

Viola Zoltán

GoboLinux

Verzió: 014-3.2

Viola Zoltán

GoboLinux

avagy

**Zárd be végleg az Ablakokat - így telepítsd első
Linuxodat!**

A GOBOLINUX 014 REJTELMEI

2008

© Viola Zoltán,
a #86-os sorszámú regisztrált, „hivatalos GoboLinux felhasználó”

Szerzői kiadás

Tartalom

Előszó	12
A könyv külalakjáról	15
A GoboLinux 014 telepítése	18
Indítás, betöltésvezérlők, s egyéb titokzatos dolgok	29
Ha nem megy az Internet	34
A hálózat leállítása és újraindítása	41
A mount parancs és az fstab állomány	43
Programok telepítése és eltávolítása	52
A GoboLinux könyvtárszerkezete	52
A magyar OpenOffice telepítése	55
A JAVA telepítése	58
Fontkészletek telepítése	59
Váltás a különböző verziójú programok között	60
Programok másolása GoboLinux disztribúciók között	61
A GoboLinux rendszereszközeinek frissítése	62
A Scripts csomag frissítése	63
A Compile csomag frissítése	64
A forrásból való telepítés lényege	65
Na de mi is az, hogy „függőség”?!	66
NumlockX	68
XBindkeys	71
Az nVidia grafikus driver telepítése	74
A 3D gyorsítás meglétének letesztelése	75
Egy példa az xorg.conf -ra	76
A magyar nyelvű Firefox	78
Betűméret a Firefoxban	82
A Firefox okosítása	82
Djvu plugin telepítése	82
Flashplayer plugin telepítése	83
Mrxvt, avagy megszabadulunk a Konsole termináltól	84
Forrógombok, aliasok, és egy konzolos zenelejátszó prog	87
A hangkártya bekonfigurálása	87
Az XBindkeys bekonfigurálása	88
Multimédiás billentyűzet speciális billentyűinek kódjai	92
Aliasok, azaz parancsok rövidítései	96
Saját forróbillentyűk definiálása a moc programnál	100
Böngészés mélykonzolból és virtuális terminálból	102
CD-írás parancssorból	104
DVD írása parancssorból	106
Fájltypusok programokhoz társítása az mc progiban	107
Nyomtató telepítése	112
A rendszer bővítése saját szkriptekkel	114
A cdfs kernelmodul és konverzió az audiállományok közt	117

MP3-WAV konverzió	118
OGG-WAV konverzió	118
WMA-MP3 konverzió	118
WAV-OGG konverzió	119
MP3-OGG konverzió	120
OGG-MP3 konverzió	121
WAV-MP3 konverzió	122
WAV-MIDI konverzió	122
MIDI-WAV konverzió	123
Az Mplayer és a filmek konvertálása-vágása	125
MPG file AVI-ba tömörítése	125
A bitráta kiszámolása	126
MPG filmek összefűzése	127
Filmek darabolása	127
AVI filmek összefűzése	127
Egy videoállomány hanganyagának leszedése	128
Hang nélküli videó készítése	129
Más hangsáv felrakása a videóra	129
DVD legreggelése (azaz parancssori DVD rip)	129
Video-CD (VCD) betömörítése	131
A GoboHide és a rejtett tartalomjegyzékek	132
Hogyan törjük fel a saját rendszerünket?	134
Áttérés a SithWM ablakkezelőre	137
Óra és dátum kijelzése a grafikus felületen	144
Az ION3 ablakkezelő indítása	145
A Ratpoison ablakkezelő	146
A Metisse ablakkezelő	147
Saját billentyűzetkiosztás definiálása	147
Egy eszperantó billentyűzetkiosztás	152
Csináljunk extra módosítóbillentyűt!	153
Hogyan készíts szerveret az otthoni számítógépedből?	156
THHTTPD	162
Recipe új verziójának elkészítése	166
Youtube-os videók letöltése	169
Magyar man-oldalak	168
Néhány hasznos konzolos okosság	171
A partíciók foglaltságának kijelzése	172
Konzolos hexa editor	173
Konzolos számológépek	174
FTP kapcsolat intézése parancssorból	175
Szótárprogi	186
A Conky rendszermonitor	175
Nagy fájlok darabolása és összefűzése	181
A CHMOD parancs	179
A mikrofon használata	181

A kill parancs, azaz hogyan gyilkoljunk programokat	182
Konzolos naptár	184
Filozofálgatunk – miért épp a GoboLinux?	189
A legjobban testreszabható Linux disztribúció	189
Az Unix-struktúra újragondolása	193
Az ötlet, amin a Compile alapul	198
Mítoszok és félreértések a GoboLinux tervezésével kapcsolatban	201
Oka van annak, hogy a dolgok olyanok, amilyenek	201
A hosszabb nevek állítólagos felhasználó-jóakarata	203
Meg akarod változtatni a szabványt?	204
Fáradtságos küzdelem, hogy megváltoztassanak minden alkalmazást	205
Könnyebb lefordítani egy programot relatíven ugyanahhoz a könyvtárszerkezethez alkalmazkodva	206
Windows-szerűvé akarod változtatni a Linuxot!	207
Egy disztribúció vizsgálata	208
Internacionalizálás	209
Integráció más disztrókkal	210
A rendszergazda	211
Következtetések	212
A regisztrált GoboLinux felhasználók	212
GoboLinux, haladóknak	214
AddUser	214
Compile	215
A Compile használata	215
Compile.conf	217
CompileProgram	218
A hálózat beállítása	219
Automatikus indítás	220
Programok automatikus indítása	220
Autostart a GoboLinux rendszer alatt	221
Az inicializálás testreszabása	221
A billentyűzet késleltetési ideje és sebessége	222
Indítórutin taszkok	222
Taszkok készítése	223
Billentyűzetkiosztás	223
Boot témák	224
BootDriver	224
Boot téma készítése	225
Boot téma konfigurálása	225
Boot téma megvalósítása	226
ThemeInit	226
ThemeFile	226
ThemeBefore és ThemeAfter	227

ThemeFinish	227
Boot téma tesztelése	228
Bootolási opciók	228
CleanModules	229
CreatePackage	229
DEB és UHU csomagok kezelése	230
DeduceName	230
Dependencies	230
DetachProgram	232
DisableProgram	233
Démonikus felhasználók	233
EDITOR variable	234
EditScript	234
FetchArchive	235
FiboSandbox	235
File konvertálása UTF-8 kódolásba	236
FilterColors	236
FilterLines	236
FindPackage	237
FindQuick	238
FixAttributes	238
FixInfo	239
FixLibtoolLa	239
Fordítás kézzel	240
Egy tipikus kézzel való fordítás menete	242
Freshen	243
GAMPPS	245
MySQL	245
OpenSSL	246
Apache (HTTPD)	246
PHP	247
Python	248
GenRecipeStore	248
GetAvailable.conf	249
GetRecipe	250
Gobo-gombok	250
GoboPath	250
Görgős egér görgőkezelésének beállítása	250
GrepQuick	251
GrepReplace	251
Gyors csomagkezelési útmutató	252
A Manager, azaz a grafikus csomagkezelő felület	254
Hashes	254
InstallPackage	255
IsExecutable	256

JAVA plugin beállítása Firefoxhoz	256
Kernel modulok betöltése induláskor	256
Kernelfoltok (kernelpatch-ek)	257
KeyManager	257
KillProcess	257
Konzol fontok	258
Környezeti változók	258
List	259
MakeRecipe	259
MergeTree	260
Más disztrók csomagjainak telepítése	260
NamingConventions	262
NewVersion	262
Pluginek	263
Pppoe	263
PrepareProgram	264
Programok elnevezése	265
RebuildLinks	266
Recipe típusok	266
Recipék konfigurálása	267
Makefile recipék	268
Perl recipék	269
Python recipék	269
Xmkmf recipék	269
Scons recipék	269
Manifest recipék	269
Egy hack	270
Recipék készítése	270
A Compile.conf beállítása	270
Régi recipék frissítése	270
Új recipe készítése	271
RemoveBroken	274
RemoveEmpty	274
RemoveProgram	275
RenameCaps	275
Rendszeróra	276
Rootless	276
Rootless (= „Root nélküli”) GoboLinux	276
A Rootless installálása	277
Követelmények	277
A Rootless uninstallálása	278
Miért jó ez nekünk?	278
Rootless installálása – részletesebb leírás	279
SMF hack	281
SandboxInstall	284

Scripts	285
Shutdown	288
SignProgram	288
StartTask	289
StopTask	289
Swap	289
SymlinkProgram	289
SystemFind	291
SystemInfo	292
TestBootTheme	292
Thunderbird	292
UpdateRecipes	293
Új kernel installálása	294
Szenzorok	295
ChrootCompile	298
Ismert hibák a GoboLinux 014-ben	300
Frissített csomagok	300
Frissített recipék	300
AM2 gond a SquashFS-sel	300
Egyéb problémák	301
Saját programokat is írhat sz...	302
Vimperator	303
Titkosított állományok és könyvtárak létrehozása	310
Saját, személyre szabott Telepítő- és LiveCD készítése	316
Hasznos linkek	326

A világ legjobb dolgai *ingyenesek*:

- **napfény...**
- **levegő...**
- **Linux...**

Előszó

Mindenekelőtt: az e könyvben leírtakat mindenki a maga felelősségére használja! A tőlem tellő legnagyobb gondossággal állítottam össze az anyagot, ennek ellenére nem vállalom semmiféle anyagi, erkölcsi, büntetőjogi, egészségügyi, pszichológiai, esztétikai vagy más felelősséget semmiért sem! (Ebből is látszik, hogy teljesen felelőtlen fickó vagyok...)

Továbbá, mindjárt itt a legelején bevallom minden szégyenkezés és pirulás nélkül, hogy szegény ember vagyok, tehát szívesen fogadok adományokat! Azaz, amennyiben valaki úgy véli, hogy könyvem az ő hasznára volt, és megengedheti magának az adakozást, és támogatni is kívánja személyemen keresztül a Szabadszoftver-mozgalmat, úgy vegye fel velem a kapcsolatot a violazoli@gmail.com címen. Szívesen fogadok pénzt, hardware-okat, de akár egy jól (JÓL!) fizetett linuxos állást is, ahol tapasztalataimat kamatoztathatnám sőt bővíthetném is...

Na ennyit a könnyes gyermekkoromról. Térjünk a tárgyra.

E könyv, ahogy a címből is kitűnik, egy olyan mű, mely szándékai szerint arra buzdít, hogy a kedves Kezdő Felhasználó (akit neveznek a számítástechnikai szakzsargonban úgy is, hogy „user”, „júzer”, sőt „láma”) végleg megszabaduljon a Windows operációs rendszertől, és végre át merjen térni a Linux remélhetőleg kizárólagos használatára. Erre nyílt utalás a címben szereplő „Ablakok”-szó – az ugyanis hogy „windows”, annyit jelent angolul, hogy „ablakok”.

Akadnak már linuxos könyvek a könyvpiacra, nem is kis számban. E mostani mégis különleges. Ez ugyanis mindenekelőtt nem (elsősorban) a már ügyis „profikhoz” szól, továbbá, a kezdőket sem úgy óhajtja tanítani, hogy elkezdje ismertetni velük a bizony több százra rúgó konzolos linuxos parancsokat, hogy szerencsétlenek azt mind bemagolják. Ezt már amiatt sem teszem, mert erre semmi szükség. Most valljuk be őszintén, ki az, aki ilyesmit SZERET tanulni?! Sőt még ha akad is ilyen ember, a legtöbben akik a winről a Linuxra áttérést fontolgatják, nem amiatt akarják ezt, mert magát a Linuxot megszerették (hogy is szerethetnék, hogy még nem is ismerik...), hanem egyszerűen menekülnek a win elől annak instabilitása, meg az ottani temérdek vírus miatt, vagy egyszerűen megunták, hogy ott mindenért fizetni kell, ha meg nem teszik hát tolvajnak érezhetik magukat. Ám szeretnék a Linuxra áttérést a lehető legcsekélyebb „munkával” megúszni!

Nos, aki a Linuxot használni óhajtja, az a tanulást semmi esetre sem úszhatja meg, ezzel tisztában kell lenni. A Linux nem Windows, és nem is óhajt azzá válni. Erős a hitem azonban abban, hogy e tanulás egyrészt cseppet sem olyan sok, mint azt számosan hiszik, s mint ahogy azt némely nagyképű

„linuxguru” is mondja, elriasztván ezzel a kezdőket, másrészt a tanulás *folyamata* is lehet élvezetes, szórakoztató szellemi kaland, sőt még humoros is! És mindenekelőtt: nem kell mindent tudni rögtön az elején ahhoz, hogy a rendszerünket a legalapabb dolgokra azonnal használhassuk, úgy mint internetezés, szövegszerkesztés, zenehallgatás.

E könyv azt a célt tűzte tehát ki magának, hogy... hogy hármat is.

1. Mindenekelőtt LÉPÉSRŐL-LÉPÉSRE, szinte dedós szájbarágással leírja, hogyan tegyük fel életünk első Linux disztribúcióját az otthoni számítógépünkre. Ezt természetesen egy bizonyos konkrét „Linux-változat” kapcsán írja le, azaz minden parancs arra vonatkozik, bár megfelelő fantáziával más „linuxokra” is vonatkozik a leírás. Ez a „Linux-változat” az úgynevezett „GoboLinux 014” nevű disztribúció. (Meg lesz magyarázva az is, mit jelent a „disztribúció”-szó).

2. Miután feltettük a rendszert amúgy „natúr” és nyersen, ez már rögtön használható, de ezután számos fejezeten át szépítjük, csinosítjuk, bekonfiguráljuk azt, megtanulunk alá programokat telepíteni és eltávolítani, meg betűkészleteket is; különböző finombeállításokat végzünk majd, s apránként megismerkedünk az úgynevezett „konzol” illetve a „virtuális terminálok” használatával, melyek nagyjából az úgynevezett „DOS parancssor” linuxos megfelelői. Nem amiatt, mintha a Linuxnak ne volna grafikus képernyője, mert VAN, de látni fogjuk majd, hogy ennek a „perverziónak”, hogy sok dolgot akkor is a „parancssorban” intézünk el (abban a ronda fekete ablakban...) amikor nem is lenne muszáj hogy épp ott, ennek értelme van, mert ott minden sokkal egyszerűbb és hatékonyabb, mint a grafikus képernyőn. Különben kiderül majd, hogy a parancssor gyakran, sőt, az esetek többségében a Linux alatt igenis grafikus... tudniillik alig van eset, hogy az „igazi” parancssort, a „mélykonzolt” kéne használnunk, mert általában (99%-ban) elég nekünk egy „virtuális terminál”, más néven „terminálemulátor” is, amit természetesen a grafikus képernyőre varázsolunk elő. Parancssoros kalandjaink közben lépésről-lépésre elsajátítjuk a „linuxos gondolkodásmódot”, mondhatni „filozófiát”. Nem száraz leckék lesznek ezek, hanem bőséges magyarázat is lesz mindenhez, habár a parancsok betűről-betűre le lesznek írva. Megismerkedünk majd bizonyos fontosabb rendszerállományokkal, ezekbe bele is turkálunk, egyáltalán, mire e könyvet elejétől végéig „végigrágja” valaki, már nem lesz kifejezetten kezdő a Linuxban. Persze mindig lesz mit tanulni: én magam már lassan fél évtizede CSAK Linuxot használok, nincs is idehaza Windowsom, s még rengeteg tanulnivalóm van. Na de ez RÁM vonatkozik – a legtöbb ember nem akar az otthoni gépéből szervert csinálni dinamikus honlapkiszolgálással, adatbáziskezeléssel, nem akar külön programnyelvet írni mint én... bár, ha a programnyelvírás nem is szerepel e könyben, az IGEN, hogy miként készíts saját szervert az otthoni gépedből.

3. A „hardcore linuxosok”, a haladók is hasznát vehetik e könyvnek: ők amiatt, mert e könyv kifejezetten épp a GoboLinuxot mutatja be, márpedig e nagyon érdekes disztribúció elég kevéssé ismert Magyarországon – könyv még nem is jelent meg róla. A könyv vége afféle nemhivatalos (és nem is teljes, bár a legfontosabb dolgok azért benne vannak...) referencia a GoboLinuxról, annak fontosabb szkriptjeiről és azok paraméterezéséről.

No és most lássuk, milyen kezdő tudást tételez fel e könyv!

- Feltételezem, hogy az Olvasó tudja a Windows alatt alapszinten kezelni a Norton Commander programot. Amiatt tételezem ezt fel, mert a Linux alatt létezik egy MC nevű program (ez a „Midnight Commander” rövidítése), és ez nagyon hasonlít a Norton Commanderre. Ezt rengetegszer fogjuk használni a Linux alatt. Ugyanis sok dolgot megkönnyít majd nekünk.
- Azt is feltételezem, hogy Windows alatt képes az Olvasó particionálni a merevlemezt. Természetesen ez megoldható Linux alatt is, semmi baj, ha úgy esik neki a telepítésnek az Olvasó, hogy nincs a lemeze particionálva. A GoboLinux telepítőlemezén ott van a GParted nevű progi, amivel ezt már a Linux indulása után elvégezheti. De én nem fogok belemélyedni abba, mi az a partició, miként kell azokat létrehozni. Ez nem kifejezetten Linux-specifikus téma, hanem az általános számítógépes ismeretek része.
- Azt is feltételezem, tudja (Windows alatt...) mi az a „parancssor”, még ha nem is használja. Tehát tudjon róla annyit, hogy „az valami ronda fekete képernyő ahol nem lehet kattintgatni az egérrel, hanem különböző fura szavakat kell beírni, amire a számítógép valamit csinál”. Ebbe beleértendő az is, hogy tudja, mi az, hogy „könyvtárrendszer”, ismerje az olyan fogalmakat, hogy „szülőkönyvtár”, „alkönyvtár”, „gyökérkönyvtár”.
- Feltételezem azt is, hogy ismeri a Firefox böngészőt. Ez működik Windows alatt is, és aki még soha életében nem használta, hanem csupán az IE-vel böngészett, annak melegen ajánlom, hogy most Win alatt kissé ismerkedjék meg vele. Linux alatt ugyanis nincs Internet Explorer, hanem helyette van nem is egy böngésző, ezek egyike a Firefox. Miután feltettük a „linuxunkat”, lesz rögtön 2 böngészőnk is, egyik a Konqueror nevű, a másik a Firefox. A Firefoxszal megismerkedhetünk már Windows alatt is, tehát tegyünk is így! Azaz, alapszintű Internetes ismereteket sem fogok közölni, és a Firefox kezelését SEM magyarázom el e könyvben, ezt az Olvasó részéről már tudottnak tételezem fel.
- Feltételezem továbbá, hogy rendelkezik az Olvasó alapszintű szövegszerkesztő ismeretekkel. Ez amiatt fontos, mert rengetegszer lesz olyasmi írva, hogy valami állományt szerkesszen (szakszóval: editáljon) a rendszerében. Ez rendszerint olyasmi lesz, hogy az MC programban az adott állományra ráállunk a kiválasztást jelző inverz csíkkal, és megnyomjuk az **F4** gombot. Ekkor az állomány szerkeszthető az MC beépített belső primitív kis szövegszerkesztőjével. Nem fogom elmagyarázni, miként kell sorokat vagy

szavakat törölni, beírni, stb. Magyar a menü, és mindez nagyon hasonló más ismert karakteres felületű szövegszerkesztők kezeléséhez. Őszintén-szólva az a véleményem, aki ennyire sem képes egy amúgy magyar menürendszerrel rendelkező program kezelését illetően, az ne akarjon Linuxra áttérni, mert úgysem sikerülhet neki: ehhez vagy túl buta, vagy túl önállótlan! Tehát ha azt írom, hogy „az EZÉSEZ nevű könyvtárban van egy NEKERESD nevű állomány, ebben keresd meg a NEMKELLIDE kezdetű sort és cseréld le arra a sorra, hogy 'EZKELLIDE 33.47.22.0' ” - akkor az Olvasó tudja azt, hogy neki most az úgymond „parancssorban” be kell ütni az **mc** karaktereket, **enter**t kell nyomnia, el tud navigálni a megfelelő könyvtárba, ki tudja választani a mondott állományt, és azt a megfelelőképp tudja szerkeszteni is az mc progi szövegszerkesztőjével (amit mint írtam az F4-el aktivizálhat). Mindezt persze csak azután, hogy elmondtam neki, miként léphet be a Linux parancssorába. Az a maga idejében el lesz mondva. De onnantól már meg kell tudja ezt csinálni.

- Feltételezem azt is, hogy Windows alatt tud CD-t írni valami akármiféle CD-író programmal. Ez multhatatlanul szükséges, ugyanis a „Linuxot” úgynevezett „CD-képmás” formában töltheti le az „internetről”, amit csak úgy tud használni, hogy előbb CD-re menti, majd „bebútol” róla. Emiatt tudnia kell azt is, hogyan tud „bootolható” („butulható”, hehe...) CD-eket írnia. Azaz a letöltött, „**.iso**” kiterjesztésű fájlt nem közönséges adatként kell kiírnia a CD-re, hanem „CD-képmásként”, ami a különböző CD-író szoftverek menüiben olyan neveken szerepelhet mint funkció, hogy „CD-képmás írása”, vagy „CD-kép írása”, „Képmás-fájl írása”, esetleg angolul: „Burning CD-image”.
- Azt is feltételezem, be tudja állítani a saját számítógépe BIOS-át úgy, hogy a gépe CD-ről induljon, s ne a merevlemezről. Ha erre nem képes, reménytelen kísérleteznie a Linuxszal, attól tartok.

Ennyi tudást tételezek tehát fel az Olvasóról, nem többet. Nem hiszem, hogy ez valami hallatlan igény volna.

És most lássuk az első fejezetet, amit rögtön azzal kezdünk, hogy feltelepítjük a „mi” Linuxunkat! Ezután jön csak némi „általános duma”, hogy miért épp ezt a „linuxot”, azaz ezt a disztribúciót választottam, egyáltalán mi az hogy disztribúció, meg mindenféle egyéb infók. Azért ez a sorrend, mert eddigre legalább fel lesz telepítve a rendszer, s így lesz az Olvasónak egy jó adag sikerélménye! Előtte azonban még pár infó...

A könyv külalakjáról

Nem sokat törődtem a szöveg úgymond „szépségével” és eleganciájával. Az elegancia az úriszabók és divatdiktátorok dolga. Azt amit nekünk kell parancsként bepötyögni, általában **félkövér** betűkkel kiemeltem. Gyakran

azonban a közölt konzol output szövegeket csúnya módon méretcsökkentettem, igen picire. Ennek oka, hogy így a könyv rövidebb és olcsóbb lehetett, még inkább azonban mert így a hosszú sorok nem törnek több sorba a könyv lapjain, ami megtéveszthetné az Olvasót, aki úgy vélhetné hogy az nem egy parancs hanem kettő, s így hibásan ütné azt be.

Továbbá: sok úgynevezett „konzol outputot” idézek a könyvben, illusztrálásul annak, mit láthatsz a képernyőn az egyes parancsok lefutása során. Ezt ugyanolyan fix szélességű betűtípussal szedem, mint amit e bekezdésben láthatsz, kedves Olvasóm! E konzol-szövegekből azt, amit neked kell „bepötyögnöd”, azt általában szintén félkövér betűkkel emeltem ki, **ilyenekkel, mint amit most olvasol!**

Így egy kezdő szerintem jobban eligazodik a könyvön.

Na de tényleg, ha ez kezdőknek való könyv, miért nem a sokkal könnyebb grafikus felületet ismertetem?! - kérdezhetné valaki.

Amiatt nem, mert az úgyis magától értetődő. Hamar rájön arra mindenki magától is. A GoboLinux az úgynevezett KDE ablakkezelővel indul (alapértelmezés szerint), s az nagyonis Windows-szerű. Egyáltalán nem magától értetődő azonban sok esetben a konzol kezelése, márpedig e könyv célja HASZNOS információk adása. De azért is így cselekszem, mert szándékaim szerint e könyv afféle „Bevezetés a Kockásodásba”-tanfolyam lenne. Azaz szeretném, ha e könyv elolvasása után az Olvasó úgy érezné, ő külön (számítástechnikai téren), mint a számítógépfelhasználók többsége. Ő ugyanis nemcsak *használja*, de *érti* is a számítógépének (operációs rendszerének) működését, legalábbis sokkal jobban érti, mint a legtöbben! Nos, ez el sem képzelhető a parancssor legalább alapszintű ismerete nélkül.

Továbbá: a GoboLinux egyik legnagyobb előnye, hogy „jól hackelhető” (hekkelhető). Azaz, tetszőlegesen személyre szabható. Mindent átállíthatsz benne. Azoknak való, akik szeretik az ilyesmit, szeretnek „turkálni” a rendszerben, szeretik, ha urai a gépüknek. Ehhez is kell a parancssor.

E mű első megjelenése óta néhány dolog megváltozott. Például kijött a **GoboLinux 014.01** verziója. Ennek nyilván más a közvetlen letöltési linkje, mert más az állományneve. És más opciókkal van forgatja a „kernele”, rendszermagja, például valami „paravirtualizációt” elősegítő „XEN” opcióval, emiatt nem települ fel rá az nVidia driver. Más dolgok azonban amiket leírtam e könyvben, működnek alatta. Továbbá, megszűnt honlapom, a <http://linux.birodalom.net> és emiatt az oda korábban feltett programcsomagok nem találhatók már meg ott. Ellenben ezeket beszerezheti a Tisztelt Olvasó a

<http://gobolinux.cwi.hu> oldalról, illetve részben a

http://www.gobolinux.info/index.php?option=com_remository&Itemid=61&func=select&id=3

oldalról. Bevallom azonban becsületesen, túl lusta voltam ahhoz, hogy e könyvben mindenütt kijavítsam a linkeket az új linkekre... ez tehát bizony rád

vár, Tisztelt Olvasó, ez, hogy megkeresd a megfelelő csomagokat! Időmet inkább arra fordítottam, hogy további trükkökkel és leírásokkal bővítsem a könyvet.

Lépj tehát az első fejezetre, tisztelt Olvasóm!

A GoboLinux 014 telepítése

Mindenekelőtt be kell szereznünk a GoboLinuxot. Mint a „Linuxok” általában, természetesen „ő” is ingyenes. Annyi kell csak, hogy el kell caplatnunk az Internet kies, virtuális országútjain a <http://gobolinux.org/> webcímre, ami a GoboLinuxot készítő baráti kör honlapja. No most ez elvileg magyarul kéne hogy bejöjjön az odatévedőnek, habár ez nem magyar készítésű disztró, mert Brazíliában fejlesztik, s a fejlesztők többnyire angolul társalognak egymással. Nem kell megijedni, a feltelepített rendszer magyarítva van már... Mindenesetre, kezdetben a honlapjuk cseppet sem volt magyarul elérhető. Ez mára már megváltozott, ugyanis lefordítottam a magyar anyanyelvűek kedvéért. Ha mégsem magyarul jelenne meg, akkor kattints az oldal fejlécében található „Magyar” felíratra, vagy rögvest e linket kövesd:

<http://gobolinux.org/index.php?lang=hu> HU

Az oldalon nyugodtan elidőzhatsz, találsz sok érdekes linuxos cikket. No de te le akarod tölteni a „Gobót”... Ezt a „Letöltések” oldalon teheted meg, aminek közvetlen linkje ez:

<http://gobolinux.org/index.php?page=downloads>

Ezen az oldalon sok linket találsz, különböző országokból, valamelyikről le kell töltsd a **GoboLinux-014-i686.iso** nevű állományt. Ezt akár torrenttel is megteheted. Egy konkrét link az állományra a sok közül:

<http://gobo.calica.com/gobolinux-iso/GoboLinux-014-i686.iso>

Miután ez egy teljes CD-nyi, azaz 699 megabájtos anyag, célszerű olyan letöltővel töltened, mely képes az esetleg megszakadó letöltések folytatására. Mindezt természetesen még valószínűleg Windows alatt végzed. Ez ne aggasszon, ez nem szégyen. Én is „egységsugarú Win-user” voltam valamikor régen. Ez szerencsére gyógyítható betegség, amit erős akarattal le lehet küzdeni. És könnyebb, mint a dohányzást abbahagyni.

Nos, tegyük fel letöltötted. Most írd ki CD-re, de CD-képmás formátumban! Állítsd be a számítógéped BIOS-át úgy, hogy CD-ről „butuljon”, és indítsd el így az imént kiírt GoboLinux rendszert!

Tudnod kell, hogy ez egy úgynevezett „LiveCD”, azaz közvetlenül erről működik most az operációs rendszer. SEMMIT nem ír a merevlemezre, természetesen csak addig, amíg el nem indítod a telepítést! De mert MÉG nincs az oprendszer a merevlemezeden, emiatt mindent a CD-ről olvas, s így ez elég lassú. De ez se aggasszon téged, ez nem azt jelenti, hogy „a Linux szar”, mert a feltelepített rendszered természetesen már gyors lesz! Sőt, „hasít mint az atom” majd *bizonyos egyéb* oprendszerekhez képest!

Most amíg töltődik a rendszer, különböző furcsa kiírásokat látsz a cseppet sem grafikus képernyőn. Várj türelemmel. Egy idő után meg kell jelenjen egy olyan felirat, hogy

Choose your language

No most azt ígértük, hogy a rendszer magyar lesz... hát ez mintha nem lenne olyan! Nos igen, de csak amiatt, mert szegény GoboLinux még nem tudja, hogy ő épp egy magyar illető gépére készül felmászni! Ellentétben azzal amit sokan hisznek, ez a szöveg nem azt jelenti, hogy „Válaszd ki a langusztádat”, a Linuxnak ugyanis nem a languszt, hanem a pingvin a jelképe. Ez azt jelenti, hogy „Válaszd ki a nyelvet”. Nos, neked a kurzorbillentyűk segítségével ki kell választanod a

hu_HU Magyar

sort, majd az **Enter** billentyűt megnyomni. Innentől tudja a masina, hogy te a hírs-vezetős Magyar néphez tartozol, s innentől már magyarul próbál majd társalogni veled.

No most mi van akkor, ha valamiért nem jelenik meg ez a „langusztás” szöveg, hanem vagy semmi, vagy egy rakás fura szöveg, amiben sűrűn szerepel az „**error**” szó, ami angolul hibát jelent?!

Nos, akkor marha nagy gáz van. Az azt jelenti, hogy a számítógéped „vasa”, azaz hardware-konfigurációja nem támogatott e Linux-disztribúció „kernele”, tehát „rendszermagja” által. Ekkor sajnos csak 2 dolgot tehetsz:

1. Veszelsz egy másik számítógépet, de úgy, hogy megvásárlása előtt a boltban elindítod e Linux-CD-t, hogy működik-e benne. Ha eljutsz e „langusztás” sorig, valószínűleg jó lesz a számodra. Mindenesetre jó tudnod, hogy e CD olyan rendszert tartalmaz, mely az úgynevezett „Intel” processzoros gépeket támogatja, mással tehát ne nagyon próbálkozz. Ugyan más architektúrákon is lehet életet lehelni a Linux-ba, még speciálisan a GoboLinuxba is, de az nagyon nem kezdőknek való feladat.
2. Azt is megteheted, hogy keresel egy másik Linux disztribúciót tartalmazó telepítő CD-t. Ezen esetben ennek a könyvnek sajnos nem sok hasznát fogod venni.

Ez a könyv természetesen az én saját otthoni számítógépemen lett megírva, minden egyes parancsa kipróbálva hogy működik-e, de másnak lehet más összeállítású gépe is. Ideírom, hogy kifejezetten az én számítógépem felépítése milyen „csipekből” áll. Ezt szerencsére nem kell fejből tudnom, mert Linux alatt ezt mind kiírja nekem az

lspci

parancs, melyet természetesen „konzolban” kell kiadni. Íme a kimenete:

```
vz@Csiszilla /Depot/Letolt]lspci
```

```
00:00.0 Host bridge: Intel Corporation 82865G/PE/P DRAM Controller/Host-Hub Interface (rev 02)
```

```
00:01.0 PCI bridge: Intel Corporation 82865G/PE/P PCI to AGP Controller (rev 02)
```

```

00:1d.0 USB Controller: Intel Corporation 82801EB/ER (ICH5/ICH5R) USB UHCI Controller #1
(rev 02)
00:1d.1 USB Controller: Intel Corporation 82801EB/ER (ICH5/ICH5R) USB UHCI Controller #2
(rev 02)
00:1d.2 USB Controller: Intel Corporation 82801EB/ER (ICH5/ICH5R) USB UHCI Controller #3
(rev 02)
00:1d.3 USB Controller: Intel Corporation 82801EB/ER (ICH5/ICH5R) USB UHCI Controller #4
(rev 02)
00:1d.7 USB Controller: Intel Corporation 82801EB/ER (ICH5/ICH5R) USB2 EHCI Controller (rev
02)
00:1e.0 PCI bridge: Intel Corporation 82801 PCI Bridge (rev c2)
00:1f.0 ISA bridge: Intel Corporation 82801EB/ER (ICH5/ICH5R) LPC Interface Bridge (rev 02)
00:1f.1 IDE interface: Intel Corporation 82801EB/ER (ICH5/ICH5R) IDE Controller (rev 02)
00:1f.2 IDE interface: Intel Corporation 82801EB (ICH5) SATA Controller (rev 02)
00:1f.3 SMBus: Intel Corporation 82801EB/ER (ICH5/ICH5R) SMBus Controller (rev 02)
00:1f.5 Multimedia audio controller: Intel Corporation 82801EB/ER (ICH5/ICH5R) AC'97 Audio
Controller (rev 02)
01:00.0 VGA compatible controller: nVidia Corporation NV34 [GeForce FX 5200] (rev a1)
02:06.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL-8139/8139C/8139C+ (rev 10)

```

Tehát az e könyvben leírtak ezen az architektúrán/konfiguráción lettek tesztelve. Emellé megjegyzem, a gépemben 512 mega RAM van - bár a GoboLinux működik 256 megával is. Sőt! **A minimális hardware-követelmény mindössze Pentium-II számítógép, 128 megabyte RAM-mal!**

A fenti lista első sora:

```
vz@Csiszilla /Depot/Letolt]lspci
```

azt mutatja, hogy én mint a „vz” nevű felhasználó vagyok bejelentkezve a rendszerbe, a számítógépemnek a „Csiszilla” nevet adtam, és a „Depot” nevű könyvtáram „Letolt” alkönyvtárában adtam ki az „lspci” parancsot. Már láthatod ebből tisztelt Olvasóm, hogy a Windowssal ellentétben a Linux alatt a könyvtárakat elválasztó szeparátorjel nem az a hülye „fordítottperjel”, angolul „backslash”, azaz ez az idétlenség: „\” amit a magyar billentyűzeten az

AltGR+Q

billentyűkombinációval lehet előcsalogatni, hanem a közönséges „per”-jel. Ezt ajánlatos megjegyezni örök időkre...

Ha már az „error” szóbakerült: a képernyő alján egy olyan feliratot láthatunk esetleg, hogy

```
end_request: I/O error, dev fd0,sector 0
```

Nos, ettől NE IJEDJÜNK MEG! Ez a floppy egységre vonatkozik, ha van olyan a gépünkben (nekem van). Linux alatt a floppy drive szokásos jele az „fd”. Ez annyit jelent, hogy valamiért nem képes most lekezelni a floppy meghajtónkat a rendszer. Ez nem baj, mert úgysem akarunk most vele foglalkozni. A feltelepített rendszer természetesen képes lesz már a floppyt is kezelni. Ez egy szépséghiba, kétségtelen, de a disztró készítői egyelőre nem foglalkoztak vele, mert telepítés közben úgysem akar senki sem floppykat dugdosni a gépbe. Az efféle hülyeség csak a „Má\$ik Oprendszer” szokása.

Na de térjünk már vissza szegény langusztáinkhoz... miután kiválasztottuk a sok langusztá közül a magyarszkit, meg kell jelenjen egy olyan felirat, hogy

ÜDVÖZÖL A GOBOLINUX

Itt egy hosszabb szöveg is látható, melynek lényege, hogy választanod kell, a telepítés további részét grafikus vagy konzolos(=parancssoros) felületen hajtod-e végre. Bár e könyv hátralevő részében igen sűrűn mélyedünk majd el a konzol fekete perverzítésében, most még nem sokkoljuk ezzel magunkat – a grafikus telepítés mellett döntünk. Ehhez be kell pötyögnünk a

startx

parancsot (és a végén az Enter billentyűt megnyomni. Ezt most írtam le utoljára, tessék megjegyezni, hogy minden parancs az Enterrel küldendő el Őfelségének, Gobo bácsinak, aki az oprendszer a gépünkön!)

No most ezen a ponton egy normális szakkönyvben illene hogy szép színes képecskék legyenek, bemutatván, mit lát a kedves Felhasználó. Ez elmarad. Ebben a könyvben egyetlen kép sem lesz. Ennek az az oka, hogy így olcsóbb. Ez a mai árak mellett nem elhanyagolható. Továbbá, mert mi a csudának kép, amikor gyakorlatilag mindig a parancssorból oldjuk majd meg a dolgokat, ott meg nincs kép, csak szöveg. Szövegeket természetesen bőven idézek majd e könyvben.

Mindenesetre most már egy grafikus képernyőt látunk, bár igazán nem sok ikonnal. Linuxunk lassan egész állatsereglet lesz, mert az iménti languszták mellé felsorakozott még egy állat: van mindenki öröme már egerünk is! Nosza ne is húzzuk az időt, kattintsunk arra az ikonra, mely szándékai szerint egy eszkimó ruhába öltözött kölyökpingvint ábrázolna – a felirata az, hogy „Install GoboLinux”. Természetesen jó Windowsos szokás szerint gyorsan kétszer kell kattintani rá.

Megjegyzendő, hogy ha eddig nem particionáltuk volna a merevlemezünket, akkor ne erre az ikonra kattintsunk, hanem a **GParted** nevűre, amivel egy grafikus particionáló progit indítunk, és csak ha ezzel megvagyunk, azután indítsuk a telepítőt a fent említett módon. Bár mint a bevezetőben ígértem, nem fogom részletezni a particionálás menetét, de annyit elárulok, milyen MÉRETŰ partíciókat kell létrehoznunk! Nos, egy „fullos” telepítés a CD-ről, minden progival, az elfér mindössze 3 gigabájton. Mégis azt ajánlom, hogy 10 gigásnál kisebb méretűre ne nagyon rakjuk rá GoboLinuxunkat, s ennek az az oka, hogy ez egy forrásalapú disztró – hogy ez mit jelent azt a következő fejezetben magyarázom el – és emiatt *később*, amikor mindenféle olyan progival akarjuk bővíteni a rendszert ami most a CD-n nincs, bajba kerülhetünk, mert sok hely kell majd a forrásprogramoknak, meg hely kell

ezek lefordításához is! Plusz hely kell a saját adatainknak is. Én eredetileg egy 10 gigásra raktam a rendszert, még e GoboLinux előző, 013-as változatát, de idővel kicsi lett, s most ezen újabbat egy 20 gigás partícióra raktam.

Tehát kell egy legalább 9-10 gigás partíció, ahová telepítjük, s kell egy úgynevezett „swap” partíció. A swap szó ismerős lehet a Windowsban járatosabbaknak – a win *swap fájl*t használ. Linux alatt ez egy fix partíció, ami lényegesen gyorsabb, mintha fájl használna. E swap partíció legjobb ha nincs megformázva semmiféle fájlrendszerre sem. A mérete legyen úgy nagyjából 512 mega, de 1 gigánál ne nagyobb, mert úgysem használja ki. Különbö a Linux többnyire elboldogul 512 mega RAM mellett swap nélkül is – én eredetileg 630 megát akartam adni az én rendszeremben swapnak, de elírtam és 63 lett belőle. Most meg lusta vagyok „átcsinálni”. És nem is szükséges, mert alig nyúl hozzá ehhez is...

Ezen kívül melegen ajánlom, hogy legyen legalább még egy partíciónk a Linuxunk számára, célszerűen ugyanakkora mint az, ahová fel fogjuk rakni. Illetve ha nagyobb, az nem baj. Ennek célja egy későbbi fejezetben lesz elmagyarázva, lényeg az, hogy ide készítünk majd úgymond „biztonsági másolatot” a rendszerünkről.

No most e könyv jelölismódjai: a továbbiakban feltételezem, hogy **rendszerünket** az úgynevezett **HDA4** nevű partícióra telepítjük, a **swap** partíció a **HDA3**, az a bizonyos másik pedig lesz mondjuk a **HDA7**. Ha nálad Olvasóm nem ezek a partíciók számai, akkor sincs baj, csak ha ilyesmit látsz nálam a parancsokban, értelemszerűen cseréld ki a jelölésekben a számokat. (Ezen partíciónevekben a „HD” arra utal, hogy „hard disk”, azaz merevlemez. Már például pendrive-ok és MP3 lejátszók esetén a név többnyire „SD”-vel szokott kezdődni).

Hogy miért nem a **HDA1**-re telepítjük a Linuxot? Nos amiatt, mert a HDA1-en nagy valószínűséggel egy Windows „csücsül”. Amit esetleg (nagyon helytelenül...) nem szeretne letörölni még a kedves Felhasználó, azaz Te, Tisztelt Olvasóm! No most, én a wint utálom mint a bűnt, s nem fogok semmi különösebb segítséget megadni ahhoz, hogy a Linuxot a win mellé telepítse valaki, mert mély meggyőződése, hogy az a legjobb, ha azonnal megszakit minden „közösséget” a „Nagy Kék” oprendszerrel. „Bizony mondom néktek feleim, jobb a Felhasználónak a Windowst nem ismernie” - hogy kissé „biblikusan” fogalmazzak. Mindazonáltal, ha nem arra a partícióra telepíti valaki a Linuxot, amelyen a Windows van, akkor a Windows (sajnos...) nagy valószínűséggel meg fog menekülni, a Linux ugyanis általában van olyan okos, hogy ezesetben úgy telepítse magát, hogy bekapcsoláskor egy pici kis butuska progí, a „Grub” nevű induljon el, aminek menüjéből ki lehet választani, épp mi induljon: a Windows vagy a Linux. Na de ehhez az kell, hogy a Linux ne a Windows helyére legyen telepítve. Bár nekem nincs idehaza Windowsom, de

gondolván a Windowst megőrizni kívánó perverzebb (vagy épp óvatosabb...) felhasználók igényeire, e leírásban feltételezem, hogy a leendő Linuxunk telepítési helye a **HDA4**.

Na tehát ott tartottunk, hogy már egészen elállatiasodtunk, mert van sok állatunk: langusztáink, egerünk, kölyköpingvinünk... s utóbbira rákattintva megjelenik egy magyar nyelvű szövegecske egy kis ablakban (ami azonban teljesen win-szerű módon kinagyítható), hogy ez a GoboLinux telepítője, stb-stb, tessék végigolvasni. Különben semmi fura vagy különleges nincs a szövegben. Ezután lent a „Next” feliratú gombra kell kattintanunk. Ezt hogy mire kell kattintanunk, ezentúl a rövidség kedvéért így fogom jelezni:

=>Next

Most ki kell válasszuk a megjelenő képernyőről, hogy melyik partícióra fogjuk telepíteni a Linuxunkat. Jelöljük be a HDA4-et. (Vagy amelyiket óhajtod). Tegyük egy **x**-et az „**A partíció formázása**” szöveg elé, azzal, hogy odakattintunk. Erre megjelenik egy figyelmeztető ablak, hogy minden adat el fog veszni e partícióról. Semmi gond, ezt tudjuk úgyis! Okvetlenül meg kell formáztatnunk a partíciót akkor, ha Windows alatt csináltuk, mert majdnem biztos, hogy a Windows nekünk NTFS fájlrendszerre formázta meg vagy FAT-ra esetleg, és nekünk, azaz a Linuxnak, egyik sem jó. Most ki kell választani, milyen típusúra legyen formázva a filerendszer, választhatunk a **ReiserFS**, az **ext2** és **ext3** között. Nem baj, ha nem tudod mit jelentenek e szavak, lényeg, hogy nekünk ezek bármelyike jó. Régen a Linuxok ext2-re települtek, aztán jött az ext3, de most a „bölcsek” a ReiserFS-t ajánlják. Nekem is ez van idehaza mindegyik partíciómon, és semmi bajom vele. Fogalmam sincs, miért jobb (állítólag), mint az ext3, de hiszek a nálam okosabbaknak. Valószínűleg a GoboLinux is eleve ezt ajánlja fel neked. Fogadd el, vagy ha nincs beixelve, tedd azt.

=>Next

Most jön a „Válogatás a csomagokból”. A telepítő felajánl egy „tipikus” csomaglistát, hogy ezeket telepítené. Elvileg beleegyezhetnél, de melegen ajánlom, jelöld be azt, hogy „full”. Erre rögvest ki lesz jelölve az összes csomag. No most ebből te törölgethetnél, kiszedve a pipákat a neked nem kellőek előtt, de mert most még halvány fogalmad sincs, melyik csomag mire való – némelyikről én magam sem tudom – szerények leszünk. Csak néhány mellől szedjük ki a jelet, amiről tutibiztosan tudjuk, hogy nekünk nem kell. Keresd meg például a LILO nevű progit, s ezelől szedd ki. Ez nekünk egészen biztosan nem fog kelleni. Ez egy úgynevezett „betöltésvezérlő”, hasonló feladatot lát el tehát, mint a Grub, amiről fentebb szoltam pár szót. Minthogy mi Grubot telepítünk majd (mert én azt szeretem), emiatt a LILO-ra semmi szükségünk, csak foglalná a helyet. Egyben szedjük ki a jelet a

KDE-I18N-de

és a

KDE-I18N-pt_BR

csomagok elől, mert ezek sem fognak kelleni. Ezek a KDE ablakkezelő program (egészen precízen nem ablakkezelő, hanem „desktop environment”, magyarul „munkaasztali környezet”) német illetve „brazíliai portugál” nyelvi fájljai. Minthogy merőben valószínűtlen, hogy ne magyarul szeretnéd használni a rendszeredet, ezekre semmi szükséged. (Ha mégis akkor bocsika, és hagyd meg a pipát). Ellenben ellenőrizd, nem törölted-e netán a

KDE-I18N-hu

progi elől a jelölést, ha ugyanis azt is törlöd, a rendszered nem fog tudni magyarul, és kénytelen lesz angolul értekezni vele, ami elég kínos lehet, ha nem ismered azt a nyelvet.

=>Next

A betöltésvezérlő (Bootloader) telepítése. Nos, most két eset van. Ha van már valamilyen Linux rendszer telepítve a gépedre netán, akkor szedjük ki az **x-et** a **„Betöltésvezérlő (bootloader) telepítése”** sor elől! Ugyanis a régebben felrakott Linux rendszer úgyis felrakott neked már valamilyen betöltésvezérlőt. Ebbe majd te valamiképp beleírod a GoboLinuxot is. Le fogom írni a megfelelő helyen, hogyan. Ha viszont még semmilyen Linux rendszer nincs a gépeden – s ez a valószínűbb eset! - akkor ellenőrizd, hogy igenis **LEGYEN** bejelölve a fent említett sor!

=>Next

Jönnek az **„Általános beállítások”**.

Gépnév: itt felajánlja, hogy **„NewMachine”**. Ez elég hülyén hangzik, írd be helyette, amilyen nevet adsz a te masinádnak. Én a **„Csiszilla”** nevet adtam neki, mert nekem ez tetszik. Használhatsz bármilyen nevet, kis- és nagybetűket vegyesen is. Olyan nevet írsz be, amilyent csak akarsz, te találd ki. Különleges karaktereket azonban ne használj a névben. Azaz lehetőleg ne az legyen a masinád neve, hogy **<(:-)]=**

Ezen a képernyőn még olyasmiket állíthatsz be, hogy **billentyűzetkiosztás** – de itt már a masina tudja, hogy te magyar vagy, ezt állítja be amúgy is. A betöltésvezérlő kinézete, a BIOS órájának beállítása – ezt mind jobb, ha nem piszkálsz, mert felesleges.

=>Next

Időzóna – felajánlja, hogy „Europe/Budapest”. Ha Magyarországon élsz, ez neked jó, ezt sem kell piszkálnod.

=>Next

A hálózati interfész beállítása: eth0

Ha „Ethernet” csatlakozón át mászkálsz az Internetre, és csak egy hálókártya van a gépedben (ez majdnem mindenkinél így van), akkor ez neked jó. No most ha azt akarod, hogy indulás után automatikusan legyen lehetőség az Internetre rácsatlakozni, például böngészni, anélkül, hogy ennek érdekében különböző varázsszavakat kéne beütögetned a konzolba, akkor tégy róla, hogy legyen bejelölve az **„Az interfész engedélyezése induláskor”** sor. Többnyire ugyanis azt akarjuk, hogy a saját gépünkön ne legyen letiltva önmagunk elől az Internet...

A következő sorok fontosak! Arról van szó, hogy neked úgynevezett **„Fix IP-címed”** van-e, vagy **„dinamikus”**. Utóbbi azt jelenti, hogy a gép minden ki-be kapcsolásánál megváltozik (legalábbis megváltozHAT) az IP címed, mert hogy épp milyen IP-t kapsz, azt akkor közli veled (pontosabban a gépeddel) az internetszolgáltatód szervergépe. Ennek számos előnye van, például nehezen tudnak IP alapján kitiltani valami fórumról. Én mégis jobban szeretem a fix IP címet, mert ennek alapján könnyen lehet akár saját szervert is készítened az otthoni gépeden, amire felrakhatsz valami honlapot... Mindenesetre, azt hogy az IP címed fix-e vagy dinamikus, neked kell tudnod, mert ezt a szerződéskötéskor a Szolgáltató közli veled. Ez már a Windows hálózati bekonfigurálásánál is fontos volt, azaz tudnod kell rég. Nos, ha dinamikus IP címed van, akkor azt kell bejelölnöd, hogy „Automatikus konfigurálás DHCP segítségével”. Ezután =>Next. Ez azt mondja a masinádnak, hogy el kell indítson minden induláskor egy „dhcp” nevű úgynevezett „démont”, azaz háttérben futó progit, ami elég okos ahhoz, hogy felderítse, most épp milyen IP címet kapsz a szervertől, s ennek megfelelően működjön.

No most, ezzel az esettel nem is foglalkozom a könyv hátralevő részében, egyszerűen mert NEKEM fix IP címem van, így nem tudom a dhcp-változatot letesztelni. De gyanúm szerint úgyis az az egyszerűbb eset, mert a fent említett „démon” mindent el kell intézzen automatikusan amúgy is, hiszen ez a dolga, ezért van, erre lett kitalálva, azaz azzal a továbbiakban (elvileg) nem kell hogy bármit is törődjünk.

Ha fix IP címed van, akkor NE legyen bejelölve a fenti DHCP-vel kapcsolatos sor, hanem írjuk be a címet pontokkal elválasztva az **IP:** - kezdetű sorba, mondjuk így:

82.144.172.154

Adjuk meg az úgynevezett „Netmaszk”-ot is. Ez általában 255.255.255.0 szokott lenni, de te kellett kapj a Szolgáltatótól ügyis leírást, hogy ez mennyi, na azt kell beírnod. Egyáltalán, ha a Windowsodnak volt internethozzáférése, akkor minden ilyen adatot kell tudj.

Van még ott egy „Broadcast” feliratú sor is, no az maradhat üresen.

=>Next

Átjáró IP – ezt nevezik úgy is, hogy „Alapértelmezett átjáró”, angolul „Default Gateway”, nos, itt felajánlja a gép, hogy „192.168.1.1”. Ennek helyére természetesen azt kell írnod, ami nálad a jó szám. Megjegyzem, az idevaló érték többnyire kitalálható az IP címből, mert azzal megegyezik, csak a legkisebb helyiértékű bájtt értéke 1-es, azaz a fenti példa szerint majdnem biztos, hogy ez itt 82.144.172.1 lenne. Ez majdnem mindig így van. De csak majdnem...

Most állítsd be a Szolgáltatótól kapott leírás szerint az elsődleges és másodlagos DNS-szerverek címét is.

=>Next

Következik a gép rendszergazdájának a beállítása. Az első sor úgy kezdődik, hogy

Bejelentkezés: - no és itt felajánlja, hogy a rendszergazda neve az legyen, hogy „**root**”.

Ez, hogy a rendszergazda neve „root”, egy hagyomány az Unix-rendszerekben, s a Linux is átvette. A név onnan származik, hogy angolul azt jelenti, hogy „gyökér”, mert mint a gyökérből nő ki a fa törzse s az ágai, úgy minden jog(osultság) a többfelhasználós számítógéprendszereknél e rendszergazdától, a „gyökérfelhasználótól” származik, mert ő határozhatja ezeket meg.

Nos nem tudom ki hogy van vele, de nekem nagyon nem tetszik a „root” név, már amiatt sem, mert nem magyar, s azt sem tudom, hogyan ejtsem ki: „rót”?! „Rút”?! A „rút” talán a helyes, de kikérem magamnak, hogy engem valaki rútnak tartson, akár a komputerem is, márpedig idehaza nyilván én leszek önmagam rendszergazdája!

Nos, a legtöbb „Linux” nem ad lehetőséget rá, hogy a rendszergazda neve más legyen, de a GoboLinux e téren kivétel. Írjuk át valami más, nekünk tetsző

névre a „root” szót. Javaslom, hogy kerüljük a nagybetűket és a számokat, az ékezetes karaktereket pedig pláne. A továbbiakban feltételezem, hogy ide a „**pv**” nevet állítottuk be. Megjegyzem, hogy biztonságtechnikai szempontból is előnyös, ha a rendszergazda neve nem a megszokott „root”, mert képzeljük csak el, hogy valaki a távolból fel próbálja törni a gépünket, azaz kitalálni a rendszergazdai jelszót. Nos, ez akkor sem sikerülhet neki, ha a jelszót magát megfejtené netán. Az ugyanis mit sem ér a felhasználó neve nélkül. Ő pedig feltételezi, hogy a rendszergazda neve a gépünkön a megszokott „root”. Holott nincs olyan felhasználó nálunk, hehehe...

Most be kell ütnünk a rendszergazdai jelszót kétszer. Jól fontoljuk meg, hogy mit ütünk be, mert nehogy elfelejtsük! Jó ha tudjuk, hogy a Linux különbséget tesz a kis- és nagybetűk közt, azaz teljesen külön jelszavaknak tartja az „eger”, „Eger”, „EGER”, „eGER” karaktersorozatokat! De ez csak példa volt, mert neveket ne adjunk jelszónak. Pláne ne négybetűset, mert az túl rövid! Az én jelszavam például... de nem, inkább mégsem írom le!

=>Next

Most következnek az úgynevezett „egyéb felhasználók”. Mert egy már van ugye, ő a rendszergazda, de nagyon nem ajánlatos rendszergazdaként bejelentkezve nemhogy internetezni, de egyáltalán bármit csinálni is. Ennek számos oka van, lényegében mindegyik ilyen ok a biztonsággal függ össze. Hogy mást már ne is mondjak, például tévedésből elindítasz valami progit, amit jobb lenne mégsem futtatni. Az efféle progik (a kockázatosak...) többnyire csak rendszergazdai jogkörrel futtathatóak. Azaz ha nem rendszergazdai jogban vagy, nincs baj, mert el sem indulnak neked. De ha mindig rendszergizdaként dolgozol... érted, ugye!

A lényeg tehát az, hogy létre kell hozni legalább egy db lekorlátozott jogú felhasználót, annak a nevében ténykedünk általában, ha tehát internetezünk, filmet nézünk stb, s rencergizdaként csak akkor gizdáskodunk, ha már végképp nagyon muszáj. No ezt a felhasználót vesszük fel most a felhasználói jegyzékbe. Adjuk meg a nevét, jelszavát, teljes nevét, majd kattintsunk a „Felhasználó hozzáadása” gombra. És vigyázzunk, hogy NE legyen bejelölve nála a „Rendszergazdai jogosultságok engedélyezése” sor!

Eképp itt még akárhány másik felhasználót is felvehetnénk a rendszerbe, de átlagos, tipikus esetben egy elég. A továbbiakban feltételezem e leírásban, hogy e „normál”, nem rendszergazdai felhasználó neve „**vz**”. Aki az Emberiség azon teljesen jelentéktelen hányadához tartozik, akiknek mégsem tetszik e név, az értelemszerűen vegye ezt figyelembe a leírás olvasása során. Különben nem gyakran kell majd ügyelnie erre.

=>Next

Most meg kell jelenjék a képernyő tetején a felirat:

Készen állok az installálásra

Bátran katteljünk a „**Finish**” gombra.

Erre megjelenik egy ablak a képernyő közepén, s látni fogjuk, ahogy átmásolja a különböző programokat a CD-ről a merevlemezünkre a gép. Ez eltart majd pár percig. Végül ezt írja ki majd:

Most indítsd újra a számítógépedet, és örvendj a GoboLinuxodnak!

Na most az csak a baj, hogy szabályosan nemigen tudjuk a gépet újraindítani, lévén hogy a CD-ről megy az oprendszer jelenleg, s emiatt nem hagyja, hogy a CD-t kivegyük a helyéről! (Tulajdonképpen persze lenne módszer rá, hogy szabályosan indítsuk újra a masinát, de ahhoz el kéne magyarázni egy csomó mindent Neked, kedves Kezdő Linuxos, s úgy ítélem meg, hogy annak még nem most jött el az ideje). Leghelyesebb, ha egyszerűen megnyomjuk a **Reset** gombot, majd az újraindítás közben a **Del** gombot amivel belépünk a BIOS-ba. Most kivehetjük már a GoboLinux telepítőlemezt a CD-tartóból, s különben is helyes, ha beállítjuk a BIOS-t úgy, hogy már a merevlemezről „butuljon” elsőként.

No és akkor most indítsuk is el a frissen feltelepített Linuxunkat! Hamarosan megvizsgáljuk, mit szűrhattunk el azalatt, míg végrehajtottuk a telepítés imént leírt embertpróbáló, iszonyatos szellemi erőfeszítéseit! Megtudjuk azt is, mi az hogy „disztribúció”, megismerkedünk néhány fontos rendszerállománnyal, az **mc** progival is amúgy alapszinten, s a haladó linuxosok megtudják azt is, mit tegyenek, ha meglevő Linux rendszerük mellé telepítették fel a GoboLinuxot.

Indítás, betöltésvezérlők s egyéb titokzatos dolgok

Sóval s tetvvel, feltelepítetted a GoboLinuxot, s újraindítottad a gépet. No és most számos eset elképzelhető. Például, még akár az is megtörténhet, hogy rögvest elindul neked a GoboLinux... Igen, igen, igen, ez igenis cseppet sem kizárt! Igenis előfordulhat, bár nemcsak amiatt, mert mindent helyesen csináltál, de az is kell ehhez, hogy ne legyen Windows a gépeden. Ezesetben ugyanis egyedüli oprendszer rajta maga a GoboLinux, s így minden külön sallang nélkül azonnal ez indul.

No és ha nem?

Mert mi van, ha a gépeden volt egy Windows is, te nem is annak a helyére telepítetted a GoboLinuxot, oszt' most mégis a Windows indul!

Ezesetben a baj valószínűleg ott rejtezik, hogy elfelejtetted telepítés közben bepipálni (beixelni) a „Betöltésvezérlő (bootloader) telepítése” sort, vagy ha az eredetileg be volt jelölve, kivetted onnan a jelölést. Nos, bár meg lehetne oldani kissé haladóbbaknak egyszerűbben is ezt a problémát, *neked mint kezdőnek* most az a legegyszerűbb, ha újratelepíted a rendszert, de most már figyelsz erre.

Mi van, ha volt egy Windows telepítve a gépeden, te nem annak a helyére akartad telepíteni a GoboLinuxot, de most mégis a GoboLinux indul, s nincs is lehetőséged induláskor választani a Windows és az új oprendszer közt?

Ezesetben valószínűleg elnézted amikor beállítottad, hogy melyik partícióra telepítenéd a Linuxot. Tévedésből a Windows partícióját adtad meg neki. Hát öregem ezt jól elrontottad – a Windowsodnak ez esetben örökre kampec, és egyáltalán semmiféle módon nem is hozhatod őt vissza az „életbe”! (Természetesen ha van valamiféle Windows rescue lemezed és biztonsági másolatod, arról visszatelepítheted, de lebeszéllek róla – nem éri meg a fáradságot...)

Az esetek többségében azonban, ha volt Windows a gépeden, indulásnál meg kell jelenjen egy kis menüféleség, amiből kiválaszthatod, melyik oprendszer induljon. Hát most kiválasztottad a Linuxot, s indul is...

De hoppá! Több indulási opciót ajánl fel! Például valami grafikusat meg valami konzolosat!

Nos, válasszuk bátran a konzolosat! (console). Előrebocsátok ugyanis valamit: Linux alatt a legtöbb dolgot nem egy-, hanem sokféleképp meg lehet oldani. E könyvben néha nem a legegyszerűbb(nek tűnő...) módszereket ismertetem, mert a látszatra bonyolultabbakban gyakorta több a lehetőség ismereteink

későbbi bővítésére! Válasszuk csak a konzolos indulást, és indítsuk a rendszert!

Némi töltődés után kér tőlünk két adatot, a nevet és a jelszót. NE a rendszergazdait adjuk meg neki, hanem a másikat, e könyv példája szerint tehát a „vz” felhasználó nevében és az ő jelszavával jelentkezzünk be!

Ezután pedig hagyjuk is el a karakteres felületet, s üssük be a

startx

parancsot. Hamarosan indul a KDE, s ismerős kép fogad majd bennünket: nagyon hasonló ahhoz, amit a telepítő CD képernyőjén láttunk!

Mielőtt továbblépnénk, vessünk egy pillantást arra az esetre, hogy mi van akkor, ha telepítés után egy egészen más Linux rendszer indul, nem a GoboLinux? Mert már volt telepítve egy „Linux” a gépünkre?

Nos, ez esetben kell hogy legyen azon disztrónak egy **menu.lst** állománya, s ezt többnyire a **/boot/grub/menu.lst** útvonalon leljük meg. Ezt kell szerkesztjük úgy, hogy belevezzük a GoboLinuxot indító sorokat is. Ez annyit jelent, hogy a következő sorokat kell belevennünk (feltételezve, hogy a HDA4-re van a GoboLinux telepítve):

```
title GoboLinux - Console (hda4)
kernel (hd0,3)/System/Kernel/Boot/kernel root=/dev/hda4 vga=0
```

Ha a GoboLinux a HDA6-ra van telepítve, akkor a sorok ezek lesznek:

```
title GoboLinux - Console (hda6)
kernel (hd0,5)/System/Kernel/Boot/kernel root=/dev/hda6 vga=0
```

HDA7-re telepítés esetén:

```
title GoboLinux - Console (hda7)
kernel (hd0,6)/System/Kernel/Boot/kernel acpi=off root=/dev/hda7 vga=0
```

Ha pedig a HDA8-ra telepítjük:

```
title GoboLinux - Console (hda8)
kernel (hd0,7)/System/Kernel/Boot/kernel root=/dev/hda8 vga=0
```

Amint látszik a titokzatos **(hd0,7)** részen, a vessző utáni szám eggyel kisebb, mint a partíció száma amire telepítettünk, - ez a titka az egésznek. Ez a Grub progí valami hülyesége, nem tudom az okát, de így van. A **hd** utáni szám pedig azt jelenti, hányas merevlemezről van szó.

Annak nem vagyok a híve, hogy valaki ne konzol módban indítsa a Linuxot – igaz „Kocka” nem cselekszik másként, az szentségtörés volna! - de a nagyon perverzek kedvéért íme egyetlen példa a grafikus indítás menübe való

huzalozására:

```
title Gobo HDA4 Grafikus mod (hda4)
kernel (hd0,3)/System/Kernel/Boot/kernel root=/dev/hda4 vga=0 Boot=Graphic
```

E sorokat lényegében teljesen mindegy hogy hova írjuk a **menu.lst** file-ban, de célszerű a legvégére, nehogy valami előző bejegyzést elsűrjünk vele. Mentsük el a fájlt, indítsuk újra a masinát, s ezután ha mindent jól tettünk, a grub menüjében kiválaszthatjuk a GoboLinuxot is, hogy indítsuk ha akarjuk.

Megjegyzés: Amikor a Grub fut, még nem érvényesek a nyelvi beállításaink, emiatt a karakterkészlet-kiosztás sem. Azaz, nem jó ötlet a title-kezdetű sorokban a menüpont megnevezésében ékezetes betűket használni. Helyettük ugyanis valószínűleg valami hülyeség jelenik majd meg a képernyőn.

Na de mi van, ha valakinek a régi Linuxa nem Grubot, hanem **Lilo**-t használ betöltésvezérlőként – kérdezhetné valaki?

Nos, fogalmam sincs, hogy akkor mi van. Tényleg! Halvány kisegítő segédpót- lila dunsztom sincs róla! Ebben nem tudok segíteni, a Lilót nem ismerem.

No és mi van akkor, ha volt valahová egy „Linux” telepítve már, de mi telepítettük a GoboLinuxot, és tévedésből bepipáltuk (bepipálva hagytuk) a „**Betöltésvezérlő (bootloader) telepítése**” sort, aztán most a GoboLinux indult, holott mi azt szeretnénk, hogy azért a másik, a régi Linuxot is indíthassuk, ha akarjuk?

Semmi probléma – ha nem épp a régi Linux partíciójára telepítettük a GoboLinuxot, akkor nem kell megijedni, öreg barátunknak semmi baja sem esett, és egyszerű életet lehelniünk belé: más sem kell hozzá, mint hogy a GoboLinux **menu.lst** fájlját módosítsuk a megfelelőképpen! Ehhez az kell, hogy elindítsuk a kedvenc szövegszerkesztőnket, például az **mc** progit, mert annak az editorja jó lesz nekünk, s megkeressük a Gobo menu.lst fájlját, ami a

```
/System/Kernel/Boot/grub/menu.lst
```

útvonalon lelhető fel, s ezt szerkesszük úgy, ahogyan az kell a régi „Linuxunk” kedvéért. Hogy annak milyen sorok a számára kedvesek, azt én nem tudhatom e könyv írásakor, mert esetről-esetre más lehet, s valószínű, hogy Te sem emlékszel rá fejből. Ez azonban nem gond, mert elég ha csak csatolod (felmountolod) a megfelelő partíciót valahová, s ott megkeresed az ottani **menu.lst** fájlt, belenézel, hogy mi van benne, s a megfelelő sorokat átmásolod a GoboLinux menu.lst fájljába. Újraindítás után minden rendben lesz, s választhatsz a két Linux között. Ezt nem írom le részletesebben, mert akinek már van valami Linux rendszer a számítógépén, az kell hogy ismerje ennyire a **mount** parancsot, az tehát ezt meg kell tudja csinálni. Ha mégsem, most már bírja ki a GoboLinuxszal mint egyedüli oprendszerével, amíg e könyvben el nem jut odáig, hogy ismertetem e fontos parancsot, a mountot.

No és mi van a Windowssal? Annak milyen bejegyzés felel meg a menu.lst fájlban?

Nos, bevallom, *ezt nem próbáltam ki*, mert nincs is idehaza Windowsom. Már lassan fél évtizede abszolúte bármiféle Windows nélkül élek, s erre kifejezetten büszke vagyok! (Igenis LEHET így élni, bármilyen hihetetlennek tűnjék is ez!) Más helyekről ellenben úgy értesültem, hogy annak ezek a sorok felelnek meg, ha az első merevlemez első partíciójára (azaz az általában szokásos helyre) van telepítve:

```
title Windows XP
root (hd0,0)
makeactive
chainloader +1
```

Hogy ez mit jelent, arról csak halvány fogalmaim vannak (mondjuk a (hd0,0) rész elég világos...). FELELŐSSÉGET NEM VÁLLALOK ÉRTE! És nem is ásom bele magamat a témába, mert abszolúte nem érdekel a Windows. Annyit viszont elárulhatok, mit jelent a **menu.lst** első 2 sora:

default 0 - az alapértelmezésként induló grub menütétel sorszáma. A számozás 0-val kezdődik. Természetesen arra írod át a nullát, amire csak akarod, csak legyen annyiadik tétel a menüben.

timeout 10 - Ennyi másodpercet várakozik az alapértelmezett menüpontban megjelölt operációs rendszer indítása előtt a számítógép, amennyiben addig nem ütünk le billentyűt. Nekem a 10 túl sok, én át szoktam írni 5-re.

Még valami. Bár nem vagyok híve a Windows használatának, de jobb meggyőződésemm ellenére leírom, miként kell a grub-ot visszatelepíteni, ha megsérül. Ugyanis gyakori kérdés a linuxos fórumokon, hogy mit tegyenek, ha a Windows harakirit követett el, tönkrement, újratelepítették, de emiatt már nincs grub (mert a Win felülírta...), s nem tudják indítani a korábban feltelepített Linuxot, most csak az XP van!

Előrebocsátom, a most leírt módszert ÉN NEM PRÓBÁLTAM KI. Nem tehetem – nincs Windowsom. De minden mérvadó helyen azt olvastam, hogy ezt úgy kell, hogy indítsz egy rendszert a LiveCD-ről (magyarán „bebutulsz” róla...), majd (terminálba menve, azaz konzolból) csatolod (tehát „felmountolod”) a partíciót, amin a Linux rendszered van (ez a hda4 a most következő példa szerint), ezt a felcsatolást e paranccsal teheted meg GoboLinux alatt:

```
mount /dev/hda4 /Mount/media
```

s ezután kiadod e parancsot:

```
grub-install --root-directory=/Mount/media /dev/hda
```


E parancsokat rendszergazdaként kell kiadnod, de a LiveCD-k többnyire úgy szoktak indulni.

Elvileg az a módszer is kell működjön, hogy (konzolban, rendszergazdaként...) a következő parancsokat adod ki:

```
mount /dev/hda4 /Mount/media  
chroot /Mount/media  
grub-install /dev/hda
```

Előfordulhat, hogy bár a Gobo jól betöltődik, működik is, de „mélykonzolban”, tehát amikor nem a grafikus felületet látod, akkor az ékezetes betűk helyett - „á,é,ó” stb – csak kérdőjelek jelennek meg neked („?”). Sőt, már betöltés közben is a grafikus karakterek helyett csupán csúnya kérdőjeleket látsz. Ez a probléma orvosolható, ha a menu.lst fájlban a GoboLinuxot indító sort kiegészítéd ennyivel: **vt.default_utf8=0**

Ezesetben tehát például így néz ki az a rész, ami a hda6 partíción levő Gobót tölti be úgy, hogy parancssorban induljon:

```
title GoboLinux - Console (hda6)  
kernel (hd0,5)/System/Kernel/Boot/kernel vt.default_utf8=0 root=/dev/hda6 vga=0
```

Ennyit tehát erről. Mint látszik, e fejezet javarészt egy kis kitérő volt a „haladóbb” felhasználók számára. A következő fejezetben visszatérünk az abszolút kezdőhöz, hogy tehát indította a rendszert, már felállt a grafikus képernyő is, és most...

Ha nem megy az Internet

E fejezetet erősen javallott akkor is elolvasni, ha nálad most épp MEGY az „Internet”, mert sok hasznos dolgot tanulhatsz a következő oldalakon, amiket a későbbi fejezetekben ismertnek tételezek már fel!

Ott hagytuk abba, hogy tehát indítottad a rendszert, már felállt a grafikus képernyő is, és most... most kissé meg szeretnél ismerkedni új operációs rendszereddel. Mindenekelőtt most egy hosszas szövegelés következik a részemről. Elmagyarázom, mi az, amit látsz. Mi az, hogy ablakkezelő, mi az, hogy „desktop environment”, magyarul „munkaasztali környezet”, mi az, hogy „disztribúció”, miért olyan drámaian eltérő kinézetre két-két Linux rendszer, s a végén elmondom azt is, mi a teendő akkor, ha most megkeresed a Firefoxot, internetezni akarsz vele, és az netán nem működne. Ami velem is megesett. Telepítés közben ugyanis rosszul írtam be az internetcsatlakozásom bizonyos adatait. No most a legnagyobb ökörség volna efféle hülyeség miatt újratelepíteni a rendszert. A Linux nem Windows, nem jellemző rá, hogy gyakran kéne újratelepíteni. Persze a „Linuxok” közt is vannak stabilabbak és instabilabbak, de egy egészen pocsék és instabil Linux is olyan, hogy ha a Windows lenne olyan, azt „atomstabilnak” neveznék a készítői. Szóval, meg fogunk ismerkedni a Linux legfontosabb, Internettel kapcsolatos rendszerállományai-val. Ez akkor is jól jöhet, ha egyszerűen a Szolgáltatód megváltoztatja mondjuk a DNS szervereinek az IP címét. (Nálam ez megtörtént...) Emiatt természetesen eszünk ágában sincs újratelepíteni egy remekül testreszabott rendszert...

Kezdjük tehát!

A „disztribúció”, az első megközelítésben a Linux egy „változatát” jelenti. Ez persze így szó szerint nem igaz, nem szakmai megfogalmazás, de a felhasználó szempontjából érthető. Az a szó ugyanis, hogy „Linux”, az tulajdonképpen csak az úgynevezett „kernelt” jelenti, magyarul rendszermagot. Ez elnagyoltan fogalmazva, a legalapvetőbb rutinok gyűjteménye. Természetesen ennek is van számos változata, mert vannak régebbiek meg újabbak (amint a Windowsnak is), de nagy vonalakban azért ez minden „Linux” esetén, azaz minden disztribúcióban azonos.

No most, önmagában egy efféle rendszermag nem sok mindenre használható, pláne nem kezdő felhasználóknak. Ebben nincs progi az internetezésre, szövegszerkesztésre, stb. Akadnak azonban egyesek – egy-egy ember, vagy egész csoportok, de még némely cégek is – akik e rendszermaghoz összeválogatnak programokat, olyanokat, amik jól működnek az adott rendszermaggal (és egymással), s amelyeket a Felhasználók gyakran igényelnek. Ezeket be is állítják, csinálnak e programgyűjteményből egy CD-t, DVD-t vagy ritkábban

valamilyen tömörített állományt, és erre mondják azt, hogy ez egy „disztribúció”. (E szót magyarul talán mint „terjesztést” lehetne lefordítani, de senki sem mondja így). Minthogy rengetegféle progí van egy-egy feladatra, például szövegszerkesztésre Linux alatt, és csakis a disztribúció – rövid nevén, azaz becenevén „disztró” - készítőin múlik, a „csomagba” mely progikat válogatják be, emiatt rengetegféle disztró létezik, ami azonban mind egy-egy „Linux”. Némelyik egészen speciális igényeket elégít ki – létezik olyan is, ami elfér mindössze egyetlen floppy lemezen, és csak annyit tud, hogy ha onnan elindítják, automatikusan elindít egy programot, ami azt tudja, hogy megkeressi a számítógép merevlemezére telepített Windows 2000 vagy Windows XP felhasználóinak nevét és jelszavát, s ezt át lehet írni általa, illetve ki lehet e jelszavakat törölni. Azaz így „fel lehet törni vele a Windowst”.

Ez természetesen rendkívül speciális alkalmazási terület... de ettől eltekintve is nagyon sokféle disztribúció létezik. Akadnak amik csak egy DVD-re férnek rá, olyan sok progít tartalmaznak; a legtöbb 1 CD-re fér fel; de ezeken belül is óriásiak a különbségek mondjuk például a kinézetet illetően. A különbségek SOKKAL nagyobbak, mint két tetszőleges Windows között. Ennek már az is az oka, hogy Windowsból nemigen van több, mint talán egy tucat fajta, Linux disztribúcióból azonban jelen pillanatban, amikor e sorokat írom, 2008.02.13.-án, 352 fajta létezik. Az adat a <http://distrowatch.com/> linken található oldalról származik, ahol a különböző disztribúciók szépen fel vannak sorolva, keresgélhetünk köztük különböző szempontok szerint. Ekkora bőségből igazán mindenki megtalálhatja a neki tetszőt! E gazdagságról nem is álmodhatnak a Windows használói! Igazán nevetséges, hogy ezek után a Linuxot nem ismerők rendszeresen azzal vádolják ezt az oprendszert, hogy Linux alatt „nincsenek programok”, hogy itt sokkal kevesebb a progí, mint a Windows alatt. A valóság ezzel éppen ellentétes.

Persze ezek után azonnal felmerül a kérdés, miért éppen pontosan a GoboLinuxot szemeltem ki e bőségből arra, hogy e könyvben ismertessem! Pláne ha a kedves Olvasóm netán hallott is már emlegetni néhány nevet az elterjedtebb disztribúciók közül, mint például Debian, Ubuntu, Kubuntu, Xubuntu, UhuLinux, SuSE, RedHat, Fedora, Slackware, PCLinuxOS, Gentoo, Frugalware, Mandriva, Blackpanther...

Ez természetesen rendkívül fontos kérdés, és eszem ágában sincs válasz nélkül hagyni! Épp csak erre nem most válaszolok majd, hanem a könyvem egyik későbbi fejezetében, melynek címe: „Filozófálgatunk – miért épp a GoboLinux?” - és azért ott, mert ez nem kezdőknek való téma még.

A disztrók egészen érthető módon csoportosíthatóak abból a szempontból is, mennyire „kezdőbarátok”. Azaz, mennyi tudás kell a kezelésükhöz. Olyan egyetlen sincs, mely abszolút nulla tudással kezelhető, de olyanok akadnak azért nem is kis számban, melyek – legalábbis akkor, ha csak nagyon hétköznapi dolgokra akarod használni – csak egészen kevés tanulást igényelnek. Ténylegesen akadnak azonban olyan disztrók is, melyek egyáltalán nem rendelkeznek grafikus felülettel. Ez természetesen nagyonis nem tekinthető kez-

dőbarátnak! Sok ilyen disztribúció akad, de ezek *nem is akarnak* kezdőbarátok lenni, ezeket többnyire valami speciális célra készítik. Megvan azonban ezeknek az az előnyük, hogy ezek többnyire kis helyigényűek. A kezdőbarát disztrók NAGYOK, mert a grafikus felület előállítására szolgáló programok, pláne amik az alatt futnak, bizony jókora méretűek. Erre konkrét példát tudok hozni: én magam írtam egy picike programot, ami annyit tudott, hogy meg lehetett adni hozzá egy szöveges fájlt, amit mi bővítettünk ha akartunk szópárokkal, s a természetesen parancssorból indított progink ebből kikereste s kilistázta az összes olyan sort, amiben a keresett szó benne volt. Azaz ez egy rendkívül primitív kis szótárprogi. S ismétlem parancssoros. Amiatt írtam meg, mert felmerült az igény az egyik fórumon, hogy valaki szeretne egy lengyel-magyar szótárat létrehozni, mert ilyet nem talált, de nem talált olyan progit sem, aminél könnyen lecserélhette volna az abban meglevő (tipikusan angol-magyar) szótárat, s a legtöbb szótárfájl szerkezete is bonyolult volt neki, mert adatbáziskezelőt igényeltek, stb. Nos én megírtam ezt a progit igen hamar, s a lefordított, működőképes progik mérete mindössze 8888 bájt jelenleg! (véletlenül ilyen szép „kerek” szám). Azaz mindössze kb 8 kilobájt! Elkezdtem ellenben írni egy grafikus felületet tartalmazó progit is, és bár még semmi nincs benne a szótárból, a progik máris 25254 bájt, azaz háromszorosa az előbbinek! Holott még grafikailag is rém primitív! És ez utóbbi nem is működik akkor, ha önmaga van csak fenn a gépen, hanem igényli mindenféle más progik meglétét is. Tehát jóval bonyolultabb és nehezkesebben használható.

A csak „parancssoros” disztrók tehát jóval kisebbek. Viszont egy kezdő számára nem annyira barátságosak.

A „parancssor versus grafikus képernyő” kérdésre még számos alkalommal kitérünk majd e könyvben. Most azonban ennyi feszültségfeloldó szövegelés után kezdjünk bele „Linuxunk” használatába.

Mindenekelőtt: amit magad előtt látsz, eléggé hasonlít valami Windows-képernyőhöz. Tudsz „egerezni”, kattintgatni, stb. A menüt a bal alsó sarokban levő „K” jelű ikonra kattintva hívhatod elő. Mindezt a KDE nevű úgynevezett „desktop environment” (röviden DE) biztosítja neked, amit magyarul „munkasztali környezet”-ként lehetne lefordítani. Van ebben minden: határidőnapló, „ragadós cetlik”, beállítóközpont ahol a rendszer fontos paramétereit állíthatod be (s amit előre bocsátom, SOHA nem fogunk használni e könyvben!). Mindez nagyon ismerős lehet a Windowst használók számára. S valóban: sokféle „DE” létezik a Linuxos világban. Legalább 3: ezeket a „desktop environmenteket” a KDE, Gnome illetve XFCE nevekkkel illetik. Ezek közül a KDE az, amely kinézetre a leginkább közelít a Windowshoz. (Emiatt is nem szeretem... de nem kell aggódni, kedves Olvasóm: el fogunk jutni e könyvben odaig, hogy lecseréljük az egész KDE-t egy sokkal egyszerűbbre! Hú hogy az milyen egyszerű lesz!)

Egy „DE” természetesen amiatt hasonlít a Windowshoz, mert a Windows is – legalábbis a Windows 2000, a Windows XP és a Windows Vista – MAGA IS tulajdonképpen egy „desktop environment”. Egy DE-re jellemző, hogy sok program bonyolult összessége, melyek nemcsak a tulajdonképpeni „ablakkezelést” végzik el, vagyis a programok ablakainak megjelenítését, mozgatását stb, de ezen felül igyekeznek az egész számítógépet vezérelni, undorító módon azon mesterkednek, hogy kitalálják a felhasználó gondolatait, emiatt elindítanak a háttérben egy rakás „fontos” szolgáltatást, s ezzel mind zabálják a memóriát... a KDE is amit látsz, már induláskor megeszik vagy 200 megát. Iszonyatos. Mondjuk ez még mindig nagyon szép ahhoz képest, hogy úgy hallottam, a Vista el sem indul 1 giga RAM alatt. De ez csak hallomásból szerzett infó, mert bevallom, sosem próbáltam ki a Vistát. Viszont nagyon sok helyről hallottam. De ez persze nem biztos hogy igaz, mert ismétlem, nem próbáltam.

No most a DE-kkel szemben vannak azok a grafikus felületet vezérlő progik, amiket az angol úgy hív, hogy „Window Manager”, röviden WM, magyarul „ablakkezelő”. Ezek nem rendelkeznek annyi szolgáltatással, mint a DE-k, sokkal kevesebb memóriát igényelnek maguknak, de többnyire nem tekintik őket annyira felhasználóbarátnak. Ezek tipikusan nem is tartalmazznak például olyasmit, hogy „vezérlőpult”. Aki nem DE-t hanem WM-et használ, az bizony a fontos rendszerbeállításait kénytelen úgy módosítani, hogy önmaga átírja valami szövegszerkesztővel a megfelelő úgynevezett „config fájlt”. Bár nekünk van KDE-nk, de mégis így cselekszünk majd, mert meg akarjuk ismerni a rendszerünket, tudni akarjuk, mi van benne, mi mit vezérel, hogyan működik. Tudod, ezt úgy szokták mondani: ha Windows vezeti a kocsidat, te az anyósülésbe kényszerülsz és csak utas lehetsz. A Linux odaenged téged a vezetőülésbe. Sajnos azonban ha te vezetsz, és nem értesz hozzá, karambolt is okozhatsz.

Mi meg fogjuk tanulni „vezetni” a Linuxunkat.

Most azonban örvendj egyelőre a frissen telepített Linuxnak. Kezdetben javaslom, próbáld megkeresni a menüben a Firefoxot. Indítsd el. Ha megvan, keresd meg valamelyik kedvenc fórumodat, és írd be, hogy te most telepítettél egy Linuxot, s már alóla Internetezel! Ez jó érzés lesz neked, mert „geek” színben tüntet fel téged a haverok előtt. Hú te milyen kemény csávó vagy, atyavilág, te Linuxot használsz! Te aztán értesz a masinához! Ez jó érzés, és nagyon alkalmas rá, hogy megnövelje az úgynevezett „E-penis” mértéket, azaz hogy szakértő színben tündökölj mindenki előtt, mert te milyen „fasza fickó” vagy. S ha ezt élvezed, azt szégyelned sem kell, mert MEGÉRDEMLED az elismerést, ugyanis ha e részletes leírás alapján telepítesz is, és még marhára mit sem értesz is az egészből, de legalább volt BÁTORSÁGOD arra, hogy KIPRÓBÁLJ valami ismeretlen dolgot! Ez igenis elismerést érdemel. A legtöbb ember túl gyáva és/vagy lusta ehhez.

Ha kifejezetten a GoboLinuxsal kapcsolatos fórumot keresel, aminél elújsághatod hogy felraktad, akkor e két linket ajánlom figyelmedbe:

Magyar nyelvű fórum:

<http://linuxempire.hu/viewforum.php?f=30&sid=6f5e5b90ef55e56aa5d66e95b55f1652>

Angol nyelvű GoboLinux fórum itt található:

<http://forum.gobolinux.org/>

No és most kezdődhetnek a problémuszok. Mert előfordulhat, hogy a Firefox olyasmit ír ki neked, hogy nem tud csatlakozni az internetre, s ellenőrizd az internetbeállításaidat!

Nos, ezesetben valószínűleg elszúrtad telepítéskor az internetbeállításaid adatainak bepötyögését. Ellenőrizni kell őket, valóban. De hogyan teheted meg?

Leírom lépésről-lépésre.

Mindenekelőtt: lenn a „tálcán” látsz egy olyan ikont, ami valami ronda fekete képet mutató monitort formáz. Na ez a „**Konsole**” nevű progí. Indítsd el. Egy úgynevezett „virtuális terminálba” jutsz, ami hasonló mint a „DOS-ablak” a Windows alatt.

Most rögtön rendszergazdai jogosultságokat kell szerezned, mert (esetleg) szerkesztened kell egy vagy több fontos rendszerállományt. Ezt úgy éred el, hogy beütöd:

su -

(a „-” jelet is üsd be, de legyen szóköz a su és a mínuszjel közt).

Ekkor megkérdi tőled a jelszót. Ez az a jelszó, amit telepítéskor beállítottál a rendszergazdának, akinek a neve nálunk „pv”. Miután megadtad, megváltozik a prompt, piros színű lesz. Ebből a rendszergazdai üzemmódból az

exit

parancs kiadásával léphetsz ki.

Most pötyögd be, hogy „**mc**” (idézőjelek nélkül, és Entert is nyomj...) és bejutsz az **mc** progiba, ami a Norton Commanderhez hasonló. Itt a kurzorbillentyűkkel el kell navigálj a **/System/Settings** könyvtárba. (a fájlok/tartalomjegyzékek közt legfelül mindig találsz majd egyet aminek az a neve, hogy **/..** - nos, ez jelenti a „szülőkönyvtárat”). A **/System/Settings** könyvtárban keresd meg a **resolv.conf** állományt, menj rá a kiválasztást jelző csíkkal, s nyomd meg az F4 gombot!

Ilyesmit fogsz látni:

```
nameserver 212.24.160.1
nameserver 212.24.164.1
nameserver 82.144.169.165
nameserver 82.144.160.179
```

A te esetedben valószínűleg csak 2 sor lesz 4 helyett. Ez nem hiba, egyszerűen az én Internetszolgáltatónál 2 helyett 4 DNS-szerver is van. Látod ezért is jó tudni, ezeket a dolgokat hova kell beírni: telepítéskor csak 2 DNS-szerver IP címét adhattad meg, de ha több van, azokat utólag kell felvenni! No most ellenőrizd le, jó értékek szerepelnek-e ebben az állományban – ha nem, akkor

javíts, és mentsd el a változtatásokat az **F2** gombbal!

Ezután – főleg ha itt nem találtál hibát – vizsgáljuk meg a
/Programs/BootScripts/Settings/NetworkOptions állomány tartalmát is!
Itt ilyesmit kell látnod:

```
NetworkInterfacesOnBoot=(  
    lo  
    eth0  
)  
  
lo_IP=127.0.0.1  
lo_Netmask=255.0.0.0  
#eth0_BootProto=Static    # DHCP (requires DHCPd package), Static (default if  
empty)  
eth0_IP=82.144.172.154  
eth0_Netmask=255.255.255.0  
#eth0_Broadcast=  
Gateway=82.144.172.1  
GatewayInterface=eth0
```

Ezekből téged főleg az **eth0_IP** kezdetű sor érdekelhet, ide a te IP címed kell beírva legyen. Ha nem az, akkor javítsd. Érdekelhet még azért a többi is, tudniillik a **Gateway** a te „Átjáród” másnéven „alapértelmezett átjáród”, azt is ellenőrizd le, meg a „**Netmask**”-ot is. Ha valami hibás, javítsd ki, és mentsd el az állományt.

Miután mindezzel megvagy, még benézhetsz ugyanebben a könyvtárban a **BootOptions** állományba is, ami valahogy így néz ki nálam:

```
HostName=Csiszilla  
  
#ClockMode=GMT  
#ClockMode=LocalTime  
ClockMode=LocalTime  
  
#BootTheme=AppleII  
#BootTheme=CheckList  
#BootTheme=Hat  
#BootTheme=Progress  
#BootTheme=Progress-II  
#BootTheme=Progress-III  
#BootTheme=Quotes  
#BootTheme=ScanDisk  
#BootTheme=Slack  
#BootTheme=SplitScreen  
BootTheme=Progress  
  
#KeymapLayout=br-abnt2  
#KeymapLayout=us  
#KeymapLayout=dvorak.map  
KeymapLayout=hu.map
```

```

MouseDevice=/dev/input/mice # Mouse Device
MouseType=imps2 # Mouse GPM protocol

ConsoleFont=/Programs/KBD/Settings/kbd/consolefonts/lat2-16.psfu.gz

# If you need to load a module on boot, use the array below.
# However, be warned that the correct way to load modules
# is using the load-on-demand facilities of /etc/modprobe.conf.
# For example: to set the module for your network card, you
# only need to add a line like
#   alias eth0 via-rhine
# to /etc/modprobe.conf. To load your ALSA sound module, use
#   alias sound-slot-0 snd-via82xx
# (replacing the module names above by those that match your
# hardware, of course).
UserDefinedModules=(
    snd-pcm-oss
    snd-mixer-oss
    snd-intel8x0
)

```

(A #-el kezdődő sorok mind megjegyzések, a számítógép nem veszi figyelembe őket).

A **HostName** után természetesen a te gépneved van megadva, amit telepítéskor beállítottál. Ezen kívül a **KeymapLayout** érdekelhet téged, ami a billentyűzetkiosztásodat határozza meg, hogy tudniillik az magyar legyen. A többi dolog egyelőre nem hiszem hogy érdekes lenne a számodra. (Egyelőre...)

Ha már ügyis terminálban vagyunk (azaz „konzolos” úgymond „üzemmódban”), ráadásul rendszergazdai jogosultságokkal, akkor beállítunk még valamit. Ha ugyanis most kilépnénk az **mc**-ből (ez az **F10** billentyűvel érhető el), akkor ott találod magadat abban a könyvtárban, ahonnan az **mc**-t elindítottad, s nem maradsz abban a könyvtárban, ahol legutoljára az **mc**-ben voltál. Márpedig ez nagyon jó lenne, ha a könyvtárnevek bonyolultak, ekkor ugyanis messze sokkal könnyebb az **mc**-ben eljutni valahová, mint kiadni a

cd könyvtárnév

parancsot (amiről eddig nem is írtam neked). Nos ezt a hibát rögvest kiküszöböljük!

Írjuk be a

```
/Programs/Scripts/Settings/zshrc
```

és a

```
/Programs/Scripts/Settings/bashrc
```

file-okba az alábbi sort (célszerűen a legvégére):

```
alias mc='. /Programs/MC/Current/Shared/mc/bin/mc-wrapper.sh'
```

Ügyeljünk rá, hogy a pont és a /Programs karaktersorozat közt van egy szóköz!

No ha mindezzel megvagyunk, lépünk ki az mc-ből, majd adjuk ki az **exit** parancsot, hogy kilépjünk a rendszergazdai (azaz „root”...) jogkörből, s visszasüllyedjünk „normál” felhasználóvá. Ezután egy újabb **exit** paranccsal be is zárjuk a Konsole ablakát.

Most újra kell indítani a rendszert, hogy meggyőződjunk róla, már működik „a Zinternet”. A KDE menüjéből válasszuk ki a **kijelentkezés** menüpontot. Vissza fogunk jutni oda, ahol a **startx** paranccsal a grafikus felületet indítottuk.

Most ismét menjünk rendszergazdai módba a

su -

paranccsal, majd kétféleképp is újraindíthatjuk a rendszert:

egyik módszer, hogy e parancsot adjuk ki:

reboot

a másik módszer az, hogy ezt:

shutdown -r now

Most rögtön elárulom azt is, ha nem újraindítani akarunk, hanem leállítani, azt hogyan kell! Nos, akkor a fenti 2 parancs helyett e két parancs valamelyikét kell kiadni:

halt

vagy

shutdown -h now

Rendszergazdai jog amiatt kell, mert ilyen fontos dolgot mint a gép leállítása vagy újraindítása, „közönséges júzer”, egyszerű „mezei” felhasználó nem végezhet.

Előfordulhat, hogy a **reboot** működik a GoboLinux rendszered alatt (A **shutdown -r now** parancs pld) de a rendszer leállítása (**shutdown -h now**) már nem! A rendszer látszólag kilép, de a számítógép nem áll le, a disk pörög tovább... Ezesetben olvasd el e könyv vége felé a „**Shutdown**” című fejezetet, ott le van írva, mit tegyél!

A hálózat leállítása és újraindítása

A fenti megoldás, hogy újraindítjuk a számítógépet a hálózati konfiguráció megváltozása miatt (ugye ha tehát például megváltoztatjuk az IP címünket, megváltozik a DNS-szerverek IP címe, vagy akármi mást megváltoztatunk), nos, ez nagyon „Windowsos” módszer! Mi linuxosok szeretjük elkerülni az efféle „életérzést”, „felhasználói élményt”. Szerencsére van módszer rá, hogy rendszerünknek csak azt a részét állítsuk le, és indítsuk újra, ami speciálisan az „internetes” (szakmaibb kifejezéssel élve „hálózati”) részt kezeli. Ezt e két

paranccsal érhetjük el:

StopTask Network
StartTask Network

Hát nem egy nagy ördögösség megjegyezni... az első parancs természetesen leállítja a hálózati interfészt, a másik újraindítja – de ekkor már az új beállításaink aktiválódnak. Persze, e két parancshoz is rendszergazdai jogkör szükségeltetik...

A mount parancs és az fstab állomány

Indítsuk újra a rendszert, de a telepítő CD-ről! Bátran csináljunk mindent úgy, mintha telepítenénk, egészen addig, míg a **startx** parancs hatására feláll a grafikus képernyő. Ezután azonban ne kattintsunk az „Install GoboLinux” ikonra, hanem indítsuk a Konsole nevű progit, azaz lépünk át a virtuális terminálba!

No és most szépen elintézzük, hogy a Linuxunk helyesen kezelje a többi partíciót, tehát ne csak azt, amire épp telepítve van. **Ez cseppet sem lesz egyszerű téma**, ami azt jelenti, hogy számos dolgot kell megértenünk. Ez mind mulhatatlanul szükséges ahhoz, hogy rendszerünket kezelni tudjuk a jövőben. Valójában talán ez a rész az, mely egy abszolút kezdőnek a legtöbb nehézséget okozza majd e könyv fejezeteiben. Ez tehát kemény menet lesz, kedves Olvasóm!

Mindenekelőtt megmutatjuk, miként kell biztonsági másolatot készíteni a már feltelepített rendszerünkről – méghozzá olyat, mely akár működik is!

