

---

Bugya Titusz

**A GRASS**  
**térinformatikai rendszer kézikönyve**

**Második kötet:**  
**GRASS haladóknak; térképi- és szakadatok kezelése,**  
**előállítás, tárolása és konvertálása**

Jelen dokumentum teljes egészében  
**Creative Commons – Nevezd meg! – Így add tovább!**  
**2.5 Magyarország Licenc**  
alá tartozik.

A könyv elkészítését  
**az fsf.hu Alapítvány**  
(a magyar szabadszoftver alapítvány)  
támogatta.

**Pécs, 2010**

---

# Tartalomjegyzék

1.Bevezetés.....	5
2.GRASS haladóknak.....	6
2.1.Kinek szól ez a kötet?.....	6
2.2.Hogyan is használjuk a GRASS-t?.....	6
2.2.1.Hogyan szervezzük a munkánkat a GRASS-ban?.....	7
2.2.2.A munkaszervezésről általában.....	8
2.3.Régiók kezelése a GRASS-ban.....	9
2.3.1.Régióbeállítás a grafikus felületen keresztül.....	11
2.4.A GRASS és a vetületek.....	16
2.4.1.Új location létrehozása.....	17
2.4.2.Vetületi adatok tárolása a GRASS-ban.....	27
2.5.Digitalizálás a GRASS-ban.....	35
2.5.1.Raszteres digitalizálás a GRASS-ban.....	36
2.5.2.Vektoros digitalizálás.....	38
2.5.3.A digitalizáló beállításai.....	43
2.5.4.A digitalizálás menete a GRASS-ban.....	45
2.6.Adattáblák és adatbázisok kezelése a GRASS-ban.....	53
2.6.1.Adatbázis és adattábla.....	53
2.6.2.Vektoros térkép és adattábla csatolása.....	53
2.6.3.Adatbázis-műveletek a GRASS-ban.....	55
2.7.Exportálás és importálás a GRASS-ban.....	59
2.7.1.ASCII állományok és a GRASS.....	61
2.7.2.Raszteres állományok exportálása.....	63
2.7.3.Vektoros állományok exportálása.....	70
2.7.4.Raszteres állományok importálása.....	74
2.7.5.Vektoros állományok importálása.....	84

2.8.Parancsok használata a GRASS-ban.....	91
2.8.1.Parancssori segédeszközök.....	95
2.8.2.Automatizálási lehetőségek.....	98
2.9.Adattárolás a GRASS-ban: könyvtárszerkezet és fájlok.....	101



# 1. Bevezetés

Mindenek előtt: ebben a jegyzetben – néhány ritka kivételtől eltekintve – a „raszteres térinformatikai modell” kifejezés helyett néhol a „raszteres térkép” megnevezés szerepel, a „vektoros térinformatikai modell” helyett pedig „vektoros térkép”. Feltételezem, hogy az olvasó tisztában van a „térkép” valamint a „térinformatikai modell” kifejezések jelentésével és ennek megfelelően azt is érti, hogy e két fogalom jelentése nem azonos. Feltételezem azt is, hogy az olvasó tisztában van vele: a megjelenített (monitoron látható vagy kinyomtatott) térinformatikai modell, vagyis egy adott térinformatikai modell megjelenítése még nem térkép. Ennek ellenére – hétköznapi gyakorlatban megszokott módon – a kötetben szinonimaként szerepelnek e fogalmak. Ez ugyan elvileg hibás eljárás és a téma szakértői méltán kárhoztathatnak érte, de egyrészt én előre szóltam, másrészt pedig vegyük észre, hogy „raszteres térkép” sokkal rövidebb kifejezés mint a „raszteres térinformatikai modell”. Előfordul az is, hogy a „raszteres réteg”, illetve a „vektoros réteg” megnevezést használom, ami nem tekinthető aggályosnak. Azon olvasóknak, akik a fentiekben foglaltakat nem értik, jó hír lehet, hogy az alapszintű ismeretek elsajátításához valószínűleg nem is lesz szükségük e különbségtétel mélyebb megértésére. Mindazonáltal az ajánlott irodalom tartalmaz útmutatót a kérdés iránt mélyebben érdeklődők számára e téma mélyebb elsajátításához is.

A GRASS használatával kapcsolatos alapfogalmak és alapismeretek a kézikönyv első kötetében található, mely szintén szabadon tölthető le ugyanerről a honlapról (<http://foldrajz.ttk.pte.hu/grass>).

## 2. GRASS haladóknak

### 2.1. Kinek szól ez a kötet?

A GRASS haladóknak azoknak szól, akik már képesek a GRASS alapszintű kezelésére, vagyis a *GRASS kezdőknek* című részben foglaltakat maradéktalanul elsajátították és gyakorolták. A továbbiakban olvasható gondolatmenetek és eljárások megértéséhez szükséges a használt térinformatikai eljárások és szakkifejezések ismerete is. Amennyiben az olvasó úgy tapasztalja, hogy e téren zavaró hiányosságai vannak, érdemes lehet az Internet segítségével összeszedni a hiányzó ismereteket. Erre a célra ajánlható például a *wikipedia.org*, a *mimi.hu* és a *terinformatika.hu* is, de persze nagyon sok egyéb, a témával foglalkozó oldalt találhat, aki időt szán a keresésre.

Mindenesetre azt érdemes szem előtt tartani, hogy a továbbiak megértése nem feltétlenül lesz egyszerű, esetenként nem is biztos, hogy elsőre sikerülni fog. Sajátossága még e résznek, hogy mivel a GRASS-t már nagy vonalakban ismerőknek szól, sokkal kevesebb ábra és képernyőkép található benne.

És végül: érdemes lehet ezért az olvasottakat azon melegében a számítógépen is kipróbálni, gyakorolni.

### 2.2. Hogyan is használjuk a GRASS-t?

A GRASS használata alapvetően két úton lehetséges: parancsok megadásával konzolból, illetve grafikusan, ikonokkal, a *GIS Manager*, a *Map Display* és a további megnyíló ablakok értelemeszerű használatával. Rögtön az elején szögezzük le: egyik használati mód sem jobb mint a másik. Inkább arról van szó, hogy mindegyiknek más az erőssége. Éppen ezért ajánlott mindkét megközelítés megismerése, mert így kombináltan használhatjuk azokat. Egy adott probléma megoldásakor ugyanis lehetőségünk van arra, hogy egyszerre vegyük igénybe a grafikus és a parancssoros lehetőségeket is és ehhez semmiféle bonyolult eljárásra nincsen szükségünk. A grafikus felületen való munka erőssége az intuitív használat lehetősége: nem szükséges minden részletet megtanulni vagy a kézikönyvekből kikeresni, az ikonok és menük segítségével a kicsit is gyakorlott felhasználó könnyen rátalál az éppen keresett eszközre és azt viszonylag könnyen használni is tudja majd. Bizonyos esetekben nagyon jól használható a grafikus felület kísérletezésre is: könnyen ki lehet próbálni az egyes funkciók hatását a mintaállományra. Mások a parancssor előnyei. Egyik előnye a változatlanág. A GRASS fejlődése során a parancsok neve és szerkezete változatlan

maradt és ez várhatóan a jövőben is így marad, legfeljebb bővül az utasításkészlet. Vagyis, az egyszer megtanult parancsok továbbra is használhatóak lesznek, az egyszer megírt scriptek továbbra is eredményesen futtathatók maradnak. Ugyanez az állandóság nem mondható el a grafikus felületről, hiszen annak éppen az az egyik fontos sajátossága, hogy változékony, igazodik az éppen elterjedt grafikus sablonokhoz, megjelenítési eljárásokhoz és a divathoz. Ha csak ennyire lenne jó a parancssor, már megérné megismerkedni vele, de ennél többről van szó. A parancsok a GRASS-ban jól áttekinthető rendszert alkotnak, használatuk könnyen elsajátítható. A parancsok segítségével a GRASS *minden* funkciója teljes mértékben automatizálható, mi több, arra is lehetőség van, hogy külső programokat is bevonjunk az automatizálásba. A parancssor használata tehát mindazon esetekben nagyon kényelmes, amikor a munka jól automatizálható, mondjuk azért, mert sokszor ismétlődő lépésekből áll. De akkor is nagy hasznát vesszük, amikor az egymásra épülő, egymás után következő részfeladatok futtatása nagyon sokáig tart. Ilyenkor nem kell a gép előtt ülve hosszasan várni arra, hogy a következő műveletet egér segítségével kijelölhessük, hiszen a GRASS ezt automatikusan elvégzi helyettünk. Bizonyos esetekben pedig kísérletezésre is jobb, mint a grafikus munkafelület. Ha ugyanis a kísérletezés abból áll, hogy több egymás utáni művelet hatását vizsgáljuk különböző paraméterek megadása mellett, akkor érdemes lehet megírni a feladat parancsait és azokat lefuttatni. Egészen meglepő időnyereségre lehet így szert tenni. És végül ne feledjük: a parancssoros műveletvégzés arra is lehetőséget ad, hogy távoli gépen futtassuk a GRASS-t (mondjuk egy nagy teljesítményű szerveren) és azt is megengedi, hogy a feladatokat ütemezve, adott időben hajtassuk végre (mondjuk munkaidő után, amikor kicsi a terhelés).

Láthatjuk tehát, hogy a parancssor és a grafikus felület erősségei – és persze hátrányai – nem ugyanazok, vagyis komplementerek, kiegészítik egymást. Célszerű tehát a hatékony munkavégzéshez mind-egyiket megismerni.

### 2.2.1. Hogyan szervezzük a munkánkat a GRASS-ban?

A GRASS munkaszervezési lehetőségei nagyon kiterjedtek. Nem csak arra terjednek ki, hogy lehetővé teszik az éppen a gép előtt ülő egyetlen felhasználó térképeinek áttekinthető kezelését. Lehetővé teszik azt is, hogy egymástól távol eső felhasználók egyidőben ugyanarra a szerverre, ugyanazon a területen dolgozzanak úgy, hogy munkájukat egységes szempontok és nevezérendszer szerint végzik, de egymás munkáját nem zavarják. Röviden: a GRASS magas szinten támogatja a hálózati csoportmunkát és az időben elhúzódó vagy periodikus munkavégzést. A munkaszervezés megértése nem különösebben bonyolult dolog, de kétségtelenül igényel némi figyelmet és gyakorlást a használata, mert meglehetősen sokrétű. Mindenesetre az elmondható, hogy érdemes alaposan megismerni a lehetőségeket, mert – különösen az összetett vagy hosszú időtávlatba mutató feladatok esetén – nagyon megkönnyíti a felhasználó dolgát a GRASS. Persze csak akkor, ha a felhasználó hagyja...

## 2.2.2. A munkaszervezésről általában

Legyünk őszinték magunkhoz: a munkaszervezés fontosságát senki sem veszi komolyan mindaddig, míg bele nem bukik egy olyan feladat (határidőre történő) elvégzésébe, melynek nehézsége abban áll, hogy az alábbiak közül egyszerre több is teljesül:

- a munka igen sokrétű;
- sok, gyakran csak egy-két apróságban különböző állományt kell kezelni, de pont e kicsi különbségek a lényegesek;
- a feladat egyszerű elemek elvégzéséből áll, de ezek sokféleképpen kombinálódnak;
- nagyon sok állománnyal kell dolgozni;
- a munkát többen végzik, akik esetleg nem is találkoznak egymással;
- mások részeredményeit kell beépíteni a saját munkába úgy, hogy azok végül egy egységes egészet képezzenek;
- a feladat jellegzetessége, hogy időben nagyon elhúzódik, akár évek múlva kell pontosan ott folytatni, ahol abbahagytuk, vagy ahol az elődünk abbahagyta;
- több, egymástól merőben eltérő kimenetet is kell készíteni különböző célokra;
- a munkát pontosan, reprodukálhatóan, ellenőrizhetően kell dokumentálni;
- a feladat elvégzéséhez több, egymástól független számítógépet is igénybe kell venni;
- egyéb nehezítések, amik a fenti listában nem szerepelnek.