Figyelmeztetés: legyünk különösen körültekintőek most, amikor parancsokat írkálunk a „konzolba”, mert már a prompt színén is látszik – mely piros – hogy most rögvest rendszergazdai joggal fut a rendszer, mely a CD-ről „butult”.

Most indítsuk az **mc** progit, és lépünk be a **/Mount** könyvtárba. *Ez természetesen nem a feltelepített rendszer Mount könyvtára*, hanem azon rendszeré, mely most a CD-ről fut. Itt hozzunk létre két új könyvtárat, melyek neve legyen HDA4 és HDA7 – ez mc-ben az F7 gomb megnyomásával érhető el. Ezután hagyjuk el az mc-t az F10 gomb segítségével, és adjuk ki a következő két parancsot sorra egymás után:

```
mount /dev/hda4 /Mount/HDA4  
mount /dev/hda7 /Mount/HDA7
```

(Tudjuk ugye: hda4 helyett mindig azt kell írni, ahová telepítettük a rendszert, a hda7 helyére pedig azt a bizonyos másik, létrehozott partíciót, mely ugyanakkora legalább, mint amire telepítettünk!)

Ügyeljünk rá, hogy a Linux rendszer megkülönbözteti a parancsokban és a könyvtár- valamint állománynevekben a kis- és nagybetűket! Ezt korábban írtam már, de nagyon fontos!

No most, mit is csinál ez a két parancs?

Tudni kell, hogy a Linux alatt nincsenek holmi „A:”, „C:” stb. betűkkel jelzett úgymond „meghajtók”. A Linux alatt minden – fájl. (file). Tehát állomány. Eképp egy CD is egyszerűen egy állomány. No de minket pillanatnyilag

nem a CD-k érdekelnek – azokkal később foglalkozunk – hanem a partíciók! Nos, egy-egy partíciót is kezelhetünk állományként. Természetesen egy CD vagy egy partíció bizonyos értelemben különleges állomány – úgynevezett „device” állomány. Ezekre a hivatkozások össze vannak gyűjtve egy speciális könyvtárban, aminek a neve: **/dev**. Ezzel többnyire nem kell foglalkoznunk, ami nagyon megnyugtató tudat, mert nem kezdőnek való téma. *Néha* azonban kell vele foglalkoznunk mégis, egy-egy parancs erejéig, azért, hogy „fogyaszthatóbb” formára hozzunk valamit onnan.

Konkrétan, a Linux lehetőséget biztosít rá, hogy megmondjuk, a mi magunk által szokásosan „látott” könyvtárszerkezetén belül melyik könyvtáron át érjük el valamely speciális dev könyvtárat. A fenti két parancs példáját megmagyarázva, az első például azt mondja a gépünknek, hogy „Hé masina, ezentúl ha bármi parancsban is a /Mount/HDA4 könyvtárra hivatkozom, akkor tudjad, hogy az valójában a /dev/hda4 könyvtárra vonatkozik, tehát az „a” jelű merevlemezem 4-es partíciójára”! Ezt úgy mondjuk, hogy „felcsatoltuk” („felmountoltuk”) a partíciót a /Mount/HDA4 könyvtárba.

Ezt a gép megjegyzi, egészen addig, míg el nem felejtetjük vele. Ez az „elfelejtetés” az **umount** paranccsal történik, amit a fenti példában úgy is megadhatunk, hogy

```
umount /dev/hda4  
és úgy is, hogy  
umount /Mount/HDA4
```

vagyis megadható e parancs akár az eszközfáljra, akár a csatolási pontra hivatkozással. Ez a parancs nevezetetik úgy, hogy „lecsatoljuk”, azaz „umountoljuk” a partíciót (vagy amit korábban felcsatoltunk, például CD-t, ISO képmás-fájlt, stb).

No most amíg nem csatoljuk le a felcsatolt akármit, addig ez azt eredményezi, hogy most ha be akarunk lépni a csatolási könyvtárba, például a HDA4-be, akkor be is léphetünk oda, s itt láthatjuk az adott partíción levő összes állományt, és ezekkel műveleteket végezhetünk: másolhatunk, törölhetünk, könyvtárakat hozhatunk létre, akármit csinálhatunk. Megjegyzendő, hogy a csatolási pont kiválasztása teljesen önkényes. Hagyomány szerint Linux rendszerek alatt csatolni dolgokat általában a **/mnt** vagy **/media** nevű könyvtárak(ban létrehozott alkönyvtárak)ba szoktak; GoboLinux alatt erre a célra a **/Mount** nevű könyvtár szolgál. De megadható teljesen más, létező könyvtár is e célra. Akár az is lehetne a neve, hogy **/Users/vz/HogyItyeg_a_Fityeg** – feltéve, hogy e könyvtár létezik.

No most lépjünk be újra az mc progiba, lépjünk ott be az egyik panelen a /Mount/HDA4 könyvtárba, a másik panelen a /Mount/HDA7 könyvtárba, majd a HDA4-en jelöljük ki az összes könyvtárat és fájlt (az Insert gombbal próbáld...) s másold ezeket át a HDA7-re. (F5 gomb...)

E másolás el fog tartani egy darabig – ne türelmetlenkedjünk! Már a másolás megindulására is várni kell kicsit (nem is kicsit...), mert mielőtt elkezdene másolni, előbb beolvassa magának az összes fájl és könyvtár nevét (rekurzívan).

Miután ezzel megvagyunk, neked elvileg kell legyen egy teljesértékű Linux-rendszered a HDA7 könyvtárban, ami a hda7 partíciónak felel meg! (Feltételeztem, hogy a HDA7-en semmi állományod nem volt eredetileg – ha mégis, akkor töröld le – már ha nem kell neked).

Persze „nem ilyen szép a gyerek”, mert hiába van ott a telepített rendszered összes állománya, ez nem tud még futni. Ehhez az kell, hogy megmondd a betöltésvezérlőnek, hogy van egy „Linuxod” már a HDA7-en is! Ehhez menj el a **/System/Kernel/Boot/grub/menu.lst** fájlhoz, és szerkeszd úgy, hogy beleveszed a következő sorokat:

```
title GoboLinux - Console (hda7)
kernel (hd0,6)/System/Kernel/Boot/kernel acpi=off root=/dev/hda7 vga=0
```

Egyéb részleteket ezzel kapcsolatban lásd az „Indítás, betöltésvezérlők s egyéb titokzatos dolgok” című fejezetben.

Te most tehát tulajdonképpen „klónoztad” a rendszeredet. Ez nagyon jó arra az esetre, ha már van egy nagyjából jól beállított rendszered, s te tovább akarsz „turkálni” benne, de félsz, hogy valamit elrontasz. Ekkor mielőtt tovább haladnál, készítesz egy másolatot róla – ehhez előbb esetleg le kell törölni a klón számára fenntartott könyvtárban (itt a HDA7-ben) ott levő régebbi másolatot. Így ha az „éles” rendszeredben elrontasz is valamit, semmi vész, „bebutulsz” a LiveCD-ről, és onnan ügymódra letörölsz mindent a HDA4-ről, majd visszamásolod a HDA7 tartalmát.

Ha nem egy de két klónod van a rendszerről, akkor már LiveCD sem kell: az egyikbe „bebutulsz”, s innen ügymódra másolsz.

Na jó, és most lássuk, miért e nagy óvatosság!

Csak. Biztos ami biztos. Mindenesetre ha most már itt vagyunk, belenézünk kissé a „hírös-nevezetös” fstab állományba!

Ugye, a merevlemezünkön sok partíció lehet. Ezeket induláskor megpróbálja felderíteni a számítógép (pontosabban a GoboLinux), és amit talál, azt fel is csatolja a **/Mount** alatt önmaga által létrehozott alkönyvtárakba. No de rögvést felmerül néhány probléma:

1. Lehet, hogy neked nem tetszik az a név, amit ő kioszt a csatolási pont számára. Lehet hogy te nem „/Mount/HDA6” néven akarod látni azt a könyvtárat, hanem így: „/Mount/Zene”. Sőt, esetleg nem is a /Mount könyvtárba akarod csatolni, hanem rögvést a gyökérkönyvtárba, így:

„/Filmek”.

2. Az is lehet, hogy a GoboLinux netán nem ismerte fel az összes partíciót, vagy rosszul. Például nagy valószínűséggel ez lesz a helyzet, ha 4-nél több partíció van egyetlen merevlemezen. Ekkor ha a tovább darabolt elsődleges partíció eredetileg egy FAT partíció volt, könnyen lehet, hogy a másodlagos partíciókat mind FAT-nak tekinti, akkor is, ha azok külön-külön amúgy mondjuk ReiserFS fájlrendszerűek. Nálam például ez így van.
3. Az is lehet, hogy te netán nem is akarsz, hogy automatikusan fel legyen csatolva induláskor minden partíció. Vagy azt akarsz, hogy némelyek csak rendszergazdai joggal legyenek elérhetőek...

Mindezen dolgok tisztázására érdemes megismerkedni az fstab állománnyal. Itt tárolja a rendszer, hogy miként viselkedjék az egyes partíciókkal kapcsolatban.

Tudni kell ugyanis, hogy fentebb nem mondtam el mindent a mount paranccsal kapcsolatban. Holott ez nem kerülhető el, mert ez az egyik **létfontosságú** parancs a Linuxban! Szerencsére a Linux alatt létezik egy **man** nevű parancs is, ami a „manual” rövidítése. Ez azt jelenti, hogy ha a „man” után (szóközzel elválasztva...) beírjuk egy másik parancs nevét, akkor kiírja a képernyőre azon parancsról a legfontosabb tudnivalókat. Eképp a

man mount

parancs kiírja nekünk, amit a mountról tudni érdemes! Van manuálja magának a man parancsnak is, ami a

man man

paranccsal érhető el. E „manuál” oldalakból a **q** gomb lenyomásával léphetünk ki. (Hogy én mit szenvedtem, mire erre rájöttem! Mindenáron az Esc-vel akartam kilépni, de az nem működött...)

Erősen javallott e man oldalak tanulmányozása!

Na de a mounthoz visszatérve, a legfontosabb tudnivalók róla a következők: Általános szintaxisa:

mount [kapcsolók] [eszköz] [csatolási_hely]

Új filerendszer csatolását végzi egy könyvtárstruktúrába. *A meg nem adott paramétereket a parancs az **/etc/fstab** file megfelelő bejegyzéseiből veszi.*

A sok lehetséges kapcsoló közül egy nagyon fontos:

mount -t típus Megadja a csatolandó filerendszer típusát. Például:

mount -t iso9660 /dev/hdc /Mount/CDROM – ez természetesen egy CD felcsatolásakor használatos.

Minthogy írtuk, hogy a mount parancs kiadásakor a meg nem adott paramétereket a parancs az `/etc/fstab` file megfelelő bejegyzéseiből veszi, ebből logikusan következik, hogy ezen állományban viszont **muszáj** hogy meglegyen **minden** szükséges infó a rendszer számára! Ezt tehát be kell állítanunk neki! (Illetve ellenőrizni, hogy jól állította-e be magának telepítéskor). No most ehhez hogy ezt megtehessük, VALÓBAN ismernünk KELL a rendszerünket – tudni kell legalább azt, hogy milyen fájlrendszertípust használnak a partícióink!

Íme a példa kedvéért az én fstab állományom, némi magyarázattal fűszerezve:

```
vz@Csiszilla ~]cd /System/Settings
vz@Csiszilla /System/Settings]cat fstab

# Proc filesystem.
proc /System/Kernel/Status proc defaults 0 0

# Sys filesystem.
none /System/Kernel/Objects sysfs defaults 0 0

# Pts filesystem.
none /System/Kernel/Devices/pts devpts defaults 0 0

# USB filesystem.
# none /System/Kernel/Status/bus/usb usbfs defaults 0 0

# Shared memory.
none /System/Kernel/Devices/shm tmpfs defaults 0 0

# Swap partition.
/System/Kernel/Devices/hda3 none swap pri=1 0 0

# Root partition.
/System/Kernel/Devices/hda6 / auto defaults 1 0

/System/Kernel/Devices/hda1 /Mount/HDA1 auto defaults 1 0
/System/Kernel/Devices/hda4 /Mount/HDA4 reiserfs defaults 1 0
/System/Kernel/Devices/hda5 /Mount/HDA5 reiserfs defaults 1 0
/System/Kernel/Devices/hda7 /Mount/HDA7 reiserfs defaults 1 0
/System/Kernel/Devices/hda8 /Mount/HDA8 auto defaults 1 0
/System/Kernel/Devices/hda9 /Mount/HDA9 reiserfs defaults 1 0

# CD-ROM drive.
/System/Kernel/Devices/hdd /Mount/CD-ROM auto ro,noauto,users,exec 1 0

/System/Kernel/Devices/hdc /Mount/CDFS iso9660,cdfs noauto,users,ro 0 0

#USB
/System/Kernel/Devices/sda1 /Mount/PEN vfat noauto,users 1 0
```

Az első két sor amit a fenti listában látsz, nem az fstab-om része, hanem az

első azt mutatja, hogy elmentem a megfelelő könyvtárba (erre szolgál a `cd` parancs, ami az angol „change directory”, azaz „könyvtárváltás” rövidítése), a másik parancs pedig – annak ellenére, hogy az angol „cat” szó macskát jelent (látszik, hogy a Linux egy állatbarát rendszer... bizony, nemcsak felhasználóbarát, de állatbarát is...) – nem cicusokkal foglalkozik, hanem kilistázza a megadott állományt a képernyőre! (Gondolom, a „**catalog**” rövidítése lehet...) Íme, a Linux az állatszeretők oprendszere: nemcsak langusztánk van, meg egerünk, meg kölyökpingvinünk, hanem macskánk is! Van itt minden, épp csak vírusokat nem találunk majd... ez kérem egy steril rendszer! (Amíg ki nem nyitjuk az Ablakokat, hogy bejöjjön mindenféle szemét...)

Hogy miért a `/System/Settings` könyvtárban van az `fstab` állomány, ha korábban azt írtuk, hogy a `/etc` könyvtárban kell(ene) lennie? Erről később szólok majd, abban a fejezetben, amiben a GoboLinux felépítéséről ejtek pár (sok...) szót, hogy miért olyan, amilyen.

Na tehát most nem ússza meg a Tisztelt Olvasó, most „gondolkodnia lesz muszáj”, hogy ilyen ékes magyarsággal fejezzem ki magamat. Össze kell hasonlítani a saját `fstab` állományát (célszerűen mc, azon belül F4 gomb...) a fenti példával, elvégeznie a megfelelő változtatásokat, figyelembe véve persze azt, hogy ugye nem biztos, hogy ugyanannyi partíciója van, mint nekem!

Sóval s tetvvel, lássuk a medvét!

Kezdjük azzal, hogy a #-el megjelölt sorok úgynevezett „commentek”, azaz megjegyzések, olyan mintha nem is lenne egy ilyen sor, a rendszer nem veszi figyelembe. Tehát ha az `fstab`-odban valami sor nem kell, óvatosan elég ha egy ilyen kettőskeresztet teszel az elejére, így könnyen „visszacsinálhatod”, ha rájössz, hogy de mégis kell.

A `Proc`, `Sys`, `Pts` filesystem-eket nem kell piszkálnunk, úgysem tudnánk jobban beállítani, mint a gép magának. Az USB filesystem nekem nem működött úgy, ahogyan az „gyárilag” be lett állítva, emiatt kikommenteztem, s az `fstab`-om végére beírtam másként – most jó. A „swap partition”-nál ellenőrizzük le, hogy a megfelelő partíció van-e swap-nak kijelölve. A „Root partition” természetesen az úgynevezett „gyökérpartíciót” jelenti, azt tehát, amiről a rendszerünk indul. Mint látható, nálam ez jelen esetben a hatos, azaz a `hda6`. Nálad a példánk szerint a `hda4` kell ott szerepeljen.

A további sorok a többi partíció adatait tartalmazzák. Egy sor felépítése olyan, hogy az első mező meghatározza a partíció nevét (helyét), hogy melyiknek az adatait adjuk meg. A második mező a csatolási pontot írja le, ezután következik a fájlrendszer típusa – ha ez nincs megadva, akkor a rendszer maga próbálja azt felderíteni. Az én `fstab`-omban látunk példát erre is, és arra is, hogy megadjuk közvetlenül a típust, jelen esetben azt, hogy az `reiserfs`. Ezután mindenféle egyéb opciókat lehet még megadni, amiket nem nagyon akarok kirészletezni, mert többnyire feleslegesek (és függhetnek az adott

fájlrendszertől is). Egyre térek csak ki: meg lehet határozni az úgynevezett „felhasználói jogosultságokat”. Hogy például kinek mihez van joga azon a partíción. Ha ide azt írjuk: defaults, akkor a rendszer által alapértelmezettnek tekintett beállításokat állítja be, ami nekünk nem okvetlenül jó! Például, tegyük fel, hogy van egy FAT típusú partíció a gépünkön. Ha ez mondjuk a hda8, ennek nagy valószínűséggel a következő sort kreálja a GoboLinux:

```
# Windows partition.  
/System/Kernel/Devices/hda8 /Mount/HDA8 vfat defaults 1 0
```

No de ennek rögvest az lesz a következménye, hogy bár arról a partícióról tudunk majd olvasni, ám nem tudunk onnét fájlokat törölni. Ha azt akarjuk, hogy e partíción is jogunk legyen „normál felhasználóként” mindenhez, e sort vegyük be az fstab-ba:

```
# Windows partition.  
/System/Kernel/Devices/hda8 /Mount/HDA8 vfat umask=000 1 0
```

A csatolási pontoknak megfelelő könyvtárakat természetesen létre kell hozni (a /Mount alkönyvtárban), illetve nyugodtan törölhetjük onnan azokat, amelyek feleslegesek.

Visszatérve a root partícióhoz: ajánlatos lesz elmenni abba a partícióba, amibe a klón-rendszerünket átmásoltuk, s az ottani **/System/Settings** könyvtárban levő **fstab** állományt is szerkeszteni, úgy, hogy e sorban azt a partíciót adjuk meg root partícióként, amin a rendszermásolat csücsül! (Persze csak akkor, ha működtetni is akarjuk, mert ha tényleg csak egy egyszerű biztonsági másolat, akkor nem számít).

Látható, hogy az USB eszközöknek a neve Linux alatt **sda**, és **vfat** formátumot adtam meg, gondolván rá, hogy egy pendrive az szokott lenni, mert azt Windows alatt is lehet akkor használni. Megadható olyan paraméter is, hogy „auto” illetve „noauto”. Ez nem arra vonatkozik, hogy van-e idehaza autónk, hanem arra, hogy próbálja-e meg rendszerindításnál a rendszer automatikusan csatolni az adott partíciót. Pendrive esetén mint láttuk, ezt nem tartom szükségesnek. Az „users” paraméter azt jelenti, hogy rendszergazdai jogosultságok nélkül is engedélyezem az eszköz felcsatolását, tehát „normál” usereknek.

A CD-ROM eszközök esetén azt kell megemlítenem, hogy nekem ebből 2 van: egy CD-író/olvasó ami a **hdc**-re van dugva, és egy CD-DVD író-olvasó, ami a **hdd** eszközöm. A „ro” azt jelenti, hogy az eszköz csak olvasható. A cdfs jelentését máshol magyarázom majd el – mindenesetre elhagyható (az előtte levő vesszővel együtt).

Most tehát el kell végezd a megfelelő beállításokat itt, meg létrehoznod a

csatolási pontoknak megfelelő könyvtárakat, azután indítsd újra a rendszert, de már a merevlemezzről! Ha minden jól megy, gratulálhatsz magadnak. Ez komoly fejezet volt, ahol *nagyon* észnél kellett lenned.

Szerencsére olyan igazán nagy baj akkor sincs, ha bár ezt elrontottad, de legalább a gyökérpartíciód jól megmaradt. Maximum nem látod a többi partíciót, kísérletezgethatsz vele akkor tovább, amíg jó nem lesz minden.

Megjegyzendő, hogy a sima **mount** parancs kilistázza neked a felcsatolt partíciókat. Efféle listát kapsz eredményül:

```
vz@Csiszilla /System/Settings]mount
/System/Kernel/Devices/hda6 on / type auto (rw)
none on /System/Kernel/Objects type sysfs (rw)
tmpfs on /System/Kernel/Devices type tmpfs (rw)
devpts on /System/Kernel/Devices/pts type devpts (rw)
none on /System/Kernel/Devices/shm type tmpfs (rw)
/System/Kernel/Devices/hda1 on /Mount/HDA1 type reiserfs (rw)
/System/Kernel/Devices/hda4 on /Mount/HDA4 type reiserfs (rw)
/System/Kernel/Devices/hda5 on /Mount/HDA5 type reiserfs (rw)
/System/Kernel/Devices/hda7 on /Mount/HDA7 type reiserfs (rw)
/System/Kernel/Devices/hda8 on /Mount/HDA8 type reiserfs (rw)
/System/Kernel/Devices/hda9 on /Mount/HDA9 type reiserfs (rw)
```

Látható, kiírja, hogy mi hová milyen filerendszerrel van felcsatolva. Az (rw) azt jelenti: read/write, azaz írásra-olvasásra is engedélyezett.

(Az ne zavarjon meg, hogy szerepel a sorok közt a hda4-es partíció: e témát épp a hda6-ról indított Linux rendszeremből írom kivételesen, azaz nekem most a hda6 a gyökérpartícióm (root) és ide felcsatoltam a hda4-et).

Ezek után egyszerűbben csatolhatunk valamit. Például ha beteszünk egy CD-t, elég ennyit beírni:

```
mount /dev/hdc
```

azt ugyanis hogy HOVA csatolja az eszközt, már tudja a masinánk: kiolvassa az fstab-ból, hogy neki ezt a /Mount/CDFS tartalomjegyzékhez kell csatolnia.

Aki már használt valami másik Linux disztribúciót, most erre csóválhatja a fejét, hogy ez így nagyon nehézkes. Más disztrókbán megoldották, hogy amikor beteszünk egy CD-t, automatikusan csatolja azt a rendszer, még valami kis ikon is megjelenik róla a képernyőnkön! Miért nem lehet ezt megoldani a GoboLinux alatt?!

Bizonyára meg lehetne oldani, bár nem tudom hogyan, mert nem néztem utána. Nem érdekel ugyanis. Nem vagyok híve az „automount” szolgáltatásnak. Általában véve ki nem állhatom, ha a számítógép automatikusan végez dolgokat, anélkül, hogy tudnám mit csinál, anélkül, hogy én azt elrendeltem volna neki. Ezt tekintheted egyfajta „Kocka-filozófiának” is. Most persze erre

mondhatja valaki, hogy de hiszen nyilván azért teszünk be egy CD-t, hogy használjuk azt! Ahhoz úgyis fel kell mountolni!

Igen, ez többnyire igaz, de azért nem mindig. Az is lehet, hogy amiatt teszem be, hogy ezután arról indítsam a rendszert. Vagy az egy újraindítási CD, emiatt nem is kell csatolni, mert épp az a célom, hogy írjak rá valamit. Vagy ha csatolni akarom is de máshová, mint ahová általában szoktam. Továbbá: az efféle automatikus szolgáltatások elkényelmesítik a felhasználót, a végén már el is felejt, mi az a mount parancs, s emiatt ha egyszer olyan disztribúció elé kerül, ahol nincs automount, nem is fog tudni vele mit kezdeni. Általában véve az automatikus szolgáltatások nemcsak biztonsági szempontból veszélyesek, de amiatt is, mert konzerválják a Felhasználó tudatlanságát, azaz így soha nem is lesz belőle igazi *profî*, azaz Kocka!

Akkor most a következő fejezetben megismerkedünk a GoboLinux speciális eszköztárával, ami a csomagok (magyarán: progik) telepítését/eltávolítását végzi. Ez remélhetőleg fontos fejezet lesz, ugyanakkor remélem egyszerűbb, mint ez az iméti.

Programok telepítése és eltávolítása

Ez egy nagyon fontos fejezet. A programok telepítése más disztrók esetén is fontos téma, de a GoboLinux esetén pláne, mert az egyetlen CD-re csak kevés progí fér fel.

Természetesen ezt is mind „parancssorból” intézzük el. Ugyan van egy speciális grafikus program e célra, az a neve, hogy „Manager”, de előre bocsátom, e könyvben azt soha nem fogjuk használni. Sőt, én magam sem használom soha. Nincs rá ugyanis szükség. GoboLinux alatt vannak a progik telepítésére és eltávolítására speciális szkriptek (azaz – mondjuk kissé elnagyoltan - „programok”), és a „Manager” nevű grafikus progí semmi más, mint a parancssori szkripthívogatást elfedő egyfajta „frontend”, azaz „grafikus előtét”. Valójában ő is pontosan azokat a rutinokat hívogatja, amiket mi fogunk a parancssorból.

Nem kell megijedni, kevesebb mint egy tucat parancsról van csak szó, de még ezekből is az esetek 99%-ában csak kettőt használunk majd. Ezek pedig az

InstallPackage

és a

Compile

parancsok.

Az **InstallPackage**-vel úgynevezett „**bináris csomagokat**” telepíthetünk, a **Compile** pedig arra szolgál, hogy **forrásprogramokat** töltsünk le, s ezeket lefordítsa végrehajtható állománnyá (azaz épp az előbb említett bináriszá) a masinánk. Ügyeljünk rá, hogy – bár ezt egypárszor már írtam, de megemlítem – nem mindegy, hogy a kiadott parancsok nevét hogy írjuk, azaz ne tévesszük el a kis- és nagybetűket!

Természetesen az InstallPackage és a Compile is egy-egy program, amik maguk is telepítve vannak a rendszerünkbe valahová. Más Linux disztribúciókkal ellentétben a GoboLinux könyvtárszerkezete nagyon logikus. Például az összes program a **/Programs** mappából „nyílik”. Itt van egy könyvtára egy adott programnak, ezen belül egy mappa a megfelelő verziószámmal, s ez alatt a program összes saját tartalomjegyzéke. Más könyvtáraink is vannak a GoboLinux alatt, ezek teljes jegyzékét most felsorolom a funkciókkal együtt, hogy meglegyen egy helyen, áttekintés végett:

A GoboLinux könyvtárszerkezete

A GoboLinux filerendszer gyökerénél hat tartalomjegyzék van: Programs, Users, System, Files, Mount és Depot. Magyarázatukat lásd lentebb.

- **/Programs/** - ez a könyvtár egy-egy alkönyvtárat tartalmaz mindegyik programhoz, ami telepítve lett a számítógépre. Mindegyik efféle alkönyvtár egy vagy több további alkönyvtárat tartalmaz, az adott

program különböző verzióihoz (Pld /Programs/OpenOffice/1.0.3 és /Programs/OpenOffice/2.0.1) és opcionálisan lehet benne egy *Settings* és *Variable* alkönyvtár is. A *Settings* az adott program verziófüggetlen beállításait tartalmazza. Példák: /Programs/Bash/3.0/bin/bash és /Programs/Xorg-Server/Settings/X11/xorg.conf

- **/Users/** - ez a tartalomjegyzék tartalmazza a felhasználók „Home” könyvtárait, úgyhogy egy - mondjuk - „bob” nevű felhasználó „Home” könyvtára az /Users/bob útvonalon lenne elérhető.
- **/System/** - Kritikus rendszerállományok. Leginkább a létfontosságú rendszeralkalmazások használják (például /System/Settings/passwd) és a GoboLinux szkriptek (például /System/Links).
 - **Links/** - Olyan könyvtárakat tartalmaz, amik a /Programs alatti fájlokat indexelik.
 - **Environment/** - Linkek a környezeti állományokra. Ezeket egy Cache fájlba fordítják be, és a shell tölti be, miközben a programok betöltik a saját környezeti változóikat.
 - **Executables/** - Linkek a programok **bin** és **sbin** alkönyvtáraiban található fájlokhoz. (Tehát az ún. „végrehajtható” állományokhoz).
 - **Headers/** - ez a jegyzék tartalmazza a linkeket a programok fejléckönyvtáraiban található állományokhoz.
 - **Libraries/** - Linkek a programok **lib** könyvtárainak állományaira.
 - **Manuals/** - A *manuals* és *info* alkönyvtárak tartalma.
 - **Shared/** - Linkek a programok *share* könyvtáraiban található fájlokba.
 - **Tasks/** - Linkek programok *Resources/Tasks*-tartalomjegyzékeikből való indítórutinjaira.
 - **Settings/** - Konfigurációs fájlok és linkek a programok *Settings* könyvtáraiban található fájlokra.
 - **BootScripts/** - A rendszerindítás alatt használt szkriptek. Ez egy szimbolikus link a /Programs/BootScripts alatti Settings/BootScripts/ könyvtárra.
 - **Variable/** - a változó adatokat tartalmazó sokrétű könyvtár: ide kerülnek az ideiglenes állományok, spool file-ok, log állományok és a tranzien্স adatok.
 - **tmp/** - Ideiglenes adatok.
- **Kernel/** - A kernellel kapcsolatos tartalomjegyzékek.
 - **Boot/** - Az operációs rendszer indulása alatt használt programok és konfigurációs fájlok. Itt van elhelyezve a rendszermag és a rendszerbetöltő konfigurációs fájljai is.
 - **Devices/** - eszközfájlok (Az *Udev* kezeli őket).

- **Modules/** - Különböző kernelmodulokat tartalmaz, amiket a kernel változatok használnak.
- **Objects/** - A kernel-eszközökről ad képet (A 2.6 sorozatú kernelek sysfs fájlrendszerével együtt vezették be).
- **Status/** - Kernel státuszfájlok (a *proc* fílerendszer által használt).
- **/Files/** - A Files olyan strukturált adatokat tartalmaz, amiket bizonyos alkalmazások igényelnek, de nem szükségképpen képezik az ő részeit. Ilyenek a különböző önálló entitások, mint például a betűtípusok, kodekek és pluginok (és hasonló egyebek, melyek nem igényelnek csomagmenedzselést). Továbbá az alkalmazások úgy definiálhatják a saját alkönyvtáraikat, hogy helyszínspecifikus adatokat tároljanak – a Compile, a GoboLinux csomagkezelője, használja is ezt.
 - **Codecs/** - A multimédia-lejátszók, mint például az MPlayer, úgy konfiguráltatnak be, hogy ezen útvonalat használják, megtalálendő az ő külső kodekeiket.
- **Documentation/** - how-to dokumentumok
- **Fonts/** - Betűkészletek. Több alkönyvtárat is tartalmaz, mint például a **TrueType**. A fontkészletek útvonalait az X-hez megtaláljuk a következő állományokban: `/System/Settings/X11/xorg.conf` és `/System/Settings/fonts/fonts.conf`.
- **Plugins/** - pluginek, melyek például böngészőkhöz kellhetnek.
 - **Compile/** - Recipék, csomagok, források, stb. A Compile eszköz használja őket. ***Ha a rendszerünk partícióján már gyanúsán kevés a hely, akkor töröljük le az e könyvtár alkönyvtáraiban található állományokat és további alkönyvtárakat, mert valószínűleg teli van forrásprogramokkal és letöltött csomagokkal, amik eszméletlenül sok helyet képesek elfoglalni!***
- **/Mount/** - további helyi vagy távoli fájlrendszerek csatolási pontja. A tipikus alkönyvtárak ezen belül például a CD-ROM, Floppy és Zip.
- **/Depot/** - a felhasználók fájljainak tárolóhelye, ide kerülhetnek például a letöltött állományok. A /Depot tulajdonképpen egy szabad terület, hogy a dokumentumaidat, mint például a médiafájlokat, letöltött anyagokat, stb, tárolja. Gondolhatsz rá afféle „közösségi területként” is, mint afféle „HOME minden felhasználó számára”. (néhány UNIX-rendszernek van egy /pub könyvtára erre a feladatra). A GoboLinux mint rendszer tulajdonképpen figyelmen kívül hagyja a /Depot tartalmát, ez csak azért létezik, hogy a felhasználókat arra biztassa, hogy egyetlen helyen tárolják a mindenféle fájljaikat, és tartsák a rendszer többi részét tisztán. A /Depot -on belül nincsenek előre meghatározott alkönyvtárak, mindazonáltal része a szabványos GoboLinux fájlrendszer-hierarchiának.

Ezek után nézzük meg, honnét tudunk progikhoz jutni! Nos, bináris program aránylag kevés van a GoboLinuxhoz – de nincs is szüksége ilyesmire nagyon...

Figyelemre méltó azonban magyar felhasználók számára a

<http://gobolinux.info>

link, ahová én magam tettem fel néhány speciálisan a magyaroknak fontos, mert magyarított program bináris csomagját! De ide is feltettem efféle csomagokat:

<http://gobolinux.cwi.hu>

Mindjárt ezek közül a legfontosabb az OpenOffice, ami az MS-Office linuxos megfelelője. E progik már fel kell legyen telepítve a rendszeredre, csak hogy az angol változatban! Most ezt el fogjuk távolítani, és telepítjük a magyar változatot. Telepítjük az úgynevezett „JAVA environmentet” is, mert JAVA sok programhoz kell, speciálisan épp az OpenOffice némely funkciójához is!

A magyar OpenOffice telepítése

Mindenekelőtt le kell tölteni a

<http://linux.birodalom.net/gobowiki/index.php/Csomagok>

oldalról a megfelelő link alapján. Ezt természetesen meg lehet oldani úgy, hogy az ember rákattint a Firefoxban, aztán... Szóval, ahogy az ilyesmit szoktad. De „Igaz Kocka” megvetéssel fordul el az efféle módszertől! Az úgy cselekszik, hogy a böngészőben jobb klikk a link fölé, majd a linket a helyi menüből vágólappra másolja, „terminálba megy” (ugye a Konsole...), elnavigál abba a könyvtárba, ahová le akarja tölteni a progit (ez most célszerűen a saját *home* könyvtárad illik legyen, bár ehhez minden bizonnyal „navigálnod” sem kell, mert eleve ott indul neked a Konsole...), s ezután a **wget** progival tölti le a csomagot. A „wget” lényegében egy parancssoros „webszivattyú”, amivel akár egész honlapokat is lementhetsz, rekurzívan. Esméletlenül sok hasznos funkciója van. Most mi egyszerűen használjuk majd: beütjük hogy **wget** majd szóközt nyomunk, majd a **Shift+Insert** gombbal bemásoljuk a parancssorba a vágólapon levő linket. Ezután Enter. S íme, a wget letölti nekünk! Az egésznek itt a folyamata, íme:

```
vz@Csiszilla ~]wget http://linux.birodalom.net/gobocsomagok/openoffice/OpenOffice--2.3.0--i686.tar.bz2
--19:42:05-- http://linux.birodalom.net/gobocsomagok/openoffice/OpenOffice--2.3.0--i686.tar.bz2
=> `OpenOffice--2.3.0--i686.tar.bz2'
IP keresés linux.birodalom.net... 213.163.8.130
Connecting to linux.birodalom.net[213.163.8.130]:80... kapcsolódva.
HTTP kérés elküldve, várom a választ... 200 OK
Hossz: 138.078.789 (132M) [application/x-tar]

100%[=====>] 138.078.789 119.11K/s ETA 00:00

19:59:06 (132.07 KB/s) - `OpenOffice--2.3.0--i686.tar.bz2' saved [138078789/138078789]
```

Hasonlóképp töltsük le a JRE-1.5.0.08--i686.tar.bz2 csomagot is.

```
vz@Csiszilla ~]wget http://birobdalom.net/linux/gobocsomagok/JRE/JRE--1.5.0.08--i686.tar.bz2
--20:31:49-- http://birobdalom.net/linux/gobocsomagok/JRE/JRE--1.5.0.08--i686.tar.bz2
=> `JRE--1.5.0.08--i686.tar.bz2'
```

```
IP keresés birodalom.net... 213.163.8.130
Connecting to birodalom.net[213.163.8.130]:80... kapcsolódva.
HTTP kérés elküldve, várom a választ... 200 OK
Hossz: 26.146.807 (25M) [application/x-tar]
```

```
100%[=====] 26.146.807 144.24K/s ETA 00:00
```

```
20:35:00 (133.99 KB/s) - 'JRE--1.5.0.08--i686.tar.bz2' saved [26146807/26146807]
```

Ezután következik a rendszerünkben levő angol nyelvű OpenOffice eltávolítása. Ez ráadásul régebbi is, mint amit mi fel fogunk tenni, s ami magyar nyelvű lesz.

Programok eltávolításának tulajdonképpen két módszere is van a GoboLinuxban: egyik a „végleges”, amikor töröljük is a progit, ezt a **RemoveProgram** parancs végzi el. A másik a „kíméletesebb” módszer, amikor csak „letiltjuk” a nekünk nem kellő progit. Ennek megvan az a haszna, hogy könnyen vissza lehet hozni „az életbe”, ha rájövünk hogy hoppá, nekünk az mégis kéne! Mi most ez utóbbit használjuk. Mindenekelőtt váltsunk rendszer-gazdai módba, majd... de minek is írkalok ennyit, kéretik figyelemmel kísérni az alábbi eseményeket a konzolból idemásolt listán! Félkövérrel kiemelem a teáttalad írt (írandó) szövegeket!

```
vz@Csiszilla ~]su -
Jelszó:
pv@Csiszilla ~]DisableProgram OpenOffice
DisableProgram: Disabling OpenOffice, version 2.2.0.
DisableProgram: Removing links...
/System/Links/Executables/sbase
/System/Links/Executables/scalc
/System/Links/Executables/sdraw
/System/Links/Executables/smath
/System/Links/Executables/simpress
/System/Links/Executables/soffice
/System/Links/Executables/spadmin
/System/Links/Executables/swriter
/System/Links/Executables/spadmin.bin
/System/Links/Executables/soffice.bin
/System/Links/Shared/dtd
/System/Links/Shared/xdg
/System/Links/Shared/dict
/System/Links/Shared/xslt
/System/Links/Shared/basic
/System/Links/Shared/fonts/truetype
/System/Links/Shared/samples
/System/Links/Shared/autocorr
/System/Links/Shared/autotext
/System/Links/Shared/template
/System/Links/Shared/uno_packages
/System/Links/Shared/gallery
/System/Links/Shared/config/images.zip
/System/Links/Shared/config/psetupl.xpm
/System/Links/Shared/config/javavendors.xml
/System/Links/Shared/config/images_industrial.zip
/System/Links/Shared/config/images_crystal.zip
```



```
/System/Links/Shared/config/symbol
/System/Links/Shared/config/soffice.cfg
/System/Links/Shared/config/wizard
/System/Links/Shared/config/images_hicontrast.zip
/System/Links/Shared/config/webcast
/System/Links/Shared/config/psetup.xpm
/System/Links/Shared/Scripts
/System/Links/Shared/wordbook
/System/Links/Shared/readme
/System/Links/Shared/extension
/System/Links/Shared/registry
/System/Links/Shared/psprint
DisableProgram: OpenOffice 2.2.0 disabled.
pv@Csiszilla ~]
```

Most következik az, hogy elnavigálj abba a könyvtárba, ahová letöltötted a telepítendő csomagokat. Ez valószínűleg a `/Users/vz` könyvtár lesz. Vedd észre ugyanis, hogy mindig amikor a

su -

parancsot kiadod, átkerülsz egy másik könyvtárba, tudniillik a **pv** nevű rendszergazda könyvtárába! Ez elkerülhető lenne, ha csak a sima

su

parancsot adnád ki, leahagyván a mínuszjelet. Ekkor ugyanis csak simán rendszergazdai jogokat kapsz. A mínuszjeles változat annyiban másabb, hogy egyben (ideiglenesen) átírja az úgynevezett „környezeti változókat” is arra, amire azok rendszergazdai esetben beállítva kell legyenek, azaz ez épp olyan, mintha rendszergazdaként jelentkeztél volna be a rendszerbe. A rendszer afféle „megműtése” hogy programokat távolítunk el, illetve telepítünk, egy egész sereg rutin együttműködését kívánja meg, rendszergazdaként. Helyesebb „teljesértékű” rendszergazdaként végezni e feladatot. (Őszinténszólva, még nem próbáltam ki, sikerül-e hibamentesen telepíteni, ha csak a sima

su

paranccsal megyek rendszergazdai üzemmódba...).

Minthogy a „**su -**” miatt más könyvtárba kerültél át, vissza kell térned oda, ahol a csomagok vannak letöltve. Ekkor aztán telepítened kell őket az **InstallPackage** paranccsal. Íme:

```
pv@Csiszilla /Users/vz]InstallPackage OpenOffice--2.3.0--i686.tar.bz2
InstallPackage: Installing OpenOffice, version 2.3.0.
InstallPackage: Uncompressing to /Programs...
InstallPackage: Package signed with unknown signature. Unable to verify package
InstallPackage: Continue install?
InstallPackage: Press Enter to continue or Ctrl-C to cancel.
```

Ez a kérdés azt jelenti, hogy a csomag ismeretlen digitális aláírással van ellátva, és emiatt nem tudja a csomagot ellenőrizni. Kérdezi, folytassa-e a telepítést. A folytatáshoz Entert kell nyomni, a megszakításhoz Ctrl-C -t. Nyugodtan nyomj Entert.

```

UpdateSettings: No settings exist
SymlinkProgram: Symlinking OpenOffice 2.3.0.
SymlinkProgram: Symlinking global settings...
SymlinkProgram: Symlinking tasks...
SymlinkProgram: Storing variable files...
SymlinkProgram: Symlinking libraries...
SymlinkProgram: Symlinking headers...
SymlinkProgram: Symlinking info...
SymlinkProgram: Updating info dir...
SymlinkProgram: Symlinking manuals...
SymlinkProgram: Symlinking executables...
SymlinkProgram: Processed 7 files.
SymlinkProgram: Symlinking wrappers...
SymlinkProgram: Symlinking shared...
SymlinkProgram: Processed 32 files.
SymlinkProgram: Removing unused directories...
SymlinkProgram: Done.
InstallPackage: Checking dependencies...

```

Most következik JRE, tehát a JAVA (JAVA Runtime Environment) telepítése:

A JAVA telepítése

```

pv@Csiszilla /Users/vz]InstallPackage JRE--1.5.0.08--i686.tar.bz2
InstallPackage: Installing JRE, version 1.5.0.08.
InstallPackage: Uncompressing to /Programs...
InstallPackage: No signature. Unable to verify package
InstallPackage: Continue install?
InstallPackage: Press Enter to continue or Ctrl-C to cancel.

UpdateSettings: No settings exist
SymlinkProgram: Symlinking JRE 1.5.0.08.
SymlinkProgram: Symlinking global settings...
SymlinkProgram: Symlinking tasks...
SymlinkProgram: Storing variable files...
SymlinkProgram: Symlinking libraries...
SymlinkProgram: Processed 64 files.
SymlinkProgram: Updating library database (ldconfig)...
SymlinkProgram: Symlinking headers...
SymlinkProgram: Symlinking info...
SymlinkProgram: Updating info dir...
SymlinkProgram: Symlinking manuals...
SymlinkProgram: Processed 14 files.
SymlinkProgram: Symlinking executables...
SymlinkProgram: Processed 16 files.
SymlinkProgram: Symlinking wrappers...
SymlinkProgram: Symlinking shared...
SymlinkProgram: Removing unused directories...
SymlinkProgram: Done.
InstallPackage: Checking dependencies...

```

Fontkészletek telepítése

Fel van tehát telepítve a két progi. No most, van egy kis bibi a dologban. Az OpenOffice valami érthetetlen okból saját, külön könyvtárban óhajtja tárolni a **TrueType** fontokat. Véleményem szerint abszolút felesleges, hogy a rendszerben két külön helyen tárolj **TTF** fontokat. Elintézzük, hogy az OpenOffice is ugyanazokat a fontkészleteket lássa és használja, mint az összes többi progi!

Az OpenOffice truetype könyvtárát a

`/Programs/OpenOffice/2.3.0/Shared/fonts/truetype` link alatt leled meg. No most ebben a telepítés után mindössze egyetlenegyszem TTF fontkészletet találsz. Az mc segítségével ezt másold át a `/Files/Fonts/TrueType` mappába! Ezután (szintén az mc-vel) töröld le az OpenOffice könyvtárból ezt a truetype nevű mappát! (De CSAK ezt!) Ha pedig ezzel megvagy, lépj ki az mc-ből, és ad ki ezt a parancsot:

```
pv@Csiszilla /Programs/OpenOffice/2.3.0/Shared/fonts]ln -s /Files/Fonts/TrueType truetype
(Ehhez persze legyél abban a mappában, ahonnet a letörölt könyvtár nyílott...)
```

Ha úgy léptél ki az mc-ből, hogy a

`/Programs/OpenOffice/2.3.0/Shared/fonts` könyvtárban vagy, akkor itt most létre kellett jöjjön egy „truetype” nevű úgynevezett „szimbolikus link”, röviden **symlink**, ami a `/Files/Fonts/TrueType` mappára mutat. A szimbolikus linkek némileg hasonlóak a Windows úgynevezett „parancsikonjaihoz”, bár nem ikonok, és nem csupán végrehajtható állományra mutathatnak. Egy symlink, az tulajdonképpen egy másik állomány „helyettesítő neve”. Ezek után ha az OpenOffice az ő truetype mappáját keresi, erre a kis szimbolikus linkre talál rá, ami azt mondja majd neki, a programnak, hogy „hé öreg, helyettem használd azt a másik könyvtárat, aminek mindjárt mutatom a címét...”

A fentiekből rögtön rájöhetsz arra is, **hogyan telepítsd kedvenc Windowsos TTF betűkészleteidet** a GoboLinux alá: mindössze annyi a dolgod, hogy azokat bemásold a `/Files/Fonts/TrueType` könyvtárba... ennél egyszerűbben igazán nem lehet!

Mégvalamit meg kell oldanunk. Az említett könyvtárba telepített fontokról egyelőre sajnos nem vesz tudomást az úgynevezett Xorg szerver, ami egy olyan progi, ami általában a grafikus megjelenítést „műveli” a számítógépünkön. Az ugyanis a TrueType fontokat a `/Files/Fonts/TTF` könyvtárban keresi, de olyan könyvtárunk momentán nincs. E probléma kijavítására belenyúlhatnánk az Xorg progi konfigurációs fájljába (lesz is példa rá a későbbiekben, hogy belenyúlunk!), de sokkal egyszerűbb, ha a következő parancsot adjuk ki (rendszergazdaként), úgy, hogy a `/Files/Fonts` mappában vagyunk épp:

```
pv@Csiszilla /Files/Fonts]ln -s TrueType TTF
```

Ezzel csináltunk egy TTF nevű szimbolikus linket a TrueType könyvtárra. Most már ez is oké.

Egy **FONTOS** tudnivaló! Én is csak most futottam bele, miután megírtam a fejezetet, és sokat kinlódtem vele, mire rájöttem. Bár nem mindig, de néha előfordulhat, hogy bár bemásoltad a TTF fontkészletet a könyvtáradba, de az nem működik – ezalatt azt kell érteni, hogy az OpenOffice betűkészlet-listájában látod ugyan a fontkészlet nevét, ki is tudod választani, de a progik mégsem változtatja meg arra a betűtípusra a kijelölt szöveget. Nos, ekkor csináld azt, hogy szépen bezárod az OOo-t, és RENDSZERGAZDAKÉNT kiadod a

fc-cache -force

parancsot. Ezután már jó lesz. (Nem kell megijedni, ezt nem kell kiadni a fontkészlet minden használata előtt, csak a telepítés után egyszer...)

Ez tehát a bináris csomagok telepítésének módja: letöltöd, elmész a könyvtárba ahová a tömörített bináris csomag le lett töltve, aztán az InstallPackage paranccsal telepíted. Telepítés után a letöltött tömörített csomag(ok) törölhető(k).

Váltás a különböző verziójú programok között

Ha most elmész a /Programs/OpenOffice mappába és megnézed azt, azt látod, hogy ott 2 alkönyvtár is van: az egyik a 2.2.0 a másik a 2.3.0 nevet viseli. Van egy **Current** nevű link is, mely az utóbbira mutat. No most ez azt jelenti, hogy a rendszeredbe tulajdonképpen 2 különböző verziójú OpenOffice progik van integrálva, de ezek közül az aktuális a 2.3.0 verziójú (hiszen erre mutat a Current link). Ott van azért a másik is, csak az nem működik!

Mit kéne tenned, ha valamiért mégis a régi, angol nyelvű változatot akarnád használni?

Csekélység. A következő parancsokat kéne kiadnod:

DisableProgram OpenOffice

(Ez letiltja az aktuális OpenOffice-ot).

SymlinkProgram OpenOffice 2.2.0

(Ez pedig megmondja a GoboLinuxnak, hogy hozza létre a megfelelő szimbolikus hivatkozásokat erre az OpenOffice-változatra, mert ezentúl ezt kell használnia).

Ha aztán rájössz, hogy de neked mégis az újabb kell, ennyit csinálsz:

DisableProgram OpenOffice

SymlinkProgram OpenOffice 2.3.0

Azt hiszem világos minden – két parancs mindössze!

Ha végleg meg akarsz szabadulni egy letiltott progitól, egyszerűen töröld a megfelelő – korábban már letiltott - verzió alkönyvtárát. Vagy használd e parancsot:

RemoveProgram OpenOffice 2.2.0

Ez nyom nélkül kitörli a rendszeredből a régi OpenOffice-ot. Gondolom világos a parancs paraméterezése: a törlendő progí nevét várja és a verzióját.

Na de hogyan használd az OpenOffice-odat, hiszen ez nem szerepel a KDE grafikus menüjében!

Semmi vész. Mindenekelőtt elindíthatod parancssorból is a szövegszerkesztőt az

swriter

parancs kiadásával. Na de ha ehhez nincs kedved, akkor egyszerűen használd a (grafikus menüből elérhető) „Menüszerkesztő” funkciót a KDE-ben! Ennek kezelését nem írom le, könnyű rájönni. Minden magyarul van benne... Azt írom csak le, hogy az OpenOffice több részből áll, mert nemcsak szövegszerkesztővel rendelkezik, de CorelDraw-szerű vektorgrafikus rajzolóprogival is, meg egyéb effélékkel. Ezeknek meg kell add az elérési útját a menüszerkesztőben, nos ezeket a következő útvonalakon leled, ezeket „huzalozd be” a menübe:

/Programs/OpenOffice/2.3.0/bin/sbase
/Programs/OpenOffice/2.3.0/bin/scalc
/Programs/OpenOffice/2.3.0/bin/sdraw
/Programs/OpenOffice/2.3.0/bin/simpress
/Programs/OpenOffice/2.3.0/bin/smath
/Programs/OpenOffice/2.3.0/bin/soffice
/Programs/OpenOffice/2.3.0/bin/swriter

Hogy melyik mire való, nem ismertetem: indítsd el, és meglátod! E könyvnek nem célja OpenOffice tanfolyamot tartani.

Programok másolása GoboLinux disztribúciók között

Az előbb olvasottakból már kikövetkeztethető, hogy ha nekünk van két (vagy több...) GoboLinux rendszerünk telepítve a merevlemezünkre, mondjuk egyik a HDA4-en, a másik a HDA7-en, akkor miként kell egyikből átmásolni a másikba egy programot. Mondjuk a HDA4-ről a HDA7-re. Mert ugye, akár eszméletlenül hosszú időbe is telhet egy progí lefordítása, s nem akarjuk ezt eljátszani mindkét rendszernél! Például telepítettük mondjuk a HDA4-re a

KedvencProgi programot. Nos, semmi más dolgunk nincs, mint hogy a `/Programs/KedvencProgi` könyvtárat átmásoljuk a `/Mount/HDA7/Programs/` könyvtárba, így ott is létrejön a **KedvencProgi** tartalomjegyzék. Ezután kilépünk a rendszerből, „bebutulunk” a HDA7-en levő rendszerünkbe, s rendszergazdai jogkörben (azaz e könyv példája szerint **pv** felhasználóként...) kiadjuk e parancsot:

SymlinkProgram KedvencProgi

Ennyi az egész! Persze, arra azért ügyeljünk, így tényleg csak ezt az egy programot vittük át a másik partícióra, a FÜGGŐSÉGEIT, azokat NEM...

A GoboLinux rendszereszközeinek frissítése

Mielőtt megismerkednénk a forrásból való telepítés eszközeivel, szólnunk kell róla, hogy amióta megalkották a GoboLinux készítői a GoboLinux 014 -et, azóta természetesen tökéletesedtek a különböző speciális GoboLinux-eszközök, tehát az **InstallPackage** és a **Compile** progik is. Ezek közül a **Compile** egy külön programként van megalkotva, az **InstallPackage** viszont a „**Scripts**” nevű csomag része, amiben még több hasznos mindenféle is van. No most tehát, ezen progik legújabb verziójával te nem rendelkezel, ezokból ezeket most helyes lesz, ha letöltöd és felinstallálod a rendszeredbe, hogy az a lehető legújabb, legnaprakészebb legyen!

Szerencsére nem kell tudnod róla, honnét szerezhetőek ezek be. Az **InstallPackage** paranccsal telepítheted ezeket, s e parancs van olyan értelmes, hogy ha a megadott állománynak nem találja meg a csomagját abban a könyvtárban, ahol épp állsz a parancs kiadásakor, akkor körülnéz az Interneten, a GoboLinux disztribúció csomagbázisában, hogy létezik-e ott a megadott progik! Ha igen, letölti annak legújabb változatát, és telepíti a gépedre. Telepítés előtt neked célszerű letiltani a régi változatot (a **DisableProgram** paranccsal), de speciálisan épp a **Scripts** és a **Compile** parancsok esetében **nem**. Mindkét parancsra jellemző, hogy ha a program neve után verziószámot is adsz meg, akkor nem a legújabbat telepíti, hanem az adott verziójút. Tehát mondjuk egy **XXX** nevű progik esetén az

InstallPackage XXX

a legújabb **XXX** progik telepíti a gépedre (ha az megvan a GoboLinux csomagbázisában), míg az

InstallPackage XXX 0.3.17

parancs a 0.3.17 verziójút, akkor is, ha annál van már akár jelentősen újabb is.

Ugyanez igaz a **Compile** parancsra is szintaktikailag.

A Scripts csomag frissítése

Most tehát ellenőrizd, hogy rendszergazdai jogosultságokkal rendelkez-e a parancssorban (azaz „pv” felhasználó vagy-e, és piros színű-e a promptod), majd add ki a következő parancsot:

InstallPackage Scripts

A dolog fontosságára való tekintettel idemásolom, amit látsz:

```
pv@Csiszilla ~]InstallPackage Scripts
InstallPackage: Locating a binary package for Scripts ...
InstallPackage: Installing
http://kundor.org/gobo/packages/official/Scripts--2.8.3--i686.tar.bz2
InstallPackage: Press Enter to continue or Ctrl-C to cancel.

InstallPackage: Downloading package to /System/Variable/tmp/Scripts--2.8.3--
i686.tar.bz2.
--09:22:58-- http://kundor.org/gobo/packages/official/Scripts--2.8.3--
i686.tar.bz2
=> `/System/Variable/tmp/Scripts--2.8.3--i686.tar.bz2'
IP keresés kundor.org... 134.53.148.149
Connecting to kundor.org|134.53.148.149|:80... kapcsolódva.
HTTP kérés elküldve, várom a választ... 200 OK
Hossz: 463.986 (453K) [application/x-tar]

100%[=====>] 463.986 56.44K/s ETA 00:00

09:23:08 (44.35 KB/s) - `/System/Variable/tmp/Scripts--2.8.3--i686.tar.bz2'
saved [463986/463986]

InstallPackage: Installing Scripts, version 2.8.3.
InstallPackage: Uncompressing to /Programs...
InstallPackage: Package Signature Verified
SymlinkProgram: Symlinking Scripts 2.8.3.
SymlinkProgram: Symlinking global settings...
SymlinkProgram: Processed 7 files.
SymlinkProgram: Symlinking tasks...
SymlinkProgram: Storing variable files...
SymlinkProgram: Symlinking libraries...
SymlinkProgram: Conflict: /System/Links/Libraries/python2.3/site-
packages/PythonUtils.pyc
SymlinkProgram: Conflict: /System/Links/Libraries/python2.3/site-
packages/GetAvailable.pyc
SymlinkProgram: Conflict: /System/Links/Libraries/python2.3/site-
packages/GuessProgramCase.pyc
SymlinkProgram: Conflict: /System/Links/Libraries/python2.3/site-
packages/FindPackage.pyc
SymlinkProgram: Conflict: /System/Links/Libraries/python2.3/site-
packages/GuessLatest.pyc
SymlinkProgram: Processed 11 files.
SymlinkProgram: Updating library database (ldconfig)...
SymlinkProgram: Symlinking headers...
SymlinkProgram: Symlinking info...
```

```

SymlinkProgram: Updating info dir...
SymlinkProgram: Symlinking manuals...
SymlinkProgram: Symlinking executables...
SymlinkProgram: Processed 67 files.
SymlinkProgram: Symlinking wrappers...
SymlinkProgram: Processed 1 file.
SymlinkProgram: Symlinking shared...
SymlinkProgram: Processed 15 files.
SymlinkProgram: Symlinking environment...
SymlinkProgram: Removing unused directories...
SymlinkProgram: Done.
InstallPackage: Removing downloaded package
/System/Variable/tmp/Scripts--2.8.3--i686.tar.bz2.
pv@Csiszilla ~]

```

Amikor azt írja ki, hogy „Press Enter to continue or Ctrl-C to cancel.”, akkor természetesen Enter-t kell nyomnod!

A fenti listán látható, hogy a végén előzékenyen el is távolítja a letöltött tömörített Scripts csomagot, mert már nincs rá szükség.

A Compile csomag frissítése

Most a Compile legújabb változatát telepítsd. A teendők, szintén a teljes telepítést leköszölve ide neked:

```

pv@Csiszilla ~]InstallPackage Compile
InstallPackage: Locating a binary package for Compile ...
InstallPackage: Installing
http://kundor.org/gobo/packages/official/Compile--1.10.2--i686.tar.bz2
InstallPackage: Press Enter to continue or Ctrl-C to cancel.

InstallPackage: Downloading package to /System/Variable/tmp/Compile--1.10.2--
i686.tar.bz2.
--09:27:46-- http://kundor.org/gobo/packages/official/Compile--1.10.2--
i686.tar.bz2
=> `/System/Variable/tmp/Compile--1.10.2--i686.tar.bz2'
IP keresés kundor.org... 134.53.148.149
Connecting to kundor.org|134.53.148.149|:80... kapcsolódva.
HTTP kérés elküldve, várom a választ... 200 OK
Hossz: 79.255 (77K) [application/x-tar]

100%[=====>] 79.255          72.35K/s

09:27:47 (72.15 KB/s) - `/System/Variable/tmp/Compile--1.10.2--i686.tar.bz2'
saved [79255/79255]

InstallPackage: Installing Compile, version 1.10.2.
InstallPackage: Uncompressing to /Programs...
InstallPackage: Package Signature Verified
UpdateSettings: Current and default settings match
SymlinkProgram: Symlinking Compile 1.10.2.
SymlinkProgram: Symlinking global settings...
SymlinkProgram: Processed 1 file.

```



```

SymlinkProgram: Symlinking tasks...
SymlinkProgram: Storing variable files...
SymlinkProgram: Symlinking libraries...
SymlinkProgram: Processed 5 files.
SymlinkProgram: Updating library database (ldconfig)...
SymlinkProgram: Symlinking headers...
SymlinkProgram: Symlinking info...
SymlinkProgram: Updating info dir...
SymlinkProgram: Symlinking manuals...
SymlinkProgram: Symlinking executables...
SymlinkProgram: Processed 11 files.
SymlinkProgram: Symlinking wrappers...
SymlinkProgram: Symlinking shared...
SymlinkProgram: Processed 16 files.
SymlinkProgram: Removing unused directories...
SymlinkProgram: Done.
InstallPackage: Checking dependencies...
InstallPackage: Removing downloaded package
/System/Variable/tmp/Compile--1.10.2--i686.tar.bz2.
pv@Csiszilla ~]

```

Oké, hát most már nyugodtan alhatunk – ami a GoboLinux fejlesztőeszközeit illeti, vadi-új zsír-frissek vagyunk!

A forrásból való telepítés lényege

No most ideje megismerkednünk a **Compile** paranccsal. Jó tudni, hogy ahányszor csak valami progit telepíteni akarunk a GoboLinuxba, az esetek 99%-ában e parancsot fogjuk használni.

A szintaktikájáról különben nem kell sokat szólnom, említettem már, hogy az megegyezik azzal, amit megismertünk az InstallPackage esetében. Épp csak „ő”, a Compile, forrásból fordít.

Minden számítógépprogramnak van olyanja, hogy „forrás”. A Windowsos progiknak is, épp csak azt az esetek többségében nem közlik, mert akkor azt le lehetne fordítani a Felhasználónak odahaza, s akkor nem fizetne a progért... Windows alá tehát az esetek többségében a *binárist* telepítik.

Linux alatt ez szerencsére másként van. Alig akad példa rá, hogy a forrás ne lenne elérhető. Ebből aztán a különböző Linux-disztribúciók szoktak mindefféle bináris csomagokat készíteni a maguk rendszeréhez, ezek különböző kiterjesztésekkel bírnak, mint például .rpm, .deb, .uhu, stb. Általában mindegyik bináris csomag csak ahhoz a speciális Linux disztribúcióhoz jó. (Mondjuk ez így nem egészen igaz: akadnak csomagtípusok, melyek több disztróhoz is jók, de olyan nincs, ami MINDEGYIKHEZ megfelelő lenne). E kiterjesztéseket teljes nyugalommal felfoghatjuk úgy, mint az „.exe” linuxos megfelelőit. (Vannak ugye „önkicsomagolós” exe állományok is win alatt). No most miután rengetegféle Linux disztribúció van, teljesen logikus volt az ötlet, amit a GoboLinux készítői követtek: az egyetlen ami biztosan elérhető minden

progi esetében, az a forrás. Azt kell felhasználni. Azt szépen le kell tölteni a progi készítőinek honlapjáról, és lefordítani az otthoni számítógépen végrehajtható programmá. Igaz, ez némileg lassabb, mint egy előfordított bináris csomagot telepíteni, viszont a mi gépünkön lefordított progi legalább valamiképt gyorsabban működik majd (legalábbis némely esetekben).

Az ötlet kitűnő, mindössze két szépséghibája van: az nem valami kényelmes, hogy a Felhasználó elkezdjen utánanyomozgatni, honnan tölthető le egy progi forrása! Aztán meg, a GoboLinux architektúrája jelentősen különbözik egy „hagyományos” UNIX-szerű Linux-disztribúció architektúrájától, holott a progi készítői biztos nem számoltak a GoboLinux specialitásokkal!

Erre találták ki a GoboLinux készítői az úgynevezett „recipéket”. A recipe egy apró kis szöveges állomány. Amikor te telepíteni akarod a Compile segítségével mondjuk az XXX progit, akkor a Compile letölti a GoboLinux recipetárolójából a recipét ami ehhez a progihoz tartozik, s feldolgozza azt. E recipében van ugyanis leírva, honnan (milyen internetcímről) töltheti le az XXX progi forrását, s azt hogyan, milyen módon kell lefordítsa ahhoz, hogy az elkészült bináris progi aztán működjék a Felhasználó GoboLinux rendszerében. Az egész tehát teljesen automatikus – jó esetben. Időnként felléphetnek különböző komplikációk, főként a függőségek (angolul „dependencies”) miatt, *ami bizony a Linux operációs rendszer egyik legnagyobb rákfeneje*. Csúnya szóval élve, Linux alatt ezzel, a függőségi problémákkal fogsz a legtöbbet „szívni”. No nem kell megijedni, a dolog nem annyira vészes, mint a vírusok a Windows alatt, és nem is teszi tönkre a rendszeredet, maximum az derülhet ki, hogy bizonyos progit nem fogsz tudni használni. Ennek ellenére, merem állítani, a függőségek több problémát okoznak Linux alatt, mint az összes többi létező probléma együtt.

Na de mi is az, hogy „függőség”?!

A dolog furcsa módon amiatt létezik mint probléma, mert a Linux annyira kiváló operációs rendszer. Aki Linuxot használ, az minden vicc és túlzás nélkül a számítástechnika élvonalában jár! A létező legmodernebb progikat használhatja – ha akarja. Ezzel együtt jár természetesen, hogy a rendszere mégsem okvetlenül lesz hibátlan – ha valamiből mindig a legmodernebb izémizét használjuk, előfordulhat, hogy beleszaladunk az izémizé valami rejtett hibájába, ami még senki másnál „nem jött elő”, s így nincs kijavítva. Új dolgoknak mindig vannak kockázatai. Emiatt, különösen kezdő felhasználóknak, megfontolandó, hogy talán ne mindig a legújabb verzióját használják valami programnak.

Erre ügyelni szoktak a disztribúciók készítői, *emiatt amikor te feltelepítesz egy CD-ről vagy DVD-ről egy disztrót, abban minden bizonnyal nem a legmodernebb verziói vannak a programoknak. Ám amit oda beraktak, az nagy valószínűséggel megbízhatóan működik*. E telepített progikat frissítheted – erre láttunk példát az OpenOffice esetében: a CD-n a 2.2.0 verziójú volt, de lecseréltük a 2.3.0 verziójúra. (Én jelenleg már nem is azt használom, hanem

egy annál is újabbat, ami a készítői szerint is még fejlesztés alatt álló, bevallottan kísérleti példány...).

Elképzelhető azonban, hogy mondjuk a rendszeredben van egy XXX nevű progí, az ő 0.2.6 verziójával, s te frissíteni akarod a 0.3.8 változatra, mert abban van pár új, neked tetsző funkció. Ekkor váratlan meglepetések érhetnek. Mert lehet, hogy az XXX készítői ahhoz, hogy ezen újabb verzió működjék az új csicsákkal-funkciókkal, felhasználtak valami másik progít vagy annak egy részét, azaz feltételezik, hogy a rendszer alatt amiben az újabb verziójú XXX fut majd, jelen van egy YYY nevű progí is! És lehet hogy neked ez a progí nincs meg! Ekkor ezt is telepítened kell. Vagy épp neked megvan a rendszeredben az YYY progí, de annak csak a 2.6.8 verziója, míg az új XXX progí okvetlenül az YYY 2.7.0 változatát igényli! Ekkor bizony frissítened kell az YYY progít is, de az meg lehet, hogy a ZZZ és QQQ progik valami új változatait igényli, azok megint más akármiket, és így tovább a végtelenségig!

Hogy még jobban elszörnyesszelek, *ez még a jobbik eset*. Megtörténhet ugyanis, hogy te valami olyan progít akarsz a rendszeredbe integrálni, ami még egyáltalán nem volt meg. S mondjuk ez egy viszonylag régi progí. És az szól, hogy neki az YYY progí 0.4.3 verziója kéne. S neked van is YYY progíd, de egy jelentősen újabb, a 2.7.0 verziójú! No most ekkor megpróbálhatsz továbblépni az XXX progí fordításával, remélve, hogy az az újabb YYY progival is együtt tud működni. Szerencsés esetben sikerül is neked a fordítás, és működni is fog. Lehet azonban, hogy bár a fordítás működik, hiba nélkül lezajlik, a progí indítás után mégsem működik majd. Vagy már a fordítási folyamat is „elhalálozik”. Ilyenkor vagy lemondasz a vágyott XXX progí használatáról, vagy eltávolítod a rendszeredből az új YYY progít, s felteszed annak azt a régebbi verzióját, amit az XXX igényel. Ez viszont azt az óriási kockázatot rejt magában, hogy egy egész rakás progí ami most jelen van a rendszeredben, s mind számít az YYY épp ezen újabb változatának meglétére – ettől a pillanattól kezdve esetleg nem fog működni! Akár az is megtörténhet szélsőséges esetben, hogy amikor legközelebb újraindítod a rendszeredet, egyszerűen nem indul majd el a grafikus felület. (A parancsor valószínűleg igen. *Elvileg* az is összeomolhat, de *gyakorlatilag* nem jellemző rá. Nagyon kevés olyan progí van csak, aminek piszkálásával a parancssor tönkreteszhető. Ilyen például a **Glibc** – azt tehát jobb, ha **SOHA** nem piszkáljuk! A grafikus felület megóvása érdekében pedig a **GLib** progít ne maceráljuk). De még ha e kockázatot fel is vállalod, már az sem biztos hogy könnyű, hogy te lecseréld az új YYY progít a régebbire, mert lehet, hogy a régebbi YYY progí meg más akármiféle progikból igényel régebbi verziókat, mint amik a rendszeredben telepítve vannak.

A fentiek kellemetlen hírek, de ez úgy van, hogy egy Linux rendszer lényegében egyfajta „szoftver-alkatrészek” - programok – gyűjteménye, melyeket különböző „cégek”, tehát programozók fejlesztenek, a maguk elképzelései sze-

rint. Ezeket a disztribúció készítői egy aránylag jól együttműködő egészbe összeválogatják, de ha ezen te változtatsz, az a te saját felelősséged. S most képzelj el, hogy neked van egy Ferrari autód, és megtetszik neked egy Mercedes kocsi kormánykereke. Egyáltalán nem biztos, hogy azt be tudod szerelni működőképesen a te Ferrari kocsidba! Ha mégis, lehet, hogy emiatt ki kell cserélned az egész műszerfalat is! Ez nem azt jelenti, hogy az a kormánykerék rossz, azt sem jelenti, hogy a te kocsid rossz. Egyszerűen nem valók egymáshoz. Így van ez a Linux világában a programokkal is. Nem tudsz okvetlenül mindent használni minden disztribúció alatt. Néha nagyon komoly hekkerkedéssel életet lehet lehelni egy-egy nem odaváló programba is, de nem mindig, s az már nagyon nem kezdőknek való feladat.

Továbbá: nehogy azt hidd, hogy ez a probléma valami különleges Linux-specialitás! Pontosan ugyanez megvan – ha nem is ekkora mértékben talán – a Windows világában is. Köztudomású, hogy nem fut minden program minden Windows-változat alatt. Például akadnak progik, melyek futnak XP alatt, de nem futnak a Vista alatt, és fordítva. Sőt, bizonyos „szervízpakkok” (SP1, SP2, stb) feltevése után is abban a rendszerben egyes programok egyszerűen nem működnek. Ott is így van ez tehát, csak a probléma nem annyira feltűnő, mert Windowsból nincs annyi, mint Linux disztribúcióból.

Na de szerencsére ez azért nem mindennapos probléma. Kezdjünk is neki életünk első fordítási feladatának! Annyit bocsátok csak előre, hogy a függőségek miatt minden esetben közlöm az adott progiz verziószámát is!

A verziószámokról tudni kell, hogy többnyire háromjegyűek, **x.y.z** alakban megadva, s ha csak a legkisebb helyiértékű (**.z**) változik, az oly pici javítást jelent, hogy azt általában gond nélkül felrakhatod a rendszeredbe a régebbi (vagy akár újabb) ugyanolyan, de más verziójú program helyére. Ha az általam **x** vagy **y** -al jelzett verziószám változik, az már komoly gondot jelenthet – főleg az **x**...

S még egy infó: bár mint írtam, a Linux rendszer szigorúan különbséget tesz kis- és nagybetűk közt, az InstallPackage és a Compile programok a paramétereiket tekintve nem „case-sensitívek”, azaz nem kisbetű/nagybetű-érzékenyek. Elég nekik a progiz nevét csupa kisbetűsen is megadni, tudni fogják, mire gondolsz.

NumlockX

Mindjárt kezdetnek lefordítunk egy picike kis progit. Ez a **NumlockX 1.1-r2** verziója. Most kivételesen közlöm a fordítás teljes listáját, mert rövid, s hogy lássuk, miként néz ki egy efféle. A többi proginál illet már nem (mindig) teszek.

Ja és hogy ez a program mit csinál? Nos, ez fogja elintézni nekünk, hogy induláskor a NumLock azonnal bekapcsolt állapotban legyen! Engem rém-

ségesen idegesít, ugyanis, ha ez nem úgy van. Micsoda hülyeség, hogy ezért nekem meg kell nyomnom azt a gombot! Minden rendszerindításnál egy gombbal többet kelljen megnyomnom?! Na ne, ez már aztán igazán barbár nyomor! Legyen kényelmes a rendszerem!

Tehát a kiadandó parancs (rendszergazdaként!):

Compile numlockx

A fordítás menete:

```
pv@Csiszilla /Programs/OpenOffice/2.3.0/bin]Compile numlockx
Compile: Locating a recipe for numlockx ...
Compile: Found recipe for numlockx
GetRecipe: Trying to get http://gobolinux.org/recipe-store/NumlockX--1.1-r2--
recipe.tar.bz2
GetRecipe: Downloading recipe from http://gobolinux.org/recipe-
store/NumlockX--1.1-r2--recipe.tar.bz2
Compile: Recipe placed in /Files/Compile/Recipes/NumlockX/1.1-r2
Compile: Checking dependencies...
Compile: Compiling NumlockX version 1.1, revision r2.
--10:12:38-- http://ktown.kde.org/~seli/numlockx/numlockx-1.1.tar.gz
=> `numlockx-1.1.tar.gz'
Resolving ktown.kde.org... 131.246.120.250
Connecting to ktown.kde.org|131.246.120.250|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 75,777 (74K) [application/x-gzip]

100%[=====>] 75,777          138.96K/s

10:12:39 (138.56 KB/s) - `numlockx-1.1.tar.gz' saved [75777/75777]

Compile: Unpacking file /Files/Compile/Archives/numlockx-1.1.tar.gz...

PrepareProgram: Preparing...
PrepareProgram: Autoconf configure script detected.

checking for a BSD-compatible install... /System/Links/Executables/install -c
checking whether build environment is sane... yes
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ANSI C... none needed
checking for style of include used by make... GNU
checking dependency style of gcc... gcc3
checking for ranlib... ranlib
```

```

checking how to run the C preprocessor... gcc -E
checking for X... libraries /Programs/Xorg/7.2/lib, headers /Programs/Xorg/7.2/
include
checking for IceConnectionNumber in -lICE... yes
checking for libXext... yes
checking for egrep... grep -E
checking for ANSI C header files... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for unistd.h... yes
checking X11/XKBlib.h usability... yes
checking X11/XKBlib.h presence... yes
checking for X11/XKBlib.h... yes
checking for XkbLockModifiers in -lXext... yes
checking X11/extensions/XTest.h usability... yes
checking X11/extensions/XTest.h presence... yes
checking for X11/extensions/XTest.h... yes
checking for XTestFakeKeyEvent in -lXtst... yes
checking for Xsetup file... /usr/X11R6/lib/X11/xdm/Xsetup_0
checking for xinitrc file... /usr/X11R6/lib/X11/xinit/xinitrc
configure: creating ./config.status
config.status: creating Makefile
config.status: creating config.h
config.status: executing depfiles commands

```

```

/bin/make all-am
make[1]: Entering directory `/Files/Compile/Sources/numlockx-1.1'
if gcc -DHAVE_CONFIG_H -I. -I. -I. -I/Programs/Xorg/7.2/include -O2
-march=i686 -fomit-frame-pointer -pipe -Wall -pedantic -MT main.o -MD -MP -MF
".deps/main.Tpo" -c -o main.o main.c; \
    then mv -f ".deps/main.Tpo" ".deps/main.Po"; else rm -f
".deps/main.Tpo"; exit 1; fi
gcc -O2 -march=i686 -fomit-frame-pointer -pipe -Wall -pedantic -o numlockx -
L/Programs/Xorg/7.2/lib main.o -lXtst -lXext -lX11
make[1]: Leaving directory `/Files/Compile/Sources/numlockx-1.1'
Compile: Asserting that requirements are met...
SandboxInstall: unionfs is available. Using UnionSandbox.
SandboxInstall: Installing NumlockX...

```

```

make[1]: Entering directory `/Files/Compile/Sources/numlockx-1.1'
test -z "/Programs/NumlockX/1.1/bin" || mkdir -p --
"/Programs/NumlockX/1.1/bin"
/System/Links/Executables/install -c 'numlockx'
'/Programs/NumlockX/1.1/bin/numlockx'
make[1]: Nothing to be done for `install-data-am'.
make[1]: Leaving directory `/Files/Compile/Sources/numlockx-1.1'
UnionSandbox: Cleaning up.
UnionSandbox: Moving entries to: /Programs/NumlockX/1.1/.SandboxInstall_Root
SandboxInstall: Postprocessing Sandbox

```

```

Compile: Stripping executables...
UpdateSettings: Current and default settings match
SymlinkProgram: Symlinking NumlockX 1.1.
SymlinkProgram: Symlinking global settings...
SymlinkProgram: Symlinking tasks...
SymlinkProgram: Storing variable files...
SymlinkProgram: Symlinking libraries...
SymlinkProgram: Updating library database (ldconfig)...
SymlinkProgram: Symlinking headers...
SymlinkProgram: Symlinking info...
SymlinkProgram: Updating info dir...
SymlinkProgram: Symlinking manuals...
SymlinkProgram: Symlinking executables...
SymlinkProgram: Processed 1 file.
SymlinkProgram: Symlinking wrappers...
SymlinkProgram: Symlinking shared...
SymlinkProgram: Processed 1 file.
SymlinkProgram: Removing unused directories...
SymlinkProgram: Done.
Compile: Generating package's build information...
Generating dependency report. Please wait: |
Compile: Generating package's signature...
SignProgram: Creating FileHash
pv@Csiszilla /Programs/OpenOffice/2.3.0/bin]

```

XBindkeys

Rögvest lefordítjuk az XBindkeys 1.8.2 progit is. Ez nekünk nagyon fontos, mert ezzel érjük el, hogy mindenféle progikat billentyűzetről, gombnyomogatással indítsunk!

A fordítás elején azonnal függőségi problémába ütközünk. Íme a kapott konzol-output eleje:

```

pv@Csiszilla ~]Compile xbindkeys
Compile: Locating a recipe for xbindkeys ...
Compile: Found recipe for xbindkeys
GetRecipe: Trying to get http://gobolinux.org/recipe-store/XBindkeys--1.8.2-r1--recipe.tar.bz2
GetRecipe: Downloading recipe from http://gobolinux.org/recipe-store/XBindkeys--1.8.2-r1--recipe.tar.bz2
Compile: Recipe placed in /Files/Compile/Recipes/XBindkeys/1.8.2-r1
Compile: Checking dependencies...

CheckDependencies: Guile is not installed.
CheckDependencies: Compile recipe for Guile 1.8.3 or skip this dependency?
[C]Compile/[S]Skip/[CA]Compile All/[SA]Skip All
c

```

Itt ugye azt írta ki, hogy neki kéne egy Guile nevű progi, ami nincs installálva nekünk, és kérdi, hogy telepítse-e! Persze hogy ezt akarjuk, emiatt

megnyomtuk a „c” gombot, hogy azt is „Compilézze” le nekünk. Az „s” válasz azt jelentené, hogy a Guile-ot ne fordítsa le, hanem enélkül próbálja lefordítani nekünk az épp telepítendő progot (az XBindkeys-et), már ha tudja. A „ca” válasz azt jelenti, hogy minden esetleges egyéb függőséget is megpróbál rögvest lefordítani, anélkül, hogy zavarna bennünket a kérdéseivel. Az „sa” pedig azt jelenti, hogy minden függőséget „átlép”, azaz egyiket sem fordítja le.

Mi ugye „c”-t válaszolunk neki. Ezután lefordítja előbb a Guile nevű progit, aztán az Xbindkeys progit. No ezt a függőségi problémát könnyedén megúsztuk, mindössze lett eggyel több progink – gazdagabbak lettünk egy Guile-vel. Bár minden más függőség is ilyen könnyen megoldható lenne...

Miután e progit telepítetted, a következőt kell még tenned: írd be a `/Programs/BootScripts/Settings/BootScripts/BootUp` file végére e sorokat:

```
INITTY=/dev/tty[1-8]
for tty in $INITTY; do
setleds -D +num < $tty
done
```

De akár e sorokat is beveheted oda, ugyanazt csinálják:

```
for i in 1 2 3 4 5 6; do
setleds +num < /dev/tty${i}
done
```

Ha használsz a KDE-t is (és te jelenleg még használsz!), akkor a következőket is tedd meg:

A `/Programs/KDE-Base/Current/Shared/config/kdm/Xsetup` file-ba azt kell beírni, hogy

numlockx on

Ha a grafikus felületről nyitott virtuális termináloknál is rögvest bekapcsolva szeretnéd látni a numlockot, akkor a saját Home mappád (Tehát a `/Users/vz` mappád!) `.xinitrc` állományába tégy egy `numlockx on` & sort az elejére! Nálam az xbindkeys progi is indítva van itt (neked is ajánlom!), s emiatt így néz ki:

```
xbindkeys &
numlockx on &
[ -x /bin/startkde ] && exec startkde
[ -x /bin/wmaker ] && exec wmaker
[ -x /bin/twm ] && {
xsetroot -solid '#203040' -cursor_name 'left_ptr'
exec twm
}
```

Ha a tied nem ilyen, javítsd ki! De ehhez előbb lépj ki a rendszergazdai üzemmódból az

exit

paranccsal. Ugyanis azt az állományt kell szerkeszteni, ami az `/Users/vz/.xinitrc` útvonalon van.

És ennyi az egész! Ezután a következő indításkor már a kívánt módon viselkedik a masinád. Azaz, a NumLock led „on” (=bekapcsolt) fázisban lesz. Már a bejelentkezésnél is, ami nagyon kényelmes, ha a felhasználónevünk vagy a jelszavunk számokat is tartalmaz.

Azt hiszem most már mindenki láthatja, mi a helyzet ezzel a könyvvel: amint haladsz benne előre kedves Olvasó, egyre több mindent tudsz meg, egyre kevésbé leszel kezdő, s közben a rendszered is egyre barátságosabbá válik, a lépésről-lépésre haladó fejlesztéssel! És közben elmúlik a félelmed is a „konzoltól”, mert sokat dolgozol benne, megismered azt... egyre „Kockább” leszel! A következő fejezetben feltesszük az nVidia grafikus kártyák driverét. Ha a te konfigurációdnak nem nVidia a grafikus kártyája, akkor bocs, de nem tudok segíteni neked, mert az enyémben az van.

Az nVidia grafikus driver telepítése

Előfordulhat, hogy épp nVidia grafikus kártyád van. Ennek illik hogy telepítsd a driverét. Ez is a Compile paranccsal fog történni. No most ez amiatt nincs fenn a GoboLinux LiveCD-n, mert mindenféle jogi problémuszok miatt tilos együtt szállítani a telepítőlemezrel. Ennek ellenére ingyen letölthető az nVidia cég honlapjáról.

No most, eddigi telepítéseinket úgynevezett „virtuális terminálban” végeztük, azaz a **Konsole** progit felhasználva. Tehát valójában grafikus képernyőn dolgoztunk, épp csak volt egy ablakunk, ami olyan volt, mintha konzolban lettünk volna. No most az nVidia driver esetében ezt nem tehetjük meg. Ehhez ténylegesen ki kell lépnünk a grafikus felületből, hogy az azt „előidéző” Xorg szerver ne működjön. Tehát jelentkezzünk ki a KDE-ből, hogy ugyanaz a kép fogadjon bennünket, mintha le akarnánk állítani a rendszert. Menjünk itt is rendszergazdai módba (azaz a „**pv**” felhasználóként dolgozzunk, piros prompt-tal), és most, hogy úgynevezett „mélykonzolban” vagyunk, tehát nem működik a grafikus felület, adjuk ki a következő parancsot:

Compile nvidia

Erre ő letölti magának az Nvidia-169.07_bin-r1-recipe.tar.bz2 nevű állományt, ami jó nagy, 17 megás. Ezután a képernyőn fent lesz két kocka, hogy Accept és Do Not Accept. Ez arra vonatkozik, elfogadod-e a feltételeket. Nyomj egy **tab**-ot, hogy a kiválasztást jelző fehér csík az „Accept”-re vándoroljon, majd

=>Enter, azaz Entert kell nyomj.

Ezután kapsz egy Warning üzenetet kék mezőben, hogy mindenféle útvonalak megváltozását érzi a progi – ez nem számít, =>Enter.

Ezután még sok mindenféle szöveget ír ki, aminél Entert kell nyomj, e szövegeknek csak az első néhány szavát írom ki, hogy beazonosíthasd:

Kernel Source Path:... =>Enter

Building kernel module

Itt megjelenik egy haladásjelző csík, vársz amíg 100% lesz, majd Enter. Ennél a pontnál nálam sajnos totál elrontja a 100% elérése után a képernyőt, nem tudom ezután mit ír ki, de nyomj egy Entert. (Remélem, e gond csak nálam jelentkezik, mert ősrégi és picike a monitorom).

X library installation path... =>Enter

X module ... =>Enter

The Directory '/Programs/Nvidia/169.07_bin/lib/xorg/modules' does not exist; create? =>Enter

OpenGL installation prefix... =>Enter

Documentation... =>Enter

Utility... =>Enter

Install NVIDIA's OpenGL header files? =>Enter

The following operations... =>Enter
Would you like... =>Enter
=>Enter
do you want to install the unmanaged files?

Itt üss egy **y** betűt, majd =>Enter

Ezután még egy dolog hátra van. Indítsuk az mc progit, és a
/Programs/Xorg/Settings/X11/xorg.conf
állományt szerkesszük úgy, hogy megkeressük benne a
Section "Device"
részt, és ezen belül a
Driver "nv"
sort. Ezt írjuk át úgy, hogy
Driver "nvidia"

Ezután exit paranccsal lép ki a rendszergazdai üzemmódból, majd add ki a
startx
parancsot. Ha mindent jól csináltál, fel kell villanjon a KDE betöltődése előtt
az nVidia cég logója. Ha mégsem, az nagy pech. Nálam működött.

Ez azonban még nem elég. Ahhoz hogy menjen a Beryl s más efféle látványos
hóbelevancok, kell a 3D gyorsítás beállítása is. A következő fejezetben
megnézzük, ennek megléte/meg_nem_léte hogyan tesztelhető le, az
azutániban pedig (az Egy példa az xorg.conf -ra -ra című fejezetben) példát
adok rá, hogyan kell azt beállítani.

A 3D gyorsítás meglétének letesztelése

Biztos ami biztos, teszteljük le, van-e 3D gyorsítási lehetőség a gépünkön!
Ehhez normál userként is „elegek vagyunk”, üssük be mindössze a következő
parancsot:

```
glxinfo | grep "direct"
```

Ha erre az a válasz, hogy „direct rendering: Yes”, akkor megnyugodhatunk.
Ekkor nagy sóhaj, mert minden oké!

És ha nem ezt írja ki, hanem mindenféle más furcsaságot? Nos, az marha
nagy gáz. Esetleg egyvalamit megpróbálhatunk. Adjuk ki rendszergazdaként e
parancsot:

```
SymlinkProgram -f nvidia
```

Ez azt jelenti, hogy „besymlinkeljük” a programot, méghozzá „forced”, azaz
„erőltetett” módban. Vannak ugyanis olyan esetek, amikor a GoboLinux létre
akarja hozni a System könyvtárban az újonnan telepített progí szimbolikus

hivatkozásait, de egypár néven már ott szerepel valami más. Ilyen eset lehet, ha például egy programcsomag visz magával valami drivert ami általában szokott kelleni neki, de e driver már telepítve van nálad valami más csomag által. Ilyen esetben a GoboLinux alapértelmezett beállítása az, hogy meghagyja a régi program-alkotórészre való hivatkozást, nem cseréli le az újra mutatóra, nem minden alap nélkül gondolván rá, hogy az újról mit se tudunk, de a rendszerünk EDDIG tökéletesen működött a régivel is, azaz okosabb dolog azt meghagyni, remélhetőleg a most telepített új progi is működni fog vele!

E módszer majdnem mindig beválik. NÉHA azonban nem. Ilyenkor segíthet a „-f” kapcsoló megadása amint a fenti példánál láttuk. Ekkor kímélet nélkül átír minden hivatkozást az újra. (Ez persze bajt okozhat egyes régebben telepített programoknál elméletileg). Nálam például csak így sikerült az nVidiát rendesen telepíteni – nem hagyományos módon! Azaz felraktam a rendszert, majd lusta voltam az nvidia driverét telepíteni csomagból ahogy azt kéne, egyszerűen átmásoltam egy másik partícióra telepített GoboLinux rendszeremből a /Programs könyvtárba, és besymlinkeltem. Beállítottam az xorg.conf állományt is, újraindítottam a rendszert... látszólag ment is szépen, de a 3D gyorsítás nem. Abba egyedül akkor tudtam életet lehelni, amikor így „forced” módban symlinkeltem be. (Ezt persze megspórolhattam volna, ha szabályosan telepítem a drivert, de hát lustaság fél egészség, s így tanul az embör gyermeke...)

Egy példa az xorg.conf -ra

Hiába van nVidia driverünk, ettől még nem megy majd a Beryl s más ilyen „csicsa” nyalánságok. Ugyanis 3D gyorsítás csak az xorg.conf állomány megfelelő beállítása után várható.

Lássuk tehát az én efféle állományomat! (Emlékeztetőül: ennek teljes elérési útja a **/Programs/Xorg/Settings/X11/xorg.conf**).

```
vz@Csiszilla /Programs/Xorg/Settings/X11]cat xorg.conf
```

```
Section "ServerLayout"
Identifier "X.org Configured"
Screen 0 "Screen0" 0 0
InputDevice "Mouse0" "CorePointer"
InputDevice "Keyboard0" "CoreKeyboard"
EndSection

Section "Files"
RgbPath "/System/Links/Shared/X11/rgb"
ModulePath "/System/Links/Libraries/xorg/modules"
FontPath "/Files/Fonts/Misc/"
FontPath "/Files/Fonts/TrueType/"
FontPath "/Files/Fonts/TTF/"
```

```
FontPath "/Files/Fonts/OTF"
FontPath "/Files/Fonts/Type1/"
FontPath "/Files/Fonts/100dpi/"
FontPath "/Files/Fonts/75dpi/"
EndSection
```

```
Section "Module"
Load "GlcCore"
Load "dbe"
# Load "dri"
Load "extmod"
Load "glx"
Load "record"
Load "xtrap"
Load "freetype"
Load "type1"
EndSection
```

```
Section "InputDevice"
Identifier "Keyboard0"
Driver "kbd"
Option "XkbLayout" "hu"
Option "XkbModel" "pc104"
EndSection
```

```
Section "InputDevice"
Identifier "Mouse0"
Driver "mouse"
Option "Emulate3Buttons" "yes"
Option "ZAxisMapping" "4 5"
Option "Protocol" "IMPS/2"
Option "Device" "/dev/input/mice"
Option "ZAxisMapping" "4 5 6 7"
EndSection
```

```
Section "Monitor"
VertRefresh 43-75
HorizSync 28-50
Identifier "Monitor0"
VendorName "nVidia Corporation"
ModelName "Amaga"
Option "backingstore" "True"
Option "TripleBuffer" "True"
EndSection
```

```
Section "Device"
### Available Driver options are:-
### Values: <i>: integer, <f>: float, <bool>: "True"/"False",
### <string>: "String", <freq>: "<f> Hz/kHz/MHz"
### [arg]: arg optional
#Option "SWcursor" # [<bool>]
#Option "HWcursor" # [<bool>]
#Option "NoAccel" # [<bool>]
#Option "ShadowFB" # [<bool>]
#Option "UseFBDev" # [<bool>]
```

```

#Option "Rotate" # [<str>]
#Option "VideoKey" # <i>
#Option "FlatPanel" # [<bool>]
#Option "FPDither" # [<bool>]
#Option "CrtcNumber" # <i>
#Option "FPScale" # [<bool>]
#Option "FPTweak" # <i>
Identifier "GeForce FX 5200"
Driver "nvidia"
VendorName "nVidia Corporation"
BoardName "NV34 [GeForce FX 5200]"
BusID "PCI:1:0:0"
EndSection

Section "Screen"
Identifier "Screen0"
Device "GeForce FX 5200"
Monitor "Monitor0"
Option "AllowGLXWithComposite" "True"
Option "RenderAccel" "True"
Option "AddARGBGLXVisuals" "True"
DefaultDepth 24
SubSection "Display"
Depth 8
Modes "1024x768" "800x600" "640x480"
EndSubSection
SubSection "Display"
Depth 16
Modes "1024x768" "800x600" "640x480"
EndSubSection
SubSection "Display"
Depth 24
Modes "1024x768" "800x600" "640x480"
EndSubSection
EndSection
Section "Extensions"
Option "Composite" "Enable"
EndSection

```

Azt hiszem a legtöbb sor „magáért beszél” - ahol mégsem, például a *Section "Module"* részben, meg a *Section "Screen"* részben az általam félkövérrel kiemelt 3 sor esetében – ott van szó épp a 3D gyorsításról, de ez a számomra is misztikus homály, így nem magyarázom. Ezen infókat mindenféle fórumokon és ilyen-olyan leírásokból buheráltam össze. Mindenesetre így működik nálam. Ami fontos: a Beryl például csak „TrueColor” üzemmódban működik, azaz „24 bites színmélységben”. Na ezt határozzák meg a „Depth” kezdetű sorok, hogy milyen felbontás mellett hány bites színmélységre képes a grafikus kártyád/képernyőd. Látható, minden színmélységhez megadtuk az elérhető felbontásokat. Meg kell adni azt is, mi a „DefaultDepth”, azaz alapértelmezett színmélység, ez muszáj hogy 24 legyen, ha Berylt illetve Compiz-Fusion-t akarsz használni. (Tudod, a pörgő kocka meg hasonlók...)

A magyar nyelvű Firefox

Épp itt az ideje, hogy az install CD-vel együtt érkező angol nyelvű Firefoxot is lecseréljük egy magyar nyelvűre! Először is töltsük le a bináris csomagját a **<http://gobolinux.cwi.hu>** oldalról. Majd mindenekelőtt (rendszergazdaként...) tiltsuk le a régít:

DisableProgram Firefox

majd menjünk el a letöltési könyvtárba, és telepítsük:

```
pv@Csiszilla /Users/vz]InstallPackage Firefox--2.0.11--i686.tar.bz2
InstallPackage: Installing Firefox, version 2.0.11.
InstallPackage: Uncompressing to /Programs...
InstallPackage: Invalid signature. FileHash could not be verified.
InstallPackage: Suspect package in /Programs/Firefox/2.0.11
pv@Csiszilla /Users/vz]
```

Hoppá! Ez nem úgy néz ki, hogy sikerült volna... No de sebj, ez csak amiatt van, mert a csomag nincs digitálisan aláírva. (Lusta voltam...) Sebj, ettől még feltelepítjük – az InstallPackage progi már bemásolta amit kell a **/Programs** könyvtárba, csak a megfelelő szimbolikus kötéseket kell létrehozunk! Tehát:

```
pv@Csiszilla /Users/vz]SymlinkProgram Firefox 2.0.11
SymlinkProgram: Symlinking Firefox 2.0.11.
SymlinkProgram: Symlinking global settings...
SymlinkProgram: Symlinking tasks...
SymlinkProgram: Storing variable files...
SymlinkProgram: Symlinking libraries...
SymlinkProgram: Symlinking headers...
SymlinkProgram: Symlinking info...
SymlinkProgram: Updating info dir...
SymlinkProgram: Symlinking manuals...
SymlinkProgram: Symlinking executables...
SymlinkProgram: Processed 1 file.
SymlinkProgram: Symlinking wrappers...
SymlinkProgram: Symlinking shared...
SymlinkProgram: Removing unused directories...
SymlinkProgram: Done.
pv@Csiszilla /Users/vz]
```

Oké, ez megvan. Most azonban ha a KDE menüjébe megyünk, rájövünk, hogy nem találjuk ott a Firefoxot! Persze hogy nem: amikor letiltottuk a régít a DisableProgram paranccsal, törölte a menüből is. Sebj, létrehozunk itt egy újabb menüpontot neki! A menüszerkesztőben a „parancs” mezőbe ezt kell írunk:

'/Programs/Firefox/Current/bin/firefox'

tudniillik a progink amit indítani kell, ezen az útvonalon helyezkedik el.