Ezekkel a problémákkal az átlagos felhasználó az egyetemi, illetve általában az iskolai tanulmányok alatt, egy-egy ritka kivételtől eltekintve nem találkozik. Az iskolai feladatok jellemzően egyszerűek (még a bonyolultnak tűnők is), rövid gondolkodás után, egy- vagy néhány állomány felhasználásával megoldhatóak. A valóságban azonban tényleg előfordulhatnak olyan munkák, amelyek gondos tervezés- és az azt követő munkaszervezés nélkül nem végezhetőek el, de legalábbis nem határidőre vagy nem a kívánt minőségben.

Első tehát a munka megtervezése aztán erre építve a munka megszervezése. Csak ezután érdemes nekifogni a tulajdonképpeni munkavégzésnek.

A GRASS esetében a munkaszervezés az alábbiakat jelenti

- Használjuk ki a GRASS adattárolási sajátosságaiból eredő előnyöket! Egy-egy új földrajzi terület feldolgozásához hozzuk létre az adott terület location-ját!
- A munka tervezése során ne féljünk új mapsetek létrehozásától! Ne feledjük: a PERMANET mapset kötelező eleme minden location-nak, de célszerű létrehozni továbbiakat is, amelyekben a tulajdonképpeni munkát végezzük! Így a PERMANET-ben tárolt állományok változatlan formában megmaradnak, az eredmények reprodukálhatóak lesznek és ellenőrizhető lesz a munka is.
- Használjuk ki a *GIS Manager* listakezelési lehetőségeit! A feldolgozandó térképeket ne ömlesztve helyezzük a lástába, hanem szervezzük őket mappákba, ha kell almappákba, használjunk be-

szédes neveket és mentsük el a listát! Ha több gépen is ugyanazzal a problémával dolgozunk, használjuk mindenütt ugyanazt a listát!

- Fordítsunk figyelmet az *Output* ablakban megjelenő üzenetekre! Olvassuk el a megjelenő üzeneteket minden elvégzett művelet után – de legalábbis vessünk egy pillantást –, mert sok fölösleges munkától óvhatjuk meg magunkat egy véletlen hiba után.
- Ha a feladat jellege megkívánja, rendszeresen mentsük el az *Output* ablak tartalmát! Így egyrészt a munkánk ellenőrizhetővé válik, másrészt a később előkerülő hibák forrása könnyebben visszakereshető lesz, harmadrészt pedig a munkamenet pontosan reprodukálható lesz. Ez utóbbi különösen akkor jön jól, ha hosszú távú monitorozásos vizsgálatok keretében kell feladatokat elvégezni.

## 2.3. Régiók kezelése a GRASS-ban

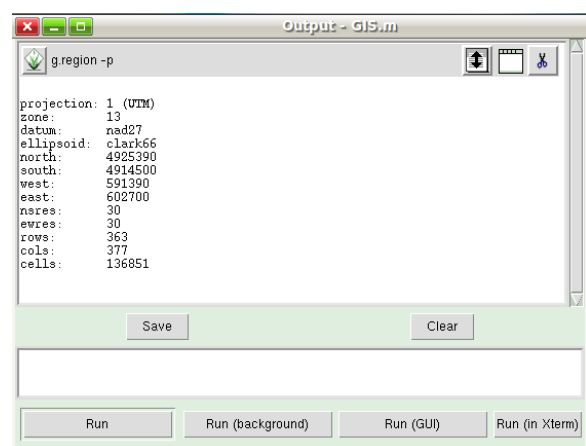
A régiókezelés elsajátítása nagyon fontos a GRASS használatához. A GRASS számára a régió az a terület, amelyre vonatkozóan a műveleteket el kell végezni. Másképpen fogalmazva ez azt jelenti, hogy a számítási műveleteket a GRASS az adott mapset térképeinek arra a részére végzi el, amely részt az éppen aktuálisan beállított régió lefed. A régió tulajdonsága a kiterjedésén kívül a felbontása is. Jó magyar fordítása lehet a „terjedelem”.

A régiók kezelése grafikus felületen a *Config* menü *Region* almenüjéből lehetséges. A *Display region settings* megjeleníti az éppen aktuális beállítási értékeket, míg a *Change region settings* pontra kattintva magunk adhatjuk meg a régió kívánt paramétereit. Terminálból a *g.region* parancs megadásával indítható a régióbeállítás.

A GRASS-ban, ha mást nem definiálunk, az alapértelmezett régió és az éppen aktuális régió, vagyis amelyen a GRASS a műveleteket elvégzi, egybeesik. Mind az alapértelmezett, mind az aktuális régió megváltoztatható. A változtatás lehetősége azt jelenti, hogy módosíthatjuk a határait és a felbontását, természetesen mindegyiket akár három dimenzióban is.

Az *alapértelmezett régió* tulajdonságai lekérdezhetőek a *Config* menü *Region* almenüje *Display region settings* pontjával. A lekérdezés eredménye az *Output* ablakban jelenik meg (2.1. ábra). Látható, hogy a régió adatai az alábbiak:

- **projection:** a vetület száma a GRASS-ban és neve,
- **zone:** a zóna száma (ha az adott vetületben ez értelmezhető),



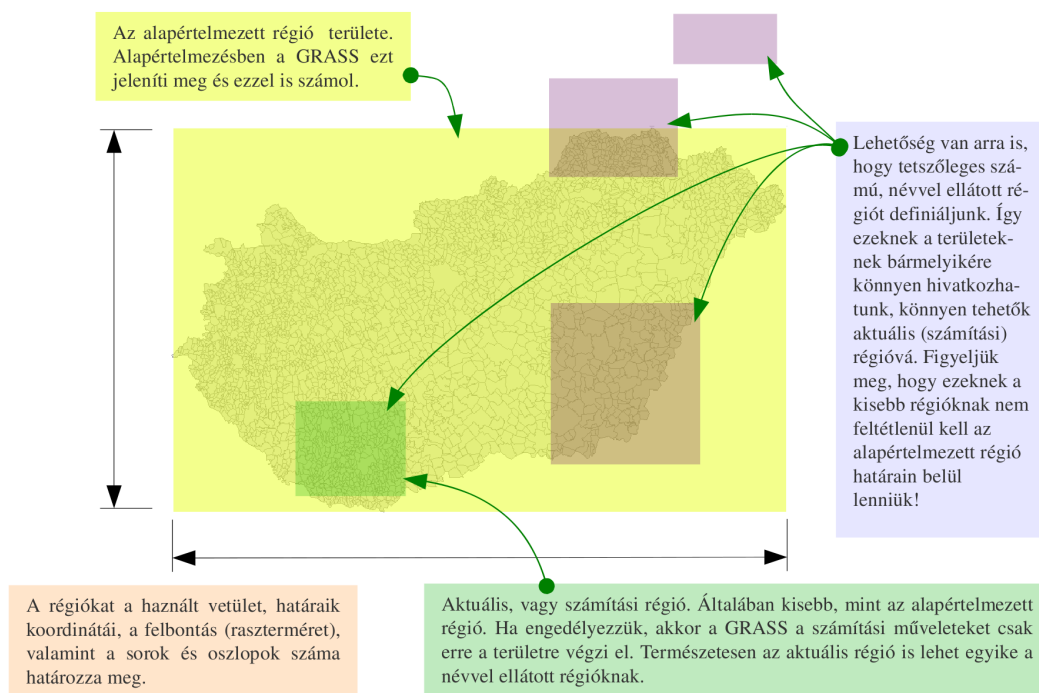
2.1. ábra

A *Display region settings* kimenete a *spearfish60* esetében (alapértelmezett beállításokat használva)

- **datum:** a használt geodéziai dátum,
- **ellipsoid:** a használt ellipszoid megnevezése,
- **north:** a régió északi szegélye a használt vetületi rendszerben mérve,
- **south:** a régió déli szegélye a használt vetületi rendszerben mérve,
- **west:** a régió nyugati szegélye a használt vetületi rendszerben mérve,
- **east:** a régió keleti szegélye a használt vetületi rendszerben mérve,
- **nsres:** észak-déli irányú felbontás a vetületi rendszer mértékegységében mérve,
- **ewres:** kelet-nyugati irányú felbontás a vetületi rendszer mértékegységében mérve,
- **rows:** sorok száma (raszteres térképekre vonatkozóan),
- **cols:** oszlopok száma (raszteres térképekre vonatkozóan),
- **cells:** raszterek (cellák) száma.

Ezeket az adatokat a GRASS az adott location PERMANENT mapsetjében levő DEFAULT\_WIND nevű állományban tárolja (részletesen lásd: *Vetületi adatok tárolása a GRASS-ban* című rész).

Az *aktuális régió*, más néven *számítási régió* (*computational region*), a GRASS-ban azt a területet jelenti, amelyre vonatkozóan a GRASS elvégzi a kívánt műveleteket (2.2. ábra). Így lehetőség nyílik arra, hogy egy térképnek csak egy kiválasztott területén végezzünk műveleteket. Ez azért jó, mert így nem kell



**2.2. ábra**  
**Az alapértelmezett régió, a számítási régió és a névvel ellátott régió értelmezése a GRASS-ban**

kisebb résztérképet (vagy résztérképeket) létrehozunk, ugyanakkor a szükséges számítási kapacitás is kisebb lehet, valamint a műveletvégzési idő is lerövidülhet. Az aktuális régió kijelölése történhet egerrel, a grafikus felületen, illetve a határoló élek koordinátáinak megadásával, numerikusan. Lehetőség van arra is, hogy egy régiót a definiálása után elmentsünk valamilyen néven és a továbbiakban így hivatkozunk rá. Vagyis, ha mondjuk egy nagyobb térkép részét képező kisebb területet külön régióként kívánunk kezelni, akkor megadjuk a határait, adunk neki egy nevet és ha később meg kívánjuk jeleníteni, e névre hivatkozva könnyen megtehetjük.

## 2.3.1. Régióbeállítás a grafikus felületen keresztül

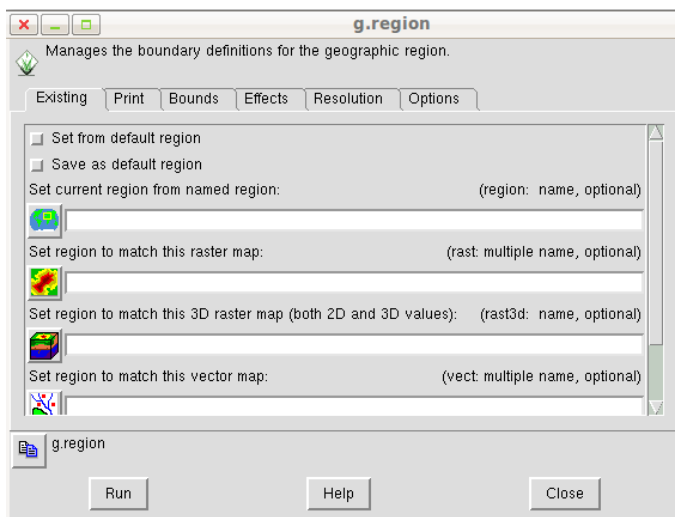
A régióbeállítások módosítására használjuk a *Config* menü *Region* almenüjének *Change region settings* pontját! A megjelenő ablakban (2.3. ábra) az alábbi beállítások tehetők.

### Existing

Ha a régiót határait úgy kívánjuk definiálni, hogy valamely már létező térkép vagy régió kiterjedésével egyezzen meg, akkor itt adhatjuk meg, hogy mely térkép vagy régió legyen az. Ha a régió határainak megadására ezt a módot választjuk, a *Bounds* részben NE adjunk meg semmit sem, mert az ellentmondáshoz vezethet.

Értelemszerűen, ha például azt kívánjuk, hogy a régió határai az *elevation.dem* nevű raszteres állomány határaival essenek egybe, akkor írjuk be az *elevation.dem* nevet a *Set region to match this raster map* mezőbe. Ne feledjük, hogy a GRASS adattárolási sajátosságai miatt az ugyanazon mapsetben található térképek általában, de nem feltétlenül, ugyanazt a területet is fedik le (vagyis „ugyanakkorák”). Ebből következően általában mindegy, hogy egy adott mapset melyik térképét választjuk, mindegyik egyforma. Ez azonban nem *feltétlenül* igaz, csak általában, éppen ezért, ha bizonytalanok vagyunk, inkább válasszuk annak a térképnek a nevét, amelyik biztosan azt a területet fedi le, amelyikre szükségünk van. A *spearfish60* location PERMANENT mapsetjében például mindegyik térkép (tehát a raszteresek és a vektorosak is) pontosan ugyanazt a területet ábrázolják, a megfelelő oldalaik koordinátái ugyanazok.

Jegyezzük meg, hogy mivel itt csak a térképek által lefedett területről van szó, vagyis annak a határaitól, ezért teljesen mindegy, hogy vektoros, vagy raszteres térképet választunk a régió határainak definiálására. Csak az a lényeg, hogy olyan térképet válasszunk, amely pontosan a kívánt területet fedi le. Így, ha például a *spearfish60* location PERMANENT mapsetjéből a *roads* nevű vektoros térképet választjuk a határok definiálására, a beállítás természetesen a raszteres térképekre is vonatkozni fog.



2.3. ábra  
A régióbeállítás ablaka

Az *Existing* részben választható a *Save as default region* opció. Ha ezt engedélyezzük, akkor az általunk megadott paraméterek alapján létrehozott régió lesz az alapértelmezett régió. A *Set from default region* engedélyezésével az alapértelmezett régiót tesszük meg aktuális régiónak.

A *Set current region from named region* mezőben megadhatjuk valamely már korábban definiált és névvel ellátott régiót is, ebben az esetben az lesz megtéve aktuális régiónak.

### Print

Az aktuális régióra vonatkozó beállítások értékeit jeleníthetjük meg vele. Válaszuk ki a megjeleníteni kívánt értékeket, majd nyomjuk meg a *Run* gombot! Az eredmény az *Output* ablakban olvasható.

- **Print the current region** – Megjeleníti az aktuális régió beállításait.
- **Print the current region in lat/long using the current ellipsoid/datum** – Úgy jeleníti meg az aktuális régió sarokpontjainak és középpontjának koordinátáit, hogy azokat földrajzi szélesség és hosszúság formába számolja át, az adott régió ellipszoidját és geodéziai dátumát figyelembe véve.
- **Print the current region extent** – Csak az aktuális régió ÉD és KNy irányú kiterjedését írja ki.
- **Print the current region map center coordinates** – Az aktuális régió közepének a koordinátáit adja meg az adott location vetületi rendszerét használva.
- **Print region resolution in meters (geodesic)** – Az adott régió felbontását adja meg méterben.
- **Print the convergence angle (degrees CCW)** – Az adott régió középpontjában a vetületi észak és a valódi északi irány közötti eltérést írja ki. Az irányt az óramutató járásával szemben kell érteni, a kiírt érték pedig tizedestört alakban jelenik meg és fokban értendő (tehát nem fok, perc, másodperc!).
- **Print also 3D settings** – A 3D régióbeállításokat is megjeleníti, akkor is, ha az adott térkép nem ilyen.
- **Print the maximum bounding box in lat/long on WGS84** – WGS84 dátumon értelmezhető földrajzi szélesség és hosszúság formában adja meg annak a foktrapéznek az oldalait, amelyet belülről érint az aktuális régió (vagyis, amelyikbe pont beleillik az aktuális régió).
- **Print in shell script style** – Bármelyik előző opcióval használható. Hatására a kimenet úgy változik, hogy a GRASS automatizálásához is használható (lásd később) script formátumban jelenik meg.

### Bounds

Ha a régió határait úgy kívánjuk definiálni, hogy megadjuk a határoló északi-, déli-, keleti- és nyugati oldal koordinátáját, akkor azt itt tehetjük meg. Ha ezt a meghatározási módot választjuk, az *Existing* részben NE adjunk meg semmilyen értéket, hiszen ellentmondás merülhet fel.

- **Value for the northern edge** (*southern edge, eastern edge, western edge, top edge, bottom edge*) E mezőkben rendre adjuk meg az északi/déli/keleti/nyugati/felső/alsó oldal megfelelő koordinátáját! Ha a definiálandó régióknak csak két dimenziós, akkor a *top edge* és a *bottom edge* mezőket hagyjuk üresen! Ne feledjük, hogy a tizedes törtek megadásakor tizedes pontot kell használnunk, nem pedig vesszőt!

Ha például a *spearfish60* locationban akarunk egy régiót definiálni, úgy, hogy az csak az alapértelmezett régió DNY-i sarkára terjedjen ki, megadhatjuk az alábbi koordinátákat:

north: 4920372

south: 4916289

east: 597077

west: 591639

- **Shrink region until it meets non-NULL data from this raster map.** Úgy módosítja a bállított régiót, hogy addig mozgatja a határait, míg az itt megadott térképen el nem éri az első nem-NULL értékű rasztert (2.4. ábra).

Ez a meglehetősen bonyolult tűnő meghatározás az alábbiakat takarja. Tétélezzük fel, hogy van egy olyan olyan raszteres térképünk, amelynek csak valamely jól meghatározható részén csoportosulnak nem-NULL értékű raszterek és a térkép nagyobb, összefüggő részei, mondjuk az egyik fele, teljesen üres. Ha feladatunk során ugyanebben a mapsetben levő, ugyanekkora kiterjedést ábrázoló térképekkel kell fedvényeket létesítenünk, akkor érdemes lehet ezt az opciót igénybe vennünk. Ekkor a beállított régió úgy változik meg, hogy annak mindegyik határa a munkatérképünk legalább egy nem-NULL értékű elemével legyen szomszédos. Így biztosak lehetünk benne, hogy minden, valódi jelentéssel bíró (tehát nem-NULL rasztert) figyelembe vetünk, de a számítási igény és idő mégis jelentősen csökken, mert csökkentettünk a fölöslegesen elvégzendő munkát. Ez abból következik, hogy figyelmen kívül hagyjuk a NULL értékű rasztereket, melyeken a legtöbb művelet úgyse végezhető el.

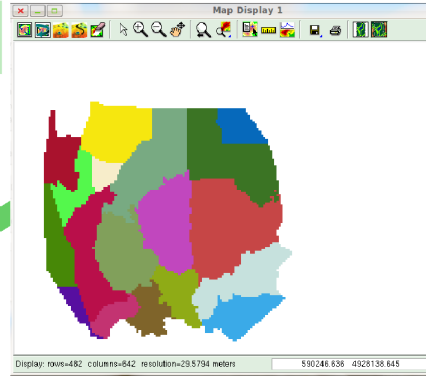
Még egyszerűbben fogalmazva: a számítási régió határai úgy módosulnak, hogy a lehető legtöbb NULL értékű raszter maradjon rajtuk kívül, viszont mindegyik nem NULL értékű raszter a régióon belül maradjon.

Mindenesetre egy példa valószínűleg jól megvilágítja ennek az opciónak a lényegét.

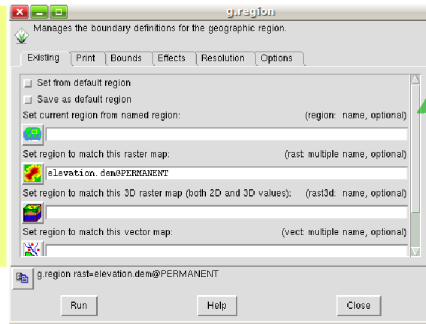
1. Jelenítsük meg a *trn.sites* raszteres térképet, úgy, hogy a megjelenítőablak *Zoom to...* menüjében a *zoom to default region* pontot válasszjuk! Látjuk, hogy a térkép elég nagy része nem tartalmaz adatot, mert NULL értékű (vagyis fehér). Csak a nyugati oldalán vannak ábrázolható adatok, de azok is úgy, hogy sehol sem érnek ki a régió széléig.
2. Most válasszuk régióbeállítás ablakot (*g.region*), és az *Existing* részben adjunk meg egy térképet, amellyel régiót definiálunk! Mivel a *spearfish60* location PERMANENT mapsetjében mindegyik térkép ugyanazt a területet ábrázolja, akármelyiket választhatjuk (lásd erről fentebb bővebben). Adjuk meg mondjuk a *Set region to match this raster map* mezőben az *elevation.dem*-et! Lépünk a *Bounds* részhez és a *Shrink region until it meets non-NULL data from this raster map* mezőben válasszuk ki a *trn.sites*-t! A *Run* megnyomása után az *Output* ablakban láthatjuk, hogy lefutott a kérés.
3. Térjünk vissza a megjelenítéshez! A megjelenítési listában helyezzünk el két raszteres elemet és a felsőhöz rendeljük mondjuk a *landuse*, az alsóhoz pedig a *trn.sites* térképet! A felsőt állítsuk kissé átlátszóra, hogy jól lássuk a *trn.sites*-t is!

## 2.3. RÉGIÓK KEZELÉSE A GRASS-BAN

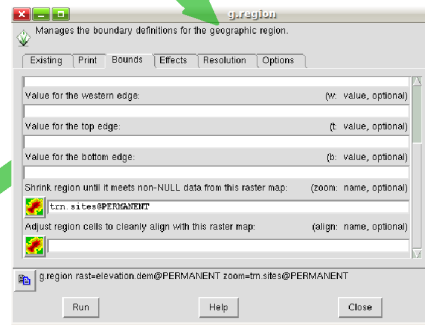
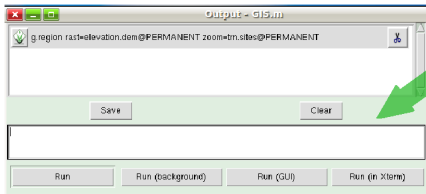
A *trn.sites* térkép teljes területe, az alapértelmezett régióban megjelenítve. Jól látható, hogy a terület zöme NULL értékű (fehér).



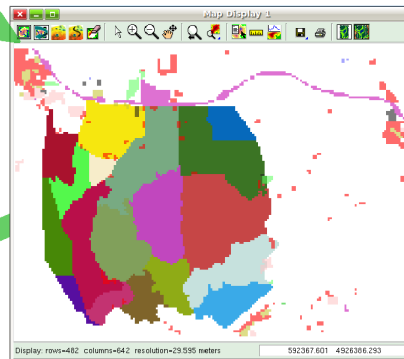
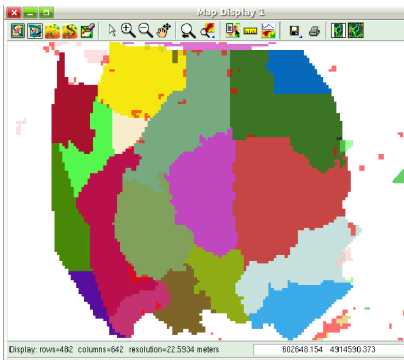
A *g.region* ablakban, az *Existing* részben kell megadni azt a térképet, mellyel a használandó régiót definiáljuk.



A Run megnyomása után az Output ablakban tájékoztatást a GRASS a parancs lefutásáról



A *g.region* ablak *Bounds* részében adjuk meg azt a térképet, mellyel a régió határait igazítjuk.



A *landuse* térkép és a *trn.sites* együttes megjelenítése a művelet előtt.

A *landuse* térkép és a *trn.sites* együttes megjelenítése a művelet után, a *Zoom display to computational region* hatására csak a számítási terület látszik, mely a *trn.sites* legszélső nem NULL rasztereig ér. Így a műveleteket is csak ebben a régióban végzi el a GRASS.

## 2.4. ábra

*A Shrink region until it meets non-NULL data from this raster map* funkció használata

4. Ha most a *Map Display* ablak *Zoom to...* menüjéből a *Zoom display to computational region (set with g.region)* pontot választjuk, akkor láthatóvá válik az eredmény. A régió határai úgy módosultak, hogy éppen érintsék a *trn.sites*-t, így a lehető legkevesebb NULL raszter maradt a régióban. Természetesen a *landuse*-ből is csak annyi látszik, ami éppen erre a területre esik. Így, ha fedvényműveleteket végzünk (mondjuk összeszorozzuk a térképeket), akkor, mivel a műveletek csak az aktuális régióon belül hajtódnak végre, sokkal gyorsabb lehet a munka, mint ha a teljes alapértelmezett régiót használtuk volna.

Persze, a régióbeállítás és a *mapset* sajátosságai miatt ugyanez lett volna az eredmény, ha a *g.region* ablakban nem az *elevation.dem*-et, hanem valamelyik vektoros térképet adjuk meg és a megjelenítésnél is vektoros térképet használunk a *landuse* helyett (érdemes kipróbálni!).