No most azonban örömről enyhémszólva nem teljes, mert ha indítjuk Őfelségét, Firefox kisasszonyt, - akkor nem történik semmi. Ilyen esetben az a helyes megoldás, ha indítjuk parancssorból (terminálból) is, mert ilyenkor a linuxos progik ki szokták írni, mi a bánatuk. A Firefox nekünk ezt írja ki:

```
vz@Csiszilla ~]firefox
/Programs/Firefox/2.0.11/firefox-bin: error while loading shared libraries:
libstdc++.so.5: cannot open shared object file: No such file or directory
```

Ó, anyám! Ez biza egy függőségi probléma! Kell neki egy állomány a **libstdc++** progiból.

No most ezt megpróbálhatjuk telepíteni az InstallPackage segítségével:

```
pv@Csiszilla ~]InstallPackage LibStdC++
InstallPackage: Locating a binary package for LibStdC++ ...
InstallPackage: Installing http://kunder.org/gobo/packages/official/LibStdC+
+--4.1.2--i686.tar.bz2
InstallPackage: Press Enter to continue or Ctrl-C to cancel.

InstallPackage: Downloading package to /System/Variable/tmp/LibStdC+--4.1.2--
i686.tar.bz2.
--15:59:08-- http://kunder.org/gobo/packages/official/LibStdC+--4.1.2--
i686.tar.bz2
=> `/System/Variable/tmp/LibStdC+--4.1.2--i686.tar.bz2'
IP keresés kunder.org... 134.53.148.149
Connecting to kunder.org|134.53.148.149|:80... kapcsolódva.
HTTP kérés elküldve, várom a választ... 200 OK
Hossz: 716.942 (700K) [application/x-tar]

100%[=====>] 716.942          67.93K/s    ETA 00:00

15:59:22 (52.81 KB/s) - `/System/Variable/tmp/LibStdC+--4.1.2--i686.tar.bz2'
saved [716942/716942]

InstallPackage: Installing LibStdC++, version 4.1.2.
InstallPackage: Uncompressing to /Programs...
InstallPackage: Package Signature Verified
UpdateSettings: No settings exist
SymlinkProgram: Symlinking LibStdC++ 4.1.2.
SymlinkProgram: Symlinking global settings...
SymlinkProgram: Symlinking tasks...
SymlinkProgram: Storing variable files...
SymlinkProgram: Symlinking libraries...
SymlinkProgram: Conflict: /Programs/GCC/4.1.2/lib/libstdc++.la
SymlinkProgram: Conflict: /Programs/GCC/4.1.2/lib/libstdc++.so.6.0.8
SymlinkProgram: Conflict: /Programs/GCC/4.1.2/lib/libstdc++.a
SymlinkProgram: Conflict: /Programs/GCC/4.1.2/lib/libstdc++.so.6.0.8
SymlinkProgram: Conflict: /Programs/GCC/4.1.2/lib/libstdc++.so.6.0.8
SymlinkProgram: Processed 0 files.
SymlinkProgram: Updating library database (ldconfig)...
SymlinkProgram: Correcting directory references...
SymlinkProgram: Symlinking headers...
SymlinkProgram: Symlinking info...
```



```

SymlinkProgram: Updating info dir...
SymlinkProgram: Symlinking manuals...
SymlinkProgram: Symlinking executables...
SymlinkProgram: Symlinking wrappers...
SymlinkProgram: Symlinking shared...
SymlinkProgram: Removing unused directories...
SymlinkProgram: Done.
InstallPackage: Checking dependencies...
InstallPackage: Removing downloaded package /System/Variable/tmp/LibStdC+
+--4.1.2--i686.tar.bz2.
pv@Csiszilla ~]

```

A telepítés sikerült, sajnos amikor a Firefoxot akarjuk indítani, ugyanaz a hibaüzenet fogad. No igen: Annak a **libstdc++.so.5** állomány kell, de a fenti listán látható, hogy ez a most installált progik már a 6-os verziójú efféle állományokat telepítette!

Nincs más megoldás: ezt letiltjuk, mert nem jó nekünk:

DisableProgram libstdc++

és kell keresnünk egy régebbit e programból! Ezt meg is találjuk, ugyanazon az oldalon, ahonnan a Firefoxot letöltöttük: ez a **LibStdC++ 3.3.1** progik. Ezt letöltjük, és... De hoppá! Mivel találjuk meg, és mivel töltjük le, ha egyszer nem működik a Firefoxunk?!

Nos, erre legalább 2 módszer kínálkozik: egyrészt egy

SymLinkProgram Firefox 2.0.0.3

paranccsal átmenetileg életet lehelhetünk a régi, angol nyelvűbe – legfeljebb utána megint letiltjuk. Ennél sokkal bölcsebb azonban, ha a rendszerünkben levő másik böngészőt, a **Konqueror** nevűt használjuk erre az alkalomra.

Sőt, harmadik megoldás is van: egyszerűen letöltöd parancssorból, gondosan bemásolva ezt az utasítást:

```
wget http://birodalom.net/linux/gobocsomagok/LibStdC+--3.3.1--i686.tar.bz2
```

Szóval letöltjük ezt a progit is, és telepítjük. Ezt már nem írom le részletesen, mindenki tudja, hogy az InstallPackage paranccsal történik. Ha netán reklamálna hogy nincs digitális aláírás, akkor kell egy **SymLinkProgram** parancs is, aminek 2 paramétere van: a progik neve és verziószáma. Tehát:

```
SymLinkProgram libstdc++ 3.3.1
```

Na de honnét tudhatja a gyakorlatlan felhasználó a sok angol szövegből, sikerült-e rendesen a csomag telepítése?!

Egyszerű. El kell menni a **/Programs** könyvtárba, ott kell legyen egy altartalomjegyzék a progik nevével. Ebbe belemegyünk, itt kell legyen egy még „alabb” altartalomjegyzék a progik verziószámával, és egy **„Current”** nevű szimbolikus link, ami ezen al-altartalomjegyzékre mutat. Ha mindez így van, akkor valószínűleg jól telepítettünk mindent. Ha nincs „Current” link, akkor

nem sikerült a telepítés. Ha már a progí könyvtárát sem találtuk meg a **/Programs**-ban, akkor pláne nem sikerült a telepítés.

Betűméret a Firefoxban

No és akkor egy kis hekkerkedés. Előfordulhat, hogy valaki úgy találja, túl nagyok a betűk a Firefox (és a Thunderbird) progí menüiben. Sajnos a Firefoxban egyáltalán semmi menüpont nincs rá, hogy ezen változtassunk. A megoldás:

Menjünk el a saját Home könyvtárunkba (Tudod: **/Users/vz** a neve...), és itt kell lennie egy **.gtkrc-2.0** nevű állománynak. Ennek egyelőre valószínűleg valami ilyesmi a tartalma:

```
include ".kde/share/config/gtkrc-2.0"  
gtk-font-name="sans 14"
```

No ebben írjuk át a **gtk-font-name** kezdetű sort erre:
gtk-font-name="sherif 10"

A Firefox újraindítása után már helyes kis apró betűkkel jelennek majd meg a menüben a szövegek. Természetesen a „10” szám a betű méretét jelenti, azaz válasszuk a nekünk tetsző nagyságot. Nekem a tízes tetszik.

A Firefox okosítása

Vagyis, hogyan kell a Firefox alá mindenféle plugineket telepíteni? Például Flash playert és Djvu-nézegető plugint?

Djvu plugin telepítése

Mindenekelőtt: Az efféléket a Firefox a **/Programs/Firefox/verziószám/plugins** tartalomjegyzékben keresi. Oda kéne bemásolni a megfelelő programmodult. Na de hát ez elég durva, ugyanis elárulom, például a Djvu-fájlok megnyitását úgy érhetjük el, hogy telepítjük a Djvulibre nevű programcsomagot. (Nálam a 3.5.20_2 verziójú van fenn). No most miután ezt lefordítottuk a Compile paranccsal, ezután – híven a GoboLinux hagyományaihoz – ennek összes bizgentyűje-állománya a maga saját könyvtárába fog kerülni, nem a Firefoxéba! Na de semmi baj. Csináljunk egy szimbolikus linket a **/Programs/Firefox/verziószám/plugins** könyvtárba, mely a

/usr/local/lib/netscape/plugins/nsdejavu.so állományra mutat, s máris oké lesz minden! Ez természetesen a következő paranccsal érhető el (rendszergazdaként):

```
ln -s /usr/local/lib/netscape/plugins/nsdejavu.so  
/Programs/Firefox/verziószám/plugins/nsdejavu.so
```

(Ez nem két sor hanem egyetlen parancs, ami nem fért ki egy sorba írva...)

Éppenséggel írhattuk volna azt is, hogy

```
ln -s Programs/DjVuLibre/verziószám/lib/netscape/plugins/nsdejavu.so  
/Programs/Firefox/verziószám/plugins/nsdejavu.so
```

de jobb az előző változat, mert az akkor is működik, ha lecseréljük a Djvulibre progit egy modernebb változatra. Ha a második parancsot adnánk ki, akkor a létrejött szimbolikus link nem működne a Djvulibre frissítése után, azaz a linket is „frissítenünk kellene”. Az első változatnál azonban a létrejött szimbolikus link nem közvetlenül a megfelelő állományra mutat, hanem maga is egy szimbolikus linkre. E link a **/System/Links/Libraries** könyvtárban van. Itt van ugyanis egy „**netscape**” nevű link, ami a

/Programs/DjVuLibre/verziószám/lib/netscape könyvtárra mutat. S ez a link mindig aktualizálódik, amikor a Djvulibre programcsomagot frissítjük...

Valójában persze egy további láncolás is van a dologban: az **/usr/local/lib** is már egy szimbolikus link az előbb említett **/System/Links/Libraries** könyvtárra...

Flashplayer plugin telepítése

Flashplayer esetén is hasonlóan járhatunk el, ha van külön flashplayer-progink. Itt azonban többnyire készen szokta ezt letölteni magának az embör gyermeke... például a www.adobe.com oldaláról. Töltsük le mondjuk a .tar.gz kiterjesztésűt. (Gyakorlatilag mindegy melyiket, elvileg az rpm kiterjesztésű is lehetne, lényeg, hogy a Linuxhoz készültet, és ne mondjuk a Windows-osat...) No most ez a letöltött állomány tartalmaz egy installáló progit, na az nekünk nem kell. Hanem miután kicsomagoltuk valami ideiglenes helyre, találunk a könyvtárában egy **libflashplayer.so** nevű állományt. Na ezt kell bemásolnunk a **/Programs/Firefox/verziószám/plugins** könyvtárba, és készen is vagyunk.

Azaz dehogy vagyunk készen... bármi efféle izémizét telepítünk a Firefoxba, az csak akkor fog működni, ha előbb újraindítjuk... no nem a számítógépet, hanem a Firefoxot!

Mrxvt, avagy megszabadulunk a Konsole termináltól

Ha már olyan perverzek vagyunk, hogy mindent a „konzolban” intézünk el, akkor azt hiszem könnyű belátni, hogy esetünkben létfontosságú követelmény egy VALÓBAN profi terminálemulátor program! A Konsole nevű, ami a KDE egyik rendszerkomponense, sajnos nem ilyen. Talán már Te is észrevetted, kedves Olvasóm, hogy amikor az mc editorát használod (tudod, amit F4-el hívsz elő), akkor például nem használhatod a sorok kijelölésére a **Shift+kurzorbillentyűk** billentyűkombinációt! Nos, ez annyira idegesített engem, hogy soká kutattam, vajon melyik terminálemulátor progi képes erre? A Gnome-terminál „okéságos” volna, csak annak sajnos hihetetlenül sok függősége van! Végül ráleltem az **Mrxvt** terminálemulátorra, ami jobb még a Gnome-terminálnál is.

A dolog különben rávilágít arra, hogy Linux alatt akkora programbőség van, amiről egy Win-user nem is álmodik: terminálemulátorból is van rengetegféle, csak győzd őket megismerni, s kiválasztani a neked tetszőt!

Az Mrxvt rengeteg hasznos funkcióval rendelkezik. Ezeknek én bevallom, töredékét sem használom. Gyakorlatilag tényleg csak amiatt döntöttem mellette, hogy használhassam az említett billentyűkombinációkat. De akit érdekel, annak érdemes megismerkednie a (sajnos angol nyelvű) leírásával is, amit a

<http://materm.sourceforge.net/wiki/>
oldalon talál meg.

Az Mrxvt telepítése egyszerű:

Compile mrxvt

s ez telepíti nekünk a **0.5.3** verziójú mrxvt programot.

Ezután vegyük fel valahová a KDE menüjébe is a proginkat! A „parancs” mezőbe ezúttal a következő útvonal kerül:

'/Programs/Mrxvt/Current/bin/mrxvt'

Ezután már indítható is a drága – sajnos azonban iszonyatos a színbeállítása! Természetesen egy Linux rendszerben minden állítható, az egyes programok egyéni viselkedése is. Még hozzá mert a Linux, az egy többfelhasználós rendszer, így minden egyes felhasználónak lehetnek saját, egyéni beállításai. Ez többnyire úgy valósul meg, hogy a te „Home” mappádban – jelen esetben az **/Users/vz** könyvtárban – vannak bizonyos állományok vagy akár egész alkönyvtárak is, melyek a „.”, azaz a „pont” jellel kezdődnek! Ezek tartoznak az egyes programokhoz, s a te egyéni beállításaidat tartalmazzák. Ahhoz, hogy az mrxvt esetén is legyen neked ilyen, menj el a

/Programs/Mrxvt/Settings/mrxvt

könyvtárba, s itt találsz egy **mrxvtrc.sample** nevű állományt. Ezt másold a saját könyvtáradba, (Az **/Users/vz** nevűbe!), de úgy, hogy a neve legyen **.mrxvtrc** – FONTOS a „pont” a név elején! - ezután ezen állomány sorainak átírásával

egyéniiséget adhatsz e terminálemulátornak. Ezen állományba nézzünk be nyugodtan az editorunkkal. Bőséges megjegyzés is van benne ahhoz hogy tudjuk, mi mire szolgál – igaz, ezek angol nyelvű megjegyzések. De nem nehéz rájönni, mi mire való. Ajánlom mindenki figyelmébe főleg a „**Colors**” részt, ahol a fontosabb színeket határozhatjuk meg!

Végre van egy terminálemulátorunk, ahol kijelölésre használhatjuk kedvenc Shift+kurzorbillentyűinket... ezek után nem is használjuk többé a Konszole progit. Szerepét betöltötte, kész, vége.

Egy gyors megjegyzés a használatával kapcsolatban: Itt is kijelölhetünk szöveget az egér segítségével, mint a Konszole alatt. De itt nem kell a jobb egérgombbal klikkelni s a helyi menüből kiválasztani a „másolás” parancsot ahhoz, hogy a kijelölt szöveg a vágólapra kerüljön – amint kijelöltük azt, az máris a „vágólapra” kerül! A beillesztés viszont – legyünk akár a Firefox, akár az OpenOffice alatt – a **KÖZÉPSŐ EGÉRGOMB** megnyomására következik be! Soká tartott, míg erre rájöttem...

Továbbá: a GoboLinux előző változatában, a 013 alatt az Mrxvt-ben nem működtek a Delete, Home és End gombok, csak egy tilde-karaktert azaz hullámvonalat (~) rajzoltak ki. Bár e hibát a 014-es GoboLinux kiadásban már orvosolták a fejlesztők, talán nem felesleges ha leírom ide a megoldást, mert lehet belőle tanulni! Tehát a megoldás: Töltsük le a

<http://linux.birodalom.net/gobocsomagok/zkbd-rxvt-pc-linux-gnu>

állományt és másoljuk be a /Programs/Scripts/Current/Shared/zsh/ könyvtárba! Ezután szerkesszük a /Programs/Scripts/Settings/zshrc fájlt úgy, hogy kitöröljük belőle a ["\$DISPLAY" != ""] && export TERM='xterm-color' részt (vagy megjegyzéssé tesszük, eléje egy "#" jelet írva). Ezután

SymlinkProgram Scripts

majd bezárjuk és újra megnyitjuk az mrxvt progit, és örülünk, mert elvileg minden jól kell működjék.

A letöltendő és másolandó fájl tartalma különben a következő:

```
typeset -g -A key
key[F1]='^[11~'
key[F2]='^[12~'
key[F3]='^[13~'
key[F4]='^[14~'
key[F5]='^[15~'
key[F6]='^[17~'
key[F7]='^[18~'
key[F8]='^[19~'
key[F9]='^[20~'
key[F10]='^[21~'
key[F11]='^[23~'
key[F12]='^[24~'
key[Backspace]='^?'
```

```
key[Insert]='^[[2~'  
key[Home]='^[[7~'  
key[PageUp]='^[[5~'  
key[Delete]='^[[3~'  
key[End]='^[[8~'  
key[PageDown]='^[[6~'  
key[Up]='^[[A'  
key[Left]='^[[D'  
key[Down]='^[[B'  
key[Right]='^[[C'
```

A következő fejezetben megkönnyítjük további munkánkat, néhány hasznos billentyűparanccsal és rövidítéssel.

Forrógombok, aliasok, és egy konzolos zenelejátszó progi

No hát kezdjük is e fejezetet a zenelejátszó progi telepítésével! Ez a **Moc** nevű program **2.4.3-r1** változata lesz. Telepítése:

Compile moc

Mindjárt az elején azt mondja nekünk, hogy

```
CheckDependencies: FAAC is not installed.  
CheckDependencies: Compile recipe for FAAC 1.26 or skip this dependency?  
[C]Compile/[S]Skip/[CA]Compile All/[SA]Skip All
```

Azaz kell neki valami „FAAC” nevű progi. Oké, nyomjunk neki **c**-t, telepítse csak amire szüksége van! Igen, de kiderül, hogy kell neki egy FAAD2 nevű progi is. Megint nyomunk egy **c**-t. Erre kiderül, hogy kell neki FFmpeg progi is. Megunván hogy annyit nyomkodjuk a **c**-t, ezúttal nyomjuk azt, hogy **ca** – ez azt jelenti, hogy fordítson csak le magától mindent ami kell, és ne rikoztassék nekünk minden egyes függőségénél!

A fordítási folyamat jó sokáig el fog tartani – ezalatt nyugodtan igyunk meg egy kávét, teát, vagy böngésszük az Internetet.

Na, végre lefordított mindent nekünk a kis drága. Most éppenséggel indíthatnánk ökelmét, de ne tegyük. Ugyanis hiába indítjuk el, annak amit lejátszik, nem lesz hangja! Eddig a hangkérdésekkel nem foglalkoztunk, de most már épp itt az ideje. Sajnos telepítés után minden hangesatorna alapértelmezés szerint 0 hangerőre van állítva. Ez elég hülye dolog, de így van.

Ezen rögtön változtatunk.

A hangkártya bekonfigurálása

Mindenekelőtt, be kell konfigurálnunk a hangkártyánkat. Ennek érdekében szerkesszük a

`/System/Settings/BootScripts/BootUp` file-ot! Abban kell legyen egy kikommentezett

```
Exec "Loading sound settings..."
```

```
Alsactl Start
```

sor. Töröljük a sor elejéről a keresztet („#”). Mentsük el a változtatást! Hasonlóképp szerkesszük a `/System/Settings/BootScripts/Shutdown` file-ot is, abban ugyanis kell legyen egy szintén kikommentezett

```
Exec "Storing sound settings"
```

```
Alsactl Stop
```

sor, ennek elejéről is távolítsuk el a kommentjelet (a keresztet), s mentsük el az állományt!

E változtatásokkal azt tettük, hogy megparancsoltuk a rendszernek, hogy

leálláskor mentse el a hangkártya aktuális beállításait, és rendszerinduláskor töltsse vissza a korábban elmentett beállításokat.

Most keressük meg a

`/Programs/BootScripts/Settings/BootOptions` file-ot, és keressük meg benne ezt a részt (a legvégén van):

```
UserDefinedModules=(  
    snd-pcm-oss  
    snd-mixer-oss  
)
```

Ezt írjuk át így, azaz egészítsük ki:

```
UserDefinedModules=(  
    snd-pcm-oss  
    snd-mixer-oss  
    snd-intel8x0  
)
```

Ezzel tulajdonképpen azt mondtuk a gépnek, hogy induláskor töltsse be a hangkártyát kezelő modult. Ez természetesen csak akkor jó, ha ennek megfelelő hangkártyával rendelkezünk; szerintem a legtöbb esetben megfelel ez, „átlagos” gépek esetén; ha neked más hangkártyád van, nem tudok segíteni benne, hogyan keltsd életre – annyi bizonyos, hogy az **snd-intel8x0** helyére kell írnod valami modulnevet, de hogy mit, azt neked kell kiderítened.

Ezután indítsuk újra a számítógépet. Sajnos muszáj. (Illetve, végeredményben nem lenne muszáj, de most nem akarjuk olyasmikkel sokkolni magunkat, hogy kernelmodulok kézi betöltése, meg hasonlók).

Most már ideje végre megparancsolni a rendszernek, hogy vegyen tudomást a hangkártyánkról. Miután nekünk (nekem...) **ALSA** típusú „hangkarcsim” van, tegyük azt, hogy (rendszergazdaként) kiadjuk az

alsaconf

parancsot! Erre elindul egy kis progí – elég mindenütt Entert nyomni a kérdéseire. Ha jól számoltam 5 esetben kell Entert nyomni. Amikor véget ért a futása, adjuk ki az

alsactl store

parancsot is, hogy elmentse ezen beállításokat, amiket épp most derített fel.

Az XBindkeys bekonfigurálása

Most következnek a tulajdonképpeni zenelejátszó programot vezérlő mindenféle hasznos okosságok. Beállítunk mindjárt billentyűzetparancsokat, hogy „gombnyomorgatással” a billentyűzetről tudjuk szabályozni a hangerőt. Ezt

természetesen a már korábban feltelepített **xbindkeys** progival tesszük meg.

Ha most elindítjuk azt, a parancssorba írva az xbindkeys szót, ezt a választ kapjuk:

```
vz@Csiszilla ~]xbindkeys
Error : /Users/vz/.xbindkeysrc not found or reading not allowed.
please, create one with 'xbindkeys --defaults > /Users/vz/.xbindkeysrc'.
or, if you want scheme configuration style,
with 'xbindkeys --defaults-guile > /Users/vz/.xbindkeysrc.scm'.
vz@Csiszilla ~]xbindkeys --defaults > /Users/vz/.xbindkeysrc
vz@Csiszilla ~]
```

Ez azt jelenti, hogy nem találta meg az xbindkeys a maga config állományát a mi Home könyvtárunkban, amely állománynak a neve **.xbindkeysrc** kéne legyen, és javasolja nekünk, hogy hozzunk létre egy ilyen nevűt, alapértelmezés szerinti beállításokkal, kiadván az

```
xbindkeys --defaults > /Users/vz/.xbindkeysrc
```

parancsot. Hát legyen meg az öröme, s adjuk ki e parancsot! Ezután ezt rögtön szerkesszük is.

Mielőtt továbblépni, megjegyzem, hogy feltételezem, az Olvasónak olyan billentyűzete van, melyen szerepelnek különböző úgynevezett „multimédiás billentyűk” is. Ezeknek többnyire olyan nevük szokott lenni, hogy „Home”, „Back”, „Forward”, „My favorites”, stb. Ezekhez – és néhány más gombhoz – különböző parancsokat fogunk most mindjárt hozzárendelni. Akinek nincs ilyen billentyűzete, ne habozzon beszerezni egy ilyet, nem egy nagy összeg manapság. Mindazonáltal az itt leírt parancsokat értelemszerűen más billentyűkombinációkhoz is hozzárendelheti, megfelelően módosítva a leírásomon.

Tehát, lépünk be az mc-be, és vegyük szemügyre az F4 segítségével a **.xbindkeysrc** állományt! A dolog fontosságára való tekintettel közlöm az egésznek a listáját, bár tele van angol szöveggel. A #-el kezdődő sorok természetesen mind megjegyzés.

```
# For the benefit of emacs users: -*- shell-script -*-
#####
# xbindkeys configuration #
#####
#
# Version: 1.8.2
#
# If you edit this file, do not forget to uncomment any lines
# that you change.
# The pound(#) symbol may be used anywhere for comments.
#
# To specify a key, you can use 'xbindkeys --key' or
# 'xbindkeys --multikey' and put one of the two lines in this file.
#
```

```

# The format of a command line is:
#   "command to start"
#       associated key
#
#
# A list of keys is in /usr/include/X11/keysym.h and in
# /usr/include/X11/keysymdef.h
# The XK_ is not needed.
#
# List of modifier:
#   Release, Control, Shift, Mod1 (Alt), Mod2 (NumLock),
#   Mod3 (CapsLock), Mod4, Mod5 (Scroll).
#

# The release modifier is not a standard X modifier, but you can
# use it if you want to catch release events instead of press events

# By defaults, xbindkeys does not pay attention with the modifiers
# NumLock, CapsLock and ScrollLock.
# Uncomment the lines above if you want to pay attention to them.

#keystate_numlock = enable
#keystate_capslock = enable
#keystate_scrolllock= enable

# Examples of commands:

"xbindkeys_show"
    control+shift + q

# set directly keycode (here control + f with my keyboard)
"xterm"
    c:41 + m:0x4

# specify a mouse button
"xterm"
    control + b:2

#"xterm -geom 50x20+20+20"
#   Shift+Mod2+alt + s
#
## set directly keycode (here control+alt+mod2 + f with my keyboard)
#"xterm"
#   alt + c:0x29 + m:4 + mod2
#
## Control+Shift+a release event starts rxvt
#"rxvt"
#   release+control+shift + a
#
## Control + mouse button 2 release event starts rxvt
#"rxvt"
#   Control + b:2 + Release

#####
# End of xbindkeys configuration #

```

#####

Aki ért egy picit angolul, ebből már ki tudja következtetni, miként is „kezelődik” ez. Itt bizonyos gombkombinációkhoz lehet parancsokat hozzárendelni. Például ez a rész:

```
# set directly keycode (here control + f with my keyboard)
"xterm"
  c:41 + m:0x4
```

azt mondja, hogy az „xterm” progit (ami egy primitív terminálemulátor) a **Control+f** billentyűkkel lehessen indítani.

No most ezt a fájlt kell szerkeszteniünk a mi igényeinknek megfelelően. Mindenekelőtt e két sort vegyük bele az elejére:

```
keystate_numlock = disable
keystate_capslock = disable
```

Ez azt mondja, hogy ne vegye figyelembe a parancsoknál a NumLock és a CapsLock billentyűk állapotát. Ugye azt akarjuk, hogy a parancsaink akkor is működjenek, ha ezek akárhogyan is állnak épp. Aztán „kommentezzük ki” az xterm-re vonatkozó sorokat – egyáltalán, mindent, azt kivéve ami az xbindkeys_show-ra utal. Az hasznos lesz, mert megmutatja majd nekünk, épp milyen billentyűparancsaink „élnek”.

Ezután pedig fel kell vennünk a megfelelő parancsokat! Ehhez tudnunk kell a megfelelő billentyűkombinációk kódjait. Ezek úgy tudhatóak meg, hogy kiadjuk az

```
xbindkeys -k
```

parancsot, majd lenyomunk egy billentyűkombinációt. Erre kiírja nekünk annak a kódját. Például nyomjuk csak meg ilyen esetben a multimédiás „My favorites” gombot! Ezt fogja kiírni:

```
vz@Csiszilla ~]xbindkeys -k
Press combination of keys or/and click under the window.
You can use one of the two lines after "NoCommand"
in $HOME/.xbindkeysrc to bind a key.
"(Scheme function)"
  m:0x10 + c:230
  Mod2 + NoSymbol
```

Ha viszont kikapcsoljuk a NumLock gombot, s úgy kérdezzük meg tőle a „My favorites” gomb kódját, már ezt válaszolja:

```
vz@Csiszilla ~]xbindkeys -k
```

Press combination of keys or/and click under the window.
 You can use one of the two lines after "NoCommand"
 in \$HOME/.xbindkeysrc to bind a key.
 "(Scheme function)"
 m:0x0 + c:230
 NoSymbol

Nos, nekünk ebből a m:0x0 + c:230 rész az érdekes, ez a „kód”. S amiatt ezt választjuk, mert mint írtuk, úgysem akarjuk figyelembe venni a NumLock állapotát a parancshozzárendelésnél, márpedig ebből az első rész, az „m:0x0” a módosítóbillentyűk állapotára utal, s a billentyű tulajdonképpen saját kódja itt a „c:230” rész.

Ezek után az összes billentyű kódját meghatározhatjuk, de mert ezek már úgyis kvázi-szabványosak, ide is írom az összeset:

Multimédiás billentyűzet speciális billentyűinek kódjai

Billentyű	Kód
Home	178 <i>(Ez nem azonos az „End” gomb fölötti „normális”, régen megszokott „home” gombunkkal!)</i>
Back	234
Forward	233
My favorites	230
E-mail	236
Play/Pause	162
Previous	144
Next	153
Sleep	223
Fn+PowerOff	222
Fn+WakeUp	227

És ezek után szépen elkezdhetjük a különböző funkciók felvételét a **.xbindkeysrc** fájlunkba! Azt hiszem helyes lesz, ha közlöm a teljes, módosított fájl tartalmát.

```
# For the benefit of emacs users: -*- shell-script -*-
#####
```

```

# xbindkeys configuration #
#####
#
# Version: 1.8.2
#
# If you edit this file, do not forget to uncomment any lines
# that you change.
# The pound(#) symbol may be used anywhere for comments.
#
# To specify a key, you can use 'xbindkeys --key' or
# 'xbindkeys --multikey' and put one of the two lines in this file.
#
# The format of a command line is:
#     "command to start"
#         associated key
#
#
# A list of keys is in /usr/include/X11/keysym.h and in
# /usr/include/X11/keysymdef.h
# The XK_ is not needed.
#
# List of modifier:
#   Release, Control, Shift, Mod1 (Alt), Mod2 (NumLock),
#   Mod3 (CapsLock), Mod4, Mod5 (Scroll).
#

# The release modifier is not a standard X modifier, but you can
# use it if you want to catch release events instead of press events

# By defaults, xbindkeys does not pay attention with the modifiers
# NumLock, CapsLock and ScrollLock.
# Uncomment the lines above if you want to pay attention to them.

keystate_numlock = disable
keystate_capslock = disable
#keystate_scrolllock= enable

# Examples of commands:

"xbindkeys_show"
    control+shift + q

#Shift+PageUp

"amixer set PCM 5+"
    m:0x11 + c:99

#Shift+PageDown

"amixer set PCM 5-"
    m:0x11 + c:105

#Ctrl+PageUp

"amixer set Master 5+"

```

```

m:0x14 + c:99

#Ctrl+PageDown

"amixer set Master 5-"
m:0x14 + c:105

# MyFavorites

"mrxvt"
m:0x0 + c:230

# Play/Pause

"mocp -G"
m:0x0 + c:162

# Previous

"mocp -r"
m:0x0 + c:144

# Next

"mocp -f"
m:0x0 + c:153

"gkrellm"
control+shift+alt + g

#Shift+PrintScreen

"import -window root `date +%y%m%d%H%M%S`.png "
c:111 + m:0x11

# set directly keycode (here control + f with my keyboard)
#"xterm"
# c:41 + m:0x4

# specify a mouse button
#"xterm"
# control + b:2

#"xterm -geom 50x20+20+20"
# Shift+Mod2+alt + s
#
## set directly keycode (here control+alt+mod2 + f with my keyboard)
#"xterm"
# alt + c:0x29 + m:4 + mod2
#
## Control+Shift+a release event starts rxvt
#"rxvt"
# release+control+shift + a
#
## Control + mouse button 2 release event starts rxvt

```

```

# "rxvt"
# Control + b:2 + Release

#####
# End of xbindkeys configuration #
#####

```

E sok új funkció működéséhez az kell, hogy most lépjunk ki a terminálból, majd lépünk be oda újra...

Az aktuális billentyűhozzárendeléseket az

xbindkeys_show

paranccsal irathatjuk ki. Sajnos ha most ezt ki akarjuk próbálni, ezt a választ kapjuk:

```
vz@Csiszilla ~]$xbindkeys_show
```

```
/System/Links/Executables/xbindkeys_show: line 3: exec: wish: not found
```

A megoldás kézenfekvő. Telepíteni kell a **wish** programot. Sajnos, nem „wish” néven találjuk meg – tudni kell, hogy ez a „Tk” progí része. Tehát:

Compile tk

Ez nekünk lefordítja majd, amit kell. Megjegyzendő, hogy e proginak függősége a **Tcl** progí, rá is kérdez majd, akarjuk-e telepíteni – persze hogy akarjuk! Ez tehát telepíti nekünk a **Tcl 8.5.1** és a **Tk 8.5.1** verzióit. Ezután már működik az **xbindkeys_show** parancs, ami egy szép grafikus ablakba kiírja nekünk az aktuális hozzárendeléseket. Sőt, most már elég ehhez a **Shift+Ctrl+Q** billentyűkombináció is!

No de mi igazi „Kockák” vagyunk, parancssor-mániások! Mi van, ha „mély-konzolban” vagyunk, s ott akarjuk kiírni ezeket, holott ott nincs grafikus ablak?!

Semmi vész. Akkor használjuk az

xbindkeys --show

parancsot!

No de most már épp itt az ideje annak, hogy elmeséljük, miféle parancsokat kötöttünk miféle billentyűkhöz a fenti, példaként közölt **.xbindkeysrc** állományban! Íme!

Shift+Ctrl+Q = **xbindkeys_show**

Shift+PageUp = PCM hangerő növelése 5 egységgel

Shift+PageDown = PCM hangerő csökkentése 5 egységgel

Ctrl+PageUp = Master hangerő növelése 5 egységgel

Ctrl+PageDown = Master hangerő csökkentése 5 egységgel

My favorites = Mrxvt terminál indítása

Play/Pause = Moc zenelejátszón játszott zene megállítása/indítása

Previous = Moc zenelejátszó listájában az előző zeneszámra ugrás

Next = Moc zenelejátszó listájában a következő zeneszámra ugrás

Shift+Ctrl+Alt+G = GKrellM rendszermonitor indítása

Shift+PrintScreen = képernyőkép készítése png formátumban

Ja és hogy miért van két-két funkció a hangosításra és a halkításra? Mert két csatorna van erre, az egyiknek a neve PCM, a másiké Master. Ennek jelentőségét – bevallom büntudatosan – nem egészen értem, de úgy különösebben nem is érdekel. (Ettől még megírhatja nekem, aki tudja, de csak jól szájbarágósan, hogy én, a csekélyértelmű medvebocs is felfogjam!) A lényeg, hogy *van* lehetőségünk szabályozni mindegyiket, és kész. Használjuk egészséggel!

Szép lista, hasznos, jó funkciók! Megoldódott az is, hogy legyen képernyőkép-pünk. Hogy miért nem a sima PrintScreenre kötöttem? Mert így nem szemeltethetjük tele a rendszerünket képernyőképekkel, ha véletlenül nyomjuk meg azt a gombot. Azaz, a Shifttel együtt biztonságosabb. Megjegyzem, hogy tudjuk hol keressük: a saját HOME könyvtárunkba menti el a képernyőképeket, azaz a /Users/vz -be, és a kép állományneve az aktuális dátum és idő számadataiból képződik. A kiterjesztése meg természetesen **.png** lesz. Megjegyzendő, hogy e funkció különösképp hasznos. Tele van a világ képernyőkép-készítő progikkal. Mi a nyavalyának?! Amit lehet, oldjunk meg egyszerűen! (Erős a gyanúm különben, hogy az összes képernyőkép-készítő progi is *végső soron* ezt a rutint „rángatja”, hívogatja, csak szép „ruhát”, „csomagolást” ad hozzá).

E listának csak az a szépséghibája, hogy a **GKrellM** rendszermonitor még fel sincs telepítve... Nosza, tegyünk róla, hogy legyen!

Compile gkrellm

Ez a **GKrellM 2.3.1-r1** verzióját telepíti nekünk. Nagyon szuper rendszermonitor! Bővíthető is, lehet hozzá letölteni mindenféle skineket, plugineket, stb... A kezelését most nem magyarázom el, nem e könyv célja. Lényeg, hogy a különböző részein, azaz pici képernyőin az egér jobb gombjával kattintva felugrik azon rész beállítóablaka, ahol beparaméterezhetjük, például hogy milyen magas legyen az a kisablak. A fejlécen kattintva így egy általános konfiguráló ablak jelenik meg, ahol monitorunk szélességét is beállíthatjuk (pixelben).

Aliasok, azaz parancsok rövidítései

És akkor most egyelőre elég a gombnyomorgatásból – következze a rövidítések világa! Ez némileg egyszerűbb téma lesz.

Arról van szó, hogy ha most a **moc** paranccsal akarjuk indítani a konzolos zenelejátszó proginkat – az nem fog működni. Nincs ugyanis olyan progink,

hogy „moc”. Az, amit a mocból indíthatunk, ugyanis a „**mocp**” névre hallgat... Elég hülye dolog, de így van. Jobb lenne, ha a sima „moc” parancs is működne... Sőt! Ahogy a könyvtárában böngészünk, meg ahogy olvasgatjuk a kezelési utasítását (már aki olvasgatja... sajnos ezzel még én is úgy vagyok, mint a legtöbb ember: idő hiányában az az elvem, hogy „ha végleg nem megy, olvasd el a kezelési utasítást...”), kiderül, hogy különböző „témák” is vannak hozzá, a kinézetére! Ezeket a

/Programs/Moc/2.5.0-alpha2/Shared/moc/themes könyvtárban leljük fel. Nosza készítsünk rögvest egy másolatot a **yellow_red_theme** nevűről mondjuk **viola_theme** névvel – ezentúl ezt fogjuk használni, s ha kell, átírni is! A másolat is legyen ugyanebben a könyvtárban! No most egy meghatározott témával a zenelejátszót úgy kell indítani, hogy – például -

```
mocp -T viola_theme
```

No most ezt nagyon nehézkes minden alkalommal beírni! Milyen jó is volna, ha egyszerűen egy

```
moc
```

parancsra máris így cselekedne!

Van megoldás. Amikor elindítasz egy terminálemulátort, az beolvassa a maga config fájlját a te HOME könyvtáradból. Ezt kell módosítanod.

Egészen pontosan nem a terminálemulátor config fájljára kell gondolnunk persze, bár van annak is. Ugye kedvenc Mrxvt-nknek például a **.mrxvtrc** file. Hanem a terminálemulátor elindít egy úgynevezett parancsfeldolgozót, ami nagyjából megfelel a hírhedt DOS-nak. Parancsfeldolgozóból több fajta létezik a Linux világban, neked a GoboLinux rendszeredbe kettő is telepítve van, egyiknek a neve **Bash**, a másiké **Zsh**. A disztribúciókban többnyire a Bash az alapértelmezett parancsfeldolgozó, szakszóval „**shell**” - a GoboLinux alatt azonban a **Zsh**. (a parancsok szintaxisa tudomásom szerint lényegében ugyanaz e kettőnél). Ezeknek a config fájljai a **.bashrc** illetve **.zshrc** nevet viselik. Neked tehát ezeket, de legalábbis a **.zshrc** fájljt kell módosítanod, hogy saját kedved szerint működjék. Ez most a következőképp nézhet ki nálad, ha kilistázod a cat paranccsal:

```
vz@Csiszilla ~]cat .zshrc
alias pico="nano -p"
alias rm="rm -i"
alias cp="cp -i"
alias d="ls --color"
```

```
export PATH=$PATH:.
```

```
if [ "$SSH_CONNECTION" ]
then prompt lode yellow
else prompt lode cyan
fi
```

```
# function to start one instance of gpg-agent for current user
#if [ -z "$(echo $(ps -C gpg-agent -o user=) | grep $(whoami))" ]
```

```
#then
#  gpg-agent --daemon -s --write-env-file ~/.gnupg/gpg-agent.env 1> /dev/null
#fi
#export $(cat ~/.gnupg/gpg-agent.env)
```

A #-el jelölt sorok természetesen itt is megjegyzések.

A fájl elején sok olyan sort látunk, amiknek a kezdete „alias”. E szó nem az „alias” hibásan írt változata, hanem azt jelenti, hogy nagy aljasul bizonyos parancsoknak álneveket adunk...

Nosza, egészítsük ki a fájlt néhány sorral, hogy így nézzen ki:

```
vz@Csiszilla ~]cat .zshrc
alias pico="nano -p"
alias rm="rm -i"
alias cp="cp -i"
alias d="ls --color"
alias ú="ls -l"
alias á="ls -l"
alias pen="mount /dev/sda1"
alias upen="umount /dev/sda1"
alias oh="cd /Mount/HDA7/Zene/OfraHaza"
alias ol="cd ~/ofralink"
alias z="cd /Mount/HDA7/Zene"
alias le="cd /Depot/Letolt"
alias m="cd /Mount"
alias xmms=audacious
alias x=exit
alias p="cd /Programs"
alias wiki="cd /Programs/Xampp/1.6.5.a/htdocs/dokuwiki"
alias o="cd /Programs/Os/0.0.0/bin"
alias moc="mocp -T viola_theme"
export PATH=$PATH:

if [ "$SSH_CONNECTION" ]
then prompt lode yellow
else prompt lode cyan
fi

# function to start one instance of gpg-agent for current user
#if [ -z "$(echo $(ps -C gpg-agent -o user=) | grep $(whoami))" ]
#then
#  gpg-agent --daemon -s --write-env-file ~/.gnupg/gpg-agent.env 1> /dev/null
#fi
#export $(cat ~/.gnupg/gpg-agent.env)
```

Látható a lényeg: különböző hasznos rövidítéseket határoztunk meg az „alias” parancs segítségével! Például, ezentúl a sima

moc

parancs egyenértékű azzal, mintha a mocp progit a viola_theme-vel indítottuk volna! A „z” parancs hatására elugrik abba a könyvtárba, ahol a zenéim vannak, az „oh” parancs hatására oda, ahol kedvenc énekesnőm, „Ofra Haza”

zenéi vannak összegyűjtve, stb. A „pen” és „upen” szolgál arra, hogy könnyen fel- illetve lecsatolhassam a pendrive-omat.

E ponton jegyezném meg, hogy „mp3 lejátszókat”¹ ugyanúgy kell csatolni, mint a pendrájvokat.

Azt hiszem ez annyira világos, hogy nem is kell többet írnom róla. A fentiek természetesen az énnekem fontos rövidítések, a Kedves Olvasó bátran cserélje ki a neki nem kellőeket valami hasznosra, vagy teljesen hagyja ki őket, s kitalálhat újabbakat is.

A moc progi kezeléséről (sajnos csak angolul) ezen az oldalon tájékozódhatsz:

<http://moc.daper.net/>

E könyvnek nem célja azt részleteiben bemutatni. A kinézete mindenestre kedvenc **mc** proginkéhez hasonlatos. És ha elindult, a **h** betű lenyomására szépen kilistázza neked az összes billentyűparancsát (melyek persze csak akkor „élnek”, amikor épp a **moc** ablakát látod a képernyőn... a listát lefelé a kurzorbillentyűvel görditheted, a kilépés belőle pedig az **Esc**-vel történik). A moc-ból ki a **q** betűvel léphetünk, de a zene tovább fog szólni! Csak akkor hallgat el, ha kiadjuk a

mcop -x

parancsot, ez ugyanis leállítja a démont.

Felhívom azonban a figyelmet egy fontos kényelmi szempontra. Ahogy a **help** képernyőjén is láthatjuk az „f” gomb lenyomása azt csinálja, hogy „**Toggle ReadTags option**”. Azaz magyarul, ezzel ki/be kapcsolhatjuk azt a funkcióját, hogy a lejátszandó számokból kiolvasson mindenféle ott leírt tulajdonságokat, azaz nem szigorúan zenei infókat, mint pl az ott letárolt zeneszám-név, stb. Amennyiben ez be van nála kapcsolva (és alapesetben be van...), akkor a **moc** ablakában nem a zene állománya neve jelenik meg, hanem ez a „tags”-ban letárolt zenénév, kivéve persze ha ott semmi nincs letárolva, az esetben ugyanis mégis az állománya nevet mutatja. Ez többnyire nagyon jól is van így, csak az a baj, hogy némely esetben van „tags” letárolva a zeneszámokban, de bármiféle érdemleges infó nélkül, mert zenénév helyett olyasmik vannak ott, hogy „**Track 1**”, „**Track 2**” stb, ami nekünk semmi hasznosat nem mond, legfeljebb hogy hányadik zeneszám volt a CD-n, amiről digitalizálták, de ezzel nem sokra megyünk, mert kell a halálnak, ki kíváncsi erre?! Ezesetben mégis jobb lenne talán az állománya nevet mutatni, remélve hogy annak elnevezése információgazdagabb. Nos ekkor nyomjuk meg az „f” gombot, hogy azt mutassa...

¹ Egy kis nyelvújítási kezdeményezés: Javasolom hogy a különböző zenelejátszó küttyüket nevezzük úgy, hogy „**zenemorzsa**”, s ne úgy, hogy „mp3 lejátszó”. Pláne mert nemcsak mp3 formátumot játszanak le de wma-t is, sőt a jobbak a kedvencemet, az *ogg vorbis* fileformátumot is. A zenemorzsa-szó jó, mert nem 2 szó mint az, hogy „mp3 lejátszó”, nincs is szám a névben, illeszkedik a magyar nyelvbe, rövidebb is mint az „mp3 lejátszó”, sőt, nincs benne ékezetes karakter sem, ami bizonyos esetekben jól jöhet a számítástechnika világában.

Saját forróbillentyűk definiálása a moc programnál

Természetesen arra is van lehetőség, hogy a moc-ot vezérlő forróbillentyűket mi magunk határozzuk meg. Ehhez csináljunk egy másolatot a

`/Programs/Moc/Current/Shared/doc/moc/keymap.example`

fájlról valamilyen néven a \$HOME könyvtárunk **.moc** alkönyvtárába! Én a file nevének stílszerűen (és roppantmód fantáziadúsan...) azt választottam, hogy „keymap”, azaz most van egy `/Users/vz/.moc/keymap` nevű file-om. Ebben akármit megváltoztathatok. Ahhoz hogy a moc tudja, hogy ezt kell használnia, csinálnunk kell ugyanebbe a `/Users/vz/.moc` könyvtárba egy **config** nevű file-ot is, aminek ezt a sort kell tartalmaznia:

Keymap = keymap

vagyis látható, egyszerűen megadjuk a billentyűhozzárendeléseket tartalmazó file nevét.

S most már például megkereshetem a keymap file-ban a

```
quit_client =      q  
sort, s átírhatom erre:
```

```
quit_client =      :
```

S ezentúl nem a **q**, hanem a **:** billentyűre lép ki. Ha ellenben valamely funkcióbillentyű lenyomására akarjuk kiléptetni, arra itt a példa, hogyan kell azt definiálnunk:

```
quit_client =      F10
```

Én magam is ezt használom, mert ez megegyezik az `mc` progi beállításával, s így nem kell két külön kilépési billentyűt megtanulnom.

Vegyük fel ezeket a sorokat is a `/Users/vz/.moc/config` fájlba:

```
Fastdir1 = "/Users/vz/ofralink"
```

```
Fastdir2 = "/Mount/HDA7/Zene"
```

```
ExecCommand8 = "rm %f"
```

Ez azt jelenti, hogy a moc proginak meghatározhatunk előre betáplált „gyorskönyvtárakat”, ahová elugrik a megadott billentyűparancsra, illetve meghatározhatunk végrehajtandó parancsokat is, melyeket természetesen szintén bizonyos gyorsbillentyűre, azaz billentyűlenyomásra hajt végre. És mi most meghatároztunk neki két efféle „gyorstartalomjegyzéket”, a `Fastdir1`-et és a `Fastdir2`-t, természetesen a Kedves Olvasó cserélje ki az idézőjelek közti szövegrészt a neki tetsző tartalomjegyzék-linkre... valamint azt mondtuk neki, hogy a 8-as sorszámú parancs az legyen, hogy letörli azt a zeneszámot, amin épp állunk a kiválasztást jelző csíkkal a `moc` zenelistájában. Ez nagyon hasznos, ha kapunk egy zenekollekciót amiből ki akarjuk selejtezni a nekünk nem tetszőeket. Minek legyen emiatt nyitva egy külön terminálablak az `mc`-

vel!

Na de mik az e tartalomjegyzékekhez és a parancshoz dukáló billentyű-kombinációk?

Természetesen ezeket is a **keymap** nevű fájlban találjuk meg. Ott a vége felé például ez a sor szerepel:

```
exec_command8 =      F8
```

Ez jelenti azt, hogy az előbb beállított zeneszámtörlésre utasított 8-as parancsot az **F8** billentyű lenyomásával aktiválhatjuk. Beállíthattam volna akármi más billentyűre is, de ez nekem nagyon megfelelt, mert az általam sokat használt mc progiban is az F8 a fájltrörlő gomb. A „gyorstartalom-jegyzékek” eléréséhez más speciális jelek vannak meghatározva, azok, melyek a „go_to_fast_dir” kezdetű sorokban szerepelnek, ezek alapértelmezés szerint a következők:

```
go_to_fast_dir1 =      !
go_to_fast_dir2 =      @
go_to_fast_dir3 =      #
go_to_fast_dir4 =      $
go_to_fast_dir5 =      %
go_to_fast_dir6 =      ^
go_to_fast_dir7 =      &
go_to_fast_dir8 =      *
go_to_fast_dir9 =      (
go_to_fast_dir10 =     )
```

E beállítások természetesen csak a **mc** progi újraindítása után lépnek életbe.

Böngészés mélykonzolból és virtuális terminálból

Nem kell megijedni, eszünk ágában sincs örökre lemondani szeretett Firefoxunkról! De megtörténhet, hogy ellenállhatatlan kíváncsiságunktól és kísérletezési vágyunktól vezettetvén, valami olyasmit cselekszünk, amivel úgy elrontjuk a rendszerünket, hogy többé nem indul a grafikus felület, s nem tudunk életet lehelni belé. Ekkor 2 dolgot tehetünk:

1. Indítjuk a LiveCD-t, s erről bejelentkezvén valami linuxos fórumra, tanácsot kérünk a nálunk okosabbaktól. Vagy épp e-mailban kérünk tanácsot.
2. Ugyanezt tesszük, azaz tanácsot kérünk, épp csak ezért nem indítunk LiveCD-t, hanem egy parancssori böngészőt futtatunk.

No most az 1. eset világos, azzal nem is foglalkozunk. A második eset érdekesebbnek tűnik, de elkedvetlenítően egyszerű lesz... ugyanis mindössze arról van szó, hogy Linux alá van számos „parancssoros” böngésző. Ezek egyike telepítve is van nekünk, a neve: Links. Ezt természetesen a **links**

parancssal indíthatjuk, akár a „mélykonzolból” (tudod, ahonnet a startx parancssal léphetsz be a grafikus felületbe), akár valamely virtuális terminálból.

Egy efféle böngésző képességei természetesen szerények – ne várjuk el tőle, hogy JAVA-t mutogasson nekünk, de még azt sem, hogy egyáltalán jpg vagy png képeket. De a weboldalakon az infók többsége úgyis a szöveg, az a lényeg, s azt jól megmutatja. A szöveges linkeket is. Be tudunk jelentkezni bárhová a jelszavainkkal, kommunikálni tudunk, stb. A kezelését nem írom le, hamar rájössz – lényeg, hogy a KurzorLe/Fel billentyűkkel mozoghatsz a szövegben, s az mindig ráugrik a következő linkre, akkor =>Enter, ha arra a linkre akarsz ugrani...

A Links indítható – virtuális terminálból! - úgy is, hogy

links -g

Ez az ő úgynevezett „grafikus üzemmódja”. Ez annyiból másabb, hogy itt már megjelennek menük is benne – de csak ha **Esc** gombot nyomsz. A menü szövegei természetesen angolul vannak. Vagy az egérrel kell kattints az ablak fejlécén. (Igen, itt már lehet használni az egeret is, ebben az üzemmódban!) Ahhoz, hogy URL-t írj be, a „**g**” gombot kell megnyomnod. Ezek a legfontosabb tudnivalók. És még az, hogy itt már megjelennek bizonyos esetekben a grafikák, azaz a képek is a weboldalokról. De némely oldalak mégis furán mutatnak, mert mintha nem kezelné a kereteket (frames). De használható azért, ha minden kötél szakad.

Egy másik ilyen böngésző is létezik, ami így hirtelenjében eszembe jut, ez a **Lynx**, és ez a **Compile lynx** parancssal telepíthető.

Ezek nem sok helyet foglalnak, érdemes megtartani őket, minden eshetőségre készen. Elvégre sosem lehetünk eléggé pesszimisták és paranoiások. Különben is, igaz Kocka nehogy már ne rendelkezék a parancssori böngészés lehetőségeivel!

CD-írás parancssorból

Amint meglátod e címet, szerintem kiráz téged a hideg. „Nagy egek, ez meg minő sátánista perverzió?! Hát ezeknek a linuxos fickóknak nincs valami Néróhoz hasonlatos normális CD-író szoftverük, hogy erre az iszonyatra kényszerülnek?!”

Persze hogy van olyan. Az a neve, hogy **K3B**, és megtalálod őt a KDE menüjében. És még magyar nyelvű is lesz ökelme!

Mégsem tartom feleslegesnek ismertetnem ennek módját, ugyanis VÉGTELENÜL EGYSZERŰ, legalábbis ha az ember semmi különleges „extrát” nem akar. A lényeg, hogy tudjuk, hogy CD-t mindig **.iso** képmásból írunk! Amit természetesen el kell készítenünk, mert csak utána írhatjuk ki. Ehhez mindenekelőtt hozd létre azt a könyvtárat, aminek a tartalmát fel akarod írni, benne pontosan olyan mappákkal, amiket a CD-n látni óhajtasz, persze a mappák legyenek teli a megfelelő tartalommal is, azaz a fájlokkal... majd...

E paranccsal készíthetsz egy ISO képmásfájlt:

```
mkisofs -J -V "proba" /Users/vz/Dokumentumok > proba.iso
```

A fenti parancs elkészíti az /Users/vz/Dokumentumok könyvtárból (tehát az itt tárolt fájlokból) és ennek összes altartalomjegyzékéből a proba.iso nevű fájlt (CD-képmást), melyen pontosan a megadott /Users/vz/Dokumentumok könyvtár tartalomjegyzékstruktúrája lesz. A kötet címke az lesz, hogy "proba", és Joliet-bejegyzések is lesznek, azaz ennek megfelelő formátumú lesz a képmás. (Vagyis e CD-t amit írsz, „szeretett” Windowsod alól is olvashatod majd). A képmásfájl abba a könyvtárba lesz elhelyezve, amiben épp állsz a parancs kiadásakor.

Ellenőrzéshez mountoljuk fel:

```
mount -o loop proba.iso /Mount/PEN
```

Ez a /Mount/PEN könyvtárba mountolja fel a fájlt. (Már ha létezik ez a könyvtárad... És ezt többnyire csak root-ként tehetjük meg). Ezután ha bemész mondjuk az mc progival a /Mount/PEN könyvtárba, láthatod a proba.iso képmásfájl tartalmát, épp úgy, mintha ez egy külön partíció volna, vagy egy már kész CD.

Az umountolás:

```
umount /Mount/PEN
```

A képmás lemezre égetése:

```
cdrecord dev=/dev/hdc proba.iso
```

Na ez sokáig fog dolgozni... És amíg el nem készül, a promptot sem kapod vissza. Érdekességképpen ideírom, milyen kimenetet látsz akkor, ha a GoboLinux 014 LiveCD iso-képmását írod ki egy lemezre ezzel a módszerrel:


```

pv@Csiszilla /Mount/HDA9/iso]cdrecord dev=/dev/hdc GoboLinux-014-i686.iso
cdrecord: No write mode specified.
cdrecord: Assuming -sao mode.
cdrecord: If your drive does not accept -sao, try -tao.
cdrecord: Future versions of cdrecord may have different drive dependent
defaults.
Cdrecord-ProDVD-Clone 2.01.01a23 (i686-pc-linux-gnu) Copyright (C) 1995-2006
Jörg Schilling
cdrecord: Warning: Running on Linux-2.6.23.8-Gobo
cdrecord: There are unsettled issues with Linux-2.5 and newer.
cdrecord: If you have unexpected problems, please try Linux-2.4 or Solaris.
scsibdev: '/dev/hdc'
devname: '/dev/hdc'
scsibus: -2 target: -2 lun: -2
Warning: Open by 'devname' is unintentional and not supported.
Linux sg driver version: 3.5.27
Using libscg version 'schily-0.9'.
Device type      : Removable CD-ROM
Version         : 0
Response Format: 2
Capabilities     :
Vendor_info      : 'HL-DT-ST'
Identifikation  : 'CD-RW GCE-8480B '
Revision        : '1.03'
Device seems to be: Generic mmc CD-RW.
Using generic SCSI-3/mmc CD-R/CD-RW driver (mmc_cdr).
Driver flags    : MMC-2 SWABAUDIO BURNFREE
Supported modes: TAO PACKET SAO SAO/R96P SAO/R96R RAW/R16 RAW/R96P RAW/R96R
Starting to write CD/DVD at speed 48 in real SAO mode for single session.
Last chance to quit, starting real write    0 seconds. Operation starts.
cdrecord: WARNING: Drive returns wrong startsec (0) using -150
Track 01: Total bytes read/written: 699897856/699897856 (341747 sectors).
pv@Csiszilla /Mount/HDA9/iso]

```

No jó, oké – mondhatja erre valaki – de hogy a nyavalyába teszünk szert a GoboLinux CD képmására?!

Na de kérem, hiszen ez PONT AZ a képmásfájl, amit eredetileg még Windows alatt letöltöttél, s onnan lemezre írtál! Ha valamiért letörölted volna, s most mégis kéne, mert a havernak is lemezre akarod írni, semmi gond, ha megvan maga a CD még – ugyanis BÁRMELY lemezről könnyedén készíthetsz iso képmásfájlt ezzel a paranccsal:

```
dd if=/dev/hdc of=A_képmásfájl_kívánt_neve
```

(Persze, előbb tedd be a CD-t a meghajtóba...)

Ha valaki még használna floppyt, annak kedvéért itt van, hogyan készítsen floppyról is képmásfájlt:

```
dd if=/dev/fd0 of=A_képmásfájl_kívánt_neve
```

A képmásfájl alapján új floppyt írni:

```
dd if=A_képmásfájl_neve of=/dev/fd0
```

(Persze, ehhez tegyük be a floppyt, és az sem árt, ha abban a könyvtárban

állunk épp, ahol a képmásfájl van...)

Újraírható CD törlése parancssorból:

cdrecord -blank=fast dev=/dev/hdc

Illetve ha ez valamiért nem válna be, akkor:

cdrecord -blank=all dev=/dev/hdc

Ez utóbbi persze tovább tart...

Most már remélem világos, hogyan másolhatsz le egy CD-t parancssorból: megcsinálod a képmásfájlt róla, majd kiírod azt. A szép az egészben az, hogy mindegy, a CD bootolható-e vagy sem. (zenei cédék esetén nem tudom, működik-e ez a módszer. Ha valaki kipróbálja, írja meg nekem az eredményt).

Ennyit tehát a parancssori CD-írásról. Persze rengeteg paramétere lehetséges az **mkisofs** és a **cdrecord** proginak is, de nem célom ezek mélységeibe belemerülni – az esetek többségére a fenti ismeretek épp elegendek. Nem ismertetem azt sem, hogyan lehet parancssorból zenei CD-ket másolni/írni. Az már bonyolultabb eset ugyanis.

DVD írása parancssorból

A DVD-íráshoz (a parancssorihoz) a **growisofs** progi kell. Használata:

growisofs -Z /dev/dvd -R -J /kiirandó/alkönyvtár

A -R a „RockRidge”, a -J az úgynevezett „Joliet” bejegyzések létrehozását jelenti. Persze e parancsot rendszergazdai jogkörrel („root” userként, azaz mint „pv”) illik kiadni...

DVD képmás lemezre égetése:

growisofs -dvd-compatible -Z /dev/dvd=image.iso

Fájltypusok programokhoz társítása az mc progiban

Miután eddig mi lényegében mindent parancssorból (na jó, virtuális terminálból...) intéztünk el, s ott is főleg az mc program segítségével, nyilvánvalóan alapvető követelmény, hogy meg tudjuk határozni, ha valami állományon állunk a kiválasztást jelző inverz csikkel, és Entert nyomunk, akkor milyen progit indítson az mc, mely azt az állományt kezeli valamiképp. E tudást e fejezet közli veled, tisztelt Olvasóm!

Nyilván nem lepődöl meg azon, hogy azt, miféle állományhoz milyen progit kell használnia, az mc egy config file-szerű szöveges állományban tárolja. Ezen állományt csak rendszergazdaként szerkesztheted, és a **/Programs/MC/4.6.1/Shared/mc** könyvtárban leled meg, aholis a neve **mc.ext** – beszédes név, mert azt jelenti, az mc progi extensionokkal, azaz kiterjesztésekkel foglalkozó állománya! És elég hosszú.

Nem kell megijedni, nem fogom idemásolni az egészet. Csak a bennünket érdeklő részeket. Ez nem lesz olyan hosszú, ellenben annál hasznosabb! Kezdjük is!

Mindenekelőtt keressünk benne egy

```
### Archives ###
```

részt. Ez után vannak összegyűjtve az úgynevezett archivált, azaz tömörített, „becsomagolt” fájlokra vonatkozó tudnivalók az mc számára. A többségét felesleges piszkálnunk, de találunk egy ilyen részt:

```
# deb
regex /\.u?deb$
Open=%cd %p#deb
View=%view{ascii} dpkg-deb -c %f
```

Nos, ez természetesen a Debian nevű Linux disztribúció csomagjaira vonatkozik, melyeknek a kiterjesztése természetesen – ki hinné? - **.deb** szokott lenni... Ne menjünk bele annak rejtelseibe, miért ilyen bonyolult módon van megadva a kiterjesztés; ami most bennünket érdekel az az, hogy van egy Magyarországon készített Linux disztribúció is, aminek UhuLinux a neve (elérhető a <http://uhulinux.hu> oldalon), és ennek a csomagjai bár **.uhu** kiterjesztésűek, de felépítésükben megegyeznek a debian csomagokéval. Szeretnénk, ha a rendszerünk ezeket is kezelni tudná. Nem mintha ez olyan égetően fontos lenne, de miért is ne! Nos, az MC nemcsak azt tudja mint a legtöbb más fájlkezelő progi – nemcsak a megnyitáshoz tud alkalmazást társítani, de ahhoz is, hogy **belenézzünk** az állományba! A megnyitási infókat természetesen az **Open=** - kezdetű sorban, a nézegetésit meg a **View=** - kezdetű sorban kell megadnunk neki. No most, ha a kiválasztott állományon Entert nyomunk, vagy kétszer kattintunk az állományon az egérrel, azt tekinti „Open”-nek, azaz megnyitásnak (ha úgy tetszik hát futtatásnak), és az F3 gomb megnyomását „View”-nek.

Látjuk, hogy a debian csomagokat a **dpkg-deb** nevű progival óhajtja megnyitni. Ilyen progink momentán nincs, de a

<http://linux.birodalom.net/gobowiki/index.php/Csomagok>

oldalról le lehet tölteni azt, amit én hekkerkedtem össze. Ez természetesen bináris csomag. Töltsük le és telepítsük az

InstallPackage DPKG-1.13.25--i686.tar.bz2

paranccsal. Kiabálni fog, hogy nincs aláírva a csomag, de üssünk bátran Entert, hogy telepítse csak.

Ezután meg kell mondjuk az mc proginak, hogy az .uhu csomagokat is úgy kezelje, mint a .deb csomagokat. Ehhez a fent bemutatott részt cseréljük ki a következőre:

```
# deb
#regex/\.u?deb$
# Open=%cd %p#deb
# View=%view{ascii} dpkg-deb -c %f

# deb
regex/\.(deb|uhu)$
Open=%cd %p#deb
View=%view{ascii} dpkg-deb -I %f control ; echo ; dpkg-deb -c %f
```

Látható, nagy óvatosan meghagytuk a régi beállításokat, csak megjegyzéssé „fokoztuk le” annak sorait. Ajánlom e módszert mindenki figyelmébe, ha ezt az állományt piszkálja! Így könnyen vissza lehet csinálni mindent, ha elrontunk valamit.

Következik a

Sound files

rész. Itt nagyot fogunk kaszálni. Itt kötjük rá szeretett Moc zenelejátszónkat a megfelelő hangállományokra. Konkrétan az Mp3-ra és Ogg-ra. De az itt bemutatott példa alapján a többi is megoldható.

Rögvest látunk egy ilyen részt:

```
regex/\.([wW][aA][vV]|[sS][nN][dD]|[vV][oO][cC]|[aA][uU]|[sS][mM][pP]|[aA][iI][fF][fF]|[sS][nN][dD])$
Open=if [ "$DISPLAY" = "" ]; then play %f; else (xmms -e %f 1>/dev/null
2>&1 &); fi
```

Ezt cseréljük ki erre:

```
regex/\.([sS][nN][dD]|[vV][oO][cC]|[aA][uU]|[sS][mM][pP]|[aA][iI][fF][fF]|[sS][nN][dD])$
Open=if [ "$DISPLAY" = "" ]; then play %f; else (audacious -e %f 1>/dev/
null 2>&1 &); fi
```

E cserének az az oka, hogy a rendszerünkben nincs xmms program, van ellenben audacious, ami különben teljesen ugyanúgy néz ki mint az xmms – sőt, mint akár a Winamp a Windows alatt! (Ráadásul simán meg lehet adni neki a Winamp „skin”-jeit is, tudja kezelni őket...)

Van aztán ez a rész:

```
regex/\.([wW][aA][wW]22)$
    Open=vplay -s 22 %f

regex/\.([mM][pP]3)$
    Open=if [ "$DISPLAY" = "" ]; then mpg123 %f; else (xmms %f &); fi
    View=%view{ascii} mpg123 -vtn1 %f 2>&1 | sed -n
    '/^Title/,/^Comment/p;/^MPEG/,/^Audio/p'

regex/\.([oO][gG][gG])$
    Open=if [ "$DISPLAY" = "" ]; then ogg123 %f; else (xmms %f &); fi
    View=%view{ascii} ogginfo %s

regex/\.([mM][iI][dD][iI]?|[rR][mM][iI][dD]?)$
    Open=timidity %f

regex/\.([wW][mM][aA])$
    Open=mpplayer -vo null %f
```

Ezt kéretik kicserélni erre:

```
regex/\.([wW][aA][vV])$
    Open=audacious %f

#regex/\.([mM][pP]3)$
# Open=if [ "$DISPLAY" = "" ]; then mpg123 %f; else (xmms %f &); fi
# View=%view{ascii} mpg123 -vtn1 %f 2>&1 | sed -n
# '/^Title/,/^Comment/p;/^MPEG/,/^Audio/p'

regex/\.([mM][pP]3)$
    Open=if [ "$DISPLAY" = "" ]; then mpg123 %f; else (mocp -T viola_theme %f);
fi
    View=%view{ascii} audacious -vtn1 %f 2>&1 | sed -n
    '/^Title/,/^Comment/p;/^MPEG/,/^Audio/p'

#regex/\.([oO][gG][gG])$
# Open=if [ "$DISPLAY" = "" ]; then ogg123 %f; else (xmms %f &); fi
# View=%view{ascii} ogginfo %s

regex/\.([oO][gG][gG])$
    Open=if [ "$DISPLAY" = "" ]; then ogg123 %f; else (mocp -T viola_theme %f);
fi
    View=%view{ascii} ogginfo %s

regex/\.([mM][iI][dD][iI]?|[rR][mM][iI][dD]?)$
    Open=timidity %f

regex/\.([wW][mM][aA])$
    Open=audacious %f

# Open=mpplayer -vo null %f
#
```

Ez azt eredményezi, hogy ezentúl a WMA állományokat az audacious progival akarja majd lejátszani, például. Ellenben az mp3 és ogg számokat a moc progival. Ilyenkor indítja a moc-ot, annak jobb paneljába bekerül a lejátszani kívánt szám, ekkor elég ott egy Enter, s a zene máris szól! Valamint, a midi számokat a Timidity++ nevű progival kísérli meg lejátszani. Persze, ehhez fel kell azt telepítenünk. Ezt a

Compile timidity++

paranccsal érjük el. Rögvest meg fogja kérdezni tőlünk, akarjuk-e a FreePats progit telepíteni, persze hogy akarjuk – ez jó 25 megás cumó, de muszáj, mert ebben vannak a hangszerdefiníciós állományok!

A

Play lists

rész alatti dolgokat erre cseréljük ki:

```
regex/\.([pP][lL][sS])$
  Open=if [ -z "$DISPLAY" ]; then mplayer -vo null -playlist %f; else
(audacious -p %f &); fi

regex/\.([mM]3[uU])$
  Open=(mocc -T viola_theme %f)
```

Amint látható, a moc progit mindig rögvest úgy „huzaloztuk be” ide, hogy a saját megjelenési témánkkal együtt induljon, amit átírhatunk nekünk megfelelőre, ha akarjuk. Igaz, ezt egyelőre csak rendszergazdai joggal szerkeszthetjük... de sebj! Menjünk el a

/Programs/Moc/2.4.3/Shared/moc/themes

könyvtárba, s itt adjuk ki a következő parancsot (rendszergazdaként):

chown vz viola_theme

Ez átadta a viola_theme file tulajdonjogát a vz nevű felhasználónak. Ezután már nem rendszergazdai jogkörben is megváltoztathatjuk azt.

Még egyéb mindenféle dolgokat is állíthatunk az mc.ext file-ben, de ez már a kedves Olvasó fantáziájára és igényeire van bízva. Az eddigi példák áttanulmányozása alapján menni fog, legfeljebb kell hozzá némi kísérletezés. (kísérletezés előtt mentjük el az eredeti mc.ext file-ot valami sokatmondó néven valahová, s akkor nem kerülünk bajba, ha nagyon elrontjuk. Ugyanis bármit rontunk is el benne, annak kizárólag az mc működésére lesz kihatása, másra nem. Ha visszamásoljuk a régit az elrontott helyére, kilépünk mc-ből is és a terminálból is, akkor ha új terminálablakot nyitunk, már minden a régi módon fog működni).

Egy hasznos infó befejezésül: a fenti listákban gyakran látunk egy efféle feltételtesztelést:

if ["\$DISPLAY" = ""]; then ... ; else ...

Nos, itt azt teszteli le, van-e grafikus képernyő, vagy „mélykonzolban” va-

gyunk. Ettől függően ugyanis más-más progikat rendelhetünk hozzá az adott funkcióhoz. Ha NINCS grafikus képernyőnk, azaz a „mélykonzolban” vagyunk, akkor a **then** utáni részben meghatározott progival játssza le a számot (illetve nézhetjük meg a belsejét), ha VAN grafikus képernyőnk, akkor az **else** utáni rész aktivizálódik. A **%f** jelölés meg természetesen a szám (általában a kiválasztott állomány) nevére utal, mint paraméterre.

Nyomtató telepítése

Természetesen a nyomtató telepítésének folyamatát is egy konkrét példán át mutatom be. Nekem van egy USB-csatlakozós **Samsung ML-1520** típusú lézernyomtatóm, ezt fogjuk telepíteni most. Előtte természetesen csatlakoztassuk a nyomtatót a géphez, és kapcsoljuk is be.

Aztán (rendszergazdaként):

```
pv@Csiszilla ~]$StartTask cups
```

```
StartTask: CUPS: Returned code 0 => OK.
```

```
pv@Csiszilla ~]
```

A cups egy úgynevezett „démon”, azaz háttérben futó progí. Ő a „nyomtató-kezelő démon”. Így nevezik. Minthogy a nyomtatás nem mindennapos feladat még azoknál sem akiknek van nyomtatójuk – márpedig az nincs is mindenkinek – emiatt ez nem indul el automatikusan a rendszerindulásnál, nekünk kell elindítani. Ha már nincs szükség rá, leállítható (rendszergazdaként) a

StopTask cups
paranccsal.

Ahhoz képest hogy eddig mindent a terminálban intéztünk el, most kellemes meglepetés ér bennünket: a nyomtató telepítését és konfigurálását egyszerűen a BÖNGÉSZŐNKBŐL végezhetjük el. A Firefoxból, de akár más böngészőből is. Most ugyanis, hogy már el van indítva a cups, üssük be a böngészőnkbe, hogy

<http://localhost:631/>

Ez egy internetcím, mely azonban a mi számítógépünkre mutat, annak a 631-es portjára, ami a cups démon kommunikációs portja mifelénk. Ugyanez elérhető a

<http://csiszilla:631/>

link beírásával is, tudniillik mert a „localhost” mindig a mi saját gépünk neve, de telepítéskor megmondtuk azt is a rendszerünknek, hogy e gépnek az is a neve, hogy „Csiszilla”.

No és most egy rendes grafikus ablak ugrik fel, tulajdonképpen egy weboldal, bár nem magyar nyelvű. De azért könnyű rajta eligazodni. Kattintsunk az „**Administration**” fülre. Rögvest szemünkbe ötlük a kiírás:

New Printers Found: Samsung ML-1520 (Samsung ML-1520 USB #1)

és ott van mellette a gomb is, hogy „**Add this printer**”. Remek, ez kell nekünk! Kattintunk rá. Milyen okos gépecske, rögvest felismerte a nyomtatónkat! Tiszta Windows-érzés!

Némi várakozás után megjelenik egy ablak, benne sok Samsung nyomtató

nevével, melyek közt inverz csikban lesz a

Samsung ML-1520 Foomatic/gdi (en)

sor. Alatta egy üres sor, azzal a névvel, hogy **Or Provide a PPD File:**

Ott lehetne egy **gdi** kiterjesztésű csomagot adni a démonnak, ami a nyomtató úgymond „meghajtója”. Erre most nem lesz szükségünk – elfogadjuk amit felajánl. Kattintsunk az **„Add Printer”** gombra. Erre felugrik egy ablak, melyben kéri a rendszergazdai nevet és jelszót. Adjuk meg neki. Most pár másodperc múlva kiírja angolul, hogy sikeresen telepítette a nyomtatót, s megjelenít egy ablakot, ahol a nyomtató mindenféle adatait lehet beállítani – nekünk egyelőre jó az alapértelmezett. Kattintsunk a **„Printers”** feliratú gombra, majd a **„Set as defaults”**-ra. Ezzel alapértelmezett nyomtatóvá tesszük.

Megjegyzés: Talán jó tudni, hogy a nyomtató „PPD” fájlját ezek után a `/Programs/CUPS/Settings/cups/ppd` könyvtárban leljük fel. Ez különben egy szöveges fájl, amiben mindenfélét átírkálhatunk – bár ez nagyon nem ajánlott a hozzá nem értők számára! Ugyanezt a könyvtárat elérhetjük a `/System/Settings/cups/ppd` szimbolikus linket követve is. E könyvtárban nekem most egy `Samsung_ML-1520_USB_1.ppd` nevű fájlom van a telepítés után.

Elvileg, ha a létező világok legjobbjában élnénk, kész is volnánk minden-
nel, amivel kell – sajnos azonban ha nyomtatni akarunk, a következő hiba-
üzenet fogad majd:

**Samsung_ML-1520_USB_1 (Default Printer) "Printer not connected; will
retry in 30 seconds..."**

Ez ténylegesen egy hiba. De nem mi rontottunk el valamit. Arról van szó, hogy a nyomtatónk be van dugva, de mégis úgy érzékeli a rendszer, mintha nem lenne bedugva. Ennek oka, hogy nem tudja írni azt az eszközfájlt, amit pedig kéne hogy írjon. Ez afféle jogosultsági probléma, s valószínűleg a rendszerünkben levő „udev” nevű progival függ össze. Nos, ezt megoldhatjuk úgy, hogy (rendszergazdaként) kiadjuk a

chmod o+rw /dev/usb/lp0

parancsot. Ezzel írhatóvá tettük a megfelelő portot, s a nyomtatás lehetségessé válik. Minthogy a cups démon ahogy írta, 30 másodpercenként próbál csatlakozni a nyomtatóhoz, e parancs kiadásától számítva 30 másodpercen belül el kell induljon a nyomtatás. (Viszont az az angol üzenet hogy a nyomtató nincs csatlakoztatva, csak akkor tűnik el a képernyődről, ha frissítéd az oldalt a böngésződben).

Mielőtt felsóhajtanánk hogy minden oké, sietve közlöm, hogy NEM, nem oké minden. Ha újraindítod a számítógépedet, nemcsak a **StartTask cups** parancsot kell kiadnod újra – ami még hagyján lenne – de sajnos a fenti ronda `chmod`-kezdetű parancsot is! Ugyanis valami általam ismeretlen okból e beállítást elfelejti a rendszer. Erre még nem találtam megoldást (aki tudja,

írja meg!), de jobb híján hogy kényelmesebbé tegyem a rendszert, javasolom, csináljunk egy **alias**-t erre a feladatra, amit nevezzünk mondjuk „printer”-nek. Ezt persze úgyis csak rendszergazdaként adhatjuk ki, tehát célszerű a **pv** felhasználó **.zshrc** fájljába beírakni.

A rendszer bővítése saját szkriptekkel

Másik, talán nehezebbnek tűnő, de hosszú távon előbbremutató megoldás, ha egy kicsi szkriptet készítünk e célra. Ez egy egyszerű szöveges állomány, amiben mindössze ez a `chmod-os` sor áll. Ezen állomány neve legyen – ki hinné – **printer**. Ezt hogy mindig tudjuk hívni, bármi könyvtárban is vagyunk, ugyanúgy „behuzalozzuk” a rendszerbe, mintha csak egy normálisan megírt, telepített program volna! Ennek érdekében készítünk a **/Programs** könyvtárban egy **Szkriptjeim** nevű alkönyvtárat. Adjuk ki a következő parancsot:

```
chown vz /Programs/Szkriptjeim
```

Ezzel a `vz` felhasználó tulajdonába adtuk ezt a könyvtárat. Ebbe készítünk (immár `vz`-felhasználóként) egy **0.0.0** nevű mappát – ez a verziószámra utal természetesen. Ezután – a **/Programs/Szkriptjeim** könyvtárban állva – adjuk ki e parancsot:

```
ln -s 0.0.0 Current
```

Ezzel létrehoztunk egy `Current` nevű szimbolikus linket, ami a `0.0.0` könyvtárra mutat. Ha most kilistázzuk e könyvtárat, mutatja is nekünk, hogy arra mutat:

```
pv@Csiszilla /Programs/Szkriptjeim]ls -l
összesen 0
drwxr-xr-x 4 vz pv 96 jan 17 19.57 0.0.0
lrwxrwxrwx 1 pv pv 5 febr 18 12.21 Current -> 0.0.0
pv@Csiszilla /Programs/Szkriptjeim]
```

Most csináljuk meg a `0.0.0` könyvtár belsejébe a **bin** nevű könyvtárat, s ide másoljuk be az elkészített **printer** nevű szkriptet! E könyvtárban adva alkalmazzuk e parancsot, hogy az állományt futtathatóvá tegyük:

```
chmod 777 printer
```

(Biztonsági szempontból elvileg jobb volna a `chmod 755 printer` parancs, de a `777`-et könnyebb megjegyezni. És egyelőre ne feledjük, hogy kezdők vagyunk, s így vélhetőleg nem a Pentagon rendszerére ügyelünk jelenleg...) Ezután adjuk ki a következő parancsot – immár rendszergazdaként:

SymLinkProgram Szkriptjeim

Ez belinkeli a megfelelő állományokat – jelen esetben az egyszem printer file-ot. Ettől kezdve a rendszerünkben létezik a **printer** nevű parancs, ami azt csinálja, ami ebbe az állományba bele van írva.

E módszer amiatt jó, mert így mostantól tetszőleges saját szkriptekkel bővíthetjük a rendszert – FOGJUK IS a későbbiekben! Annyi most már csak a dolgunk, hogy megírjuk a szkriptet, az állományt bemásoljuk a

/Programs/Szkriptjeim/0.0.0/bin nevű könyvtárba, ott a **chmod 777** paranccsal futtathatóvá tesszük, majd ismét kiadjuk a

SymLinkProgram Szkriptjeim

parancsot, s így immár az új szkriptünk is használhatóvá válik!

FIGYELMEZTETÉS! Ha ezt a módszert követjük amit fent leírtam, ezt a „sajátszkriptes” módszert, akkor NE csináljunk „printer” néven alias-t erre a chmod-os utasításra, mert nem találja meg akkor a „printer” nevű szkriptünket!

Mostantól tehát ha újraindítjuk a gépet, úgy tudunk nyomtatni, hogy

- csatlakoztatjuk a nyomtatót és bekapcsoljuk azt
- rendszergazdaként kiadjuk a **StartTask cups** parancsot
- rendszergazdaként kiadjuk a **printer** parancsot
- örülünk, mert nyomtathatunk.

Elvileg ezután le se kell állítanunk a démont, de ha mégis azt akarjuk, hogy ne tudjunk nyomtatni, rendszergazdaként a

StopTask cups

parancsot alkalmazzuk.

Most már minden jó?

Nem. Hamar rájövünk, hogy az OpenOffice alól remekül tudunk nyomtatni, de sajnos ha az mc-ben rákattintunk egy **pdf** dokumentumra, akkor az bár megnyílik az azt kezelő **xpdf** progiban, de nyomtatni bizony nem lehet ökelmét! Ha megpróbálkozunk a nyomtatással, felugrik a nyomtatási dialógusablak, de olyasmit kér tőlünk, hogy „**Print with command**”, s ezalatt egy fehér mező, ami azt várja, hogy valami parancsot üssünk be neki!

Ne késlekedjünk. Üssük be neki egészen egyszerűen azt, hogy

lpr

s ezután kattintsunk a Print gombra.

Rögvest ezután ellenőrizzük is le, bekerült-e a megfelelő állományba e parancs! Ez azt jelenti, hogy keressük meg a

/Programs/Xpdf/Settings/xpdfrc

fájlt, és benne ezt a részt:

Set the default PostScript file or command.

```
#psFile          "|lpr -Pmyprinter"  
psFile "|lpr"
```

A lényeg a félkövérrel kiemelt sor. Ha ez nincs, pótoljuk. A helye felismerhető az itt közölt szövegkörnyezetből.

Hát ennyit a nyomtatásról.

A cdfs kernelmodul és konverzió az audiállományok közt

A cdfs egy kis modul. A forrásból való fordítása a

<http://linux.birodalom.net/gobowiki/index.php/CDFS>

link alatt le van írva, picit talán macerás – egyszerűbb, ha a Tisztelt Olvasó egyszerűen letölti a kész modult amit csináltam, ezen, már talán jólismert oldalról:

<http://linux.birodalom.net/gobowiki/index.php/Csomagok>

A modul neve: **cdfs.ko**

Ezt aztán másold be a `/System/Kernel/Modules/2.6.23.8-Gobo/kernel` könyvtárba.

A CDFS használata előtt ki kell adnod az

`insmod /System/Kernel/Modules/2.6.23.8-Gobo/kernel/cdfs.ko` parancsot!

No most ezt mindig beírni elég macerás; természetesen csinálhatunk e parancsra egy alias-t, de egyszerűbb elintézni, hogy minden rendszerindítás-nál betöltse a modult! Ezt úgy érjük el, hogy az előbbi sort beírjuk a

`/Programs/BootScripts/Settings/BootScripts/BootUp` file-be mondjuk a végére.

Ha mindezzel megvagyunk, ellenőrizzük a `/System/Settings/fstab` állományt is! Ennek a sornak kell benne lennie:

`/System/Kernel/Devices/hdc /Mount/CDFS iso9660,cdfs noauto,users,ro 0 0`

Ugye emlékszünk rá, amikor először beszéltem az fstab állományról, írtam, hogy a „cdfs” jelentésére később térek majd ki! Nos, ez az a pillanat! A fenti sor azt mondja, hogy ha a CD amit beraktunk csatolható iso-állományként, akkor a gép tegye azt. Ha úgy nem lehetséges csatolnia, próbálja csatolni „cdfs” állományként! A csatolási pont a (remélhetőleg létező!) `/Mount/CDFS` tartalomjegyzék.

Na de mit a fene az a CDFS?!

Ez az, hogy (ha újraindítottuk a masinánkat...) most már bármikor mountolhatjuk a CD-nket e paranccsal:

`mount /dev/hdc`

Mondjuk ezt eddig is megtehettük, de MOST MÁR ha az egy audioCD, automatikusan aként fogja felmountolni, s a `/Mount/CDFS` tartalomjegyzékben láthatjuk az összes számot **wav**-ban. Ha meg nem audioCD, akkor normál adatCD-ként mountolja ugyanoda. És ez nagyon jó, mert íme milyen egyszerű számokat grebbelnünk most már audioCD-ről: egyszerűen egy fájlmásoló utasítás, s máris ott van a merevlemezünkön wav-ban! Amit aztán betömöríthetünk ha akarunk mp3-ba vagy ogg-ba... hogy hogyan? Nos ezzel foglalkozunk a következőkben!

MP3-WAV konverzió

Mindenekelőtt megtanulunk a fontosabb fájltypusainkból wav-ot készíteni. Kezdjük az mp3-mal! Ezt a következő paranccsal érhetjük el:

```
vz@Csiszilla ~]lame --decode n.mp3 n.wav
ID3v2 found. Be aware that the ID3 tag is currently lost when transcoding.
input:  n.mp3  (44.1 kHz, 2 channels, MPEG-1 Layer III)
output: n.wav  (16 bit, Microsoft WAVE)
skipping initial 1105 samples (encoder+decoder delay)
Frame# 7162/7163   128 kbps
```

Látható: a parancs után meg kell adni az átkódolandó mp3 file nevét, majd azt is, mi legyen az eredményként kapott wav file neve.

OGG-WAV konverzió

Ogg Vorbis fájlból wav gyártásához a Vorbis-Tools nevű progira van szükségünk. Függősége a Libao program. Tehát:

Compile vorbis-tools

Ez telepíti nekünk a Vorbis-Tools 1.1.1 progit és a függőség miatt a LibAO 0.8.8. programot is.

Ezután így használhatjuk:

```
vz@Csiszilla ~]oggdec v.ogg -o v.wav
OggDec 1.0.1
Decoding "v.ogg" to "v.wav"
      [ 99.5%]
vz@Csiszilla ~]
```

Azt hiszem ez is teljesen világos.

WMA-MP3 konverzió

A wma számok átkonvertálása:

Ez a kis szkript a könyvtárban levő összes wma kiterjesztésű fájlt átkonvertálja mp3 formátumúba. Itt már meg sem tartjuk a közben keletkező wav állományt. Ezt a szkriptet nevezzük el mondjuk úgy, hogy „**wmamp3**”, mentsük el a korábban megcsinált

/Programs/Szkriptjeim/0.0.0/bin könyvtárba, adjunk rá futási jogot a

chmod 777 wmamp3

paranccsal, majd adjuk ki e parancsot (rendszergazdaként):

SymLinkProgram Szkriptjeim

Ezután használhatjuk. Szóval a szkript tartalma:

```
#!/bin/bash
current_directory=$( pwd )
#Szóközök aláhúzássá konvertálása
for i in *.wma; do mv "$i" `echo $i | tr ' ' '_'`; done
#Nagybetűk kisbetűsítése
for i in *.[Ww][Mm][Aa]; do mv "$i" `echo $i | tr '[A-Z]' '[a-z]'`; done
#Az Mplayerrel wav-ba alakítjuk / majd bekódoljuk mp3-má a LAME-vel (Az alább
következő 2 sort egy sorba kell írni a for-tól a done-ig!)
for i in *.wma ; do mplayer -vo null -vc dummy -af resample=44100 -ao pcm
-waveheader $i && lame -m s audiodump.wav -o $i; done
#A filenév konvertálása
for i in *.wma; do mv "$i" ``basename "$i" .wma`.mp3"; done
rm audiodump.wav
```

Megjegyzendő, arra nem adok módszert, hogyan lehet wma-ba kódolni. A wma a Microsoft cég saját zenei formátuma, és ilyesmit „igaz Kocka” nem használ. Ha wma-hoz jut, azt átkódolja, és törli utána a wma-t!

WAV-OGG konverzió

Egyetlen wav szám ogg-ba konvertálása:

```
vz@Csiszilla ~]oggenc v.wav
Opening with wav module: WAV file reader
Encoding "v.wav" to
    "v.ogg"
at quality 3,00
    [ 99,8%] [ 0m00s remaining] |

Done encoding file "v.ogg"

    File length:  3m 23,0s
    Elapsed time: 0m 21,1s
    Rate:        9,6527
    Average bitrate: 99,9 kb/s

vz@Csiszilla ~]
```

Igazán egyszerű: az oggenc után meg kell adni a wav file nevét, s erre ő készít egy ugyanolyan nevű állományt, csak a kiterjesztése lesz ogg !

Ha a könyvtárban levő összes wav file-ból óhajtanánk ogg-ot készíteni:

```
#!/bin/bash
current_directory=$( pwd )
#Bekódoljuk ogg-gá az oggenc-vel
for i in *.wav ; do oggenc $i; done
```

A fenti szkriptet mentsük el mondjuk **mindenwavogg** néven a szkriptjeink közé, és... de a többit már úgyis tudod, kedves Olvasó! A használata egyszerűen:

mindenwavogg

MP3-OGG konverzió

Ezután könnyű olyan szkriptet írunk, ami rögtön ogg-ba kódol mp3 számot.

Ez a kis szkript a paraméterként megadott nevű mp3 állományt átkonvertálja ogg formátumúvá. Mentsük el mondjuk **mp3ogg** néven, adjunk rá futtathatósági jogot, tegyük be ugyanoda, ahová a **wmamp3** szkriptet is tettük, symlinkeljük be újra a szkriptjeinket, s ezután a használata mondjuk az xxx.mp3 nevű file-ra: **mp3ogg xxx**

A szkript:

```
#!/bin/bash
lame --decode $1.mp3 $1.wav
oggenc $1.wav
rm $1.wav
ls
```

A végén mint látható, még előzékenyen ki is listázza nekünk az aktuális könyvtárat. Ha ezt nem akarjuk, a végéről hagyjuk le az „ls” parancsot. E szkriptnél a paraméternél elég megadni az mp3 file nevét a kiterjesztés nélkül, azaz ne úgy hívjuk, hogy mp3ogg ezaneve.mp3 hanem így: mp3ogg ezaneve

A szkript először készít egy ideiglenes wav állományt, azt ogg-ba kódolja, majd törli az ideiglenes állományt.

Egy példa a futására, az input file neve v.ogg:

```
vz@Csiszilla ~]mp3ogg v
input: v.mp3 (44.1 kHz, 2 channels, MPEG-1 Layer III)
output: v.wav (16 bit, Microsoft WAVE)
skipping initial 1105 samples (encoder+decoder delay)
Frame# 7810/7810 128 kbps MS
Opening with wav module: WAV file reader
Encoding "v.wav" to
      "v.ogg"
at quality 3,00
      [ 99,6%] [ 0m00s remaining] -

Done encoding file "v.ogg"

      File length: 3m 23,0s
```



```
Elapsed time: 0m 21,2s
Rate:          9,5997
Average bitrate: 104,4 kb/s
```

```
Desktop n.mp3 n.wav v.mp3 v.ogg zardbe.odt
vz@Csiszilla ~]
```

Ennek a fordítottja, az ogg-mp3 konverzió:

OGG-MP3 konverzió

Ez a szkript a megadott ogg formátumú állományt átkonvertálja mp3 formátumúvá. A szkript:

```
#!/bin/bash
oggdec $1 -o tmp_wav
lame -h tmp_wav $1.mp3
rm -f tmp_wav
```

Ezt is nevezzük el, mondjuk **oggmp3** néven, s integráljuk a rendszerünkbe a **Szkriptjeim** könyvtárba a korábbi példákhoz hasonlóan. Megjegyzés: itt a teljes nevet kell megadni input paraméterként, tehát így hívod:

```
oggmp3 ezaneve.ogg
és NE így:
oggmp3 ezaneve
```

Ha ez túl kényelmetlen, akkor ezt a szkriptet használd helyette, itt már nem kell megadnod a kiterjesztést is:

```
#!/bin/bash
oggdec $1.ogg -o tmp_wav
lame -h tmp_wav $1.mp3
rm -f tmp_wav
```

Én is ezt az utóbbi változatot használom. Egy példa a futására (az állomány neve v.ogg):

```
vz@Csiszilla ~]oggmp3 v
OggDec 1.0.1
Decoding "v.ogg" to "tmp_wav"
[ 99.5%]
LAME 3.97 32bits (http://www.mp3dev.org/)
Using polyphase lowpass filter, transition band: 16538 Hz - 17071 Hz
Encoding tmp_wav to v.mp3
Encoding as 44.1 kHz 128 kbps j-stereo MPEG-1 Layer III (11x) qval=2
Frame          | CPU time/estim | REAL time/estim | play/CPU | ETA
```

```

7810/7810 (100%)| 0:34/ 0:34| 0:35/ 0:35| 5.9515x| 0:00
-----
kbps      LR    MS  %      long switch short %
128.0     27.8  72.2      99.5  0.3  0.2
Writing LAME Tag...done
ReplayGain: -4.3dB
vz@Csiszilla ~]

```

WAV-MP3 konverzió

Ez a szkript a megadott wav file-ból gyárt MP3-at. Mentsük el a szkriptjeink közé **wavmp3** néven, symlinkeljük be... Használata mondjuk a v.wav file esetén:

```
wavmp3 v
```

Látható, a kiterjesztést nem kell megadni neki, tudja, hogy az .wav, s ebből gyárt egy ugyanolyan nevűt .mp3 kiterjesztéssel. A szkript tartalma:

```
#!/bin/bash
lame -h $1.wav $1.mp3
```

Ha a könyvtárban levő összes wav file-ból óhajtánánk mp3-at készíteni:

```
#!/bin/bash
current_directory=$( pwd )
#Bekódoljuk mp3-má a LAME-vel
for i in *.wav ; do lame -m s "$i" "`basename "$i" .wav`.mp3"; done
```

A fenti szkriptet mentsük el mondjuk mindenwavmp3 néven a szkriptjeink közé, és... de a többit már úgyis tudod, kedves Olvasó! A használata egyszerűen:

```
mindenwavmp3
```

WAV-MIDI konverzió

Igen, ilyet is lehet csinálni! Ehhez a „**waon**” nevű progira van szükségünk. A projekt honlapja: <http://waon.sourceforge.net/>

E sorok írásakor az aktuális verzió a 0.9 számú.

A telepítése forrásból történik: a letöltött forrást kicsomagoljuk valami nekünk tetsző könyvtárba, abba belépünk, majd kiadjuk a

```
make
```

parancsot. Ezután (root-ként) hozzuk létre a `/Programs` könyvtárban a következő könyvtárszerkezetet:

```
/Programs/Waon
/Programs/Waon/0.9
/Programs/Waon/0.9/bin
/Programs/Waon/0.9/man
/Programs/Waon/0.9/man/man1
```

A `/Programs/Waon` könyvtárba belépve adjuk ki e parancsot:

```
ln -s 0.9 Current
```

majd abból a könyvtárból amiben a `make` parancso kiadtuk korábban másoljuk át a

```
gwaon
pv
waon
```

fájlokat a `/Programs/Waon/0.9/bin` könyvtárba, és másoljuk át a

```
waon.1
```

fájlt is, ezt a `/Programs/Waon/0.9/man/man1` könyvtárba. Majd adjuk ki e parancsot:

```
SymlinkProgram waon
```

Ezután a progink már használható. Használata egyszerű, és parancssoros:

```
waon -i inputfile.wav -o outputfile.mid
```

vagyis megadjuk paraméterként a bemeneti wav fájlt, s hogy milyen néven hozza létre a midi fájlt.

Csodát persze ne várjunk tőle. Az eredmény minősége nagyban függ attól, hány hangszer szól és milyen tisztán az input fájlban...