- **Adjust region cells to cleanly align with this raster map** – úgy állítja át a régió felbontását, vagyis a raszterméretét, hogy a lehető legjobban illeszkedjék az itt megadott raszteres térképhez.

### Effects

- **Save the current region to region definition file** – Ha ebben a mezőben megadunk egy nevet, akkor a régióbeállításainkat ezen a néven tárolja a GRASS. Így lehetőségünk lesz rá, hogy később egyszerűen visszatérjünk az ekkor megtett beállításokhoz. Ha ilyet szeretnénk tenni, akkor vagy az *Existing* részben levő *Set current region from named region* mezőben kell megadnunk a használni kívánt régió nevét, vagy a *Map display* ablak *Zoom to...* menüjéből kell választanunk a *Zoom to saved region* pontot és kiválasztani a használandó régiót.
- **Do not update current region** – Ha azt szeretnénk, hogy a megadott beállítások egyelőre ne lépjenek érvénybe, akkor engedélyezzük ezt az opciót. Ilyenkor a Run megnyomása után az ablakban tett beállításokat a megadott állományban tárolja a GRASS, de nem hajtja végre azokat. Az opció alapértelmezése a kikapcsolt állapot.

### Resolution

A régió felbontásának megadására szolgál. Vagy azt adjuk meg, hogy hány oszlop és sor alkossa a raszteres térképet, vagy azt, hogy az egyes irányokban (K-Ny, É-D, fent-lent) mekkora legyen a felbontás számszerű értéke (a térkép mértékegységében, például méterben). Ha a sorok és oszlopok számát kívánjuk megadni, használjuk az első két mezőt, ha pedig közvetlenül a felbontást, akkor a többit. Ezek között választhatunk, hogy a felbontás minden irányban ugyanakkora legyen, vagy irányonként különböző.

- **Number of rows in the new region** – sorok száma az új régióban.
- **Number of cols in the new region** – oszlopok száma az új régióban.
- **Grid resolution 2D (both north-south and east-west)** – itt azt adhatjuk meg, hogy mekkora legyen a 2D felbontás É-D-i és K-Ny-i irányban, de mindkét irányban ugyanakkora,
- **3D grid resolution (north-south, east-west and top-bottom)** – azt adhatjuk meg, hogy mekkora legyen a 3D felbontás É-D, K-Ny és fent-lent irányban, de mindhárom irányban ugyanakkora,
- **North-south grid resolution 2D** – az észak-déli irányú felbontás értéke a térkép mértékegységében. Eltérhet a kelet-nyugati és a fent-lent irány felbontásától.
- **East-west grid resolution 2D** – a klet-nyugati irányú felbontás értéke a térkép mértékegységében. Eltérhet az észak-déli és a fent-lent irány felbontásától.

- **Top-bottom grid resolution 3D** – a fent-lent irányú felbontás értéke a térkép mértékegységében. Eltérhet a kelet-nyugati és az észak-déli irány felbontásától.

### Options

Két opciót tartalmaz.

- **Allow overwrite** – engedélyezi a meglévő állomány felülírását (például elnevezett régió mentésekor).
- **Run quietly** – a Run megnyomása után a program lefut, de nem ír üzeneteket az Output ablakba.

## 2.4. A GRASS és a vetületek

A vetületi adatok és a geodéziai dátumok korrekt kezelése a térinformatikai alkalmazások és a térinformatikai munka egyik sarokköve. A GRASS a vetületi adatokkal és a geodéziai dátumokkal való műveleteket magas szinten támogatja, mert nemcsak igen sok vetületi rendszert képes használni, de az ezek közötti transzformációkat is elvégezhetjük vele, és természetesen egyéni dátumokat is létrehozhatunk.

Vetületek és geodéziai dátumok kezelésére akkor lehet szükségünk, ha:

- új location-t kell létrehoznunk,
- meglévő location-ba kell egy másik, eltérő vetületű location-ból térképet importálnunk,
- meglévő location vetületi adatait kell módosítanunk,
- vetületi információkat kell kérnünk egy térképről és azt külső célra átadni.

Ezeket a feladatokat a GRASS-ban mind megoldhatjuk, némelyiket több, különböző úton indulva is.

Ha a GRASS-szal való ismerkedés után valamilyen munkát is szeretnék végezni vele, az első probléma az lesz, hogy új locationt kell készíteni. Ennek az új locationnak olyannak kell lennie, hogy benne a munkatérképek – melyek vélhetően nem a spearfish60 location által leírt régió területére esnek – problémamentesen feldolgozhatók legyenek. Vagyis, ha mondjuk a Magyarországon szokásos EOVRa alapuló térképeket kell feldolgoznunk, akkor először is szükségünk lesz egy EOVRa szerinti (EOV vetületű) locationra. Éppen ezért kezdjük a GRASS és vetületek viszonyával való ismerkedést az új location létrehozásának mikéntjével.

### 2.4.1. Új location létrehozása

Új location-t létrehozhatunk a GRASS indításakor is és akkor is, ha már elindítottuk a programot valamely location és mapset megadásával. Elvi különbség a két eljárás között nincsen, de maga az eljárás különböző. Az új location létrejöttéhez szükséges adatokat az alábbiak egyikével adhatjuk meg:

- már létező, georeferált térkép,
- egyszerű szöveges állományban tárolt vetületi információk,
- EPSG kód,
- egyéni vetületi adatok.

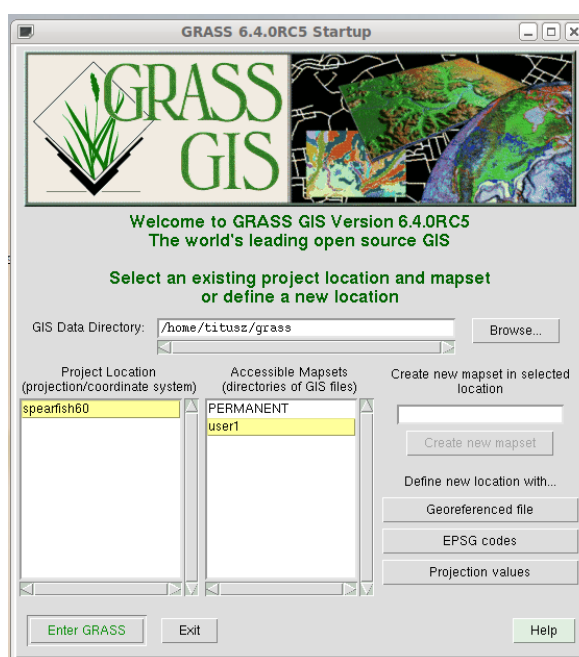
**FONTOS!** Ne feledjük el, hogy új location létrehozása után abban csak a PERMANET mapset-et találjuk, így szükség lehet a munka megkezdése előtt új (saját) mapset elkészítésére!

### 2.4.1.1. Új mapset és új location készítése a GRASS indításakor

A GRASS indításakor (részletesen lásd az első kötetben) nem csak a már meglévő locationokból választhatunk, hanem, ha szükséges, újat is definiálhatunk. Az indítási ablak jobb oldala tartalmazza az ehhez szükséges gombokat (2.5. ábra).

#### Új mapset készítése

Új mapset készítéséhez először a *Create new mapset in selected location* mezőben kell megadni a kívánt nevet (ékezetek és szóköz nélkül), aztán az alatta levő *Create new mapset* gombbal jóváhagyni a kérést. Ezután a kijelölt locationban már ki is választhatjuk az új mapsetet és kezdőhet a munka!



#### Új location georeferált állományból

Az első lehetőség, hogy úgy hozunk létre új locationt, hogy egy, már meglévő georeferált állományból, például térképből olvassuk ki a szükséges vetületi adatokat. Így bármilyen vetületű location definiálható.

Ennek a lehetőségnek a használata roppantul egyszerű. Nincs más dolgunk, mint a GRASS indításakor a *Define new location with...* alatt megnyomni a *Georeferenced file* feliratú gombot és a megjelenő ablak két mezőjét kitölteni. A felső mezőben a létrehozni kívánt új location nevének adjuk meg (ez alapértelmezésben *newLocation*, de természetesen felülírható), az alsóban pedig annak az állománynak a nevét, amely a használni kívánt vetületben tartalmaz adatokat és belőle a szükséges vetületi információkat a GRASS ki tudja olvasni. Ha mindkét mezőt kitöltöttük, nyomjuk meg a *Define location* gombot! Ezzel elkészül az új location és benne a kötelező PERMANENT mapset. A továbbiakban úgy járunk el, mint bármely más location választása esetében.

2.5. ábra

A GRASS bejelentkező ablaka

Felmerül a kérdés: miféle georeferált állományokat fogad el a GRASS, új location létrehozásához? Itt alapvetően olyan, már létező térképi állományokra kell gondolnunk, amelyek tartalmazhatják a szükséges adatokat és a GRASS azokat egyébként képes importálni. Ilyen például az *shp*. Tekintve, hogy az *shp* ráadásul igen gyakori állománytípus, gyakorlatilag az összes térinformatikai rendszer képes írni és olvasni, így használata erre a célra is bátran ajánlható. Azaz, ha valamely terület térképe helyesen georeferált *shp* állományban rendelkezésünkre áll, akkor a bemutatott eljárással minden további nélkül előállíthatunk vele egy GRASS locationt. Így ez a location (pontosabban ennek az alapértelmezett régiója) pontosan az eredeti térkép területét fogja lefedni. Ezek után elindíthatjuk a GRASS-t az új location-nel (és ha kell, új mapset-tel!) és beimportálható a kérdéses *shp* térkép is.

### Új location EPSG kód alapján

A GRASS indításakor választhatjuk azt is, hogy a létrehozandó új location vetületi adatait EPSG kód alapján adjuk meg. Ekkor a megjelenő ablakban meg kell adnunk a létrehozandó location nevét, az EPSG kódot tartalmazó állomány elérési útját és nevét, valamint a használni kívánt EPSG kódot. (Az EPSG egyébként az *European Petroleum Survey Group* »Európai Kőolajkutató Csoport« rövidítése). Ezzel a módszerrel már nem annyira egyszerű az új location létrehozása, mint a georeferált állománnyal, de azért nem is nehéz. A probléma az, hogy míg az előző esetben elég volt megadni egy általunk kiválasztott állomány nevét és elérési útját, most egy vetülethez rendelt azonosítót kell előkeresnünk és azt megadni a GRASS-nak. Az alábbiakban leírt, EOV alapú location elkészítését érdemes kipróbálni és magát a location-t is megcsinálni, mert később még visszatérünk rá.

#### Egyszerűbb esetben

Mindenek előtt értelemszerűen adjunk meg egy nevet az elkészítendő location-nek! Arra azonban figyeljünk, hogy csak olyan nevet adjunk meg, ami még nem foglalt, ellenkező esetben hibüzenet lesz a művelet végeredménye! Figyeljünk arra is, hogy az új location neve ne tartalmazzon ékezetet és szóközt sem.

Ezután térjünk át a második mezőre, ahol az EPSG kódokat tartalmazó állomány elérési útját és nevét kell megadni. Operációs rendszertől és a környezeti beállításoktól függően a GRASS felkínál egy elérési utat és állományt, mely általában tartalmazza a keresett vetület adatait. Ha mégsem így lenne, akkor magunknak kell megadni ezeket az adatokat, erről lentebb még lesz szó.

Ha minden rendben levőnek tűnik, keressük meg a használni kívánt vetület (és geodéziai dátum) EPSG kódját és adjuk meg a harmadik mezőben! Ahhoz persze, hogy meg is találjuk amit szeretnénk, tudnunk kell a nevét, esetleg több formában is. Előfordulhat ugyanis, hogy több formában is szokás hivatkozni egy vetületi rendszerre, de a táblázatban ezek közül az adott esetben csak az egyiket használják és az is lehet, hogy ugyanannak a dátumnak kisebb eltérésekkel több változata is létezik. A magyarországi gyakorlatban általánosan elterjedt EOV-ből szerencsére csak egy van, így könnyű lesz megtalálni. A példa kedvéért hozunk létre tehát egy EOV alapú locationt (2.6. ábra)!

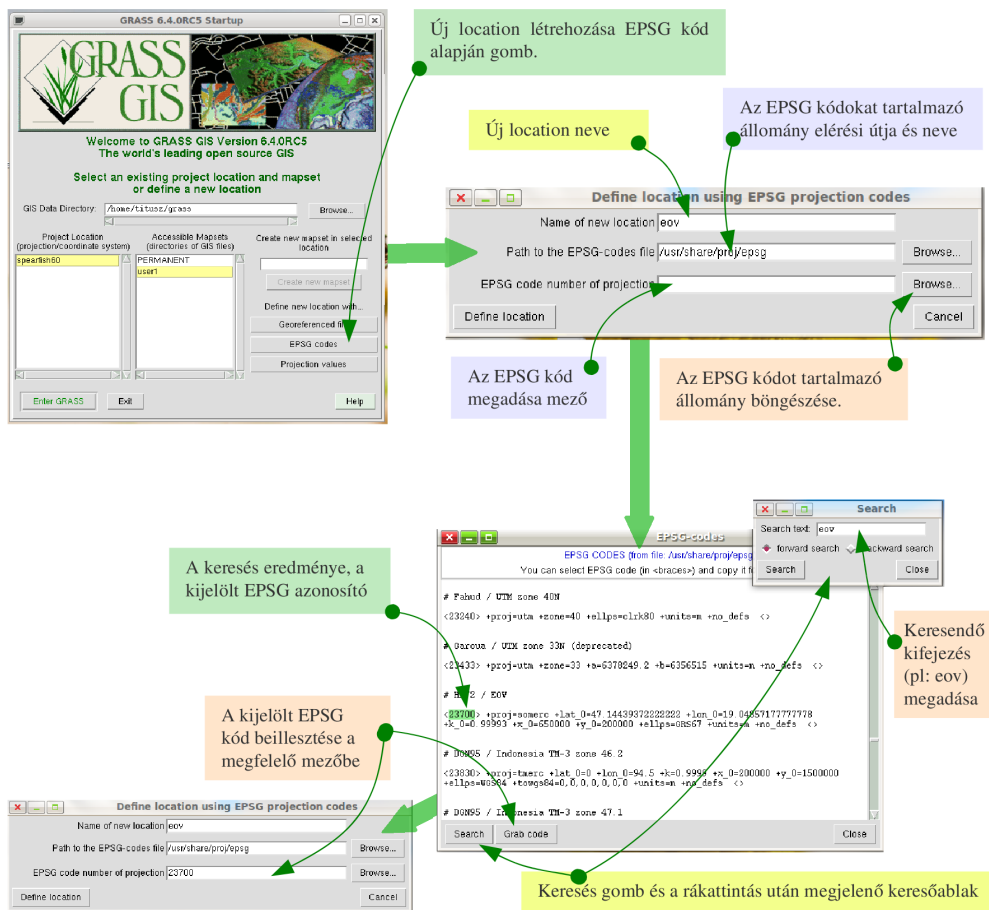
Legyen a location neve mondjuk *eov*, a második mezőben felkínált elérési úton és állományneven pedig ne változtassunk. A harmadik sorban kell megadni a szükséges EPSG kódot, amit – egyelőre – nem tudunk. Tudjuk azonban, hogy amit keresünk, az az EOV és van egy *Brow-*

se (böngéssz) gomb az ominózus harmadik sor végén. Kattintsunk rá! A megnyíló, *Define location using EPSG projection codes* nevű, ablakban tanulmányozhatjuk mindazon geodéziai dátumokat, melyeket a második sorban megadott állomány tartalmaz. Válasszuk most az ablak aljáról a *Search* (keress) gombot és kattintsunk rá! A felugró ablakban adhatjuk meg a kereső-kifejezést, jelen esetben azt, hogy *eov*. Itt nem számít a kisbetű-nagybetű különbség! Az ablaknak két opciója van:

- *forward search* – keresés a szövegben előre felé,
- *backward search* – keresés a szövegben visszafelé.

Mivel a keresőablak megnyitásakor a kurzor a szöveg elején áll, természetesen a *forward search* az alapértelmezett opció.

A *Search* gombra kattintva a keresés elindul, és ha eredményes, akkor a talált kifejezést a szövegben kiemelve jeleníti meg a GRASS. Legyünk azonban körültekintőek és ne elégedjünk meg az első találattal! Lehet, hogy a keresett kifejezés más EPSG kódnál is előfordul. Éppen ezért ha van is találat, keressünk tovább! Mivel jelen esetben a *Search* további lenyomására



2.6. ábra  
Új location elkészítése EPSG kód megadásával

sem jelenik meg újabb találat, fogadjuk el a felkínáltat! Tehát csak egyetlen találatot kapunk, mely így néz ki:

```
# HD72 / EOVS
<23700> +proj=somerc +lat_0=47.14439372222222 +lon_0=19.048571777777778
+k_0=0.99993 +x_0=650000 +y_0=200000 +ellps=GRS67 +units=m +no_defs <>
```

Az első sor a dátum nevének rövidítését (Hungarian Datum 1972), utána pedig a vetületi rendszer nevét (EOV – Egységes Országos Vetület) tartalmazza. A következő sorban van definiálva maga a vetület. A sor első eleme maga az EPSG azonosító, jelen esetben 23700, a továbbiakban pedig a vetület sajátosságait leíró paraméterek következnek. Ezek magyarázatával itt ugyan nem foglalkozunk, de azt érdemes megfigyelni, hogy a vesszők elválasztó karakterekként funkcionálnak, a tizedes törteket pedig tizedes pontok tagolják, vagyis az angolszász (amerikai) jelölésrendszert használja az állomány, hasonlóan a GRASS egészéhez.

Látjuk tehát, hogy az EOVS-nek az EPSG azonosítója 23700, ezt kell megadni az új location létrehozására megnyitott ablak harmadik mezőjében. Ezt megtehetjük egyszerűen a kód beírásával, de a *Grab search* (a keresés eredményének beillesztése) gombra kattintással is.

Ha az új location-t létrehozó ablak mindhárom mezőjét kitöltöttük, kattintsunk a *Define location* gombra és elkészül az *eov* nevű location-ünk.

### Egyéb esetekben

Előfordulhat, hogy nem találjuk meg a használni kívánt EPSG kódot a felkínált állományban, vagy bármi egyéb ok miatt nem felel meg ez a fájl, vagy egyáltalán nem is kínálja fel a GRASS a fájlt. Például akkor kerülünk ilyen helyzetbe, ha olyan locationt kell létrehozunk, amely a *Krovak*-ot veszi alapul. A *Krovak* a hajdani Csehszlovákia térképezéséhez használt vetületi rendszer, és persze ezt használja ma Csehország és Szlovákia is. Ha olyan terület térinformatikai elemzését kell elkészítenünk, melynek egyik része Magyarországra, másik része pedig Szlovákiba esik, szükség lehet arra, hogy ilyen térképeket is fel tudjunk dolgozni. *Krovak*-ból azonban több is van, de csak egyet találunk az előbb bemutatott eljárással, mégpedig ezt:

```
# S-JTSK (Ferro) / Krovak
<2065> +proj=krovak +lat_0=49.5 +lon_0=42.5 +alpha=30.28813972222222
+k=0.9999 +x_0=0 +y_0=0 +ellps=bessel +pm=ferro +units=m +no_defs <>
```

Ha másakra van szükségünk (és ez valószínű), magunknak kell előkeresni azt. A feladat tehát az, hogy keressünk egy olyan állományt, mely tartalmazza a megfelelő geodéziai dátumot és a hozzá tartozó EPSG kódot! Ilyenkor az alábbi megfontolásokat tehetjük.

A legegyszerűbb megoldás, ha először abban a könyvtárban nézünk szét, amelyet a GRASS ajánlott fel amikor EPSG kód alapján akartunk location-t definiálni. Ez a könyvtár a */usr/share/proj/* volt (ez az elérési út csak Linux esetében megfelelő, OSX-et használva ugyanezt alap esetben a */Applications/GRASS-6.4/@@@*, Windows alatt pedig a *C:/GRASS/@@@* útvonalon érhetjük el.) Ha kilistázzuk a könyvtár tartalmát, az alábbi állományokat találjuk:

alaska	hawaii	ntv1_can.dat	stlrc
conus	IGNF	null	stpaul
epsg	MD	nzgd2kgrid0005.gsb	TN
esri	nad27	other.extra	WI
esri.extra	nad83	proj_def.dat	WO
FL	nad.lst	prvi	world
GL27	ntf_r93.gsb	stgeorge	

Ezek mindegyike vetületeket és geodéziai dátumokat ír le. Egyesek közülük kizárólag valamely kisebb területi egységre vonatkoznak (például a *hawaii*), míg mások általánosak, több különféle dátumot tartalmaznak (akár több százat is). Ez utóbbiakban érdemes keresni, ha olyan dátumot kell használnunk, melyet a GRASS által eredetileg felkínált állományban nem találtunk. Ezek az alábbiak:

epsg	GL27	nad83
esri	IGNF	other.extra
esri.extra	nad27	world

Ezeket érdemes tehát megnyitni és a szükséges dátumot keresni bennük. A fájlok megnyitását úgy is elvégezhetjük, ahogy azt fentebb, az „*Egyszerűbb esetben*” című részben láttuk, de most a második sorban, a GRASS által felkínált állománynevet átírjuk valamelyikre a fentiek közül. Ezután rögtön kereshetünk is bennük. Ennél is egyszerűbb bármilyen fájlkezelőben (például *Midnight Commander*) ebbe a könyvtárba lépni, sorban megnyitni az állományokat és rákeresni bennük a kérdéses dátum nevére.

Ha a dátum és az azonosítója megvan, akkor úgy járunk el, mint az első, az „egyszerűbb” esetben: a második sorban adjuk meg a fájlnak a nevét a hozzáfető elérési úttal együtt, a harmadik sorban pedig az ebben talált, nekünk megfelelő azonosítót. A *Define location* gombra kattintás után a location létrejön.

További hasonló, EPSG kódokat leíró állományokat találunk a GRASS könyvtárában, de ezeket *nem* ugyanúgy kell kezelni, mint az előbb bemutatottakat. E könyvtár helye a GRASS könyvtárából nézve: *etc/ogr\_csv/*.

- Vagyis, ha például a GRASS a */usr/lib/grass64* könyvtárban található, akkor:  
*/usr/lib/grass64/etc/ogr\_csv/*  
Ez a helyzet a legtöbb Linux disztribúció esetében.
- Apple OSX operációs rendszer alatt, GRASS 6.4 esetében az elérési út általában valahogy így néz ki:  
*/Applications/GRASS-6.4.app/Contents/MacOS/etc/ogr\_csv/*
- Ha Windows környezetben futtatjuk a GRASS-t, akkor jellemzően itt találjuk a keresett állományt:  
*C:\GRASS\etc\ogr\_csv*

Az itt levő fájlok *.csv* kiterjesztése arra utal, hogy egyszerű szöveges állományokról van szó. Közülük először a *pcs.csv* kipróbálása ajánlott, de ki tudja, lehet, hogy valakinek valamelyik másikra lesz szüksége. Ezek a fájlok megnyithatók bármelyik szövegszerkesztővel és kereshetünk is bennük, de a *Define location using EPSG projection codes* ablakban erre *nincs* lehetőségünk. Ha az ebben az ablakban a GRASS által felkínált elérési utat és állománynevet átírjuk erre (vagyis például, ha Linux alatt a */usr/share/proj/epsg* sort kicseréljük */usr/lib/grass64/etc/ogr\_csv/pcs.csv-re*), akkor EPSG kód keresése nem lesz lehetséges a *Browse* gombbal, mert csak hibaüzenetet kapunk. Ez az üzenet arról tájékoztat, hogy a példában megadott *pcs.csv* nem epsg formátumú állomány. Ez önmagában igaz is, de ettől még nagyon jól használható. Ha tehát ezekben az állományokban is EPSG kódot akarunk keresni, akkor előbb nyissuk meg a kiválasztott állományt egy egyszerű szövegszerkesztőben és ott keressük ki a szükséges dátumhoz tartozó azonosítót!

Ezután adjuk meg az operációs rendszerünknek megfelelő módon az elérési utat és a fájlnevet az új locationt létrehozó ablak (*Define location using EPSG projection codes*) második mezőjében, aztán pedig a kikeresett EPSG kódot a harmadikban. Végül csak a *Define location*-ra kell egyet kattintani és készen is vagyunk.