MIDI-WAV konverzió

Ez sokkal egyszerűbb, mint az előző, WAV-MIDI konverzió esete. Ehhez csak a `Timidity++` progira van szükségünk, aami különben is lejátssza nekünk a midi számokat. Mi sem egyszerűbb annál, hogy így indítsuk, mint az az alábbi példán látszik:

```
vz@Csiszilla ~/waon]timidity -Ow ali.mid
```

```
Playing ali.mid
MIDI file: ali.mid
Format: 1 Tracks: 12 Divisions: 240
Track name: BrassSection Arranged by...
Track name: SynthVoice ... Gene Cowan
Track name: Violin
Track name: Grand Piano
Track name: Pizz Strings
Track name: Fantasia
Track name: Sitar
Track name: StrEnsemble2
Track name: StrEnsemble2
Track name: Drums
Track name: Acoustic Bass
Output ali.wav
Playing time: ~199 seconds
Notes cut: 0
Notes lost totally: 0
vz@Csiszilla ~/waon]moc
Running the server...
Trying JACK...
JACK tmpdir identified as [/tmp]
Trying ALSA...
```

És kész is az egész, a lényeg a **-ow** kapcsoló, s erre az ali.mid számból készített nekem egy ali.wav nevű fájlt! Jó tudni azonban, a konvertálás „valós idejű”, azaz kb ugyanannyi ideig tart, mint ameddig a tényleges zeneszám lejátszása tartana...

Az Mplayer és a filmek konvertálása-vágása

Ehhez mindenekelőtt telepítsük az **xvidcore** csomagot.

Compile xvidcore

Ez az **xvidcore 1.1.0-r2** változatát telepíti.

Ezután tiltsuk le a régi mplayerünket:

DisableProgram mplayer

majd fordítsunk egy újat:

Compile mplayer

Ez az **Mplayer 1.0rc2-r3** változatát telepíti.

Minderre amiatt van szükség, hogy az mplayer úgy legyen lefordítva, hogy együtt tudjon működni a rendszerünkbe az imént telepített **xvidcore** csomaggal (is).

MPG file AVI-ba tömörítése

Miután ezzel megvagyunk, lássuk, miként kell átkonvertálni egy filmet! Mondjuk van egy avi filmünk, legyen a neve mondjuk k.avi, s ezt át szeretnénk konvertáltatni hogy kisebb legyen a mérete, s legyen az új változat neve mondjuk kk.avi – ekkor adjuk ki e parancsot:

```
mencoder k.avi -ovc xvid -xvidencopts bitrate=700 -oac mp3lame -o kk.avi
```

Ha a filmünk mondjuk mpg formátumú, gyakorlatilag ugyanezt a parancsot kell kiadnunk:

```
mencoder k.mpg -ovc xvid -xvidencopts bitrate=700 -oac mp3lame -o kk.avi
```

(Feltételezzük a parancs kiadásakor, hogy ugyanabban a könyvtárban állunk, ahol a konvertálandó állomány is van).

Elemezzük ki e parancs jelentését! Mindenekelőtt: a mencoder progit indítjuk, ez együtt települt nekünk az mplayer-rel. Ez végzi nekünk a legtöbb filmes konverziót. Utána megadandó a konvertálandó file neve. Majd azt mondjuk, hogy az **output video codec** (ennek rövidítése az „ovc”) xvid legyen – sok másfélét is beállíthatnánk, akit érdekel olvassa el az mplayer man oldalát –, s megadjuk ezen codec egy fontos paraméterét (opcióját): a bitráta legyen 700. Minél nagyobb e szám, annál jobb a minőség, de annál nagyobb az állományméret is. Nekem, kisképernyős monitorra, a 700-as szám bevált. Utána az **output audio codec**-et adjuk meg („oac”), hogy tehát a film hanganyaga mp3-ba legyen kódolva. Lehetne ogg-ba is, de akkor win alatt nemigen lehetne lejátszani a filmet (illetve le lehetne, csak hangja nem lenne...) majd megadjuk az output file nevét.

Megjegyzendő, hogy nem csak avi vagy mpg fájl adhatunk meg az input file

helyén! Ide tetszőleges fájlformátumú fájlt berakhatunk, már amennyiben rendelkezünk az ehhez szükséges codeckel. Minthogy élünk a gyanúperrel, hogy a CD-re helyhiány miatt nem tették fel az összeset, telepítsük őket mind (13 Megás cumó tömörítve!):

Compile mplayer-codecs-all

Ez a **20071007-r1** verziójú csomagot telepíti.

Ezek után könnyű filmeket átkonvertálnunk xvid-be. Mindegy milyen az input file formátuma. (Többnyire mpg fájlokat szoktunk így AVI-ba tömöríteni). Megjegyzem, az Mplayernek is és a mencoder-nek is rengeteg opciója van, TÉNYLEG érdemes áttanulmányozni a kezelési utasítást!

A bitráta kiszámolása

Előfordulhat, hogy valaki pontosan akarja tudni, mekkora bitrátát (ugye a „bitrate=...” rész a parancsban) kell megadnia valamekkora mérethez. Íme a képlet:

**bitráta = (célméret MegaBájtban - hangsávok mérete MegaBájtban) * 1024 * 1024 /
hossz másodpercben * 8 / 1000**

Példa: Ha van egy másfél órás filmed, amin mondjuk 40 mega a hangsáv (átlagos értékek...) és ezt te egy 700 megás CD-re akarod betömöríteni, akkor az

bitráta = (700-40)*1024*1024/(90*60)*8/1000 s ennek eredménye:

1025.274 ami azt jelenti, hogy adj meg 1000-et. (A 90*60 a másfél óra másodpercben...) Megjegyzendő, hogy nekem ici-pici monitorom van, így nekem a 700-as bitráta is elég. Tehát itt az egyéni igények dominálnak.

Na de honnan tudjuk, mekkora a hangsáv mérete?! Hát kérem, például onnan, hogy leszeded a hangsávot róla, s megnézed, hány mega. Ez itt van leírva: Egy videoállomány hanganyagának leszedése

Jó tudni azért, hogy még az így kiszámított adatok is csak közelítésnek tekinthetők, továbbá, az ideális bitráta nagy mértékben függ attól, milyen filmet tömörítünk: amiben kevés a mozgás, oda kisebb bitráta is jó minőséget ad, ahol azonban sok és gyors mozgás van (például a *Mátrix*, vagy valami „karatés” film, meg olyan amiben számos üldözéses jelenet van) oda bizony nagyobb bitráta kell, másképp nagyon „kockásnak” látjuk majd az eredményt. Általában a videókódolásról **magyar nyelven** remek, részletes cikk található az Mplayer weboldalán e link alatt:

<http://www.mplayerhq.hu/DOCS/HTML/hu/menc-feat-dvd-mpeg4.html>

MPG filmek összefűzése

Ha már szóba kerültek az mpg fájlok: hogyan lehet összefűzni őket?! Nos, ennek csak akkor van értelme, ha ugyanolyan a bitráta, képméret stb. De többnyire ez szokott lenni a helyzet, mert általában azért darabolják a filmeket, hogy el lehessen küldeni e-mailben darabonként.

Az mpg file-ok összefűzése Linux alatt még egyszerűbb mint W\$ alatt - ott a copy paranccsal kell megoldani a /b kapcsolót használva, mi linuxosok azonban ezt így tesszük parancssorból:

```
cat file1 file2 ... fileN > egyben.mpg
```

Azaz a cat után felsoroljuk az összefűzni kívánt állományokat a kívánt sorrendben (kiterjesztésüket is megadva), majd az outputot átirányítjuk egy file-ba, itt a példában „egyben.mpg” lesz a kívánt összefűzött file neve.

Konkrét példa:

```
cat film1.mpg film2.mpg film3.mpg > film.mpg
```

Filmek darabolása

És hogyan lehet darabolni egy filmet? Ezt is megmutatjuk.

Gyakori feladat ugyanis, hogy valami nagy filmet több darabra vágjunk (többnyire kettőbe) például hogy ráférjen egy CD-re. Természetesen ezt is a **mencoder**rel oldjuk meg.

50 percnél elvágja kétfelé:

```
mencoder -ovc copy -oac copy -endpos 0:50:00 -o film_első_resz.avi teljes_film_neve.avi
```

```
mencoder -ovc copy -oac copy -ss 0:50:00 -o film_második_resz.avi teljes_film_neve.avi
```

Ha a film közepéből akarunk kivágni egy darabot, mondjuk a huszadik perctől kezdve 5 percnyi:

```
mencoder -ovc copy -oac copy -ss 0:20:00 -endpos 0:5:00 -o darab.avi teljes_film_neve.avi
```

Az időadatok megadását a következőképp értsd: **óra:perc:másodperc**

AVI filmek összefűzése

Az avi fájlokat nemcsak darabolni kell tudnunk, de összefűznünk is. Erre ajánlanak mindenféle cuki kis programcskákat az Internet különböző

virtuális sikátoraiban, de mi ezt is nyersen és durván parancssorból végezzük el, méghozzá így:

```
cat file1.avi file2.avi >ideiglenes.avi  
mencoder ideiglenes.avi -noidx -ovc copy -oac copy -o teljes.avi -
```

Az összefűzött file neve a példában „teljes.avi”. A mencoder-rel kezdődő parancssor végére kötelező a záró „-” jel, enélkül nem működik! A végén az „ideiglenes.avi” file letörölendő, nem kell már. (pld törölhető az `rm ideiglenes.avi` paranccsal vagy az mc (azaz a Midnight Commander) segítségével.

Egy másik probléma:

Egy videoállomány hanganyagának leszedése

Ez is igen gyakori kérdés. Többnyire ugyan nem az egész hanganyag kell, csak valami rész belőle, de ehhez a hangsávot előbb le kell menteni, csak azután lehet megvágni a hanganyagot (pld az Audacity progival).

A hangsáv lementése nem a mencoder hanem az mplayer programmal történik (bár e kettő gyakorlatilag ugyanaz). Rém egyszerű:

```
mplayer videofile_neve -dumpaudio
```

Konkrét példa:

```
mplayer film.avi -dumpaudio
```

Erre ez létrehoz a könyvtárban egy `stream.dump` nevű file-ot, és EZ NEM MÁS MINT A HANGSÁV, MP3 FORMÁTUMBAN! Feltéve, hogy a filmben a hang eleve mp3 formátumban volt kódolva... Ezesetben tehát csak át kell neveznünk a `stream.dump` file-ot mondjuk `film.mp3` -ra, és máris hallgathatjuk!

Honnan tudjuk, mp3-ban van-e? Egyszerű. Átnevezzük a `stream.dump`-ot valami.mp3 -ra, rákattintunk, s ha az xmms vagy az audacious vagy a moc vagy más, az épp aktuális zenelejátszónk nem játssza le, gáz van: nem mp3!

Ekkor a megoldás: mielőtt kidumpoltatnánk, át kell konvertálni a videót, hogy a hangsáv az mp3lame-vel legyen kódolva, tehát mp3-ban, ez a korábban idézett példa szerint:

```
mencoder miből -ovc xvid -xvidencopts bitrate=700 -oac mp3lame -o mibe.avi
```

(Sok más módszer is lehetséges, a lényeg a `-oac mp3lame` rész. Például gyorsabb, ha nem kódoltatjuk vele a video-streamet mert mi a csudának, ha úgyis csak a hangsáv átalakítása a célunk! Ezesetben ezt a parancsot adjuk ki:

```
mencoder miből -ovc copy -oac mp3lame -o mibe.avi
```


). És ezután adjuk ki az
mplayer mibe.avi -dumpaudio
parancsot, majd átnevezzük a stream.dump file-ot valami.mp3-ra és
hallgathatjuk vagy nekikezdünk vágni...

Hang nélküli videó készítése

Azaz hogyan távolítsuk el a hangsávot a videóról, hogy néma legyen? Így:

```
mencoder video_vanhangja.avi -ovc copy -nosound -o video_nincshangja.avi
```

Más hangsáv felrakása a videóra

Ehhez először a „Hang nélküli videó készítése” részben leírtak szerint távolítsuk el róla az eredeti hangsávot. Majd:

```
mencoder video_nincshangja.avi -ovc copy -audiofile ujhangja_ez_lesz.mp3 -oac  
copy -o video_ujhangja_van.avi
```

(A fenti parancs EGY SORBA írandó!)

DVD legrebbelése (azaz parancssori DVD rip)

Na de miért parancssorból?! Hát nem jó valamelyik rippelőprogram?! Lehet hogy jó, nem próbáltam végig mindet. Az azonban amivel próbálkoztam, egy másolt DVD-t úgy szedett le AVI-ba, hogy nem volt hangja. Egy barátom azt mondta, ez amiatt lehetséges, mert ez már eleve másolt DVD, s a DVD másolása közben valahogy „összetömrítették”, s így nem szabványos a formátuma! De különben is, eddig is mindent parancssorból intéztünk... A megoldás tehát *igenis* parancssorból:

```
mencoder dvd:// -ovc xvid -xvidencopts bitrate=700 -oac mp3lame -o film.avi
```

Ebben az esetben a kimenő állomány neve film.avi lesz, s az alapértelmezett hangsávval szedi le. Na de ez nekünk magyaroknak csak akkor jó, ha a filmhez egyetlen hangsáv tartozik, másképp valószínűleg angolul halljuk majd... S pláne ha „nem szabványos” a film, akkor aztán tényleg ki kell deríteni, hol a magyar hangsáv!

Ez így történik:

```
mplayer -v dvd://
```

Erre eszméletlenül sok szöveget listáz ki a képernyőre (a control-c-vel

lelőhető), s ebben vissza kell keresnünk azt a részt, amikor a hangsávokról hangcsatornákról-audiosávokról povedál. Itt számokat fogunk látni, valószínűleg egy 128-as számjegyűt is, na ez lesz majdnem bizonyosan az alapértelmezett, de ez nekünk nem jó, mert ha jó lenne, az előző megoldás az alapértelmezett hangsávval is jó lenne nekünk! Kell látnunk még más számokat is, tegyük fel a következő a 130! Ha mázlink van és ez a magyar (amit csak úgy tudunk meg, hogy kipróbáljuk, méghozzá az alábbi utasítással), akkor ezt a parancsot kell kiadnunk:

```
mencoder dvd:// -aid 130 -ovc xvid -xvidencopts bitrate=700 -oac mp3lame -o film.avi
```

Ha még sok más szám is van, akkor javasolt, hogy tömörítsünk így be mondjuk 5 percet, azután lőjük le a progit (Ctrl-C) és hallgassunk bele! Ha nem magyar (vagy az a nyelv ami kell nekünk), próbálkozzunk egy másik számmal a **-aid** után!

Egy példa, félkövérrel kiemelve a hangsávokról szóló részt:

```
vz@Csiszilla /Mount/HDA7/Users/vz]mplayer -v dvd://
MPlayer 1.0rc2-4.1.2 (C) 2000-2007 MPlayer Team
CPU: Intel(R) Celeron(R) CPU 2.40GHz (Family: 15, Model: 2, Stepping: 9)
CPUflags: MMX: 1 MMX2: 1 3DNow: 0 3DNow2: 0 SSE: 1 SSE2: 1
Compiled for x86 CPU with extensions: MMX MMX2 SSE SSE2
get_path('codecs.conf') -> '/Users/vz/.mplayer/codecs.conf'
Reading /Users/vz/.mplayer/codecs.conf: Can't open
'/Users/vz/.mplayer/codecs.conf': No such file or directory
Reading /Programs/MPlayer/1.0rc2/etc/mplayer/codecs.conf: Can't open
'/Programs/MPlayer/1.0rc2/etc/mplayer/codecs.conf': No such file or directory
Using built-in default codecs.conf.
Configuration: --prefix=/Programs/MPlayer/1.0rc2 --enable-gui
--win32codecsdir=/System/Links/Shared/mplayer-codecs
--codecsdir=/Programs/MPlayer/1.0rc2/share/mplayer-codecs --enable-largefiles
--language=all --disable-arts
CommandLine: '-v' 'dvd://'
init_freetype
get_path('font/font.desc') -> '/Users/vz/.mplayer/font/font.desc'
font: can't open file: /Users/vz/.mplayer/font/font.desc
font: can't open file: /Programs/MPlayer/1.0rc2/share/mplayer/font/font.desc
Using MMX (with tiny bit MMX2) Optimized OnScreenDisplay
get_path('fonts') -> '/Users/vz/.mplayer/fonts'
Using nanosleep() timing
get_path('input.conf') -> '/Users/vz/.mplayer/input.conf'
Can't open input config file /Users/vz/.mplayer/input.conf: No such file or
directory
Can't open input config file /Programs/MPlayer/1.0rc2/etc/mplayer/input.conf:
No such file or directory
Falling back on default (hardcoded) input config
get_path('.conf') -> '/Users/vz/.mplayer/.conf'

Playing dvd://.
get_path('sub/') -> '/Users/vz/.mplayer/sub/'
URL: dvd://
```

Reading disc structure, please wait...
There are 1 titles on this DVD.
There are 23 chapters in this DVD title.
There are 1 angles in this DVD title.
DVD successfully opened.
audio stream: 0 format: ac3 (5.1) language: en aid: 128.
audio stream: 1 format: ac3 (5.1) language: hu aid: 129.
number of audio channels on disk: 2.
subtitle (sid): 0 language: en
subtitle (sid): 1 language: hu

Most persze mondhatja valaki, mi a fenének kipróbálni, az adott számnál szereplő hangsáv-e a magyar, amikor oda van írva mellé a nyelvkód, hogy „hu”! Nos, óvatosságból. Találkoztam már ugyanis olyan videóval, aminél más volt a nyelv, mint amit írtak kódként. Konkrétan, a magyar nyelvet a francia helyén találtam meg. A magyar helyén pedig az orosz volt...

Video-CD (VCD) betömörítése

Ez rém egyszerű. Csatoljuk fel a lemezt (mondjuk így: `mount /dev/hdd`), s feltételezve hogy a csatolási pontunk a `/Mount/CD-ROM` e paranccsal érünk célt:

```
mencoder /Mount/CD-ROM/mpegav/avseq01.dat -ovc xvid -xvidencopts bitrate=700  
-oac mp3lame -o film.avi
```

(Ez egyetlen parancs, nem 2, csak nem fért ki 1 sorba!)

E lemezeken ugyanis általában van egy **mpegav** nevű könyvtár, benne az **avseq01.dat** nevű fájjal, ami maga a tulajdonképpeni videóanyag. (amennyire tudom, ez egy egyszerű mpg file, csak a kiterjesztése más. Persze tévedhetek is).

A GoboHide és a rejtett tartalomjegyzékek

No ez igazi GoboLinux-specialitás! Kezdő ismerkedésként add ki e parancsot:

```
vz@Csiszilla ~]gobohide --help
gobohide: Elrejtí/Megmutatja a tartalomjegyzéket

-h, --hide      A tartalomjegyzék elrejtése
-u, --unhide    A tartalomjegyzék engedélyezése
-l, --list      A rejtett tartalomjegyzékek kilistázása
-f, --flush     A rejtési lista kiürítése
--version      A program verziójának kiírása
--help         Ennek az üzenetnek a megjelenítése
```

Na, ez a parancs legalább magyarul beszél hozzánk! Angolul ezen üzenetek a következőképp festenek (egy nem magyar telepítésű GoboLinux alatt például):

```
vz@Csiszilla ~]gobohide --help
gobohide: Hide/Unhide a directory

-h, --hide Hide the directory
-u, --unhide Unhide the directory
-l, --list List the hidden directories
-f, --flush Flush the list
--version Show the program version
--help Show this message
```

Na de mit csinál ez a progí?!

A GoboHide egy a GoboLinuxhoz tartozó kernelpatch (=kernelfolt). A GoboLinux speciális könyvtárstruktúráját kezeli oly módon, hogy az abban levő, a hagyományos linux-disztribúciókban alkalmazott filerendszer-hierarchiát emuláló szimbolikus kötéseket listázáskor elrejtí a felhasználók elől. Elvileg azonban más könyvtárak is elrejthetőek vele, illetve az elrejtettek felszabadíthatók, azaz e kernelpatch tényleges rejtett tartalomjegyzékeket valósít meg kerneltámogatással a GoboLinuxban. Erre a GoboLinuxnak különben kizárólag „esztétikai okokból” van szüksége (A GoboHide által elrejtett linkekre viszont amiatt, mert maga a GoboLinux nem POSIX-kompatibilis a filerendszer-hierarchiáját illetően).

Vagyis ha csináltunk a gyökérkönyvtárba egy **/opt** nevű könyvtárat, és azt akarjuk, hogy ez ne jelenjék meg listázáskor, akkor adjuk ki e parancsot:

```
gobohide -h /opt
```

Ez lehetővé teszi, hogy a fájlrendszer egy darabját elrejtjük. De ne aggódj, a rendszergazdának megvan a hatalma ahhoz, hogy lekérdezze a kernelt a rejtett bejegyzésekkel kapcsolatban, tehát mindig biztos tudomásod lehet

afelől, mit rejtettek el a rendszeredben:

```
pv@Csiszilla /]gobohide -l
Hidden directories:
/opt
```

Illetve magyar nyelven:

```
pv@Csiszilla /]gobohide -l
Rejtett tartalomjegyzékek:
/opt
```

És a legszebb az egészben: még mindig hozzá tudsz férni a fájljaidhoz ezeken a rejtett belépési pontokon keresztül! Be tudsz lépni e könyvtárakba például a `cd` paranccsal.

A GoboHide különben jelenleg a GoboLinux 014-ben még épp hogy csupán angol és magyar nyelven tud. Más nyelvekre még nincs internacionalizálva. A magyarítását én magam készítettem, engem kell szídni érte.

E kernelpatch értelme az, hogy a GoboLinux fájlrendszer-hierarchiája nem követi a POSIX-ajánlásokat. A hagyományos Linux disztribúciók gyökérkönyvtára tele van olyan könyvtárakkal, hogy

```
/etc
/usr
/var
/tmp
/sys
/bin
/sbin
/lib
/dev
/proc
```

- a Gobóé ehhez képest roppantul egyszerű és áttekinthető! Emiatt is szeretem ezt a disztrót... igen ám, de a programok többnyire a régi könyvtárrendszerhez vannak megírva. Emiatt a régi struktúrát a GoboLinux emulálja úgy, hogy e hagyományos könyvtárak helyére ugyanilyen nevű szimbolikus linkeket rak, melyek e könyvtárak funkcióját ellátó GoboLinux-beli könyvtárakra mutatnak. E symlinkekre tehát a GoboLinuxnak a kompatibilitás miatt van szüksége. Még így is rondák, látni sem akarjuk őket! No épp ezen elrejtést – hogy ne lássuk őket – végzi el nekünk a GoboHide.

Hogyan törjük fel a saját rendszerünket?

Rögvest felmerül a kérdés: miért kéne feltörni épp a saját rendszerünket?! Nos, ennek legalább 2 oka lehet. Egyik, hogy túl hülye voltál, és elfelejtetted a jelszavadat. Bocs de erre nincs szebb szavam. Saját jelszónk elfelejtése iszonyú „lamerság”, hülyeség. Kizárólag a teljesen „egységsugarú”, abszolút kezdő valakikre jellemző, akiket tulajdonképpen nem is érdekel a számítástechnika. Sajnos általában ráadásul épp a rendszergazdai jelszót felejtik el egyesek, ha sokáig nem kell használniuk azt, mert nem végeznek semmi komolyabbat a rendszerükkel. Hát ez irtó ciki.

Ritkábban az is megtörténhet, hogy valaki belépett a rendszerünkbe, és hogy kicsesszen velünk, megváltoztatta a jelszót. Ekkor is fel kell törnünk a saját rendszerünket.

Azt mondják, a Linux biztonságos – az is, ennek ellenére szinte kétségbeejtően egyszerű feltörni, azesetben, ha a számítógép hardvere a kezünkbe kerül. Ellenkező esetben, távolból feltörni borzalmasan nehéz. Na de ha odaadják a kezünkbe... mert jegyezzük meg: nem létezik olyan módszer, mellyel megvédehetnénk a számítógépet akkor, ha azt odaadják egy hacker kezébe!

Nos, a mi gépünk remélhetőleg ott van a mi kezünkben. Na nehogy már ne tudjunk bejutni a rendszerébe!

Persze hogy megteesszük. Nem kell más hozzá, csak a GoboLinux LiveCD, amiről telepítettük a rendszert. De akármiféle más LiveCD is megteszi, tehát más disztrók LiveCD-je is. Lényeg, hogy legalább egy parancssoros kis Linux rendszer elinduljon róla. Ebből rögvest következik is, hogy természetesen úgy kell beállítsuk a BIOS-t, hogy a CD-ről „butuljon”. A LiveCD rendszerek elindulnak minden jelszó nélkül, vagy pedig ráírják a CD-re a jelszót, azzal tehát nem lesznek gondjaink.

Miután a Live rendszer „felállt”, csatoljuk fel a merevlemez azon partícióját valahová, amely partíción a feltörendő rendszer van. Legyen ez mondjuk a `/Mount/media` könyvtár. Teljesen mindegy hova csatoljuk, a lényeg, hogy ezután be tudunk lépni abba a partícióba, célszerűen az `mc` progi segítségével. Ezután lépünk be a `/System/Settings` könyvtárba! Itt kell találjunk két file-ot, melyek a

shadow

és a

shadow-

nevekkel büszkélkedhetnek.

Ezek tartalma valami ilyesmi:

```
pv:aZxY.jEl6jiug:13834::::::  
nobody:!:9797:0::::::  
uucp:!:9797:0::::::  
ftp:!:12105:0:99999:7:::  
fibo:!:12154:0:99999:7:::
```

```
sshd!:12341:0:99999:7:::  
vz:aS1YyIAQcMoPA:12990:0:99999:7:::  
bin!:13885:0:99999:7:::
```

Látható, e fájlban szerepel mindkét felhasználó azonosítója, amit létrehozunk: ott a **pv**, aki rendszergazda a rendszerünkben, s ott a **vz** is. Ami ezután szerepel, a sok értelmetlen betű és szám, az a jelszó maga kódolva. Valamiért több darabban kódolva tárolja a rendszer, vagy talán más infókat is letárol itt a jelszóval kapcsolatban, például lejáratí időt – bevallom ezt nem tudom. De a feltöréshez nem is kell tudni. Lényeg az, hogy elég mindössze annyit tennünk, hogy annál a felhasználónál, amelynek jelszavát úgy mond „fel akarjuk törni” - s ami többnyire ugye a rendszergazda – e részt brutálisan kitöröljük, ügyelve azonban rá, hogy a kettőspontok megmaradjanak. Azokból ha nem tévedek nyolcnak muszáj lenni. Azaz ha a pv és a vz felhasználókat egyaránt „jelszótlanítjuk”, ezen állomány így kell kinézzen utána:

```
pv::::::::::  
nobody*:9797:0:::::  
uucp*:9797:0:::::  
ftp!:12105:0:99999:7:::  
fibo!:12154:0:99999:7:::  
sshd!:12341:0:99999:7:::  
vz::::::::::  
bin!:13885:0:99999:7:::
```

A változást (a megváltoztatott állományt) ne feledjük elmenteni...

E változtatást MINDKÉT állományban el kell végezni, a **shadow** -ban is és a **shadow** -nevűben is!

Megjegyzés: e könyvem első változatának közzététele után számos gúnyos kritikát kaptam amiatt, hogy minek írok olyan hülyeséget, hogy a változtatást el kell végezni a **shadow** -nevű állományban is, amikor az csak másolat! Nos, nem próbáltam ki, enélkül is feltörhető-e a rendszer, de ha igen, akkor sem tartom feleslegesnek ezen állomány megváltoztatását is. Ez ugyanis a számomra egyszerűen elvi kérdés. Az elv az, hogy ha van valami, ami valami másnak a másolata, akkor illik, hogy ugyanúgy nézzen ki. Enélkül nem mondhatjuk, hogy beavatkozásunk után konzisztens állapotban hagytuk a rendszert. Különben is: a másolat amiatt másolat, mert valamiért megőrzi a rendszert mintegy biztonsági okokból, tartaléknak. Ez lehet például olyan ok is, hogy valamikor vissza akarja állítani ebből az eredeti állapotot. S mi ugye nem szeretnénk, ha visszaállítaná az általunk ismeretlen jelszavakat...

De ismétlem, az elv a lényeg: ha valamit turkálunk a rendszer mélyén, hagyjuk magunk után a rendszert rendben, tisztességes állapotban. Azaz igyekezzünk olyan végeredményt produkálni, amit a szokásos rendszerrutinok is produkálnának, ha azok meghívása árán jutnánk el a jelenlegi állapothoz. Ez a módszer, ha megszokjuk, minimalizálja a lehetséges hibákat, bakikat.

Ezután indítsuk újra a rendszert, ezúttal a merevlemezről. Felhasználói nevet fog kérni, de utána jelszót már nem! S ha mondjuk először a vz felhasználóként jelentkeztünk be, majd utána a

su -

paranccsal rendszergazdai jogkört szeretnénk magunknak, akkor sem fog jelszót kérni.

A módszer különben működik a nem GoboLinux disztróknál is, csupán ott a két shadow-fájlt a **/etc** könyvtárban kell keresned, mert ott lesznek.

Ez igazán nem volt ördögösség, mi?!

Áttérés a SithWM ablakkezelőre

Na de miért térnénk át másik ablakkezelőre a KDE-ről?! - kérdezhetné valaki. Nos amiatt, mert a KDE nem is ablakkezelő, hanem „desktop environment”, ezt korábban már említettem. Aztán, túlságosan hasonlít a Windowshoz. Igaz Kocka pedig nem mondhatni, hogy különösképp szeretné azt az oprendszert. (Most igyekeztem nagyon udvariasan fogalmazni...) Megjegyzem, ahogy a KDE a Windowst igyekszik utánozni külsőségeiben és funkcionalitásában, úgy a másik legnépszerűbb „DE”, a Gnome, az Apple/MacIntosh komputerek felhasználói felületére hasonlít.

Nekünk Kockáknak nem kell desktop environment, „munkaasztali környezet”, elég egy egyszerű ablakkezelő. Ezekből sok van, mi most bekeményítettünk, s a talán legegyszerűbbet rakjuk fel. Ha ettől nem szaladsz világgá, már jó úton haladsz valóban a teljes „bekockásodás” felé!

Ez egy igazán minimalista és pici ablakkezelő. Látszik a telepítési lista itt közreadott elejéből, hogy a forráskódja is mindössze 29 kbyte!

```
pv@Csiszilla ~]Compile sithwm
Compile: Locating a recipe for sithwm ...
Compile: Found recipe for sithwm
GetRecipe: Trying to get http://gobolinux.org/recipe-store/SithWM--1.2.2-r1--
recipe.tar.bz2
GetRecipe: Downloading recipe from http://gobolinux.org/recipe-
store/SithWM--1.2.2-r1--recipe.tar.bz2
Compile: Recipe placed in /Files/Compile/Recipes/SithWM/1.2.2-r1
Compile: Checking dependencies...
Compile: Compiling SithWM version 1.2.2, revision r1.
--02:03:47-- http://sithwm.darkside.no/sn/sithwm-1.2.2.tgz
=> `sithwm-1.2.2.tgz'
Resolving sithwm.darkside.no... 195.0.152.146
Connecting to sithwm.darkside.no|195.0.152.146|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 29,826 (29K) [application/x-tar]

100%[=====>] 29,826      101.30K/s

02:03:48 (101.23 KB/s) - `sithwm-1.2.2.tgz' saved [29826/29826]

Compile: Unpacking file /Files/Compile/Archives/sithwm-1.2.2.tgz...
```

A lefordítása is készen van úgy egy percen belül.

Most azonban be kell állítani ennek is a konfigurációját, mert mint gyakorlatilag minden linuxos proginak, ennek is van olyan. Linux alatt a konfigurációs állományok neve általában úgy néz ki, hogy VALAMI.rc vagy VALAMIRC – a VALAMI helyére a progijét kell elképzelned, tisztelt Olvasóm! Ilyesmit kell tehát létrehoznunk a SithWM ablakkezelő számára is. És nyilván a saját könyvtárunkba. Na de mi legyen benne, s mi legyen a pontos neve e fájlknak?!

Szerencsére találunk egy példafájlt, a
vz@Csiszilla /Programs/SithWM/1.2.2/Shared/sithwm/default.sithwmrc
útvonal alatt.

Ezt gyorsan átmásoljuk saját HOME könyvtárunkba, még hozzá
.sithwmrc

névvel. Belenézünk, s látjuk, hogy ebben olyasmik vannak megadva, hogy
milyen gomb lenyomására mit csináljon, stb. Ezt egyelőre nem piszkáljuk –
hanem az elején nagyon érdekes dolgot látunk – egy példát arra, hogyan
definiálja a SithWM a menüket! Erről a részről van szó:

```
# menu "Apps"
#   menu "Viewers"
#     cmd "Eye of GNOME" "eog"
#     cmd "Xpdf" "xpdf"
#   end
# end
# menu "XShells"
#   cmd "Aterm" "aterm"
#   cmd "XTerm" "xterm"
# end
```

Persze mind kikommentezve. Nos, ezt a részt töröljük ki, és írjuk a helyére a
magunk által igényelt menüszerkezetet – persze kommentjeleket (#) ne írjunk
a soraink elé! Az enyém e pillanatban így néz ki:

```
menu "Internet"
  cmd "Firefox" "firefox"
  cmd "Azureus" "azureus"
  cmd "jDictionary" "jdictionary"
  cmd "NVU" "nvu"
  cmd "Thunderbird" "thunderbird"
  cmd "gFTP" "gftp"
  cmd "Pidgin" "pidgin"
  cmd "Konqueror" "konqueror"
end
menu "Terminálemulátorok"
  cmd "Mrxvt" "mrxvt -name FullScreen"
  cmd "XTerm" "xterm"
  cmd "Konsole" "konsole"
end
menu "Fájlkezelők"
  cmd "MC" "mc"
end
menu "OpenOffice"
  cmd "Writer" "swriter"
  cmd "Táblázatkezelő" "scal"
  cmd "Rajzoló" "sdraw"
  cmd "Adatbáziskezelő" "sbase"
  cmd "Bemutatókészítő" "simpres"
  cmd "Képletszerkesztő" "smath"
  cmd "Nyomtatókezelő" "spadmin"
end
```

```

menu "Grafika"
  menu "Nézegetők"
    cmd "Kview" "kview"
    cmd "Xpdf" "xpdf"
  end
  menu "Rajzolók"
    cmd "KolourPaint" "kolourpaint"
    cmd "Gimp" "gimp"
    cmd "Fontforge" "fontforge"
    cmd "Inkscape" "inkscape"
  end
  cmd "XaOS" "xaos"
  cmd "Stellarium" "stellarium"
  cmd "Celestia" "celestia"
end
menu "Rendszer"
  cmd "Kdiskfre" "kdf"
  cmd "Gkrellm" "gkrellm"
end
menu "Hang és videó"
  cmd "Audacious" "audacious"
  cmd "gMplayer" "gmpayer"
  cmd "Audacity" "audacity"
  cmd "K3B" "k3b"
end
menu "Játék"
  cmd "C-64" "x64"
end

```

A menü szintaxisa ebből a példából is jól látható: Az „end” szolgál egy menü végének jelzésére. A menük tetszőleges mélységben egymásba ágyazhatóak. A cmd-vel adjuk meg a menüparancsot. Az első paramétere a menüben megjelenő felirat, a második lényegében egy shell parancs, és paramétereket is tartalmazhat az idézőjelek közt, ha szükséges. Ügyeljünk rá, hogy az Mplayer progi esetében az Mplayernek ahhoz az indulásához, aholis látjuk a grafikus kezelőfelületet, ahhoz a „**gmpayer**” parancs tartozik!

No most ha megvan ez a file, hozzákészülhetünk az indításához. Ehhez a saját **.xinitrc** fájlunkat kell átírnunk, úgy, hogy ez legyen benne:

```

#!/bin/sh
until xsetroot -solid '#36444f';do sleep 1;done
xmodmap -e 'clear Lock'
xbindkeys
numlockx on
xset r rate 400 30
xset s 180
xrdb $HOME/.Xdefault
if xsetroot -display :0.1 -solid '#36444f'
then
  AUXDIS=:0.1
else
  AUXDIS=:0

```

```

fi
export AUXDIS
exec sithwm
exec xterm -display $AUXDIS -C
}

```

Amint látható, az xbindkeys progit továbbra is indítottuk, hogy kedvenc billentyűparancsaink a SithWM alatt is használhatóak legyenek. Valamint továbbra is bekapcsolt állapotban szeretjük a NumLock gombot...

Az **xset r rate 400 30** sor a billentyűzet ismétlési sebességét állítja be. Az első szám (itt a 400) a késleltetést jelzi milliszekundumokban, mielőtt tehát ismételni kezdene a billentyűzet a lenyomás után, a második szám azt mutatja meg, hány karaktert ismételjen egy másodperc alatt.

Mielőtt ezt a változtatást a **.xinitrc** állományon megejtenénk, nagy óvatosan készítsünk róla egy másolatot a könyvtárunkba, legyen a neve az, hogy **kde.xinitrc** – ez jó lesz nekünk, ha valamiért olyan perverzek leszünk, hogy mégis a KDE-t akarjuk indítani valamiért!

Tehát a **kde.xinitrc** tartalma ez kell legyen:

```

xbindkeys &
numlockx on &
[ -x /bin/startkde ] && exec startkde
[ -x /bin/wmaker ] && exec wmaker
[ -x /bin/twm ] && {
xsetroot -solid '#203040' -cursor_name 'left_ptr'
exec twm
}

```

Ezután írjuk csak át a **.xinitrc** fájlt az előbb mondottakra. S ha megvagyunk vele, erről is készítsünk egy másolatot, ezúttal **sithwm.xinitrc** néven.

És most csináljunk két állományt, egyiknek legyen a neve az, hogy **kdestart**, míg a másiké legyen mondjuk **sithstart** – sejtjük már, mi célt szolgálnak majd ezek?!

A **kdestart** file tartalma:

```

rm /Users/vz/.xinitrc
cp /Users/vz/kde.xinitrc /Users/vz/.xinitrc
startx

```

Azt hiszem világos, mit csinál: letörli a **.xinitrc** fájlunkat, és átmásolja a **kde.xinitrc** fájlunkat ennek helyére **.xinitrc** névvel. Ezután indítja a grafikus felületet. Ami a KDE lesz, mert a **.xinitrc** tartalma annak megfelelő.

A **sithstart** tartalma:

```
rm /Users/vz/.xinitrc
cp /Users/vz/sithwm.xinitrc /Users/vz/.xinitrc
startx
```

Ennek a feladata tökéletesen az, csak a sithwm.xinitrc fájlt állítja be a .xinitrc helyére.

Ezeket mozgassuk át (felesleges hogy a mi könyvtárunkban megmaradjanak) abba a könyvtárba, ahol saját szkriptjeinket gyűjtjük, tehát a

/Programs/Szkriptjeim/0.0.0/bin

nevűbe, majd adjunk mindegyikre futtatási jogot a

```
chmod 755 kdestart
```

és

```
chmod 755 sithstart
```

parancsokkal. Ezután (rendszergazdaként!) symlinkeljük be őket:

SymlinkProgram Szkriptjeim

Mostantól a grafikus felületet 3 féleképp indíthatjuk:

kdestart

Ez a KDE desktop environmentet indítja nekünk.

sithstart

Ez a SithWM ablakkezelőt indítja nekünk.

startx

Ez azt indítja a KDE és a SithWM közül, amelyiket legutóbb használtuk.

Érdekességgéppen leírom, hogy a SithWM indítható másképp is. Ekkor az **.xinitrc** állomány tartalma ez kell legyen:

```
#!/bin/sh
until xsetroot -solid '#36444f';do sleep 1;done
xmodmap -e 'clear Lock'
xbindkeys
numlockx on
xset r rate 400 30 xrdp $HOME/.Xdefault
if xsetroot -display :0.1 -solid '#36444f'
then
AUXDIS=:0.1
else
AUXDIS=:0
fi
export AUXDIS
exec sithwm
exec xterm -display $AUXDIS -C
```

És ezzel a szkripttel indíthatjuk az X-et, az xinit helyett:

```
#!/bin/sh
DISPLAY=:0
export DISPLAY
NO=0
while ls /tmp/.X${NO}-lock
do
let NO++
done
DISPLAY=${NO}
nohup X $DISPLAY -noreset -v -s 1 -ar1 300 -ar2 30 $XARGS &
nohup sh $HOME/.xinitrc &
sleep 20
```

Ez a szkript megpróbál egy szabad szerverszámot találni. Ekkor elindítja a szerveret ezzel a számmal, és futtatja a `.xinitrc`-t. Mellesleg e módszer működik akkor is, ha mégsem e szkripttel indítjuk a SithWM-et, hanem a `startx` paranccsal.

A fentiek fényében – ha lefordítjuk az XFCE ablakkezelőt a **Compile xfce** paranccsal – akár még azt is „behuzalozhatjuk” a rendszerünkbe hasonlóképpen. Azt már nem írom le részletesen, csak annyit szólok róla, hogy az ő esetében elvileg (ha nem tévedek...) ez kéne legyen a `.xinitrc` tartalma:

```
[ -x /bin/startxfce4 ] && exec startxfce4
[ -x /bin/wmaker ] && exec wmaker
[ -x /bin/twm ] && {
    xsetroot -solid '#203040' -cursor_name 'left_ptr'
    exec twm
}
```

A SithWM-ből legegyszerűbb kilépni a **Ctrl-Alt+Backspace** billentyűkombinációval. Ekkor mélykonzolba jutunk, abba tehát, ahonnét a grafikus felületet indítottuk.

A SithWM indulásakor egy abszolút üres, szürke képernyő fogad minket, középen egy kis `x` jellel, ami az egérkurzor. De semmire nem tudunk vele kattintani, mert egy szál ikon sincs a képernyőn. Továbbá, nemhogy ikon, de még szöveg sem. A menü (amit ugye mi állítottunk össze az első indulás előtt!) a Win+Enter lenyomásával érhető el. Benne a kurzorbillentyűkkel mozoghatunk, s az Enterrel választhatunk. Emellett működnek az összes olyan, xbindkeys-re kötött billentyűparancsaink is, melyeknél a módosítóbillentyű nem a Win gomb. (Ideje volna már olyan billentyűzeteket gyártani, ahol nem a Windows képe díszelge e gombokon, hanem a pingviné... vagy legalább legyenek ilyen öntapadós matricák...)

Hogyan lehet beállítani a SithWM-ben a nekünk tetsző billentyűzetismétlési sebességet?

A megoldás: végy fel egy

xset r rate 400 30

sort az .xinitrc fájlodba! Az első szám (itt a 400) a késleltetést jelzi milliszekundumokban, mielőtt tehát ismételni kezdene a billentyűzet a lenyomás után, a második szám azt mutatja meg, hány karaktert ismételjen egy másodperc alatt.

Egyéb hasznos billentyűparancsok:

A Sithwm az explicit és implicit fókusz kényelmes keverékével rendelkezik. Az épp aktuális ablakkeret fekete, az egyéb ablakoké baloldalt szürke.

Használhatod az egeret az ablakok manipulálásához a Win billentyűt lenyomva és kattintva a gombbal bárhol az ablakban. Az alapértelmezett funkciók a következők:

"1-es gomb"

Az ablak mozgatása.

"2-es vagy 3-as gomb"

Az ablak átméretezése.

A legtöbb billentyűparancs úgy történik, hogy le kell nyomni a Win gombot, és még egy billentyűt. Az elérhető funkciók:

Az aktuális ablak maximalizálása: **Win+A**

Az aktuális ablak törlése (bezárása): **Win+Esc**

Az ablak mozgatása balra, lefelé, fel vagy jobbra (16 pixellel): **Win+"S, X, E, D"**. A Shift lenyomásával együtt mozgatás helyett átméretezi az ablakot.

Az ablak mozgatása a képernyő balfelső, jobbfelső, balalsó vagy jobbalsó sarkába: **Win+"W, R, Z, C"**.

Infókat jelenít meg az aktuális ablakról – méret, stb: **Win+I**

Váltás sorra az ablakok közt: **Win+Tab**

Még számos – számomra kevésbé fontosnak tűnő – funkciója is van, melyeket megtalálhatsz a

/Programs/SithWM/1.2.2/man/man1/sithwm.1

útvonalon elérhető man állományában. (angolul...). E man-oldal html-ben is olvasható e link alatt: http://sithwm.darkside.no/man_sithwm.html

A SithWM honlapja: <http://sithwm.darkside.no/sithwm.html>

De szerintem ennyi tudás a kezeléséről épp elég a napi munkához. Az ablakkezelő arra van, hogy az ablakokat kezelje, és nem másra!

No és hogy miért használ valaki épp SithWM-et? Például, mert fel akar vágni a haverok előtt, hogy ő milyen „übergeek” és Kocka csávó. Különböző is, jó érzés azt tudni, hogy mi olyan különlegesek vagyunk, hogy olyan ablakkezelőt használunk, amit a világon csak mintegy 2 tucat személy! (Ennél ugyanis a SithWM fejlesztője szerint sem lehet sokkal több ezen WM használóinak a száma). Én most például csaknem 100%-ig biztosra vehetem, hogy én vagyok az egész Világon az *egyetlen* olyan személy, aki SithWM-et használ, és éppen konkrétan GoboLinux alatt.

No de eltekintve a hiúsági okoktól (amit viszont nem szégyelek mert mindenki hiú, legfeljebb pofátlanul letagadja...), van értelme is ennek, hogy valaki SithWM-et használ. A KDE például egy memóriazabáló szörnyeteg – a SithWM nem az. A KDE binárisai is tekintélyes helyet foglalnak el a merevlemezünkön – a SithWM csak 23908 bájtot. Azaz cseppet sem megabájtok tömkelegét! Mindez nem mellékes, ha például egy kis minimál-disztrót akarunk építeni, vagy ha szűkiben vagyunk a memóriának. De ha sok a memóriánk, akkor is előnyös, ha nem a KDE foglalja le annak felét-harmadát, hanem a hasznos, lényegi alkalmazások.

Személy szerint én főként két másik okot tartok legelőnyösebbnek. Egyik, hogy itt minden átváltható, s billentyűzetről vezérelhető, a másik pedig, hogy nincs teli kéretlen és zavaró szolgáltatásokkal.

Meg kell említenem a SithWM egy érdekes tulajdonságát. Ha több, egymást részben fedő ablakod van, hiába kattintasz rá az alsóra az egérrel, az bár aktívvá válik, de nem fogja eltakarni az őt (részben) fedőt! Engem ez eleinte zavart, de aztán rájöttem, hogy ez nagyon jó, mert így például az aktív ablakom egy nem fontos részére áthúzhatom mondjuk a GKrellM rendszermonitort, s látom, hogy amit csinálok a másik ablakon, az hogyan befolyásolja mondjuk a memóriahasználatot vagy a hálózat állapotát. Sőt, ez hozzásegít bennünket ahhoz, hogy a grafikus felületre egy órát mesterkedjünk.

Óra és dátum kijelzése a grafikus felületen

Ehhez szerencsére semmit nem kell telepítenünk. Ezt az **xclock** progi fogja elvégezni nekünk. Ezt egyszerűen indíthatjuk terminálemulátorból:

```
xclock -digital -update 1
```

E parancs azt mondja, hogy az **xclock** progit digitális módon indítsa (azaz e stílusú kijelzéssel), és a mutatott értékeket egy másodpercenként frissítse. Ennek hatására egy kis fehér grafikus ablakban efféle stílusban jelenik meg a dátum és idő:

2008 jún. 18., szerda, 14.10.19 CEST

Azt hiszem világos, mi mit jelent... de analóg, azaz „mutató” órát is elővarázsolhatunk, csak ahhoz így kell indítanunk:

```
xclock -analog -update 1
```

Különb en ez utóbbi az alapértelmezés, azaz:

ha nem adunk meg **-digital** vagy **-analog** paramétert, úgy az *analog* az érvényes.

A megjelenített kis ablakot aztán oda húzzuk a win billentyű lenyomva tartása mellett az egérrel, ahová akarjuk. S ezután ha mondjuk az OpenOffice-ban dolgozunk vagy Firefoxban vagy bárhol, a Win-Tab billentyűvel előcsalogatjuk az ablakát, amit elhúzzunk a képernyő egy nem használt részére, de közben a másik ablakon dolgozunk, például szöveget szerkesztünk, s hála a SithWM-nek az épp aktív ablak nem fogja eltakarni az xclock mutatta időt!

Különb en amiatt kerestem egy efféle jópofa kis órácskát, mert zavart, hogy az OpenOffice újabb változataiból kihagyták az alsó státuszbaron korábban ottlevő apró órát.

Amennyiben nem akarjuk az xclock-ot parancssorból indítani, huzalozzuk be a SithWM menüjébe, a következőképp:

```
cmd "Xclock" "xclock -digital -update 1"
```

Az ION3 ablakkezelő indítása

Ha valaki nem a SithWM hanem az Ion3 ablakkezelőre akar áttérni, az előzőleg írtak alapján már egyszerű. Mindenekelőtt persze le kell fordítsa azt, majd ha már telepítve van, ez kell legyen a **.xinitrc** tartalma:

```
#!/bin/sh
xbindkeys
numlockx on
ion3
```

Ezt tehát tegye be egy **ion.xinitrc** nevű file-ba a saját \$HOME könyvtárába (tehát a /Users/vz -be...), majd csináljon egy **ionstart** nevű szkriptet a **/Programs/Szkriptjeim/0.0.0/bin** könyvtárba, aminek ez a tartalma:

```
rm /Users/vz/.xinitrc
cp /Users/vz/ion.xinitrc /Users/vz/.xinitrc
startx
```

Persze symlinkelje is be a **Szkriptjeim** csomagot újra. A használata gondolom világos az eddigiek alapján...

A Ratpoison ablakkezelő

A Ratpoison (a név jelentése magyarul „patkányméreg”) ablakkezelő indítása esetén a két szkript tartalma:

```
vz@Csiszilla ~]cat ratpoison.xinitrc
#!/bin/sh
xbindkeys
numlockx on
ratpoison
```

illetve

```
vz@Csiszilla /Programs/Szkriptjeim/0.0.0/bin]cat ratstart
rm /Users/vz/.xinitrc
cp /Users/vz/ratpoison.xinitrc /Users/vz/.xinitrc
startx
```

Az edigiekből gondolom már világos a használatuk.

Megjegyzendő, hogy a Ratpoison az alapértelmezett (default) terminálemulátornak az **Xterm**-et használja. Amennyiben azt szeretnénk, hogy ez inkább az **Mrxvt** legyen, akkor célszerű már eleve ezzel az opcióval fordítani. Ehhez az kell, hogy fordítás előtt kissé belenyúljunk a recipéjébe! A recipe végéhez kell hozzáírni néhány sort. Íme a recipéje, félkövérrel kiemelve az általunk hozzáírandó sorokat:

```
# Recipe (MakeRecipe) for Ratpoison by Viola Zoli; Hungary,
violazoli@gmail.com, on 2008. márc. 20., csütörtök, 22.14.12 CET
# Recipe for version 1.4.3 by Viola Zoli; Hungary, violazoli@gmail.com, on
2008. márc. 20., csütörtök, 22.14.12 CET
compile_version=1.10.2
url="http://savannah.nongnu.org/download/ratpoison/ratpoison-1.4.3.tar.gz"
file_size=418364
file_md5=c1899e3e2549d29825cdb1b8d042c836
recipe_type=configure
configure_options=(
"--with-xterm=mrxvt"
)
```

Ami pedig magának a Ratpoisonnak a használatát illeti, annak ismertetése nem e könyv feladata, ellenben szép részletes – bár angol nyelvű – leírás található róla e link alatt:

<http://www.nongnu.org/ratpoison/doc/index.html>

Annyit jegyeznék csak meg, hogy a Ratpoison esetén az ablakkezelővel kapcsolatos összes funkció úgy érhető el, hogy nyomunk egy speciális billentyűkombinációt, majd még egy billentyűt. E speciális billentyűkombináció alapértelmezés szerint a Ctrl-t gomb. Például ahhoz hogy egy terminálemulátort indítsunk, le kell nyomni, hogy Ctrl-t majd ezután (miután tehát a „t” gombot felengedtük) egy **c** gombot.

Sokak számára túl bonyodalmas lehet ez, és esetleg szeretnénk kicserélni ezt a Ctrl-t prefix gombsorozatot mondjuk a baloldali Windows gombra. Ennél a linknél:

<http://ratpoison.antidesktop.net/cgi-bin/wiki/ChangeEscapeKeyHOWTO> azt olvastam, hogy kell csinálnom egy .ratpoisonrc file-ot a saját \$HOME könyvtáramba (/Users/vz) ezekkel a sorokkal:

```
exec xmodmap -e "keycode 115 = F30"
escape F30
```

és akkor állítólag oké lesz. Nos nem lett oké, semmi nem változott. De én rájöttem! A file-ot persze létre kell hozni valóban, de mindössze ennyi kell bele:

```
escape Super_L
```

A Metisse ablakkezelő

Ehhez e két szkript kell:

```
vz@Csiszilla /Programs/Szkriptjeim/0.0.0/bin]cat metissestart
rm /Users/vz/.xinitrc
cp /Programs/Szkriptjeim/0.0.0/xinitrc/metisse.xinitrc $HOME/.xinitrc
startx
```

és

```
vz@Csiszilla /Programs/Szkriptjeim/0.0.0/xinitrc]cat metisse.xinitrc
Xmetisse -ac -depth 24 -geometry 1024x768 :1 >& .Xmetisse-log &
metisse-start-fvwm >& .compositor-log &
mrxvt
killall Xmetisse
```

(Persze miután megírtuk e szkripteket s elhelyeztük őket ahová kell, ne feledjük el beszimlinkelni újra a **Szkriptjeim** csomagot...)

A -depth a színmélységet határozza meg, a -geometry meg természetesen azt, mit bír a monitorunk „felbontásilag”...

Saját billentyűzetkiosztás definiálása

Máris látom a tanácstalan tekintetedet, kedves Olvasóm: mi a csudának saját billentyűzethozzárendelés, ha GoboLinuxunk eddig remekül működött a magyar billentyűzet magyar kiosztásával?!

Nos, ennek számos oka lehet. Például, nem vagyunk megelégedve a betűk szokásos kiosztásával. Előfordulhat ugye, hogy te korábban olyan billentyűzeten írtál, ahol az „Y” és „Z” betűk meg voltak cserélve a mostani állapothoz képest. Vagy neked is az a véleményed mint nekem, hogy teljesen felesleges a „Q” gombot olyan jól elérhető helyre rakni, amikor az abszolút ritkán használatos karakter a magyar szövegekben, helyette jobb lenne valami gyakoribb betűt ahhoz a billentyűhöz rendelni! Vagy esetleg eszperantista vagy, és szeretnéd, ha amúgy magyar billentyűzeteden azért az eszperantó „kalapos” (azaz ékezetes) betűket is elérhetnéd valamiféle speciális billentyűkombinációk segítségével. Vagy netán thai/japán/héber (stb...) karakterekkel akarod bővíteni a billentyűzet karakter-repertoárját...

Nos, a GoboLinux alatt (persze más disztróknál is) erre van mód. Mindezekelőtt azonban tudni kell, hogy amit most leírok, csak a grafikus felületre vonatkozik – beleértve a virtuális termináljainkat, tehát kedvenc Mrxvt termináljainkat is. Amit most írunk, az tehát nem lesz érvényes „mélykonzolban”, abban tehát, amiből bejelentkezünk a grafikus felületre a „startx” paranccsal!

Szóval, saját billentyűzethozzárendelést legegyszerűbb úgy csinálni, hogy meghagyjuk e mostanit, ami valószínűleg jó nekünk nagyjából, mindössze megváltoztatunk benne néhány apróságot. Ehhez mindeneelőtt mentsük ki e mostanit egy fájlba, az

```
xmodmap -pke >myxmodmap
```

paranccsal. Ezután nézzünk ebbe bele. Nem listázom ide az egészet, de efféle sorokat látunk:

```
keycode 24 = q Q backslash Greek_OMEGA backslash Greek_OMEGA
```

Ez azt jelenti, hogy ha a billentyűzetünk egy „24” kódú jelet küld a számítógépnek, akkor az milyen karakterkódot állítson elő belőle. Mint látható a szám után több variáció is fel van sorolva. A legelső a shift nélküli variációt jelenti – ez itt a „q” betű. Utána jön az, ha a Shift módosító-billentyűvel nyomtuk meg a gombot, ez itt a „Q”-t eredményezi. Utána az AltGr-el való együttes megnyomás jele szerepel, na ez az a hülye „visszaperjel” vagy „fordítottperjel” amit a Windowsos korszakunkból jól ismerünk, s aminek angolul „backslash” a neve. Idáig világos a dolog, most jön az, ami a számomra homály: az ezutáni adat azt jelenti, hogy mi van, ha Shift+AltGr-el együtt nyomnánk meg a billentyűt, s még ez is oké lenne, de úgy tapaszt-

taltam, hogy az ezutáni oszlopban megint az jön, hogy mi van ha AltGr-el nyomjuk meg, s az ezutániban megint az, mi van ha Shift+AltGr-el együtt! Vagyis az AltGr-es és Shift+AltGr-es kombinációk kétszer szerepelnek!

Bevallom becsületesen, ennek nem tudom az okát. Mint azt már bizonyára sejti a Tisztelt Olvasóm, e fájlban módosítani fogjuk a bejegyzéseket. Annyi bizonyos, hogy nálam, az én gépemen kizárólag akkor működik az AltGr-es és Shift+AltGr-es kombináció, ha a megfelelő sorban mindkét helyen megadom az adatot!

Én mostanában azzal ismerkedem, ami még „linuxos berkekben” is sokak rémálma: a Vim nevű szövegszerkesztő. Emellett felraktam a Firefoxomhoz a Vimperator nevű kiegészítőt, mely által a böngészőmet a Vim-hez hasonló módon lehet kezelni. No most, a Vim esetében és a Vimperátornál is különböző okokból melyekre itt most nem térek ki, elég sokszor kell a kettőspont-jelet nyomkodni, s elhatároztam, hogy ezt a Q gomb helyére rakom, mert úgy gyorsan, kényelmesen elérhetem, ráadásul a Shift, tehát módosítóbillentyű lenyomása nélkül. Ehhez a fájlunkban a már idézett sort a következőképp módosítottam:

```
keycode 24 = colon Q backslash Greek_OMEGA backslash Greek_OMEGA
```

Azaz, a nagy Q továbbra is elérhető a Shift billentyű segítségével, de ha anélkül nyomjuk, a kettőspontot kapjuk eredményül. Azt sajnos a „colon” névvel kell megadnunk a fájlban.

Megemlítenéd, hogy Greek_OMEGA jelet én bizony nem bírok kicsikarni a billentyűzetből! Ha a virtuális terminálban nyomom a Shift+AltGr-Q gombot, semmit nem ír ki, ha az OpenOffice alatt nyomom, akkor ezt a szép 1/2-es jelet: ½

Gondolom ennek az lehet az oka, hogy amit itt megadunk, nem maga a karakter amit látunk, csupán egy karakterkód, s a megjelenített kép nem okvetlenül az, amit elvárunk – függ ugyanis a használt karakterkészlettől is! Szóval, a dolog bonyolult. El tudom képzelni, hogy elvileg a Shift+AltGr-Q kombináció ha lenyomom, igenis kiküldi a megfelelő jelet, de az a most használt fontkészletemben az ½ jelet eredményezi, miközben más fontkészlet alatt talán valóban a görög omegát látnám. A virtuális terminál alatt meg azért nem látok semmit ekkor, mert az az eredeti „mélykonzol” terminál beállításait emulálja, s abban nincs megadva görög omega jel, legalábbis nem a magyar karakterkészletben, ami eltér attól, amit az X felület alatt használunk!

Ki is próbáltam: OpenOffice-ban ha a most használt **HBookman** betűtípusról átváltok **Ljuba Serif**-re, megjelenik az omega-jel: Ω

Ha most ezeket olvasván szörnyülködve csóválod a fejedet, s azt mondd, ez iszonyú zűrös és nagy a kavarás, akkor – akkor igazat adok neked! Sajnos, jelenleg ilyen a számítástechnika állapota. Ezen a helyzeten próbál segíteni az

úgynevezett Unicode-kódolás, s annak UTF-8 nevű implementációja, aholis teljesen egyértelmű lesz, milyen kód milyen karakterhez tartozik, mert nem lesznek külön kódlapok csak egyetlenegy, igaz ott egy karakterhez olykor több bájtt is tartozik majd, mert a kód csak annyi bájton fér el. Mindenesetre, akkor a világ összes jele egyetlen, egyértelmű kódrendszer alapján lesz azonosítható. Ez a jövő útja, de jelen pillanatban még nem terjedt el kizárólagosan.

Szóval, a kettőspontot már rátettem a „q” jel helyére, de azért szeretném, ha amikor kell, nyomhatnék még „q” betűt is! El kell helyeznem őt is valahová. Nos, e célra azt találtam ki, hogy az AltGr+k-ra teszem a kis „q” betűt, az AltGr+L helyére pedig a nagy Q betűt! Az eredetileg e billentyűkön pompázó áthúzott „L” betűket úgysem használom semmire sem.

A módosítandó sorok a módosítás után így néznek ki:

```
keycode 45 = k K q ampersand q ampersand
keycode 46 = l L Q Lstroke Q Lstroke
```

Ezenkívül úgy döntöttem, megcserélem a nem-numerikus tasztatúrán levő „4” és a „!” jeleket, azaz hogy ezentúl a felkiáltójelet érhessem el a Shift lenyomása nélkül, mert az nekem sokkal többször kell (ugyanis regényíró vagyok), és különben is, számokat mindig a numerikus tasztatúrán (angolul „keypad”) szoktam bepötyögni. Hasonlóképp járok el a „7” és az „=” jelekkel is. Ehhez a következő sorok kellenek:

```
keycode 13 = exclam 4 dead_breve breve dead_breve breve
keycode 16 = equal 7 grave dead_grave grave dead_grave
```

Van aztán ez a sor:

```
keycode 91 = KP_Delete colon
Ezt kicseréltem erre:
keycode 91 = KP_Delete period
```

Ez azt csinálja, hogy a jobbszálen levő numerikus tasztatúrán lent van egy gomb, amire felül egy pont van rajzolva, alulra meg a „Del” szöveg van írva. No most engem halálra idegesített eddig, hogy hiába van oda pont rajzolva, az mégis vesszőt eredményezett, ha lenyomtam. Így ugyanis nem üthettem be tört számokat a billentyűzeten. Tudom, magyar szabvány szerint nem tizedespont van hanem tizedesvessző, de ez egyrészt hülye szabály szerintem, másrészt ha szabály is, rajtam nem sokat segít, ha a gépem igenis a tizedesPONTOT várja el a számításoknál törtszámok esetén! Nos a fenti módosítással végre igenis pontot ír nekem az a billentyű, amire pont van rajzolva.

Ahhoz hogy a fenti módosítások érvénybe lépjenek, ki kell adni az

```
xmodmap myxmodmap
```

parancsot. Ha azt akarjuk, hogy ez az új billentyűzetkiosztás már rögzítve a grafikus felület indulásakor érvénybe lépjen, ahhoz... nos, elvileg csak annyi kéne, hogy a „myxmodmap” nevű fájlkunkat átnevezzük úgy, hogy

.xmodmaprc

(Persze ezen fájl a mi \$HOME könyvtárunkban kell legyen, azaz a /Users/vz mappában).

Sajna, nekem ez az egyszerű módszer nem működik – de azért megoldjuk a dolgot! Módosítsuk a SithWM-et indító **sithwm.xinitrc** állományunkat úgy, hogy ez legyen a tartalma:

```
#!/bin/sh
until xsetroot -solid '#36444f';do sleep 1;done
xmodmap .xmodmaprc
xmodmap -e 'clear Lock'
xbindkeys
numlockx on
xset r rate 400 30
xset s 180
xrdb $HOME/.Xdefault
if xsetroot -display :0.1 -solid '#36444f'
then
  AUXDIS=:0.1
else
  AUXDIS=:0
fi
export AUXDIS
exec sithwm
exec xterm -display $AUXDIS -C
}
```

A fenti listában kiemeltem félkövérrel azt a sort, amivel ki kellett bővítsük, ahhoz, hogy betöltse a mi kiosztásunk fájlját. Igazán nem ördöngösség, ugyebár?!

Felmerülhet a kérdés, honnan találhatod ki a speciális karakterek neveit, melyeket meg kell adnunk a táblázatunkban! Nos ezeket itt leled fel:

/Programs/Xorg/7.2/include/X11/keysymdef.h

mindössze van a nevek előtt egy **XX_** rész is, amivel nem kell törődnöd.

Annak akit érdekel, az eszperantó betűk elnevezései, és hogy én hová raktam őket:

Egy eszperantó billentyűzetkiosztás

Az eszperantó betű	A betű neve	Billentyűzet-kombináció
ĉ	ccircumflex	AltGr C
Ĉ	Ccircumflex	Shift+AltGr C
ĝ	gcircumflex	AltGr T
Ĝ	Gcircumflex	Shift+AltGr T
ĥ	hcircumflex	AltGr H
Ĥ	Hcircumflex	Shift+AltGr H
ĵ	jcircumflex	AltGr J
Ĵ	Jcircumflex	Shift+AltGr J
ŝ	scircumflex	AltGr S
Ŝ	Scircumflex	Shift+AltGr S
ŭ	ubreve	AltGr Ü
Ŭ	Ubreve	Shift+AltGr Ü

A fenti kiosztáshoz szükséges sorok:

```

keycode 20 = udiaeresis Udiaeresis ubreve Ubreve ubreve Ubreve
keycode 28 = t T gcircumflex Gcircumflex gcircumflex Gcircumflex
keycode 39 = s S scircumflex Scircumflex scircumflex Scircumflex
keycode 43 = h H hcircumflex Hcircumflex hcircumflex Hcircumflex
keycode 44 = j J jcircumflex Jcircumflex jcircumflex Jcircumflex
keycode 54 = c C ccircumflex Ccircumflex ccircumflex Ccircumflex

```

Hogy miért nem a G betűhöz rendeltem hozzá az eszperantó „kalapos G”-t? Nos, ennek oka, hogy nem óhajtok lemondani a „csukószögleteszárójelről” (erről ni: „|”) ami oda van rajzolva, s hagyományosan elérhető az AltGr+G kombinációval. Az nekem kell, mert elég sokat programozom. Hasonlóképp nem akartam lemondani az eurójelről sem, emiatt tettem a „kalapos-U” jelet az Ü-höz és nem az U-hoz.

Mindemellett belevettem egy érdekes sort is a fájlba, ez az:

keycode 78 = Scroll_Lock ampersand

Ez eredetileg csak ennyi volt:

keycode 78 = Scroll_Lock

Így elértem, hogy a Shift+ScrollLock billentyűvel elő tudom csalogatni a „&” jelet, ami most eltűnt szokásos helyéről, az AltGr+C helyről, az odatett „kalapos-C” betű miatt.

Azt hiszem most már semmi akadálya annak, hogy a kedves Olvasó a neki tetsző billentyűhozzárendelést valósítsa meg!

Csináljunk extra módosítóbillentyűt!

E könyvem előző változatának megírása óta súlyos szerencsétlenség szakadt rám: tönkrement a billentyűzetem! A nejem vett nekem egy újabbat, de persze a legolcsóbbat, amin nincsenek multimédiás billentyűk... ki kellett találnom valamit helyettük. Persze az xbindkeys progival hozzárendelhetem más billentyűkombinációkhoz a kedvenc funkcióimat – de *miféle* billentyűkombinációkhoz?! A „Win” billentyűt nem használhatom, azt lefoglalja a SithWM ablakkezelő, hasonlóképp a CapsLock-ot is. A Ctrl illetve Alt gombokra meg jobb ha nem számítok, mert tuti, hogy valami progiban, például az OpenOffice-ban (vagy akár a Firefoxban, pláne ha Vimperator kiegészítést használok hozzá...) e billentyűknek fontos egyéb szerepeket szánnak! Kell készíteni egy saját, extra módosítóbillentyűt!

E feladatra a „0” gombot szemeltem ki a nem-numerikus tasztatúrán, azon tehát, ahol ő balra fent helyezkedik el. Szépen, kényelmesen elérhető ugyebár... és nincs is rá igazán szükség, tisztességes Kocka a számokat a numerikus tasztatúrán pötyögi be jobboldalt, „magyarul” a keypad-on...

A feladat nem volt egyszerű. Egyszerre kellett összepasszintani 3 állományt nálam: a \$HOME/.xmodmaprc, a \$HOME/.xinitrc és a \$HOME/.xbindkeysrc állományokat! Persze a dolog rém egyszerű, ha már tudjuk, mit kell tenni – csak ugye amíg rájöttem...

De most már megvan.

Most ilyen az **xmodmap** parancsom kimenete:

```
vz@Csizsilla ~]$xmodmap
xmodmap: up to 3 keys per modifier, (keycodes in parentheses):

shift      Shift_L (0x32),  Shift_R (0x3e)
lock
```

```
control      Control_L (0x25), Control_R (0x6d)
mod1         Alt_L (0x40), Alt_L (0x7d), Meta_L (0x9c)
mod2         Num_Lock (0x4d)
mod3         Hyper_R (0x31)
mod4         Super_L (0x7f), Hyper_L (0x80)
mod5         Mode_switch (0x5d), ISO_Level3_Shift (0x71), ISO_Level3_Shift
(0x7c)
```

Ebből a mod3 kezdetű sor úgy jött létre, hogy a `$HOME/.xmodmaprc` file-ben megkerestem azt a sort, ami így néz ki eredetileg:

```
keycode 49 = 0 section notsign notsign notsign notsign
```

s ezt átírtam erre:

```
keycode 49 = Hyper_R section notsign notsign notsign notsign
```

Most már lett tehát Hyper_R billentyűm, de meg kellett mondani a masinának, hogy ez egy „mod3”-nak tekintett módosítóbillentyű ezentúl! Emiatt most így néz ki a `$HOME/.xinitrc` fájlom:

```
#!/bin/sh
until xsetroot -solid '#36444f';do sleep 1;done
xmodmap .xmodmaprc
xmodmap -e 'clear Lock'
xmodmap -e 'add Mod3 = Hyper_R'
xbindkeys
numlockx on
xset r rate 400 30
xset s 180
xrdb $HOME/.Xdefault
if xsetroot -display :0.1 -solid '#36444f'
then
  AUXDIS=:0.1
  else
    AUXDIS=:0
  fi
  export AUXDIS
  exec sithwm
  exec xterm -display $AUXDIS -C
}
```

Ennek persze az eleje az érdekes, az

```
xmodmap -e 'add Mod3 = Hyper_R'
```

sor, amit a fenti listában félkövérrel ki is emeltem, a többi a SithWM ablakkezelő betöltéséhez kell.

Ezek után lehet parancsokat hozzárendelnem a billentyűkombinációkhoz a `$HOME/.xbindkeysrc` fájlban, az új gombomra mint „mod3” hivatkozva (a

módosítók nevei e fájlban szerencsére nem case-sensitivek):

```
"mrxvt"  
Mod3 + a  
  
# Play/Pause ( 0+Tab)  
  
"mocp -G"  
Mod3 + c:23  
  
# Previous (0+Q)  
  
"mocp -r"  
Mod3 + colon  
  
# Next (0+I)  
  
"mocp -f"  
Mod3 + c:10
```

Megjegyzésre érdemes, hogy a „Previous” azaz az előző szám lejátszásánál miért mint „colon” hivatkozom a „q” gombra. Amiatt, mert én ugyan a „Q” jelű gombot nyomom, de a `$HOME/.xmodmaprc` egy másik helyén már korábban a kettőspontot definiáltam rá e billentyűre.

Most már boldog vagyok. Van egy saját, abszolút külön módosítóbillentyűm, amire egyáltalán semmi progí nem számít, jól elérhető helyen, s azt kötök rá, amit csak akarok!

Hogyan készíts szerveret az otthoni számítógépedből?

Ennek természetesen számos módja van. Itt most egy aránylag könnyebb módszert ismertetünk majd. E módszer amiatt „könnyebb”, mert egy szerverhez többnyire a következő progikat kéne külön-külön telepíteni:

- HTTPD (becenevén Apache)
- MySQL
- PHP
- egyéb apróságok, például SSL

Mi mindettől megkíméljük magunkat. Letöltünk egy olyan bináris csomagot, amiben ez már mind egyben megvan, integrálva. Ennek neve: XAMPP

Előrebocsátom azt is, e fejezet eleve feltételezi azt, hogy fix IP címed van! Gondolom, az itt leírt módszerek többsége dinamikus IP cím esetén is használható, de bizonyára némi módosítással, s ezt nem kísérleteztem ki.

Tehát, először is töltsük le a XAMPP progit, tehát ezt:

<http://kent.dl.sourceforge.net/sourceforge/xampp/xampp-linux-1.6.6.tar.gz>

```
vz@Csiszilla ~]wget http://kent.dl.sourceforge.net/sourceforge/xampp/xampp-  
linux-1.6.6.tar.gz
```

```
--16:07:05-- http://kent.dl.sourceforge.net/sourceforge/xampp/xampp-  
linux-1.6.6.tar.gz
```

```
=> `xampp-linux-1.6.6.tar.gz'
```

```
IP keresés kent.dl.sourceforge.net... 212.219.56.167
```

```
Connecting to kent.dl.sourceforge.net|212.219.56.167|:80... kapcsolódva.
```

```
HTTP kérés elküldve, várom a választ... 200 OK
```

```
Hossz: 60.711.704 (58M) [application/x-tar]
```

```
100%[=====] 60.711.704 135.51K/s ETA  
00:00
```

```
16:14:27 (134.03 KB/s) - `xampp-linux-1.6.6.tar.gz' saved [60711704/60711704]
```

Ezután nézzünk bele a letöltött tömörített csomagba! Látunk benne egy **lampp** nevű könyvtárat, benne számos programösszetevővel. Ezt kell valahogy a rendszerünkbe integrálni. Nos, ez nem lesz triviális feladat, tekintve hogy GoboLinuxunk alá eddig csak forrásból telepítettünk, vagy ha mégis binárisokat, akkor olyan bináris csomagokat, melyeket valaki kifejezetten a GoboLinux számára készített. Ez a progí pedig nem olyan!

Továbbá, eszünk ágában sincs elrontani GoboLinuxunk szép, áttekinthető könyvtárstruktúráját, aholis a **/Programs** átnézésével azonnal láthatjuk, milyen összetevők vannak a rendszerünkben. Ezekből készítsünk egy **Xampp** nevű könyvtárat a **/Programs** alá, ebbe pedig egy verziókönyvtárat **1.6.6** néven. Ide telepítjük a cuccost, s majd megküzdünk érte, hogy innen is működjön. Ide tehát másoljunk át mindent, ami csak akad a tömörített csomag **lampp** könyvtárában.

Ezután elblattyogunk a <http://www.apachefriends.org/en/xampp-linux.html> linkre, ahol (angolul) leírják, hogyan kell ezt a programcsodát telepíteni! Nos, azt látjuk, hogy nekünk a kitömörített állományt a **/opt** könyvtárba kéne kitömöríteni, ezzel a paranccsal:

```
tar xvfz xampp-linux-1.6.6.tar.gz -C /opt
```

No de nekünk nincs **/opt** könyvtárunk, s különben is, már kitömörítettük egészen máshová. Holott mert ez nem forrásprogram, semmiképp sem módosíthatjuk úgy, hogy ne a **/opt**-ban próbálja megkeresni önmaga részeit!

Sebaj! Készítsünk egy **/opt** nevű könyvtárat:

```
mkdir /opt
```

lépjünk oda be, majd itt hozzunk létre **lampp** néven egy szimbolikus linket, mely a **/Programs/Xampp/1.6.6** könyvtárra mutat! Ezek után, ha a progit indítottuk, az bármit keresne is a **/opt/lampp** könyvtárban, e könyvtár helyett a symlinket találja majd meg, ami olyan, mint amikor a postás visz egy levelet egy házhoz, de az zárva, s a kapun ez a tábla áll: „Elköltöztem. Új címem:...” - ekkor a postás továbbküldi a levelet az új címre. Eképp a symlink most a másik, igazi könyvtárra küldi a program utasításait, oda, ahol ténylegesen szerepelnek a programösszetevők.

A baj csak az, hogy most elrontottuk a GoboLinux szép, tiszta gyökér-könyvtárrendszerét. Ott az a csúnya **/opt**. No de erre találták ki a **GoboHide** programot:

```
pv@Csiszilla [/]gobohide -h /opt
```

és máris nem látjuk a **/opt** -ot! De attól még működni fog...

A telepítési leírást olvasgatva látjuk, hogy nekünk a **lampp** paranccsal kell indítanunk ezt a szerver-tcsodát. Na de erről sajnos csak akkor venne tudomást a rendszerünk, ha épp abban a könyvtárban állunk, ahol e **lampp** progí van! Be kéne symlinkelnünk; ám ha nem akarunk kézzel felvenni symlinkeket a **/System/Links/Executables** könyvtárba, ami rém rossz ötlet mert inkonzisztenssé teheti a rendszerünket, akkor másképp kell ezt megoldanunk. Beütjük a teljes elérési utat, legfeljebb ha gyakran szerverkedünk, majd csinálunk rá egy alias-t: azaz így indítjuk:

```
/opt/lampp/lampp start
```

Leállítás:

```
/opt/lampp/lampp stop
```

A sima

```
/opt/lampp/lampp
```

parancsra némi tájékoztatást közöl az indítási lehetőségekről:

pv@Csiszilla /Programs/Xampp/1.6.6]/opt/lampp/lampp
Usage: /System/Links/Executables/lampp <action>

start	Start XAMPP (Apache, MySQL and eventually others)
startapache	Start only Apache
startssl	Start only SSL support
startmysql	Start only MySQL
startftp	Start only ProFTPD
stop	Stop XAMPP (Apache, MySQL and eventually others)
stopapache	Stop only Apache
stopssl	Stop only SSL support
stopmysql	Stop only MySQL
stopftp	Stop only ProFTPD
reload	Reload XAMPP (Apache, MySQL and eventually others)
reloadapache	Reload only Apache
reloadmysql	Reload only MySQL
reloadftp	Reload only ProFTPD
restart	Stop and start XAMPP
security	Check XAMPP's security
php5	Activate PHP5
php4	Activate PHP4
phpstatus	Which version of PHP is active?
backup	Make backup file of your XAMPP config, log and data files
panel	Starts graphical XAMPP control panel

FIGYELEM! ATTÓL A PILLANATTÓL FOGVA, HOGY ELINDÍTOTTAD A LAMPP PROGRAMOT, A RENDSZERED FOKOZOTT VESZÉLYNEK, BIZTONSÁGI KOCKÁZATNAK VAN KITÉVE, ÉS ÉRZÉKENY A KÜLSŐ TÁMADÁSOKRA!

Minél gyorsabban be kell tömnünk legalább a legnyilvánvalóbb biztonsági réseket, azaz konfigurálnunk kell a rendszerünket. Ehhez a böngészőnkben keressük fel a

<http://localhost>

linket. Mármint ezt ajánlja a honlapon levő tájékoztató. Nekem érdekes módon ez abszolút nem működik, csak az a jó, ha a számítógép nevét ütöm be:

csiszilla

Erre a böngészőben megjelenik egy oldal, melyen különböző nyelvek közül válogathatok. Ugyanez lesz az eredménye annak is, ha az IP címed számaidat ütöd be a böngésződ címsorába.

Nos, a nyelvek közül még az angol a legalkalmasabb nekem, mert a magyar nincs köztük. Ezt választván bejön ez a link:

<http://csiszilla/xampp/index.php>

Kattintsunk a Security fülre! Egy rakás biztonsági figyelmeztetés fogad. Kezdjük őket orvosolni! Ehhez adjuk ki (rendszergazdaként) a következő parancsot: `/opt/lampp/lampp security`
Ez a következő párbeszédet eredményezi:

```
pv@Csiszilla /]/opt/lampp/lampp security
XAMPP: Quick security check...
XAMPP: Your XAMPP pages are NOT secured by a password.
XAMPP: Do you want to set a password? [yes]
XAMPP: Password:
XAMPP: Password (again):
XAMPP: Password protection active. Please use 'lampp' as user name!
XAMPP: MySQL is accessible via network.
XAMPP: Normally that's not recommended. Do you want me to turn it off? [yes]
XAMPP: Turned off.
XAMPP: Stopping MySQL...
XAMPP: Starting MySQL...
XAMPP: The MySQL/phpMyAdmin user pma has no password set!!!
XAMPP: Do you want to set a password? [yes]
XAMPP: Password:
XAMPP: Password (again):
XAMPP: Setting new MySQL pma password.
XAMPP: Setting phpMyAdmin's pma password to the new one.
XAMPP: MySQL has no root password set!!!
XAMPP: Do you want to set a password? [yes]
XAMPP: Write the password somewhere down to make sure you won't forget it!!!
XAMPP: Password:
XAMPP: Password (again):
XAMPP: Setting new MySQL root password.
XAMPP: Change phpMyAdmin's authentication method.
XAMPP: The FTP password is still set to 'lampp'.
XAMPP: Do you want to change the password? [yes]
XAMPP: Password:
XAMPP: Password (again):
XAMPP: Reload ProFTPD...
XAMPP: Done.
```

Ha ezután kattintunk a „Security” fülre, megjelenik egy ablak. Itt a felhasználói névnek azt kell beütni: **lampp**, jelszónak meg természetesen azt, amit beállítottunk.

Ez nagyon szép és rém komolyan hangzik, DE HOGYAN LESZ EBBŐL SAJÁT HONLAPUNK?!

Egyszerű. Tegyük fel, hogy nem is akármilyen közönséges honlapot óhajtunk készíteni, hanem egyenesen egy dokuwiki szoftvert! (Ez előnyös, mert ez olyan wiki-fajta, amihez nem kell adatbázis-kezelő). Nos, más dolgunk sincs, csak annyi, hogy a

`/Programs/Xampp/1.6.6/htdocs`

könyvtárba készítsünk egy dokuwiki alkönyvtárat (lehet más is a neve, ami

nekünk tetszik), ebbe felmásoljuk a szoftver megfelelő állományait, majd ha készen van, örülhetünk, mert a

<http://csiszilla/dokuwiki/doku.php>

link böngészőbe írására működni kell a szoftverünknek! A „csiszilla” helyére persze a te géped neve kell kerüljön. De kerülhet ide az IP címed is, például:

<http://83.145.173.156/dokuwiki>

E módszerek elvileg akkor is kell működjenek, ha ki van húzva az internet-kábeled a számítógépből. Ekkor persze csak te láthatod a saját honlapodat a gépen, más nem.

DokuWiki helyett persze akármi más is lehet a könyvtárban, akár önmagunk által összekínlódott html-oldal is. De „rendes” wiki-szoftvert is telepíthetünk, ehhez persze le kell töltsük azt a

<http://download.wikimedia.org/mediawiki/1.11/mediawiki-1.11.1.tar.gz>

linkről.

Ezt csomagold ki a neked tetsző könyvtárba (amit előbb létre kell hoznod, ott, ahol a fenti példában szereplő dokuwiki is volt, legyen a neve mondjuk **wikike**), és utána tedd írhatóvá a config alkönyvtárat:

```
chmod a+w config
```

Ezután keresd fel a

<http://csiszilla/wikike>

linket, és telepítsd a wikidet! (Ehhez előbb létre kell hoznod egy adatbázist, ezt lásd e fejezetben pár bekezdéssel később).

Néhány megjegyzés: Ha a phpMyAdmin földre kattintasz, felhasználót és jelszót kér majd tőled. No most hiába hogy nálunk a **pv** a root jogú felhasználó, ha ezzel és az ő jelszával akarsz belépni, nem enged be. Ha **pv**-ként akarsz belépni, de jelszó nélkül, érdekes módon beenged, de nem lesz jogod létrehozni adatbázist. A megoldás: azt írd be felhasználónévnek, hogy **root** és jelszónak azt add meg, amit a security beállításoknál mint **MySQL root password**-ot beállítottál!

Ezután hozd létre az adatbázisodat.

Ha ezután a MySQL konzolt használni akarod (már ha értesz hozzá...) az úgy megy, hogy menj el a **/Programs/Xampp/1.6.6/bin** könyvtárba, és menj be a MySQL konzolba így:

```
pv@Csiszilla /Programs/Xampp/1.6.6/bin]mysql -u -p
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 66
Server version: 5.0.51a Source distribution
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql>
```


Ha netán van már egy máshonnan lementett adatbázisod, tegyük fel az a neve, hogy „birodalom”, azt így importálhatod:

```
pv@Csiszilla /Programs/Xampp/1.6.6/bin]mysql -u root -p < /Users/vz/birodalom.sql
Enter password:
```

Add meg a MySQL Root Password-öt, és várj türelmesen, amíg importálja.

Ezután a phpmyadmin főoldalán *privilegiumok* menüpontból létrehozol egy user-t akinek van joga használni az előbb létrehozott adatbázist. Majd ezzel a mysql usernek az adataival feltelepíted/beconfigolod a MediaWikit, vagy ami szoftvert épp telepítesz. Ügyelj azonban rá, hogy ezen user létrehozásakor a „Belépési információ” menüpont alatt a „Hoszt” sorban az legyen, hogy „Helyi” kiválasztva, erre a mellette levő ablakocskában meg kell jelenjen a „localhost” felirat. Különben nem fogadja majd el a telepítendő szoftverből az user adatait a MySQL motor.

A MediaWiki esetén telepítéskor megadandó adatoknál ügyeljünk rá, hogy pontosan adjuk meg őket! Jó telepítés esetén a **LocalSettings.php** legfontosabb részlete így kell kinézzen:

```
$wgDBtype           = "mysql";
$wgDBserver         = "localhost";
$wgDBname           = "Ez az adatbázisod neve";
$wgDBuser           = "ide jön a MySql user neve, akit létrehoztunk a
phpMyAdminban";
$wgDBpassword       = "Ide jön az user jelszava";
```

THTTPD

E fejezet megírásához felhasználtam adatokat az e link alatti cikkből is:
<http://www.hwsz.hu/oldal.php3?cikkid=657>

Ez a fejezet a THTTPD „mini webszerver démon”-ról szól. Azaz, szintén arról, hogyan lehet saját honlapod a saját gépeden. Ez a kis okosság, a THTTPD azonban csak statikus honlapokat fog neked kezelni. Nem tud annyit, mint az előző fejezetben bemutatott változat, de jóval egyszerűbb kezelni is. A THTTPD weboldala: <http://www.acme.com/software/thttpd/>

Telepítése

0. Szerkeszd a `/System/Settings/group` file-ot, írd bele a következő sort:

```
www:x:200:
```

1. Add ki a következő parancsokat rotként:

```
/usr/sbin/useradd -s /bin/false thttpd  
useradd -g bin bin
```

2. Compile thttpd

(Erre letölti a recipét és lefordítja). A fordítás előtt esetleg érdemes átszerkeszteni a forrásfa gyökerében lévő `config.h` fájlt, amelyben számos később már nem módosítható beállítást végezhetünk el. Ha tervezzük CGI programok futtatását, akkor a `CGI_TIMELIMIT` értékét módosíthatjuk az alapértelmezett 30 másodpercről 60-ra, esetleg még többre, ugyanis a 30 másodpercből könnyen kifuthatunk komplex CGI szkriptekkel megoldott feladatok esetén. Ugyanitt állítható be azoknak a fájlneveknek a mintája, amelyeket a szerver egy könyvtárra való hivatkozás során megpróbál kiszolgálni (`INDEX_NAMES`, értéke tipikusan „`index.html`”, „`index.htm`”).

A további opciókról vagy magában a fájlban, vagy a thttpd weboldalán olvashatunk, de általában az alapértékek megfelelőek. A totális kontrollra vágyó felhasználók számtalan egyéb opciót találnak `config.h`-ban, akiket pedig a szerver alapértelmezett hibaüzeneteinek bordó háttere, vagy a könyvtárlisták zöld háttere zavarja, azok a `libhttpd.c` fájlban végezhetik el a szükséges módosításokat.

3. `mkdir /Programs/THTTPD/Settings`
4. `ln -s /Programs/THTTPD/Settings /usr/local/thttpd`
5. `mkdir /Programs/THTTPD/Settings/etc`
6. `mkdir /var/log/thttpd`

7. `chown tthttpd /var/log/tthttpd`

8. Ezután ha nem lenne meg a `/Programs/THTTPD/Settings/etc/tthttpd.conf` file, akkor létre kell hoznod, ha megvan, akkor pedig szerkesztened kell, hogy így nézzen ki:

```
port=80
dir=/Mount/HDAx/www
nochroot
user=tthttpd
logfile=/var/log/tthttpd/access.log
pidfile=/var/run/tthttpd.pid
cgipat=/cgi-bin/*
```

Természetesen a `dir=/Mount/HDAx/www` sor helyére annak a könyvtárnak a pontos címét kell írnod, ami a web számára hozzáférhetővé akarsz tenni.

10. Indítsd el a `tthttpd` programot a következő paranccsal:

```
tthttpd -C /Programs/THTTPD/Settings/etc/tthttpd.conf
```

és sok szerencsét...!

Fontos megjegyezni, hogy a THTTPD érzékeny a config file szerkezetére! Üres sorok nem lehetnek benne, és egy sor csak egy opciót vagy megjegyzést tartalmazhat. A fájlt nekünk kell a semmiből létrehozunk (`touch tthttpd.conf`), majd feltöltenünk a kívánt értékekkel. A `port` segítségével azt a TCP portot adhatjuk meg, amelyre ráül a szerver, ez a legtöbb esetben a 80-as lesz. A `dir` a web felé mutatott könyvtárszerkezet gyökere, minden fájl és könyvtár, amit ebben a könyvtárban helyezünk el, látható lesz a böngészőprogramokból (persze csak akkor, ha jól be vannak állítva az attribútumok). A `nochroot` hatására a program nem `chroot`-ol a `dir`-ben megadott könyvtárba az indulása után, ezt csak akkor kapcsoljuk be, ha olyan CGI programokat szeretnénk futtatni, amelyeknek a `www` könyvtáron kívüli dolgokra is szüksége van (pld. perl interpreter). Ha nem kívánunk CGI szkripteket futtatni, akkor javasolt a konfigurációs fájl `chroot` opciójának használata. A `logfile` a webszerver tevékenységéről vezetett naplófájl helyét adja meg, ebben a fájlban csak a HTTP-kéréseket, illetve az azokra adott válaszokat naplózza a rendszer a szokásos formátumban, magának a `tthttpd` démonnak az üzenetei a `syslogon` keresztül a `/var/log/messages` fájlba kerülnek. A `pidfile` önmagában nem túl hasznos, a `tthttpd` processzazonosítóját fogjuk megtalálni az itt megadott fájlban, erre csak akkor lehet szükségünk, ha a disztribúciónk initszkriptjeihez illeszkedő indítószkriptet is gyártunk a programhoz. A `cgipat` pedig egy általános CGI minta; azok a fájlok, amelyekre illik a `cgipat` és megfelelően be vannak állítva a jogosultságaik, CGI programokként futtathatóak lesznek. Ezek a legfontosabb opciók, a program természetesen ezernyi apró lehetőséget tartalmaz, ezekkel most nem foglalkozunk, inkább próbáljuk meg elindítani.

Ha a böngészőnket a <http://localhost/> címre (vagy nálunk: <http://Csiszilla>) irányítjuk, és megjelenik a **www** könyvtárunk tartalma, nyert ügyünk van. Amennyiben valami mégsem sikerült volna, megnézhetjük az üzeneteket a rendszernaplóban, a **top** vagy a **ps** paranccsal pedig megvizsgálhatjuk, hogy fut-e **thttpd** néven valami.

Most kell a fix IP cím! Mert ezután ha eddig eljutottunk az van, hogy minden alkalommal amikor újraindítjuk az átitógépünket, root-ként ki kell adnunk a démon indító fenti parancsot, ezt tehát:

```
thttpd -C /Programs/THHTTPD/Settings/etc/thttpd.conf
```

s ezután MÁSOK, a haverjaink, akiknek megadjuk az IP címünket, így hívhatják meg a `/usr/local/thttpd/www` könyvtárba betett `index.html` állományunkat:

<http://101.102.103.104/>

feltéve hogy az IP címünk 101.102.103.104 (ez egy teljesen kitalált cím a példa kedvéért). Vigyázzunk azonban rá, hogy ha a fenti linket beírjuk próbának (jó IP címünkkel) a démon elindítása NÉLKÜL, akkor is megjelenik a böngészőben az `index.html`, mert a belső hálózaton át kommunikál önmagával a gép, s emiatt azt hihetjük, hogy fut a szerver! De nem fut. Próbának a teljes elérési utat, a

<http://101.102.103.104/index.html> -t kell beütni, na ekkor nem jelenik csak meg, ha nincs a szerverdémon elindítva!)

A `thttpd` apró mérete ellenére is sok lehetőséggel rendelkezik. A HTTP protokoll 1.1-es verzióján kívül ismeri a CGI 1.1-et, az SSI-t (Server Side Includes), támogatja az IPv6-ot, jelszavas oldalakat készíthetünk vele (erre való a csomagban lévő `htpasswd` program), de talán a legnagyobb tulajdonsága, amellyel más webserverek nem rendelkeznek, az a forgalomszabályozás (szelep)². E lehetőség segítségével URL-ekre és URL-csoportokra definiálhatunk egy byte(!)/másodpercben megadott értékeket, amit a szerver semmilyen körülmények között nem fog átlépni. A funkció használatához fel kell vennünk a konfigurációs állományba a

`throttles=/usr/local/thttpd/etc/thttpd.throttles` opciót, és létre kell hoznunk a `/usr/local/thttpd/etc/thttpd.throttles` fájlt. A forgalomszabályozási fájl szintaxisát szintén a `thttpd` kézikönyvdala írja le, a fájl szerkezete nem túl bonyolult, először azt az URL-mintát kell megadnunk, amelyre a korlátozás vonatkozik (pl. a `"**"` minden fájlra fog vonatkozni, míg a `"**.*jpg|**.*gif|**.*png"` csak a képfájlokra), utána pedig az adott mintához társított byte/másodperc értéket kell definiálnunk. Ne szerénykedjünk, a korlátozásokat akár fájlonként is megadhatjuk, így lehetnek olyan állományaink, amelyeket nagyobb sebességen kínálunk, míg másokhoz akár vérlázítóan alacsony értéket is megadhatunk. Az alábbi minta a webserverünk összes fájljához 6000

² Megjegyzem, a forgalomszabályozást, annak itt közölt módját én NEM PRÓBÁLTAM KI!

byte/szekundumban határozza meg a hozzáférést:

```
# /usr/local/thttpd/etc/thttpd.throttles ** 6000
```

A szerver egyébként részben e tulajdonságáról kapta a nevét, a thttpd t-je tetszőlegesen behelyettesíthető a tiny, turbo és throttling szavakkal. A szoftver nagy site-okon ritkán szokott felbukkanni.

Használata

A program elindítása után egyszerűen elérhető a megfelelő könyvtárba helyezett weblapod (index.html állománya) a következőt beírva a böngésző címsorába:

<http://xxx.xxx.xxx.xxx/index.html>

feltéve, hogy fix IP-d van, és az xxx-ek helyébe a saját IP címedet írod.

Recipe új verziójának elkészítése

Észrevehettük eddig, hogy programok telepítéséhez többnyire úgynevezett *recipe*-ket használtunk. Nos, előfordulhat, hogy telepíteni akarunk egy programot, aminek azonban csak régebbi változatához van recipe a GoboLinux recipetárolójában. Nem kell kétségbeesni, többnyire a régi recipe is használható, csak új verziót kell készíteni belőle, melyben mindenképp meg kell adni az újabb verziójú program forráskódjának helyét. Persze ez is automatikus.