Érdemes tudni, hogy a GRASS a vetületi adatok felhasználását valójában a PROJ nevű programon keresztül végzi és annak adatbázisát használja fel. Ezt abból is láthatjuk, hogy az EPSG kód kiválasztásánál is a PROJ könyvtárát kínálja fel használatra (*/usr/share/proj*). Ez a program a GRASS-tól független, de a GRASS telepítésekor ez is felkerül a gépre. Hogy hová, az operációs rendszerenként változó, sőt, az is lehet, hogy ugyanazon operációs rendszer két különböző verziójában sem ugyanott található.

Látható, hogy alapesetben, ha minden rendben van, nem nehéz EPSG azonosítóalapján új location-t készíteni, de ha elsőre nem találjuk meg a szükséges azonosítót, akkor már kissé bonyolultabb lehet a munka. Ez azonban nem kell, hogy elkeserítse a felhasználót: inkább azt vegyük észre, hogy a GRASS lehetővé teszi számunkra azt, hogy tulajdonképpen az összes geodéziai dátumból és ezekre épülő térképészeti rendszerből választhassunk, ha kell.

Még egy fontos apróság: honnan tudhatjuk meg, hogy melyek mondjuk a Horvátországban használatos geodéziai dátumok és mikor melyiket kell választani? Ilyen esetben sajnos nincs más hátra, mint leülni a számítógép elé és interneten utánajárni a válasznak.

### Új location egyénileg megadott értékek alapján

Ennek a résznek az elkezdése és gyakorlása előtt érdemes lehet egy csésze kakaót/teát/kávét, mindenkinek ízlése szerint, a számítógép mellé készíteni. Ez ugyanis a legtöbb háttérismeretet követelő eljárás, igaz viszont, hogy így gyakorlatilag olyan location-t hozunk létre, amelyet csak akarunk. A helyzet az, hogy egyéni location létrehozása, egyedileg megadott adatokkal, olyan mélységű és mennyiségű vetületi ismeretet és a geodéziai dátumok közötti eligazodás olyan szintjét igényli, amellyel még a térinformatikai eljárásokat rutinosan használók nagy része sem rendelkezik. Jó hír viszont, hogy erre elég ritkán van szükség, mert a leggyakrabban a már meglevő georeferált állományokat érdemes felhasználni, amikor pedig ilyen éppen nem áll rendelkezésre, az EPSG alapbeállításai is megfelelőek szoktak lenni. Amikor viszont egyik eddigi módszer sem vezet eredményre, csak az egyedi értékadás marad.

A példa kedvéért három egyszerűen elkészíthető location létrehozásának menetét tekintjük át : egy egyszerű *x,y* koordináta rendszerre alapuló location-ét, egy *földrajzi szélesség-hosszúság* alapúét és egy *UTM* vetületűét!

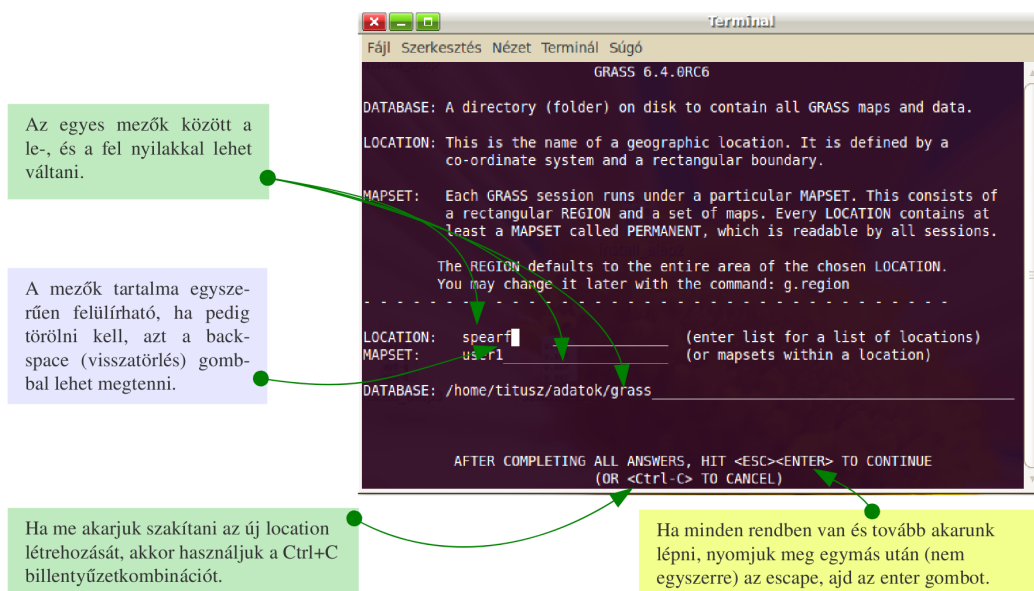
A munka megkezdéséhez a GRASS indítóablakából válasszuk a *Projection values* gombot!

A megnyíló ablak merőben eltér az eddigiektől, mert egy egyszerű szöveges terminálba kerülünk (2.7. ábra). Ennek használatáról azt kell tudnunk, hogy:

- az egyes mezők között a le- és a fel nyilakkal lehet váltani,
- a mezőkben levő értékek felülírhatók, ha pedig törölni akarjuk őket, akkor először a felülírandó érték végére kell menni a jobbra nyíllal, aztán a *visszatörölés* (backspace) gombbal kitörölni, ezután lehet új értéket megadni,
- a beírt adatok jóváhagyása és a következő ablakra váltás az escape és az enter egymás *utáni* lenyomásával lehetséges (nem egyszerre kell lenyomni, hanem egymás után!),
- az ablak elhagyása a CTRL+C megnyomásával történik.

Lássuk ezután az egyes lépéseket!

1. Az első ablakban tulajdonképpen csak a létrehozandó location nevét kell megadni, mert a PERMANET mapset neve is meg van már adva és a DATABASE mező is ki van töltve. Ha a nevet megadtuk, nyomjuk meg az escape-et, aztán pedig az enter-t!
2. Ezután a program rákérdez, hogy valóban el akarjuk-e készíteni az új locationt. Ha a válasz igen, üssünk *enter*-t, ha nem, akkor előbb nyomjuk le az *n* gombot (no) és azután üssünk *enter*-t!
3. A megjelenő feliratok arról tájékoztatnak, hogy milyen adatokat fog bekérni a GRASS a továbbiakban, egyúttal rákérdez, hogy rendelkezünk-e ezekkel. Ha igen, üssünk *enter*-t, ha nem, akkor *n* után üssünk *enter*-t. A bekérendő adatok:



2.7. ábra

Új location készítése egyedileg megadott értékekkel: a terminálablak használata

1. a létrehozandó rendszer koordinátái;
  2. a koordináta rendszer adatai, beleértve a zónát is, ha UTM;
  3. az alapértelmezett régió határai és felbontása;
  4. rövid, egysoros leírás a location-ről.
- 4.** Adjuk meg az új location koordináta rendszerét!
- A – x, y, vagyis egyszerű sík-koordináta rendszer;
- B – földrajzi szélesség és -hosszúság alapú rendszer;
- C – UTM;
- D – egyéb.

Arról is tájékoztatást kapunk, hogy az *enter* (return) megnyomása a mégsem gombbal egyenértékű. Ha egyszer megnyomjuk, kilépünk az aktuális részből, ha még egyszer megnyomjuk, visszatérünk az első lépéshez.

### 4A

- x,y alapú koordináta rendszert úgy választunk, hogy az „a” megnyomása után *entert* ütünk. A GRASS rákérdez, hogy tényleg ezt akarjuk-e, ha igen, üssünk egy *enter*-t!
- Adjunk egy egysoros leírást az új location-ről, ha kész, *enter*, majd hagyjuk jóvá.
- Adjuk meg az új location alapértelmezett régiójának határait, valamint É-D és K-Ny irányú felbontását. Ha a location-ban mondjuk egy 100×210 m méretű régészeti feltárást akarunk kezelni, akkor az északi él lehet 100, a déli él 0, a keleti él pedig 210, a nyugati él pedig 0 értékű. Ebben az esetben a felbontás legyen mindkét irányban mondjuk 10 cm, vagyis írjuk azt, hogy 0.1, mindkét felbontáshoz.
- Tovább lépés után tájékoztatást kapunk a megadott adatinkról, valamint az így létrejövő alapértelmezett régió rasztereinek számáról. Ha egyetértünk, hagyjuk jóvá!
- A location létrejött, visszatérünk az első lépéshez. Innen a *CTRL+C* billentyűkombináció, majd (ha kell) az *enter* lenyomásával léphetünk ki. A továbbiakban a szokott módon válasszunk locationt, és ha kell, hozzunk létre benne új mapsetet.

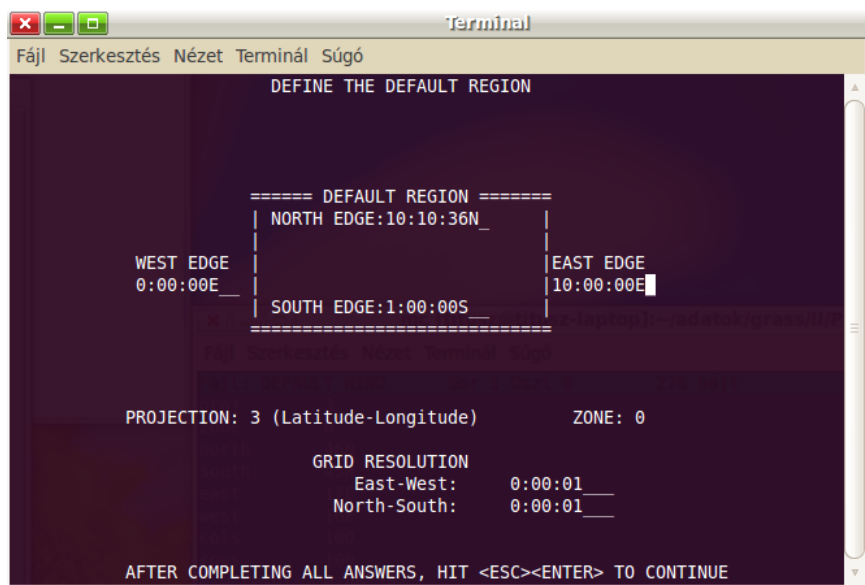
### 4B

- Szélesség-hosszúság alapú koordináta rendszert úgy választunk, hogy a „b” megnyomása után *enter*-t ütünk. A GRASS rákérdez, hogy tényleg ezt akarjuk-e, ha igen, üssünk egy *enter*-t!
- Adjunk egy egysoros leírást az új location-ről, ha kész, *enter*, és hagyjuk jóvá.
- El kell döntenünk, hogy kívánunk-e valamilyen geodéziai dátumot használni, avagy nem. Ha nem, akkor üssünk „n”-t és *enter*-t, ha igen, üssünk *enter*-t! Persze világos, hogy általában célzerű választani valamit, úgyhogy menjünk tovább ezen az úton!

- Most megadhatjuk a használni kívánt dátum nevét. Ha nem tudjuk, akkor írjuk be azt, hogy „list” és utána üssünk *enter*-t! Ennek hatására a választható dátumokat megjeleníti a GRASS. A megjelenő listában az *enter*-rel mozoghatunk lefelé soronként, a *space*-szel pedig egyszerre egy oldalt lép lefelé. Ha megvan, amit kerestünk, akkor lépjünk ki a listából a „q” megnyomásával (quit, kilépni) és írjuk be a megfelelő nevet. Általános esetre a legcélszerűbb választás alighanem a *wgs84*, tehát gyakorlásként válasszuk ezt! Ha megadtuk a nevet, az *enter*-rel lehet továbblépni.
  - Ha nem listából választunk dátumot, hanem egyedileg akarjuk definiálni, akkor „list” helyett aztírjuk be, hogy „custom”, aztán nyomjuk le az *enter*-t. A következő lépésben a használandó ellipszoid nevét kell megadni, aztán pedig az alapértelmezett régió határait és felbontását.
- Ha kiválasztottuk a használni kívánt dátumot (mondjuk a *wgs84*-et), akkor a következő lépésben a már ismert módon, listából választhatunk transzformációs paramétereket. Ha nem akarunk ilyesmit használni, akkor az *enter*-rel lehet továbblépni
- Utolsó lépésként az alapértelmezett régió határait és felbontását adjuk meg. Ezt, mivel szélesség és hosszúság alapú a location-unk, természetesen fok, perc és másodperc mértékegységben kell megadnunk, a 2.8. ábrán látható formában.
- Ha mindent megadtunk, akkor továbblépés után a GRASS megmutatja az általunk megadott értékeket és azok jóváhagyását kéri (2.9. ábra). Ha ezt megtesszük, elkészül az új location a megadott paraméterekkel.