Ezt természetesen a

NewVersion

parancs végzi nekünk. Legegyszerűbb azonban, ha úgy fogunk hozzá, mintha új recipét készítenénk:

Példaként nézzük a Links programot. Nekem most a rendszerben a 2.1pre26 verziójú van meg, de felfedeztem, hogy van már egy újabb változata. Ha nekem az kell, mindenekelőtt letiltom a régit:

DisableProgram Links

Ezután:

```
pv@Csiszilla ~]MakeRecipe http://links.twibright.com/download/links-2.1pre33.tar.gz
MakeRecipe: There is already a recipe for Links, at
MakeRecipe: /Files/Compile/Recipes/Links/2.1pre26-r1
MakeRecipe: Use 'NewVersion Links 2.1pre33' instead of MakeRecipe? [Y/n]
y
NewVersion: Locating a recipe for Links...
NewVersion: Found recipe for Links at /Files/Compile/Recipes/Links/2.1pre26-r1
GetRecipe: Trying to get /Files/Compile/Recipes/Links/2.1pre26-r1
NewVersion: Recipe placed in /Files/Compile/Recipes/Links/2.1pre26-r1
NewVersion: Creating recipe for Links 2.1pre33 based on 2.1pre26-r1.
NewVersion: Checking URL 'http://links.twibright.com/download/links-2.1pre33.tar.gz'...
NewVersion: Downloading source code...
--01:26:51-- http://links.twibright.com/download/links-2.1pre33.tar.gz
=> `links-2.1pre33.tar.gz'
Resolving links.twibright.com... 81.0.239.25
Connecting to links.twibright.com|81.0.239.25|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4,170,585 (4.0M) [application/x-gzip]

100%[=====] 4,170,585 135.54K/s ETA 00:00

01:27:24 (128.67 KB/s) - `links-2.1pre33.tar.gz' saved [4170585/4170585]

NewVersion: Recipe template for Links 2.1pre33 created
NewVersion: Printing result (/Files/Compile/LocalRecipes/Links/2.1pre33/Recipe):
# Recipe for version 2.1pre33 by Viola Zoli; Hungary, violazoli@gmail.com, on 2008. febr.
21., csütörtök, 01.26.51 CET
# Recipe (MakeRecipe) for Links by Lucas Correia Villa Real <lucasvr@gobolinux.org>, on Tue
Nov 23 18:09:18 BRST 2004
compile_version=1.8.0
url="http://links.twibright.com/download/links-2.1pre33.tar.gz"
file_size=4170585
file_md5=8c4bc1ebbf4962d290df8218bfc21413
recipe_type=configure
```

```

pre_build() {
    # the shipped configure doesn't recognize parameters such as CC
    echo "Running autoconf..." && autoconf
}
NewVersion: Printing recipe for i686 architecture:
configure_options=(
    "--without-libfl"
    "--without-directfb"
    "--enable-javascript"
    "--enable-graphics"
    "--with-ssl"
    "--with-x"
)
NewVersion: Done
pv@Csiszilla ~]

```

Ezután fordítsuk le a

Compile Links

parancssal! A végén, ha minden sikerült, kiírja:

```
PackRecipe: Packed /Files/Compile/PackedRecipes/Links--2.1pre33--recipe.tar.bz2.
```

Ez azt jelzi, hogy ide rakta el ilyen néven a becsomagolt recipét. Ezt küldjük el a gobolinux-recipes@lists.gobolinux.org címre, örülni fognak neki!

Azt hiszem ez annyira világos, hogy nem is kell többet írnom róla.

Magyar man-oldalak

Ahhoz, hogy szeretett anyanyelvünkön is olvasgathassuk a man-oldalakat (már amelyiknek megvan a fordítása...) telepítenünk kell valahová a magyar man-oldalakat. Ehhez éppenséggel belenyúlhatnánk a `/System/Links/Manuals` könyvtárba, de én idegenkedem ettől a módszertől. Nem helyes, ha „kézi vezérléssel” igazgatjuk a rendszerkönyvtárakat, ha nem nagyon muszáj! S most nem muszáj nagyon. Még kicsit sem.

Ehelyett: Mindenekelőtt töltsük le az UhuLinux disztribúciótól a ftp://ftp.uhulinux.hu/sources/man-pages-hu/man_hu_2001_01_05.tar.gz csomagot, majd a tartalmát másoljuk ki egy nekünk tetsző könyvtárba. Én e célra a `/Programs/Szkriptjeim/0.0.0` könyvtárban hoztam létre egy „human” nevű alkönyvtárat.

A többi már pofonegyszerű. A saját `.zshrc` fájlunkba felvettem két alias, e megfeleltetésekkel:

```
alias human="man -M /Programs/Szkriptjeim/Current/human"  
alias Human="man -M /Programs/Szkriptjeim/Current/human"
```

Amint újraindítottam a terminálemulátort, gyönyörűen működött minden... Ja, hogy miként? Nos, úgy, hogy a közönséges „man” parancs továbbra is az angol man-oldalakat hozza be – természetesen. Magyar man oldalt így lehet hívni:

```
human bc
```

Ez behozza az „bc” parancs magyar man oldalát. A „bc”, az egy konzolos számológép. Látható, hogy hívhatom a `human`-parancsot kis- és nagy kezdőbetűvel is.

Sajnos magyar man oldalakból kevés van e csomagban, de semmi nem akadályozza meg ezek után, hogy ha valahol rábukkanunk egy nekünk kedves man-oldalra, azt egyszerűen bemásoljuk a mi human-könyvtárunkba, s utána már használhatjuk is...

Youtube-os videók letöltése

Gyakori kérdés az internetes fórumokon, hogyan lehetne letölteni a YouTube-ról videókat. Tudniillik „normális esetben” azokat csak nézni lehet. Eleve direkt úgy csinálták meg a YouTube-ot, hogy csupán nézegetni lehessen ezeket, nem „hazavinni”!

Na de gondold bele kérlek, Olvasóm, ha már nézed a SAJÁT monitorodon a videót, amit a SAJÁT géped játszik le neked – akkor csak meg kell legyen valahol a masinádban a videóállomány! Nehogy már meg ne találád!

A GoboLinux alatt az efféléket a rendszer a `/System/Variable/tmp` könyvtárba tölti le. A neve mindig úgy kezdődik, hogy **Flash** és e név után van mindenféle egyéb karakter. Teendőd tehát csak annyi, hogy kattintasz a YouTube oldalán a videóra, az elkezd lejátszani magát. Te figyeled, mikor ér véget a videofo-lyam letöltését jelző csík haladása (ha befejezte a videó lejátszását, biztos véget ért...), s ekkor, de mindenesetre **mielőtt** bezárnád a böngésző ezen ablakát, elmész e könyvtárba, s ezen állományt átmásolod a neked tetsző könyvtárba valami neked tetsző névvel, úgy, hogy a kiterjesztése **.flv** legyen. S már is megvan a videó neked!

Persze, minthogy mi szeretünk mindent parancssorból intézni, megadatik nekünk, hogy egyszerűbben intézzük ezt el. Mert amit fenn leírtam, működik ugyan, de írtó kényelmetlen. Gondoljuk csak el: várni, amíg letölti a videót! Megkeresni valami idióta néven a tmp könyvtár halomnyi szemetében! Átmásolni! Brrr! Elképesztő, hogy ez mekkora barbár nyomor!

Szerencsére van erre egy jó kis szkript. E szkriptet nem én írtam, hanem az e link alatti fórumtpikba:

<http://www.linuxempire.hu/viewtopic.php?f=11&t=108>

küldte be egy „ferx” néven regisztrált felhasználó. De olyan jó, hogy bemásolom ide. Minthogy ő maga tette közzé, s nem írta hogy copyrights volna, ez úgy vélem nem törvénytelen. Lám, ilyen a Linux világa: csak úgy odalökik nekünk ingyér' a szuper dolgokat!

A szkript neve legyen mondjuk **ywget**. Az ywget szkript tartalma:

```
WGET=/usr/bin/wget
GETFILE="http://www.youtube.com/get_video.php"
SED=/bin/sed

if [ -z "$1" ]; then
    echo "$0 <url> <nev>"
    exit 1;
elif [ -z "$2" ]; then
    echo "$0 <url> <nev>"
    exit 1;
fi;

PARMS=$(WGET -O - "$1" | grep "fullscreen" |\
$SED -e 's/.*\watch_fullscreen/')

$WGET -O $2.flv "$GETFILE$PARMS"
```

E szkriptet mentsük el a **/Programs/Szkriptjeim/0.0.0/bin** könyvtárba, tegyük futtathatóvá, symlinkeljük be, és használjuk! Használata: paraméterként megadjuk a letöltendő videó linkjét és a nevet (kiterjesztés nélkül) amin le akarjuk menteni. Ügyeljünk rá, hogy a youtube-os linkekben szokott lenni kérdőjel-karakter is, emiatt e paramétert idézőjelek közé téve kell megadnunk! Ha anélkül írjuk be a paramétert, a következő példán látható hibaüzenetet kapjuk:

```
vz@Csiszilla /Depot[ywget http://www.youtube.com/watch?v=ceEEaL3kwNM sharbat1
zsh: no matches found: http://www.youtube.com/watch?v=ceEEaL3kwNM
```

Egy példa, ahol a legendás izraeli énekesnő (kedvencem...) „Ofra Haza” egyik híres klipjét töltöm le:

```
vz@Csiszilla /Depot/Letolt[ywget "http://www.youtube.com/watch?v=L8Co7Iz0yhw" Yerushalaim
--17:03:06-- http://www.youtube.com/watch?v=L8Co7Iz0yhw
=> '-'
IP keresés www.youtube.com... 208.65.153.253, 208.65.153.238, 208.65.153.251
Connecting to www.youtube.com|208.65.153.253|:80... kapcsolódva.
HTTP kérés elküldve, várom a választ... 200 OK
Hossz: 113.272 (111K) [text/html]

100%[=====>] 113.272          94.53K/s

17:03:20 (94.25 KB/s) - '-' saved [113272/113272]

--17:03:20-- http://www.youtube.com/get_video.php?
fs=1&video_id=L8Co7Iz0yhw&l=326&sk=_eMFtTz3DZ8L-paer08tDQC&t=0EgsToPdskJkX5WwSV_pUo7n8-
xfDnd&hl=en&plid=AARG_bt0SN-JwQHfAAAAoAAcIAE&title=Ofra%20Haza%20-%20Yerushalaim%20Shel
%20Zahav';
=> `Yerushalaim.flv'
IP keresés www.youtube.com... 208.65.153.253, 208.65.153.238, 208.65.153.251
Connecting to www.youtube.com|208.65.153.253|:80... kapcsolódva.
HTTP kérés elküldve, várom a választ... 303 See Other
Hely: http://chi-v54.chi.youtube.com/get_video?
video_id=L8Co7Iz0yhw&signature=1554B2B199C5C61187ED2B841B53884B29018D28.1E4F901B744C5FDA11E
6DB8510E8CD9DCBEFE118&ip=82.144.172.155&ipbits=16&expire=1203977107&key=1 [következik]
--17:03:36-- http://chi-v54.chi.youtube.com/get_video?
video_id=L8Co7Iz0yhw&signature=1554B2B199C5C61187ED2B841B53884B29018D28.1E4F901B744C5FDA11E
6DB8510E8CD9DCBEFE118&ip=82.144.172.155&ipbits=16&expire=1203977107&key=1
=> `Yerushalaim.flv'
IP keresés chi-v54.chi.youtube.com... 64.15.116.149
Connecting to chi-v54.chi.youtube.com|64.15.116.149|:80... kapcsolódva.
HTTP kérés elküldve, várom a választ... 200 OK
Hossz: 11.754.304 (11M) [video/flv]

100%[=====>] 11.754.304    127.28K/s    ETA 00:00

17:05:14 (118.26 KB/s) - `Yerushalaim.flv' saved [11754304/11754304]
```

A fenti trükk példa volt rá, hogy Linux alatt és pláne parancssorban minden sokkal könnyebb. Kényelmesebb. Szinte előkelő...

Néhány hasznos konzolos okosság

Kedves Olvasó, mire eljutottál e fejezetig, már sok mindent „átéltél” e könyvben. Talán felmerült benned a kérdés, mi a jó fittyfenéért foglalkozom én ennyit a konzollal/virtuális_terminállal, amikor jól látod, hogy – legalábbis a KDE menüjében – csilivili progik *tömkelege* vár rád, amikkel kényelmesen, néhány egérgattintással megoldhatsz mindent, akár fontos rendszerbeállításokat is! Minek fektetek ennyi energiát akkor abba, hogy okvetlenül „konzolban” oldjak meg valamit?!

Nos, azért, mert aki nem tud valamit konzolban megoldani, az egyszerűen NEM ISMERI A RENDSZERÉT. Az NEM ISMERI A DISZTRÓT, AMIT HASZNÁL. Sőt, ÚGY ÁLTALÁBAN A LINUXOT SEM ISMERI. E könyvnek pedig elméletileg az lenne a célja, hogy a kezdő linuxosokat kissé megismertesse a Linuxszal. Most őszintén, volna-e értelme azt magyarázgatnom, a KDE menüjében melyik progik hol található? Egykettőre megtalálod ott a progikat magad is. Ha e könyvben olyasmit írnék le, amit 3 perc alatt magad is meglesz könyv nélkül, akkor ha netán valamikor személyesen találkozol velem, fejemhez vágod a könyvet, mondván, hogy feleslegesen adtál ki pénzt rá!

Különben is: senki nem tiltja meg neked, hogy a csilivili progikat használd. Jó azonban ha tudod, ha ezekhez szoksz hozzá, abban a pillanatban óriási bajba kerülsz, amint egy olyan Linux disztró elé ülsz le, amin ezek a progik nem lesznek fenn, vagy netán, urambocsá', nem is lesz grafikus felülete a disztrónak! És ilyen igenis RENGETEG van, tudniillik majdnem mindegyik „Linux” ilyen, amit például szerver célra alkalmaznak! A rendszergazdák ugyanis abból indulnak ki, hogy szervergépen csak az a szoftverösszetevő legyen, ami múlhatatlanul szükséges. Minthogy a grafikus felület nem ilyen, azt nem szokták feltenni, mert plusz potenciális biztonsági résnek tartják. Szerintem ebben teljesen igazuk van!

Emellett, a grafikus felületű programok sokkal több erőforrást igényelnek, mint a konzolosak. Adott esetben ez sem lehet mindegy. Egy grafikus felület nélküli disztró egy ősrégi gépen is „elketyeg”. (Itt jegyzem meg, hogy a GoboLinux minimális hardware-igénye Pentium-II számítógép, 128 megabájt RAM-mal. Ezen már OpenOffice-ot is tudunk futtatni, bár lassan).

A „parancssornak”, a „konzolnak” tehát igenis *vannak* előnyei. Például az is, hogy a legtöbb linuxos progik rendelkezik egy egész rakás olyan szolgáltatással, opcióval, amit egyszerűen nem is tudsz elérni grafikus felület alól, még akkor sem, ha csináltak neki valami grafikus kezelőfelületet, azaz, szakszóval, GUI-t. (ez a *Graphical User Interface* rövidítése, ha nem tévedek). Tudniillik a GUI-ba csak a progik leggyakoribb funkcióinak a vezérlését „huzalozták be”, főleg mert a legtöbbször képtelenség minden létező funkcióra valami jelölőnégyzetet alkotni, annyi mindenfélét tud egy-egy linuxos progik. S így aki nem képes elszakadni a GUI-tól, az a rendszerén elérhető progiknak csak a leg-alapabb funkcióit használhatja! Hatalmas pocséklás!

A partíciók foglaltságának kijelzése

Gyakorta nincs is szükség egy progira, ha ismerjük a „parancssort”. Nem szándékozom itt most a *bash* vagy a *zsh* shell sok-sok parancsát ismertetni, különben én is csak ezek egy töredékét ismerem. Akad viszont néhány, amit nagyon is hasznos megtanulnunk. Például ha meg akarjuk tudni, a partícióinkon mennyi szabad hely van, indíthatjuk a nagyon látványos, csicsás **KDiskFree** programot. Természetesen grafikus program. Lehet azonban, hogy nem szándékozunk erre memóriát pazarolni. Különben is, idő, amíg betöltődik. Mennyivel egyszerűbb ehelyett ezt beütni a terminálba:

df

Az eredmény efféle lesz:

```
vz@Csiszilla /Mount/HDA4/Users/vz]df
Fájlrendszer      1K-blokk   Foglalt    Szabad Fo.% Csatl. pont
/System/Kernel/Devices/hda6
                    18948024   9818120    9129904  52% /
tmpfs              254916     92         254824   1% /System/Kernel/Devices
none               254916     0          254916   0% /System/Kernel/Devices/
shm
/System/Kernel/Devices/hda1
                    10016136   9663088    353048   97% /Mount/HDA1
/System/Kernel/Devices/hda4
                    9405724    4084348    5321376  44% /Mount/HDA4
/System/Kernel/Devices/hda5
                    1028060    831952     196108   81% /Mount/HDA5
/System/Kernel/Devices/hda7
                    33791604   31421072   2370532  93% /Mount/HDA7
/System/Kernel/Devices/hda8
                    3895604    3808132     87472   98% /Mount/HDA8
/System/Kernel/Devices/hda9
                    1028060    717012     311048   70% /Mount/HDA9
```

Némi töprengés után világos előttünk minden: 1K-blokkonként értelmezendők az értékek, azaz kilobájtokban kapjuk az eredményt, s azt is kijelzi nekünk, mi a partíció csatolási pontja, azaz hová van felmountolva. A foglaltságot százalékban is megadja. Kell ennél több?! Minek ide valami grafikus izémizé?!

Ha csak az aktuális partíció foglaltságára vagyunk kíváncsiak, használhatjuk az „**l**” parancsot, ami GoboLinux-specialitás, tulajdonképpen az **ls** parancs előtétprogija. Ez kilistázza az aktuális könyvtár tartalmát, szépen, színesben, s a végére még előzékenyen kiírja azt is, hogyan állunk „területileg”. Íme egy példa:

```
vz@Csiszilla /Mount/HDA7/OfraHazaVideoCatalog]l
```

```
30/06 23:08 rwxr-xr-x      800 Album
03/07 20:14 rwxr-xr-x      304 Concerts and live Performances
05/07 13:13 rwxr-xr-x      496 Miscellaneous
24/06 02:52 rwxr-xr-x      256 New Videos
23/06 11:11 rwxr-xr-x      232 Soundtracks
23/06 11:09 rwxr-xr-x      120 Video Collections
25/03 22:38 rwxr-xr-x     3352 flv
28/11 09:06 rwxr-xr-x       736 kepek
09/03 00:00 rw-rw-r--    23923 OfraSongs.csv
26/04 00:00 rw-rw-r--    32102 indexOfra.txt
26/04 00:00 rw-rw-r--    36130 inventoryOfra.txt
```

```
=====
98451 in 11 files - reiserfs: 30850068 kB used (91%), 2941536 kB free
```

Bár itt nem látszik, de a könyvtárak – amiknek rwxr-xr-x az attribútumuk – kék színben jelennek meg, továbbá, a méretek kijelzésénél a gigabájtokat jelző helyiértékek a számban fehér színnel vannak írva, a megabájtoké türkizkékkel, a kilobájtoké halványkékkel, s csak a többi szürkével. Azaz így sokkal szemléletesebben kapjuk meg az eredményt. Természetesen az „used” előtti szám jelzi kilobájtban a foglalt helyet, és a „free” előtti a szabad helyet. A foglaltságot még százalékban is kiírja. Minek ide grafikus felület? Kell több?! Nem mond ez eleget?!

Konzolos hexa editor

Vagy ott van a KDE hexa editora. Minek egy hexa editornak grafikus felület?! Azért, hogy ha épp a grafikus felület megy tönkre valami esetleges programhiba miatt, a hexa editort se tudjuk használni?! Töltsük le ehelyett a picike kis „Heme” csomagot és telepítsük! Letöltési linkje:

<http://birodalom.net/linux/gobocsomagok/Heme--0.4.2--i686.tar.bz2>

Leírást itt találunk hozzá: <http://linux.die.net/man/1/heme>

vagy nézzük meg a man lapját. Annyit elárulok, hogy csinálhatunk egy

.hemerc

nevű config fájlt neki a mi Home tartalomjegyzékünkbe, ha egyéni beállításokat akarunk. Nekem például nem tetszett, hogy olyan ronda kékek a színei, emiatt az én **.hemerc** fájlom tartalma most ilyen:

```
fg_color = green
bg_color = black
```

Így most fekete alapon zöld betűket látok. Tiszta „Mátrixos” érzés...

Persze, a legtöbben nem akarnak hexa editort. Nem annyira „Kockák”. De talán nincs kifogásuk egy jó konzolos számológép ellen. Hiszen milyen jó is az, hétköznapi feladatokra is!

Konzolos számológépek

Örömmel közlöm, van ilyen a GoboLinuxban, ott kell legyen a te rendszeredben is, a neve: **bc** és sokan nagyon kedvelik. Egyszerűen gépeld be a bc parancsot, amire kapsz egy speciális promptot (aminek semmiféle jelzése sincs, csak a villogó kurzor...), és itt nyugodtan üsd be a kiszámítandó képletet, majd enter. E promptból a **quit** paranccsal léphetsz ki.

Rengeteget tud a bc, részletes magyar nyelvű leírást itt találsz róla:

<http://www.linuxcertif.com/man/1/bc/hu/>

Mindazonáltal mert a Linuxra jellemző, hogy egy feladatra többnyire nagyon sok progi létezik, emiatt rögvest bemutatok még két konzolos számológépet! Az egyik az, aminek egyszerűen csak „**E**” a neve. Ezt töltsd le innen:

<http://birodalom.net/linux/gobocsomagok/E--0.02718--i686.tar.bz2>

A

man e

parancsra kiírja a progi funkcióit neked a gép. Ez nem ad speciális promptot, mint a bc, hanem a program neve után (szóközzel elválasztva...) kell írd paraméterként a kiszámolandó képletet. Viszont mert a műveleti jelek nagy része illetve a zárójelek speciális karakternek számítanak a parancssorban, emiatt a paramétert – a képletet – idézőjelek közé kell rakd. Továbbá, az egyszerű **e** parancsra nálam (s biztos nálad is...) az **mc** szövegszerkesztője indul, egy üres dokumentumot megjelenítve. Hasznos ez is... de emiatt kénytelen voltam megcsinálni, hogy az **e** proginak kreálok egy **ó** álnevet. Javasolom ezt neked is... A progit tehát a következőképp használd:

```
vz@Csiszilla ~]ó "3*6"  
18  
vz@Csiszilla ~]ó "tan(45)"  
1.61977519054386
```

Amint látható a példán, a szögfüggvényeket alapértelmezésben radiánban számolja.

Az **E** progi nagy előnye, hogy picit – mindössze 7000 bájt...

Egy másik nagyon jó efféle progi a **wcalc**. Ezt telepítsd forrásból a **Compile wcalc**

paranccsal – nálam a 2.3.1 verző van fenn. Ez is indulás után speciális promptot ad. A **\r** paranccsal lehet váltani a radián és fok értelmezés közt. Itt egy rövid példafuttatás:

```

vz@Csiszilla /Programs/Wcalc]wcalc
Enter an expression to evaluate, q to quit, or ? for help:
-> tan(45)
    = 1
-> \r
Now Using Radians
-> tan(45)
    = 1.61978
-> \r
Now Using Degrees
-> tan(45)
    = 1
-> quit
vz@Csiszilla /Programs/Wcalc]

```

FTP kapcsolat intézése parancssorból

Tudtad-e, hogy FTP kapcsolat létesítéséhez sem szorulsz rá bonyolult grafikus programokra?! Az FTP kapcsolat képessége bele van építve az **mc**-be. Amikor fut, egyszerűen gépeld be az alsó sorba:

```
cd /#ftp:felhasználónév:jelszó@internetesím
```

(és utána =>Enter, vagyis nyomj Entert...)

A „cd” természetesen a könyvtárváltó utasítás, a többi a paraméter. Egy konkrét (bár képzeletbeli...) példa: ha be akarok jelentkezni a <http://boszorka.info> honlap tárhelyére, **devil** felhasználói névvel, s a jelszavam mondjuk **LuciFeri666** – akkor azt e paranccsal érhetem el:

```
cd /#ftp:devil:LuciFeri666@boszorka.info
```

Azt hiszem, ezt a témát sem érdemes tovább ragozni, mert világos...

Aki torrentezni akar, azt is megteheti parancssorból. Csak telepíteni kell a CTorrent progit. De van sok más hasonló is. Parancssorból tehát majdnem minden megoldható.

A Conky rendszermonitor

Bár a GKrellM remek rendszermonitor, de mintha nehézkes volna a beállítása, aztán meg túlságosan grafikusan néz ki! Túl sok a függősége is. Szerencsére azonban találtak nekem a barátaim egy megfelelőbb rendszermonitort. Ennek telepítése a

Compile conky

paranccsal lehetséges. Én a multimédiás billentyűzetem „Sleep” feliratú gombjára kötöttem rá, úgy, hogy ezt írtam a **.xbindkeysrc** fájlomba:

```
#rendszermonitor
"conky"
m:0x0 + c:223
```

Ahhoz hogy működjön, kell neki a „home” könyvtárunkba egy **.conkyrc** fájl. Ez határozza meg a kinézetét (sőt, főleg azt). Efféle fájlokat képpel illusztrálva bőven lelhetünk a <http://conky.sourceforge.net/screenshots.html> oldalon – én itt most közlöm az általam választottat, aminek a kiírandó szövegeit már magyarítottam is. Aztán, bele is nyúltam kissé az eredetibe itt-ott.

```
# Conky sample configuration
#
# the list of variables has been removed from this file in favour
# of keeping the documentation more maintainable.
# Check http://conky.sf.net for an up-to-date-list.

# set to yes if you want Conky to be forked in the background
background no

# X font when Xft is disabled, you can pick one with program xfontsel
#font 5x7
#font 6x10
#font 7x13
font 8x13
#font 9x15
#font *mintsmild.se*
#font -*-*-*-34-***-***-***

# Use Xft?
use_xft no

# Xft font when Xft is enabled
xftfont Bitstream Vera Sans Mono:size=10

# Text alpha when using Xft
xftalpha 0.8

# Print everything to stdout?
# out_to_console no

# MPD host/port
# mpd_host localhost
# mpd_port 6600
# mpd_password tinker_bell

# Print everything to console?
# out_to_console no

# mail spool
#mail_spool $MAIL

# Update interval in seconds
update_interval 5.0

# This is the number of times Conky will update before quitting.
# Set to zero to run forever.
total_run_times 0
```



```

# Create own window instead of using desktop (required in nautilus)
own_window yes

# If own_window is yes, you may use type normal, desktop or override
own_window_type normal

# Use pseudo transparency with own_window?
own_window_transparent no

# If own_window_transparent is set to no, you can set the background colour here
# own_window_colour hotpink
own_window_colour black

# If own_window is yes, these window manager hints may be used
#own_window_hints undecorated,below,sticky,skip_taskbar,skip_pager

# Use double buffering (reduces flicker, may not work for everyone)
double_buffer yes

# Minimum size of text area
minimum_size 280 5

# Draw shades?
draw_shades yes

# Draw outlines?
draw_outline no

# Draw borders around text
draw_borders yes

# Draw borders around graphs
draw_graph_borders yes

# Stippled borders?
stippled_borders 8

# border margins
border_margin 4

# border width
border_width 1

# Default colors and also border colors
default_color green
default_shade_color black
default_outline_color black

# Text alignment, other possible values are commented
#alignment top_left
#alignment top_right
alignment bottom_left
#alignment bottom_right
#alignment none

# Gap between borders of screen and text
# same thing as passing -x at command line
gap_x 12
gap_y 12

# Subtract file system buffers from used memory?
no_buffers yes

```

```

# set to yes if you want all text to be in uppercase
uppercase no

# number of cpu samples to average
# set to 1 to disable averaging
cpu_avg_samples 2

# number of net samples to average
# set to 1 to disable averaging
net_avg_samples 2

# Force UTF8? note that UTF8 support required XFT
override_utf8_locale no

# Add spaces to keep things from moving about? This only affects certain objects.
use_spacer no

# mldonkey_hostname      Hostname for mldonkey stuff, defaults to localhost
# mldonkey_port          Mldonkey port, 4001 default
# mldonkey_login         Mldonkey login, default none
# mldonkey_password      Mldonkey password, default none

# boinc (seti) dir
# seti_dir /opt/seti

# Allow for the creation of at least this number of port monitors (if 0 or not set, default
is 16)
#min_port_monitors 16

# Allow each port monitor to track at least this many connections (if 0 or not set, default
is 256)
#min_port_monitor_connections 256

# none, xmms, bmp, audacious, infopipe (default is none)
#xmms_player none

# variable is given either in format $variable or in ${variable}. Latter
# allows characters right after the variable and must be used in network
# stuff because of an argument

# stuff after 'TEXT' will be formatted on screen

TEXT
$nodename - $sysname $kernel on $machine
$stippled_hr
${color lightgrey}Uptime:$color $uptime ${color lightgrey}- Load:$color $loadavg ${color
#ffffee}- CPU hőfok:$color ${acpitemp}C

${color lightgrey}CPU használat:${color #cc2222} $cpu% ${cpubar}
${color red}${cpugraph 0000ff 00ff00}
${color lightgrey}RAM használat:$color $mem/$memmax - $memperc% ${membar}
${color lightgrey}Swap használat:$color $swap/$swapmax - $swapperpc% ${swapbar}
${color lightgrey}Processzek:$color $processes ${color grey}Running:$color
$running_processes
$color$stippled_hr
${color lightgrey}${offset 260}Hálózat:
${color #ffffee} Memória ${offset 100}Le:${color lightgreen} ${downspeed eth0} k/s${color
lightgrey} ${offset 80}Fel:${color #22ccff} ${upspeed eth0} k/s
${color #ffffee}${memgraph 32,150 222222 ffffff} ${color #0000ff}${downspeedgraph eth0
32,150 33dd00 ddbb00} ${color #22ccff}${upspeedgraph eth0 32,150 3300ff ff0000}

${color lightgrey}Fájrendszer:
```

```

/ $color${fs_used}/${fs_size}/${fs_bar}/
/Mount/HDA4 $color${fs_used}/Mount/HDA4}/${fs_size}/Mount/HDA4} ${fs_bar}/Mount/HDA4}
/Mount/HDA5 $color${fs_used}/Mount/HDA5}/${fs_size}/Mount/HDA5} ${fs_bar}/Mount/HDA5}
/Mount/HDA7 $color${fs_used}/Mount/HDA7}/${fs_size}/Mount/HDA7} ${fs_bar}/Mount/HDA7}
/Mount/HDA8 $color${fs_used}/Mount/HDA8}/${fs_size}/Mount/HDA8} ${fs_bar}/Mount/HDA8}
/Mount/HDA9 $color${fs_used}/Mount/HDA9}/${fs_size}/Mount/HDA9} ${fs_bar}/Mount/HDA9}
${color}Név      PID      CPU%      MEM%
${color #ddaa00} ${top name 1} ${top pid 1} ${top cpu 1} ${top mem 1}
${color lightgrey} ${top name 2} ${top pid 2} ${top cpu 2} ${top mem 2}
${color lightgrey} ${top name 3} ${top pid 3} ${top cpu 3} ${top mem 3}
${color lightgrey} ${top name 4} ${top pid 4} ${top cpu 4} ${top mem 4}
${color}Memóriahasznál
${color #ddaa00} ${top_mem name 1} ${top_mem pid 1} ${top_mem cpu 1} ${top_mem mem 1}
${color lightgrey} ${top_mem name 2} ${top_mem pid 2} ${top_mem cpu 2} ${top_mem mem 2}
${color lightgrey} ${top_mem name 3} ${top_mem pid 3} ${top_mem cpu 3} ${top_mem mem 3}
${tail -var/log/Xorg.0.log 3 30}
$stippled_hr
${color #ddaa00}Portok${alignr}#Kapcsolatok
$color Inbound: ${tcp_portmon 1 32767 count} Outbound: ${tcp_portmon 32768 61000 count}$
${alignr}ALL: ${tcp_portmon 1 65535 count}
${color #ddaa00}Bejövő kapcsolatok ${alignr} Local Service/Port$color
${tcp_portmon 1 32767 rhost 0} ${alignr} ${tcp_portmon 1 32767 lservice 0}
${tcp_portmon 1 32767 rhost 1} ${alignr} ${tcp_portmon 1 32767 lservice 1}
${tcp_portmon 1 32767 rhost 2} ${alignr} ${tcp_portmon 1 32767 lservice 2}
${tcp_portmon 1 32767 rhost 3} ${alignr} ${tcp_portmon 1 32767 lservice 3}
${tcp_portmon 1 32767 rhost 4} ${alignr} ${tcp_portmon 1 32767 lservice 4}
${tcp_portmon 1 32767 rhost 5} ${alignr} ${tcp_portmon 1 32767 lservice 5}
${color #ddaa00}Kimenő kapcsolatok ${alignr} Remote Service/Port$color
${tcp_portmon 32768 61000 rhost 0} ${alignr} ${tcp_portmon 32768 61000 rservice 0}
${tcp_portmon 32768 61000 rhost 1} ${alignr} ${tcp_portmon 32768 61000 rservice 1}
${tcp_portmon 32768 61000 rhost 2} ${alignr} ${tcp_portmon 32768 61000 rservice 2}
${tcp_portmon 32768 61000 rhost 3} ${alignr} ${tcp_portmon 32768 61000 rservice 3}
${tcp_portmon 32768 61000 rhost 4} ${alignr} ${tcp_portmon 32768 61000 rservice 4}
${tcp_portmon 32768 61000 rhost 5} ${alignr} ${tcp_portmon 32768 61000 rservice 5}

```

Ezzel a kinézettel bár grafikus felületen jelenik meg, de úgy néz ki, mintha egy kis konzolos ablakban lenne, s ez nagyon megfelel „Kocka-lelkületemnek”. Mindenki figyelmébe ajánlom e file középső részét, ahol tanulmányozhatja, miként oldottam meg a különböző partícióim foglaltságának kiírását. Annyit jegyeznék meg, hogy mint látható, a színeket RGB módban hexadecimális számokkal adjuk meg a picike grafikonokhoz, s mindegyik ablakocskához 2 szín jár, az egyik a baloldali, a másik az ablak jobboldalára eső, s e két szín között képez a progi színátmeneteket ahogy múlik az idő. Ja, és alapesetben a conky 5 másodpercenként frissíti magát...

A CHMOD parancs

Többször szóbakerült már könyvünkben a **chmod** parancs. Ideje kissé részletesebben beszélnünk róla. Ugyanis nagyon hasznos. Ez határozza meg, hogy ki mivel mit csinálhat.

Némileg pongyolán fogalmazva, mert minden részletbe nem merülök bele e könyvben, e parancs általános alakja:

chmod xyz állománynév

ahol az xyz egy oktális, azaz 8-as számrendszerbeli szám 3 számjegye, az állománynév meg sejtjük ugye, hogy mi... Az xyz ennek megfelelően bármi olyan 3 jegyű szám lehet, aminek egyes számjegyei a 0 és 7 közt mozognak. A legnagyobb helyiértékű számjegy azt mondja meg, hogy mit művelhetsz a fájlal te (általánosabban: a fájl tulajdonosa), a második azt, mit művelhet vele az, aki a tulajdonossal egy csoportban van, a harmadik, azaz legkisebb helyiértékű azt, mit művelhet a fájlal mindenki más, azaz akárki.

A lehetséges jogok a következők: olvasás, írás, végrehajtás (azaz futtatás). Minden oktális szám 3 bitből tevődik össze, a legnagyobb helyiértékű az olvasást, a középső az írást, a legkisebb helyiértékű a végrehajtást jelenti. Ezt legkönnyebb úgy megjegyezni, hogy minden oktális számjegy felírható úgy, hogy $4+2+1$, aholis a 4, a 2 és az 1 helyén is lehet az összegben nulla, ha azonban nem nulla van ott akkor a 4 jelenti az olvasást, a 2 az írást, s az 1 a végrehajtást. Eképp például a

chmod 755 enyifile

azt jelenti, hogy te mint tulajdonos az enyifile fájlt olvashatod, írhatod és futtathatod (mert $4+2+1=7$), a csoportod tagjai illetve bárki más viszont csak olvashatja és futtathatja (de nem írhatja át), mert $4+1=5$).

Gyakoribb számkombinációk és jelentéseik:

777 Bárki bármit tehet a fájlal. Biztonsági szempontból nem javallott.

755 A tulajdonos bármit tehet a fájlal, mások csak olvashatják és futtathatják. Tipikus alkalmazási területe ha te írsz egy mások által is használható szkriptet.

700 Csak a tulajdonos birizgálhatja a fájlt, ő mindent megtehet vele, mások azonban semmit.

666 Ez nem a Sátán száma: azt jelenti, a fájlt bárki írhatja és olvashatja. Ideiglenes fájlok esetén előfordulhat, hogy jól jön ez a beállítás.

644 A tulajdonos írhatja-olvashatja a fájlt, mások csak olvashatják. Nem végrehajtható fájlok esetén használatos.

600 A tulajdonos írhatja és olvashatja a fájlt, mások semmit nem csinálhatnak vele.

A `chmod` parancs értelmezett nemcsak közönséges fájlokra, de tartalomjegyzékekre (mappákra...) is, a következőképp:

- az „olvasási” jog azt jelenti, hogy láthatjuk a tartalomjegyzékben szereplő bejegyzéseket (fájlokat illetve alkönyvtárakat). (Ez azt jelenti, hogy például az `ls` paranccsal kilistázhatod a tartalomjegyzéket. Nem jelenti viszont azt, hogy be is léphetsz a könyvtárba!)
- Az „írási” jog azt jelenti, hogy létre tudunk hozni benne bejegyzéseket, illetve efféle bejegyzéseket törölhetünk benne. Mindezt természetesen a

- szokásos parancsokkal, például a `cp`, `ln`, `mkdir`, `mv`, `rm`, `rmdir` parancsokkal.
- A „végrehajtási” jog azt jelenti, hogy például beléphetsz a könyvtárba, és indíthatod innen a végrehajtható fájlokat.

Miután az olvasási és végrehajtási jog nehezen érthető még ezek után is könyvtárakra, íme egy kis táblázat, a különböző variációkra. A „+” azt jelenti, sikerülni fog a megkísérelt művelet, a „x” azt jelenti, nem fog sikerülni:

Mit akarsz csinálni	milyen jogaid vannak a könyvtáron	
	<i>csak olvasási</i>	<i>csak végrehajtási</i>
Belépni a könyvtárba	x	+
Kilistázni a könyvtár tartalmát	+	x
Kimásolni onnan egy fájlt	x	+
Futtatni onnan egy fájlt	x	+

És még egy fontos dolog: a jogosultságok rekurzív megváltoztatása. Ha mondjuk az „Innen” nevű alkönyvtár összes (akár alkönyvtárakban levő...) fájljának jogosultságait akarod megváltoztatni mondjuk 755-re, így használd a parancsot:

```
chmod -R 755 Innen
```

A mikrofon használata

Természetesen ez is parancssorból. Semmit nem kell telepítened hozzá! A szükséges progí már a rendszeredben van, s a neve: **arecord**
Használatához ajánlott tanulmányozni a man-oldalát (man arecord), de íme az esetek legnagyobb százalékában megfelelő három példa-parancs:

```
arecord -f cd fileneve.wav
```

Ez CD minőségben fogja neked felvenni a mikrofonról az infót (16 bit little endian, 44100, stereo)

```
arecord -f dat ezaneve.wav
```

ez pedig úgynevezett „dat” minőségben (16 bit little endian, 48000, stereo)

Ha pedig monóban akarunk felvenni:

```
arecord -f S16_LE -c1 -r44100 fileneve.wav
```

A kill parancs, azaz hogyan gyilkoljunk programokat

Normális esetben, ha le kell állítani egy progit, legcélszerűbb a programban eleve benne levő kilépési funkciót használni. Ennél durvább módszer, ha egyszerűen bezárjuk azt az ablakot, amiben a progit fut. Erre van megfelelő ikon az ablak fejlécében általában, illetve SithWM alatt erre szolgál a **Win+Esc** billentyűkombináció.

Előfordulhat azonban, hogy e módszerek nem elegendőek. Mert például mi van, ha programhiba miatt egyszerűen lemerevedik az adott progit, és emiatt az istennek sem tudod aktivizálni a kilépési funkcióját?! Cikisebb esetben az ablak bezárása sem működik valamiért. Bár a Linux megbízható oprendszer, NÉHA előfordulhat efféle esemény. Olykor nem is amiatt, mintha a program hibás lenne, de előfordulhat, hogy amúgy normális működése közben megbalt minden memóriát, iszonyúlassan dolgozik, kinszenvedés várni, míg reagál az utasításodra pld egy egérrel való klikkelésre, és inkább úgy döntesz, hagyod a fenébe az egészet, s „le akarod löni”.

Aztán az is lehet, hogy olyan programot akarsz „elhaláloztatni”, ami nem is ablakban fut, hanem mint valami „démon”, a háttérben.

Nos ezen óhajodat, hogy kiélhesd vérengző hajlamaidat és szabadon gyilkolászhasd a programokat, a **kill** parancs valósítja meg. Használata nagyon egyszerű:

kill pid

aholis a „pid” helyére valamiféle szám irandó, ami a meggyilkolni óhajtott program úgynevezett **„Process ID”**-je, azaz magyarán valamiféle azonosítója. Vagyis ehhez tudni kell a progit „pid”-jét. Ezt egyszerűen megtudhatjuk, ha kiadjuk „konzolban”, leginkább persze virtuális terminálban (ez lehet például az Mrxvt) a

ps aux

parancsot. Erre sok mindent kilistáz nekünk, íme egy futásának eredménye a rendszeremen:

```
vz@Csiszilla ~]ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
pv         1  0.0  0.1  1552   536 ?        Ss   16:33   0:01 init [2]
pv         2  0.0  0.0      0      0 ?        S<   16:33   0:00 [kthreadd]
pv         3  0.0  0.0      0      0 ?        S<   16:33   0:00 [migration/0]
pv         4  0.0  0.0      0      0 ?        S<   16:33   0:00 [ksoftirqd/0]
pv         5  0.0  0.0      0      0 ?        S<   16:33   0:00 [events/0]
pv         6  0.0  0.0      0      0 ?        S<   16:33   0:00 [khelper]
pv        107  0.0  0.0      0      0 ?        S<   16:33   0:00 [kblockd/0]
pv        110  0.0  0.0      0      0 ?        S<   16:33   0:00 [kacpid]
pv        111  0.0  0.0      0      0 ?        S<   16:33   0:00 [kacpi_notify]
pv        209  0.0  0.0      0      0 ?        S<   16:33   0:00 [cqueue/0]
pv        213  0.0  0.0      0      0 ?        S<   16:33   0:00 [ata/0]
pv        214  0.0  0.0      0      0 ?        S<   16:33   0:00 [ata_aux]
pv        217  0.0  0.0      0      0 ?        S<   16:33   0:00 [ksuspend_usbd]
```

pv	223	0.0	0.0	0	0 ?	S<	16:33	0:00	[khubd]
pv	226	0.0	0.0	0	0 ?	S<	16:33	0:00	[kseriod]
pv	258	0.0	0.0	0	0 ?	S	16:33	0:00	[pdflush]
pv	259	0.0	0.0	0	0 ?	S	16:33	0:03	[pdflush]
pv	260	0.0	0.0	0	0 ?	S<	16:33	0:00	[kswapd0]
pv	313	0.0	0.0	0	0 ?	S<	16:33	0:00	[aio/0]
pv	334	0.0	0.0	0	0 ?	S<	16:33	0:00	[cifsoplockd]
pv	335	0.0	0.0	0	0 ?	S<	16:33	0:00	[cifsnotifyd]
pv	342	0.0	0.0	0	0 ?	S<	16:33	0:00	[jfsIO]
pv	343	0.0	0.0	0	0 ?	S<	16:33	0:00	[jfsCommit]
pv	344	0.0	0.0	0	0 ?	S<	16:33	0:00	[jfsSync]
pv	346	0.0	0.0	0	0 ?	S<	16:33	0:00	[xfslogd/0]
pv	347	0.0	0.0	0	0 ?	S<	16:33	0:00	[xfsdatad/0]
pv	350	0.0	0.0	0	0 ?	S<	16:33	0:00	[xfs_mru_cache]
pv	356	0.0	0.0	0	0 ?	S<	16:33	0:00	[unionfs_siod/0]
pv	1051	0.0	0.0	0	0 ?	S<	16:33	0:00	[scsi_tgttd/0]
pv	1112	0.0	0.0	0	0 ?	S<	16:33	0:00	[scsi_eh_2]
pv	1114	0.0	0.0	0	0 ?	S<	16:33	0:00	[scsi_eh_3]
pv	1168	0.0	0.0	0	0 ?	S<	16:33	0:00	[exec-osm/0]
pv	1175	0.0	0.0	0	0 ?	S<	16:33	0:00	[block-osm/0]
pv	1183	0.0	0.0	0	0 ?	S<	16:33	0:00	[khpsbpkt]
pv	1245	0.0	0.0	0	0 ?	S<	16:33	0:00	[kpsmoused]
pv	1257	0.0	0.0	0	0 ?	S<	16:33	0:00	[kcryptd/0]
pv	1258	0.0	0.0	0	0 ?	S<	16:33	0:00	[kmpathd/0]
pv	1259	0.0	0.0	0	0 ?	S<	16:33	0:00	[rdac_wkqd]
pv	1260	0.0	0.0	0	0 ?	S<	16:33	0:00	[ksnapd]
pv	1261	0.0	0.0	0	0 ?	S<	16:33	0:00	[kondemand/0]
pv	1267	0.0	0.0	0	0 ?	S<	16:33	0:00	[rpciod/0]
pv	1269	0.0	0.0	0	0 ?	S<	16:33	0:00	[reiserfs/0]
pv	1346	0.0	0.1	1760	624 ?	S<s	16:33	0:00	/System/Links/Executables/
udev	-daemon								
pv	2381	0.0	0.4	3356	2364 ?	S	16:33	0:00	klogd -n
pv	2382	0.0	0.1	1688	728 ?	Ss	16:33	0:00	syslogd -n -m 0
27	2491	0.0	0.1	1956	736 ?	Ss	16:33	0:00	/bin/dbus-daemon --system
26	2498	0.0	0.7	5456	4020 ?	Ss	16:33	0:00	/bin/hald --daemon=yes
pv	2499	0.0	0.1	2776	984 ?	S	16:33	0:00	hald-runner
pv	2503	0.0	0.2	2944	1120 ?	S	16:33	0:00	hald-addon-input:
Listening on /dev/input/event5 /dev/input/									
26	2511	0.0	0.1	1960	860 ?	S	16:33	0:00	hald-addon-acpi: listening
on acpi kernel interface /proc/ac									
pv	2523	0.0	0.2	2944	1028 ?	S	16:33	0:02	hald-addon-storage:
polling /dev/hdc (every 2 sec)									
pv	2524	0.0	0.2	2944	1028 ?	S	16:33	0:02	hald-addon-storage:
polling /dev/hdc (every 2 sec)									
pv	2526	0.0	0.2	2944	1028 ?	S	16:33	0:02	hald-addon-storage:
polling /dev/hdd (every 2 sec)									
pv	2527	0.0	0.2	2944	1028 ?	S	16:33	0:02	hald-addon-storage:
polling /dev/hdd (every 2 sec)									
pv	2562	0.0	0.0	0	0 ?	S	16:33	0:00	[kcdfsd]
vz	2622	0.0	0.4	5692	2172 tty1	Ss	16:33	0:00	-zsh
pv	2623	0.0	0.0	1544	496 tty2	Ss+	16:33	0:00	/System/Links/Executables/
agetty tty2 9600									
pv	2624	0.0	0.0	1544	500 tty3	Ss+	16:33	0:00	/System/Links/Executables/
agetty tty3 9600									
pv	2625	0.0	0.0	1544	496 tty4	Ss+	16:33	0:00	/System/Links/Executables/
agetty tty4 9600									
pv	2626	0.0	0.0	1544	496 tty5	Ss+	16:33	0:00	/System/Links/Executables/
agetty tty5 9600									
pv	2627	0.0	0.0	1544	496 tty6	Ss+	16:33	0:00	/System/Links/Executables/
agetty tty6 9600									
vz	2646	0.0	0.3	5280	1620 tty1	S+	16:33	0:00	/bin/sh
/System/Links/Executables/startx									
vz	2662	0.0	0.1	2624	772 tty1	S+	16:33	0:00	xinit /Users/vz/.xinitrc

```
-- -auth /Users/vz/.serverauth.2646
pv      2663  1.6  5.3  33064 27376  tty7      SLs+ 16:33      5:27 X :0 -auth
/Users/vz/.serverauth.2646
vz      2679  0.1  0.2  2636 1068  tty1      S   16:33  0:28 sithwm
vz      2686  0.0  0.3  4976 1732  ?        S   16:33  0:00 xbindkeys
vz      3107  0.3  14.5 178356 74392 ?        Sl           17:08      1:02
/Programs/OpenOffice/2.4.1_MagyarOS_20080618/program/soffice
vz      3377  0.2  0.4  7432 2232  ?        S   18:25  0:36 conky
vz      3919  0.0  0.3  5296 1640  ?        S           21:56      0:00 /bin/sh
/System/Links/Executables/firefox
vz      3922  0.0  0.3  5336 1672  ?        S           21:56      0:00 /bin/sh
/System/Links/Executables/run-mozilla.sh /System/Lin
vz      3953 11.4 13.6 173676 69584 ?        Sl           21:56      1:39
/System/Links/Executables/firefox-bin
vz      4176  0.1  4.3 220764 22036 ?        Sl  21:56  0:01 java_vm
vz      4208  0.2  0.7  8276 3872  ?        S   22:06  0:00 mrxvt
vz      4209  0.0  0.4  5680 2208  /System/Kernel/Devices/pts/0 Ss 22:06  0:00 -zsh
vz      4502  0.0  0.1  4312  844  /System/Kernel/Devices/pts/0 R+ 22:10  0:00 ps aux
```

Azt hiszem világos úgy nagyjából: ez kilistázta nekünk az összes izémizét, ami fut a rendszerünkben. Nagyon hasznos. Így mindig tudjuk, mivel foglalkozik a masina. A jobbszélső „COMMAND” feliratú oszlopban látjuk a progí nevét, többnyire teljes elérési úttal együtt, a „PID” oszlopban meg – ki hinné – az előbb említett „pid” azonosítót! Mint látható, nálam épp fut egy „OpenOffice”, 3107-es pid-del, s ezek után ha ezt valamiért „le akarnám löni”, ezt kéne beírnom:

```
kill 3107
```

ami így festene a prompt után megadva:

```
vz@Csiszilla ~]kill 3107
```

Sikeres gyilkosság után egyszerűen visszkapjuk a promptot bármiféle visszajelzés nélkül.

A fenti táblázatban egyéb hasznos infók is szerepelnek:

A %CPU azt mutatja, hogy mekkora részét foglalja le a CPU-időnek az adott folyamat, a %MEM azt, hogy a memória mekkora részét zabálja meg a processz, a TTY egyszerűen a terminálazonosító (már ahol ez értelmezhető), a START ugye hogy mikor indult az adott folyamat...

Na most már ezt is tudjuk.

Konzolos naptár

Nagyon jó, ha tudjuk, melyik dátum melyik napra esik. Linuxunk alatt még ehhez sem kell grafikus felület, sőt, külön progít sem kell telepítenünk hozzá. Elég a következő parancs:

```
cal hó év
```

aholis a „hó” helyén egy hónap sorszámát adjuk meg, az „év” helyén meg, biztos kitalálad, egy évszámot. E parancs futása egy példával szemléltetve

ilyesmit eredményez:

```
vz@Csiszilla ~]cal 6 2008
      június 2008
v  h  k  sz  cs  p  sz
1  2  3  4  5  6  7
8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30
```

Gondolom, a táblázat értelmezése elég világos. Persze ha a mai napra vagy kíváncsi, és még azt sem tudod, ma hányadika van, sőt, melyik hónapot vírunk, az elég ciki, de sebaj, ebben segít téged a **date** parancs, még az időt is kiírja neked, nemcsak a dátumot:

```
vz@Csiszilla ~]date
2008. jún. 24., kedd, 23.26.16 CEST
```

Nagy fájlok darabolása és összefűzése

A megoldás:

```
split -b 1024m inputfile outputfile
```

A fenti példa 1 gigabájtos darabokra hasít. Amint látható ugyanis, a **-b** kapcsoló után megabájtkban adtuk meg a méretet. Kilobájtkban is megadhatjuk eképp:

```
split -b 64k inputfile outputfile
```

vagy 512 bájtos blokkokban:

```
split -b 8b inputfile outputfile
```

A kimeneti fájlnevek egy előtagból (prefix) és egy ezt követő betűcsoportból állnak. A betűcsoport olyan lesz, hogy a kimeneti fájlokat névsorban összefűzve az eredeti fájlokat kapjuk vissza. Az alapértelmezett kimeneti előtag az **`x'**. Ha az outfile-prefix argumentum adott, ezt használja előtagnak. Például, ha a következő parancsot adjuk ki:

```
split -b 100k bochs.pdf proba.ff
```

akkor a következő nevű fájlokat kapjuk:

```
proba.ffaa
proba.ffab
...stb...
```

A fájlok összefűzése:

```
cat outputfile.* > eredetifile
```

Ha tehát valaki mondjuk egy tar file-ot darabol, 99 megás darabokra és nem ad meg állománynevet:

```
split -b 99m eredetifile.tar
```

A fájlok összefűzése ekkor:

```
cat x** > eredetifile.tar
```

Illetve ha olyan perverz, hogy Windows alatt akarja egyesíteni őket (vagy nem ő, de a haverja...) akkor:

```
copy /b xaa + xab + xac + ... filename.tar
```

Szótárprogi

Ez egy igazán luxus szótárprogi, merthogy nemcsak konzolon működik majd nekünk, de még grafikus képernyőn is...

Tehát: arról a progiról van szó, melynek neve: **xqed**, és honlapja:

<http://xqed.sourceforge.net/>

Ezt természetesen a

Compile xqed

paranccsal **KÉNE** telepítenünk, de egyelőre ne kísérletezzünk vele! Ne, ugyanis bár látszatra jól fel fog telepedni, valójában nem fog dolgozni nekünk. Elvileg úgy kéne használni parancssorból, hogy a progí neve után megadjuk a keresendő angol szót, de ha például rákeresünk a „home” (vagy bármelyik) szó jelentésére, ezt a hibaüzenetet küldi:

```
vz@Csiszilla ~]xqed home
Traceback (most recent call last):
  File "/System/Links/Executables
/xqed", line 21, in <module>
    import Xqed
  File "/Programs/Python/2.5/lib/python2.5/site-packages/Xqed.py", line 21, in
<module>
    from Tkinter import *
  File "/Programs/Python/2.5/lib/python2.5/lib-tk/Tkinter.py", line 38, in
<module>
    import _tkinter # If this fails your Python may not be configured for Tk
ImportError: No module named _tkinter
vz@Csiszilla ~]xqed dog
Traceback (most recent call last):
  File "/System/Links/Executables/xqed", line 21, in <module>
```

```

import Xqed
File "/Programs/Python/2.5/lib/python2.5/site-packages/Xqed.py", line 21, in
<module>
    from Tkinter import *
File "/Programs/Python/2.5/lib/python2.5/lib-tk/Tkinter.py", line 38, in
<module>
    import _tkinter # If this fails your Python may not be configured for Tk
ImportError: No module named _tkinter

```

Hát ez nagyon csúnya dolog tőle! Csekély ánglius-tudásunkkal azért annyit kihámozunk belőle, hogy valami „tkinter” nevű Python-modul hiányzik neki. (annak aki nem tudná: a Python egy programnyelv neve). Továbbá, ez valami összefüggésben van a Tk-progival is.

No most ha megnézzük a **/Programs** tartalomjegyzéket, látjuk, hogy nekünk VAN feltelepített **Python** progink és **Tk** progink is! Emlékezzünk csak: a **Tk**-t akkor telepítettük, amikor az **Xbindkeys** progit kínlasztottuk fel a rendszerünkre! Mi lehet akkor a baj?!

Nyilván arról van szó, hogy ezek fenn vannak ugyan a rendszerünkben, de nem működnek jól együtt. Valamiért úgy látszik a Python nem vesz tudomást a később telepített modulokról! Nyilván, mert nem úgy lett fordítva a rendszerünkkel együtt telepedő Python. Nos, ezen segítünk! Mielőtt tehát az Xqed-t telepítenénk, egyszerűen telepítsük a Pythont újra.

Normális esetben egy efféle procedúra azzal kezdődne, hogy a **DisableProgram** paranccsal letiltjuk a régi Pythont, de most ezt nem ajánlanám, mert a GoboLinux oly fontos eszköze mint épp maga a Compile, igényli a Pythont sajnos! Azaz durván és nyersen mindössze ezt írjuk be:

Compile Python

Erre ő letölti és lefordítja a 2.5.1 verziójú Pythont. Ezután jöhet a

Compile xqed

parancs, s erre már szépen lefordul a szótárprogí, sőt, még használni is tudjuk majd! Íme egy példafuttatás, rákeresve az angol „dog” szóra:

```

vz@Csiszilla /Programs/Python]xqed dog
dog
to dog [dóg, Br dag] nyomon követ, nyomában van, sarkában van, vaskapoccsal
lefog, vaskapoccsal megerősít, vaskapoccsal rögzít, üldöz
dog [dóg, Br dag] kutya, eb, fickó, hím, vaskapocs, láb, alak, befogópofa,
esztergamenesztő, esztergaszój, esztergaszív, flótás, jártató, kampó,
kanfarkas, kan kutya, közelgő vihart jelző fény a látóhatár szélén, menesztő,
nyargaló, padszolga, pofa, szorító, tuskóbak, tűzikutya, ékszorító pofa

```

Azt hiszem ebből máris világos a parancssori használat.

Bővítsük azonban ki a grafikus menüt! Azaz, ha éppen a SithWM-et

használjuk, oda (a .sithwmrc fájlba) e sort vegyük fel valamelyik almenübe:

```
cmd "XQed" "xqed"
```

Ha újraindítjuk a grafikus felületet hogy már e menüpont is aktivizálható legyen, akkor ezt kiválasztva feltűnik egy ablakocska a képernyőnkön. Üresen. Ellenben ha ezen ablak mögött ott van például a Firefox ablaka (vagy ha akár az OpenOffice-é is...) akkor az egérrel válasszunk ki egy angol szót, például kétszer gyorsan kattintva rá (ekkor inverz csík jelzi hogy kiválasztott), és az XQed ablakában megjelenik a szó angol fordítása! Ha szükségünk van rá, vágólapra is másolhatjuk a fordítást, úgy, hogy az egérmutatóval kijelöljük (katt a bal gombbal és húzás) a kis ablakban levő szöveget, majd Ctrl-C. Példának okáért itt a „power”-szó fordítása:

Power

to power [páUöR] meghajt (motoros járművet), áram alá helyez, áramot ad
power [páUöR] erő, gépi erő, erősség, energia, képesség, hatalom, uralkodás, meghatalmazás, hatvány, befolyás, elektromos áram, fegyveres erő, felhatalmazás, gépesítés, haderő, hadsereg, hajtóerő, hatalmasság, hatáskör, jogkör, lehetőség, magasfeszültségű elektromos áram, mód, nagy, nagyítás, nagyítóképesség, sok, tehetség, tekintély, teljesítmény, teljesítőképesség, testi erő, uralom, áram, élesség
POW [páU] prisoner of war, hadifogoly

Amint látható, még a szó kiejtését is közli egyfajta transzszkripcióban. Sőt: egyszerre egy egész bekezdést is kijelölhetünk, s akkor mindegyik szónak ami abban van, kiírja a jelentését!

Kifejezetten hasznunkra válik most a SithWM azon tulajdonsága, hogy az aktív ablak nem takarja el azt, ami épp felette van, így nem tűnik el az XQed ablaka, amikor újabb meg újabb keresendő szavakra kattintunk mondjuk a Firefoxban!

Persze kényelmetlen mindig a menüből indítani, de hát semmi gond: csinálunk erre is egy „forrógombot”! Én a multimédiás „Home” gombra kötöttem rá, úgy, hogy ezt írtam a .xbindkeysrc fájlba:

```
"xqed"
```

```
m:0x0 + c:178
```

A progi lehet hogy nem olyan csicsás mint sok más, de irtózatosan nagy szóanyaga van!

A kis ablak (SithWM alatt...) a Win+Esc billentyűkombinációval zárható be (ha a SithWM ablakkezelőt használjuk).

Filozofálgatunk – miért épp a GoboLinux?

Bevallom: lusta disznó vagyok! Emiatt e fejezetet úgy írom meg, hogy egyszerűen bemásolok ide egypár cikket, melyek már itt-ott fellelhetőek az Interneten. E cikkeket vagy én írtam - e tényt a cikk címénél jelzem – vagy a GoboLinux hivatalos oldaláról letöltött cikkről van szó, melyeknek azonban én készítettem el a fordítását.

Kezdjük is el!

Viola Zoltán: A legjobban testreszabható Linux disztribúció

Hogy melyik disztró a leginkább testreszabható? A témában kicsit járatosabbak többnyire az LFS-t említik meg elsőként. S annyi való igaz, hogy annál személyesebb disztró nehezen képzelhető el, mint amit te magad építasz fel a semmiből, az alapokból!

Igen ám, csak hogy az mégsem disztró, hanem a te magad személyes rendszere. Az nem egy kész disztró amit aztán személyre szabsz, hanem tényleg az alapoktól kell felépítened, s ez, valljuk be, sok idő, s nem is mindenkinek van kedve annyit tanulni, amennyi kell ehhez!

Most tehát nem az LFS-ről lesz szó. Mindenekelőtt azonban pár szót arról, miért írom e cikket. Nos egyszerűen amiatt, hogy felhívjam a Linux világába most bemerészkedő, s még egyik disztró iránt sem különösebben elkötelezett felhasználók figyelmét egy egészen különleges disztribúcióra, mely azonban nem sok hírverést kapott eddig. A GoboLinuxról van szó. Én is véletlenül találtam rá.

Mintegy 3 éve használok Linuxot, s elsőként az UhuLinux-szal ismerkedtem meg. Máig nagyon szeretem. Tényleg remekül összeválogatott rendszer, s csakis ajánlani tudom minden magyar embernek. Mindazonáltal én egy telhetetlen valaki vagyok, s hamarosan olyasmiket akartam művelni szerencsétlenül, amire nem igazán tervezték! Konkrétan: addig minden remekül működött benne, amíg csak a disztróval együtt szállított programokat használtam, vagy esetleg a direkt hozzá készített UHU-csomagokat telepítettem. Minél több időt töltöttem azonban vele, - márpedig állandóan őt használtam, nincs is idehaza Windowsom! – annál több mindent kellett felraknom alá forrásból, mert nem volt hozzá csomag. No és ekkor szembesültem egy olyan problémával, mely tulajdonképpen nem az UHU hibája, mert jellemző minden más bináris alapú disztribúcióra is! Sőt, ez 2 hiba mindjárt.

Először is, a csomagkezelő egyszerűen semmi progiról nem vesz tudomást, amit nem rajta keresztül telepítenek, hanem például forrásból fordítunk le, vagy mint például a Firefoxot, elég csak kicsomagolni valahová. Ha meggondolatlanul kézzel letörlök valami csomagkezelővel felrakott progit netán, arról sem értesül a csomagkezelő, azt hiszi még fenn van. Azaz, mindezek összevarják a függőségkezelését. Nem tudom ugyanannak a proginak több verzióját

is a gépemen tartani. Továbbá, engem mint kezdő felhasználót rémségesen zavart a Linux „misztikus” könyvtárszerkezete: sosem tudhattam, hogy a telepíteni kívánt progik éppen melyik könyvtárba kerül:

```
/bin;  
/sbin;  
/opt;  
/usr/bin;  
/usr/sbin;  
/usr/local/bin;  
/usr/local/sbin;  
vagy éppen a saját home könyvtáramba...
```

A zűrzavart fokozta számomra, hogy ezeknek a progiknak vannak mindenféle olyan állományai is, amik az **etc**-be kerülnek, de ilyen nevű könyvtárat az **/usr/local**-ban is találtam...

Mondhatja valaki, hogy ez a kezdők baja. És igen, azóta már én is találtam leírást róla, minek alapján szelektálódniuk a programok telepítési helyei. De egyrészt egy kezdőnek ezt senki nem magyarázza meg rögvest az elején, másrészt ez nem is olyan egyértelmű, mert disztribúció-specifikus sok program esetében, hogy mi hová kerül! Számomra messze logikusabb lett volna, hogy legyen valami „Programok” könyvtár, mint a – különben hön utált – Windows esetében, s ott minden proginak legyen egy saját altartalomjegyzéke! Ezt az egyet jónak tartottam a Windowsban, persze, ez még nem volt elegendő ok ahhoz, hogy visszaszokjam rá...

Az is bosszantott, hogy nem tudtam csomagokat készíteni az UHU-hoz. Persze, több tudással biztos sikerült volna, de manapság mindenki mindent hamar akar – bevallom, én is!

Mindenesetre, a fenti okok arra késztettek, hogy más disztrókkal is megismerkedjem többé-kevésbé, keresve magamnak egy jobban megfelelőt. Miért is ne, amikor azt hallottam, hogy több száz disztribúció létezik! Ám most forrásalapú disztribúcióval próbálkoztam, mert már annyit azért tudtam, hogy a bináris disztrók nem kifejezetten barkácsolásra és utólagos módosítgatásokra készültek. Nekiestem a Gentoo-nak. Sajnos ráment az egész karácsonyom, de végül valamiért becsődölt a GCC lefordításakor, s nem volt egy újabb hetem rá...

Következett az LFS. A probléma nagyjából itt is hasonló volt. S így megérlelődött bennem egy elképzelés arról, mi a számomra ideális Linux disztribúció követelményrendszere:

1. Rögvest az elején, installálás után legyen egy komplett, működőképes rendszerem (ellentétben tehát az LFS-sel és Gentoo-val, ahol az alapoktól kell építkeznem).
2. E rendszer rendelkezze grafikus felülettel (is).
3. Meglegyen benne minden szükséges program a parancssori (terminál)

használathoz is, például az MC. Utóbbit számomra érthetetlen okokból sok kisebb disztribúcióból egyszerűen kihagyják.

4. A rendszer az első pillanattól kezdve rendelkezze a forrásból telepítés eszközeivel, például GCC.
5. Legyenek azért hozzá bináris csomagok is a legfontosabb dolgokhoz.
6. Legyen hozzá a függőségeket jól kezelő csomagkezelő,
7. ami a forrásból feltelepített programokat is nyilvántartja.
8. Jól át lehessen tekinteni, mi van a rendszeremben.
9. A csomagkezelő képes legyen csomagokat maradéktalanul eltávolítani, beleértve forrásból fordított programokat is.
10. Ha kézzel letörölök valamit, ez ne zavarja össze a csomagkezelőt (hogypld azt hiszi, megvan a csomag, holott nincs meg).
11. Ha mégis inkonzisztenssé válik a csomagkezelő nyilvántartása, képes legyen önmagát egyetlen parancsra aktualizálni.
12. Ugyanannak a csomagnak/programnak több verzióját is megtarthassam egyszerre a rendszerben (ez libraryknél fontos lehet), és a csomagkezelő azokat is tisztességesen kezelje le. Több változat esetén természetesen mindig legyen egy „alapértelmezett”, azaz aktuális, default, current...
13. Könnyen lehessen benne kernelt fordítani, azaz a kernele ne legyen „agyonpecselt”.
14. Grub legyen a betöltésvezérlője. (Mert én azt szeretem).
15. Legyen LiveCD változata is.
16. Könnyedén készíthessek én is bináris csomagokat alá, még hozzá a már felrakott, jól működő programjaimból.
17. Fel lehessen rakni alá valami bináris csomagból a MAGYAR OpenOffice-ot is. Ez számomra abszolút nélkülözhetetlen program ugyanis. Ezekből eleve kiesett a szórásból sok külföldi fejlesztésű disztró.
18. Általában UHU csomagokat azért fel lehessen rakni alá, mert miért mondjak le róluk?!
19. Full UTF-8 kódolású legyen, hogy lehessen ékezetes állomány- és könyvtárneveket használni.
20. Magyar nyelvű legyen a telepítője.
21. A honlap elérhető legyen magyarul.

Mindez jó sok igény egyszerre! Nos, anélkül hogy belemennék mélyebb történelmi részletekbe, a lényeg az, hogy végül rátaláltam a GoboLinuxra, ami kapásból kielégítette a fent felsorolt 21 pont közül az első 16-ot! Továbbá, hamarosan kiderült – hála Udvari Zsolt barátomnak – hogy csekély trükközéssel fel lehet bűvészkedni alá lényegében bármi RPM csomagot is, s ezekből a magyar OpenOffice is felrakható alá, az RPM csomagból. Továbbá, minthogy az UHU-csomagok felépítése is nagyon hasonló az RPM-hez, ez azokra is érvényes. Tehát most már teljesült az első 18 pont.

Azóta újabb két pont teljesült: „Magad uram ha szolgád nincsen” - e jelmondat szellemében lefordítottam én magam a disztró honlapját, az tehát

immár elérhető a magyar linuxosoknak is az anyanyelvünkön.

(E link alatt érhető el: http://gobolinux.org/index.php?lang=hu_HU)

A minap végeztem a telepítő magyarra fordításával, a következő kiadásban tehát már az is szólhat magyarul hozzánk.³ Az UTF-8 komolyabb dolog, de nem olyan vészes hiány: a legtöbben úgysem használnak ékezetes neveket a filerendszerben.

No és most hogy miért olyan jó a GoboLinux? Itt maga a filerendszer a csomagkezelő! Igen, a GoboLinux egy alternatív Linux disztribúció, amely újra-definiálja a fájlrendszer-hierarchiát. A GoboLinuxban nincs szükség külön csomagkezelőre, minthogy maga a fájlrendszer a csomagkezelő: minden program a maga saját, külön mappájában van elhelyezve, mint például `/Programs/Xorg/7.1/` és `/Programs/KDE-Libs/3.5.3.`

Mindezt természetesen különböző szkriptek hajtják végre nekünk, melyek kezelése azonban nem nehéz. És természetesen van grafikus csomagkezelő is, ha nem akarunk parancssorral bajmolódní, de ez a csomagkezelő végeredményben mégsem „igazi” csomagkezelő, mert mint látható maga a filerendszer az adatbázisunk tulajdonképpen – e „csomagkezelő” pusztán egy a parancssori szkript hívogatást elfedő grafikus előtétprogram!

Csomagokat itt a legkönnyebb csinálni: az egész egyetlen parancs kiadásából áll. Például ha a rendszerünkbe telepített XXX programból csomagot akarunk készíteni, elég e parancsot kiadnunk:

```
CreatePackage XXX
```

Ennél egyszerűbben tényleg nem lehet...

További hasznos linkek:

A GoboLinux immár magyarul is elérhető honlapja:

http://gobolinux.org/index.php?lang=hu_HU

A GoboLinux hivatalos magyar fóruma: <http://linux.birodalom.net/smf>

A GoboLinux wiki alapú magyar TudásBázisa:

<http://linux.birodalom.net/gobowiki>

Magyar GoboLinux csomagok (például a magyar OpenOffice):

<http://linux.birodalom.net/gobowiki/index.php/Csomagok>

Befejezésként még arra térek ki, kiknek ajánlható a GoboLinux! Mindenekelőtt SENKINEK SEM, aki egy „full-extra-de-Luxe” rendszert akar, amely a gondolatait is kitalálja. Ez nem „fapados” rendszer, rögtön használható, van grafikus felület is, de nem arra hegyezték ki, hogy már előre kitalálja a felhasználó szándékait. Ez arra van, hogy majd te magad hozzáigazítod a saját elképzeléseidhez! Épp emiatt azoknak ajánlható, akik nem kezdenek el sikoltozni, ha valamit csak úgy lehet beállítani, hogy kimennek parancssorba, és ott az MC editorával átírnak valamit egy config fileban. Itt te lényegében kapsz egy natúr Linux-rendszert, amit egészen a grafikus felületig felépítettek neked, mondhatni egy LFS-t, amit aztán a saját elképzeléseid szerint átgtyúr-

³ Amiről e könyv szól, az már ez a magyar telepítőjű kiadás, azaz Te, Tisztelt Olvasó, vélhetőleg épp ezt a magyarított GoboLinuxot használod már!

hatsz, de úgy, hogy nem kell mindig előről kezdened a változtatgatást, rögvessz „harcra kész” az egész, mindig annyit módosítasz rajta, amennyire időd és kedved van, a rendszered eképp végig használható marad, nem kell egy hetet eltotojáznod a legelején, mint a Gentoo-nál...

Ez tehát azoknak való, akik használni akarják a rendszert, azonnal, de kísérletezni is akarnak vele, hekkelni is szeretnék azt. Nem mindent – ha mindent, akkor marad az LFS nekik – de sok mindent! A GoboLinux remek választás azoknak, akik úgy érzik, nem képesek még az alapjaitól összeállítani egy LFS-t, esetleg nem is óhajtanak ennyire mély tudásra szert tenni, talán nem akarnak saját szkripteket írogatni, de mindenképp igényt tartanak rá, hogy végső hatalmuk legyen afölött, milyen progikat tartalmazzon a rendszer, mik legyenek forrásból fordítva és mik binárisból telepítve, mi hová legyen installálva, könnyen lehessen mindent frissíteni... És egyáltalán, egyetlen könyvtár átnézésével rögtön tudják, mi van és mi nincs a rendszerükben! Ha ilyen igényeid vannak, akkor most már csak annyi van hátra, hogy töltsd le, és próbáld ki!

Hisham Muhammad: Az Unix-struktúra újragondolása

Mostanában sok vita folyt a Linux desktop rendszerként betöltött helyzetéről, és néha előbukkannak itt és ott nagyon jó ötletekkel is bizonyos cikkek. Mindazonáltal semelyik nem lepett meg engem jobban, mint ez. Bár nagyon hipotetikus volt, de elég radikális ötleteket tartalmazott arról, hogy a Linux fájlrendszer-hierarchiát át kellene szervezni. Világos, hogy ez volt a cikk legvitathatóbb része, és gerjesztett is sok vitát, hogy ezt ténylegesen miként is valósíthatnánk meg. És igazán meglepő volt, hogy ezt sikerült is megvalósítani egy év alatt. A GoboLinux egy Linux disztribúció egy alternatív fájlrendszer-hierarchia alapján, ami egy egyedi LFS installációból nőtte ki magát egy olyan disztróba, amit a mai napig is fenntart és használ egy kicsi csoport. Érdekes volt látni, hogy sokan vannak emberek, akik a mieinkhez hasonló ötletek iránt érdeklődnek. Tehát talán ez a megfelelő idő ahhoz, hogy e disztró kilépjen az ismeretlenségéből.

Egy kis történelem

Mindannyian emlékszünk arra az időre, amikor a desktop Linux disztribúciókról folyó vita az jelentette, hogy arról vitáztunk, melyiknek van a legjobb telepítőprogramja. Azóta sok minden fejlődött: a könnyű, grafikus telepítő-programok már rendelkezésünkre állnak, de még nem vagyunk a célnál. A szokásos hanták között – „miért van az, hogy (ide beilleszthetsz tetszőleges problémát)”, - néhány érdekes ötlet is felbukkan néha-néha. Érdekes módon néhány ember úgy gondolta, hogy talán ez a kalandosabb kísérletekhez megfelelő időpont.

Furcsamód a GoboLinux nem efféleképp kezdődött. Az egész úgy indult, hogy az egyetemen elkezdtem programokat telepíteni. Mivel nem volt jogom írni a standard Unix könyvtárakat, elkezdtem elkészíteni egy könyvtárrendszert a \$HOME-om alatt. Nem használtam csomagkezelőt, a programokat folyamatosan forrásból telepítettem. A megoldást abban láttam, hogy minden programot a saját könyvtárában helyeztem el, például: `~/Programs/AfterStep`. Hamarosan a környezeti változók (`PATH`, `LD_LIBRARY_PATH`...) egyre nagyobbak és nagyobbak lettek, úgyhogy központi könyvtárakat készítettem a különböző típusú fájloknak, szimbolikus linkeknek: `~/Libraries`, `~/Headers` és így tovább. A természetes evolúció következtében írtam egy shell szkriptet, amely kezelte a linkeket, konfigurációkat és a Makefile-okat.

A rendszer nagyon használhatónak bizonyult. Az otthoni rendszeremen elkezdtem fokozatosan eltávolítani az előre-fordított (pre-compiled) csomagokat, és újrafordítottam őket forrásból a szkriptek segítségével. Szépen haladtam tovább egy teljesen személyreszabott Linux rendszer felé, amelyet viccesen LodeLinux-nak hívtam. Amikor körülbelül 80%-ban kész voltam, jött a Nagy Filerendszer Pusztulás. Itt volt az idő, hogy nekilássak újra az egésznek, de most egy más úton haladva: ahelyett, hogy „lepusztítottam” volna egy létező disztribúciót, egy barátom (André Detsch), és én (Hisham Muhammad) két napi munkával létrehoztunk egy módosított *Linux From Scratch* rendszert. 2002. március 20.-án minden különösebb hűhó nélkül megszületett a GoboLinux. Egy hónappal később bemutattunk egy cikket a Free Software harmadik workshopján, amely az „Egy új javaslat a Unix könyvtárstruktúrára” címet viselte.

Mi ez az egész?

A GoboLinux határozottan nem csupán egy újabb desktop Linux disztró. Teljes egészében egy alternatív könyvtárszerkezeten alapul. Minden program a saját alkönyvtárában van: A `/Programs/XFree86/4.3/` -ban fogod megtalálni az XFree86 4.3 -at, és a ping-et a `/Programs/Netkit-Base/0.17/bin/ping` -nél. Ahhoz hogy lásd, milyen programok vannak telepítve a rendszeredbe, minden, amit tenned kell, mindössze ennyi: `ls /Programs`.

A fájlok mindegyik típusának ott egy mappa a `/System/Links` alatt, mely szimbolikus linkek formájában csoportosítja ezeket mindegyik alkalmazásból: **Executables**, **Libraries**, **Headers**, **Shared** és **Manuals**. Kompatibilitási okok miatt, mindegyik „hagyományos” tartalomjegyzék létezik egy szimbolikus link formájában, mely a megfelelő kategóriára mutat. Azaz, a `/bin`, `/sbin`, `/usr/bin`, `/usr/local/bin` (és így tovább) mindegyike egy-egy symlink a `/System/Links/Executables`

könyvtárra. A környezeti változókat emiatt szintén leegyszerűsíthetjük: Elég mindössze az

```
export PATH=/System/Links/Executables
```

Röviden, amit kaptunk, az egy adatbázis nélküli csomagkezelő-rendszer: **maga a fájlrendszer-hierarchia szervezi a rendszert** (vagy talán nem az

volt az eredeti célja végülis?!). Mindegyik program-mappa (például `/Programs/KDE`)

rendelkezik verzióbejegyzéssel (`/Programs/KDE/3.0`, `/Programs/KDE/3.1.1`), és egy verzió-semleges alkönyvtárral, ami a beállításokat tartalmazza (`/Programs/KDE/Settings`), ami általában a `/etc` -ben volna. Semmiféle gondot nem jelent az sem, ha egy program két vagy több verzióját is meg akarjuk tartani egyszerre. Amikor a legtöbb disztró átváltott a **GCC 3** egy újabb major verziójára, javarészt inkompatibilissé váltak a korábbi változatokkal. Amikor a **GoboLinux 006** sorozata áttért a **GCC 3** -ra, ez neki csak annyi gondot jelentett, hogy az újak mellett megtartotta a libraryk régebbi verzióit is, amíg azokat fokozatosan megszüntették. Nincsenek bonyolult „compact” csomagok. A legtöbb feladatot a GoboLinuxban szkriptek egy gyűjteményével automatizáljuk. Ahhoz, hogy létrehozz egy GoboLinux csomagot, gépeld be például, hogy: `CreatePackage CoreUtils`. Amit e parancs csinál, az az, hogy eltárolja a `CoreUtils/5.0/` és a `CoreUtils/Settings` tartalmát egy `.tar.bz2` fájlban, aminek a neve `CoreUtils--5.0--i686.tar.bz2` lesz. A `/Programs/CoreUtils/Current` nevű link jelzi, hogy melyik verzió az aktuális éppen. Ezt alapértelmezett verzióként használják a szkriptek. Egy program telepítése 3 lépésből áll: `PrepareProgram`, ami létrehozza a `/Programs/` tartalomjegyzék-hierarchiát, és konfigurálja is a megfelelő opciókkal. A `SymlinkProgram` létrehozza a szimbolikus kötések a `/System/Links` könyvtárban. A `Compile` szkript hajtja végre a hagyományos `configure`, `make` és `make install` parancsokat (a parancssori opciók esete speciális kezelést igényel).

Alternatív indítórutinok

Mióta elkezdjük a „semmiből építeni” az egészet, és igazán egy olyan rendszert akartunk csinálni, ahol minden értelmet nyer a számunkra, elérkezettnek láttuk az időt, hogy újragondoljuk a rendszerindítás folyamatát is. Úgy éreztem, a két történelmi modell (System V és BSD) feleslegesen bonyolult egy desktop telepítés kedvéért. A GoboLinux egy egyszerűbb rendszert használ: két szkript, az `Init` és a `Done`, a munka legnagyobb részét elvégzi. A további szkriptek, mint például `Multi` és `Single`, gondoskodnak a futási szintekről. Ezek a fájlok egyszerűen parancsok sorozatai, mely parancsokat a bevezető „`Exec`” szócska, s az utána következő üzenetstring vezet be. Itt van példaként az `Init` egy részlete:

```
Exec "Setting clock..." SetClock
Exec "Loading keymap..." loadkeys "$KeymapLayout"
Exec "Bringing up the loopback interface..." ifconfig lo 127.0.0.1
```

Több bonyolult feladatot, mint például `SetClock`-ot, shell funkcióként határozzunk meg egy `Tasks` fájlban (ezeket a feladatokat szintén meg lehet hívni parancs-sorból a `RunTask` szkriptet használva). A konfigurálható beállítások környezeti változókként vannak definiálva az `Options` fájlban. Az `Exec` parancs egy további feladatot is remekül ellát: a betöltőképernyő témáinak

kezelését. A betöltési sorrend hasonlóan tűnhet a Slackware-hoz (a szabványos (hiba)üzenetek miatt), a RedHat-hez (a sok „OK” miatt), vagy a GoboLinux-hoz (az utóbbi a Linux Progress Patch egy módosított verzióját használja).

A „hagyományos” tartalomjegyzék-struktúra

Sajnos, nem minden program olyan rugalmas, hogy bárhová lehessen telepíteni. Néha fixen „behuzalozott” útvonalak vannak még olyan programokban is, amik pedig a felhasználók számára íródtak, és ezekből – legalábbis elméletileg – meg kellene engedjék, hogy bármely alkönyvtárba telepíthessék őket, mondjuk, egy felhasználó HOME könyvtárába.

Hosszú távra előretekintve, minden alkalmazást átírni nem egy használható alternatíva. Ezen okból kifolyólag a GoboLinux rendelkezik a „hagyományos” fájlrendszer-hierarchiával is, ahol minden szokásos Unix-útvonalat leképeznek a GoboLinux-ban rendelkezésre álló megfelelőire. Ha például a **Makefile** a `/usr/X11R6/include/X11/Xaw3d/XawInit.h` -t keresi, úgy meg is fogja találni azt, bár ez a `/Programs/Xaw3d/1.5/include/X11/Xaw3d/XawInit.h` -nál helyezkedik el. Amikor két alkalmazásnak ugyanazzal a névvel van egy tartalomjegyzék-bejegyzése, a GoboLinux szkriptek kiterjesztik azokat. Például az **XFree86** és az **Xaw3d** is tartalmaz **X11** bejegyzést az „include” alatt. A `/System/Links/Headers/X11` bejegyzés automatikusan létre lesz hozva, miközben be lesz oda linkelve mindkét **X11** alkönyvtár.

Egy másik érdekes jellemző az, hogy a GoboLinux szkriptek a **make install** végrehajtásához speciális user ID-t használnak, aminek csak a programok forráskönyvtárára van írási engedélye, és magára a programkönyvtárakat tartalmazó `/Programs` könyvtárra. Ezáltal a fájlok nem kóborolhatnak el a GoboLinux hierarchiából a „hagyományos” filerendszerbe.

A GoboLinux fájlrendszer-hierarchia friss, tiszta áttekintést nyújt a Linux rendszerre, de a régi struktúra jelenléte, amíg szükséges, sajnos elviszi ennek a szépségnek egy részét. A Mac OS X arra használ egy „aljas trükköt”, hogy titkolja az Unix-természetét: a Finder nem fogja megmutatni az Unix-mapákat, de láthatod őket a parancssorból. A Linuxban nekünk számos lehetőségünk adódik arra, hogy keresgéljünk a fájlrendszerben (shellek, állománykezelők, böngészők...), tehát nekünk mélyebben kellett belenyúlnunk a rendszerbe valami hasonló eredmény elérése érdekében. A GoboHide egy (nyilvánvalóan szabadon választható) kernel patch és ehhez való felhasználói felület, melyet *Lucas Correia Villa Real* és *Felipe Damásio* írt, s amely tényleges „rejtett állományokat” valósít meg Linuxon (túl a pont-fájlokon, melyek az UI-szinten vannak implementálva, a Finderhez hasonlóan).

Itt vannak az `ls /` által mutatott bejegyzések a GoboLinuxban:

Depot
Mount
System
Files

Kapcsolódó fogalmak

Ahogy olvasod ezt, valószínűleg sok dolgot ismerősnek találtál (például nem megmutatni bizonyos tartalomjegyzékeket). A GoboLinux nyilvánvalóan ihletet merített más operációs rendszerekből, mint a Mac OS X, BeOS és AtheOS, de az volt az elképzelésem, hogy építsünk „valami különbözőt”, ami egy létező Unix-alapot használ (legyen ez egy Unix-kernel mint az az OS X-ben van, vagy a GNU eszközök az AtheOS-ban), és magába olvasztja a legjelentősebb találmányokat. Jelenleg több más project is folyamatban van különböző fejlesztési stádiumban, amik a Linux kernelt használják, és alternatív fájlrendszer-struktúrát valósítanak meg. Érdekes módon a legtöbbjük klón, vagy nagyonis egy speciális saját operációs rendszer által ihletett: a ROX OS egy RiscOS-hez hasonló rendszer kíván lenni, a LinuxSTEP egy projekt, ami a GNUSTEP-en alapulva egy NeXT-hez hasonló rendszer felépítését célozza meg, és a BlueEyedOS esetén a cél egy forrás-szinten a BeOS-szal kompatibilis klón.

Másfelől a GoboLinux nem klónja bárminek is. Szabványos Linux desktop szoftvert használ. Úgy véljük, hogy az átszervezett könyvtárstruktúra jó tesztelési környezetet nyújt új ötletek kipróbálásának, -- a lehetőségek tágra nyíltak.

Hol tartunk most?

A GoboLinux az utolsó év folyamán mérhetetlenül sokat fejlődött. A gobo-linux.org-nál találhatsz egy ISO-t a legújabb verzióval, ami a 006. ((Már a **014** - a fordító megjegyzése)). Ez egy bootolható LiveCD, ami a GoboLinuxot egy chroot-olt csak olvasható fájlrendszeren futtatja, tehát megismerkedhetsz a módosított könyvtárszerkezettel, és kipróbálhatod, mennyire tetszik neked a GoboLinux. A CD tartalmazza a szokásos telepítőszkripteket és extra csomagokat is. Holmi csilivili grafikus telepítőprogram nincs még, (már **VAN** - a fordító megjegyzése) de inkább a könnyű telepíthetőségre s a rendszer könnyű használhatóságára összpontosítottunk (sok más disztribúciótól eltérően).

A nagyon kicsi felhasználói bázis ellenére (mi igazából még sehol sem jelentettük be azt), a projekt gyorsan halad előre és teljesen használható. (a legtöbb GoboLinux fejlesztő az egyetlen operációs rendszereként használja azt, és akadnak még páran, akik szintén így tesznek).

Hisham Muhammad: Az ötlet, amin a Compile alapul

Megjegyzés: tegnap éjszaka az IRC-n igazán jól beszélgettem Nathannel, és mivel úgy gondolom, amiket ott a Compile filozófiájáról mondtam, mások számára is elolvasásra érdemes, lementettem a beszélgetést. Mindazonáltal mindenki tudja, hogy az IRC log-ok olvasása kinszenvedés, emiatt megpróbálom azt átdolgozni. Íme:

Többször is neveztem már tréfásan a forrásból való telepítést a „szegény ember csomagbázisának”. Nem azért, mert nem jellemző rá a változatgazdagság (bár ez természetesen igaz), hanem mert ez csak minimális tervezést igényel.

A Linux-világban amelyben ma élünk, a programok több tonna terítési formátuma: - .tgz, .deb, .rpm... - vesz körbe minket - sőt még ugyanannak a formátumnak is számos variánsa - (SuSE rpm, RedHat rpm, Mandrake rpm...). Egy program elméletileg ugyan rendelkezésre állhat ezen formátumok mindegyikében, de, különösen a kevésbé alapvető programok esetében, általában csak e terítési formátumok egy részhalmazában hozzáférhető, végső soron tehát a felhasználó számára csupán egyetlen formátum léte az, mely minden program esetében biztosan garantált: az eredeti **.tar.gz** állomány, mely a forrást tartalmazza.

Az esetek legnagyobb részében ezt az **.tar.gz** fájlt aztán egyszerűen kicsomagolják, és három „bűvös parancsot” kiadva - „configure, make, make install”, sikerül is üzembehelyezni a programot. A függőségeket a configure ellenőrzi, és a program a megfelelő, a rendszerben mutatkozó könyvtárakhoz lesz belinkelve, inkompatibilitások nélkül. Annyira, hogy ha egy **.tar.gz** fájl tisztességesen lett összeállítva, és követ néhány GNU előírást, nincs ok, amiért ez ne működne bármely Linux disztribúcióban. Ennélfogva ideális esetben nem is lenne szükség terítési formátumokra, csak a **.tar.gz** állományokra.

Sok project próbálja „egységesíteni” a disztribúció-formátumokat egy vagy több módon: AutoPackage, LSB, stb. Ezek többségében azt a problémát látom, hogy általában megelégednek azzal, hogy - mindegyikük a maga módján - egy project-specifikus „új réteget” adnak hozzá, megfertőzve így a (mondjuk így) eredeti, „tiszt” **.tar.gz** állományt.

Másfelől az olyan projectek mint a GNU Autoconf és a GNU Libtool, könnyűvé teszik a programok különböző környezetbe való beillesztését, anélkül, hogy megkövetelnék a rendszertől, hogy létezzék ez vagy az az útvonal a csomagok együttműködése érdekében. Néhány Autoconf eszközre van csak szükségünk ahhoz (**--prefix**, **--sysconfdir**...), hogy a program GoboLinux-barát legyen. Nincs szükségünk arra, hogy legyen holmi saját csomagformátumunk, ami tárolja és szervezi a forrást, illetve valami „saját verziót” tárol a project forrásaiból. Ahogy Nate azt nagyon helyesen mondta, „el tudsz menni a project honlapjára, aztán töltsd le az alkalmazást és telepítsd... és nem kell speciális csomagokat keresned, csak a te „saját disztród oldalát” használva. Bár egyre inkább javul az Autoconfnak és egyéb GNU projekteknek nyújtott támogatás, annyira, hogy ezek ténylegesen a Szabad Szoftverek Világának

szabványaivá váljanak, tudjuk eközben, hogy a valóság most még nem ilyen szép. Emiatt rendelkezünk jelenleg a Compile eszközzel és az ő recipe-gyűjteményével.

A Compile eszköz ezen az eszmén alapul. A recipék nem egy újabb típus az ebuild-ok, rpm specifikus fájlok és hasonlók közt. Ideális esetben ezeket nem szabad utasításgyűjteménynek tekinteni, mely megszabja, miként építsük be a programot, de elég alaposan deklarálja, miként is játszódják le a fordítási eljárás.

A Compile azon a tényen alapul, hogy adott néhány szabványos út, melyet a források követnek, és ezeknek akad néhány kisebb variánsa, amelyeket megfelelően részleteznek. Ezek mindegyike egy úgynevezett „eljárásmodell”, melyeknek külön támogatást valósítanak meg. Van három fő végrehajtási modell: compileprogram (amelyik configure-bázisú fájlokat használ), makefile (amely configure-nélküli make/make-install projectekben használatos) és az xmkmf (az X imake eszközökön alapuló alkalmazások számára). Van egy további módszer is, a meta, amelyik meta-csomagolást támogat (több csomag egybecsomagolása).

Egy ideális recipe így néz ki:

```
url=$httpSourceforge/gphoto/libgphoto2-2.1.3.tar.bz2
is_compileprogram=yes
```

Nem több mint két sor. És terveim közt szerepel egy olyan compileprogram elkészítése, mely alapértelmezés szerint mindössze egy sorra lemenne. Mielőtt ezt az egész Compile-projectet elindítottam volna, tudtam, hogy a Recipe-tároló karbantartási szükséglete lesz az egésznek a legkeményebb része (Voltaképpen épp ez akadályozta meg, hogy nem indíthattam el korábban). Így az elsődleges prioritásunk a Recipe-tároló frissítéseinek a lehető legalacsonyabb szinten tartása. Azt gondolom, ezt nem lehet sokkal lejjebb szorítani annál, mint amit eddig megvalósítottam.

Amint neked van egy alapvető végrehajtási módszered, abba „besompolyognak” mindenféle „egyéni jellegzetességek”. A Compile támogatja a különböző kisebb variációk leírását, melyek előfordulnak a programok összeállítására szolgáló egyéb rendszerekben. Például a PSUtils a Makefile-ot Makefile.unix-nak hívja, vagyis: makefile=Makefile.unix

Amint kicsomagolod az **audacity-src-1.0.0-2.tar.gz** állományt, kiderül, hogy az Audacity forrásai nincsenek az **audacity-src-1.0.0-2** -ban, ahol várnánk, ellenben az **audacity-src-1.0.0** -ban, azaz **dir=audacity-src-1.0.0**

És így tovább. Akad jónéhány érdekes deklaráció, mint például configure-opció a 'compileprogram' végrehajtási modelljében, és 'make-változó' a 'makefile' modellben (ezek szintén támogatottak a 'compileprogram'-ban). Sajnos nem csekély számú csomag igényli az installációja előtt illetve után bizonyos speciális parancsállományok végrehajtását, ahhoz, hogy rendben működjenek. Ezek az esetek shell szkript funkciókkal vannak megtámogatva.

Azaz, a recipegyűjtemény célja, hogy legyen egy olyan tudásbázis, melyből tájékozódhatsz afelől (pontosabban nem te, hanem a Compile), hogy mi a program lefordításának a szabványos módja. Reméljük, hogy a szabad szoftverek fejlődésével a recipe-gyűjtemény internetcímeinek csokra is rohamosan bővülni fog. :-)

Jelenleg a **Compile/Recipe** gyűjteményt fájl-bázisú rendszer jellemzi, helyben tárolván minden recipét. A recipék szerveren tárolása és azonnali letölthetősége egy logikus és tervezett jövőbeli irányzat. A legutóbbi **Compile/Recipe** lista tartalmazza az összes szkriptet és recipét, amivel eddig rendelkezünk. Ez 150-nél is több csomagot jelent, mindössze 68K méretben, úgyhogy ez valószínűleg egy jó dolog.

Perspektívaként megjegyezhetem, hogy a Debian kb 4000 csomagot támogat, és a Gentoo-t is hasonló szám jellemzi már. Hallottam azt is valahol, hogy a BSD gyűjteménynek valami 8000 csomagja van. Arra való tekintettel, hogy a Compile project kevesebb mint egy hónapja indult, úgy gondolom hogy meglehetősen gyorsan fejlődünk. A legfontosabbnak azt tartom, hogy a projectünk magában rejti annak lehetőségét, hogy jóval magasabb szintre kúszik fel, mint a .deb vagy hasonló fájlok gyűjteménye, miközben a karbantartása nagyságrendekkel kevesebb munkát jelent.

Arra biztatlak, hogy próbáld ki a Compile-ot, és küldj nekünk recipéket. A Compile használatához csomagold ki a legújabb változatot a **/Depot**-ba. A legutolsó változatot itt leled meg: <http://gobolinux.org/~lode/GoboLinux/> (ez általában a legutóbbi Scripts csomagot is igényli, amit mindig itt találhatsz meg: gobolinux.org).

A Compile afféle alkalmi módon van összecsomagolva, mert a szkripteket és a recipéket egyetlen csomagban akarom tárolni... gyors a fejlődés. Kissé stabilizálom őket, és a terv az, hogy ezeket a **/Programs/Compile** és a **/Files/Recipes** könyvtárak tartalmazzák. A következő lépés természetesen a recipéknek a szerverre költöztetése.⁴

(Most lehet hogy arra gondolsz: e beszélgetés egésze a forrásokról szőtt... Mi köze van ennek a bináris csomagokhoz? Nos, bináris csomagokat forrásokból lehet készíteni, emiatt leginkább a forrásokra kell fókuszálni a figyelmünket/erőfeszítéseinket, aztán a binárisok már mintegy „következnek” ebből).

⁴ Ez időközben megtörtént. (A fordító megjegyzése).

Hisham Muhammad: Nem vagyok tanácsstalan

avagy

mítoszok és félreértések a GoboLinux tervezésével kapcsolatban

*"Azok a szerencsétlenek akik nem értik az Unixot,
arra vannak kárhoztatva, hogy újra kitalálják."*

- Henry Spencer, 1987

Ezen a héten kibocsátottuk a GoboLinux egy újabb verzióját, és sok ember ismét, még ha közvetett módon is, „tanácsstalan”-nak nevezett engem, sok dolgot kifogásolván a disztribúció struktúrájával kapcsolatban. Ezen kifogások közül egyik sem volt új; sokszor hallottam már mindegyiküket. Ez a cikk egy kísérlet e kifogások összefoglalására, és annak megmagyarázására, hogy miért döntöttem épp így vagy úgy. Ezzel remélhetőleg tisztázok minden felmerülő félreértést. Nem ringatom magam olyan illúziókban, hogy ez megakadályozza majd az embereket ezen kifogások újbóli előrángatásában, de legalább lesz egy szövegem, amit egyszerűen az orruk alá dughatok. Ez a cikk azon általános félreértésekről ad számot, melyek azoktól erednek, akik soha nem használtak még GoboLinuxot, és jó szándékú, de át nem gondolt ötletekről is szól, amik állandóan felbukkannak a GoboLinux levelezőlistában, miközben gyakran okoznak hosszú, terméketlen vitákat. Szekciókban fogom elválasztani a pontokat, annak érdekében, hogy közvetlenül ahhoz ugorhass, ami érdekel téged, amennyiben nem érzel kedvet az egész írás végigolvasásához.

Oka van annak, hogy a dolgok olyanok, amilyenek

Ezt valamit, amit állandóan hallok, gyakran követi egy magyarázat a `/`, `/usr` és `/usr/local`, és/vagy a `/bin` és `/sbin` közti különbségről. Értem a különbséget⁵. Ha eltöröltem ezt a három szintű megkülönböztetést, annak az az oka, hogy hiszek abban, miszerint van más mód is azon problémák orvoslására, amik e hagyományos megkülönböztetést életre hívták. Egy GoboLinux rendszerben semmi érv nem szól amellett, hogy legyen különálló `/usr` és `/usr/local` azért, hogy elkülönítse a disztribúció által szállított programokat azoktól, amiket a felhasználó fordított magának. Mindegyik program természetes módon elkülönített, és pontosan ez is a legelső helyen emlithető szándékunk azok közül, melyek végül a GoboLinux létrejöttéhez vezettek.

Az a történelmi ok, amiért az Unix-rendszerek tartalomjegyzékeinek egy

⁵ Azok számára, akik még mindig csodálkoznak (illetve akik meg szeretnének győződni róla, valóban értem-e a három könyvtár közti különbséget), a három fő könyvtárba csoportosítás lényege: Fájlok, amiket egyfelhasználós rescue-üzemmódban el kell tudni érni a root-partícióról; a disztribúció által szolgáltatott állományok; és a rendszergazda által elkülönítetten telepített programok (ez a megkülönböztetés változhat a különböző Unixokon, de főleg ez az, ami szerint a Linux disztribúciók dolgoznak, olvass utána). A hagyományos programok mennek a `/bin`-be, és a programok, amiket a rendszergazdának szántak, az `/sbin`-be.

része közvetlenül a gyökérkönyvtárból ered (`/bin`, `/lib`, `/sbin`), ellentétben azokkal amik a `/usr` könyvtárból nyílnak, nos az nem más, mint hogy így módodban áll bebootolnod egyfelhasználós üzemmódban, csupán ezen, a gyökérből nyíló fájlokat használva, megjavítandó velük a `/usr` könyvtárfa esetleges hibáit. Ez azonban csak „vallási hittétel”. Amikor meg kell mentenem a rendszeremet, inkább egy teljesértékű Linux rendszerrel felszerelt LiveCD-t használok, ami még grafikus környezetet is biztosít a számomra, az megengedi nekem, hogy böngésszem a világhálót és ott keressek megoldást a problémámra, és egyáltalán, egy teljes rendszer minden lehetőségét felhasználhassam a javítás érdekében. Tisztában vagyok azzal, hogy mi volt az értelme a régi rendszermentési megoldásnak évtizedekkel ezelőtt, de mostanság már sokkal jobb megoldással is rendelkezünk.

A `bin` és `sbin` közti megkülönböztetésének sincs értelme a jelenlegi kontextusban. A történelmi evolúció örült önkényes megkülönböztetésekre vezetett, mint például a `ping` és `traceroute` külön tartalomjegyzékbe helyezése (képtelen vagyok felfogni, miként tarthatja valaki bármiképp is különböző programosztályba tartozónak e kettőt).

Egy Unix-rendszer az engedélyek rendszere. Ha az az óhaj, hogy valamely parancs csak rendszergazdai joggal legyen futtatható, akkor a megoldás: `chmod 700` a megfelelő állományra, és kész. Azt gyanítom, hogy a programok megkülönböztetésének hagyományos rendszerét talán amiatt találták ki, hogy csökkentse a programok számát a normál felhasználók `$PATH`-jában. A mai Linux rendszerekben, lévén hogy akad akár 400 vagy 500 program is a `$PATH`-odban, ennek a megkülönböztetésnek semmi értelme.

Utolsó érvként megemlíthető, mindazonáltal, ami a Linux rendszereket a mai napig is jellemzi: a particionálás és a távoli menedzsment. Ez ugyanannak a dolognak két különböző oldala, és – különösen a távolról menedzselhetőség – a szememben a kritikusaim legjogosabb aggodalma. Az erről szóló vitákra a legfrappánabb érvelés azonban, úgy vélem, az, hogy „a merevlemezek ma olcsók, és a jó teljesítmény érdekében valószínűleg amúgyis helyben lesz telepítve minden programod”. Bár egyetértek ezzel, de szintűgy megértem azokat is, akik szeretnének dolgokat központosítani, adminisztrációs célok érdekében. Ám egy aránylag nem mindennapos feladat érdekében tovább bonyolítani az egész rendszer komplexitását, hát az általában nem igazán jó dolog, sőt, a hagyományos Unix-megoldás még ezesetben sem elég általános: mi van, ha három vagy négy ablakkezelővel rendelkezel? Telepítesz egyet a `/usr`-be, egyet a `/opt`-ba, azután mi lesz? Ott a hagyományos Unix-struktúra. Valójában, a nagyobb Unix-hálózatok többségében amivel kapcsolatba kerültem, a helyi konfigurációk igényelték az Unix-hierarchia nem-standard tartalomjegyzékekkel való kibővítését.

Szerencsére, hála a LiveCD-nek, manapság már a technológiai haladás egy olyan fokát értük el, ami a probléma egy igazi megoldásaként szolgál: ennek neve angolul az „union mount”⁶, ami „overlay filesystem”⁷ néven ismert. Az

⁶ Bocs de nem találtam megfelelő magyar kifejezést rá – a fordító megjegyzése.

⁷ Talán „átlapoló fájlrendszerek”?

ötlet az, hogy több partíciót is felcsatolhatsz ugyanabba a tartalomjegyzékbe. Ezáltal megtartható a **/Programs** azon értelme, hogy ez „a rendszerben elérhető programok összgyűjteménye”, függetlenül azok aktuális fizikai elhelyezkedésétől. A fájlrendszerek is pusztán absztrakciók (nem említünk fájlokat a sávjuk, szektoruk és cylinderük alapján), ez tehát pusztán további haladó előrelépés. Az átlapolt fájlrendszerek nagyon rugalmasak: a rendszeradminisztrátor például helyszínspecifikus beállításokat rögzíthet alapértelmezésként az állományok számára. Sajnos érthetetlen okokból ez nincs általánosan elterjedve. A Plan 9 operációs rendszer alapvető fájlrendszerkezelő műveletei közül az egyik a **bind** parancs (A Plan 9-ben például nincs szükséged **\$PATH** változóra, mert minden tartalomjegyzéket, ami végrehajtható állományokat tartalmaz, egyetlen mappában fognak össze). Az átlapolt fájlrendszer egy Linux alá készült implementációja az „**ovlfs**”.

A hosszabb nevek állítólagos felhasználó-jóakarata

Sok-sok ember, amikor rábukkannak a Gobo-Linuxra, és meglátják a hosszú, leíró-jellegű tartalomjegyzék-neveket, így kiált fel: „Nini! Megváltoztatták a neveket azért, hogy a rendszer barátságosabb legyen!”. Aztán némelyek ezt jónak, mások rossz dolognak tartják. Mindkét oldal téved.

Számos oka van annak, hogy az útvonalnevek a GoboLinuxban olyanok, amilyenek, de ezen okok egyike sem az, hogy „magához csábítson olyan felhasználókat, akiket megijeszt a **/etc**, és hasonló”. Az első számú ok ez: nem ellenkezik az Unix-névtérrel. És amikor Unix-névteret mondok, ténylegesen a Linux névtérre gondolok, ami nem egy nagyon jól megállapodott dolog. Ez nem olyan, mint egy fenntartott kulcsszavakból álló készlet valamely programnyelvből, ami megtiltja például az *if*, *while*, *repeat* változónévként való használatát, a többi viszont oké. Soha nem tudhatod, hogy milyen tartalomjegyzékek, fájlok és programok fognak felbukkanni holnap, tehát úgy gondoltam, a legjobb amit tehetek ha olyan neveket választok, amiket mások valószínűleg soha nem használnak majd. Mások is megtették már ezt előttem, és működött, úgyhogy követtem a példájukat: a NeXT-nek és a Mac OS X-nek el kellett érniük, hogy a saját mappáik Unix-tartalomjegyzékekkel éljenek egymás mellett, úgyhogy „nagybetűsítették” a neveket, és rövidítések helyett teljes szavakat használtak. A rövidítések a régi idők egy jele az Unix eredetének korszakából. Dennis Ritchie egyszer azt mondta, ha vissza tudna menni az időben és csak egy dolgot megváltoztatni az Unix-ban, átkeresztelné a **creat** rendszer-hívást **create**-ra.

Az az egyetlen dolog, ami megnyugtat engem a döntésem igazsága felől, az az, hogy amikor a GoboLinuxszal kezdtünk, úgy nagyjából a Linux 2.4. napjaiban, akkor azt kérdezte valaki, hogy „miért nem a **/sys**-t választottad a **/System** helyett? Hiszen azt könnyebb volna gépelni.” El tudod képzelni, mi történt volna, most, hogy a kernelfejlesztő fickók lefoglalták a **/sys**-t a sajátjuknak?! Valójában állandóan felvetődik a GoboLinux hierarchiáról szóló vitákban a „gépelés-barátság”. Ehhez csak annyit tudok hozzászólni, hogy egy

megfelelően beállított shellben, mint abban is, ami a alapértelmezett a GoboLinuxnál, a `/Programs` begépelése ugyanannyi billentyűleütést igényel, mint a `/usr` bevitele: egy `„/”` jel, kis `„p”` betű, `Tab`.

Egy másik felvetett kérdés: de miért kell a tartalomjegyzék-struktúra megváltoztatásával indítani? Miért nem használjuk egyszerűen a hagyományos tartalomjegyzékfát, és érjük el, hogy „GoboLinuxosan” viselkedjék? Igen, azt gyanítom, hogy ezt nem lett volna lehetetlen elérni, de egy operációs rendszer tervezése szempontjából nem kedvelem az ötletet. Nem tartom kényelmesnek egy olyan rendszer ötletét, ahol jól ismert tartalomjegyzékek más jelentésekkel bírnak azokhoz képest, amikre a legtöbb ember számít. Az AtheOS-nak például megvan ez a problémája. Látsz egy `/usr` mappát, de az nem `/usr`. Az AtheOS-ban ez inkább viselkedik úgy, mint az `/opt`, de a név történelmileg azt jelentette: „user” (= „felhasználó”), később tekintették csak az *Unix System Resources* betűszavának. S még akkor is, ha Kurt Skauen hívta azt `/opt`-nak, ez még mindig furcsa volna; azok nem „opcionális (szabadon választható) csomagok”.

A GoboLinux tartalomjegyzékek szintén rendelkeznek különböző, az Unix-mappákból ismert jelentéssel. A `/Programs` a rendszer számára rendelkezésre álló összes program gyűjteménye, ahol mindegyik alkönyvtár tartalmaz minden, az adott programhoz tartozó fájlt (egy programcsomag megkülönböztetése mindegyik projekt esetében a fejlesztőin áll; a különféle eszközök a *CoreUtils*-ből egyetlen programot alkotnak). Mindegyik alkönyvtár a `/System/Links`-ben egy „lekérdezést” tartalmaz (a kifejezés adatbázis-értelmében) a programgyűjtemény minden egyes fájlosztályáról: könyvtárak, végrehajtható állományok, fejlécek, és így tovább⁸. Mint láthatod, ezek a tartalomjegyzékek nem az Unix-tartalomjegyzékek, adminisztrációs szempontból eltérően működnek. Azt hiszem, hogy jó döntés ezt már a nevekben is világossá tenni.

Szigorúan kompatibilitási okokból mindazonáltal rendelkezünk Unix-nevű szimbolikus linkek egy extra készletével, melyek a legközelebbi GoboLinux megfelelőkre mutatnak (még azon az áron is, hogy némi engedményt teszünk a GoboLinux rendszer felépítésében, elősegítve ezzel a kompatibilitást). A tény, hogy ezek linkek, és mi hagyományos bejegyzéseknek nevezzük őket, ezt az elképzelést nagyon tisztán tartja. *Lucas Villa Real* és *Felipe Damasio* GoboHide-on végzett munkája, a kernel patch mely igazi rejtett tartalomjegyzékeket valósít meg a Linuxon, méginkább az elszigetelt tartozékok szintjére fokozza le a hagyományos tartalomjegyzék-bejegyzéseket.

Meg akarod változtatni a szabványt?

Természetesen nem. Először is, nem vagyunk annyira naivak, hogy azt higgyük, sikerülhetne. De az a tényleges ok, amiért nem akarjuk megváltoz-

⁸ A Plan 9-ben ez megoldható lenne egy `bind` paranccsal; amióta ez nem áll rendelkezésre a vanilla Linux kernelben, szimbolikus kapcsolatokkal tesszük ezt. De el kell ismernem, hogy kedvelem az elképzelést, hogy a lekérdezéseket állandó módon eltárolják.

tatni a szabványt, az az, hogy úgy véljük, *nem kellene szabványnak lennie*. Tudom, hogy ez a nyilatkozat még merészebbnek hangozhat az arról való beszédnél, hogy megváltoztatnánk egy szabványt, de a lényege annak amit mondok, hogy azt hisszük, hogy mindegyik alkalmazásnak kötelessége megengedni azt, hogy bárhová is telepítsék, és együttműködni minden neki szükséges másik alkalmazással, bárhová is vannak azok telepítve (többet erről a következő pontban). Ha volna most egy szabvány, ami kijelenti ezt, aláírnák egy kérvényt, ami támogatja azt. Valójában létezik is ilyen: ugyanis ennek felelnek meg a GNU által kibocsátott szabványok, melyek a GNU Autotools használatát ajánlják, a kapcsolók **-prefix** családjának támogatásával, felderítendő a configure szkript segítségével az alkalmazások helyét. De azon csak nevetek, amikor egy javasolt szabvány mint az FHS, közli velem a binárisok egy önkényes listáját, hogy azokat nekem meg nem nevezett okok miatt különálló tartalomjegyzékben kellene tartanom.

A különböző helyzetek különböző szükségleteket hoznak létre, és olyan úgynevezett szabványok, amik minden lábat ugyanabba a cipőbe próbálnak beilleszteni, eleve kudarcra vannak ítélve. Ehelyett a rugalmasság a szabványosítandó. Vagyis nem, mi nem javasoljuk, hogy a GoboLinux hierarchiát bárki is szabványként kövesse. Öt, tíz vagy húsz év múlva a maitól teljesen különböző szükségünk lehetnek. Nem akarom azt, hogy a GoboLinux fájlrendszerrel való akkori esetleges elszakadás olyan nehéz legyen, mint a hagyományos Unix-fájlrendszerrel való mostani eltávolodás. Mely elvezet minket természetesen a következő ponthoz.

Fáradtságos küzdelem, hogy megváltoztassanak minden alkalmazást

Nem olyan nehéz ez, mint amilyennek látszik. Mielőtt a GoboLinux első verzióját teljesen megépítettük volna, már egy egész éven át dolgoztam rajta, és javítottam azt. Amikor André Detsch és én végül nekikezdtünk és felépítettük az alapjaiból két nap alatt, belészerkesztve az elképzeléseinket, már tudtam, hogy ez tökéletesen megvalósítható.

Egyetemi környezetben dolgozom sok év óta. Ott nem én vagyok a rendszergazda, úgyhogy minden extra alkalmazást amire szükségem van, a saját **\$HOME** könyvtáramba kell telepítenem. Ez teljesen hétköznapi állapot, azaz várható, hogy bármilyen tisztességes alkalmazás meg fogja engedni ezt, és hatalmas többségük így is viselkedett. Valójában, egy erre képtelen alkalmazás tönkretetheti a rendszeredet. Ha telepíteni tudod a Gimpet a **/usr**, a **/opt**, vagy a **/home/hisham** könyvtárakba, akkor képes vagy azt a **/Programs/Gimp/2.0/**-ba is telepíteni. A tapasztalat megmutatta, hogy nagyon kevés alkalmazásnak van arra szüksége, hogy az együttműködés kedvéért átírkáljuk a Makefile-ját. Még rendszergazda-központú szoftver is rendelkezik (vagy kellene hogy rendelkezze) ezzel a rugalmassággal: egy szabályos Unix-rendszerben a rendszergazdának kéne hogy legyen lehetősége arra, hogy válasszon, mondjuk, a **/sbin** és a **/usr/sbin** között. Nincs oka annak, hogy a programokban és telepítő-

szoftverekben holmi „fixen behuzalozott útvonalak” legyenek⁹.

Érdekesebb az a probléma mely akkor merül fel, amikor egy program bár megengedi, hogy bármely altartalomjegyzékbe telepítsék, ám ugyanakkor mérőben helytelenül feltételezi, hogy egy másik program amitől ő függ, ugyanoda van telepítve, ahová ő maga. Amint azt sejtethed, ez a GoboLinuxra nehezedő problémák legfőbb forrása, de közbevetőleg megjegyezném, hogy ez az egész szabad szoftverközösség érdekében megoldandó feladat! Térjünk vissza azonban a `$HOME` tartalomjegyzékhez! Mi van, ha a kedvenc GNOME összetevőmet nem telepítette a rendszergazda, és emiatt telepíteni akarom azt a `$HOME`-omba, miközben a GNOME többi használatban levő része a `/usr`-be van telepítve? Az ehhez hasonló helyzetek, különösen a sok összetevőből álló szoftverekben gyakorta problematikusak. Ezekben sok program egy `$PATH`-hoz hasonló környezeti változó használatával oldja meg a problémát: `$GTKPATH`, `$PERL5LIB`, `$KDEDIRS`, `$PYTHON_PATH`, és így tovább. Azaz nincs indoka a monolitikus telepítésnek.

Tehát a csata, amit a GoboLinux a telepítési útvonalakkal kapcsolatban vív, nem csupán ránk jellemző; mi csupán felfedjük az alkalmazások rugalmasságában (vagy rugalmatlanságában...) rejlő problémákat, mely nemcsak a mi tartalomjegyzék-struktúránkban, de bárhol előbukkanhat, ahol egy felhasználónak valami egyéni telepítésre lehet szüksége. Úgy látom különben, hogy a helyzet az elmúlt néhány évben jelentősen javult, amint egyre több project használja a GNU Autotools-t.

Könnyebb lefordítani egy programot relatíven ugyanahhoz a könyvtárszerkezethez alkalmazkodva

Biztos így van. Ez egy olyan pont, ami néha-néha előbukkan a GoboLinux levelezőlistán, amikor az emberek azt javasolják nekünk, hogy tervezzük át a `/System/Links`-et egy hagyományos Unix-hierarchia szerinti alkönyvtárakkal, mint például a `bin`, `lib`, `stb.`; - vagy éppen fordítsunk mindent a `/usr`-hez viszonyítva, hogy biztosítsuk a hagyományos tartalomjegyzék-struktúra állandó működőképességét. Az ezt javasló emberek implicite e két megoldás egyikét javasolják nekünk: lefordítani egy könyvtárstruktúrához viszonyítva relatívan, azután telepíteni relatívan egy másikba; lefordítani egy könyvtárstruktúrához viszonyítva relatívan, azután a telepítés során használjunk átirányításokat. Nem kedvelem a két megközelítés egyikét sem. Az elsőben bizonyos rugalmasságra számíthat az telepítőrendszer részéről, amivel pedig az nem mindig rendelkezik, épp csak másfajta, mint amely pontokat az előző szekcióban felemlítettem. Márpedig nem igazolható, hogy az alkalmazás telepítőrendszere okvetlenül kell rendelkezzezzék ezzel a rugalmassággal¹⁰. A második esetben pe-

⁹ Egy önmagában való telepítőprogram ritka fogalom Linuxban; az, amire a telepítőprogram kapcsán ehelyütt gondolok, az általában egy Makefile által végrehajtott install.

¹⁰ Azok kedvéért, akik érdeklődnek eziránt a megközelítés iránt, felhívom a figyelmet a GNU Stow-ra. Nem tudom mindazonáltal, hogy bármilyen rendszer alatt működik-e 100%-osan.

dig, hát nem kedvelem egy olyan operációs rendszer ötletét, amit olyan módszer köré építettek, amit bármelyik pillanatban megkerülhet egy új rendszerhívás, vagy valami szokatlan hozzáférési mód. Lehet, hogy néhányan azt mondják, a GoboHide maga is beleesik a „szokatlan hozzáférési módok” tartományába. Ezenkívül rá szeretnék mutatni arra, hogy a GoboHide használata nálunk nem elengedhetetlenül szükséges, másrészt úgy terveztük, hogy szabályosan együttműködjék a vanilla Linux kernellel¹¹.

De ahelyett, hogy a javasolt alternatívák hibáit pécézzem ki, jobban szeretném alkotó módon megvédeni a tervvel kapcsolatos az eredeti döntéseimet. A GoboLinuxszal kapcsolatos eszménk, hogy elkülönített tartalomjegyzékekkel gyakoroljuk ezt az új megközelítést. Felbecsültük a hatását a rendszermenedzsmentre, és izgalmas eredményeket gyűjtöttünk. Ha helyette arra használtunk volna minden lehetséges módszert, hogy „megkönnyítsük az alkalmazások lefordítását”, azt hiszem, hogy csak távolabb kerültünk volna a céltól. Amikor lefuttatom a

```
ls -l /System/Links/Executables
```

parancsot, és minden végrehajtható állományt látok amit a rendszerem tartalmaz, nos, akkor egy tiszta képpel rendelkezem a rendszerről. Utálnék a /System/Links-re nézni (vagy hívják akárhogy), ha azon belül is csak az Unix-féle rendetlenséget látnám emulálva, holmi /bin, /sbin, és /usr/X11R6 által.

Windows-szerűvé akarod változtatni a Linuxot!

Ha mindent elolvastál eddig a pontig, azt hiszem, elég világos kell legyen számodra, hogy nem ez a célunk. Ha az lenne a célunk, hogy magunkhoz csábítsuk a Windows-felhasználókat, az esetben egy szerkezeti átszervezés volna az az utolsó dolog, amit cselekednénk. Ehelyett arra koncentrálnánk, hogy a felhasználói interfész úgy nézzen ki, mint a Windows, miközben Windowshoz hasonló témákat alkalmaz, ikonokat rángatva eközben, talán némiképp a Wine-ot integrálva a disztribúcióba, és így tovább. És léteznek is ilyen disztrók manapság, ezek a Lindows, Lindash, Linspire, vagy bármi is a nevük – de mi nem tartozunk közéjük.

Rendkívül paradoxnak hangozhat, de arra törekszünk, hogy megtartsuk a rendszer Linux-identitását. Hogy pontosabbak legyünk, küzdünk mindegyik projekt önazonosságáért. Amikor csak lehetséges, változatlan forrásokkal szállítunk minden alkalmazást. Ha vetted valaha a fáradságot arra, hogy belenézz bármely jelentősebb disztribúció fájljaiba (Pld `.src.rpm`), akkor tudod, hogy miről beszélek: a csomagok hatalmas többsége patch-okat tartalmaz, az alkalmazások viselkedésének csekély módosítása érdekében: legyen ez akár az, hogy megváltoztassák egy kapcsolónégyzet alapértelmezett beállítását, vagy akár az is, hogy eltávolítsák egy alkalmazás „About” dobozát! Mi nem tesszük ezt. A mi K menünk a KDE logót jeleníti meg, miként annak a KDE-

¹¹ Azt is megtehetjük, hogy a /System/Links-et union-ként mountoljuk fel, ovfls segítségével.

ben lennie kell. Bár van egy témánk amit egyedi tapétával szállítunk, de azt opcióként mutatják be a telepítőprogramban.

Komoly erőfeszítésekre is képesek vagyunk annak érdekében, hogy csomagjaink ne tartalmazzanak GoboLinux-specifikus hibákat. A legrosszabb amit egy Linux felhasználó felfedezhet, hogy egy adott szoftver az X disztrón dolgozik, ám nem működik egy másik Y disztrón, és nem tudja, hogy vajon amiatt, mert az Y disztró valami olyan patchet vezetett be, ami oka a hibának, avagy mert az X disztró vezetett be egy olyan patchet, ami kijavította a hibát. Ez fejlesztőként is jelentősen megnehezíti a dolgunkat. Alexandre Julliard a Wine fejlesztői közül egyszer azt mondta, hogy a Linux disztribúciók állandó változásai jobban lelassítják a projectet, mint a Win32 API változásai.

Az a tény, hogy a GoboLinuxon minden Unix library-könyvtár bemutat minden libraryt a rendszerből, minden fejléc-könyvtár bemutat minden fejléct és így tovább, a disztribúciók közt fellépő számos kompatibilitási problémának kihúzza a méregfogát, miközben azt az ironikus helyzetet eredményezi, hogy a szokatlan tartalomjegyzék-struktúra ellenére a legkompatibilisebb disztribúciók egyike lettünk.

Egy disztribúció vizsgálata

Néhány ember, akiket talán felizgatott a tény, hogy oly „nagy módosítást” hajtottunk végre az operációs rendszer szerkezetében, alkalmanként azzal fordul hozzánk a levelezési listában, hogy bizonyos más nagy rendszer-méretű változtatások figyelemre méltóan javítanák a GoboLinuxot. Néha a nagy ötlet alkalmazható, és alkalmazzuk is, mint például amikor Carlo Calica integrált egy démon-menedzselő eszközt, a Runit-ot a GoboLinuxba¹². Ám az esetek legnagyobb részében az ötlet olyasmi, ami megkövetelné minden alkalmazástól, hogy agyonmódosítsák, ha ugyan nem kéne akár teljesen át is írni azt. Ezt nyilvánvalóan nem tudjuk, és nem is akarjuk megtenni. Ha korlátozott számú programról lenne szó, lehet, hogy néhány efféle ötlet megvalósítható, de az embereknek észben kell tartaniuk, hogy a GoboLinuxszal használható programok világegyeteme potenciálisan végtelen, ahogy napról-napra új Linux alkalmazásokat írnak.

Itt csak néhányat tudok megemlíteni a javasolt, ám megvalósíthatatlan ötletek közül:

- áthelyezhetővé változtatni minden programot - szeretném ha ez megvalósulna, ám megcsinálni elég durva: átírni minden alkalmazást a világban, hogy libprefix-et használjon, tároljon minden függőséget mindegyik programkönyvtárban, és akár százszor is meglegyen a lemezen ugyanaz a néhány könyvtár (és még így is meg kell alkudni azokkal az összeférhetetlenségekkel, amik felmerülnek egy rendszerben, ha az nem felkészült minderre), mindent lefordítani ugyanahhoz a hierarchiához viszonyítva (fent egy egész szekciót szenteltem e kérdés-

¹² Az aktuális tendencia a GoboLinux fejlesztésben ennek az integrációnak a lazítása, opcionális Runit összetevő készítésével, ám megtartva a szükséges szerkezetet a működéshez.

körnek).

- elérni, hogy minden program egy egységesített konfigurációfájl-formátumot használjon - Természetesen az erőfeszítés, amit egy efféle átdolgozás igényelne, még nagyobb, mint az az előző javaslatnál szükségeltetne. Minden ötlet, ami úgy kezdődik, hogy „minden program...”, ugyanabba a közös problémába torkollik: abba, hogy a rááldozandó erőfeszítés potenciálisan végtelen; nem tudod ugyanis elérni, hogy a világ minden prozdsektje beleegyezzen a javaslatod használatába; de ha el is tudnád ezt érni, még akkor is kimaradna ebből számos hagyományos rendszer, amelyek akkor sem tudnának megváltozni ha akarnának, annyira inkompatibilisek. Egy másik elképzelés mely ugyan ebbe a csoportba tartozik, hogy minden program ugyanazt a grafikai eszközkészletet használja. Igen, ez valóban jó lenne, és egységesítené a Linux desktopot, de épp ez az, ami nem fog megtörténni.
- változtasd meg a `/Programs` nevét `AppDir` -rá, mert így nagyon hasonlítasz a RiscOS -ra – Mondhatják hogy ahhoz hasonlítunk, de az azért mégis jócskán különbözik tőlünk. Igen, a `/Programs/Emacs` mondjuk elhelyezhető volna holmi `AppDir` -ben. A `/Programs/LyX` szintén. Ó, sok más is. De mit mondasz a `/Programs/FindUtils` -ról? Vagy a `/Programs/KDE`, mi történne, amikor arra kattintasz? Egy másik fontos vitapont, hogy az `AppDir`-ek természetsszerűleg áthelyezhetőek, a GoboLinux csomagok pedig nem, de ezt a témát fentebb már taglaltam.

Internacionalizálás

Gyakran felemlégetik, hogy a nevek kicserélése, például a `lib` -nek `Libraries` -re változtatása túl angol-centrikus („a régi nevek legalább mindenkinek egyformán értelmetlenek”), és vennünk kéne a fáradságot rá, hogy lefordíthatóvá tegyük a tartalomjegyzék-struktúrát. Elhessegethetném ezt a pontot, mint ami sok műszaki kérdést vet fel, használhatatlanná is téve ezáltal, hacsak nem szimbolikus linkekről és GoboHide patchekről beszélünk. De nem fogom ezt tenni. Helyette megkérdem, hogy ha lenne egy tiszta és elegáns mód, hogy lefordítsák minden GoboLinux-tartalomjegyzék nevét, nos, mi változna meg ettől?

Ha az emberek annak érdekében fordítják le a tartalomjegyzék-struktúrát, hogy barátságosabbá tegyék a rendszert az angolul nem beszélők érdekében, hát sajnálom, de az nem fog segíteni. Egy olyan felhasználó, akit legyőz a tény, hogy a `/System/Settings` -et nem úgy nevezik, hogy `/Rendszer/Beállítások`, nem fog nagyon előrébb lenni, amint eljön a pillanat, hogy használnia is kell ezeket a tartalomjegyzékeket, és szerkesztenie kell a `httpd.conf`-ot. A lényeg, amit szeretnék megértetni, az az, hogy annak a fajta felhasználónak, akinek internacionalizálásra van szüksége, nem fog segíteni egy lefordított tartalomjegyzék-struktúra. Az erőfeszítéseket a dokumentációk fordítására, és a programok felhasználói interfészének lefordítására kell koncentrálni ehelyett. Ha a felhasználó el tud olvasni egy kézikönyvet ilyen vagy olyan nyelven, ami azt

mondja neki, hogy menjen a `/System/Settings`-be és végezzen el a `httpd.conf`-ban ilyen és ilyen változtatást, ez sokkal hasznosabb, mint a tartalomjegyzék nevének megváltoztatása. És ha a felhasználónak van egy barátságos GUI-ja mondjuk az Apache beállításához, hasonló ahhoz amit a Mac OS X szolgáltat, azt a felhasználó valószínűleg sokkal inkább kedveli majd.

Integráció más disztrókkal

Ez egy másik kérdés, amit különböző alakokban, méretekben és színekben vetnek fel néha-néha. Egyetlen okát látom annak, amiért felmerül ez az igény, és ez az, hogy más disztribúcióknak hatalmas készletük van használatra azonnal kész szoftverek gyűjteményéből. Első látásra, az ötlet hogy egyesítsük a GoboLinux újításai közül mindegyiket valamely „X” disztró óriási csomagválasztékával, bámulatosnak látszik. Ha azonban közelebbről is szemügyre vesszük, látni fogjuk, hogy nem az.

Először is ott van a függőségrendszerek kérdése. A GoboLinuxnak nagyon laza függőségrendszere van, amit eleve úgy terveztek, hogy rugalmas legyen, a felhasználó által testreszabható¹³. Ha a GoboLinux ezen előnyére építész, nem lesz elrészorolva az „X” disztró automatikus frissítési szolgáltatásaira, és ez fordítva is igaz. Azaz, neked választanod kellene, hogy a GoboLinuxot úgy használod, mintha az egy „X” disztró volna, feladva a GoboLinux rugalmasság nagy részét, vagy figyelmen kívül hagyod az „X” disztró remek automatikus frissítési szolgáltatásait. Akármelyiket választod is, feladnád azon okok egyikét, amiért egyáltalán leginkább elindítottad ezt az integrációs projectet.

És akkor aztán állandóan foglalkozhatnál mindkét disztribúció apró sajátosságaival is: különböző indítószkriptek, a libraryk lehetséges inkompatibilitásai, az „X” disztró állítólagos „értéknövelő” csomagmódosításai... s nem is említettük az arra való alkalmatlanságot, hogy megfelelően használják a Compile-t vagy a GoboLinux binárisok gyűjteményét, amiatt, mert például a csomagok különböző elnevezési konvenciókat használnak.

Röviden, még akkor is, ha egy egész rendszert átalakítasz az „X” disztró csomagjainak használatára, az, amivel végződni fog, nem egy „felturbózott” GoboLinux, hanem egy köntörfalazó disztró. Egyszerű megtenni, hogy vesszünk minden RPM csomagot, ami például egy RedHat rendszert alkot, kicsomagoljuk őket, és besymlinkeljük őket, hogy úgy nézzenek ki, mint egy GoboLinux rendszer. Biztos lehetsz afelől, hogy a kapott rendszer végülis sokat csinosítana a RedHat-on. Meg is valósították ezt különböző emberek különböző célokkal (néhányek építettek egy teljes disztrót, mások csak egy vagy két bináris csomagot konvertáltak), RedHattal, Slackware-ral, Debiánnal, és legtöbbször a Gentoo-val. A megfigyelésekből leszűrte általános tapasztalatom az, hogy ez nem éri meg.

¹³ A GoboLinux függőségrendszer leírása túlmutat e cikk keretén, de például, ha telepítész egy A alkalmazást ami a B-től függ, ez nem fog panaszkodni ha B nincs felinstallálva egy vanilla csomaggal, és ha B hiányzik, erről értesítést küld, de nem fogja megakadályozni az installációt.

A rendszergazda

Természetesen megjegyeztem a legjobbaknak tartottakat. A döntés, hogy a nullás felhasználót nevezzük valami másnak, mint a **root**, az egyik azok között, amiért minket a leginkább kritizálnak. Ennek az eredete visszanyúlik a GoboLinux keletkezése tájára. A kísérleti rendszeremen a normál felhasználóm neve **hisham** volt, a superuser pedig **lode**. Soha nem kedveltem az Unix azon elképzelését, hogy „egy mindenható root szemben a normál felhasználókkal”, és látni akartam, hogy egy Linux rendszer milyen jól viselkedne gyökérfelhasználó nélkül. Néhány átdolgozás után itt és ott, ez nagyon jól működött. Jó volt tudni, hogy ha valaki bármikor is megpróbálna root-ként bejelentkezni a gépembe, mindig kudarcot vallana.

Amikor végrehajtottuk azt a „hack-sorozatot”, ami aztán a Gobo-Linux első verziójával végződött, André és én úgy döntöttünk, hogy állandósítjuk ezt. A **gobo**-t választottunk, ami egy belső vicc. A szándék természetesen az, legyen egy olyan rendszer, ami tisztán támogat egy nem-root superusert, de a felhasználók (egy maroknyi ember még csak akkor) soha nem változtatott az alapértelmezésen, és így aztán a gobo valahogy beragadt. Azért még mindig lehetséges különösebb erőfeszítés nélkül megváltoztatni az alapértelmezést. Rövid időn keresztül adminisztráltam egy sereg gépet az egyetemenél, és ott, hogy könnyebben illeszkedjenek be a NIS környezetbe (a hálózat alapvetően RedHat egységekből állt), visszaváltottam a superusert gobo-ról root-tá. Most hogy a GoboLinuxnak van egy grafikus telepítőprogramja, az üzembehelyezés ideje alatt opcióként dönthetünk a superuser neve felől.

Most hogy torkig vagyok már a történelmi magyarázattal, egy dologra szeretnék rámutatni még, s ez az a jól ismert tény, hogy ez az egyetlen istenhez hasonló entitás létezése a Linux biztonsági modell gyengeségei közül az egyik, és ez az, ami nyugtalanított engem az „egy mindenható root kontra egyéb felhasználók” elképzelésben; ez hasonló az elosztott rendszerek egyetlen hibás pontjához. Minden projektnek, aminek az a célja, hogy javítsa a Linux biztonságát, legelsősorbanis növelnie kell a biztonsági modell szemcsézettségét, hígítva a root erejét: ACL-ek, képességek, SELinux... Lehet vitázni róla, hogy ezek közül néhány túlzott bonyolultságot ad hozzá a modellhez, de nem fogok belemászni ebbe a beszélgetésbe itt. Az ugyanis teljesen bizonyos, hogy a root-modell túlságosan leegyszerűsített a mai összetett rendszereknél, és az, hogy a „setuid” jogosultság a legtöbb biztonsági kérdés forrása. A Plan 9-nek például egyáltalán nincs is superuserje; ehelyett felajánlja az állományrendszer egy virtualizált nézetét mindegyik eljárásnak. A gobo kísérlet volt annak lemérésére is, hogy mennyire ivódott bele a Linux világba az elvárás egy root-felhasználó létevel kapcsolatban; szerencsére, nem igazán (ez nem számít természetesen abból a szempontból, hogy a biztonsági modell a #0 felhasználóra csatolt). Egy jövőbeli irány, ami felé szeretném, hogy a GoboLinux tartson (és valójában a Linux általában), az az, hogy át kell vennie a felsorolt technológiák közül néhányat, javítandó az irányítást a

rendszerbiztonság és az adminisztráció fölött; a root-tól való leválás volt az első lépés ebben az irányban.

Következtetések

Nos azt hiszem, sok kérdést érintettem ezzel a cikkel. Biztos vagyok abban, hogy sok kérdést felejtettem el, de azt gondolom, a legfontosabbak azért mind itt vannak. Legfőképp abban reménykedem, sikerült megértetnem, hogy a GoboLinux nem csupán a fájlrendszer esztétikusabbá kozmetikázása. Mi eléggé tudatában vagyunk annak, amit teszünk, és cselekedeteink következményeinek is.

Nem titok, hogy amikor elkészültem ennek a tartalomjegyzék-szerkezetnek az első verzióival, nem számítottam arra, hogy egy Linux disztribúcióvá válik, amit a világ minden táján használnak majd az emberek (bár elég korán disztribúciós projectté formálták, amint Guilherme Bedin csatlakozott). Az egyetlen dolog, ami boldoggá tett akkoriban, amikor ez még nem volt egy megfelelő disztró, az nem más volt, mint egy tiszta terv.

Teljes szívvel egyetértek az idézettel a cikk kezdetében, és határozottan azt hiszem, hogy ez nem az az eset.

Utóirat: **Sajtóhibák**

Köszönet Varga Péternek, aki felhívta rá a figyelmemet, hogy a híres „creat()” idézet Ken Thompsontól és nem Dennis Ritchietől való.

E dokumentumról...

Nem vagyok tanácstalan, *avagy* Mítoszok és félreértések a GoboLinux tervezésével kapcsolatban

E dokumentumot a LaTeX2HTML fordító 2002 (1.62) verziójával hozták létre. Copyright © 1993, 1994, 1995, 1996, Nikos Drakos, Computer Based Learning Unit, University of Leeds.

Copyright © 1997, 1998, 1999, Ross Moore, Mathematics Department, Macquarie University, Sydney.

A parancssori argumentumok:

latex2html -split 0 clueless.tex

A fordítást 2004.06.13-ban kezdeményezte Hisham Hashem Muhammad

Hisham Hashem Muhammad 2004-06-13

A regisztrált GoboLinux felhasználók

Ha eddig elolvastad e könyvet, minden bizonnyal komolyan érdeklődöl a GoboLinux iránt. Ideje elárulni, hogy van vezetve egy lista az úgynevezett „regisztrált” GoboLinux felhasználókról. E lista e linken tekinthető meg:

http://gobolinux.org/index.php?page=hall_of_fame

Természetesen ennél többen használnak a világon GoboLinuxot – e listára azok kerülnek fel, akik kifejezetten kérik az oda való felkerülést. Ez nyilván az „átlagosnál” komolyabb elkötelezettséget jelent e disztribúció iránt. A listában a sorszámok binárisan vannak megadva. Az enyém, e könyv írójáé a 86-os.

GoboLinux, haladóknak

Ebben a fejezetben különböző speciális GoboLinux-parancsok, szkriptek, config fájlok leírását adom meg, bármiféle előzetes elgondolás nélkül, mindössze nagyjából ABC sorrendben. (Azért nagyjából, mert az általam fontosabbnak tartottak előbbre kerülnek. Elismerem, ez némileg szubjektív vélemény). Ezek a leírások részben az eredeti angol GoboWiki megfelelő szócikkeinek általam „elkövetett” fordításai, másrészt akad olyan is, amit én magam írtam. Nem adom meg viszont a GoboHide leírását, mert azt korábban már közöltem. Ez a fejezet tehát lényegében a GoboLinux egyfajta nem hivatalos és nem is teljes, de mégis egyfajta tankönyve, referenciája, s jelen pillanatban a legteljesebb mégiscsak, mely magyar nyelven elérhető. Megjegyzem azonban, az itt szereplő parancsok nem mindegyikét próbáltam ki, nem lévén eddig szükségem mindegyikre, azaz ezekkel kapcsolatban még annyi felelősséget sem vállalom, mint az eddigi leírásokért! (Hiába, teljesen felelőtlen fickó vagyok...)

AddUser

Új felhasználói accountot készít, és bejegyzi azt a rendszerbe. E szkript segítségével egyéb személyre szabott beállítások is eszközölhetők a felhasználó home-könyvtárára vonatkozóan.

Használat: **AddUser** [**<opciók>**] **<login>**

Opciók:

- -h, --help - Megjeleníti ezt a képernyőt.
- -v, --version - Kiírja a program verzióját.
- -V, --verbose - Bőbeszédű üzemmód engedélyezése.
- -c, --comment <entry> - Szöveg eltárolása az UNIX jelszó-fájl megjegyzésmezőjében.
- -f, --full-name <entry> - A felhasználó teljes neve.
- -r, --room-number <entry> - A felhasználó sorszáma.
- -w, --work-phone <entry> - A felhasználó munkahelyi telefonszáma.
- -h, --home-phone <entry> - A felhasználó otthoni telefonszáma.
- -o, --other <entry> - Egyéb.
- -s, --shell <entry> - A felhasználó alapértelmezett parancsértelmezője (shellje). Az alapértelmezett beállítás a '/bin/zsh'.
- -p, --password <entry> - A felhasználó jelszava.
- -S, --skel - Mintakönyvtár, aminek alapján elkészül a felhasználó home hierarchiája.

A felhasználóra vonatkozó információk, mint például a teljes név, megtekinthetők olyan programokkal mint pld a finger(1) és módosíthatóak a chfn(1)

-el.

Példa:

```
AddUser -f 'Hisham Muhammad' hisham
```

Compile

A Compile egy automatizált programfordító eszköz.

Használat: **Compile** <program> [<verzió>]

Opciók:

- -h, --help – Megjeleníti ezt az üzenetet.
- -v, --version – Kiírja a program verzióját.
- -V, --verbose – Bőbeszédű üzemmód engedélyezése.
- -n, --app-name <entry> - Az alkalmazás nevének felülbírálnása.
- -e, --version-number <entry> - A verziószám felülbírálnása.
- -x, --cross-compile <entry> - Keresztfordítás más architektúrára.
- -c, --configure-options <entry> - Opciók közvetlen megadása a ./configure számára.
- -k, --keep – A programkönyvtárban meglevő fájlok megőrzése
- -b, --batch – Kötegelt üzemmód: nem tesz fel kérdéseket.
- -B, --no-build – Nem építi be a forrásokat, csak letölti őket.
- -I, --no-install – Nem installálja a programot.
- -W, --no-web – Nem vizsgálja meg a nem helyi recipe-oldalt, és nem tölt le archívumokból.
- -C, --no-configure – Nem futtatja a ./configure szkriptet.

A Compile forrásalapú csomagkezelő program. Már láthattál hasonlót: ez hasonlít egy kissé a BSD-portokra, illetve a jólismert Linux-megfelelőre, a Gentoo Linux Portage-jára (emerge). A forrásból való telepítést néha tréfásan a „szegény ember csomagbázisának” is nevezzük. Nem azért, mert nem jellemző rá a változatgazdagság (bár ez természetesen igaz), hanem mert ez csak minimális tervezést igényel.

A Compile használata

Program telepítése Compile segítségével

Ebben a példában az irssi-t, egy szöveg alapú IRC-klienset fogunk telepíteni (Internet Relay Chat). Hogy ezt lefordítsd és telepítsd, géped be:

```
Compile irssi
```

Erre le kell töltsd azt a recipe állományt, melyben le van írva az Irssi fordításának és a telepítésének menete. Az ételek receptjeihez hasonlóan, egy recipe egy olyan utasítássorozat, ami elmondja a Compile-nak, hogy hogyan lehet főzni (fordítás) és felszolgálni (telepítés) egy csomagot. A számítógéped sebességétől, internetkapcsolattól és attól függően, milyen csomagokat installáltál, az Irssi néhány perc alatt letöltődik és telepítettik. Kitűnő, nemde?

Parancssori opciók

A Compile-nek van sok más parancssori opciója is, amiket részletesen a referencia-leírásánál tárgyalunk. Néhányukat maga a Compile használja, mint például `--app-name`, meta-recipeknek, és te valószínűleg úgysem használnád közvetlenül. De vannak opciók, amiknek az a céljuk, hogy segítségükkel irányíthasd a Compile-ot, hogy az mely tevékenységet hajtson végre.

A `--no-web` opció különösen hasznos, ha nincs állandó internetkapcsolatod. Ezzel elkerülhető, hogy megpróbálja futtatni a `wget` -et a források letöltése céljából (és így elkerülhető az időtúllépés miatti késedelem), vagyis így csak a helyben tárolt fájlokat nézi át a szükséges állományok megkeresése érdekében.

A `--no-build` egy másik hasznos opció. Ezzel az opcióval meghívva a Compile-ot, ténylegesen nem fordít le semmit. Arra tudod ezt használni, hogy letöltsd a forrásokat, egy későbbi időpontban `--no-web` opcióval történő fordítás számára. Ez hasznos, például, dial-up felhasználók esetében, akik így az internetkapcsolatuk idején le tudnak tölteni mindent ami szükséges, azután a kapcsolat nélküli idő alatt fordítanak, vagy azoknak, akik nem akarnak a fordítás miatt nagy CPU terhelést, amíg dolgoznak a gépen, és jobban szeretnék a fordítást éjjel elvégezni, amíg alszanak.

A `--no-configure` opció megszakított `make` folytatásához hasznos, hogy elkerülje olyan fejlécfájlok újragenerálását, amik ismétlődésük miatt mindenféle bajokat okozhatnak.

FIGYELMEZTETÉS: A hibák eliminálása céljából újrafuttatni a Compile-ot a `-c` kapcsolóval a legbiztosabb módja annak, hogy érvénytelen recipét hozz létre, minthogy a recipe a legközelebbi futásánál nem alkalmazza a te kézi beállításaidat. Ha a gobolinux.org recipetárolója számára készítesz elő egy recipét, győződj meg róla, hogy az képes lefutni egy tisztán kicsomagolt forráson, a `-c` kapcsoló használata nélkül.

Compile.conf

A Compile.conf az az állomány, ahol be tudod állítani a Compile által használt különböző útvonalakat (path) és URL-eket (internetcímeket).

Ezek a /Programs/Compile/Settings/Compile/Compile.conf -ban vannak tárolva – mely be van linkelve a /System/Settings/Compile/Compile.conf alá (ha használod a GoboLinux fájlrendszert, akkor mostanra tudnod kell, hogy ez ugyanaz, mint az /etc/Compile/Compile.conf).

Ez a fájl szokásos tartalma:

Itt a neved, mely beleíródik az általad készített recipékbe:

```
compileRecipeAuthor="Remek Elek <remekelek@gmail.com>"
```

Ez csak példa! A név megváltoztatandó (hacsak természetesen nem te vagy Remek Elek ;)) A nevet és az e-mailt kell megadni mint látható, szigorúan a fenti formában, amiatt, hogy ezt értelmezni tudja egy másik szkript, a ContributeRecipe.

A Compile fájljainak szokásos helyei a te rendszeredben:

```
compileDir="{goboPrefix}/Files/Compile"
compileDependenciesDir="$compileDir/Recipes"
compileArchivesDir="$compileDir/Archives"
compileSourcesDir="$compileDir/Sources"
```

Itt soroljuk fel a végigkeresendő, kicsomagolt recipéket tartalmazó helyi könyvtárakat:

```
compileRecipeDirs=(
"$compileDir/LocalRecipes"
"$compileDir/Recipes"
)
```

Ezek azon internetcímek, ahonnt a recipéket letölteni igyekszik:

```
getRecipeStores=(
"http://gobolinux.org/recipe-store"
"http://gobo.calica.com/recipe-store"
)
```

A következő változó azt mutatja meg, hogy a GetRecipe hol helyezi el a recipéit. Ideális esetben ennek illeszkednie kell a \$compileRecipeDirs változóhoz:

```
getRecipeRecipeDir="$compileDir/Recipes"
```

Ez határozza meg a GenRecipeStore forrás- és célkönyvtárát:

```
genRecipeRecipeDir="$compileDir/LocalRecipes"
```

genRecipeStoreDir="\$compileDir/Store"

S íme néhány a főbb szabadszoftver-lelőhelyek (repositories) közül: a recipék ezen változókat használják az ő url deklarációikban, azért, hogy a kedvenc tükrözéseidet használhasd anélkül, hogy minden egyes recipét külön-külön szerkesztgetned kellene:

**ftpGnu=ftp://ftp.gnu.org/gnu/
ftpAlphaGnu=ftp://alpha.gnu.org/gnu/
httpSourceforge=http://unc.dl.sourceforge.net/sourceforge/**

CompileProgram

A CompileProgram egy wrapper a programok kézi fordításához. Három részből áll: „configure; make; make install”, automatikusan detektálva a paramétereket, így a program a GoboLinux-fába kerül.

Használat: **CompileProgram** <options> <archive-file> [<package-name> [<version>]]
[-- <extra-options>]

Opciók:

- -h, --help - Kiírja ezt a helpet.
- -v, --version - Kiírja a program verzióját.
- -V, --verbose - Bőbeszédű mód.
- -b, --batch - Batch mód: nem mutatja a dokumentációt és nem vár választ a jóváhagyásokra
- -k, --keep - Megtartja a létező könyvtárstruktúrát a '/Programs/<package-name>/Current'-ben.
- -r, --remove - Törli a '/Programs/<package-name>/Current' könyvtárstruktúráját a fordításhoz.
- -m, --make-target <entry> - Meghatározzuk a 'make' parancs „célját” (target).

Bármely egyéb opció '--' után közvetlenül átadódik a 'configure' szkriptnek (ha használva lesz).

Példa:

CompileProgram -m linux-i386 weirdsuperlib-1.4k.tar.gz WeirdSuperLib 1.4k -- --with-shared

A hálózat beállítása

A **NetworkOptions** file-ban (ami egy a **/SystemSettings** könyvtárban levő, s a **/Programs/BootScripts/Settings/NetworkOptions** file-ra mutató szimbolikus link) adhatja meg a felhasználó a környezetet mindegyik hálózati interfészhez. Az egyik leggyakoribb a „loopback”, a visszahurkolási interfész (**lo**), melynek alapértelmezett specifikációja a következő:

```
lo_IP=127.0.0.1
lo_Netmask=255.0.0.0
```

Megjegyzendő, hogy mindegyik bejegyzés az interfész nevével kezdődik (**lo**), ezt követi egy '_' és a beállítandó tulajdonság neve. Az interfészekhez beállítható tulajdonságok a következők:

- IP: Ha definiált, statikus IP címet állít be ezen interfész használatához;
- Netmask: A használandó alhálózati maszk;
- Broadcast: Az alhálózat üzenetszórasi címe (opcionális);
- BootProto: Meghatározza az interfész inicializálásakor használandó indítási protokollt. Jelenleg ez csak '**dhcp**' -re állítható, vagy üresként hagyható.

Lássunk példaként egy olyan rendszert, amelynek két *ethernet* interfésze van, **eth0** és **eth1**, az első **dhcp**-re konfigurálva, miközben a második statikus konfigurációt használ.

Ehhez az **eth0** a következőképp kell beállítva legyen:

```
eth0_BootProto=dhcp
```

Az **eth1** beállításai pedig eképp nézhetnek ki:

```
eth1_IP=192.168.2.13
eth1_Netmask=255.255.255.0
eth1_Broadcast=192.168.2.255
```

Bármikor indíthatsz egy interfészt a '**StartTask Network <interfész_neve>**' parancsot futtatva. Például

```
StartTask Network eth0
```

Ezzel szemben a '**StopTask Network <interfész_neve>**' lekapcsolja az interfészt.

A legtöbb rendszernél, ha ugyan nem mindnél, az interfészek már indulási időben (bootoláskor) fel kell álljanak. A **NetworkInterfacesOnBoot** változó tartalmazza az összes olyan interfész nevét, melyek már a rendszerindítás folyamata során el kell induljanak maguk is. Például:

```
NetworkInterfacesOnBoot=(
"lo"
"eth0"
"eth1"
)
```

Továbbá, meghatározhatod az átjárót (gateway) a kimenő kommunikáció számára. Ehhez az alábbi példa alapján állítsd be a **Gateway** és a **GatewayInterface** változókat:

```
Gateway=192.168.2.1
GatewayInterface=eth1
```

Megjegyzés: A DNS szerverek specifikációja (ha „kézzel” akarod beállítani) ugyanott lelhető fel, mint a hagyományos Unix rendszereknél, az `/etc/resolv.conf` állományban (illetve itt: `/System/Settings/resolv.conf`). Egy példa a `resolv.conf` fájlra:

```
nameserver 200.176.2.10
nameserver 200.199.252.68
```

Automatikus indítás

Programok automatikus indítása

A saját könyvtárunkban elhelyezett **.xstartup** fájlba beírt parancsok bármelyik ablakkezelő elindításakor lefutnak. A parancsok külön sorba kerüljenek, utánuk fontos a **&** jel, így a program(ok) elindítása után tovább folytatódik az ablakkezelő betöltése. A fájl utolsó sora üres sor legyen!

Eszerint pld a „gaim” progi automatikus indításához neked létre kéne hozni a saját home könyvtáradban a **.xstartup** nevű fájlt, benne ezzel a sorral:

```
gaim &
```

Más progi esetén a „gaim” helyett az indítani kívánt progi neve kell, az a név, amivel parancssorba írva elindítható a program.

Mivel az induláskor lefutó script „tudja”, hogy milyen ablakkezelő indul, ablakkezelőtől függő indulást is megadhatunk. Pl.:

```
--- fájl eleje ---
if [ $1 = fluxbox ]; then
gkrellm &
fi
opera &
--- fájl vége ---
```

Autostart a GoboLinux rendszer alatt

A GoboLinux teljesen személyre szabható rendszer, emiatt az autostarra lényegében akárhány lehetőségünk is kínálkozik, mert bármely indítószkriptet átírhatunk.

Autostart a grafikus felület felállításakor, ha KDM-et használunk

Ehhez írjuk át a

`/Programs/KDE-Base/Current/Shared/config/kdm/Xstartup` file-ot (ez „hagyományos” Linux rendszereknél az `/usr/share/config/kdm/Xstartup` útvonalon keresendő). Én például itt indítottam el az XBindkeys programot, igaz, emiatt a képernyőképeket a gyökérkönyvtárba menti, a `root` tulajdonaként. Ehhez e két sort vettem fel az Xstartup állományba:

```
xmodmap -e "keycode 111 = F13"
xbindkeys &
```

Az inicializálás testreszabása

Alapértelmezés szerint a GoboLinux boot szkripteket úgy konfigurálják, hogy viszonylag kevés démont inicializáljanak és konfiguráljanak, és csak azon dolgokat állítsák be, amelyek közősek a legtöbb felhasználó esetében. Ez a szekció elmagyarázza, miként veheted rá a GoboLinuxot, hogy azt tegye, amit te akarsz.

A GoboLinux boot szkriptek működésének részletes leírása másutt található (a „Boot szkriptek” alfejezet van e témának szentelve). Ezen téma áttekintéséhez annyit kell tudnod, hogy amikor megnyomod a számítógéped „power” gombját, néhány fájl csak úgy „magától”, varázslatosan végrehajtott. Ezek a fájlok itt helyezkednek el: `/System/Settings/BootScripts`; és alant találod a leírásukat.

- **BootUp** - ez az alapvető *boot szkript*. Amint a nevéből következik, a rendszer indulásakor fut le. Olyan általános inicializálásokat tartalmaz, amelyeknek minden induláskor meg kell történniük. Összetettebb indítórutinok építhetők azáltal, hogy e szkriptek meghívják előbb a BootUp-ot, majd egyéb más feladatokat is végrehajtanak. Két effélét alapból tartalmaz a rendszer:
 - **Console** - Ez meghívja a BootUp-ot, majd extra inicializálásokat futtat, a konzol üzemmód beállításának érdekében.
 - **Graphic** - Ez meghívja a BootUp-ot, majd indítja a grafikus

bejelentkezéskezelőt.

- **Shutdown** - A BootUp analogonja, ez az alapvető „leállító” szkript. Ez egyéb, specializáltabb leállítószkriptek által használt:
 - **Reboot** - (Újraindítás) Segítségül hívja a Shutdown-ot a szükséges leállítások elvégzéséhez, majd újraindítja a számítógépet.
 - **Halt** - Meghívja a Shutdown-ot, és leállítja a számítógépet (vagy egyszerűen megáll, ha nincs hardvertámogatás a szoftvervezérelt rendszerleállításhoz).

Ezen fájlok mindegyike lényegében sorok szekvenciális sorozatából áll, mint az alábbi, melynek jelentése eléggé könnyen kitalálható:

Exec "Message..." parancs-paraméterek

A billentyűzet késleltetési ideje és sebessége

Egy konkrét példán bemutatva ezt, tételizzük fel, hogy be akarod állítani a billentyűzet késleltetési idejét és ismétlési sebességét a konzolon. Megteheted ezt, a következő sort adva a `/System/Settings/BootScripts/Console` fájlhoz:

Exec "Making keyboard speedy..." kbdrate -r 30 -d 250

A boot szkriptek továbbá támogatják a „taszkokat” is. Ez némileg haladóbb témakör, melyet az „Indítórutin taszkok” részben magyarázunk el.

Indítórutin taszkok

A boot szkriptjeid alkalmazhatják a „boot taszkokat”, amelyek olyan kis szervíz szkriptek, amiket a programok szállíthatnak. Egy program a taszkjait a **Resources/Tasks** alatt tartalmazza, s ezek be vannak linkelve a `/System/Links/Tasks` könyvtárba. Ez durván megfelel a „hagyományos disztróknál” az `/etc/init.d` szkripteknek, melyek sok disztribúcióban megtalálhatóak.

Parancssorból indíthatsz vagy megállíthatsz taszkokat, a **StartTask** és **StopTask** parancsokat használva. A következő parancs például egy hálózati interfészt indít, a `/System/Settings/BootOptions`-nál megadott konfiguráció szerint (lásd még: „A hálózat beállítása”):

StartTask NetworkInterface eth0

A boot szkripteken belül neked nem kell ezeket a parancsokat használnod, szükséges azonban megadnod egy paramétert, mely jelzi, hogy az adott taszk elindítandó vagy leállítandó:

Exec "Initializing network..." Network Start

Taszkok készítése

Szigorúan véve, a taszk mindössze csupán egy shell szkript a megfelelő tartalomjegyzékben, mely elfogadja a „start” és a „stop” paramétereket. Ebben a képzeletbeli példában bemutatunk egy fájlt (/Programs/Foo/1.0/Resources/Tasks/Foo) amely ezeket a feltételeket kielégíti:

```
#!/bin/sh
case "$1" in
[Ss]tart)
# Itt indul a foo
foo --silly-walk
;;
[Ss]top)
# Itt állítjuk le a foo-t
killall foo
;;
esac # 'esac' nevetséges, tudom... de mit tehetünk, ilyen a shell szintaxisa :)
```

Jó ötlet a fenti példát sablonként használni a taszkjaid elkészítéséhez. Az "[**Ss**]" szintaxis biztosítja, hogy a „start” és „Start” parancsot egyaránt felismerje, elkerülvén így a sajtóhibákat.

Billentyűzetkiosztás

Használd a **KeymapLayout** szekciót a /System/Settings/BootOptions fájlban, kiválasztandó a megfelelő billentyűzetkiosztást, ha nem akarsz rossz billentyű-hozzárendelést.

A rendelkezésre álló billentyűzetkiosztások a **KBD** csomagban vannak; ezek .map fájlok. A konzol billentyűzetkiosztását akármikor átállíthatod a **loadkeys** -t futtatva. Például, Dvorak billentyűzetkiosztáshoz, gépeld be ezt:

```
loadkeys dvorak.map
```

Tartsd észben, hogy a billentyűzetkiosztás grafikus környezetben **más** helyen állítandó be! Keressed meg az "**InputDevice**" szekciót a /System/Settings/xorg.conf állományban. Hasonlóképp, beállíthatod a billentyűzetkiosztást futásidőben újraindítás nélkül: használd a **setxkbmap** eszközt. Például, a billentyűzetet Dvorak kiosztásra konfigurálandó, gépeld be ezt egy X Terminalba (mint például az xterm vagy a konsole):

```
setxkbmap Dvorak
```

Némely desktop környezet grafikus eszközöket is biztosít a billentyűzet-kiosztás beállításához. Például, KDE alatt ezt a KDE Control Center-nél konfigurálhatod.

Boot témák

Fizikailag a GoboLinux boot téma (betöltési képernyő) egy egyszerű szkript fájl, ami a **/Programs/BootScripts/<verzió>/Themes** címen helyezkedik el.

A témafájl az indítórutin magja tölti be. Ennélfogva a témafájl meghívódik minden futásiszint-változáskor. Bár a lényegi részt tartalmazhatja mindössze a szkript törzse, egy komoly boot szkript meg kell valósítsa a következő funkciókat:

- ThemeInit
- ThemeFile
- ThemeBefore
- ThemeAfter
- ThemeFinish

Ezek a funkciók azok a hozzáférési pontok, melyek összekapcsolják a témát az indítórutin magjával.

BootDriver

A BootDriver szkriptet nem kell futtatni közvetlenül a parancssorból. Célja, hogy az **inittab** fájlban (**/System/Settings/inittab**) meghatározott eljárásokat az 1-es, init processzel elindítsa.

BootDriver-nek egy paraméter jelzi, az adott szkriptet melyik futási szinten kell végrehajtani. Itt egy példa az inittab-ra, a BootDriver-nek szóló hívásokkal:

```
id:2:initdefault:
```

```
11:S:wait:/System/Links/Executables/BootDriver BootUp
su:S:respawn:/sbin/sulogin
```

```
12:12345:wait:/System/Links/Executables/BootDriver Console
16:6:wait:/System/Links/Executables/BootDriver Reboot
10:0:wait:/System/Links/Executables/BootDriver Halt
```

```
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -r now
```

```
1:2345:respawn:/System/Links/Executables/agetty tty1 9600
2:2345:respawn:/System/Links/Executables/agetty tty2 9600
3:2345:respawn:/System/Links/Executables/agetty tty3 9600
```



```
4:2345:respawn:/System/Links/Executables/agetty tty4 9600
5:2345:respawn:/System/Links/Executables/agetty tty5 9600
```

Boot téma készítése

A boot téma egy egyszerű szkript fájl. Valójában ha óhajtod, létrehozatsz akár olyan témát is, amely több fájlból áll (ám ez nem okvetlenül jó ötlet). A lényeg itt az, hogy egy fájl elég. Ez a fájl 5 funkciót valósít meg: **ThemeInit**, **ThemeFinish**, **ThemeBefore**, **ThemeAfter** és **ThemeFile**.

Eképp, ha meg akarod valósítani a saját GoboLinux boot témádat, minden amit meg kell tenned csupán az, hogy létrehozod a magad

```
/Programs/BootScripts/<version>/Themes/AzÉnSajátBootTémám
```

fájlt, ami megvalósítja ezen funkciókat (és opcionálisan egyéb mindenfélét is a szkript törzsében) mindennel együtt, amit csak akarsz.

Boot téma konfigurálása

A legtöbb disztribúciónak olyan boot szkriptjei vannak, melyek mindig ugyanúgy néznek ki. Természetesen a *Red Hat* boot szkriptjei különböznek mondjuk a *Slackware* boot szkriptjeitől, de amire én gondolok az az, hogy ha egyszer kiválasztottál egyet ezen disztribúciók közül, akkor mindig ugyanazt fogod látni, amikor indítod a számítógépedet.

Másfelől a GoboLinux támogatja, hogy kiválaszthasd a neked megfelelő megjelenésű és viselkedésű boot témát, mely végrehajtódik a rendszer indulásakor. Sőt, saját témákat is létrehozatsz, amint ez le van írva e fejezetben.

Válaszd ki a neked tetsző témát, beállítva a **BootTheme=<~NekedTetszőTéma>** opciót a **BootOptions** fájlban. Íme néhány, jelenleg elérhető téma:

- **GoboText** - Egy különleges téma, több tonna vezérlőkóddal „leterhelve” a terminálot.
- **Hat** - Egy Red-Hat hasonmás: színes [OK] és [FAILED] szövegek megjelenítésével, jelezve a dolgok inicializálásának sikerét vagy sikertelenségét.
- **Quotes** - Rövid véletlen idézetek kiírásával jelzi minden inicializált tétel sikeres vagy sikertelen voltát.
- **Slack** - Ezt a témát a régi Slackware bootolási módszer ihlette: nincsenek zavaró üzenetek, nincsenek színek és speciális effektek.

Nézd meg a **/Programs/BootScripts/Current/Themes/** könyvtárat, az összes hozzáférhető téma áttekintéséhez.

Használhatod a **TestBootTheme** szkriptet ahhoz, hogy a rendszer újraindítása

nélkül megtekintheted valamely téma kinézetét. A `TestBootTheme` leírása e fejezetben található: „Boot téma tesztelése”.

Idézetek

„A GoboText hibakeresése annyira nehéz volt, hogy ez motiválta a `TestBootTheme` szkript elkészítését.” -- Hisham Muhammad

Boot téma megvalósítása

Ez a rész leírja, miként valósíthatod meg a kötelező funkciók mindegyikét.

ThemeInit

Ez, amint a neve is mutatja, a szokásos helye az inicializálásoknak. Alant található egy példa arra, hogyan tudod használni a standard `$PREVLEVEL` változót (a `Sysvinit init` programból) néhány üzenet küldésére amikor a rendszer inicializálódik vagy lekapcsol.

```
if [ "$PREVLEVEL" = "N" ]
then
echo "GoboLinux is initializing..."
else
echo "GoboLinux is going down..."
fi
```

ThemeFile

A *GoboLinux boot szkriptek* fájlok egy sorozatát hajtják végre (Lásd még e fejezetet: „Az inicializálás testreszabása” a további részletekért). Minden efféle file végrehajtásának elindítása előtt az indítórutinok magja meghívja a `ThemeFile`-t, megadva neki első (és egyetlen) paraméterként az épp végrehajtandó file nevét. Talán felesleges is mondani, de téged semmi nem kötelez arra, hogy visszajelzést adj róla, épp melyik fájl hajtódik végre. (A „*Hat*” téma, például, semmit nem ír ki az ő `ThemeFile`-implementációjában).

Itt következik egy valóban egyszerű megvalósítása a `ThemeFile`-nek.

```
function ThemeFile()
{
echo "Entering file '$1'..."
}
```

ThemeBefore és ThemeAfter

Ezek a funkciók parancsokat hajtanak végre. A **ThemeBefore** egy program végrehajtása *előtt*, a **ThemeAfter**, mint várható is, *utána* hajtódik végre.

A **ThemeBefore** két paramétert igényel: egy azonosítót és egy üzenetet. Az azonosító egy numerikus id, eképp tudod párosítani a **ThemeBefore** és **ThemeAfter** hívásokat. Hasonlóképp a **ThemeAfter** is két paramétert igényel, az azonosítót és egy numerikus visszatérési kódot, mely utóbbi jelzi a sikeres (zéró) vagy a hibás (egyéb érték) végrehajtást.

Ha a témád megvalósítása csupán szekvenciális (azaz, nem használ sz Fork-ot és Wait-ot az indítótaskok végrehajtásánál), az esetben figyelmen kívül hagyhatod az azonosítót – a leggyakoribb témák szekvenciálisak, ez őket egyszerűbbé teszi. Többszálú betöltésnél mindazonáltal a programok eltérő sorrendben fejeződhetnek be, mint ahogyan elindultak; a kilépőkódokkal trükközve grafikusán is reprezentálható a párhuzamos betöltés ezen sajátossága (a **CheckList** téma egy kísérlet erre).

Íme egy gyors példa a **ThemeBefore** és **ThemeAfter** rutinokra:

```
function ThemeBefore() {
shift # ignore id
echo -n "===> $@... "
}

function ThemeAfter() {
if [ "$2" -eq 0 ]
then echo "{SUCCESS}"
else echo "{ERROR}"
fi
}
```

ThemeFinish

A **ThemeFinish** funkció utolsó lépésként hívódik meg a futási szint váltásnál (miután minden más végrehajtott). Azaz, ez a helye annak, ahol bármi végleges kódot hozzáadhatunk. Egy **ThemeFinish**-sel előkészített taszk abba a fájlba ír, melynek tartalmát a login prompt megjelenítése előtt mutatja meg a rendszer a képernyőn. Ezen fájl ("*issue file*") neve a **ThemeFinish** első (és egyetlen) paramétere.

A következő példa egy egyszerű **ThemeFinish** implementációt mutat be, ami egy ilyen "issue" fájlba ír, mely letörli a képernyőt.

```
function ThemeFinish()
{
clear > $1
echo "Welcome!!" >> $1
}
```

Boot téma tesztelése

Szerencsére nem szükséges mindig újraindítanod a számítógépedet ahhoz, hogy teszteld a boot témáidat. A **TestBootTheme** szkript a segítségedre siet ebben. Őt kell futtatnod, paraméterként megadva neki a boot téma nevét:

TestBootTheme AzÉN SajátBootTémám

Ez majd egy bootolási eljárást szimulál neked, sok dolog végrehajtásával. Némelyik csendes lesz, mások sok szöveget írnak majd ki, némelyik sikeres lesz, mások hibát jeleznek... Nyomj entert ha ez végetért a "login" prompt megjelenésével. Ha nem adsz meg e szkriptnek témanevet, kilistázza neked a az elérhető témákat a **/Programs/BootScripts/Current/Themes/** könyvtárból.

Természetesen nagyon hasznos ez a szkript ahhoz hogy lássuk, miként néznek ki a különböző boot témák, s így kiválaszthasd közülük a neked legjobbban tetszőt.

Megjegyzés: Némely téma nem jelenik meg pontosan az Xterm/Konsole -on vagy egyéb grafikus terminálon, illetve nem-standard konzol betűkészletekkel. Valószínűleg a legjobb ugyanabban a környezetben futtatni a **TestBootTheme** szkriptet, mint amilyenben ténylegesen bootolsz majd!

Bootolási opciók

Bootoláskor az indítórutin lefuttat néhány programot. Ezek célja néhány olyan dolog beállítása, mint például a billentyűzetkiosztás és a rendszeróra, illetve a swap partíció(k) inicializálása.. Léteznek bizonyos opciók arra, miként hajtodjanak végre e programok, illetve hogy e programok miként konfigurálják a rendszeredet.

Megváltoztathatod az összes efféle opciót a **/System/Settings/BootOptions** és a **/System/Settings/NetworkOptions** fájlok szerkesztésével. Ezek a fájlok a következő formátumú bejegyzésekből állnak:

Option=value

Megjegyzendő, hogy a szóköz nem megengedett az = jel (egyenlőségjel) előtt vagy után! (Érdekességképpen, igen, ez egy éppoly shell fájl, mint amit a boot szkriptek használnak).

CleanModules

Kitöröl minden eltávolítható kernelmodult.

Használat: **CleanModules**

Opciók:

- -h, --help - Megjeleníti ezt a leírást.
- -v, --version - Kiírja a program verziószámát.
- -V, --verbose - Bőbeszédű üzemmód engedélyezése.

Ez a szkript "**rmmod**"-ot hajt végre minden modulon, megismételve az eljárást a modul függőségeire is, míg nem talál több eltávolítható modult (azaz, míg ki nem töröl minden betöltött modult, kivéve az épp használatban levőket).

CreatePackage

GoboLinux csomag készítése valamely, a rendszerre már felinstallált programból. Alapvetően mindössze egy **.tar.bz2** állományt állít elő a paraméterként megadott program **/Programs** alatti könyvtáraiból és azok tartalmából.

Használat: **CreatePackage** <tartalomjegyzék>

Opciók:

- -h, --help - Megjeleníti ezt a help üzenetet.
- -v, --version - Kiírja a program verzióját.
- -V, --verbose - Bőbeszédű üzemmód engedélyezése.
- -b, --bzip2 - Tömörítés bzip2-vel.
- -D, --no-dependencies - Nem állít elő Dependencies file-ot.
- -g, --gzip - Tömörítés gzip-pel
- -p, --personal-backup - Személyes átviteli mód. NE HASZNÁLD publikus disztribúcióhoz!
- -d, --devel-version - Fejlesztői verziószámok kezelése (főként GoboLinux fejlesztőknek)
- -n, --new-version <entry> - Új verziószámot állít be a csomagnak (főként GoboLinux fejlesztőknek)
- -k, --keep - Új verzió másolása mozgatás helyett (fejlesztői illetve új verziók használatához)
- -s, --sign - Aláírja a csomagot
- -t, --target-dir <entry> - A megadott könyvtárba készíti el a csomagot a szokásos '.' könyvtár helyett.

DEB és UHU csomagok kezelése

Gyakorta jól jön, ha valamit fel tudunk tenni Gobo alá gyorsan, **.deb** vagy **.uhu** csomagokból. Ehhez szükségünk van a **DPKG** csomagkezelő szoftverre, mert az meg tudja nyitni az UHU csomagokat is, lévén hogy szerkezetét tekintve az **uhu** csomag lényegében egy **deb** csomagnak felel meg, csak **.uhu** a kiterjesztése!

Ehhez rakd fel a DPKG csomagot.

Ezután azonban szükség van még egy kis trükkre. A GoboLinux alatt az **mc** úgy van beállítva, hogy azonnal meg akarja nyitni a **deb** csomagot ha rákattintasz - bár amíg nincs **DPKG** csomagkezelő, addig nem mutatja meg neked a csomag belsejében levő **CONTENTS** könyvtárban megbújó fájlokat. Az **UHU** azonban ismeretlen kiterjesztés neki. Tanítsuk meg rá! Ezt hogy ezt a „tanítást” miként kell végrehajtani, leírtuk abban a fejezetben, melynek címe: „Fájltípusok programokhoz társítása az mc progiban” - olvasd el tehát ott! Kiemelem azonban, hogy ez a módosítás csak akkor lép majd életbe, ha ÚJRAINDÍTOD az MC-t!

Ezek után ha rákattintasz benne egy **.uhu** kiterjesztésű fájlra, ugyanúgy megnyitja neked, mintha UhuLinux alatt tennéd.

DeduceName

Megvizsgálja a file-ot, keresve benne a megadott program nagybetűsített neveit.

Használat: **DeduceName** <file> <name>

A szkript végigvizsgálja a megadott file-ot, keresve benne a megadott nevet, és visszatér azzal a nagybetűsített névvel, amit a leggyakoribbnak talál.

Példa: **DeduceName README gtkglarea**

Dependencies

Egy GoboLinux csomag függőségeit kérdezi le. Megvizsgálja a program fájljait, és ellenőrzi a más fájlokra vonatkozó dinamikus kapcsolatait (az **ldd** programot használva). Így módon kiderítheted a csomagok közti függőségi kapcsolatokat.

Használat: **Dependencies** { [-c] <csomag> [<verzió>] | -f <file> }

Opciók:

- -h, --help Megjeleníti ezt a leírást.
- -v, --version Kiírja a program verzióját.

- -V, --verbose Bőbeszédű üzemmód engedélyezése.
- -f, --file Csak egy fájl függőségeinek megjelenítése.
- -c, --check A csomag függőségfájljának vizsgálata, ahelyett hogy generálna egyet.
- -r, --reverse Inverz függőségvizsgálat: Jelzi, hogy mely programok függnék az adott programtól (verziókat is figyelembe vesz, ha adottak)
- -a, --all Minden fájlt megvizsgál, nemcsak a 'bin', 'sbin' és 'lib' könyvtárakban levőeket.
- -m, --missing-only Csak a hiányzó fájlokat jeleníti meg. (csak vizsgálati üzemmód)
- -H, --higher-or-missing-only Csak a hiányzó függőségek vagy újabb verziók kijelzése (csak vizsgálati üzemmód)
- -l, --list Függőségi állomány kilistázása, ha létezik (csinál egyet, ha nincs).
- -d, --dependencies-file Nem nézi meg a megadott függőségi fájlt.
- -e, --execute <entry> A hiányzókon vagy magasabb verziójúakon hajtja végre a parancsot. (csak vizsgálati üzemmód)
- -b, --batch Kötegelt üzemmód: nincsenek kérdések.
- -p, --programs <entry><dir> Függőségek ismételt ellenőrzése.
- -w, --write Függőségek kiírása a **Resources/Dependencies** -be.
- -k, --keep-going Ne fejezze be azonnal, ha bármely végrehajtható program hibásnak bizonyul (csak vizsgálati-végrehajtási üzemmód).

Verziószám megadásának hiányában a **Current** az érvényesnek tekintett.

A Dependencies konfigurálása

Manapság sok program különböző módokon valósít meg egy bizonyos feladatot. Egy példa erre az OpenGL API, melyet az Xorg, MesaLib és Nvidia is implementált. Mindazonáltal nem minden felhasználónak van Nvidia kártyája, s ekkor felmerül a probléma: A recipe létrehozása után miként lehetne az automatikusan generált Dependencies listáról elmaszkolni az Nvidiát? Ezt a problémát megoldja a Dependencies.blacklist-nek nevezett konfigurálható állomány.

Dependencies.blacklist

Ez az állomány felsorolja azokat a csomagokat, amelyeknek nem kell megjeleníteniük a **Dependencies** file-ban, miután a recipe létrejött. A formátuma nagyon egyszerű: minden sorban egy csomag, verziómegjelölés nélkül.

Példa a **Dependencies.blacklist**-ra:

A következő példa a **Glibc** és **Nvidia** csomagok blacklist-je. A megjegyzések és üres sorok nem lesznek figyelembe véve, azaz ezek használhatóak.

```
# A Dependencies.blacklist részletesen itt dokumentált:  
# http://wiki.gobolinux.org/Dependencies
```

Glibc
Nvidia

DetachProgram

Megadván neki egy programnevet és egy cél-tartalomjegyzéket, átmozgatja ezt a programot az ő **/Programs** -ban elfoglalt helyéről a célként meghatározott tartalomjegyzékbe, miközben elhelyez a régi helyén egy az új helyre mutató szimbolikus linket. Ez a szolgáltatás akkor lehet érdekes, ha például át szeretnéd mozgatni egy csomagot egy másik partícióra, de azt szeretnéd, hogy a csomag azért mégis látható legyen a **/Programs** -ban. Minden szimbolikus link újra lesz generálva a cél-tartalomjegyzékre.

Használat: **DetachProgram** <program> [<verzió>] [<cél-tartalomjegyzék>]

Opciók:

- -h, --help - Megjeleníti ezt a leírást.
- -v, --version - Kiírja a program verzióját.
- -V, --verbose - Bőbeszédű üzemmód engedélyezése.
- -c, --copy - Másolás. Megtartja a létező programot a gyökérfájlrendszerben.
- -b, --batch - köteget üzemmód. Minimális mennyiségű figyelmeztető üzenet.
- -D, --no-dependencies - Nem próbálja meg a függőségeket teljesíteni.

Ez a szkript 'átmozgatja' a programot a **/Programs**-hierarchiából a <cél-tartalomjegyzék> könyvtárba. Azután szimbolikus linkeket hoz létre, a **/Programs** funkcionalitását fenntartandó. Ha nem adunk meg <cél-tartalomjegyzék>-et, úgy a **\$goboInstall** környezeti változó lesz figyelembe véve.

Példa:

DetachProgram GCC 2.95.3 /Network/Programs

DisableProgram

Néha szükségessé válhat megtartani egy installált programot a maga tartalomjegyzékében, ugyanakkor viszont elérhetetlenné tenni a **/System/Links**-ben található bejegyzéseken keresztül. A **DisableProgram** arra jó, hogy eltávolítsa az adott programhoz tartozó összes szimbolikus linket, miközben a program maga még mindig megtalálható az állományrendszerben.

Használat: **DisableProgram** <program> [<verzió>]

Opciók:

- -h, --help – Megjeleníti ezt a helpet.
- -v, --version – Kiírja a program verzióját.
- -V, --verbose – Bőbeszédű üzemmód engedélyezése.

Ez a szkript „letiltja” a programot a rendszerben, ugyanakkor meghagyja azt a **/Programs** hierarchiában. A program újraengedélyezéséhez futtasd ismét a **SymLinkProgram** szkriptet. Ha nem adsz meg verziót, a „**Current**” lesz hozzárendelve.

Példa:

DisableProgram GCC 2.95.3

Démonikus felhasználók

Igen, úgy látszik a GoboLinux nemcsak állatbarát, nemcsak felhasználóbarát, de a misztikusok nagy barátja is...

Komolyra fordítva a szót: e cím nem misztikára utal, hanem a következőkre:

Felinstalláltam korábban a **THHTTPD**-t, aztán a **MySQL**-t és a **HTTPD** démont, utóbbi közismertebb neve Apache-webszerver. Ennek során különböző „álfelhasználók” jöttek létre, ami nem zavarna - ha nem az lenne a helyzet, hogy a **KDM** login-screenjénél ezek is kijelződnek a „normál” userek nevei mellett! Ilyenek, hogy

Mysql
Thhttpd
Httpd

No most ezeket nem is törölhetem, ugye... hogyan tilthatom le, hogy ezek ne jelenjenek meg a bejelentkezésnél?

A megoldás:

Szerkesszük a `/System/Settings/passwd` állományt! Ott korábban ilyen bejegyzések voltak (többek közt):

```
thttpd:x:1001:1001::/dev/null:/bin/false
mysql:x:1002:1002:added by Compile for
MySQL:/Programs/MySQL/5.0.27:/System/Links/Executables/false
httpd:x:1003:1003::/Files/WWW:/bin/false
```

Ezt írjuk át így:

```
thttpd:x:1001:1001::/dev/null:
mysql:x:1002:1002:added by Compile for MySQL:/Programs/MySQL/5.0.27:
httpd:x:1003:1003::/Files/WWW:
```

Amint látható, a különbség annyi, hogy a felhasználó shell-jének a végén semmit nem adtunk meg, azaz ami korábban ott volt, azt kitöröltük. Ezek után nem jelennek meg a KDM-ben e démonok.

EDITOR variable

Ha ilyen hibaüzenetet kapunk egy új recipe készítése közben:

```
Compile: Packing recipe directory...
PackRecipe: EDITOR variable is not set. Can't edit description.
akkor állítsuk be a megfelelő környezeti változót! Pld úgy, hogy a
$HOME/.zshrc fájlba felvesszük e sort:
export EDITOR='nano -w'
```

EditScript

Egy GoboLinux szkript készítése vagy szerkesztése.

Használat: **EditScript** <szkript-név-útvonalmegadás-nélkül>

Opciók:

- `-h, --help` - Ennek a leírásnak a megjelenítése.
- `-v, --version` - A program verziójának kiírása.
- `-V, --verbose` - Bőbeszédű üzemmód engedélyezése.

Akkor kell használnod, ha egy még nemlétező szkriptet akarsz előállítani. Az **EditScript** a **TemplateScript**-et használja sablonként az újonnan előállítandó szkriptekhez.

Példa: **EditScript EditScript**

FetchArchive

Megadva neki egy recipét, letölti a fordításhoz szükséges fájlt.

Használat: **FetchArchive** <recipe>

Opciók:

- -h, --help - Megjeleníti ezt a leírást.
- -v, --version - Kiírja a program verzióját.
- -V, --verbose - Bőbeszédű üzemmód engedélyezése.

Példa: **FetchArchive** /Files/Compile/LocalRecipes/K3B/0.10/Recipe

FiboSandbox

Program futtatása védett környezetben, lekorlátozott felhasználóként.

Használat: **FiboSandbox** [<opciók>] <program> [<parancs-paraméterek...>]

Opciók:

- -h, --help - Megjeleníti ezt a leírást.
- -v, --version - Kiírja a program verzióját.
- -V, --verbose - Bőbeszédű üzemmód engedélyezése.
- -o, --original - Eredeti tulajdonosok visszaállítása.
- -r, --restore <entry> - Tulajdonjog beállítása <entry>-re.
- -d, --directory <entry> - A program az <entry> tartalomjegyzékből futtatva. Ez az útvonal lehet abszolút, vagy relatív a sandbox gyökérkönyvtárához viszonyítva.
- -s, --sandbox <entry>[:<entry>...] - Kettőspontokkal elválasztott listája azon területeknek, melyeken a lekorlátozott felhasználónak írási joga van. Az alapértelmezett beállítás a '.'.

A sandboxon belüli használhatóság érdekében a '.' tartalomjegyzék a sandbox gyökeréhez van mountolva (mint /System/Variable/tmp/.FiboSandbox7824). Ezekből relatív útvonalak, mint például '..' használata ahhoz, hogy a '.' könyvtárnál magasabb szinteken levő tartalomjegyzékeket érhesse el, váratlan eredményeket produkálhat. Ez összezavarhatja a sandboxon keresztül megteremtett szimbolikus kapcsolatokat.

Példa:

FiboSandbox -r 0.0 -s './Programs/NaughtyApp/Current' make install

File konvertálása UTF-8 kódolásba

A megoldás, természetesen parancssorból:

```
iconv --from-code=ISO-8859-2 --to-code=UTF-8 file.html > ujfile.html
```

html file helyett természetesen sima txt is megadható.

FilterColors

Monokrómmá szűri a standard inputot (tkp annyit csinál, hogy eltávolítja belőle a színekért felelős terminál-escape-kódokat).

Használat: **FilterColors**

Opciók:

- -h, --help - Megjeleníti ezt a kiírást.
- -v, --version - kiírja a program verzióját.
- -V, --verbose - Bőbeszédű üzemmód engedélyezése.

A csővezetéken át a FilterColors-nak küldött outputból kiszűretnek a terminál escape-kódok (melyek általában színinformációkat határoznak meg, de egyéb használataik is lehetségesek).

Példa: A FilterColors általában csővezetéken keresztül használatos:

```
valamilyen_parancs | FilterColors
```

FilterLines

Egy grep-hez hasonló sokoldalú eszköz.

Példák:

```
cat foo.txt | FilterLines "bla"
```

Megjelenít minden sort a foo.txt-ből, ami a „bla” kifejezést tartalmazza.

```
cat foo.txt | FilterLines "bla" "goob"
```

Megjelenít minden sort a foo.txt-ből, ami tartalmazza a „bla” VAGY a „goob” kifejezést.

```
cat foo.txt | FilterLines "bla" "goob" -n "mac"
```

Megjelenít minden sort a foo.txt-ből, ami tartalmazza a „bla” VAGY a „goob”

kifejezést, de NEM tartalmazza a "mac"-ot.

```
cat foo.txt | FilterLines "bla" "goob" -n "mac" "ops"
```

Megjelenít minden sort a foo.txt-ből, ami tartalmazza a „bla” VAGY a „goob” kifejezést, de NEM tartalmazza sem a „mac”-ot, sem az „ops”-ot.

```
cat foo.txt | FilterLines "bla" -n "goob" "mac" "ops"
```

Megjelenít minden sort a foo.txt-ből, ami tartalmazza a "bla" kifejezést, és NEM tartalmazza sem a "goob"-ot, sem a "mac"-ot, sem az "ops"-ot.

```
cat foo.txt | FilterLines -n "bla" "goob" "mac" "ops"
```

Megjelenít minden sort a foo.txt-ből, ami a „bla”, „goob”, „mac”, „ops” kifejezések egyikét sem tartalmazza.

FindPackage

Csomag vagy recipe előfordulásai után kutat, kizárólag a program neve, illetve a név és a verzió alapján, a név esetében nem kisbetű-nagybetű érzékeny.

Opciók:

- -t [<type>, <type>,...], --types=[<type>, <type>,...] - Annak beállítása, milyen típusú csomagok közt keresgéljen. Az érvényes típusok: **localpackage** (vagyis helyi csomag), **officialpackage** (vagyis hivatalos csomag), **contribpackage** (vagyis hozzájárulási csomag), **recipe**, **tracked** (nyomon követett), **all** (mindegyik). Elegendő e típusoknak mindössze csupán az első karakterét is használni: **l**, **o**, **c**, **r**, **t**, **a**. Az alapértelmezett beállítás: **localpackage**, **officialpackage**. Megjegyzendő, hogy amennyiben a „recipe” típust használjuk, kiolvassa a **Compile.conf**-ből a recipe-tárolók és a helyi recipetárolók helyeit.
- --local-dirs=[<dir>,...] - Hol keresse a helyi bináris csomagokat. Alapértelmezés szerint a **GetAvailable.conf**-ban definiált útvonalat használja.
- -force-update - Erőltetett frissítés: Akkor is letölti a csomaglistát, ha a helyi másolat (ami a **/.Settings/cache/** -ben van) egy óránál frissebb.
- -l, --full-list - A beállított paramétereknek megfelelő összes előfordulást kiírja, nem csak a „legjobbat”.
- -s, --substring - Az összes olyan csomagot figyelembe veszi, aminek a neve tartalmazza a megadott karakterláncot. Automatikusan engedélyezi a '**--full-list**' opciót.
- -W, --no-web - Semmit nem próbál meg letölteni, és nem listáz ki távoli elérési recipéket és csomagokat (hacsak nem használtuk közvetlenül a '**--types=**' opciót). Felülbírálja a '**--force-update**' opciót.

Példák a használatra:

```
FindPackage kde
FindPackage kde 3.2.3
FindPackage KDE 3.2.3
FindPackage --types=recipe kde 3.2.3
FindPackage --types=local_package,official_package kde 3.2.3
FindPackage -t l,o kde 3.2.3
FindPackage --full-list kde 3.2.3
FindPackage --force-update --full-list kde 3.2.3
FindPackage --types=recipe kde-base 3.2.3
FindPackage --types=recipe --substring kd 3.2.3
```

FindQuick

Ez a 'find' előtétprogramja, színes kimenettel, s erősen lecsökkentett szintaxissal. Röviden az 'f' paranccsal is aktivizálható.

Használat: **FindQuick** <keresendő állomány> [<útvonal>]

Opciók:

- -h, --help - Megjeleníti ezt a leírást.
- -v, --version - Kírja a program verzióját.
- -V, --verbose - Bőbeszédű üzemmód engedélyezése.

Ha a <keresendő állomány> a csillagot ("*") tartalmazza, úgy közvetlenül ezt használja a kereséshez. Ha nem tartalmaz csillagot, az esetben a csillag ("*") hozzáadódik a keresett név elejéhez és végéhez. Ha az <útvonal>-at nem adjuk meg, úgy az épp aktuális tartalomjegyzékben keres.

Példa:

```
FindQuick "*.c" ..
```

Figyeld meg, hogy a hozzáadott idézőjelek elkerülik a shell általi behelyettesítést.

FixAttributes

Fájlok attribútumait állítja be a tartalmuktól függően. Ez hasznos lehet az engedélyek beállításához például olyan esetben, amikor olyan fájlrendszerekből másolunk állományokat, amely nem támogatja az Unix-stílusú engedélyeket (Például FAT fájlrendszerű flopick).

Használat: **FixAttributes** [opciók...] fájlok...

Opciók:

- -h, --help - Megjeleníti ezt a leírást.
- -v, --version - Kiírja a program verzióját.
- -V, --verbose - Bőbeszédű üzemmód engedélyezése.
- -R, --recursive - Rekurzív engedélybeállítás az alkönyvtárakban is.
- -t, --true <entry> - Beállítja az engedélyt <entry>-re, ha a fájl végrehajthatónak tekinthető.
- -f, --false <entry> - Beállítja az engedélyt <entry>-re, ha a fájl nem tekinthető végrehajthatónak.

Az alapértelmezett engedélyek a **FixAttributes.conf** állományból olvastatnak ki.

Példák:

```
FixAttributes -t 700 -f 600 *  
FixAttributes -R -f 664 *
```

FixInfo

Újraépíti az info tartalomjegyzékek bejegyzéseit.