#### 4C

- UTM alapú location készítése alapvetően ugyanúgy történik, mint a fentebb bemutatott szélesség és hosszúság alapúé, de két további kérdést is meg kell válaszolni. Az elsőre (*Enter zone*) azt kell megadni, hogy melyik zónában legyen az új location, a másik pedig arra kérdez rá, hogy a



2.8. ábra

Földrajzi hosszúság és szélesség megadása a GRASS-ban, új location létrehozásakor

```

projection: 3 (Latitude-Longitude)
zone: 0
north:      10:10:36N
south:      1S
east:       10E
west:       0

e-w res:    0:00:01
n-s res:    0:00:01

total rows: 40236
total cols: 36000
total cells: 1,448,496,000

Do you accept this region? (y/n) [y] >

```

**2.9. ábra**  
**Földrajzi hosszúság és szélesség alapú új location**  
**határainak és felbontásának jóváhagyása**

déli félgömbön van-e a létrehozandó terület (*South hemisphere (y/n) [n]*). Látható, hogy az alapértelmezett válasz a nem, vagyis *enter*-re továbblép és úgy hozza létre az alapértelmezett régiót, hogy az északi félgömbön levőnek tekinti. Egyebekben a korábbiakban bemutatottak szerint járunk el. És persze figyeljünk arra, hogy az UTM nem fokban mér, hanem méterben!

#### 4D

- Az eddigiekben ismertetettek szerint kell választani a listákból és válaszolni a kérdésekre, de itt sokkal több kérdés merülhet fel. Az egyénileg történő definiálás során ugyanis a válaszoktól függetlenül szükség lehet az ellipszoid kis- és nagytengelyének megadására is, a középmeridián kiválasztására és más hasonló értékek rögzítésére is. Ezeket itt nem tárgyaljuk, mert nagyon messzire vezetne e könyv eredeti témájától, másrészt pedig akik megértik a kérdéseket, azok vélhetően a válaszokat is meg tudják adni rájuk. És tegyük hozzá: ilyen módon location-t készíteni ugyanis csak az fog, aki az átlagosnál tájékozottabb a felmerülő vetületi problémák megoldásában.

### 2.4.1.2. Új location készítése a GRASS elindulása után

A *Config* menü *Manage projections* almenüjének *Manage projections* pontja lehetővé teszi, hogy a GRASS-ban dolgozva, bármelyik location-t is választottuk indításkor, új locationt hozzunk létre. Az elinduló program neve *g.proj*, mint az az ablak címsorából látható.

Új locationt úgy készítünk, hogy

- először az *Input* részben megadjuk a készítendő location vetületi információit tartalmazó állomány nevét, vagy a szükséges EPSG kódot,
- aztán a *Create/Edit* részben megadjuk a létrehozandó location nevét és engedélyezzük a *Create new location files...* opciót.

A *Run*-ra kattintás után az új location létrejön.

Az *Input* részben négy különböző módon adható meg, hogy milyen vetületi adatok alapján készüljön el az új location. E négy lehetőség közül természetesen elegendő egyet választani.

- **Georeferenced data file to read projection information from** – Annak a már létező, georeferált állománynak az elérési útját és nevét kell megadni, melynek vetületi adatai és geodéziai dátuma megegyezik a létrehozandó locationéval. Egyébként ez ugyanaz az opció mint „*Új location georeferált állományból*” című részben már taglaltunk, részletesebben lásd ott.
- **ASCII file containing a WKT projection description** – Annak az ASCII (vagyis: egyszerű szöveges) állomány elérési útját és nevét kell megadni, mely WKT (Well-Known Text) formátumban tartalmazza a létrehozandó location vetületi adatait.
- **PROJ.4 projecion description** – Annak az állománynak az elérési útját és nevét kell megadni, mely PROJ.4 formátumban tartalmazza a létrehozandó location vetületi adatait. Ez az állomány egyébként egyszerű szöveges állomány kell, hogy legyen.
- **EPSG projection code** – A létrehozandó location EPSG kódját kell megadni. Használatáról az „*Új location EPSG kód alapján*” című részben volt szó, részletesebben lásd ott.

Tegyük fel, hogy új, EOVS alapú location-t kell létrehoznunk, melynek neve az legyen, hogy *magyarország*! Akkor ezt egyszerűen megtehetjük úgy, hogy ebben az ablakban, az *Input* részben, az EPSG kód mezőjébe azt írjuk, hogy 23700, a *Create/Edit* részben pedig engedélyezzük a *Create new location files...* opciót és megadjuk a *magyarország* nevet. A *Run* lenyomása után a location-t a GRASS elkészíti.

### 2.4.2. Vetületi adatok tárolása a GRASS-ban

Minden location kötelezően tartalmaz egy olyan mapsetet, amelynek a neve PERMANENT (így, végig nagybetűvel). Ebben a mapsetben tárolja a GRASS az adott location egészére vonatkozó vetületi beállításokat. Erre a célra az több állomány szolgál, melyek mindegyike a PERMANENT mapset gyökérkönyvtárában van:

- **DEFAULT\_WIND** – Ez az állomány tartalmazza az adott location-ra vonatkozóan az alapértelmezett régió beállításait. Minden location csak egy DEFAULT\_WIND állományt tartalmaz, ennek helye az adott location PERMANENT mapset-je.
- **WIND** – Ez a fájl tartalmazza a mapset éppen aktuális régióbeállításának adatait (hiszen az aktuális régióknak nem kell ugyanannak lennie, mint aza alapértelmezettnek). A location minden mapset-je tartalmaz egy-egy WIND állományt, mert mindegyik mapsetnek lehet saját egyéni számítási régiója. A számítási műveleteket a GRASS csak azon a területen hajtja végre, amelyik a WIND-ben meg van adva. Alapértelmezésben megegyezik a DEFAULT\_WIND-del.

- **PROJ\_INFO** – A location vetületét, illetve a vetületi információkat tartalmazza (pl. geodéziai dátumot). Minden location csak egy PROJ\_INFO állományt tartalmaz, ennek helye az adott location PERMANET mapset-je.
- **PROJ\_UNITS** – A location vetületéhez tartozó mértékegység(ek)et adjuk meg ebben az állományban. Minden location csak egy PROJ\_UNITS állományt tartalmaz, ennek helye az adott location PERMANET mapset-je.

Amikor új location-t hoztunk létre, akkor tulajdonképpen az történt, hogy elkészült egy könyvtár a *Name of new location* sorban megadott néven, benne egy alkönyvtár PERMANET néven, abban pedig a DEFAULT\_WIND, WIND, PROJ\_INFO PROJ\_UNITS állományok. Ez egyúttal azt is jelenti, hogy minden további nélkül készíthető location úgy is, hogy egy egyszerű szövegszerkesztővel gyártjuk le az állományokat és töltjük fel őket tartalommal. Jelenti továbbá azt is, hogy bármely location bármely beállítása egyszerűen megváltoztatható úgy, hogy a megfelelő állományokat átírjuk.

Vegyük észre ennek a felépítésnek egy további hasznos következményét: ha a meglévő location-unk beállításait egy másik gépen levő GRASS-ba szeretnénk átvinni, elegendő csupán a megfelelő állományokat felülírni.

### 2.4.2.1. A PROJ\_INFO tartalma és értelmezése

Példán keresztül könnyeb a megértés, ezért nézzük meg a *spearfish60* location PROJ\_INFO állományát:

```
name: UTM
datum: nad27
nadgrids: conus
proj: utm
ellps: clark66
a: 6378206.4000000004
es: 0.0067686580
f: 294.9786982000
zone: 13
```

- **name** – a vetületi rendszer neve, UTM=Universal Transverse Mercator)
- **datum** – a használt geodéziai dátum neve (nad27 =North American Datum 1927)
- **ndgrids** – a dátumhoz tartozó opció, ha van egyáltalán (itt, a nad27-hez kell)
- **proj** – vetítési eljárás megnevezése
- **ellps** – a használt forgási ellipszoid neve
- **a** – az ellipszoid nagytengelyének mérete méterben
- **es** – az ellipszoid lapultsága
- **f** –
- **zone** – a használt zóna, ha a használt dátum, vagy vetületi rendszer ezt igényli

Nézzük most meg az fentebb leírt módon, EPSG azonosító alapján létrehozott EOVS alapú location PROJ\_INFO állományát:

```
name: Swiss. Obl. Mercator
proj: somerc
```

```

ellps: grs67
lat_0: 47.14439372222222
lon_0: 19.04857177777778
k_0: 0.99993
x_0: 650000
y_0: 200000
no_defs: defined
    
```

Ez a következőképpen értelmezhető (2.10. ábra): Az EOVT a GRASS úgy közelíti, hogy a *Swiss Obligated Mercator* vetületet használja (*name, proj*), az alapfelület méreteit pedig a Nemzetközi Geodéziai Unió által 1967-ben elfogadott forgási ellipszoidról veszi (*ellps*). Ezen definiál egy alappontot, mely az északi szélesség 47.14439372222222 és a keleti hosszúság 19.04857177777778 fokán található (*lat\_0, lon\_0*). A koordináta rendszer kezdőpontja e ponttól 650000 m-rel (650 km) nyugatra és 200000 m-rel (200 km) délre van (*x\_0, y\_0*).

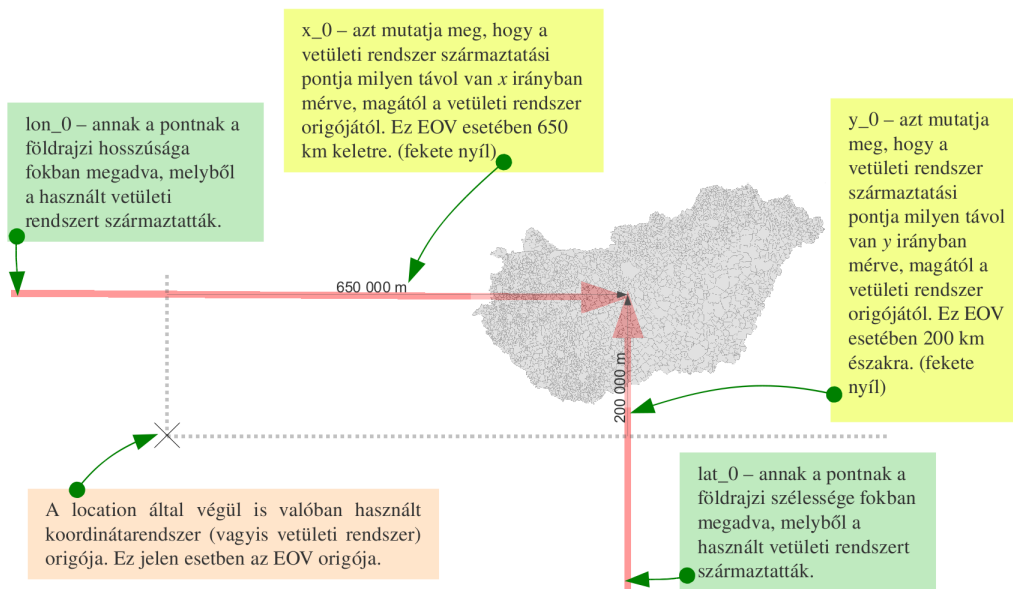
Ez a közelítés a mindennapi munka számára tökéletesen elfogadható, Magyarország területén belül csupán centiméteres nagyságrendű hibát okozhat. Egyébként általában más térinformatikai rendszerek (pl. ArcGIS) is ezt használják.