Használat: **FixInfo**

Opciók:

- -h, --help - Megjeleníti ezt a leírást.
- -v, --version - Kiírja a program verzióját.
- -V, --verbose - Bőbeszédű üzemmód engedélyezése.
- -m, --mode <entry> - Végrehajtási mód: 'links', 'dir' vagy 'both' (azaz mindkettő). Az alapértelmezett érték a 'both'.
-

FixLibtoolLa

A **libtool** által készített **.la** fájlok átkonvertálása a GoboLinux számára.

Használat: **FixLibtoolLa** <fájlok...>

Opciók:

- -h, --help - Megjeleníti ezt a leírást.

- -v, --version - Kiírja a program verzióját.
- -V, --verbose - Bőbeszédű üzemmód engedélyezése.
- -b, --backup - Biztonsági mentés készítése az eredeti .la fájlról.

Példa:

FixLibtoolLa -b *.la

Fordítás kézzel

Általában nem szorulsz rá, hogy „kézi erővel” fordíts, a Compile eszköz nélkül. A Compile magas szintű fordítóeszköz, s egyik legjelentősebb tulajdonsága a nagyfokú rugalmasság, melynek segítségével a saját céljaidnak megfelelően alakíthatod (s ha eközben netán mégis problémák adódnának a működésével, kérjük írd a levelezőlistára, hogy még rugalmasabbá tehessük). Hasznos lehet mégis tudni, mi történik eközben, a „fű alatt”, így e fejezetben végigmegyünk egy kézi fordítási eljáráson, miközben ugyanazokat a segítő szkripteket használjuk, amiket maga a Compile is.

Ha fordítottál már valaha programot Linux alatt, tudhatod, hogy ez leggyakrabban három részből áll: előkészíteni a forrást (configure), lefordítani azt (make), és telepíteni a fordítás során generált állományokat (make install). Az eljárás alapvetően a GoboLinuxban is hasonló. Mindazonáltal további beavatkozásokat is igénylünk az első lépésben, azért, hogy a forrás megfelelően legyen előkészítve a fordítás számára, figyelembe véve a GoboLinux tartalomjegyzékeit, és további teendők várnak ránk a harmadik lépésben, amikor is a /**Programs**-ba telepített fájlokat be kell linkelni a /**System** hierarchiába (azaz, hogy a programból előállított fájlokat hozzáférhetővé tegyük a rendszer számára).

A GoboLinux alatt néhány szkript segít nekünk, hogy mindezt hatékonyan véghezvessük el.

- **PrepareProgram**
- **SandboxInstall**
- **SymlinkProgram**

A **PrepareProgram** a fentebb leírt első lépést végzi el. A **SandboxInstall** pedig a harmadikat, miközben biztosítja a "make install" veszélytelen végrehajtását. A **SymlinkProgram** a végső teendőt hajtja végre, az új program rendszerbeintegrálását.

A **PrepareProgram** szkript két dolgot csinál. Megalkotja a tartalomjegyzék-hierarchiát a program számára a /**Programs** alá, és előkészíti a forrást a fordításhoz.

A **PrepareProgram** szintaxisa:

PrepareProgram <program-név> <verziószám> [-- <egyéb-opciók>]

A programnév és verziószám megadása kötelező. Ezek azok a nevek, amelyeket a /Programs alatti tartalomjegyzékekben használunk. Például a

PrepareProgram Foo 1.0

ezeket a tartalomjegyzékeket hozza létre:

/Programs/Foo/Settings, /Programs/Foo/1.0, /Programs/Foo/1.0/bin és így tovább.

A **PrepareProgram** által végrehajtott második feladat a források előkészítése. Minthogy nincs a forráskódok formátumát szabályozó standard a disztribúciók számára a szabadszoftverek világában, nincs mód az előkészítés teljes automatizálására sem. Szerencsére a GNU AutoTools népszerűsége elég közel hoz minket egyfajta standardhoz.

A **PrepareProgram** a második lépésben meghatározza, mely előkészítő-eszközök elérhetőek, és végrehajtja a következők egyikét:

1. Ha a program rendelkezik GNU autoconf által generált 'configure' szkripttel, a **PrepareProgram** futtatja azt, megadva neki a szükséges opciókat (többnyire ezeket: **--prefix**, **--sysconfdir**), és bármilyen más opciókat is, amiket a felhasználó előír a parancssorban (mint <egyéb-opciók>).
2. Néhány programfejlesztő saját 'configure' szkriptet ír, de a GNU autoconf népszerűsége miatt olyan parancssori interfészt tervez hozzá, ami hasonló ahhoz, amit amaz használ. A **PrepareProgram** megpróbálja detektálni, hogy a nem-autoconf szkript legalább a **--prefix** opciót elfogadja-e, s ha igen, használja azt.

Röviden, a **PrepareProgramot** tekintheted egy a 'configure'-ot „becsomagoló” eljárásnak. Például a következő futtatása helyett:

```
~/foo-1.0] configure --with-shared=yes
```

használd ezt:

```
~/foo-1.0] PrepareProgram Foo 1.0 -- --with-shared=yes
```

Biztonságos telepítés: SandboxInstall

Szigorúan véve, ha mindent helyesen állítottál be, egyszerűen futtathatod a "make install" parancsot is, miután a "make" már lefutott. Ám amióta „személyre szabott” útvonalakat határozunk meg az alkalmazások számára, bölcs dolognak tűnik egyfajta biztonsági réteggel gondoskodni arról, hogy a telepítés valóban a megfelelő helyen következzen be. Végülis a GoboLinux fő előnye a modularitása: legyünk biztosak abban, hogy a mi „kézi beavatkozásunk” nem fogja megtörni azt.

E feladatban siet segítségünkre a GoboLinux a **SandboxInstall** szkripttel, mely elkülönített, ellenőrzött környezetet hoz létre a biztonságos telepítés érdekében, a 'make install' futtatásának idejére. E környezetet a **FiboSandbox** szkript szolgáltatja, mely különleges engedélyekkel rendelkezik egy speciális

felhasználó, **fibo** útvonalaihoz. A telepítés e felhasználó nevében történik, akinek nincs írási engedélye semmi máshoz mint amit a **SandboxInstall** kijelölt, mint például **/Programs/Foo/Current** és **/Programs/Foo/Settings**. Ennélfogva semmi állomány nem települhet holmi „véletlen helyekre”. A **SandboxInstall** egy tipikus hívása így néz ki:

SandboxInstall Foo

Ha szükséges, egyéb opciók is megadhatóak. A részletekről e fejezetben tájékozódhatsz: **SandboxInstall**.

A program belinkelése: SymlinkProgram

A végső lépést egy program telepítése során a **SymlinkProgram** hajtja végre. Ez a szkript felelős azért, hogy a **/System/Links** hierarchiába belekerüljenek a megfelelő új linkek.

A **SymlinkProgram** szintaxisa:

SymlinkProgram <programnév> [<verziószám>]

A második argumentum opcionális. Ha nem adunk meg verziót, a **Current**ként belinkeltet veszi alapul. Egy általánosan használt parancssori paraméter a **-c overwrite**

mely azt jelenti, hogy írjon felül bármilyen létező szimbolikus linket konfliktusok esetén (az alapértelmezett beállítás a létező linkek megőrzése, hogy ne legyen hatással a már létező alkalmazásokra). Ahhoz, hogy a **Foo** programot beintegráld a **/System** struktúrába, minden linket felülírva, gépeld be ezt:

SymlinkProgram -c overwrite Foo

A **SymlinkProgram** számos parancssori kapcsolóval rendelkezik, haladóbb felhasználók szükségleteit kielégítendő. A részletekért lásd a **SymlinkProgram** fejezetet.

Egy tipikus kézzel való fordítás menete

Ez normális esetben a következőkből áll - tegyük fel a **SzuperProgi 5.1** verziójáról van szó:

1. Letöltöd a forráskódot (valószínűleg valami tömörített formátumban lesz, pld **SzuperProgi-5-1.tar.gz**).
2. Kicsomagolod: lesz egy **SzuperProgi-5-1** nevű könyvtárad.
3. Belépsz oda.
4. **PrepareProgram -t SzuperProgi 5.1**
(Ekkor létrehozza a **/Programs/SzuperProgi/5.1** könyvtárat és minden alkönyvtárat ott ami kell neki)
5. **PrepareProgram SzuperProgi 5.1**
(Ekkor állítja elő a **Configure** állományt).

6. make

(Ekkor fordítja le)

7. SandboxInstall SzuperProgi 5.1

(Ekkor másolja be a /Programs/SzuperProgi/5.1 könyvtárba amit kell)

8. SymlinkProgram SzuperProgi

(Ekkor értesíti a rendszert, hogy immár ez a program is a rendelkezésére áll - ettől kezdve a program használható)

9. Most már törölheted a **SzuperProgi-5-1** könyvtárat, azt, ahová ideiglenesen kicsomagoltad a forrást.

És készen is vagy!

Freshen

Frissített GoboLinux recipék és csomagok után kutat.

Használat: **Freshen** [<opciók>]

Opciók:

- --binaries, -b - Bináris csomagokat keres (alapértelmezés)
- --cache-only, -c - Csak a cache-ot frissíti.
- --compileroot <entry> - Felülbírálja az alap Compile tartalomjegyzéket.
- --debug - hibakereső üzemmód engedélyezése.
- --downgrades, -d - Korábbi változatok figyelembe vétele.
- --help, -h - Ennek a leírásnak a megjelenítése.
- --info, -i<entry> - Információk a <program>-ről
- --netmode, <entry> - Beállítja a netMode-ot (online | offline | always)
- --no-binaries, -B - Nem keres bináris csomagokat.
- --no-cache, -C - Nem használja a cache adatait a /Programs-hoz.
- --no-downgrades, -D - Korábbi változatokat nem vesz figyelembe. (alapértelmezés)
- --no-recipes, -R - Nem keres recipéket.
- --no-upgrades, -N - Nem keres újabb változatokat.
- --recipes, -r - Recipék keresése (alapértelmezés)
- --upgrade, -u - Újabb változatok <lista>
- --upgrade-system, -U - A hozzáférhető recipék újabb változatai, kivéve <lista>
- --upgrades, -n - Újabb változatok keresése (alapértelmezés)
- --verbose, -V - Bőbeszédű üzemmód engedélyezése.
- --version, -v - A program verziójának kiírása.

Paraméterek nélkül futtatva, a **Freshen** kiírja a frissítések listáját a következő formában:

A rövidítések jelentése:

- I - Installált
- U - Javított (újabb változat)
- D - Régebbi változat (Színkód: vörös)
- R - Recipe van hozzá (Színkód: zöld)
- B - Bináris csomag van hozzá (színkód: sárga/barna)
- X - Recipe és bináris csomag is van hozzá (színkód: kék)

Példa:

Freshen -D

Ahhoz hogy használhasd, előbb telepítened kell:

```
pv@Csiszilla ~]Compile Freshen
Compile: Locating a recipe for Freshen ...
Compile: Found recipe for Freshen
GetRecipe: Trying to get http://gobolinux.org/recipe-store/Freshen--2.2.8-r1--
recipe.tar.bz2
GetRecipe: Downloading recipe from http://gobolinux.org/recipe-
store/Freshen--2.2.8-r1--recipe.tar.bz2
Compile: Recipe placed in /Files/Compile/Recipes/Freshen/2.2.8-r1
Compile: Checking dependencies...
Compile: Compiling Freshen version 2.2.8, revision r1.
--09:03:24-- http://e.geek.nz/files/gobo/Freshen-2.2.8.tar.bz2
=> `Freshen-2.2.8.tar.bz2'
Resolving e.geek.nz... 208.113.173.240
Connecting to e.geek.nz|208.113.173.240|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24,597 (24K) [application/x-bzip2]

100%[=====>] 24,597          40.84K/s

09:03:26 (40.70 KB/s) - `Freshen-2.2.8.tar.bz2' saved [24597/24597]

Compile: Unpacking file /Files/Compile/Archives/Freshen-2.2.8.tar.bz2...
Compile: Stripping executables...
UpdateSettings: Current and default settings are different

1      new: 2.2.8/Resources/Defaults/Settings/Freshen/Freshen.conf

UpdateSettings: Quick-updating settings in /Programs/Freshen
UpdateSettings: Backing up settings in
/Programs/Freshen/2.2.8/Resources/SettingsBackup
UpdateSettings: auto: Using
2.2.8/Resources/Defaults/Settings/Freshen/Freshen.conf
SymlinkProgram: Symlinking Freshen 2.2.8.
SymlinkProgram: Symlinking global settings...
SymlinkProgram: Processed 1 file.
SymlinkProgram: Symlinking tasks...
```

```

SymlinkProgram: Storing variable files...
SymlinkProgram: Symlinking libraries...
SymlinkProgram: Processed 6 files.
SymlinkProgram: Updating library database (ldconfig)...
SymlinkProgram: Symlinking headers...
SymlinkProgram: Symlinking info...
SymlinkProgram: Updating info dir...
SymlinkProgram: Symlinking manuals...
SymlinkProgram: Symlinking executables...
SymlinkProgram: Processed 1 file.
SymlinkProgram: Symlinking wrappers...
SymlinkProgram: Symlinking shared...
SymlinkProgram: Processed 1 file.
SymlinkProgram: Removing unused directories...
SymlinkProgram: Done.
Compile: Generating package's build information...
Generating dependency report. Please wait: -
Compile: Generating package's signature...
SignProgram: Creating FileHash
pv@Csiszilla ~]

```

GAMPPS

GAMPPS = GoboLinux + Apache + MySQL + PHP + Python + SSL

E dokumentum célja, hogy lépésenkénti kézikönyvként szolgáljon arra vonatkozóan, miként kell GoboLinux alatt beállítani egy Apache webszervert PHP/SSL támogatással.

MySQL

- Először a MySQL-t kell installálni. Konzolban add ki a következő parancsot:

Compile MySQL

Ezek után elég időd marad ahhoz, hogy elkészíts magadnak egy kávé...

- Amikor végetért a fordítási folyamat, választhatsz magadnak használatra egy konfigurációs fájlt.

A MySQL konfiguráció a **/Programs/MySQL/Settings/mysql** alatt van tárolva, és az aktuálisan használt állomány neve **my.cnf**

Az alapértelmezett config file neve **my-medium.cnf**. Ha neked ez megfelelő, lépj a következő pontra! Máskülönben válassz egyet a többiből és írd felül vele a **my.cnf** állományt.

- Most eljött az ideje, hogy indítsuk a **mysql** szerveret. Ehhez üsd be:

StartTask MySQL

A szervernek most futnia kell.
Normális induláskor ezt írja ki:

Starting MySQL. SUCCESS!

StartTask: MySQL: Returned code 0 => OK.

(megjegyzés: Ha a szerver e pillanatban bedöglik, akkor ellenőrizd az engedélyeket a `/Programs/MySQL/Variable` -nál. A tulajdonos és a csoportok hozzá kell férjenek a `mysql`-hez. Ezt meg tudod valósítani a következő paranccsal:

```
chown -R mysql:mysql /P/MySQL/Variable
```

Ezután indítsd el újra a szerveret.)

- A következő lépés az, hogy beállítsuk a **mysql root user** jelszavát. Üsd be:

```
/Programs/MySQL/Current/bin/mysqladmin -u root password mYNewpAsSw0rD
```

- Ha a MySQL szervert automatikusan akarod indítani már a bootolási folyamat során, üsd be:

```
echo 'Exec "Starting MySQL Database Server..." MySQL Start' >>  
/System/Settings/BootScripts/Multi  
(A fentieket EGY SORBA kell írni!)
```

- Installációdát azzal tesztelheted, hogy begépeled:

```
mysql -u root -p
```

Bekéri tőled a jelszavadat, és láthatod a MySQL monitort, hogy minden jó lett-e.

OpenSSL

Az OpenSSL lefordítása:

Compile OpenSSL

Apache (HTTPD)

- Általában a fordítás az első lépés:

Compile HTTPD

- Fordítás után az Apache készen kell álljon az indításra. Üsd be:

StartTask HTTPD

Ha erre ilyen üzenetet kapnál:

```
httpd: Syntax error on line 428 of
/Programs/HTTPD/Settings/httpd/httpd.conf: Include directory
'/System/Settings/httpd/conf' not found
StartTask: HTTPD: Returned code 1 => FAILED.
```

akkor hozd létre neki a `/System/Settings/httpd/conf` tartalomjegyzéket.

Ha ilyen hibaüzenetet kapnál:

```
httpd: bad user name httpd
StartTask: HTTPD: Returned code 1 => FAILED.
```

akkor mindenekelőtt győződj meg róla, létezik-e **httpd** nevű csoportod a `/System/Settings/group` állományban. E tesztelés legegyszerűbb módja, ha rákeresel ezzel a paranccsal:

```
grep httpd /System/Settings/group
```

Ha nincs ilyen csoportod, hozd létre:

```
groupadd httpd
```

s ezután egy **httpd** nevű felhasználót is kell készíts e paranccsal, amely a **httpd** csoport tagja:

```
useradd httpd -s /bin/false -d /Files/WWW -g httpd
```

- Ha automatikusan, már bootoláskor indítani akarod az Apache szervert, üsd be:

```
echo 'Exec "Starting Apache Web Server..." Apache Start' >>
/System/Settings/BootScripts/Multi
(A fentieket EGY SORBA kell írni!)
```

PHP

- Az Apache PHP Modul installációjához gépeld be:

Compile Mod_PHP

Ez csak az Apache Modult installálja! A PHP és a PEAR CLI verziója nem lesz telepítve!

- Indítsd újra az Apache-ot, hogy az betöltse az új PHP modult:

StopTask HTTPD

StartTask HTTPD

Python

- A modul fordítása:

Compile Mod_Python

- Szerkeszd a **httpd.conf** állományodat (ami e könyvtárban kell legyen: **/Programs/HTTPD/Settings/HTTPD/httpd.conf**) hogy betöltse a Python modult az Apache-val együtt. Add hozzá a következő sort a többi **LoadModule** direktíva után:

LoadModule python_module modules/mod_python.so

- Indítsd újra az Apache-ot, hogy az betöltse a modult:
-

StopTask HTTPD

StartTask HTTPD

Ez minden! Helyezd el a weboldaladat itt: **/Depot/WWW/Documents/**

Az angol nyelvű cikket írta: Volker Wieban

Fordította: Viola Zoltán

Megjegyzés: Miután mindezt telepítetted, érdemes elolvasni e könyvből a „Démonikus felhasználók” fejezetet is!

GenRecipeStore

„Csomagolt recipéket” készít, elküldésre készen, hogy elhelyeztessenek a GoboLinux Recipe Store-ban.

Használat: **GenRecipeStore**

Opciók:

- **-h, --help** - Megjeleníti ezt a leírást.
- **-v, --version** - Kiírja a program verzióját.
- **-V, --verbose** - Bőbeszédű üzemmód engedélyezése.

Ez a szkript a **/Files/Compile/LocalRecipes** -ben található összes recipét összecsomagolja olyan formába, mint az eme példán látható: **Foo--1.0--recipe.tar.bz2** ; és elhelyezi őket a **/Files/Compile/Store** -ban. Ez a hely

mindazonáltal konfigurálható a `/System/Settings/Compile/Compile.conf` állományban.

GetAvailable.conf

A `GetAvailable.conf` az a fájl, ahol útvonalakat (path) és internetcímekeket (URL) állíthatsz be bináris csomagok számára.

Ez a fájl helyileg itt található:

`/Programs/Scripts/Settings/Scripts/GetAvailable.conf` - de ez természetesen be van linkelve a rendszerbe mint `/System/Settings/Scripts/GetAvailable.conf` (és ha használod a GoboLinux könyvtárrendszert, akkor mostanra már tudnod kell, hogy ez ugyanaz, mint a `/etc/Scripts/GetAvailable.conf`).

Ez a fájl szokásos tartalma:

Az időtúllépés (másodpercekben) ameddig megkísérli elhozni a csomagot a szerverről.

```
timeout=15
```

Az útvonalak, amelyeken a helyi csomagokat automatikusan megtalálja. Mindegyik lehet tömörített (pld. `/Depot/Packages/Qt--4.0.0--i686.tar.bz2`) vagy tömörítetlen (pld. `/Mount/SquashFS/Programs/Qt/4.0.0`).

```
defaultLocalPackagesPaths=(  
  "/Depot/Packages"  
  "/Mount/SquashFS/Programs"  
  "/Mount/CD-ROM/Depot/Packages/"  
  ". "  
)
```

A hivatalos bináris csomagok listáit tartalmazó állományokra mutató URL-ek (melyeket tehát az alap fejlesztőgárda készített).

```
officialPackagesLists=(  
  'http://kundor.org/gobo/packages/official/MANIFEST.bz2'  
  'http://gobo.calica.com/packages/official/MANIFEST.bz2'  
)
```

A hozzájárulási bináris csomagok listáit tartalmazó állományokra mutató URL-ek (melyeket tehát nem az alap fejlesztőgárda készített, s nincs rájuk semmi garancia).

```
contribPackagesLists=(  
  'http://kundor.org/gobo/packages/contrib/MANIFEST.bz2'  
  'http://gobo.calica.com/packages/contrib/MANIFEST.bz2'  
)
```

A nyomon követett bináris csomagok listáit tartalmazó állományokra mutató URL-ek. A nyomon követett verzió egy olyan programverzió, amely nem rendelkezik jelenleg még tényleges recipével vagy bináris csomaggal, de már elérhető a program fejlesztőinek a számára.

```
trackedVersionsLists=(  
  'http://gobolinux.org/version-tracker/TrackedVersions.bz2'  
)
```

GetRecipe

Lehív egy recipét, és elhelyezi azt a recipe-fában.

Használat: **GetRecipe** [**<recipe-url>** | **<recipe-útvonal>**]

Opciók:

- -h, --help - Megjeleníti ezt a leírást.
- -v, --version - Kiírja a program verzióját.
- -V, --verbose - Bőbeszédű üzemmód engedélyezése.

Gobo-gombok

Innen letölthető néhány jópofa „gomb”-kép a GoboLinuxról:

<http://www.dreamora.com/home/page.php?159>

illetve

<http://linux.birodalom.net/gobowiki/index.php/Gobo-gombok>

GoboPath

E szkriptet számos más szkript belsejében használják. különböző shell változókat határoz meg, mint pld a `$goboExecutables`, `$goboUsers`, `$goboKernel`, és így tovább, melyek a rendszer fontos belépési pontjait meghatározó stringeket tartalmaznak

(`/System/Links/Executables`, `/Users`, `/System/Kernel`, stb). Kerüljük el a szkriptjeinkben a „fixen behuzalozott dolgokat”, és inkább használjuk ezeket a változókat, hogy ezzel is jobba tegyük a világot... :-)

Görgős egér görgőkezelésének beállítása

A GoboLinux 013-as változata ezt még nem csinálta meg helyettünk automatikusan. A 014-es már megteszi, elég okos ehhez, ám mégsem tartom feleslegesnek leírni, mert elképzelhető, hogy valaki nem görgős egeret használva telepíti a GoboLinuxot, s később, utólag szeretne áttérni görgős egerre! Márpedig e beállítás nélkül az egere görgője nem fog működni.

Tehát: Ehhez szerkesztenünk kell a `/Programs/Xorg/Settings/X11/xorg.conf` állományt. Keressük meg benne azt a részt, ami az egérről szól – könnyen megtaláljuk, mert a „mouse” szó fog benne szerepelni. És ezt írjuk át úgy,

hogy efféleképp nézzen ki:

```
Section "InputDevice"
    Identifier "Mouse0"
    Driver "mouse"
    Option "Emulate3Buttons" "yes"
    Option "ZAxisMapping" "4 5"
    Option "Protocol" "IMPS/2"
    Option "Device" "/dev/input/mice"
    Option "ZAxisMapping" "4 5 6 7"
EndSection
```

Ezután természetesen újra illik indítani a rendszerünket (vagy legalábbis a grafikus felületet).

GrepQuick

Ez a 'grep' egyfajta előtétprogramja, színes kimenettel és erősen csökkentett szintaxissal. Röviden mindössze a 'g' paranccsal is hívható.

Használat: **GrepQuick** <minta> [<fájlok...>]

Opciók:

- -h, --help - Megjeleníti ezt a leírást.
- -v, --version - Kiírja a program verzióját.
- -V, --verbose - Bőbeszédű üzemmód engedélyezése.
- -r, --recursive - Rekurzív.
- -s, --source - Speciális mód a C/C++/Java forráskódokhoz. Átlépi a megjegyzéseket, ha a FilterComments szkript elérhető.
- -w, --whole-word - Csak teljes szavakra keres rá.
- -C, --no-color - Színezés nélkül.

A keresés kisbetű-nagybetű érzékeny, amennyiben a mintában legalább egy nagybetű szerepel. Bináris állományokat nem vizsgál át. Hibaüzeneteket nem jelenít meg. Sorszámokat megjelenít. Ha nem adunk meg fájlt, az összes fájlra értelmezi a keresést (kivéve a „source” módban, ahol ehelyett a *.**[CcHh]*** mintát tekinti alapértelmezés szerint megadottnak).

Példa:

g foo

GrepReplace

Ez egy kis előtétprogram a 'sed'-hez. Megadva neki a 'mit' mintát, a 'mire' mintát és a fájlneveket, a 'mit' minden előfordulása behelyettesítődik a 'mire'-re a megadott fájlokban.

Használat: **GrepReplace** <mit> <mire> <fájlok>

Opciók:

- -h, --help - Megjeleníti ezt a leírást.
- -v, --version - Kiírja a program verzióját.
- -V, --verbose - Bőbeszédű üzemmód engedélyezése.
- -R, --recursive - Rekurzív végrehajtás minden fájlban (felülbírálja az utolsó paramétert).

Példa:

GrepReplace "SimMcast" "simmmcast" *.java

Gyors csomagkezelési útmutató

Csomagkezelés GoboLinux alatt

GoboLinux alá a következő módszerekkel telepíthetsz programot:
Tegyük fel, az XY program 2.0.1 verzióját akarod telepíteni.

ÜGYELJÜNK A KISBETŰ-NAGYBETŰ KÜLÖNBSÉGEKRE A
PARANCSONKBAN!

Elmész terminálba, majd root jogokat szerzel a

su -

paranccsal. (a root jelszavadat bekéri). Ezután:

Bináris csomag telepítése, ha az megvan a GoboLinux hivatalos csomagbázisában

InstallPackage XY

Erre letölti az XY program legfrissebb verzióját, és telepíti. Mindent elintéz helyetted, legfeljebb a menübe nem szerkeszti be a programot, de ott lesz a /Programs könyvtárban.

VAGY:

InstallPackage XY 2.0.1

Ekkor a 2.0.1 verziót telepíti az XY programból, akkor is, ha elérhető már újabb.

A program nevét (XY) a verziótól (2.0.1) SZÓKÖZ választja el!

Telepítés forrásból, recipével

Compile XY

Ekkor letölti a recipe-fileot a GoboLinux recipetárolóból, majd letölti a forrást, és lefordítja a recipe alapján. Ez a legfrissebb verzióval dolgozik.

VAGY:

Compile XY 2.0.1

Mint fent, de a 2.0.1 verzióra.

Telepítés forrásból, ha nincs hozzá recipe. (Csak erősen haladóknak).

Íme egy példa, az XY progí 2.0.1 verziójára. (Itt kötelező verziót megadnod).

Letöltöd egy könyvtárba a progí forrását. Bemész oda terminálban, root jogokkal, majd:

```
PrepareProgram -t XY 2.0.1
```

(Ez elkészíti a **/Programs** alá az alkönyvtárstruktúrát)

```
PrepareProgram XY 2.0.1
```

Ez paraméterezi a configure szkriptet, és futtatja azt. Erre dolgozik egy keveset, majd:

```
make
```

```
SandboxInstall XY 2.0.1
```

```
SymlinkProgram XY
```

Nem hivatalos bináris csomagok feltétele

Az ilyen csomagot előbb le kell töltsd valami könyvtárba a honlapon felsorolt „hozzájárulási” csomagok közül, vagy az én honlapomról ha pld a magyar OpenOffice-ot szeretnéd. majd elmész a könyvtárba amibe letöltötted, root jogokkal, aztán

```
InstallPackage XY
```

Ha netán valamelyik **invalid signature** hibaüzenetet írna ki, akkor (de csak akkor) még ki kell adnod a

```
SymlinkProgram XY
```

parancsot is.

Feltett program letiltása (Rootként)

DisableProgram XY verziószám

Ekkor megmaradnak a progí állományai, csak a szimbolikus linkeket törli ki, azaz nem lesz a rendszernek tudomása róla, hogy van ilyen programod.

Feltett csomag eltávolítása

RemoveProgram XY verziószám

Ekkor a **/Programs** könyvtárból is törli az összes fájlt ami a progíhoz tartozik. Mondjuk én előtte egy **DisableProgram** parancsot is ki szoktam adni a fentiek szerint, hogy a linkeket is törölje előbb.

Ha töröltél valamit kézzel, és bennmaradtak a már sehova sem mutató szimbolikus linkek a rendszerben, akkor menj rootként a

/System/Links/Executables könyvtárba, majd add ki e parancsot:

```
find | RemoveBroken
```

Ez törli a sehova se mutató törött linkeket.

Ha korábban **DisableProgram**mal letiltottál valamit, de még nem törölted a fájljait **RemoveProgram**mal vagy máshogy, akkor ha mégis eszedbe jut hogy neked kéne a progí, nem kell újratelepítened, elég rootként e parancs:

SymlinkProgram XY verziószám

A Manager, azaz a grafikus csomagkezelő felület

A grafikus csomagkezelő program neve **Manager**. Leírás róla itt található:

<http://gobolinux.org/index.php?page=manager>

Hashes

A **FileHash** és **FileHash.sig** kezelése a GoboLinux csomagokban.

Használat: **Hashes** { [-c] <csomag> [<verzió>] }

Opciók:

- -h, --help - Megjeleníti ezt a leírást.
- -v, --version - Kiírja a program verzióját.
- -V, --verbose - Bőbeszédű üzemmód engedélyezése.
- -c, --check - A csomag függőségi fájljának vizsgálata.
- -s, --sign - Generálja, azután aláírja a hash fájlt.
- -g, --generate - Generálja a hash fájlt.

- -l, --list - Kilistázza a hash fájlt, ha létezik (generálja, ha nincs).
- -u, --local-user <entry> - Az <entry>-t használja user ID-ként, hogy aláírjon vele.
- -r, --keyring <entry> - GPG opció egy további keyring helyhez.

Ha nem adunk meg verziót, a **Current** használtatik.

InstallPackage

E szkriptnek paraméterként egy GoboLinux bináris csomag nevét megadva, az InstallPackage kicsomagolja azt a **/Programs** alá, és létrehozza a szimbolikus linkeket a **/System/Links** és a **/System/Settings** könyvtárakba. Megvizsgálja a függőségeket is, figyelembe véve azt is, hogy esetleg valamelyik már telepített programot frissíteni kell-e újabb verzióra.

Használat: **InstallPackage <csomagfile>|<csomagdir>**

Opciók:

- -h, --help - Megjeleníti ezt a leírást.
- -v, --version - Kiírja a program verzióját.
- -V, --verbose - Bőbeszédű üzemmód engedélyezése.
- -b, --batch - Kötegelt üzemmód, felhasználói közbeavatkozás nélkül.
- -D, --no-dependencies - Nem végez függőségellenőrzést.
- -I, --no-install - Nem installálja, csak letölti és elhelyezi a csomagot. Automatikusan engedélyezi a '--keep' opciót.
- -k, --keep - Nem törli a letöltött csomagokat.
- -c, --verify - Ellenőrzi a csomag ellenőrzőösszegét és az aláírását.
- -s, --same <entry> - Mit tegyen ha ugyanolyan már meglevő verziót talál: 'keep', 'remove', 'ask' vagy 'cancel'. Az alapértelmezett érték az 'ask'.
- -o, --old <entry> - Mit tegyen a telepített előző verzióval: 'keep', 'remove', 'ask' vagy 'cancel'. Az alapértelmezett érték a 'keep'.
- -W, --no-web - Internetkapcsolat nélkül: nem vizsgálja meg a letöltési honlapokat, és átugorja az archívumok letöltését.

Az alapértelmezett beállítások: --same = 'cancel', --old = 'keep'. Megjegyzendő, hogy a 'remove' használata veszélyes lehet fontos rendszercsomagok esetén - inkonzisztens állapotba hozhatja a rendszert a telepítési folyamat alatt.

Példa:

InstallPackage Gimp--2.0.5.tar.gz

IsExecutable

0-val tér vissza (sikert jelezve) ha a fájl végrehajtható Linux állománynak tűnik (hash-bang magic-val vagy ELF fejléccel kezdődik), illetve 1-el (hibát jelezve) minden más esetben.

Ezt a segédrutint a **FixAttributes** szkript használja.

JAVA plugin beállítása Firefoxhoz

Mindenekelőtt telepíteni kell a Java-t (például a <http://linux.birodalom.net/gobowiki/index.php/Csomagok> oldalon levő csomagból).

Ezután a

`/Programs/Firefox/verziószám/plugins`

könyvtárba helyezzünk el egy szimbolikus linket, ami a

`/Programs/JRE/1.5.0.08/plugin/i386/ns7/libjavaplugin_oji.so`

állományra mutat!

Majd indítsuk újra a böngészőt (ha épp futott volna eddig...) Ennyi az egész!

Kernel modulok betöltése induláskor

Az Udev és Hotplug segítségével a GoboLinux számos modul betöltésére képes induláskor. Előfordul mégis néha, hogy ezt a felhasználó kell megtegye helyette.

Ez egy olyan munka, amit a `/System/Settings/modprobe.conf` szerkesztésével hajthatsz végre, éppúgy, mint más disztribúciók esetén is. Mindazonáltal, a **BootScripts** csomagnak van egy különleges része a modulok s az ő paramétereik meghatározására, így könnyedén betölthetők a rendszerinduláskor.

Ez a fájl a `/System/Settings/BootOptions`. Vess rá egy pillantást, és látni fogod, hogy van benne egy terület, melyet úgy hívnak, hogy **UserDefinedModules**, mely név meg is magyarázza a lényegét (már annak, aki tud egy kicsit angolul... magyarul a jelentése: „FelhasználóÁltalMeghatározottModulok”). Ezt a területet felhasználják a boot szkriptek, minden egyes sorára futtatják a **modprobe**-ot, hozzáadva természetesen az őket követő paramétereket. Íme egy egyszerű példa arra, hogyan tölthető be az **i810_audio** (audio driver) és az **sk98lin** (ethernet driver) e módszerrel:

```
UserDefinedModules=(  
  "i810_audio"  
  "sk98lin"  
)
```


Kernelfoltok (kernelpatch-ek)

A Linux recipe néhány folttal (patch) együtt érkezik, hogy segítse a felhasználót a rendszer használatában. A foltkészlet többek között a következő módosításokat tartalmazza:

- **GoboHide:** elrejtí a hagyományos tartalomjegyzék-struktúrát a felhasználó elől.
- **SquashFS:** tömörített fájlrendszer, mely lekérdezésre tömörítetlen adatokat ad vissza. Ezt a fájlrendszert jelenleg a **GoboLinux ISO** esetén alkalmazzák, eképp fontos, hogy a CD tartalma könnyen elérhető legyen a mount parancs által.

Végezetül néhány folt a Linux kernel levelezőlistájáról van véve, különösképp bizonyos hibajavítások, a gyakrabban használt eszközmeghajtók esetében.

KeyManager

Digitális aláírások kulcsainak kezelése GoboLinux csomagok számára.

Használat: **KeyManager** { <kulcs_állomány> }

Opciók:

- -h, --help - Megjeleníti ezt a leírást.
- -v, --version - Kiírja a program verzióját.
- -V, --verbose - Bőbeszédű üzemmód engedélyezése.
- -i, --import - Import kulcs
- -l, --list - Publikus kulcsok listázása

KillProcess

A KillProcess olyasféle eljárások egységes hívőfelülete, mint például a közismert 'kill'. A kilőni kívánt alkalmazás meghatározható név szerint, vagy valamely tulajdonsága alapján (például hogy melyik fogyasztja leginkább a memóriát vagy a CPU idejét).

Használat: **KillProcess** { -c | -m | [-n] <regexps...> }

Opciók:

- -h, --help - Megjeleníti ezt a leírást.
- -v, --version - Kiírja a program verzióját.
- -V, --verbose - Bőbeszédű üzemmód engedélyezése.
- -i, --interactive - Jóváhagyást kér a process kilővése előtt.
- -d, --display - Csak információkat ír ki. Valójában nem lő ki egyetlen proceszt sem.
- -n, --name - Alkalmazás kilővése név szerint. Ez hasonló a killallhoz,

de a nevek szabályos kifejezésekként vannak összeillesztve, mint a KillProcess további paraméterei. Ez az alapértelmezett viselkedésmód, amennyiben egyéb kapcsolót nem adunk meg.

- -m, --memory - A legtöbb memóriát fogyasztó processz kilövése.
- -c, --cpu - A legtöbb CPU-időt fogyasztó processz kilövése.
- -s, --spare <entry>[:<entry>...] - Azon programnevek listája kettősponttal elválasztva, melyek nem kilövendőek automatikusan. E listához tételek adhatók a killProcessSpare résznél is a konfigurációs fájlban.

Légy figyelmes a név szerinti használatnál, ez ugyanis végigkeresi a processzek teljes sorát! Ez hasznos, ha a processz neve (amit a killall használ) eltér az eredeti névtől, (mint pld a kdeinit). Mindazonáltal van néhány veszélye is. Például, az 'ion' meghatározás illeszkedik az 'evolution'-ra is. Továbbá, még a paraméterek is illeszkedhetnek: az 'lpd'-re keresés mondjuk kiöli a 'kwrite helpdocument.txt' -t is!

Példa:

KillProcess knotify artsd kwrite

Konzol fontok

Ha nem kedveled a GoboLinux által használt alapértelmezett betűtípust, állítsd át a /System/Settings/BootOptions fájl ConsoleFont-ját az általad óhajtottá.

Továbbá, ez bármikor megtehető a normál setfont eszközzel parancssorból. Olvasd el ehhez a setfont manuálját („man setfont”), hogy megtudd a részleteket.

Emlékeztetőül: ez csak a konzol betűkészleteke vonatkozik! „X” alatt az alkalmazások saját betűtípus-beállításokkal rendelkeznek.

Környezeti változók

Bizonyos környezeti változók befolyásolják némely GoboLinux eszköz működését. Te beállíthatod ezeket. Emlékezz azonban rá, hogy **a GoboLinux a zsh (és nem a bash) shellt használja alapértelmezettként**, eképp a .zshrc (és nem a .bashrc) fájlt kell szerkesztened! A zsh egy Bourne-stílusú shell, eképp érvényes az általad használt szintaxis.

Természetesen, ha te valóban a basht preferálsz (bár mi inkább a zsh-t ajánljuk!), kicserélheted az alapértelmezett shelledet a "chsh" parancsot használva. Lásd a chsh man lapját a részletekhez.

A **\$EDITOR** változó kell hogy beállítva legyen a kedvenc szövegszerkesztődre. Amikor egy GoboLinux eszköz szükségesnek látja egy editor indítását, az ezen változó által mutatottat fogja elindítani (bár ez ténylegesen nem egy GoboLinux változó, különböző más programok is használják).

List

Ez egy kitűnő listázórutin, összevont statisztikával és különleges outputtal. Rövid névként az (**l**) is használható.

Használat: **l** [**<opciók>...**] **<minta>**

Opciók:

- -p, --pager - Oldalakra tördelt lista, hogy visszafelé is tudj scrollozni hosszú tartalomjegyzékeknél.
- -t, --time - Idő szerinti rendezés, a legújabb fájlt mutatja utoljára.
- -s, --size - Méret szerinti rendezés, a legnagyobb állományt mutatja utoljára.
- -d, --directories - Csak tartalomjegyzékeket listáz.
- -h, --hidden - Csak a '.'-tal kezdődő nevű fájlokat ("dot-files") mutatja.
- -a, --all - Minden szabályos és "dot-fájl" kilistázása.
- -L, --no-links - Csak azon fájlok listázása, melyek nem szimbolikus linkek.

MakeRecipe

Recipe sablon készítése.

Használat: **MakeRecipe** [**<alkalmazás_neve>**] [**<alkalmazás_verziója>**] **<url>**

Opciók:

- -h, --help - Megjeleníti ezt a leírást.
- -v, --version - Kiírja a program verzióját.
- -V, --verbose - Bőbeszédű üzemmód engedélyezése.
- -F, --no-fetch - Nem próbálja meg letölteni a csomagot. Feltételezi hogy az már jelen van.
- -C, --no-check - Nem vizsgálja meg, van-e a programnak létező recipéje.
- -b, --batch - Nem tesz fel kérdéseket.

Példa:

MakeRecipe Help2Man 1.33.1

http://ftp.gnu.org/gnu/help2man/help2man_1.33.1.tar.gz

MergeTree

Egy tartalomjegyzék-struktúrát egy másikba tükröz.

Használat: **MergeTree** <forrás> <cél>

Opciók:

- -h, --help - Megjeleníti ezt a leírást.
- -v, --version - Kiírja a program verzióját.
- -V, --verbose - Bőbeszédű üzemmód engedélyezése.
- -d, --delete - Konfliktus esetén felülírja az eredeti fájlt.

A <forrás> minden állománya és könyvtára szimbolikus linkként jelenik meg a <cél>-ban.

Példa:

MergeTree /Programs/KOffice/Current /Programs/KDE/Current

Más disztrók csomagjainak telepítése

E cikket eredetileg *Udvari Zsolt* írta, fórumos hozzászólásként. Én csak kissé kibővítettem a végét.

Mivel a Gobo inkább forrás-alapú, ezért viszonylag kevés (bináris) csomag áll rendelkezésre, így előbb-utóbb eljutunk arra a pontra, hogy forrásból telepítsük, ami nekünk kell, vagy pedig kihasználjuk a Gobo csomagkezelő előnyeit. Az előbbi nyilván a **Compile**-val vagy a **PrepareProgram**-mal működik (attól függően, hogy van-e recipe vagy se), ennek módját elolvashatjuk a leírásban. Ha nem akarunk fordítani (mert annyira nem fontos, hogy optimalizálva legyen, vagy túl sok időbe telne), akkor nyilván a második eset jön elő.

Lássuk akkor az idegeneket:

1. Keressük meg a kívánt programot olyan formátumú csomagban, amit tudunk kezelni (tehát ha pld. nincs rpm telepítve és nem is akarjuk, akkor ne rpm-et keressünk). Jó ötlet pl. slacki-s, frugis, zenwalk-os csomagokat (tar.gz) „lopni”.
2. Gyakorlatilag minden csomag egy tömörített állomány, egy-két telepítési infóval megspékelve. Ezt használjuk ki. Hozzuk létre a

/Programs-ban a megfelelő könyvtárszerkezetet. A legegyszerűbb megoldás a **PrepareProgram** a **-t** opcióval meghívva (ekkor csak a könyvtárakat hozza létre, nem fog **./configure**-t futtatni).

3. Tömörítsük ki a csomagot a létrehozott **/Programs/ProgramNév/Verzió** könyvtárba, még hozzá úgy, hogy az esetleges prefix-ek (pld. **/opt**, **/usr**, **/usr/local**, stb.) ne másolódjanak át. Tehát a **/Programs/ProgramNév/Verzió** könyvtár **bin** könyvtárában legyenek a futtatható fájlok, a **lib**-ben a libek, stb. Ahhoz, hogy tényleg Gobo-szabvány legyen, a **share** könyvtárat nevezzük át **Shared**-re!
4. Csináljunk egy **share** symlinket ami a **Shared**-re mutat, ugyanabba a könyvtárba, ahol a **Shared** van.
5. **SymlinkProgram ProgramNév ProgramVerzió**
6. Nézzük meg, van-e a csomagban mellékelve valami post-install-szerű fájl. Ha igen, akkor szedjük ki, hogy mik azok, ami lefutnának az „idegen” disztróra történő telepítés után (disztrófüggetlen, hogy hol tárolja ezt az infót, tessék szétnézni benne és megkeresni). Ezeket ésszel futtassuk is le! Ilyenek lehetnek pl. konfigurációs fájl módosítása, új könyvtárak vagy fájlok létrehozása, stb.

Megjegyzések:

1. Figyeljünk, hogy a függőségek is fenn legyenek, mivel ilyenkor nincs függőségkezelés, és **No library found**-okat kaphatunk. (Igaz, a függőségeket utólag is telepíthetjük).
2. Ezután akár gobo-csomagot is készíthetünk belőle a **CreatePackage** szkripttel.
3. „Gobo-szabvány” programnevet a **GuessProgramCase**-zel kérhetünk. Ez utóbbi így működik egy példán:

```
vz@Csiszilla ~]$GuessProgramCase xbindkeys
XBindkeys
```

Gondolom, értelmes angol szavakat keres benne (meg gondolom **x**-et az elején), és aszerint átírja kis/nagybetűkre.

E parancsot akárki kiadhatja. **Root**, **user**, bárki. Ez csak egy sztring-konverzió... Jelentősége: abszolút semmi, csak itt így szokás, meg ne lógjon ki a sorból.

Amikor **CreatePackage**-vel csomagot akarunk készíteni belőle, előfordulhat, hogy a rendszer efféle hibaüzenetet küld:

```
No OpenOffice/2.1.0/Resources/Dependencies file. You can run
GenBuildInformation and use the returned information as a basis when writing
it.
```

Ekkor adjuk ki ezt a parancsot:

```
pv@Csiszilla /Programs]GenBuildInformation OpenOffice 2.1.0 > Dependencies
Generating dependency report. Please wait: |
```

A létrejött **Dependencies** állománynak hozzunk létre a telepítendő program könyvtárában egy **Resources** könyvtárat, és másoljuk be oda, a jelen példa esetében tehát a **/Programs/OpenOffice/2.1.0/Resources** könyvtárba kell másolni a **Dependencies** file-ot. Ezután ismételjük meg a **CreatePackage** parancsot. Ekkor valószínűleg kiabálni fog, hogy a csomag nincs aláírva:

```
pv@Csiszilla /Programs]CreatePackage OpenOffice 2.1.0
CreatePackage: Creating package for OpenOffice 2.1.0.
VerifyProgram: No signature to verify.
VerifyProgram: No FileHash to verify.
CreatePackage: No existing sig. Re-sign?
CreatePackage: Press Enter to continue or Ctrl-C to cancel.
```

Semmi baj, nyomjunk entert, erre bekéri a digitális aláírásra vonatkozó jelszavunkat, majd elkészíti a csomagot.

NamingConventions

Meghatároz heurisztikusan egy nagybetűsített, GoboLinux-stílusú nevet.

Használat: **NamingConventions** <név>

Opciók:

- -h, --help - Megjeleníti ezt a leírást.
- -v, --version - Kiírja a program verzióját.
- -V, --verbose - Bőbeszédű üzemmód engedélyezése.

NewVersion

Új verzióra frissíti a recipét.

Használat: **NewVersion** <csomag> <verzió> [<url>]

Opciók:

- -h, --help - Megjeleníti ezt a leírást.
- -v, --version - Kiírja a program verzióját.
- -V, --verbose - Bőbeszédű üzemmód engedélyezése.

A szkript optimistán feltételezi, hogy csak a verziószám változott, és a fordítási eljárás a régi maradt. Emiatt a létrehozott recipét jobb csupán „sablonnak” tekinteni az igazi új verziójú recipéhez.

Példa:

NewVersion Allegro 4.1.12

Pluginek

Hasznos pluginek Firefox alá

Keresés a magyar GoboWikiben:

<http://birodalom.net/pluginek/gobowiki-magyar.xml>

Keresés az angol GoboWikiben:

<http://birodalom.net/pluginek/gobolinux-knowledge-base.xml>

Telepítésük: a fenti linkek tartalmát letöltöd, azután bemásolod a Firefoxod könyvtárának **searchplugins** könyvtárába, majd újraindítod a böngésződet. Ennyi az egész. Illetve, természetesen a keresősávodat meg kell jeleníttetned valamelyik eszköztáron, és rajta kiválasztani a kívánt wikit, hogy miben keressen...

Készítette: én magam, Viola Zoltán. Az alap az, amit Ajnasz¹⁴ készített a HupWikihez; azt írtam át.

Pppoe

Internet csatlakozás beállítása pppoe protokollal

Azoknak, akik internet szolgáltatója usernévvel és jelszóval ad csak hozzáférést a világhálózathoz - jellemzően a betárcsázós és wifi kapcsolatoknál - a NetworkOptions fájl beállításával még nem lesz internete! Na de azért van megoldás, igaz nem Gobo-specifikus, hanem klasszikus linuxos.

Lépések:

1. Töltsd le a Gobo honlapjáról a csomagdepóból a **ppp** és az **rp-pppoe** csomagokat, majd telepítsd fel az **InstallPackage** scripttel őket.
2. Miután feltelepült e két program, létrejöttek a beállítófájlok is. Na most ezeket állítjuk be. Lássuk sorjában:

3. **/System/Setting/ppp/pppoe.conf** beállítása:

Kód:

ETH=eth0

USER=az_en_login_nevem

DEMAND=no

DEFAULT_ROUTE=yes

4. **/System/Setting/ppp/options** beállítása:

Kód:

¹⁴ <http://hup.hu/user/4419>

name az_en_login_nevem

5. /System/Setting/ppp/pap-secrets beállítása:

Kód:

```
"az_en_login_nevem" * "az_en_jelszavam"
```

6. /System/Setting/ppp/cpap-secrets beállítása:

Kód:

```
"az_en_login_nevem" * "az_en_jelszavam"
```

```
"login_nev@valaminet.hu" * "az_en_jelszavam"
```

7. /System/Setting/ppp/resolv.conf beállítása:

Kód:

```
nameserver xxx.xx.xx.xx
```

```
nameserver xxx.xx.xx.xy
```

Ezzel kész is a beállítás!

8. Újraindítod a Gobódat és kiadod a következő parancsot:

Kód:

```
sudo pppoe-start
```

Ezzel rákapcsolódtál a netre, indulhat a Firefox a kedvenc honlapoddal!
(Megjegyzés: Miután nekem nem ilyen az internethozzáférésem, emiatt ezt én,
Viola Zoltán NEM PRÓBÁLTAM KI!)

PrepareProgram

Előkészíti az alkalmazást a telepítéshez, futtatván a 'configure' szkriptet.

Használat:

```
PrepareProgram [ <A_kívánt_név> [<verziószám>] ] [ -- <egyéb_opciók> ]
```

Opciók:

- -h, --help - Megjeleníti ezt a leírást.
- -v, --version - Megjeleníti a program verzióját.
- -V, --verbose - Bőbeszédű üzemmód engedélyezése.
- -b, --batch - Kötegelt üzemmód: felhasználói beavatkozás nélkül dolgozik.
- -t, --tree - Csak előkészíti a tartalomjegyzékeket, nem futtatja a configure-ot.
- -T, --tree-cleanup - hasonló a --tree -hez, de a tartalomjegyzékek elkészítése helyett letörli az üreseket.
- -k, --keep - Megtartja a könyvtárakat, ha azok már léteznek a /Programs tartalomjegyzékben.
- -r, --remove - Eltávolítja a könyvtárakat, ha azok már léteznek a /Programs tartalomjegyzékben.
- -c, --configure <entry> - Engedélyezi a 'configure' szkript használatát. Ez alapértelmezett beállítás.
- -C, --cross-compile - Keresztfordítás: A GoboPath export könyvtárakat

- a `$crossPrefixDir` változóból veszi.
- `-a, --autoconf` - Autoconf bázisú configure hozzárendelése, a detektálás átlépése.
- `-A, --no-autoconf` - NEM autoconf bázisú configure hozzárendelése, a detektálás átlépése.
- `-D, --no-default-options` - Minden alapértelmezett beállítás és detektálás kihagyása, kizárólag a parancssori paraméterekkel dolgozik.

A `/Programs` hierarchia a program számára a `--tree` -vel készíthető el.

Példa:

PrepareProgram KDE 2.2

Programok elnevezése

Egy fontos, ám a recipék készítése közben gyakran lenézett szempont a programok elnevezése. E részben vezérelveket és eszközöket nyújtunk e feladat megkönnyítésére, de semmi olyasmit, ami ellentmondana a józan észnek: tartsd észben, hogy az általad a recipének adott adott név az, ami meg fog jelenni a `/Programs` hierarchiában és a bináris csomag neveként is. Válassz olyan nevet, ami nem „lóg ki” a GoboLinux által telepítettek sorából.

Többnyire különben nem kell aggódj emiatt: a **Compile** és más hasonló eszközök megkönnyítik a munkádat, és javasolnak egy általuk legjobbnak tartott nevet, amennyiben nem adsz meg egyet közvetlenül. Amennyiben ezt nem találnád elfogadhatónak, szakítsd meg a fordítási eljárást, és adj meg paraméterként egy jó nevet.

Milyen a jó név? Olyan, mely eleget tesz a következő irányelveknek:

1. Az alkalmazás íróinak szuverén joguk dönteni a nagybetűsítéssel kapcsolatban, tehát ha ez definiálva van, akkor úgy van az jól. Példák erre: **XFree86**, **LyX**, **Qt**.
2. Ha a nagybetűsítés nem definiált (például mindegyik kisbetű) vagy bizonytalan (például különböző változatokban szerepel a README-ben), akkor a saját nagybetűsítési szabályainkat alkalmazzuk, melyeket a **NamingConventions** szkript definiál.
3. Az alkalmazások meghatározhatják a nevükben az aláhúzások és kötőjelek használatát is a szavak elválasztásához, amennyiben ezt következetesen teszik.
4. A csomagnevek sosem tartalmazhatnak szóközt. Amennyiben mégis, a szóköz egyszerűen nem vevődik figyelembe. (Ennek fényében az **Acrobat_Reader** név rossz).
5. Két csomag neve sosem különbözhet egymástól csupán a

nagybetűsítéstől. A **GoboLinux** szkriptek ugyanis a csomag-, program- és recipeneveket a kis- és nagybetű különbségre érzéketlen módon kezelik (case-insensitive).

A **NamingConventions**-ban definiált szabályokat nem fogod megtalálni ehelyütt részletesen leírva, mert az lényegében a szkript megismétlését jelentené, melyben e szabályok most „írodalmilag” le vannak programozva¹⁵. Röviden, szokásos elő- és utóképzők, ragok után kutat, mint például a „Tools” és „Utils” és magánhangzó-egybeesések után, heurisztikusan.

Ha nem adtunk meg a **Compile**-nak közvetlenül programnevet (például az URL-ből olvassa azt ki), úgy a **NamingConventions** által kikövetkeztetett nevet használja. A legújabb verzióban akad néhány szabály, amit akkor is kikényszerít, ha megadunk neki közvetlenül egy nevet – ezek egyike, hogy minden programnév nagybetűvel kell kezdődjék.

RebuildLinks

Újraépíti a **/System/Links** tartalomjegyzékeit.

Használat: **RebuildLinks** <opciók>

Opciók:

- **-h, --help** - Megjeleníti ezt a leírást.
- **-v, --version** - Kiírja a program verzióját.
- **-V, --verbose** - Bőbeszédű üzemmód engedélyezése.
- **-s, --shared** - Újraépíti a **/System/Links/Shared** -et.
- **-n, --environment** - Újraépíti a **/System/Links/Environment** -et.

A **/System/Links** hierarchia nem minden tartalomjegyzéke újjáépíthető ezen eszköz használatával, mert a program végrehajtása közben ez inkonzisztens állapotba hozhatná a rendszert.

Példa:

RebuildLinks --shared

Recipe típusok

A **Compile** eszköz a csomagok számos típusát képes kezelni, mindegyiket különböző technikával.

¹⁵ „Írodalmi programozás” - A Knuth által 1984-ben indítványozott programozási stílus. Az „írodalmi” programozásban te programokkal tűzdelt magyarázatokat írsz, ahelyett, hogy programokat írnál, megjegyzésekkel fűszerezve azokat. (...) „Ahh... ennek semmi köze Gobo-hoz. Még nem.” -- Leandro Motta Barros

Ha meghívjuk a **MakeRecipe**-t, az letölti a program forráskódját, kitömöríti, és megpróbálja meghatározni, milyen módszerrel telepíthető. Bizonyos azonban, hogy akadnak csomagok, melyeket a **MakeRecipe** nem képes helyesen értelmezni. Ez az a helyzet, amikor a felhasználó közbeavatkozása szükséges, hozzáadva a recipéhez bizonyos kézi módosításokat.

Ahogy a **Compile** leírásában be lett mutatva, a **Compile** a programfordítást bizonyos „recipe-típusok” szerint végzi. Mindegyik típusnak érvényes deklarációi vannak, melyeket beállíthatsz neki, meghatározandó a **Compile**-nek a program fordításához szükséges viselkedését. Lássunk néhány fontosabb opciót mindegyik típushoz:

Recipék konfigurálása

Ez autoconf-bázisú csomagoknál lehetséges, és a **recipe_type=configure** utasítással történik. Az efféle típusú recipék leghétköznapiabb változata további flagokat igényel a configure szkripthez. Ezeket a **configure_options** flaggal adhatod meg, ehhez hasonlóan:

```
configure_options=(
"--enable-shared"
"--with-foo"
)
```

Tartsd észben, hogy a configure szkriptnek történő közvetlen (explicit) flagmegadással te beleavatkozol a csomag függőségi viszonyaiba! Ideális esetben a configure szkript önmaga deríti fel azt, hogy mi elérhető a rendszerben, továbbá engedélyez vagy letilt különböző dolgokat. Sok előrehaladás történt e téren az elmúlt néhány évben, különösen a **Pkgconfig**-nak köszönhetően, de azért akad néhány program, amely flagok közvetlen megadását igényli.

A configure-bázisú recipékben a **Compile** a **PrepareProgram**ot használja arra, hogy megállapítsa, bizonyos standard paraméterek, mint a **--prefix** is, támogatva vannak-e a **configure** szkript által. Ha a program nem támogatja ezen paramétereket, és a **PrepareProgram** ezeket hibásnak detektálja, úgy használhatod az **override_default_options=yes** beállítást arra, hogy a **configure** csak az általad a **configure_options**-ban megadott paraméterekkel dolgozzon.

Egy másik flag ami néha szükséges lehet, az **autoconf_before_configure=yes**. Bizonyos programok a **configure** szkripttel generáltatják a szükséges input állományokat (mint pld a **configure.ac** vagy a **configure.in**) de nem tartalmazzák a generált szkriptet, csak az **autogen.sh** buildert. E flagot használva, az **autogen.sh** hajtódik végre legelső lépésként.

Ha a **configure** vagy az **autogen.sh** nem standard nevekkkel rendelkezik, úgy közvetlenül megadhatod őket a configure és az autogen utasításokkal, eképp:

```
configure=configure.gnu
autogen=gen_all.sh
```

Amióta a „compileprogram” recipék is futtatják a **make**-ot, azóta alkalmazható rájuk a „makefile” recipék legtöbb tapasztalata is, melyekről az alábbiakban szólnunk.

Makefile recipék

Ezek olyan csomagok, melyek közvetlenül használják a Makefile-ot. Könnyen rájöhetsz, e típusba tartozik-e az adott program: ha a program telepítési utasításaiban az áll, hogy futtasd a „**make**”-ot az előző lépés nélkül, úgy az esetben ebbe a típusba tartozik.

A recipék ezen típusánál a **Compile** kétszer futtatja a **make**-ot: a „**build**”-nál és az „**install**”-nál. Némely program csak egy futtatást igényel; bármelyiket letilthatod a **do_build=no** és **do_install=no** opciókkal.

A „makefile” recipékben neked okvetlenül meg kell adnod legalább egy további opciót, tudniillik hogy a Makefile a **/Programs/<programnév>/<verzió>** tartalomjegyzéket használja. Szerencsés esetben a Makefile-nek van egy fő változója, amely a telepítés prefixét vezérli. E változó többnyire olyasféle nevet visel, hogy **PREFIX**, **DESTDIR**, **INSTDIR**... bele kell nézz a Makefile belsejébe, hogy megtaláld. Emlékeztetőül: a telepítési prefix a **Compile** által állíttatik be, mint cél (target) shell változó. Változókat adván a **make**-nak, mi egyaránt használhatjuk a **build_variables**-t és az **install_variables**-t, melyek opciókat szolgáltatnak a „**build**”-nak és „**install**”-nak a **make** futása alatt, vagy épp használhatjuk a **make_variables**-t is, mely mindkét futás számára opciókkal szolgál. Ezek használata hasonló, mint a **configure_options** használata:

```
make_variables=(  
"DESTDIR=$target"  
)
```

Néha különböző útvonalak is definiálva vannak a Makefile-ban néhány változó segítségével. Semmi gond:

```
make_variables=(  
"BINDIR=$target/bin"  
"LIBDIR=$target/lib"  
"ETCDIR=$target/../../Settings"  
)
```

Ha nincsenek a Makefile-nak efféle változói, ha a Makefile „fixen behuzalozott” helyeket tartalmaz a telepítésre vonatkozóan, úgy az esetben sajnos meg kell foltozod a Makefile-ot (és lehet hogy a forráskódot is, rákeresve benne mondjuk **grep**-el az útvonalakra nézve releváns nevekre, mint például **/usr**).

A **configure**-hez és **autogen**-hez hasonlóan, ha a **makefile** más néven szerepel mint ami a standard (Makefile), közvetlenül megadhatod azt a **makefile**

változó használatával:

makefile=GNUmakefile

Perl recipék

Ez a recipetípus Perl modulok számára van. Megjegyzés szerint a Perl modulok számára készült recipék és csomagok neve „**Perl-**” előtaggal kell kezdődjenek. Egy „perl” recipe lényegében egy „makefile” recipe, amely első lépésként a „**perl Makefile.PL**” parancsot futtatja.

Megadhatasz a **Makefile.PL** számára opciókat a **perl_options** tömböt használva, nagyon hasonlóan mint a **configure_options** és a **make_variables**.

Python recipék

A „python” recipék azon programoknak készülnek, melyek a **Python Distutils**-t használják telepítés céljára. Amióta ez annyira elterjedt, létrejött számos kisebb variánsa (néhány csomag a **setup.py** -t, mások a **build.py** -t használják, stb.) -- a „python” megpróbálja észlelni ezeket a variációkat ha lehetséges, de végeredményben közvetlenül is megadhatók ezen speciális esetek flagok által.

A telepítő szkript neve megadható a **build_script**-tel. Ellenőrzést gyakorolhatsz a Python telepítő szkript két futása fölött ugyanazt a két opciót használva, mint amit a „makefile recipék” esetében is (**do_build=no** és **do_install=no** a futtatás letiltásához, **build_target** és **install_target** a futás megnevezéséhez). Más opciók is megadhatóak a **python_options** segítségével. Továbbá, az **override_default_options=yes** opcióval átlépi az összes saját maga által meghatározott beállítást, ha szükséges.

Xmkmf recipék

Ezek a régi „xmkmf-->-->imake” rendszeren alapuló recipék, az **X Window System** használta őket. Mára már elavultak, az Xorg 7.0 áttért a GNU autoconfra.

Scons recipék

Ezek a SCons-on alapuló recipék.

Manifest recipék

A „Manifest” recipék azon programok által használtak, melyeknek fájlokat kell másolniuk. A manifest stílusú listák kettősponttal elkülönített párokból állnak, meghatározván a forrásfájlt és a célt, relatívan a **target**-hez viszonyítva.

Egy hack

A „manifest” recipe típus akkor is használható, ha a csomag nem tartalmaz standard fordítási metódust. A felhasználó feltöltheti a **pre_install()** funkciót a szükséges lépésekkel, és egy üres manifest listát használ emellé. Persze ez a végső megoldás kell legyen amire ráfanyalodunk, tekintettel arra, hogy esetben a recipe lényegében lefokozódik egy shell szkript szintjére.

Recipék készítése

Ebben a fejezetben megtanuljuk, miként készítsünk recipéket. Alapvetően kétféleképpen: egy meglévő módosítával vagy új készítésével. A **NewVersion** és a **MakeRecipe** szkriptek lesznek segítségünkre a vázolt esetekben.

Akárhogy is csináljuk, amint a recipe kész, le van fordítva, egy csomagolt verziót találunk a **/Files/Compile/PackedRecipes** elérési út alatt (illetve a **/System/Settings/Compile/Compile.conf** állomány beállítása szerint). Kérjük, aki létrehozott egy recipét tárja a közösség elé, hogy más is használhassa (akár online módon letöltve azt). Az elkészült recipék **.tar.bz2** formátumban a következő levelezőlistára küldhetők, hogy azokat beilleszthessük a rendszerbe:

gobolinux-recipes@lists.gobolinux.org

A Compile.conf beállítása

Az állományban egyetlen dolgot kell módosítanunk: a nevünket beleírni (hogy később látsszék, hogy ki csinálta a recipéket). Nyissuk meg a **Compile.conf** állományt egy text editorban. Mi a példában a nano programot használjuk (használjuk a kedvencünket):

```
nano /System/Settings/Compile/Compile.conf
```

A második sorban találjuk a „**compileRecipeAuthor**” sort. Szedjük ki a **#** jelet a sor elejéről, és idézőjelek közt adjuk meg saját nevünket.

Dolgunk végeztével mentjük le az állományt és lépünk ki (**nano** esetén: **Control-O, Enter, Control-X**).

Régi recipék frissítése

Amennyiben a létrehozandó csomagból létezik már recipe, nem kell újrafuttatni a **MakeRecipe**-ot. A **NewVersion** szkript egy csomagnevet és verziót vár bemenetként és egy újat készít az előző csomag tartalma alapján.

Az alábbi példák mutatják meg, hogyan használjuk:

NewVersion GCC 4.4.4

Ez a parancs letölti a legfrissebb **GCC** recipét, egy „**4.4.4**” nevű könyvtárat készít a `/Files/Compile/LocalRecipes/GCC` alatt és lecseréli a csomag verzióját 4.4.4-re (\$url beállítás).

Másrészt megadhatjuk a teljes forrásURL-t is, arra az esetre, ha a csere nem futna le sikeresen, vagy az URL megváltozna a szerveren:

NewVersion GCC 4.4.4 `ftp://ftp.gnu.org/gcc/gcc-4.4.4/gcc-4.4.4.tar.bz2`

Új recipe készítése

Az új recipe elnevezése

Fontos, sokszor elhanyagolt feladat a recipe elnevezése. Léteznek javaslatok és eszközök, melyek segítik a munkát, de semmi sem pótolhatja a józan paraszti ésszt: gondoljunk arra, hogy a recipének adandó név fog megjelenni a `/Programs` elérési út alatt, valamint a bináris csomagnévben. Próbáljunk a GoboLinux egészéhez illeszkedő nevet találni.

Általában nem kell sokat aggódnunk, miképp csináljuk: a **Compile** és a kapcsolódó eszközök tudásuk legjavát nyújtják, beleértve a név kitalálását is. Arra azonban figyeljünk, hogy megfelelő nevet válasszon ki.

MakeRecipe

A következőkben egy recipét készítünk egy, a számítógépünkön található alkalmazás forráskódjából. A joe konzolos nevű text editorról van szó.

MakeRecipe `http://unc.dl.sourceforge.net/sourceforge/joe-editor/joe-3.1.tar.gz`

Az URL-ben található fájlnev alapján **MakeRecipe** észreveszi, hogy a fordítandó program a joe nevet viseli, ebből a **NamingConventions** természetesen Joe-t csinál, valamint hogy a program verziója 3.1.

Megjegyzés: amennyiben nem, adjuk meg bemenő paraméterekként:

MakeRecipe HardToDetect 2.0 `http://example.org/htd_2_0.tar.bz2`

A **MakeRecipe** ezt követően tudtunkra adja, hogy letöltötte a program forrását, amely **autoconf** programot használ. Ez jó hír, mivel azt jelenti, hogy részünkről igen kevés munka van vissza. Lefordítjuk és telepítjük hát a csomagot a gépünkre.

Compile joe

Pár perc várakozás után Joe-t lefordítja és telepíti a szkript. Immár van helyi

recipénk, a `/Files/Compile/Store` alatt található `Joe-3.1-recipe.tar.bz2` néven.

Ismét felhívom a figyelmet, hogy amennyiben olyan alkalmazásból készítünk recipét, amelyből még nem létezik, lehetőleg osszuk azt meg a közösséggel (részletek a következő bekezdésben).

A recipék megosztása

Osszuk meg elkészített recipéinket a Gobo-közösséggel! Az elkészült vagy frissített recipéket a `/Files/Compile/LocalRecipes` könyvtárban találjuk. Mielőtt elküldenénk őket, készítsünk `Description` (leírás) fájlt hozzájuk.

Megosztás módja:

`PackRecipe` [programnév]

A `PackRecipe` elkészít egy `tar.bz2` kiterjesztésű állományt a `/Files/Compile/PackedRecipes` elérési útvonal alatt. Küldjünk egy levelet a gobolinux-recipes@lists.gobolinux.org címre, melyhez csatoljuk a fenti állományt.

Ravaszabb témák

Patchek

A `Compile` támogatja a recipékbe épített patchek használatát. Néhány dolgot azonban be kell tartanunk, amikor alkalmazások patcheit készítjük. Legfőképpen: ha egy patch általános jellegű, gondoljuk meg, hogy az eredeti forrás karbantartóinak, a projektnek is elküldjük azt. Így tegyünk, ha a `Compile`-lal patchelünk:

Patchek felépítése

A `Compile` a „-p1” opciót használja patcheléskor, azaz egy könyvtárral beljebb keresi a patchelendő állományt, mint ahol a patch van. Azért így, hogy a recipe frissítése esetén is használhassuk a patchet. Ehhez a patchnek tartalmaznia kell egy további szintet a forráskönyvtárban. „diff”-fel készült példa, melyet az `rlocate` patchéből mutatok:

```
-- rlocate-0.4.3/doc/rlocate.html 2006-01-19 10:04:52.000000000 +0100
+++ rlocate-0.4.3.new/doc/rlocate.html 2006-01-19 19:10:20.000000000 +0100
@@ -223,6 +223,6 @@
permissions of the rlocate and rlocated binaries. To do this execute
the following commands.</p>
<pre>
-      chown root:rlocate /usr/local/bin/rlocate
+      chown 0:rlocate /usr/local/bin/rlocate
      chmod 2755 /usr/local/bin/rlocate
```

Ez a diff például a „01-root_to_uid.patch” nevű állományban van, az `Rlocate 0.4.3` recipe könyvtárában. Természetesen a fájlnev legyen vala-

mennyire beszédes, hogy tudjuk, mit tesz a patch. A patch neve kezdődjék számmal, így biztosítható, hogy a patcheket a megfelelő sorrendben fogjuk a forráshoz illeszteni.

Patch készítése

Patch készítéséhez szükségünk van az eredeti, valamint a módosított forrásra. Ezután használjuk a „diff” alkalmazást, amely elkészíti a patchet. A példában szereplő patch elkészítéséhez az alábbi parancsokat használtam:

```
cd /Files/Compile/Sources
diff -Naur rlocate-0.4.3 rlocate-0.4.3.new > 01-root_to_uid.patch
```

ahol **rlocate-0.4.3.new** nevű könyvtár volt a módosított forrás helye. Amennyiben több állományt változtatunk, és szeretnénk a változtatásokról külön patchet kapni, szükséges a módosított állományok megemlítése a „diff” kiadásakor, vagy használhatunk egy nagy patchet, amelyet editorral kisebb patchekre vághatunk.

Patch-konvertálás

Tegyük föl, hogy már létezik egy patch, ám nem megfelelő elérési utak vannak benne. Kipróbálható, a patch nem fog működni még a teljesen azonos forrásverziónál sem. Ebben az esetben lehet a patchet szerkeszteni, átírva a +++ és ---, valamint a --- rlocate-0.4.3/doc/rlocate.html 2006-01-19 10:04:52.000000000 +0100 tartalmú sorokat, hogy a megfelelő, a forráshoz képest egy szinttel feljebb lévő helyre mutassanak. A kézi javítás embert próbáló feladat lehet, ha a patch elég nagy méretű. Ebben az esetben könnyebb a forrást először patchelni, majd az eredeti, patcheletlen forrás közti patchet elkészíteni, éppen úgy, mint amikor patchet készítünk máskor.

Dinamikus patchek

Előfordul, hogy szükség van olyan patchek készítésére, melyek az adott gépen telepített alkalmazás elérési útjára hivatkoznak. Ilyenkor ahelyett, hogy beleírnánk a patchbe az elérési utat, használhatjuk a Compile dinamikus patchkészítési lehetőségét. Ha a **'in'** (például 01-root_to_uid.patch.in) kiterjesztéssel látjuk el a patchet, Compile felolvassa a fájlt és az adott gépre jellemző értékekkel készít egy patchet (01-root_to_uid.patch). Ugyanazok a recipékben használható változók állnak rendelkezésre, csak a **'Compile_'** szöveget kell eléteni, és **'@%'** és **'%@'** karakterek közé kell őket zárni. Például: amennyiben az rlocate alkalmazásnak szüksége van az „izé” nevű másik alkalmazásra, hogy futni tudjon, és a recipékben használatos, **izé** elérésére szolgáló változó a \$foo_path néven hivatkozhatunk, ennek a dinamikus patchben '@%Compile_foo_path%@" lesz a neve. Alant néhány használható változó, melyek a recipeben foglalt alkalmazásról illetve a szükséges **izé** alkalmazásról tartalmaznak részleteket:

- @%Compile_target%@
- @%Compile_settings_target%@

- @%Compile_variable_target%@
- @%Compile_foo_path%@
- @%Compile_foo_settings_path%@
- @%Compile_foo_variable_path%@

RemoveBroken

Egy állománynevet adva meg neki paraméterként, ez a szkript ellenőrzi, hogy az egy törött link-e vagy sem. Ha törött link, akkor törli. E szkript különösen hasznos a „**find**” paranccsal együtt használva, mint például:

```
find /System/Links | RemoveBroken
```

Ha kézzel törlek valami könyvtárat a **/Programs**-ból, én így szoktam eljárni: elmegyek a **/System/Links** könyvtárba, majd kiadom e parancsot:

```
find | RemoveBroken
```

Megjegyzem, e módszer alapvetően nem helyes – így ugyanis néhány törött link még maradhat a **/System/Settings** könyvtárban! Ekkor tehát a fenti parancsot a **/System/Settings** könyvtárra is illik megismételni!

RemoveEmpty

Az összes üres tartalomjegyzék eltávolítása az aktuálisból (vagy a megadottból).

Használat: **RemoveEmpty**

Opciók:

- -h, --help - Megjeleníti ezt a leírást.
- -v, --version - Kiírja a program verzióját.
- -V, --verbose - Bőbeszédű üzemmód engedélyezése.
- -d, --depth <entry> - Az üres tartalomjegyzékek keresési helyének megadása.
- -D, --no-defaults - Nem távolít el üres tartalomjegyzékeket az alapértelmezettből.

RemoveProgram

A paraméterként megadott program adott verzióját kitörli a GoboLinux rendszerből. Amennyiben „kézi úton”, az „rm” segítségével töröljük ki a programot a `/Programs` könyvtárstruktúrából, úgy maradhat néhány törött link a `/System/Links` és a `/System/Settings` tartalomjegyzékekben. Ekkor megtehetjük, hogy belépünk a „cd” paranccsal e könyvtárakba, és kiadjuk a „find . | RemoveBroken” parancsot, vagy e szkriptet használjuk, hogy e munkát elkerüljük. A `RemoveProgram` átvizsgálja a teljes `/System/Links` hierarchiát (közvetlenül rákeresve a program fájljaira rámutató linkekre).

Használat: `RemoveProgram <program> <verzió>`

Opciók:

- -h, --help - Megjeleníti e leírást.
- -v, --version - Kiírja a program verzióját.
- -V, --verbose - Bőbeszédű üzemmód engedélyezése.
- -b, --batch - Nem kér megerősítést.

Példa:

`RemoveProgram Qt 3.2.3`

RenameCaps

Fájlok átnevezése, miközben nagybetűsíti őket.

Használat: `RenameCaps <opciók> <fájlok...>`

Opciók:

- -h, --help - Megjeleníti ezt a leírást.
- -v, --version - Kiírja a program verzióját.
- -V, --verbose - Bőbeszédű üzemmód engedélyezése.
- -m, --media - Speciális szabályokat használva nevezi át a fájlokat: az 'Any_file.mp3' -ből 'Any File.mp3' lesz.
- -u, --upper - A fájl átnevezése csupa nagybetűssé: 'Anyfile.ext' -ből 'ANYFILE.EXT' lesz.
- -l, --lower - A fájl átnevezése csupa kisbetűssé: 'Anyfile.ext' -ből 'anyfile.ext' lesz. Ez az alapértelmezett üzemmód, amennyiben nem adunk meg más opciót.
- -i, --inverse - inverz "előadók és címek": 'Bar Bar - Foo Foo.ext' -ből 'Foo Foo - Bar Bar.ext' lesz.

Példa:

```
RenameCaps -i "Does It Really Happen - Yes.mp3"
```

Rendszeróra

A GoboLinuxnak tudnia kell, hogy a rendszeróra greenwichi (GMT) vagy helyi idő szerint működik-e (Kivéve természetesen, ha olyan szerencsés vagy, hogy éppen a greenwichi időzónában élsz). Ez elérhető a **ClockMode** szerkesztésével. Állítsd be **GMT**-re, ha a rendszerórád GMT szerint kell működjön. Állítsd **LocalTime**-ra, ha a rendszerórád helyi időre van állítva.

Működésük közben a Linux alkalmazások olyan idő-információkra számítanak, melyeket a **Glibc**, azaz a C könyvtár nyújt. A **Glibc** maga viszont a „localtime” szimbolikus linket használja az ő **Settings** tartalomjegyzékében, (**/Programs/Glibc/Settings**) mely megmutatja, melyik az éppen aktív időzóna. Ezt a szimbolikus linket a telepítőprogram állítja be, azon opció szerint, melyet az időzónákat mutató képernyőn kiválasztottál, ám bármikor átállíthatod manuálisan, a „localtime” szimbolikus linket megváltoztatva valami más helyre, ami a **/Programs/Glibc/Current/Shared/zoneinfo** könyvtárban található. A **localtime** link pontos helye:

```
/Programs/Glibc/Settings/localtime
```

Egyéb infó

A **ClockMode** információt használja még a **hwclock** alkalmazás is, amit tipikusan a bootolási idő alatt indítanak el a **SetClock** taszkon keresztül.

Rootless

A GoboLinux egy roppantmód érdekes lehetősége, mellyel még nem találkoztam más linux-rendszerek alatt, az úgynevezett „rootless”-lehetőség. Hogy mi ez? Minden felesleges szöszaporítás nélkül idemácsolom, amit erről a GoboLinux hivatalos honlapja ír e címen:

http://gobolinux.org/index.php?lang=hu_HU&page=rootless

majd ezután még ejtek pár szót róla.

Rootless (= „root nélküli”) GoboLinux

Ha ki szeretnéd próbálni valódi telepítés nélkül, hogy neked való-e a **GoboLinux**, akkor két választásod van.

Az első, amit ajánlunk, hogy letöltöd a CD-képmást, ami egy „**liveCD**” is, tehát nemcsak telepítő lemez, hanem önmagában egy teljesértékű GoboLinux

rendszer. Ennek segítségével megvizsgálhatod az alternatív fájlrendszer-struktúránkat, és láthatod, miként vannak benne szervezve a dolgok.

A második lehetőség, hogy minimális telepítést hajtasz végre a szkriptekből és a tartalomjegyzék-hierarchiából a „home”-könyvtárad belsejébe. Ezt nevezük mi úgy, hogy „**Rootless GoboLinux**”.¹⁶

A **Rootless** telepítése után lehetőséged nyílik csomagoknak a GoboLinux szkriptrendszer használatával történő installálására, s ezek futtatására a te felhasználói környezeteden belül, úgy, mint az a GoboLinuxban szokásos. Mindazonáltal nem leszel képes érvényes GoboLinux bináris csomagok előállítására, mert minden útvonal a telepítéseidben relatív kell legyen, a te „home”-részkönyvtáradhoz viszonyítva. A GoboLinux recipék azonban kompatibilisek kell legyenek egymással a normál- és a Rootless telepítések esetén.

A Rootless GoboLinuxot azoknak az embereknek hozták létre, akik nem dönthetnek maguk arról, melyik Linux disztribúciót használják (tipikusan például munkahelyeken vagy egyetemi laboratóriumokban), e módszerrel ugyanis a saját „home” könyvtárunkban, saját felhasználói jogosultságukat használva is élvezhetik a mi tartalomjegyzék-struktúránk előnyeit.

A Rootless installálása

A **Rootless GoboLinux** rendben való feltelepítéséhez minden amit tenned kell, hogy letöltöd és futtatod a **CreateRootlessEnvironment** szkriptet (illetve szükséges, hogy a

chmod +x CreateRootlessEnvironment

paranccsal futtathatóvá is tedd a letöltött állományt előbb). E szkript e link alól tölthető le:

http://cvs.savannah.nongnu.org/viewcvs/*checkout*/tools/Scripts/bin/CreateRootlessEnvironment?root=goboscripts

(A fenti link egy sorba írandó...)

A fent említett szkript néhány egyszerű feladatot végez el. Alapvetően:

- Elkészíti a „Programs” és a „System” könyvtárakat
- Telepíti a Scripts csomagot
- Megalkotja a 'StartRootless' nevű szkripteket, melyeket a ~/.zshrc vagy a ~/.bashrc használ

Követelmények

A **CreateRootlessEnvironment** és a GoboLinux szkriptek feltételezik, hogy

¹⁶ Magyarul „Root nélküli GoboLinux” lenne a neve, vagy „Gyökérfelhasználó nélküli GoboLinux”, esetleg „Rendszergazda nélküli GoboLinux” - de szerintem nincs értelme lefordítani e nagyon speciális szakkifejezést, így az angol nyelvű oldalakon is érteni fogjuk. - a fordító megjegyzése.

rendelkezésre áll egy alap GNU felhasználói tér (ez általában nem probléma egy Linux-bázisú rendszerben, de máshol nem okvetlenül egyértelmű a megléte). Mielőtt futtatod a szkriptet, győződj meg róla, hogy ezek az eszközök elérhetőek a **\$PATH**-odban:

- GNU Bash
- GNU CoreUtils
- GNU Sed
- GNU Wget

A Rootless uninstallálása

Ha bármi okból szükségét éreznéd, hogy valamikor később eltávolítsd a **Rootless**-t, ez igen egyszerű feladat - mindössze töröld ezeket a könyvtárakat:

```
~/Programs  
~/System  
~/Settings
```

Miért jó ez nekünk?

Na és akkor most már mindent tudsz amit kell, kedves Olvasó, csak annyit jegyezni még meg, hogy én kipróbáltam ezt az UhuLinux-1.2 disztribúció alatt, és kiválóan működik! A telepítéshez nem kell más, mint hogy töltsd le a szkriptet a fentebb megadott linkről, tedd futtathatóvá, és indítsd el! Ha feltesz valami kérdést, akkor a legelsőnél bármi gombot megnyomhatunk, kivéve a Ctrl-C kombinációt, a többinél egyszerűen válaszoljunk Yest (azaz nyomjuk meg az „Y” gombot. Magyarán ha mindenütt Y-t nyomunk, az jó lesz...)

A könyvtárak létrehozásán kívül csak annyi változás történik a rendszerünkben, hogy a Bash és a Zsh shellünk PATH-jába bekerül az az útvonal, amin át elérhetőek a GoboLinux szkriptek.

Na most hogy miért jó ez nekünk? Mert pld így mindent telepíthetünk forrásból a GoboLinux recipetárolóinak segítségével, anélkül, hogy görcsölnünk kellene sokat bizonyos fordítási opciók beállításával, sőt, hiába a forrásból való telepítés, hála a recipéknek, még a szükséges függőségek is feltelepülnek szükség esetén!

A saját **Home** könyvtáradon belül lesz egy **Programs** könyvtár, amiben a megfelelő altartalomjegyzékeken belül láthatod, hogy mi van a rendszeredbe telepítve. Ha nem kell valami, egyszerűen kitörölöd, ahogy azt egy GoboLinux rendszerben szokás, majd eltávolítod a **RemoveBroken** paranccsal a törött szimbolikus linkeket - például. De más módszer is lehetséges... Azaz, mindent megtehetsz, mint egy **GoboLinux** rendszerben, miközben be vagy ágyazva egy **NemGoboLinux** rendszerbe! És persze mi sem akadályoz benne téged, hogy a nem-gobo-linuxodat úgy kezeld, mint ahogy azt korábban megszoktad!

Hamar rá fogsz jönni, hogy legalábbis forrásból telepíteni dolgokat így, a Gobo-struktúrán keresztül sokkal kényelmesebb. Egyáltalán, így nem csak a hagyományos rendszered csomagkészletét használhatod, de a GoboLinux forráskészletét is, éppen ahhoz, ami a hagyományos linuxrendszerek közül a bináris alapú disztróknál a leginkább fogós kérdés: a forrásból való telepítéshez!

Magyar embernek fontos néhány program „magyarsága”. Ezen honlapon: <http://linux.birodalom.net/gobowiki/index.php/Csomagok> fel van sorolva néhány olyan csomag a GoboLinuxhoz, amit én készítettem el. Ezek közül figyelemre érdemes a magyar Firefox és az OpenOffice. Bár a GoboLinuxhoz készített bináris csomagok általában nem kompatibilisek a Rootless rendszerrel, de e kettő igen: ezekben ugyanis a „hagyományos” elérési utak szerepelnek, mert eleve binárisból hekkeltem őket! Ezek telepítése rootless rendszerbe tehát csak annyi, hogy letöltöd őket, elkészítéd a Home-od /**Programs** könyvtárába a megfelelő könyvtárat (OpenOffice esetén pld az OpenOffice könyvtárat), alá a megfelelő verzió könyvtárat, és ide szépen bemásolsz mindent a kitömörített csomagból. Majd csinálsz egy Current linket a **\$HOME/Programs/OpenOffice** alá, ami a megfelelő verzió alkönyvtárára mutat, és ezután
SymlinkProgram OpenOffice

Firefox esetén hasonlóképp kell eljárnod.

Persze, ne feledjük, hogy ezzel csak telepítve vannak a programok, de a rendszered menüjébe neked kell „beszerkesztened” őket... OpenOffice esetén pld a szövegszerkesztőt (writer) „swriter” néven keresd, a **\$HOME/Programs/OpenOffice/Current/program/swriter** link alatt.

UhuLinux alatt én eddig a régi, 1.1.3 OpenOffice-ot használtam, de most felraktam így a legújabb OpenOffice-ot alá, és kiválóan dolgozik, ám ha nagyon akarnám, használhatnám a régit is... azaz, így tényleg több programot használhatsz ugyanabban a rendszeredben, mármint egy program több változatát!

Tehát ha valamiért nem akarsz rögvest teljesen áttérni GoboLinuxra (ami felettébb helytelen...), akkor telepítsd azt **Rootless** változatban! Ehhez még csak újra sem kell particionálnod a rendszeredet!

És akkor most még egyszer a rootless-ről, de kicsit részletesebben:

Rootless installálása

A GoboRootless (amit általában egyszerűen csak „Rootless”-nek neveznek röviden) egy olyan módszer, melynek segítségével a Gobo csomagkezelő eszközöket egy másik rendszer belsejében használhatjuk. A GoboRootless nem igényel root jogosultságokat a befogadó rendszeren belül. A vendéglátó rendszer tehát lehet egy nem-Gobo Linux disztribúció. Régi tervünk - bár

alacsony prioritással dolgozunk rajta - hogy e módszert elérhetővé tegyük nem-Linux gazdarendszereken is, mint pld a BSD rendszerek, OS X, sőt az MS Windows, de pillanatnyilag ezek támogatottsága legjobb esetben is csak kísérletinek nevezhető.

Mikor célszerű a GoboRootless használata?

Akkor, ha könnyen és tisztán óhajtasz programokat forrásból fordítani és telepíteni, ám...

- Nincs rendszergazdai jogosultságod a gazdarendszerben, vagy...
- Nem akarod megkockáztatni a gazdarendszer tönkremenetelét, vagy...
- Ugyanazon program több változatát óhajtod egyidőben installálni, vagy...
- Bármely egyéb okból is, de a Gobo eszközeit óhajtod használni.

A rendszergazdai jogosultságokat igénylő csomagkezelő eszközök tönkretethetik a rendszert. Minthogy a GoboRootless nem igényel root jogokat, a GoboRootless el van különítve a gazdarendszertől, emiatt nem képes azt tönkretenni.

Szükségletei

Minden platformon kell neki:

- GNU Bash
- GNU Coreutil
- GNU Sed
- GNU Wget
- bzip2
- Python (Jaj!!!)
- beleértve a Python bz2 modulját

Figyelmeztetés: A Scripts csomag 2.6.4 verziójában a Rootless igényli, hogy a `/bin/sh` a GNU Bash-ra mutasson. Ez egy hiba, amit a jövőben majd kijavítunk. E hibának főként az Ubuntu alatt van jelentősége, aholis a `/bin/sh` egy symlink a dash shellre. Ha van root jogosultságod, kicserélheted e symlinket, hogy a bash-ra mutasson. Ellenkező esetben a Rootless sajnos nem fog neked (megfelelően) működni az Ubuntun.

A nem-i686 platformok igényei:

- GNU GCC

(Nem-i686 felhasználók: Lásd alant a platformspecifikus megjegyzéseket.)

Installálás

```
$ cd $HOME
```

```
$ wget
```

```
http://cvs.savannah.nongnu.org/viewcvs/*checkout*/tools/Scripts/bin/CreateRootl
```



```

essEnvironment?root=goboscripts
$ mv CreateRootlessEnvironment\?root\=goboscripts CreateRootlessEnvironment
$ chmod u+x CreateRootlessEnvironment
$ ./CreateRootlessEnvironment
$ . Programs/Rootless/Current/bin/StartRootless
$ InstallPackage Atool
$ InstallPackage Compile
$ Compile Mtail

```

Trükkök és csapdák

A gazdarendszer fejlécállományai és könyvtárai összezavarhatják a Rootlesst. A Rootless megpróbálja tudatni az összes GNU eszközzel, hogy hagyja figyelmen kívül a gazdarendszer állományait, de ez nem működik mindig helyesen. Ez csúnya és frusztráló hibákat okozhat.

Például, le óhajtanád fodítani (Compile) az emacs-ot (vagy bármi progit, amihez kell az ncurses) Rootless-ben Debian alatt. Az Emacs ./configure szkriptje meg fogja találni az ncurses libraryk Debian verzióját és használja azokat. Az Ncurses egy második példányának az installálása a Rootless belsejébe nem segít, mert a ./configure a Debian-féle ncursesre lel rá elsőként és mindig azt használja, nem véve tudomást a Rootless alá installált fájlokról. Továbbá, nem lenne szerencsés eltávolítani az ncurses Debian-féle verzióját, hiszen a rendszer számos fontos része függ az ncurses Debian-verziójától. Két megoldása létezik a problémának, s potenciálisan mindkettő root jogosultságokat igényel a gazdarendszeren. A két megoldás mindegyike kitérőaltatja a következő két szakaszban.

Rootless speciális platformokon

Az első megoldás, hogy a könyvtáraknak mindig a gazdarendszer-beli verzióit használjuk, és tartózkodunk attól, hogy redundáns másolatokat készítsünk a Rootless belsejébe ezekről. Ez vagy lehetséges, vagy nem.

Rootless chroot-olt környezetben

A második megoldás a Rootless chroot-olt környezetben való futtatása. Ez teljesen elszigeteli a Rootlesst (fájlrendszer-szinten) a gazdarendszertől. Mindazonáltal a „chroot” beállítása eléggé komplikált, az ebbe való belépés még bonyolultabb. Ez a megoldás itt csak a haladóbb és kalandosabb kedvű felhasználóknak van megemlítve, hogy tudják, hogy ez lehetséges, és hogy ez egy tökéletesen járható út, amennyiben már rendelkezzel kellő ismeretekkel azon meglehetősen haladóbb dolgokkal kapcsolatban, mint például: **chroot**, **suid** bitek, **GCC**-specifikus fájlok, **/proc** filerendszer, stb.

SMF hack

E fejezet azoknak van szánva, akiknek van egy (több...) SMF alapú fórumuk, és szeretnének belőle könnyen hivatkozni a magyar vagy angol GoboWiki (Sőt akkor már bármi más wiki) szócikkeire.

A fórumkódba beillesztendő kódrészlet:

Bővítsük ki a fórum **Sources/Subs.php** file-ját a következő sorokkal:

```
array(
    'tag' => 'gobo',
    'type' => 'unparsed_content',
    'content' => '<a href="http://linux.birodalom.net/gobowiki/index.php/$1"
target="_blank" style="border-bottom:1px dashed #FF0000;text-
decoration:none">$1</a><sup><span style="font-
size:7pt;color:red;">GoboWiki</span></sup>', 'validate' =>
create_function('&$tag, &$data, $disabled', '$data = strstr($data, array(\'<br /
>\' => \'\'))');), 'disallow_children' => array('email', 'ftp', 'url', 'iurl'),
    'disabled_content' => '$1', ),
array(
    'tag' => 'gobo',
    'type' => 'unparsed_equals',
    'before' => '<a href="http://linux.birodalom.net/gobowiki/index.php/$1"
target="_blank" style="border-bottom:1px dashed #FF0000;text-
decoration:none">', 'after' => '</a><sup><span style="font-
size:7pt;color:red;">GoboWiki</span></sup>', 'disallow_children' =>
array('email', 'ftp', 'url', 'iurl'), 'disabled_before' => '', 'disabled_after'
=> '$1', ),
array(
    'tag' => 'wiki',
    'type' => 'unparsed_content',
    'content' => '<a href="http://hu.wikipedia.org/wiki/$1" target="_blank"
style="border-bottom:1px dashed #FF0000;text-decoration:none">$1</a><sup><span
style="font-size:7pt;color:red;">Wiki</span></sup>', 'validate' =>
create_function('&$tag, &$data, $disabled', '$data = strstr($data, array(\'<br /
>\' => \'\'))');), 'disallow_children' => array('email', 'ftp', 'url', 'iurl'),
    'disabled_content' => '$1', ),
array(
    'tag' => 'wiki',
    'type' => 'unparsed_equals',
    'before' => '<a href="http://hu.wikipedia.org/wiki/$1" target="_blank"
style="border-bottom:1px dashed #FF0000;text-decoration:none">', 'after' => '</
a><sup><span style="font-size:7pt;color:red;">Wiki</span></sup>',
    'disallow_children' => array('email', 'ftp', 'url', 'iurl'), 'disabled_before'
=> '', 'disabled_after' => '$1', ),
array(
    'tag' => 'enwiki',
    'type' => 'unparsed_content',
    'content' => '<a href="http://en.wikipedia.org/wiki/$1" target="_blank"
style="border-bottom:1px dashed #FF0000;text-decoration:none">$1</a><sup><span
style="font-size:7pt;color:red;">EnWiki</span></sup>', 'validate' =>
create_function('&$tag, &$data, $disabled', '$data = strstr($data, array(\'<br /
>\' => \'\'))');), 'disallow_children' => array('email', 'ftp', 'url', 'iurl'),
    'disabled_content' => '$1', ),
array(
    'tag' => 'enwiki',
    'type' => 'unparsed_equals', 'before' => '<a
href="http://en.wikipedia.org/wiki/$1" target="_blank" style="border-bottom:1px
dashed #FF0000;text-decoration:none">', 'after' => '</a><sup><span style="font-
size:7pt;color:red;">EnWiki</span></sup>', 'disallow_children' =>
```

```

array('email', 'ftp', 'url', 'iurl'), 'disabled_before' => '', 'disabled_after'
=> '$1', ),
array(
'tag' => 'engobo',
'type' => 'unparsed_content',
'content' => '<a href="http://gobo.kundor.org/wiki/index.php/$1"
target="_blank" style="border-bottom:1px dashed #FF0000;text-
decoration:none">$1</a><sup><span style="font-
size:7pt;color:red;">PoWi</span></sup>', 'validate' => create_function('&$tag,
&$data, $disabled', '$data = strstr($data, array(\'<br />\' => \'\\\'));'),
'disallow_children' => array('email', 'ftp', 'url', 'iurl'), 'disabled_content'
=> '$1', ),
array(
'tag' => 'engobo',
'type' => 'unparsed_equals',
'before' => '<a href="http://gobo.kundor.org/wiki/index.php/$1" target="_blank"
style="border-bottom:1px dashed #FF0000;text-decoration:none">', 'after' => '</
a><sup><span style="font-size:7pt;color:red;">EnGoboWiki</span></sup>',
'disallow_children' => array('email', 'ftp', 'url', 'iurl'), 'disabled_before'
=> '', 'disabled_after' => '$1', ),
);

```

Használata:

Hivatkozás a magyar GoboWiki szócikkeire:

[gobo=szócikk_neve]MEGJELENÍTETT_NÉV[/gobo]

Például ha azt akarjuk, hogy valami bejegyzésben a „kedvenc disztrónk” szöveg alatt olyan link legyen, mely a magyar GoboWiki „GoboLinux” szócikkjére mutat, akkor a beszúrandó kód egy példaszöveg kapcsán:

Kód:

A [gobo=GoboLinux]kedvenc disztrónk[/gobo] megalkotója...

Természetesen van lehetőség a magyar vagy angol wikire is hivatkozni:

Magyar wiki:

A [wiki=Miskolc]Miskolc[/wiki] nevű város...

Angol wiki:

The [enwiki=buddhism]buddhism[/enwiki] is the name of the...

Az angol GoboWiki:

A [engobo=Gobo]Gobo[/engobo] név eredete...

SandboxInstall

Futtatja a 'make install' parancsot, **sandbox** környezetben.

Használat:

```
SandboxInstall [<opciók>] <program_neve> [<program_verzió>] [ --  
<extra_paraméterek> ]
```

Opciók:

- -h, --help - Megjeleníti ezt a leírást.
- -v, --version - Kiírja a program verzióját.
- -V, --verbose - Bőbeszédű üzemmód engedélyezése.
- -t, --target <entry> - A Makefile által használandó parancs. Az alapértelmezett érték az 'install'.
- -f, --makefile <entry> - Beállítja, milyen makefile használandó. Az alapértelmezett érték a 'Makefile'.
- -c, --command <entry> - A megadott parancs használata a make helyett. A --target és --makefile opciók ekkor nem lesznek figyelembe véve.
- -C, --cross-compile - Keresztfordítás: A GoboPath export könyvtárakat a \$crossPrefixDir változóból veszi.
- -a, --add-allowed - Egyéb más könyvtárak és fájlok meghatározása. Oszlopokba szeparált listában.
- -F, --no-sandbox - Nem védi le a telepítést sandboxszal.
- -e, --expand-sandbox - Alapértelmezés szerint a sandbox az aktuális '.' könyvtárhoz relatíven telepít. Használj 1-et e paraméternél, hogyha a szülőkönyvtárhoz ('..') akarsz viszonyítani, 2-t ha a '../..' -hez, és így tovább.
- -l, --allow-leftovers - UnionFS használata esetén nem tér vissza hibakóddal, sandboxon kívül eső állományok esetén.

Scripts

A GoboLinux szkriptjeinek listája

Csomagkészítés

- Compile - Automatizált programfordító eszköz.
- MakeRecipe - Recipesablon készítése.
- EditRecipe - Recipe szerkesztése.
- GenRecipeStore - „Csomagolt recipéket” készít, beküldés céljára.
- CreatePackage - GoboLinux csomag készítése.
- GetRecipe - Lehív egy recipét, és beilleszti azt a recipe-fába.
- FetchArchive - Megadva neki egy recipét letölti a fordításhoz szükséges állományokat.
- NewVersion - Új verzióra frissíti a recipét.
- PrepareProgram - Előkészíti az alkalmazást a telepítéshez, a 'configure'-ot futtatva.
- SymlinkProgram - Belinkeli a programot a /Programs hierarchiából a System-fába.
- CompileProgram - Compilation wrapper.
- UpdateRecipes - A helyi recipelistát frissíti a recipetárolókból.

Segédeszközök

- DeduceName - Megvizsgálja a file-ot, hogy megtalálja a programnév nagybetűsített változatát.
- FiboSandbox - A program védett környezetben futtatása, lekorlátozott felhasználóként.
- FindPackage - Csomagok és recipék keresése, helyileg vagy távoli eléréssel.
- FixInfo - Újraépíti az info tartalomjegyzékek bejegyzéseit.
- FixLibtoolLa - Convert a .la file created by libtool to make it GoboLinux-compliant.
- NamingConventions - Heurisztikusan meghatározza a nagybetűsített, GoboLinux-kinézetű nevet.
- SandboxInstall - Futtatja a 'make install' parancsot, elkülönített környezetben.

Csomagkezelés

- Freshen - Frissítések után kutat a GoboLinux csomagok/recipek

között.

- CreatePackage - GoboLinux csomag készítése.
- InstallPackage - GoboLinux csomag telepítése.
- DisableProgram - Megszünteti a program linkjeit a /System/Links hierarchiában.
- RemoveProgram - Eltávolítja a program verzióját a rendszerből.
- DetachProgram - Átmozgatja a programot a /Programs -ből egy másik helyre.
- Dependencies - Megvizsgálja egy GoboLinux csomag függőségeit.
- AllDependencies - Megmutatja egy csomag teljes függőségi hierarchiáját.
- MergeTree - Egy tartalomjegyzék-struktúrát egy másikba tükröz.
- RebuildLinks - Újraépíti a /System/Links tartalomjegyzékeket.
- KeyManager - A GoboLinux csomagok digitális aláírásainak kezelése.
- Hashes - A GoboLinux csomagban levő FileHash és FileHash.sig kezelése.

Kernel menedzselés

- CleanModules - Minden eltávolítható kernelmodul törlése.
- CreateInitRD – Saját LiveCD készítéséhez

Rendszerkarbantartás

- RemoveBroken - Törött linkek eltávolítása.
- AddUser - Új felhasználó felvétele a rendszerbe.
- EditScript - GoboLinux szkript készítése vagy szerkesztése.
- SystemFind - Speciális 'find' utiliti a /System hierarchiában való kereséshez.
- SystemInfo - Kiír néhány alapvető rendszerinformációt. Hasznos az /etc/issue számára.

Utilitik

- FilterColors - Monokrómmá szűri a standard inputot (tkp annyit csinál, hogy eltávolítja belőle a színekért felelős terminál-escape-kódokat).
- FilterLines - Sorok szűrése annak alapján, hogy tartalmaznak-e vagy sem bizonyos kifejezéseket.
- FixAttributes - Állományok attribútumainak beállítása a tartalmuktól függően.
- GrepReplace - Szövegek cseréje másra állományok egy sorozatán.

- IsExecutable - Meghatározza, a fájl végrehajtható-e.
- RemoveEmpty - Minden üres tartalomjegyzék törlése a current-ből (vagy a megadott tartalomjegyzékből).
- RenameCaps - Fájlok átnevezése, közben nagybetűsítve azokat.

Összevont rutinok

- FindQuick (alias: f) - Fájl keresése.
- GrepQuick (alias: g) - Előtétprogram a 'grep'-hez színes kimenettel, csökkentett szintaxissal.
- List (alias: l) - Kitűnő listázórutin, frontend az "ls"-hez.
- KillProcess - Processzek kilövésének utilitija.

Boot szkriptek

BootDriver - A betöltést vezérlő szkript.

StartTask - Indítási taszk futtatása parancssorból.

StopTask - Indítási taszk leállítása parancssorból.

TestBootTheme - Boot téma megtekintése a rendszer újraindítása nélkül.

Speciális állományok

GoboPath - különböző shell változókat határoz meg

Shutdown

Előfordulhat, hogy a **reboot** működik a GoboLinux rendszered alatt (**shutdown -r now** parancs pld) de a rendszer leállítása (**shutdown -h now**) már nem! A rendszer látszólag kilép, de a számítógép nem áll le, a disk pörög tovább...

Ebben az esetben nézd meg a **grub** bootolási paramétereit a **menu.lst** fájlban! Ez a **/System/Kernel/Boot/grub/menu.lst** helyen található.

Valami efféle részt keress, aholis meghívod a rendszert (e példa a nem grafikus indítást mutatja, de ez mindegy most):

```
title GoboLinux - Console (hda4)
kernel (hd0,3)/System/Kernel/Boot/kernel acpi=off root=/dev/hda4 vga=0
```

Na a lényeg: minden sorból, ahol az „**acpi=off**” szerepel, *töröld azt ki!* Azaz, a fenti példa így néz majd ki ezután neked:

```
title GoboLinux - Console (hda4)
kernel (hd0,3)/System/Kernel/Boot/kernel root=/dev/hda4 vga=0
```

Ezután már elvileg működnie kell. (Nálam működött). Természetesen ha megváltoztattad e sorokat, akkor ezután rögtön még nem áll le a rendszered - előbb azt indítsd újra, s ezután már elvileg le is áll, ha ezt parancsolod neki!

SignProgram

Hash fájlt generál egy GoboLinux csomaghoz, és alá is írja azt gpg-vel.

Használat: **SignProgram** { [**<opciók>**] **<csomag>** [**<verzió>**] }

Opciók:

- **-h, --help** - Megjeleníti ezt a leírást.
- **-v, --version** - Kiírja a program verzióját.
- **-V, --verbose** - Bőbeszédű üzemmód engedélyezése.
- **-u, --local-user <entry>** - Az **<entry>** használata user ID-ként aláíráshoz.
- **-S, --no-signature** - Csak FileHash előállítása. Nincs GPG aláírás.

Megjegyzés:

Ha nem adunk meg verziót, a Current használtatik. Ha nem használjuk az **-u** opciót, a kulcskészletben talált első ID lesz felhasználva.

StartTask

A taszkok tulajdonképpen bizonyos speciális csomagok, az elérhető taszkokat a **/System/Links/Tasks** tartalmazza.

A taszk indítása így történik:

StartTask TaszkNeve

A taszkok úgy vannak megírva, hogy automatikusan kezelik a képernyőt, ezzel feleslegessé téve az "&" operátor használatát az őket meghívó parancsban.

StopTask

A **StartTask**, **StopTask** parancsoknak a szintaxisa ugyanaz. A "**StopTask <TaskName>**" a futó TaskName taszkot állítja le.

Swap

Swap partíció kibővítése ideiglenesen swap fájlal.

Itt a példa rá, hogyan kell 2 gigával kibővíteni az átmenetileg esetleg szűkösen bizonyuló swap partíciónkat (jól jöhet például az OpenOffice forrásból való fordításánál):

```
dd if=/dev/zero of=/swapfile bs=1024 count=2097152
mkswap swapfile
swapon swapfile
```

(A „2097152” egy kilobájtban megadott szám).

SymlinkProgram

Belinkeli a programot a **/Programs** hierarchiából a **/System** struktúrába.

Használat: **SymlinkProgram** [**<opciók>**] **<programnév>** [**<programverzió>**]

Opciók:

- -h, --help – Megjeleníti ezt a helpet.
- -v, --version – Kiírja a program verzióját.
- -V, --verbose – Bőbeszédű üzemmód engedélyezése.
- -s, --settings <entry> - A beállítások belinkelése a **/System/Settings** -be:

- 'yes', 'no' vagy 'safe'. Az alapértelmezett a 'yes' (igen).
- -l, --libraries <entry> - A libraryk belinkelése a **/System/Links/Libraries** -be: 'yes', 'no' vagy 'safe'. Az alapértelmezett a 'yes' (igen).
- -e, --executables <entry> - A végrehajtható állományok belinkelése a **/System/Links/Executables** -be: 'yes', 'no' vagy 'safe'. Az alapértelmezett a 'yes' (igen).
- -d, --daemons <entry> - A démonok belinkelése a **/System/Links/Daemons** -ba: 'yes', 'no' or 'safe'. 'yes', 'no' vagy 'safe'. Az alapértelmezett a 'yes' (igen).
- -h, --headers <entry> - Fejlécek belinkelése a **/System/Links/Headers** -be: 'yes', 'no' vagy 'safe'. Az alapértelmezett a 'yes' (igen).
- -a, --shared <entry> - Shared állományok belinkelése a **/System/Links/Shared** -be: 'yes', 'no' vagy 'safe'. Az alapértelmezett a 'yes' (igen).
- -w, --wrappers <entry> - Wrapperek belinkelése a **/System/Links/Executables** -be: 'yes', 'no' vagy 'safe'. Az alapértelmezett a 'yes' (igen).
- -t, --tasks <entry> - A taszkok belinkelése a **/System/Links/Executables** -be: 'yes', 'no' vagy 'safe'. Az alapértelmezett a 'yes' (igen).
- -E, --no-environment - Nem linkeli be a környezeti változókat a **/System/Links/Environment** -ből.
- -A, --no-variable - Nem mozgatja át a változó állományokat a **/System/Variable** -ba.
- -M, --no-doc - Nem linkeli be a manuálokat és info fájlokat.
- -S, --no-strip - Nem tömöríti strip-pel a végrehajtható állományokat.
- -X, --cross-compile - Keresztfordítás: A GoboPath export könyvtárakat a **\$crossPrefixDir** változóból veszi.
- -C, --cleanup - Tisztítás a telepítés után.
- -c, --conflict <entry> - Mi a teendő konfliktus esetén: 'keep' (megtart) vagy 'overwrite' (felülír). Az alapértelmezett érték a 'keep' (megtart).
- -n, --no-make - Üres opció. Visszamenőleges kompatibilitás megőrzésére fenntartva.
- -r, --relative - Relatív útvonalak használata a **/Programs** fájljainak belinkelésekor.
- -t, --rootfs - A program másolása a rootfs-be, ha az egy symlink.

Ha programverziót nem adunk meg, a **Current** lesz használva.

Példa:

SymlinkProgram WeirdSuperLib 2.4

SystemFind

A **SystemFind** egy speciális 'find' utility a **/System** hierarchiában való kereséshez.

Egy szokásos Linux disztróban a binárisok, libraryk vagy a fejlécek sokféle helyen előfordulhatnak. A GoboLinuxban azonban rendelkezünk a **/System/Links** fával, amely természetes indexként funkcionál a számunkra: minden bináris megtalálható a **/System/Links/Executables** alatt, minden library a **Libraries**-ben, stb. Azt megtudandó, hogy egy program pontosan hol van (más szavakkal, melyik csomagnak is a része az), elegendő kiadni az **"l"** parancsot (a **List** alias-a, és az **ls -l** ekvivalense) és máris látjuk, hogy hová is mutat a link. Végrehajtható állományok számára a **which** parancs még kényelmesebb.

A **/System/Links** fa „fájlrendszer index”-ként való felhasználásának ötletén alapul az, hogy van nekünk egy **SystemFind** szkriptünk, mely végigvizsgálja azt, és közvetlenül a belinkelt fájjal tér vissza. A **/System/Links** -et végigvizsgálni sokkal gyorsabb, mint a **/Programs** fa végigkutatása: már amiatt is, mert kevesebb tartalomjegyzéket kell vizsgálni (és legalábbis a **ReiserFS** fájlrendszerben, a symlinkek ugyanabban az inode-ban tárolódnak, mint a tartalomjegyzék, helyet és időt takarítva meg ezáltal).

Továbbá, minthogy a **/System/Links/Libraries** és **/System/Links/Executables** tartalomjegyzékek gyakran használatnak, ezek általában amúgyis a lemezgyorsító tárban vannak.

Használat: **SystemFind** [**<flagok>**] **<keresési_minta>**

Opciók:

- **-h, --help** - Megjeleníti ezt a leírást.
- **-v, --version** - Kiírja a program verzióját.
- **-V, --verbose** - Bőbeszédű üzemmód engedélyezése.
- **-e, --executables** - Végrehajtható állományok keresése a **/System/Links/Executables** -ben.
- **-l, --libraries** - Libraryk keresése a **/System/Links/Libraries**-ben.
- **-i, --headers** - Fejlécek keresése a **/System/Links/Headers**-ben.
- **-s, --settings** - Beállítások keresése a **/System/Settings**-ben.
- **-m, --manuals** - Manuálok keresése a **/System/Links/Manuals**-ban.
- **-q, --quick** - Alapértelmezésként, a rutin a tényleges állományokat is megvizsgálja, melyek a szimbolikus linkek által mutatott tartalomjegyzékekben vannak. Ha tudod, hogy az általad keresett állomány a tartalomjegyzék-hierarchia felső szintjén helyezkedik el (man oldalak, végrehajtható állományok, C libraryk, stb.), úgy a **--quick** használata azáltal, hogy a tartalomjegyzékekre mutató szimbolikus linkeket nem vizsgálja át teljesen, jelentősen felgyorsíthatja a kereséset. (Megjegyzendő, hogy ez volt a **SystemFind** régebbi verzióinak alapértelmezett viselkedése.)

Ha nem adunk meg flagot, a rendszer összes helye megvizsgáltatik.

Példa:

SystemFind -i -l freetype

SystemInfo

Néhány alap információt jelenít meg a rendszerről. A `/etc/issue` számára lehet hasznos.

Használat: **SystemInfo**

Opciók:

- `-h, --help` - Ezt a help-et mutatja.
- `-v, --version` - A program verzióját írja ki.
- `-V, --verbose` - Bőbeszédű mód.

Futtatásának nálam például ez az eredménye:

```
vz@Csiszilla ~]SystemInfo
System.....: Linux 2.6.23.8-Gobo i686
Compilation...: #2 SMP PREEMPT Tue Nov 20 01:38:04 PST 2007
Processor.....: Intel(R) Celeron(R) CPU 2.40GHz
  Clock.....: 2459.177 MHz
  Bogomips...: 4920.90
Memory.....: 509836k
```

TestBootTheme

E szkript használatának leírását a „**Boot téma tesztelése**” című fejezetben találsz meg.

Thunderbird

Thunderbird telepítése és beállítása magyar nyelvűre

Ezt a cikket amiatt írom, mert iszonyat mennyit „szívtam” a Thunderbird magyar nyelvűre állításával! Holott egy aprócska dologon múlik az egész...

A telepítés csak annyi, hogy

Compile thunderbird

Ez jó 25 mega anyag letöltését jelenti, és sokáig fordul, azaz ezalatt

nyugodtan teázhatunk egyet...

Ezután a magyarítás következik.

Menjünk el erre a linkre:

<http://releases.mozilla.org/pub/mozilla.org/thunderbird/releases/2.0.0.0/linux-i686/xpi>

és itt töltsük le a **hu.xpi** fájlt.

FIGYELEM! Ha Firefoxot használunk **NE** kattintsunk rá az egér bal gombjával, mert akkor a Firefox azonnal önmagába akarja azt telepíteni! Töltsük le, majd a Thunderbirdot nyissuk meg. Ekkor még angolul gagyarászik természetesen. A **Tools** menü **Add-ons** menüpontjában telepítsük az előbb letöltött **hu.xpi** állományt. Ezután lépünk ki a Thunderbirdból, és indítsuk újra - de **PARANCSSORBAN!** (Terminál-emulátorban) a következő paranccsal:

thunderbird -UILocale hu-HU

Ekkor már magyarul fog beszélni, sajnos azonban ha kilépünk belőle és a grafikus menüből újraindítjuk, angolul fecseg megint. Tehát: ha kilépünk belőle, **ne** indítsuk újra a grafikus menüből, hanem az egész rendszert indítsuk újra! Azaz: **reboot** a nekünk tetsző módon. Tiszta Windows-érzés... De sajnos ez van. Aki tud valami előkelőbb megoldást, írja meg nekem!

Ezután már nálam megmaradt magyarnak...

UpdateRecipes

Frissíti a helyi recipelistát a recipetárolókból.

Használat: **UpdateRecipes** [<program>]

Opciók:

- -h, --help - Megjeleníti ezt a leírást.
- -v, --version - Kiírja a program verzióját.
- -V, --verbose - Bőbeszédű üzemmód engedélyezése.
- -a, --all - Letölti a frissített recipék tartalmát. Alapértelmezés szerint az UpdateRecipes csak a recipelistát tölti le, és üres recipe tartalomjegyzékeket generál (kivéve ha egyetlen programot frissítünk).
- -t, --thorough - Minden elérhető tükrözést átkutat frissítések után. Alapértelmezés szerint ugyanis csak az első aktív tükrözést használja (mely a Compile.conf-ban van lekonfigurálva).

Egyetlen program frissítésénél az **UpdateRecipes** az összes elérhető recipét

letölti. Ha nem adtunk meg neki programspecifikációt, az UpdateRecipes letölti a recipelistát, és létrehozza a `/Files/Compile/Recipes` -ben a megfelelő tartalomjegyzék-bejegyzéseket (de magukat a recipéket csak akkor tölti le, ha a `--all` kapcsolót használjuk).

Új kernel installálása

A **Compile**nek köszönhetően a GoboLinux alatt elég egyszerű egy új kernel telepítése. A **Linux** recipe megvizsgálja a `config.gz` létezését a `/System/Kernel/Status` könyvtárban. Ez a file tartalmazza a futó kernel aktuális konfigurációját, és új kernel opciók megadására használják.

Röviden, a
Compile Linux

parancs futtatása letölti a legutolsó elérhető recipét (amely már tartalmazza az opcionális GoboLinux kernelfoltokat). Miután ez megtörtént, a kernel automatikusan letöltődik, belékerülnek a foltok, és be lesz állítva a új konfigurációra, amit a `config.gz` állományból olvas ki.

Ekkor megjelenik a `menuconfig` képernyő, lehetőséget adva a felhasználónak, hogy módosítsa a beállított opciókat. „Exit”-et választván, s azt az opciót hogy mentse a szkript a beállításokat, befejeződik a kernel fordítása. Amint ez kész van, új bejegyzést találunk a `/System/Kernel/Modules/$KERNEL_RELEASE` cím alatt, és egy új `bzImage` és `System.map` fájlt a `/System/Kernel/Boot` könyvtárban.

A régi `bzImage` és `System.map` fájl nem íródik felül. Ők symlinkek az aktuális kernel image-re, ezáltal garantálva, hogy ha valami „zűr” van, egyszerűen vissza lehessen térni a korábbi állapothoz, a **GRUB** betöltésvezérlő promptjánál, és később visszaállítani a szimbolikus linket a régebbi rendszermagra.

Szenzorok

Mindenekelőtt le kell fordítsuk az LM-Sensors progít:

Compile lm-sensors

Ezután ki kell adjuk a **sensors-detect** parancsot, s minden kérdésére igennel, azaz yes-sel (**Y**) válaszolni. Nálam ez efféleképp ment:

```
pv@Csiszilla /Programs/LM-Sensors/Settings]sensors-detect
# sensors-detect revision 5108 (2008-01-22 13:22:47 +0100)
```

This program will help you determine which kernel modules you need to load to use lm_sensors most effectively. It is generally safe and recommended to accept the default answers to all questions, unless you know what you're doing.

```
We can start with probing for (PCI) I2C or SMBus adapters.
Do you want to probe now? (YES/no): y
Probing for PCI bus adapters...
Use driver `i2c-i801' for device 00:1f.3: Intel 82801EB ICH5
```

```
We will now try to load each adapter module in turn.
Module `i2c-i801' already loaded.
If you have undetectable or unsupported adapters, you can have them
scanned by manually loading the modules before running this script.
```

```
To continue, we need module `i2c-dev' to be loaded.
Do you want to load `i2c-dev' now? (YES/no): y
Module loaded successfully.
```

```
We are now going to do the I2C/SMBus adapter probings. Some chips may
be double detected; we choose the one with the highest confidence
value in that case.
If you found that the adapter hung after probing a certain address,
you can specify that address to remain unprobed.
```

```
Some chips are also accessible through the ISA I/O ports. We have to
write to arbitrary I/O ports to probe them. This is usually safe though.
Yes, you do have ISA I/O ports even if you do not have any ISA slots!
Do you want to scan the ISA I/O ports? (YES/no): y
Probing for `National Semiconductor LM78' at 0x290...      No
Probing for `National Semiconductor LM78-J' at 0x290...   No
Probing for `National Semiconductor LM79' at 0x290...     No
Probing for `Winbond W83781D' at 0x290...                 No
Probing for `Winbond W83782D' at 0x290...                 No
Probing for `Silicon Integrated Systems SIS5595'...       No
Probing for `VIA VT82C686 Integrated Sensors'...          No
Probing for `VIA VT8231 Integrated Sensors'...            No
Probing for `IPMI BMC KCS' at 0xca0...                    No
Probing for `IPMI BMC SMIC' at 0xca8...                   No
```

```

Some Super I/O chips may also contain sensors. We have to write to
standard I/O ports to probe them. This is usually safe.
Do you want to scan for Super I/O sensors? (YES/no): y
Probing for Super-I/O at 0x2e/0x2f
Trying family `National Semiconductor'...           No
Trying family `SMSC'...                             No
Trying family `VIA/Winbond/Fintek'...               Yes
Found `Winbond W83627HF/F/HG/G Super IO Sensors'     Success!
    (address 0x290, driver `w83627hf')
Probing for Super-I/O at 0x4e/0x4f
Trying family `National Semiconductor'...           No
Trying family `SMSC'...                             No
Trying family `VIA/Winbond/Fintek'...               No
Trying family `ITE'...                               No

```

```

Some CPUs or memory controllers may also contain embedded sensors.
Do you want to scan for them? (YES/no): y
AMD K8 thermal sensors...                           No
AMD K10 thermal sensors...                           No
Intel Core family thermal sensor...                  No
Intel AMB FB-DIMM thermal sensor...                  No

```

Now follows a summary of the probes I have just done.
Just press ENTER to continue:

```

Driver `w83627hf' (should be inserted):
Detects correctly:
* ISA bus, address 0x290
  Chip `Winbond W83627HF/F/HG/G Super IO Sensors' (confidence: 9)

```

```

Do you want to generate /etc/sysconfig/lm_sensors? (yes/NO): y
Copy prog/init/lm_sensors.init to /etc/init.d/lm_sensors
for initialization at boot time.
You should now start the lm_sensors service to load the required
kernel modules.

```

Ezután meg kell keresnünk a `/System/Settings/sysconfig/lm_sensors` fájlt, ki kell listáznunk (`cat lm_sensors`) és megnézni, milyen modulokat írt bele, mint amelyek kellenek neki. Nálam ez ebből a sorból látszik:

```
MODULE_0=w83627hf
```

Azaz, ez a sor van a file legalján. Tehát be kell töltsen ezt a modult:

```
pv@Csiszilla ~]modprobe w83627hf
```

Ezután a `sensors` paranccsal kilistáztathatom a szenzorok adatait:

```

pv@Csiszilla ~]sensors
w83627hf-isa-0290
Adapter: ISA adapter
VCore 1:      +1.50 V  (min =  +0.00 V, max =  +4.08 V)
VCore 2:      +2.61 V  (min =  +0.00 V, max =  +4.08 V)

```



```

+3.3V:      +3.31 V (min = +2.82 V, max = +3.79 V)
+5V:        +5.08 V (min = +0.05 V, max = +0.86 V)
+12V:       +12.16 V (min = +0.00 V, max = +0.00 V)
-12V:       -7.01 V (min = -14.91 V, max = -14.91 V)
-5V:        -3.69 V (min = -7.71 V, max = -7.71 V)
V5SB:       +5.43 V (min = +0.22 V, max = +0.86 V)
VBat:       +3.25 V (min = +0.26 V, max = +0.00 V)
fan1:       2721 RPM (min = 332 RPM, div = 16)
fan2:       0 RPM (min = 332 RPM, div = 16)
fan3:       0 RPM (min = 332 RPM, div = 16)
temp1:      +25.0°C (high = +0.0°C, hyst = +0.0°C) sensor = thermistor
temp2:      +35.5°C (high = +85.0°C, hyst = +80.0°C) sensor = diode
temp3:      +28.0°C (high = +85.0°C, hyst = +80.0°C) sensor = thermistor
cpu0_vid:   +0.000 V
beep_enable:enabled

```

Ha azt akarjuk, hogy e modul(ok) automatikusan betöltődjék(betöltődjenek) induláskor, írjuk be a

/Programs/BootScripts/Settings/BootOptions fájlba, az **UserDefinedModules** részbe!
Nálam a kérdéses részt így fest most – félkövérrel kiemelem a szenzormodul nevét:

```

UserDefinedModules=(
    snd-pcm-oss
    snd-mixer-oss
    snd-intel8x0
    w83627hf
)

```

ChrootCompile

A **ChrootCompile** egy olyan eszköz, mely elkülönített végrehajtási környezetet biztosít a **Compile** futtatásához. Ezen elkülönített környezet (egy „chroot” rendszer) három fő célt szolgál:

- lehetővé teszi csomagok beépítését más eszközöket használva, mint amivel a jelenleg futó rendszerben rendelkez,
- megoltalmazza a rendszert minden problémától, mely a fordítási folyamat során felmerülhet,
- és segít neked megbizonyosodni afelől, hogy a recipéd függőségi információi helyesek.

Ha egy recipét futtatsz a ChrootCompile segítségével, az ki kell olvassa a függőségi listáját, és ezen alapulva létrehoz egy tartalomjegyzéket egy minimális, de még tökéletesen dolgozó GoboLinux-rendszerrel, mely tartalmazza a Compile eszköz számára szükséges alapsomagokat (mint például egy shell, egy fordító és az alap Unix parancsok) és a recipéd függőségi listájában felsoroltakat – semmi egyebet. Ekkor kell lefusson a recipéd a Compile-on keresztül ebben az elkülönített rendszerben, és, ha minden rendben zajlott le, ennek eredménye a folyamat végén a megfelelő bináris csomag lesz.

Eképp csomagokat készíthetsz – például – a GCC és Glibc különböző verzióhoz, mint amit a rendszerben épp használsz.

Az alapértelmezett konfigurációjában a ChrootCompile az ő alapértelmezett csomagkészletét a <http://rsync.gobolinux.org/chrootcompile-packages/> címen levő chrootcompile-csomagtárból szerzi be, eképp biztosítva a csomagok beépítésének konzisztenciáját. E lépést egy külön szkript, a SetupChrootEnv hajtja végre.

A SetupChrootEnv készíti el azt a környezetet, amiben a ChrootCompile futni tud (amennyiben üres tartalomjegyzékben hívjuk meg, úgy a ChrootCompile első lépésként automatikusan meghívja a SetupChrootEnv szkriptet). E környezet kell tartalmazza a chroot rendszer elkészítéséhez szükséges csomagokat. **A ChrootCompile minden meghívásakor egy új chroot rendszer készítettetik.** Ez biztosítja, hogy az előző fordítás eredményei ne befolyásolják a következő fordítást. A ChrootCompile fő célja reprodukálható fordítások készítése: mindegy, hogy két számítógép konfigurációja mennyire különbözik, ha ugyanazzal a recipével futtatjuk a ChrootCompile-ot, ideális esetben az ugyanazt az eredményt kell produkálja.

Ezen reprodukálhatóság fontos eszköze az, hogy **a ChrootCompile által végzett fordítás nem rendelkezik hálózati hozzáféréssel.** A fordításhoz szükséges összes fájl rendelkezésre kell álljon, vagy a SetupChrootEnv szkript (alapeszköz), vagy a felhasználó által (recipék és egyéb csomagfüggőségek). A forráscsomagok tehát még a chroot-ba belépés előtt letöltendőek. Ezúton ki-kényszeríthetjük annak világos ellenőrzését, mely csomagokat használja ez a

fordítási folyamat, ahelyett, hogy megelégednénk a Compile szokásos futtatásakor a FindPackage által nyújtott lehetőségekkel. Továbbá, távoli rendszerek változásai (fájlok szervereken, CVS frissítések, stb.) nem befolyásolják a fordításainkat.

A ChrootCompile használata nemcsak annak ellenőrzését teszi lehetővé, mi épül be a csomagjainkba, de annak ellenőrzését is, mi az, ami kimarad. Alkalmadtán egy fordítási folyamat sikeres lehet az élő rendszerben, mert ott létezik egy csomag, ami nincs felsorolva a függőségi listában, vagy mert valami trükkkel megkerülik a csomagfüggőséget, amit elfelejtett a recipe készítője. Ezután e recipét elküldik a recipetárolóba, ám az képtelen lesz működni más rendszerekben melyek nélkülözik e csomagot vagy trükköt (melyet a recipe csomagfüggőségébe kellett volna beépíteni).

A ChrootCompile által **fordítások nem sikerülhetnek véletlenül**. Egy olyan recipe, mely működik a ChrootCompile alatt, sokkal nagyobb eséllyel működik bármely más rendszerben is, mely szabályos Compile-val rendelkezik – és bizonyosan működik ChrootCompile alatt ugyanazt a függőségi listát alkalmazva.

A ChrootCompile kulcsfontosságú volt a GoboLinux 013 fejlesztése közben. Gyakorlatilag minden csomag ezt használva lett lefordítva (az OpenOffice.org, Firefox és Thunderbird figyelemre méltó kivételével, melyek az ő hivatalos binárisaikból lettek elkészítve). Ez tette lehetővé a fejlesztőcsapatnak, hogy sokkal könnyebben reprodukálják a recipék hibáit, egy élő rendszerben való futtatás kockázata nélkül, és legfőképpen, hogy számos recipe függőségi listáját gondosan kijavíthassák.

A ChrootCompile a GoboLinux rendszerben a következőt begépelve installálható:

InstallPackage ChrootCompile

Ismert hibák a GoboLinux 014-ben

Amióta a GoboLinux 014 megjelent, felfedeztek néhány kisebb jelentőségű hibát benne, részben a szorosan vett disztribúcióban, részben az őt alkotó csomagokban. Ezek a következők:

Frissített csomagok

A GoboLinux 014 kibocsátása óta a felfedeztünk néhány hibát egyes csomagokban, melyeket az itt felsorolt frissítésekben kijavítottunk. Miután telepítetted a LiveCD-t, telepítsd e csomagokat is az **InstallPackage** paranccsal. Például az Xorg frissítéséhez futtasd az

InstallPackage xorg

parancsot a parancssorból, s erre az letölti és telepíti a legutóbbi verziót.

(Megjegyzendő: Xorg frissítését *nagyon nem ajánlott* „elkövetni” virtuális terminálból, például az Mrxvt alatt – ehhez tehát mindig **lépjünk ki** a „**mélykonzolba**”, azaz oda, ahonnet a grafikus felületet a „**startx**” paranccsal indítani szoktuk!)

Xorg 7.2-r6 - *LibXCB beállítási hibákat javít.*

Frissített recipék

A következő frissített recipék a GoboLinux 014 kiadása óta felfedezett hibákat javítják. Hamarosan meg kell érkezzenek az ezekből készített csomagok is. Addig is használhatod a **Compile** eszközt a verziófrissítésre.

DBus 1.1.2-r3 - *Rendszer-id problémát javít.*

AM2 gond a SquashFS-sel

Úgy tűnik van egy hiba a SquashFS-ben, mely a kétmagos AM2 AMD64 rendszerek esetén okoz problémát. Ezek a rendszerek is elindulnak azonban, ha a következő parancsot adod ki a CD rendszerbetöltő promptjánál:

```
/isolinux/kernel initrd=initrd init=/linuxrc ramdisk_size=4096 root=/dev/ram0  
video=vesafb:off noapic nolapic nosmp apic=off Boot=LiveCD max_loop=32
```

(A fenti 2 sor egyetlen parancs, azaz 1 sorba írandó!)

Ez a hiba nem jelentkezik a feltelepített rendszernél, mely SMP/HT/AM2 -vel működik.

Egyéb problémák

Ismert, hogy a LiveCD-nek problémái akadtak a VirtualBox OSE emulátor alatt bootoláskor. Ezt fixálták a disztribúció fejlesztői változatában. Az erről szóló tudnivalókat e link alatt tekintheted meg:

<http://bugs.gobolinux.org/view.php?id=232>

Saját programokat is írhatasz...

Most elárulom neked – bár lehet hogy eddigre már rájöttél – hogy a forrásalapú disztrók – s a GoboLinux is ilyen! - egyik nagy előnye, hogy szükségszerűen fel vannak szerelve minden szükséges cuccal, mely kell ahhoz, hogy egy programot lefordítsunk. Na de ebből máris következik, hogy a fordítóprogramnak teljesen mindegy, „gyári” programot fordít-e le, vagy olyasmit, amit te írtál! Azaz, írhatasz saját programokat is ha akarsz, és ehhez egyáltalán semmi külön miskulanciát nem kell feltelepítened – minden szükséges máris rajta van a gépeden!

Például ha tudod, miként kell programozni C vagy C++ nyelven, elég, ha csak megírod a progidat a kedvenc szövegszerkesztőddel egy fájlba, aztán a parancssorban elmész abba a könyvtárba, ahová lementetted ezt, s kiadod a következő parancsot:

```
g++ s.cpp -o s
```

Ennek magyarázata: a **g++** -al indítod a fordítóprogramot, az **s.cpp** az az állománynév, melyen elmentetted a programot (ez csak egy példa, természetesen más néven is lementheted), **-o s** pedig azt mondja meg, hogy az „output file”, vagyis a lefordított bináris progí neve egy sima **s** legyen. Kicsit általánosabban egy fordítási utasítás paraméterezése így néz ki:

```
g++ forrásprogramfile_neve -o bináris_neve
```

Természetesen a fordítás után futtathatóvá kell tenni a létrejött állományt ahhoz hogy kipróbálhasd, ez megtehető például a fenti példa esetében mondjuk a

```
chmod 777 s
```

paranccsal.

Na de elképzelhető, hogy te épp nem C/C++ nyelven óhajtasz programozni! Nos, örömmel közölhetem, hogy ha eddig végigkövetted az e könyvben írtakat, akkor lehet már némi sejtelked róla, hogy számos programnyelv szerepel a rendszeredben: ha értesz hozzá, írhatasz progikat a következő nyelvek bármelyikén:

Assembler, C, C++, Html, Java, Python, Perl, Sed, Tcl/Tk

Természetesen bármely neked tetsző újabbat is feltelepíthetsz, de ezek, a legismertebbek, máris telepítve vannak neked! Tehát minden csak rajtad múlik!

Vimperator

A „vimperator” nem egy külön program, hanem egy Firefox „kiegészítés”, tulajdonképpen plugin, ami erről az oldalról tölthető le:

<http://vimperator.mozdev.org/>

E plugin által **VIM**-szerű billentyűzetparancsokkal lehet vezérelni a böngészőnket! Ez **.xpi** formájában installálható a Firefoxhoz a fent megadott oldalról.

Gondoltam lefordítom nektek a legfontosabb parancsait. Nem árthat ez egy olyan könyvben, mely „billentyűzetfüggő Kockáknak” íródik! Az itt felsorolt parancsok korántsem az összes képességei a Vimperatornak, de talán a legfontosabbak. Mindenesetre ezzel már „el lehet indulni” a használatában. Íme:

Parancs	Leírás
:set guioptions+=mT	Visszahozza neked a menüket és egyéb ilyesmit, ami alapesetben el van tüntetve. Ha ezt beírod egy \$HOME/.vimperatorrc fájlba, akkor ez lesz az alapbeállításod is. Kezdőknek jól jöhet.
?szöveg	keresés visszafelé az oldalon. Utána nyomjunk Entert.
?szöveg?<CR>	keresés visszafelé az oldalon, sorvégjellel együtt. Utána nyomjunk Entert.
/szöveg	keresés előre az oldalon. Utána nyomjunk Entert.
/szöveg/<CR>	keresés előre az oldalon, sorvégjellel együtt. Utána nyomjunk Entert.
\$	A dokumentum jobbszélső részére ugrik.
0	A dokumentum balszélső részére ugrik.
d vagy :bdelete	Az aktuális fül bezárása.
F1 vagy :h vagy :help	Help
G	A weboldal legvégére ugrik.
10G	10%-al megy lejjebb az aktuális weboldalban.
gg	Az aktuális weboldal legtetéjére ugrik.
10gg	10%-al megy feljebb az aktuális weboldalban.
gH	Ugrás a kezdőlapra a következő fülön
gh	Ugrás a kezdőlapra az aktuális fülön
gn{a-zA-Z0-9}	Ugrás a gyorsbillentyűvel megjelölt linkre a következő fülön
go{a-zA-Z0-9}	Ugrás a gyorsbillentyűvel megjelölt linkre az

Parancs	Leírás
	aktuális fülön
gP	A vágólapon levő linket megnyitja egy új fülben, de nem adja át a fókuszt erre a fülre. Hasonló, mint amikor az egér középső gombjával kattintunk egy linkre.
gT	Ugrás az előző fülre
3gT	Ugrás 2 fület visszafelé. A számlálás körkörös: az első fül után az utolsó következik.
gt	Ugrás a következő fülre
3gt	Ugrás a harmadik fülre
gU	Ugrás a weboldal gyökérkönyvtárába. Például a <code>gU</code> parancs a http://www.example.com/dir1/dir2/file.html esetén az http://www.example.com/ linkre visz bennünket.
gu	Ugrás a weboldal szülőkönyvtárába.
H	Ugrás visszafelé a meglátogatott lapok listájában.
3H	Ugrás visszafelé 3 lapot a meglátogatott lapok listájában.
h	Balra scrollozás.
10h	Balra scrollozás 10 egységgel.
j	Lefelé scrollozás.
10j	Lefelé scrollozás 10 sorral.
k	Felfelé scrollozás.
10k	Felfelé scrollozás 10 sorral.
L	Ugrás előre felé a meglátogatott lapok listájában.
3L	Ugrás előre felé 3 lapot a meglátogatott lapok listájában.
l	Jobbra scrollozás.
10l	Jobbra scrollozás 10 egységgel.
M{a-zA-Z0-9}	Gyorsbillentyű adása az aktuális linkhez
N	Az előző előfordulás keresése
n	A következő előfordulás keresése
o	URL megnyitása az aktuális fülön. Ez tulajdonképpen a vimperator parancssorába visz bennünket, ahová automatikusan beírja az :open parancsot, s ez után kell írunk az URL-t (vagy egy keresendő kifejezést, lásd az :open

Parancs	Leírás
	leírásánál).
P	A vágólapon levő URL-t megnyitja egy új fülben.
p	A vágólapon levő URL-t megnyitja az aktuális fülben.
R	Az aktuális oldal újratöltése, a cache megkerülésével (figyelembe nem vevésével).
r	Az aktuális oldal újratöltése
t	URL megnyitása egy új fülben. Ez tulajdonképpen a vimperator parancssorába visz bennünket, ahová automatikusan beírja az :tabopen parancsot, s ez után kell írunk az URL-t (vagy egy keresendő kifejezést, lásd a :tabopen leírásánál).
Y	A kiválasztott szöveget a vágólapra másolja
y	Az aktuális oldal linkjét a vágólapra másolja
zI	Az aktuális oldal nagyítása 100%-al
3zI	Az aktuális oldal nagyítása 3*100%-al
zi	Az aktuális oldal nagyítása 25%-al
3zi	Az aktuális oldal nagyítása 3*25%-al
z0	Az aktuális oldal kicsinyítése 100%-al
3z0	Az aktuális oldal kicsinyítése 3*100%-al
zo	Az aktuális oldal kicsinyítése 25%-al
3zo	Az aktuális oldal kicsinyítése 3*25%-al
ZQ vagy :qall	Kilépés a Firefoxból, az aktuális állapot mentése nélkül.
ZZ	Kilépés a Firefoxból, az aktuális állapot mentésével.
:	Belépés a parancsüzemmódba. Ebben az üzemmódban kibővített parancsokat használhatsz, melyek különböző paramétereket igényel(het)nek.
: addons	A Firefoxunkba installált kiegészítők (pluginek) megjelenítése egy külön fülön; tulajdonképpen a chrome://mozapps/content/extensions/extensions.xul „linket” nyitja meg.
:ba vagy :back	Eggyel visszalép a meglátogatott oldalak között.
:beep	Megszólaltatja a rendszer-jelzőcsengőt.
:fo vagy :forward	Eggyel előrelép a meglátogatott oldalak közt.

Parancs	Leírás
:hist	A legutóbb meglátogatott honlapok listáját adja elő teljes képernyőn. A megfelelő linkre rá kell kattintsunk az egérrel. Ezután a lista eltüntetés az Esc gombbal, hogy lássuk a lista mögött megjelenő honlapot.
:hist!	A Firefox szokásos listáját adja elő a meglátogatott honlapokról.
:open példa	Google keresés a „példa” szóra az aktuális fülön megnyitva.
:open vimperator help	Google keresés a „vimperator” ÉS a „help” szó együttes előfordulására az aktuális fülön megnyitva.
:open yahoo példa	Yahoo keresés a „példa” szóra az aktuális fülön megnyitva.
:open ebay példa	Ebay keresés a „példa” szóra az aktuális fülön megnyitva.
:pa vagy :pageinfo	Információ az adott oldalról.
:prefs	A Firefox Szerkesztés/Beállítások menüjét mutatja meg.
:prefs!	Az about:config képernyőt mutatja.
:q	Az aktuális fül bezárása. Ha csak egy fül van nyitva, bezárja a Firefoxot is.
:restart	A Firefox újraindítása
:set	Kilistázza az összes aktuális nem-alapértelmezett vimperator-beállításunkat.
:tabopen példa	Google keresés a „példa” szóra egy új fülön megnyitva.
:tabopen vimperator help	Google keresés a „vimperator” ÉS a „help” szó együttes előfordulására egy új fülön megnyitva.
:tabopen yahoo példa	Yahoo keresés a „példa” szóra egy új fülön megnyitva.
:tabopen ebay példa	Ebay keresés a „példa” szóra egy új fülön megnyitva.

A h,j,k,l gombok mellett scrollozásra használhatóak a kurzorbillentyűk is.

Megjegyzendő, hogy a google, yahooo stb kereséseknél bemutatott módszer azoknál a keresőszolgáltatásoknál válik be, melyeknek van pluginjük a /Programs/Firefox/Current/Searchplugins könyvtárban.

Ahhoz, hogy a Vimperátor azt is tudja, hogy a **Shift+kurzorbillentyűkkel** válthassunk a fülek közt (azaz „lapozhassunk” jobbra-balra) vegyük fel a `.vimperatorrc` fájlba a következő sorokat:

```
" kovetkezo/elozo ful kivalasztasa
map <S-Right> gt
map <S-Left> gT
```

Ahhoz pedig, hogy az aktuális fül elhelyezkedését (sorrendjét) megváltoztathassuk a **Ctrl+kurzorbillentyűkkel**, e sorokat vegyük fel:

```
" az aktualis ful jobbra/balra mozgatasa
map <C-Right> :tabmove! +1<CR>
map <C-Left> :tabmove! -1<CR>
```

Ha nem akarjuk hogy a Firefox menüje és hagyományos eszköztárai (URL-sor stb) úgy általában látsszanak, de azt azért akarjuk, hogy alkalomadtán előcsalogathassuk őket egy gombnyomásra, és ugyanúgy egy gombnyomásra el is tüntethessük őket, akkor a `.vimperatorrc` fájlba ne a `set guioptions+=mT` sort írjuk, hanem ezt:

```
" A menuk es hagyomanyos eszkozok elohozasa es elrejtese
map <A-Up> :set guioptions+=mT<CR>
map <A-Down> :set guioptions-=mT<CR>
```

Ez a rész azt csinálja, hogy ezentúl az **Alt+kurzorfel** gomb megnyomására **megjelennek** a Firefox hagyományos „tartozékai” (menük, URL-sor, stb), míg az **Alt+kurzorle** gomb hatására ezek **eltűnnek**. Én így használom a Vimperátort. E beállítás olyan „tiszta és száraz érzés”, mert „biztonságérzetet nyújt a nehéz napokon”, hiszen miközben a lehető legélethűbb „VIM-érzést nyújtó” módon használom a böngészőmet, aközben tudom, hogy „ha minden kötél szakad”, vissza tudok térni a megszokott használathoz a szükséges minimális időtartamra, mindössze egyetlen gombnyomással! Ráadásul közben büszke lehetek arra is, hogy e beállítási lehetőséget én magam kísérleteztem ki...

Megtehetjük azt is, hogy különböző wikipédiákban való keresésekhez készítünk forrógombokat. Ehhez is a `.vimperatorrc` fájlba kell sorokat felvennünk. Íme:

```
" kereses wikipediakban
map <A-F1> :open http://hu.wikipedia.org/wiki/Special:Search?search=
map <A-F2> :open http://en.wikipedia.org/wiki/Special:Search?search=
map <A-F3> :open http://gobo.kundor.org/wiki/Special:Search?search=
map <A-F4> :open http://linux.birodalom.net/gobowiki/index.php/
map <A-F5> :open http://poliverzum.birodalom.net/powi/index.php/
map <A-F6> :open http://wiki.hup.hu/index.php/
map <A-F8> :open http://poliverzum.birodalom.net/doku.php/
```

Mint látható, az **Alt+F1** a **magyar Wikipédiában** keres, az **Alt+F2** az **angol Wikipédiában**, az **Alt+F3** a **GoboLinux Wikijében**, az **Alt+F6** pedig a **HUP**, vagyis a **Hungarian Unix Portal** wikijében. Az **Alt+F4** és **Alt+F5** a saját wikijeimben való keresésre volna beállítva; ezeknél valamiért nem működik az a linkmegadás úgy, mint például a magyar Wikipédiában, így náluk (és a HUP wikinél is) az itt látható módon oldottam meg. Az **Alt+F8** -as sorban a dokuwiki szoftverben való keresés megvalósítására láthatunk példát. A forrógombok használati módja egyszerű: lenyomjuk a billentyűkombinációt, mire az alsó Vimperator-állapotsávba bemásolódik a megfelelő **:open** parancs és utána a link, s közvetlenül ezután kell írunk a keresendő kifejezést. Többszavas kifejezés esetén a szóközök helyett az aláhúzásjel (_) használandó! És persze Entert is kell nyomnunk. A szócikk az aktuális fülön jelenik meg (ha létezik a szócikk). Ha azt akarjuk, hogy mindig külön lapon jelenjék meg a keresés, akkor a **.vimperatorrc** megfelelő soraiba ne azt írjuk, hogy **:open** , hanem azt, hogy **:tabopen** .

Ha más gombot akarunk a „browser history”-ban való mozgáshoz mint a vimperator alapértelmezett **H** és **L** gombja, akkor beállíthatjuk úgy is, hogy nekünk tetsző gombnyomásra ugorjon előre vagy hátra a korábban meglátogatott honlapok közt. Ehhez vegyük fel a **.vimperatorrc** fájlba a következő sorokat:

```
" ugras a browser-historyban hatra vagy elore
map <A-b> :ba<CR>
map <A-f> :fo<CR>
```

A visszaugrás az **Alt+b** , az előreugrás az **Alt+f** gombra lett kötve.

Keresés online szótárakban:

Ehhez a következő sorokat vegyük fel a **.vimperatorrc** fájlba:

```
" Kereses szotarokban
map <M-F12> :open http://magyarangol.dicfor.com/hun/
map <M-F11> :open http://angolmagyar.dicfor.com/eng/
```

Az angol-magyar keresést ezek után a **Win+F11** gombbal, a magyar-angolt a **Win+F12** billentyűvel végezhetjük. (Természetesen a keresendő szót is be kell írunk, majd Entert nyomni...)

Ha nem tetszik a vimperator „statuszbar”-jának, azaz az alsó kijelzősorának színe, azt is kicserélhetjük, igaz, hogy ehhez már a Firefoxba kell kissé belenyúlunk. Csináljunk egy kis szöveges fájlt **userChrome.css** néven, és ezt mentjük el a

\$HOME/.mozilla/firefox/6u5h9ypf.default/chrome könyvtárba. Azaz, könyvbeli példánknl ez a

/Users/vz/.mozilla/firefox/6u5h9ypf.default/chrome könyvtár lesz. (Elképzelhető, hogy a 6u5h9ypf helyén nálad esetleg más számok és betűk vannak). A fájl tartalma a következő legyen:

```
.hl-StatusLine {
background-color: black !important;
color: #B0FF00 !important; /* világoszöld */
}
.hl-StatusLineSecure {
background-color: yellow !important; /* sarga */
color: black !important;
}
.hl-StatusLineBroken {
background-color: #FF6060 !important; /* rozsaszin */
color: black !important;
}
```

A „black” szín természetesen mindenütt feketét jelent. A file értelmezése: A „background-color” a státuszsor háttérszínét jelenti, a „color” pedig az írásszín. A **.hl-StatusLine** részben azt adjuk meg, általában milyen legyen a státuszsorunk állapota – én fekete alapon világoszöld színt határoztam meg, ez olyan „Mátrixos” érzés. A színek mint láthatóak, megadhatóak hexakóddal is és az angol nevükkel is. A **.hl-StatusLineSecure** részben azt határozzuk meg, milyen legyen a státuszsor színe titkosított kapcsolat esetén – például a gmail használata esetén többnyire ilyen lesz. A **.hl-StatusLineBroken** helyén meghatározott színeket pedig törött linkek esetén fogjuk látni.

Titkosított állományok és könyvtárak létrehozása

Előfordulhat, hogy olyan szupertitkos dolgaink vannak, amit el akarunk rejtteni. Akár otthoni, saját gépünkön is, akár a munkahelyi gépen. Például levelezés. Vagy akármi. Ekkor jó dolog, ha bizonyos könyvtárakat titkosítottá tehetünk. Ez nem azt jelenti, hogy e könyvtárt ne látná ezek után bárki, de azt igen, hogy az ebbe bemásolt állományok (közönséges fájlok illetve egész alkönyvtárak is) titkosítva lesznek, azaz lekéódolva egy baromihosszú jelszóval. E csodát az „EncFS” progi (aminek neve angolul az „ENCrypted FileSystem”-ből ered ha jól sejtem) hajtja végre nekünk. Telepítése 2 részből áll, az egyiket root userként kell végeznünk, a másikat közönséges felhasználóként.

Rootként, azaz rendszergazdaként:

Compile encfs

A függősége az Rlog progi, azt is telepíti.

Ezután:

modprobe fuse

A **/System/Settings/group** fájl végére írjuk be e sort:

fuse:x:1001:vz

(a „vz” helyére annak a felhasználónak a nevét írva, aki e szupertitkos dolgokat műveli majd nálunk).

Ellenőrizzük le, benne vagyunk-e a **fuse** csoportban, azaz írassuk ki, mely csoportoknak vagyunk tagjai:

```
vz@Csiszilla ~]groups vz
users tty tape floppy disk lp dialout audio cdrom uucp cdrecord video plugdev
fuse
```

Ha nincs a listában benne a „fuse”, akkor elrontottunk valamit.

Nem rootként:

Létrehozunk 2 könyvtárat, melyek neve a példánkban az lesz, hogy „titkos” és „nemtitkos”.

mkdir titkos

mkdir nemtitkos

Ezután a titkosított fájlrendszerben csak teljes útvonalakat használjunk!
Felmountoljuk a titkosított fájlrendszert a nemtitkos könyvtárba:

```
vz@Csiszilla ~]encfs /Users/vz/titkos /Users/vz/nemtitkos
```

Új titkosított kötet létrehozása.

Kérem válasszon az alábbi lehetőségek közül:

"x" a haladó beállításokhoz,

"p" a meghatározott beállításokat tartalmazó 'paranoia' konfigurációhoz,
minden mással, vagy az üres sorral az alap konfigurációt választja.

```
?> p
```

A 'paranoia' konfigurációt választotta.

A konfigurálás kész. A készülő fájlrendszer tulajdonságai:

Fájlrendszer titkosítás: "ssl/aes", verzió: 2:1:1

Fájlnev titkosítás: "nameio/block", verzió: 3:0:1

Kucs méret: 256 bit

Blokk méret: 512 byte, 8 byte MAC fejléceket tartalmaz

Minden fájl 8 byte fejléceket tartalmaz egyedi IV adattal.

Fájlnev titkosítás IV láncolást használ.

A fájl adat IV adata a fájlnev IV adatához van láncolva.

```
----- WARNING -----  
The external initialization-vector chaining option has been  
enabled. This option disables the use of hard links on the  
filesystem. Without hard links, some programs may not work.  
The programs 'mutt' and 'procmail' are known to fail. For  
more information, please see the encfs mailing list.  
If you would like to choose another configuration setting,  
please press CTRL-C now to abort and start over.
```

Önnek most be kell írnia a jelszót a fájlrendszerhez.

Jegyezze meg ezt a jelszót, mert egyáltalán

nincs semmiféle visszaállítási lehetőség. Viszont a jelszó később
megváltoztatható az encfsctl-lel.

Új Encfs jelszó:

EncFS jelszó megerősítése:

A fenti szövegben az angol rész azt jelenti, hogy a titkosított fájlrendszeredben a „hard” linkek nem fognak működni, s ez azt eredményezheti, hogy némely programok ott nem funkcionálnak – ilyenek például a „mutt” és a „procmail” nevűek. Hát ezt még ki fogjuk bírni...

Ezek után másoljunk fájlt a „nemtitkos” könyvtárba, majd listázzuk ki annak tartalmát – teljesen közönséges eredményt kapunk:

```
vz@Csiszilla ~/nemtitkos]ls
```

```
10 - Fight Of The Bumble-Bee.mp3
```

Igen, mert ide van felmountolva a titkos könyvtárunk! De listázzuk csak ki a titkos könyvtárat:

```
vz@Csiszilla ~/titkos]ls
```

aVtUa29-Sks72iCEmyQ,52CSH7DdNUxV6DRA7DjgVPNF71qZXFbtbYoGFhAZ0-iZn97

Na, EZ az IGAZI tartalom, ami a titkosított adat. Elég titkos, hm? Hiába látja bárki – nem tud mit kezdeni vele...

Menjünk át root jogosultságba, és próbáljuk úgy listázni a tartalomjegyzékeket:

```
vz@Csiszilla ~]su -  
Jelszó:
```

```
pv@Csiszilla /Users/vz]ls titkos  
aVtUa29-Sks72iCEmyQ,52CSH7DdNUxV6DRA7DjgVPNF71qZXFbtbYoGFhAZ0-iZn97  
pv@Csiszilla /Users/vz]ls nemtitkos  
ls: nemtitkos nem érhető el: Hozzáférés megtagadva
```

Vagyis a rencergizda se láthatja, hogy mit dugdosunk a gépen. Azt tudni fogja, hogy VALAMI sumákságot rejtegetünk, de nem tudhatja mi az – maximum letörölheti. De elolvasni nem tudja...

A titkosított könyvtár lecsatolása külön paranccsal történik, nem a szokásos mounttal, és CSAK a „nemtitkos” könyvtárra kiadva működik, azaz, általánosabban fogalmazva, nem arra kell kiadni AMIT csatoltál, hanem arra, AHOVÁ csatoltad:

```
vz@Csiszilla ~]fusermount -u nemtitkos  
vz@Csiszilla ~]ls titkos  
aVtUa29-Sks72iCEmyQ,52CSH7DdNUxV6DRA7DjgVPNF71qZXFbtbYoGFhAZ0-iZn97  
vz@Csiszilla ~]ls nemtitkos  
vz@Csiszilla ~]
```

Ha ugyanis e parancsot arra adnád ki, AMIT csatoltál, ezt válaszolná:

```
vz@Csiszilla ~]fusermount -u titkos  
fusermount: entry for /Users/vz/titikos not found in /etc/mtab
```

Az újrafelmountolás a megszokott módon történik, de persze bekéri a jelszót:

```
vz@Csiszilla ~]encfs /Users/vz/titikos /Users/vz/nemtitkos  
EncFS jelszó:  
vz@Csiszilla ~]ls titkos  
aVtUa29-Sks72iCEmyQ,52CSH7DdNUxV6DRA7DjgVPNF71qZXFbtbYoGFhAZ0-iZn97  
vz@Csiszilla ~]ls nemtitkos  
10 - Fight Of The Bumble-Bee.mp3  
vz@Csiszilla ~]
```

Ha akkor akarjuk lecsatolni a könyvtárat amikor valami progi használja, például játszunk le onnan a zenét, ezt írja ki:

```
vz@Csiszilla ~]fusermount -u nemtitkos
```



```
fusermount: failed to unmount /Users/vz/nemtitkos: Device or resource busy
```

Ügyeljünk rá, hogy a „titkos” könyvtárban lesz nekünk egy „rejtett” fájlunk (dot-fájl), aminek neve

.encfs5

és ezt **nagyon** nem jó ötlet letörölni, mert akkor nem tudunk hozzáférni az adatainkhoz! Épp emiatt ha mentéseket készítünk az e könyvtárban tárolt adatainkról, ezt is másoljuk át...

Illik tudnunk azt is, hogy a meta-információk, tehát a fájlok tulajdonjoga, engedélyei és az utolsó módosítás dátuma nem titkosított.

Ha a jelszót egy fájlban tároljuk, azt így adhatjuk meg a csatolás pillanatában:

```
vz@Csiszilla ~]cat ./jelszo | encfs -S /Users/vz/titkos /Users/vz/nemtitkos
```

A fenti példában feltételeztük, hogy abban a könyvtárban állunk épp, ahol a jelszó el van tárolva egy „jelszo” nevű fájlban. Ez persze nem túl értelmes és biztonságos dolog, de példának jó – a lényeg az, hogy így a jelszófájlt tárolhatjuk például egy pendrive-on, ami mindig nálunk van, a csatoláskor bedugjuk, felcsatoljuk a pendrájvot, s az azon levő jelszófájl elérési útját adjuk meg. Ez a GoboLinux esetén lehetne például

/Mount/PEN/jelszo

Fájlban tárolt jelszó esetén ügyeljünk rá, hogy az igazi jelszót muszáj lezárni egy „enter” jellel (karakterrel, azaz sorvég-jellel), hogy elfogadja a rendszerünk, különben ezt írja ki:

```
vz@Csiszilla ~]cat ./jelszo | encfs -S /Users/vz/titkos /Users/vz/nemtitkos  
Hiba a kötet kulcsának meghatározásánál, a jelszó hibás
```

Ha a „fuse” kernelmodult nem töltjük be a „modprobe” utasítással, ez történik:

```
vz@Csiszilla ~]encfs /Users/vz/titkos /Users/vz/nemtitkos  
EncFS jelszó:  
fusermount: fuse device not found, try 'modprobe fuse' first  
A fuse hibásan futott le, lehetséges gyakori problémák:  
- a fuse kernel modul nincs telepítve (modprobe fuse)  
- hibás paraméterek -- nézze meg a hibaüzeneteket
```

Lehetséges azonban, hogy ezt kényelmetlennek érezzük – azt szeretnénk, hogy a rendszer a „fuse” modult automatikusan betöltse indulás után! Nos, akkor annyi a dolgunk, hogy a **/Programs/BootScripts/Settings/BootOptions** állomány végébe, az „UserDefinedModules” részhez beírjuk a „fuse” modult is. A

kérdéses rész így valahogy nézhet ki azután – a lényeg a félkövérrel kiemelt szócska:

```
UserDefinedModules=(  
    snd-pcm-oss  
    snd-mixer-oss  
    snd-intel8x0  
    w83627hf  
    fuse  
)
```

Azt is megtehetjük, hogy ugyanazon titkosított könyvtáron belül más jelszavakat adunk az egyes állományoknak vagy alkönyvtáraknak. Ehhez mindenekelőtt csatoljuk le a már felcsatolt titkos könyvtárat:

```
vz@Csiszilla ~]fusermount -u nemtitkos
```

majd csatoljuk fel újra, de úgy, hogy egy másik jelszót adunk neki. Ezt a funkciót a „**--anykey**” kapcsoló végzi:

```
vz@Csiszilla ~]encfs --anykey /Users/vz/titkos /Users/vz/nemtitkos  
EncFS jelszó:
```

Ha ezután kilistázzuk a „nemtitkos” könyvtárat, láthatjuk, hogy nem látjuk benne egyetlen, korábban a „titkos” könyvtárba betett állományunkat sem – holott a „titkos” könyvtárban jól látszódnak, persze titkosítottan! Ez nem hiba, ennek így kell lennie – mindig csak azon állományokat láthatjuk a „nemtitkos” könyvtáron keresztül, melyek az éppen a felcsatoláskor használt jelszóval lettek titkosítva! Másoljunk a „nemtitkos” könyvtárba egy állományt, és listázzuk ki a „titkos” könyvtárat. Látjuk hogy ott KÉT állomány szerepel, ellenben a „nemtitkos”-ban csak egy mutatkozik, a mostani:

```
vz@Csiszilla ~]ls titkos  
3fxxQ5W86TKVzhjHCQNVl8agwsXkkXeYi5-Kgw267fXvfBkseOPw3i5LLDHqJ2wnWC1  
aVtUa29-Sks72iCEmyQ,52CSH7DdNUxV6DRA7DjgVPNF71qZXFbtbYoGFhAZ0-iZn97  
vz@Csiszilla ~]ls nemtitkos  
04 Serenade Nicola Benedetti.mp3
```

Most csatoljuk le megint a „nemtitkos” könyvtárat, csatoljuk fel újra a régi jelszóval a „titkos”-at és listázzuk ki a két könyvtár tartalmát. A „titkos”-ban még mindig két (titkosított) fájl látnak, de a „nemtitkos”-ban csak egyet, ám most a régebben oda betett:

```
vz@Csiszilla ~]fusermount -u nemtitkos  
vz@Csiszilla ~]encfs /Users/vz/titkos /Users/vz/nemtitkos  
EncFS jelszó:  
vz@Csiszilla ~]ls titkos  
3fxxQ5W86TKVzhjHCQNVl8agwsXkkXeYi5-Kgw267fXvfBkseOPw3i5LLDHqJ2wnWC1  
aVtUa29-Sks72iCEmyQ,52CSH7DdNUxV6DRA7DjgVPNF71qZXFbtbYoGFhAZ0-iZn97
```

vz@Csiszilla ~]ls **nemtitkos**
10 - Fight Of The Bumble-Bee.mp3

Ezek alapján már bárki dugdoshatja az irtófontos dolgait bárki előtt, akit illetéktelennek tart...

Saját, személyre szabott Telepítő- és LiveCD készítése

Azt hiszem rögvést az elején kitalálhatja mindenki, hogy olyan CD-ről beszélek, mely egyszerre LiveCD is és TelepítőCD is, híven a GoboLinux hagyományaihoz.

Aki saját disztribúciót szeretne készíteni, annak legegyszerűbb egy már meglevőből kiindulni, amin elkezd ezt-azt, s nyilván egyre többet változtatni. De ha nincs is valakinek ekkora vágya, még az esetben is előnyös, ha olyan CD-t készíthetünk, melyen a kedvenc programjaink vannak elhelyezve, s ezeket onnan telepíthetjük is! Akinek vannak efféle vágyai, annak igazi kincs a GoboLinux, mert ideális, és roppant könnyű módot kínál e vágy kielégítésére, sőt, részletes MAGYAR nyelvű leírás is van hozzá. Ez az a leírás, amit most olvasol, tisztelt Olvasóm!

Igazából az a helyzet, hogy a GoboLinux LiveCD-n elég kevés „hasznos” proggi van, felhasználói szempontból, ugyanis ez egy forrásalapú disztró, s emiatt egy csomó olyasmit is kénytelen magával hurcolni, melyeket a bináris disztrók bátran nélkülözhetnek – például a GCC, Python, stb. A nagyobbik baj az, hogy például rajta van az angol OpenOffice, holott 99%-ig biztos, hogy te a magyar nyelvű OOo-t akarod majd használni. Feleslegesen foglalja tehát a helyet, pedig megvan vagy ~200 mega. EGYHARARMADA a CD helyének! Biztos nem kell neked a KDE braziliai portugál és német nyelvi fájlja sem. Talán a Firefoxot is lecserélnéd magyar nyelvűre. Sőt, lehet hogy eleve nem is KDE grafikus felületet óhajtasz... Mit teszel akkor?!

Elkészítéd a saját, személyre szabott CD-det. Állítólag Windowshoz is léteznek mindenféle cuccok, amikkel holmi személyre szabott telepítőkészletet lehet összeeszkábálni – nos, GoboLinuxunk ebben nem marad el tőle! Sőt, itt ha akarod, még a rendszerszkripteket is bátran átírhatod...

Kezdjünk tehát bele! A <http://gobolinux.org/index.php?page=livecd> link alatt van erről egy leírás, de nem eléggé részletes szerintem – emiatt írtam e mostani „tutoriált” e témáról.

Mindenekelőtt: az összes itt felsorolt okosság rendszergazdai joggal hajtandó végre, tehát legjobb, ha most a legelején átmész rencergizdába a

su -

parancs segítségével. Megjegyzendő, ez amiatt van, mert e szkriptek suid biteket piszkálnak, kicsomagolnak-becsomagolnak, meg mindenféle egyéb „varázslásokat” végeznek, amiket csak a rencergizda csinálhat. Mindez tulajdonképpen meglehetősen rizikós – ne feledd, A SAJÁT KOCKÁZATODRA CSELEKSZEL! A legértékesebb adataidról célszerű biztonsági másolatot készíteni előbb... És egyáltalán, légy észnél, amikor most parancsokat adsz

ki!!!!!!!!!!

Ezután telepítsd a LiveCD készítő szkriptcsomagot (ez ugyanis nem része a GoboLinux alaptelepítésnek, mert ez tényleg már erősen haladóknak való téma...):

Compile buildlivecd

Ezután csinálj a /Depot könyvtárba egy LiveCD nevű alkönyvtárat és lépj oda be. Ezt legegyszerűbb így:

```
mkdir /Depot/LiveCD  
cd /Depot/LiveCD
```

Megjegyzendő, hogy a LiveCD könyvtárnak OKVETLENÜL ugyanazon a partíción kell lennie, mint amelyen az „éles rendszered”, vagyis az ISO képmást készítő szkriptet futtató, épp működő rendszered van telepítve! Rém rossz ötlet valami felmountolt helyre tenni, pláne szimbolikus linkkel hivatkozni rá, vagy valamelyik altartalomjegyzékére. Ha ugyanis nem így teszel, akkor a folyamat közepe táján (a squashfs-képmások készítésekor – ezt nem tudod hogy mi, de nem baj...) ELHALÁLOZIK az „ISO-csináló” szkript! Nem ír ki hibaüzenetet, látszólag jól befejeződik a futása, valójában azonban csak valami néhány megás kutyulék lesz az, amit a kész ISO-képmás címén elébed tálal, ami ugye nem valószínű, hogy jó lenne neked, mert egy CD mérete nagyon nem akkora... Erre engem saját, szomorú tapasztalataim tanítottak meg. Ebből máris következik, hogy aki saját CD vagy pláne DVD készítésére gondol, az jól teszi, ha kellően nagy partícióra telepíti a fő rendszerét, hogy ott elférjen ez is. E munkához ugyanis, ha egy X méretű ISO fájlt akarunk készíteni, minimum 3*X mennyiségű hely kell, mert egyszer kell hely a csomagoknak, másodszor ezeket a szkriptünk ki is csomagolja majd egy ideiglenes könyvtárba, plusz hely kell az ISO-nak is. Minthogy a kicsomagolt csomagok több helyet foglalnak el, mint becsomagolva, emiatt a 3*X valójában a helyszükséglet durva **alul**becslése, inkább az 5*X közelíti meg a valóságot. Azaz, ha egy 700 megás CD-t tervezünk készíteni, minimum 3500 mega szabad hellyel kell rendelkezünk a gyökérpartíciónkon e munkához.

Ezután add ki e parancsot:

UpdateEnvironment

Erre most elkezd letölteni néhány száz mega adatot – mindenféle csomagokat. Internetsebességedtől függően ez bizony elég soká is tarthat... de muszáj. Amiatt muszáj, mert most tulajdonképpen letölt MINDENT, amiből el lehetne készíteni az *eredeti* GoboLinux LiveCD-t. Ez pedig nagyon jó nekünk, mert

nyilván úgy lesz ez, hogy a mi „saját” CD-nk alapja is az eredeti GoboLinux CD lesz, épp csak ezt-azt megváltoztatunk rajta. Ki többet, ki kevesebbet, de valószínűleg mégis megmarad eredeti formájában a nagyja.

Miután mindent letöltött és te visszakaptad a promptot, készíts a /Depot/LiveCD könyvtárról egy másolatot valahová, mert gondolom nem akarod újra eljátszani ezt az időrabló letöltögetősdit, ha kiderül, hogy elszúrtál valamit...

A /Depot/LiveCD könyvtárban 5 könyvtárat kell láss:

```
09/04 11:17 rwxr-xr-x      304 Archives
06/04 09:40 rwxr-xr-x       96 LocalSVN
09/04 12:18 rwxr-xr-x      576 Output
09/04 10:42 rwxr-xr-x    13896 Packages
09/04 02:03 rwxr-xr-x      408 Progik
```

A letöltött csomagok természetesen a „Packages” nevűben vannak.

A tulajdonképpeni „LiveCD-készítés” VÉGTELENÜL egyszerű: ugyanis másból sem áll, minthogy kiadj egyetlen, paraméterek nélküli parancsot, ezt:

CompressAndBuildISO

E parancsot nem mindegy, hol adjuk ki. Abban a könyvtárban állva kell kiadnunk, ahol az UpdateEnvironment parancsot is kiadtuk korábban, másképp ezt írja majd ki nekünk:

```
pv@Csiszilla ~]CompressAndBuildISO
=> ROLayer stage started
Basic directories could not be found.
Please check if you are running the script inside a directory populated by
UpdateEnviromment
```

De MOST még ne adjuk ki e parancsot a jó könyvtárban sem, mert akkor szimplán csak az eredeti GoboLinux LiveCD-t készítené el nekünk! Előbb ugyebár meg kell változtatnunk mindenfélét a nekünk tetszőkre.

Mindenekelőtt HELYET KELL CSINÁLNUNK. Egy CD még számunkra és a GoboLinux számára is csupán 700 mega, valamit tehát le kell törölnünk. Természetesen jobb, ha nem nyúlunk máshoz, mint a Packages könyvtár csomagjaihoz. Innen szívfájdalom nélkül letörölhetünk mindenfélét, amit nem akarunk felrakni a CD-re – bár tulajdonképpen felesleges bármit is törölnünk. A LiveCD készítő szkriptet ugyanis magasan nem érdekli, mi van itt, AZON FELÜL, ami neki kell. Ő nem e tartalomjegyzék listája alapján dolgozik, hanem van neki külön listája arról, miket kell beépítenie a CD-re, és a

Packages könyvtárból csak az alapanyagot szedi. Bátran lehet itt akárhány fölösleges cucc is, az nem zavar.

A csomaglistát kell tehát összeállítani neki. Az eredeti csomaglistát a **/Programs/BuildLiveCD/20080406-SVN/Data**

könyvtárban leljük meg, s hozzá több lista is van itt, melyek a következő neveket viselik:

Packages-Devel-i686

Packages-List-Base

Packages-List-GoboLinuxTools-SVN

Packages-List-GoboLinuxTools-Stable

Packages-List-KDE

Packages-List-NonBase

Packages-List-OpenOffice

Packages-List-Printing

Packages-List-XorgAndDependencies

A LiveCD elkészítéséhez a szkript e listák mindegyikének figyelembe veszi a tartalmát. A listák felépítése rém egyszerű: egy sor = egy csomagnév-bejegyzés, és a névnél NEM kell megadni verziót! A #-el kezdődő sorok megjegyzések, nincsenek figyelembe véve. Azaz, mindenekelőtt nézzük át e listákat, és csináljunk úgy helyet, hogy a nekünk nem kellő csomagok neve elé teszünk egy kereszt-jelet („#”). Így ha később rájövünk hogy de hoppá, mégis kellenek, könnyű „visszacsinálni” mindent. Javaslom törlésre a KDE minden nyelvi fájlját a magyar kivételével, plusz a Lilo sem kell, ha Grub a betöltésvezérlőnk. Én magam az egész KDE-t töröltem mindenestül, meg a Qt-t is. Emiatt törölhettem a K3B-t is, mert az Qt-függő progí. Persze hogy ki mit töröl, ízlés kérdése. Óvatosan kell eljárni, mert most NEKÜNK kell törődni a függőségekkel, azt a progí nem oldja meg helyettünk! Persze ha valaki nem használja mondjuk a „Bc” parancssoros számológépet vagy a „Vi” szövegszerkesztőt, ezeket is törölheti. Ez már olyan játszadozás, hogy ismernünk kell a mi szeretett disztrónkat, tudnunk kell, mi van benne és mi mire való.

Javaslom, NE töröljük a Firefox és OpenOffice bejegyzéseket, ha ugyanis ezekből saját (magyar) verziót akarunk használni, kellenek e bejegyzések - mindössze itt a CSOMAGOKAT cseréljük majd ki. A szkript ugyanis nem vesz figyelembe verziókat – látható, itt csak a nevek vannak megadva, verziók nem! A Packages könyvtárban pontosan egy példány, azaz 1 verzió lehet minden progiból. Miután tehát letörölgettük (azaz megjegyzéssé fokoztuk le) a nekünk nem kellő programokat, vegyük is fel valamelyik listába a nekünk kellő progik neveit! NAGYON ügyeljünk rá, hogy e lista úgynevezett „case-sensitiv”, azaz kisbetű-nagybetű érzékeny, azaz minden progí nevének minden betűjét PONTOSAN úgy kell írunk, ahogyan az majd a csomagjának a nevében szerepel! Egyetlen betű eltérés megghiúsíthatja a sikert!

Na de melyik listába írjuk, amit akarunk? Először is, vannak eleve kikommentezett programnevek a listákban, ilyen például a „speex” is. Ha nekünk ez mégis kell, logikusan egyszerűen eltávolítjuk a név elől a kommentjelet. Ha a progink neve nincs a listában, szintén felettebb logikusan célszerű a Packages-List-NonBase nevűbe írni, mert ezek „non-base”, azaz „nem-alapvető” dolgok. Különbösen ez csak egy személyes vélemény a részemről, mert alapvetően *mindegy*, melyikbe írjuk őket.

Továbbá: a könyvtárban már meglevő listák mellé újakat is felvehetünk, nekünk tetsző névvel (mondjuk nem tudom, az ékezetes betűket hogyan tűrné el a szkript...), amennyiben a név úgy kezdődik, hogy **„Packages-List-”**. Ezután bármit írhatunk a listanév folytatásaként. Én például eképp csináltam egy **„Packages-List-NonFree”** nevű listát, amibe az **Nvidia** és a **JRE** programokat vettem fel, ezek ugyanis elvileg – jogi okokból – nem lehetnének fenn egy hivatalos disztróban. De ugye ki tudja mi van az én saját CD-men, meg különben is, magának azt csinál az ember amit akar, csak ekkor terjeszteni nem szabad (ha jól tudom). Mindenesetre ha ezek után akarok egy terjesztésre szánt disztrót is csinálni ami ugyanaz mint a magam személyes lemeze, csak a tilos dolgok nincsenek rajta, akkor egyszerűen kitörölöm a **Packages-List-NonFree** listát a könyvtárból, s úgy futtatom le a LiveCD készítő eljárást.

Ezután menjünk a mi Packages könyvtárunkba, és ha valami csomag helyett újat akarunk betenni, töröljük le a régi csomagot, majd (ugyanebben a könyvtárban maradván) hozzuk létre az újat! Ez nyilván azt jelenti majd, hogy a saját disztrónk már kipróbált, működő progijából csinálunk csomagot. Például az OpenOffice-ből csomagot e paranccsal készítünk:

CreatePackage OpenOffice

Megjegyzendő, hogy az OpenOffice-ban van olyan, hogy „gallery” nevű könyvtár, s ebben rengeteg fotó is az egyik alkönyvtárban. Ha a csomagkészítés előtt ezeket töröljük (s ezek igazán nem létfontosságúak a működéséhez...), akkor kb 30 megát megspórolunk! S ez biza nagy kincs – CD-kapacitásunk egyhuszada!

Ha efféle törlést végzünk – vagy bármiféle módosítást a feltelepített programok állományain – akkor csomagkészítés közben valószínűleg reklamálni fog nekünk a csomagkészítő szkript, hogy nem stimmel az úgynevezett „hash”, amivel ellenőrzi a csomag érintetlenségét! Azaz nem csinál nekünk csomagot, hanem efféleképp rikoltozik, mint most nekem a vim esetén, amiből kitöröltem (szándékosan) az olasz nyelvű tutorial fájlt:

```
pv@Csiszilla ~]CreatePackage vim
CreatePackage: Creating package for Vim 7.0.
VerifyProgram: No signature to verify.
VerifyProgram: Verifying FileHash...
VerifyProgram: FileHash check failed: Shared/vim/vim70/tutor/tutor.it is
missing
CreatePackage: Unable to verify package. Do not redistribute
```


Ez nem gond. El kell menni az adott progi

/Programs/progi_neve/verziószám/Resources könyvtárába, és itt durván letörölni a **FileHash** nevű fájlt. Ezután már hajlandó megcsinálni a csomít, csak előtte megjegyzi, hogy nincs „szignálva”:

```
pv@Csiszilla ~]CreatePackage vim
CreatePackage: Creating package for Vim 7.0.
VerifyProgram: No signature to verify.
VerifyProgram: No FileHash to verify.
CreatePackage: No existing sig. Re-sign?
CreatePackage: Press Enter to continue or Ctrl-C to cancel.
```

Itt bátran nyomjunk neki Entert, s erre elkészíti a csomagot.

Felmerül a kérdés, mi van a verziószámokkal? Nos, illik, hogy azokat megváltoztassuk, amennyiben változtattunk a csomag összetételén. Például én a VIM progi esetén beraktam oda a magyar tutoriált, de az összes többit kiirtottam belőle. Eredetileg a VIM verziója az volt, hogy 7.0 – ám én átneveztem a verziót úgy, hogy **7.0_MagyarOS** és így csináltam belőle csomagot. Efféle verziószám-átnevezést úgy a legegyszerűbb végrehajtani, hogy éles, futó rendszerünkben először letiltjuk a csomagot:

DisableProgram vim

majd átnevezzük a verzióbejegyzést a megfelelő könyvtárban, e vim-példa szerint tehát a /Programs/Vim/7.0 könyvtárat átnevezzük 7.0_MagyarOS -ra, majd

SymlinkProgram vim

és ezután csinálunk csak csomagot belőle a CreatePackage paranccsal.

Fontos megjegyeznünk, hogy ha egy csomagot kitömörítesz, megváltoztatsz, majd újra betömörítesz, s esetleg NEM a CreatePackage paranccsal hozod létre a tömörített csomagot, akkor a neve legyen a Gobo-szintaxis szerinti, azaz végződjön úgy, hogy „--i686.tar.bz2”. Tehát egy szabványos csomagnév felépítése:

ALAPNÉV--verziószám--i686.tar.bz2

Különben a CD-készítő progi nem találja majd meg vagy az alapnevet, vagy a verziószámot, s valami hülyeséget ókumulál ki belőle, ami miatt nem fog működni a progi. Azaz ÜGYELJÜNK A DUPLA MÍNUSZJELEKRE!

Felmerülhet a kérdés arra vonatkozóan, hogy saját fontkészletekkel hogyan tudjuk ellátni a készülő rendszerünket? Nos, a fontok speciális helyen vannak tárolva: a **/Depot/LiveCD/Archives** könyvtárban van egy

Files-Fonts.tar.bz2 állomány. Természetesen mi sem egyszerűbb, mint ezt kicsomagolni valahová, kitörölni belőle ami benne van de nekünk esetleg nem kell, bemásolni ami nekünk tetszik, „visszacsomagolni” újra tar.bz2 fájlá DE UGYANEZEN A NÉVEN!!, és bemásolni az eredeti állomány helyére... Ügyeljünk rá, hogy több alkönyvtár van ezen fájlban, s nem mindegy, hogy mondjuk a TTF fontjainkat melyikbe másoljuk! Természetesen a többi ttf állomány könyvtárában van a helyük...

Na de te esetleg szeretnél rögvést úgy telepíteni, hogy azonnal saját rendszerbeállításai legyenek a létrehozott új felhasználóknak! Például biztos ez a helyzet akkor, ha lecserélnéd az alapértelmezett KDE-t valami más ablakkezelő progira. Akkor ugye meg kell változtasd a Felhasználó \$HOME/.xinitrc fájlját, mert az most a KDE-t töltené be, de az akkor már nem lesz...

Ez is megoldható. Ehhez a Packages tartalomjegyzékünkben kell megváltoztatnunk az **EnhancedSkel** nevű csomag tartalmát. Épp úgy módosíthatjuk, mint bármi más szokásos progí csomagjával tennénk. Ez – ha belenézünk láthatjuk – egy „testuser” nevű mappát tartalmaz csupán, melyben egyszerűen minden benne van, ami létrehozattatik MINDEN létrehozandó felhasználó számára a Kezdet Kezdetén, amikor Ő, a Felhasználó létrejön. Nyugodtan ki lehet egészíteni hát ezt nekünk tetsző mappákkal, fájlokkal, módosíthatjuk vagy törölhetjük a meglevőket... ha kész vagyunk, csomagoljuk be, és leszen majd csoda: ha el nem rontottuk, működni kell majd...

Ha olyan módosításokat szeretnénk, hogy hogyan induljon a (már telepített) rendszer, ahhoz a BootScripts mappába kell benyúlnunk. Tudjuk: kicsomagoljuk, megváltoztatjuk, új csomagot készítünk belőle... hogy hogyan változtassuk meg, azt nem írom le, ahhoz tessék megismerni a rendszert, különben is, itt csak a fantázia szabhat határt! Én például alapértelmezetten betöltötté tettem a cdfs és az nvidia kernelmodulokat, mert nekem úgy tetszett (sőt, tetszik még most is...). Itt az egyéni igények dominálnak.

Ha mindezzel megvagyunk, adjuk ki a

CompressAndBuildISO

parancsot (a megfelelő könyvtárban, erről fentebb már írtam), és várjunk türelmesen. Jó sokáig elszöszmötöl majd, de ezalatt egy másik ablakban nyugodtan internetezhetünk vagy szöveget szerkeszthetünk. Amennyiben semmi durva hibát nem követtünk el, azért jó esélyünk van rá, hogy kb egy óra múlva visszakapjuk a promptot. Ekkor nézzünk be a

/Depot/LiveCD/Output

könyvtárba. Itt lesz mindenféle, de kell találnunk egy

GoboLinux-NoVersion.iso nevű fájlt is, ami gondolom úgy 600 és 700 mega közt lehet „méretileg” - hacsak nem valami minimáldisztrót gyártottál épp. Na

ez lesz az az ISO képmás, amit a szokásos módon lemezre kell égetned, majd „bebútolnod” róla. EZ A TE SAJÁT CD-d.

E létrejött file neve a

/Programs/BuildLiveCD/20080406-SVN/bin/CompressAndBuildISO szkript majdnem legvégén van meghatározva, ott ugyanis szerepel e sor:

iso_name="Output/GoboLinux-NoVersion.iso"

Ha ezt a szkript futtatása előtt átírjuk, meghatározhatjuk a létrejövő állományunk nevét. Véleményem szerint azonban abszolút felesleges a szkripthez hozzápiszkálni – messze sokkal egyszerűbb elkészülése után az állományt azonnal átnevezni. Különbözik meg ennél fontosabb lenne a telepítőben az üdvözlőszöveg kicserélése, ilyesmi, de nem érdekel annyira, hogy pillanatnyilag foglalkozzam vele. Aki kikísérletezi, majd írja meg. Mindenesetre az eddigiek alapján bárki írhat egy teljesen korrektül működő saját LiveCD-t. Természetesen a rákerült csomagok közt továbbra is lehet válogatni telepítéskor, de arra ügyeljünk, hogy a GRAFIKUS installálórésze csak akkor működik, ha nem piszkáljuk a Qt csomagot. (Sőt, lehet hogy kifejezetten maga a KDE is kell neki – ez utóbbit nem kísérleteztem ki). Mindenesetre a parancssoros installálórésze tökéletes minden Qt és KDE nélkül is, ezt kipróbáltam.

A telepítés során mint tudjuk, a GoboLinux felajánl egy alap (base) és egy tipikus (typical) telepítési listát. (ezek mellett választhatjuk az összes csomagot (full) is). Nos hogy mi kerül az „alap”-ba és a „tipikus”-ba, természetesen azt is mi határozhatjuk meg: ki hinné, ez az „Installer” nevű progiban van „elrejtve” (az Installert mondjuk illik azért mindenképp rávenni a lemezünkre...). Tehát keressük meg a

/Programs/Installer/verziószám/Shared/Installer/Profiles/Base

és

/Programs/Installer/verziószám/Shared/Installer/Profiles/Typical

szöveges fájlokat, és írjuk át őket tetszésünk szerint...

Még az Installer progi ikonját is kicserélhetjük: őt a

/Programs/Installer/verzió/Shared/Installer/Images/que.png néven találjuk meg.

Amennyiben a létrejött cédéd nem találtatik tökéletesnek – ami több mint valószínű, ha első próbálkozásod – akkor nem kell eljátszanod az összes lépést a legelejétől. Elvileg a CompressAndBuildISO szkriptnek több belépési pontja van, meg lehet neki mondani hogy innen vagy onnan induljon, ez le van írva a belinkelt angol oldalon. Én itt nem ismertetem, mert nálam először működött, másodjára már nem – bár tutibiztos én szúrtam el valamit, de ha én is elszúrom, egy még kezdőbb még inkább. Mindenesetre nekem csak valami kis 300 megás kutyulékot csinált a 700 megás ISO helyett... A biztos az az, hogy igenis lefuttatjuk a teljes CompressAndBuildISO szkriptet újra, de előtte töröljük a /Depot/LiveCD/Output könyvtár TARTALMÁT, azaz nem az Output könyvtárat, hanem csak az ezen belüli fájlokat és alkönyvtárakat!

Viszont NEM kell lefuttatni újra az UpdateEnvironment szkriptet (**nem is szabad**, mert átírja a megváltoztatott csomagjainkat, meg melléjük tölt más verzióúakat, s akkor a CD-csináló szkript majd rikoztzik nekünk, hogy egy progiból ő csak egy változatot tűr meg...), és természetesen csak azon csomagokkal szórakozunk, melyeket továbbmódosítani akarunk valamiért (illetve a saját éles rendszerünkben levő BuildLiveCD program Data könyvtárában levő listák tartalmát írjuk át esetleg, ha újabb csomagokat akarunk beszerkeszteni, vagy ellenkezőleg, elhagyni). Ügyeljünk rá, ha a LiveCD-re is rá akarjuk tenni a BuildLiveCD csomagot, akkor a módosításokat ne (csak) a CD-re kerülő változaton vezessük végig, mert NEM abból dolgozik a szkript, hanem az élő rendszerünkben levő „élesített” program adatait veszi figyelembe!

Befejezésként térjünk vissza rá, hogy honnan szerzünk HELYET a kedvenc progijainknak?!

Nos, összegyűjtve a tippjeim:

- A KDE egy helypocsékoló szörnyeteg. LE VELE! Vannak SOKKAL kisebb ablakkezelők is. A SithWM például...
- A Qt majdnem csak a KDE miatt kell és a K3B miatt. De K3B nélkül is lehet CD-t írni parancssorból – hogy hogyan, ezt is leírtam korábban... Azaz a Qt is nélkülözhető.
- Minden progi felesleges, amit nem használunk rendszeresen. Például a Vim a legtöbb embernek misztikus homály, s mert úgysem használja, törölheti. Én nem törölöm, de az én vagyok.
- A magyar embereknek többnyire felesleges a KDE minden nyelvi fájlja a magyar kivételével, akkor is, ha a KDE-t amúgy megtartja.
- Az OpenOffice BIZTOSAN KELL NEKED?! Ne feledd: Ő foglalja el a CD helyének HARMADÁT!
- Ha kell is az OpenOffice, tuti hogy egy rakás kép belőle felesleges. Nem hiszem hogy kell belőle az összes sablon, bemutató, példák, varázslók, stb... Ezeket mind kiirthatod, s karcsúsíthatsz.
- ÉRDEMES végigmenni a CD-re szánt ÖSSZES progin, s MINDBŐL kiírtani a tutorialokat, man-oldalakat és LC_MESSAGES részeket, AMIK NEM MAGYAROK. Meglehetősen valószínűtlen ugye, hogy ha megtartod is mondjuk a Vim-szövegszerkesztőt, de ennek amúgysem evidens kezelését mondjuk koreai nyelven óhajtanád elsajátítani.
- Hasonlóképp érdemes kiírtani az olyan fájlokat, hogy „license, TODO, readme, news”... Persze, NEM ILLIK közreadni egy disztrót úgy, hogy nincsenek meg benne a felhasznált progik licenszfeltételei, a készítők neve, stb, de ha ez a TE SAJÁT CD-d lesz, kizárólagos használatra, ami felfogható eképp afféle „karcsúsított biztonsági másolatnak” is, akkor miért is ne!
- Számos progi – például az OpenOffice is! - tárol mindenféle hangfájlokat hangjelzésre, ráadásul WAV formátumban. Ezek iszonyúsok helyet tudnak

elfoglalni. Ha amúgysem vagy oda a hangjelzésekért, ezeket nyugodtan kitörölheted. Az OOo például majdnem 3 mega WAV hangfájlt hurcol magával. Ez első pillantásra csekélységnek tűnhet, holott nem az – a Wcalc nevű konzolos kalkulátorprogi – aminek biztos több hasznát veszed – például csak kb egytized mega tömörítve! Azaz 3 megába is sok-sok hasznos izémizé férhet be!

- Ha Grub-ot használasz, nem kell neked a Lilo.
- Szerintem nem kell neked a CD-re az Aspell-en nevű angol helyesírás-ellenőrző sem.
- Általában véve a parancssoros alkalmazások sokkal kisebbek mint a grafikusak, emiatt belőlük több fér fel a CD-re.
- Ha nem használasz KDE-t, felesleges meghagynod a Manager progit is, ami a GoboLinux csomagkezelőinek grafikus „frontendje”. Ugyanis KDE-függő. A parancssoros szkriptekkel történő csomagkezelés természetesen működni fog.
- Ha nem nagyon akarsz a kész rendszeredre forrásból telepíteni – habár a GoboLinux forrásalapú disztró! - de te „készre” szánod, akkor felesleges a Compile program, s a GCC is meg a Linux kernel és a kernel-headers is.

És most a végén egy fontos infó: Ha nagyon kevésnek tűnik a hely a CD-n, akkor ne LiveCD, hanem LiveDVD legyen az, amit alkotsz! Arra egy KICSIVEL mintha több adat férne rá, ugye... Ja, hogy hogyan csinálj LiveDVD-t?! Pontosan úgy, ahogy fentebb leírtam a CD-csinálást. Semmiben nem különbözik tőle, egyszerűen a létrejövő ISO fájl értelemszerűen nagyobb lesz, akkora, hogy nem fér fel egy CD-re. Hát akkor ne CD-re írd ki, hanem DVD-re, ennyi az egész...

Hasznos linkek

Itt adok meg néhány hasznos internetes linket.

A GoboLinux hivatalos honlapja:

<http://gobolinux.org/>

A GoboLinux honlapja magyar nyelven:

http://gobolinux.org/index.php?lang=hu_HU

A GoboLinux angol fóruma:

<http://forum.gobolinux.org/>

Itt kereshetsz recipéket valamely progizhoz:

<http://gobo.dreamhosters.com/recipes/>

Az angol GoboWiki, azaz TudásBázis:

http://gobo.kundor.org/wiki/Main_Page

A magyar GoboLinux fórum:

<http://linux.birodalom.net/smf/>

Ez is egy magyar GoboLinux fórum:

<http://linuxempire.hu/viewforum.php?f=30>

A magyar GoboWiki:

<http://linux.birodalom.net/gobowiki/>

Itt is kereshetsz csomagokat: <http://gobolinux.uw.hu>

és itt is: <http://gobolinux.info>

Egy általam „elkövetett” linuxos tanulmány (Értekezés az ingyenes szoftverekről):

<http://portal.linuxempire.hu/node/29>

A Vimperator wikijének linkje:

http://vimperator.cutup.org/index.php?title=Main_Page