### 2.4.2.2. A PROJ\_UNITS tartalma és értelmezése

A spearfish60 location PROJ\_UNITS állománya:

```

unit: metre
units: metres
meters: 1
    
```



2.10. ábra  
A PROJ\_INFO-ban foglaltak értelmezése az EOVT példáján

Azt tartalmazza tehát, hogy a PROJ\_INFO-ban megadott vetületi rendszert használó location-ban a használt mértékegység neve (*unit*) méter (*metre*), többszáma (*units*) *metres* (méterek), és ez a mértékegység egyenlő 1 méterrel (*metres*), vagyis a használandó konverziós együttható=1.

Tekintve, hogy az EOVS is méterben mér, ugyanezt tartalmazza az *eov* nevű location-unk PROJ\_UNITS állománya is.

### 2.4.2.3. A DEFAULT\_WIND tartalma és értelmezése

A megértés segítésére vizsgáljuk meg a spearfish60 location PERMANENT mapsetjének DEFAULT\_WIND állományát!

```
Proj: 1
zone: 13
north: 4928010
south: 4913700
east: 609000
west: 589980
cols: 634
rows: 477
e-w resol: 30
n-s resol: 30
top: 1
bottom: 0
cols3: 634
rows3: 477
depths: 1
e-w resol3: 30
n-s resol3: 30
t-b resol: 1.
```

- **proj** – A location vetületét adja meg. Ez jelen esetben 1, vagyis UTM.
- **zone** – Ha a vetület zónákra osztott szerkezetű, akkor itt kell megadni a zóna számát (most tehát a 13-as zóna)
- **north** – A location északi szélének észak-déli koordinátája az adott vetületi rendszer szabályai szerint megadva (itt tehát UTM vetületi rendszerben É=4928010 méter)
- **south** – A location déli szélének észak-déli koordinátája az adott vetületi rendszer szabályai szerint megadva (itt tehát UTM vetületi rendszerben É=4913700 méter)
- **east** – A location keleti szélének kelet-nyugati koordinátája az adott vetületi rendszer szabályai szerint megadva (itt tehát UTM vetületi rendszerben K=609000 méter)
- **west** – A location nyugati szélének kelet-nyugati koordinátája az adott vetületi rendszer szabályai szerint megadva (itt tehát UTM vetületi rendszerben K=589980 méter)

- **cols** – Oszlopok száma a locationban (ez a location tehát 634 oszlopra bomlik). Ennek csak raszteres térképek esetében van jelentősége.
- **rows** – Sorok száma a locationban, ennek is csak raszteres térképek esetében van jelentősége.
- **e-w resol** – Kelet-nyugati felbontás a location mértékegységében.
- **n-s resol** – Észak-déli felbontás a location mértékegységében.
- **top** – A felső szél értéke, csak 3D állományok esetében van értelme (alapértelmezésben 1). Részletesen lásd a 3D objektumok részben!
- **bottom** – Az alsó szél értéke, csak 3D állományok esetében van értelme (alapértelmezésben 0). Részletesen lásd a 3D objektumok részben!
- **cols3** – Oszlopok száma, ha a valódi 3D állományokat kezelünk a régióban (alapértelmezésben egyezik a cols értékével)
- **rows3** – sorok száma, ha a valódi 3D állományokat kezelünk a régióban (alapértelmezésben egyezik a rows értékével)
- **depths** – 3D állományok mélysége, vagyis, hogy hány lépésközzel érhetünk el az aljától a tetejéig (alapértelmezésben 1) Figyelem: ez nem a rétegek (layer) számát adja meg! Részletesen lásd a 3D objektumok részben!
- **e-w resol** – Kelet-nyugati felbontás értéke, 3D állományok esetében (alapértelmezésben egyezik az e-w resol értékével)
- **n-s resol** – észak-déli felbontás értéke, 3D állományok esetében (alapértelmezésben egyezik az n-s resol értékével)
- **t-b resol** – A fent-lent irányú felbontás értéke, csak 3D állományok esetében van értelme (alapértelmezésben 1) Részletesen lásd a 3D objektumok részben!

Az előbb leírtakat szem előtt tartva, egy EOV location-beli DEFAULT\_WIND tartalmát is érdemes megvizsgálni! Erre tökéletesen megfelel a korábban, EPSG kód alapján, általunk létrehozott, *eov* nevű locationon.

```
proj: 99
zone: 0
north: 1
south: 0
east: 1
west: 0
cols: 1
rows: 1
e-w resol: 1
n-s resol: 1
top: 1
bottom: 0
cols3: 1
rows3: 1
```

```
depths: 1
e-w resol3: 1
n-s resol3: 1
t-b resol: 1
```

Hogyan értelmezhetők ezek az adatok? Emlékezzünk csak: az egyedi értékekkel létrehozandó location készítése során választanunk kellett, hogy egyszerű x,y alapú, szélesség-hosszúság alapú, vagy UTM alapú lesz maga a location. Ha egyik sem felelt meg, akkor választhattuk az „egyéb” (*other*) pontot. Amikor az EPSG azonosító megadásával készítettük el az új location-t, akkor olyan location-t alkottunk, mely a fentiek egyikének sem felel meg. Ez látható is a PROJ\_INFO megnyitásakor (lásd fentebb), hiszen itt az van megadva, hogy

```
name: Swiss. Obl. Mercator
proj: somerc
```

Vagyis az alkalmazott vetület az „egyéb” kategóriába tartozik, hiszen nem x,y, nem szélesség-hosszúság és nem UTM, hanem *somerc*. Ezért van az EOV alapú location-unk DEFAULT\_WIND-jében a *proj* értékének 99 megadva: ez jelenti azt, hogy „egyéb vetület”. Hogy ez az egyéb vetület milyen, az le van írva a PROJ\_INFO-ban, innen tudja maga a GRASS is. Ez a rendszer nem tartalmaz zónákat, ezért a *zone* értéke 0 (nulla).

A további rész definiálja az alapértelmezett régió kiterjedését:

```
north: 1
south: 0
east: 1
west: 0
```

Ez jelen állapotában azt jelenti, hogy az alapértelmezett régió déli éle az adott vetületi rendszer 0 értékénél van, északi éle pedig az 1-nél. Mivel a PROJ\_UNITS-ból kiderül, hogy a megadott értékek méterben értendők, ez azt jelenti, hogy az alapértelmezett régió észak-déli kiterjedése mindössze egy méter! Ugyanígy kell értelmeznünk a további két sort is, vagyis az alapértelmezett régió kelet-nyugati kiterjedése is csak egy méter.

A következő két sorból az derül ki, hogy az alapértelmezett régiót mindössze egyetlen oszlop és egyetlen oszlop alkotja, vagyis mindössze egy darab raszter:

```
cols: 1
rows: 1
```

Ennek megfelelően az észak-déli és a kelet-nyugati felbontás is egy méter (hiszen az alapértelmezett régió egy méterszer egy méteres és ebben egyetlen sorban és oszlopban csak egy raszter van):

```
e-w resol: 1
n-s resol: 1
```

Mivel az alapértelmezett régióknk 2 dimenziós, ezért a régió teteje 1, alja pedig 0 értékű:

```
top: 1
bottom: 0
```

Tekintve, hogy az alapértelmezett régióknak 2 dimenziós, ezért a 3D értékek megegyeznek a megfelelő 2D értékekkel:

```
cols3: 1
rows3: 1
depths: 1
e-w resol3: 1
n-s resol3: 1
t-b resol: 1
```

Ezek az értékek így azt eredményezik, hogy a frissen elkészített EOVS alapú (és *eov* nevű) locationunk a gyakorlatban használhatatlan. A probléma megoldása a következő részben olvasható.

### 2.4.2.4. A DEFAULT\_WIND megváltoztatása

Az előző példából látható volt, hogy az EPSG azonosító alapján, újonnan létrehozott, EOVS alapú, *eov* nevű location alapértelmezett régiója a gyakorlatban – még – használhatatlan. Összesen egy darab rasztert tartalmaz, mely 1×1 m méretű és az is a koordinátarendszer sarkában, az origónál helyezkedik el. Ha valóban használhatóvá kívánjuk tenni, módosítani kell a DEFAULT\_WIND tartalmát. Erre a munkára a legegyszerűbb szövegszerkesztő is megfelel, így alapesetben bármelyik fájlkezelő szövegszerkesztője is tökéletes választásnak tekinthető.

Első lépésként adjuk meg az alapértelmezett régió határait. Magyarország esetében, EOVS-t használva, a szélső értékek *közelítően* az alábbiak:

```
legészakibb pont É-D koordinátája: 363 868,
legdélibb pont É-D koordinátája: 43822,
legkeletibb pont K-Ny koordinátája: 937288,
Legnyugatibb pont K-Ny koordinátája: 426 507.
```

Ezek alapján célszerű úgy eljárni, hogy a fenti értékeket északon és keleten felfelé, délen és nyugaton pedig lefelé kerekítjük a következő ezres értékig, mert így biztos, hogy az ország teljes területe benne lesz az alapértelmezett régióban. Vagyis:

```
észak: 364000
dél: 43000
kelet: 938000
nyugat: 425000
```

A következő lépés a használandó felbontás meghatározása. Észak-déli és kelet-nyugati irányban is olyan felbontást kell megadni, amelyiknek a régió észak-déli, illetve kelet-nyugati kiterjedése, egész számú többszöröse. Mindazonáltal az észak-déli és a kelet-nyugati felbontás lehet különböző értékű. Jelen esetben az alapértelmezett régió észak-dél irányú kiterjedése  $364000-43000=321000$  m, a kelet-nyugati pedig  $938000-425000=513000$  m. Vagyis, Magyarország területét egy  $321\text{ km} \times 513\text{ km}$  oldalhosszúságú, EOVS vetületű felszínen helyeztük el. Ha mondjuk azt szeretnénk, hogy a felbontás  $1\text{ km} \times 1\text{ km}$  legyen, akkor látható, hogy ez problémamentesen megvalósítható, mert az alapértelmezett régió oldalhosszúságai egész számú többszöröse a felbontás által meghatározott raszterméretnek.

Most már csak az van hátra, hogy az aktuális számítási régió is ugyanezeket az értékeket vegye át. Ehhez csak annyit kell tenni, hogy a DEFAULT\_WIND állományt lemásoljuk WIND néven. Mivel a másolás ugyanabba a könyvtárba történik, ezért felül kell írni a meglevő WIND fájlt. Ezek után a két állomány tartalma ugyanaz, és minden további nélkül nekiállhatunk a location használatának.

Az eddigi okfejtés eredményét tartalmazza az alábbi táblázat.

DEFAULT_WIND változói	Eredeti érték	Újronnan megadott érték
proj:	99	99
zone:	0	0
north:	1	364000
south:	0	43000
east:	1	938000
west:	0	425000
cols:	1	513
rows:	1	321
e-w resol:	1	1000
n-s resol:	1	1000
top:	1	1
bottom:	0	0
cols3:	1	513
rows3:	1	321
depths:	1	1
e-w resol3:	1	1000
N-s resol3:	1	1000
t-b resol:	1	1

Ha tehát úgy módosítjuk az *eov* nevű (és EOV vetültű) location DEFAULT\_WIND-jét és WIND-jét, ahogyan az a fenti táblázatban látható, akkor az alapértelmezett régiónkba teljes terjedelmében befér Magyarország területe és e területet 1×1 km-es felbontással kezelhetjük.

A bemutatott eljárás természetesen bármilyen más területre és vetületre is alkalmazható.

### 2.4.2.5. A WIND tartalma és értelmezése

A WIND tartalmazza az aktuális (számítási) régió paramétereit. A WIND állomány tartalma alapértelmezésben megegyezik a DEFAULT\_WIND-ével. Másképpen fogalmazva ez azt jelenti, hogy az aktuális számítási régió megegyezik az alapértelmezett régióval. Ha egyéni számítási régiót definiálunk (például a Map Display ablakban, a Zoom to... menüvel), akkor ennek a határait, felbontását és minden egyéb

paraméterét a WIND-ben tárolja a GRASS. A WIND szerkezete és értelmezése mindenben megegyezik a DEFAULT\_WIND-ével.

Fontos sajátossága a WIND-nek, hogy minden mapsetben van belőle egy, ellentétben a DEFAULT\_WIND-del, mely csak a PERMANENT mapsetben helyezkedhet el. Ez érthető is, hiszen egyazon location mindegyik mapset-jének ugyanaz kell, hogy legyen az alapértelmezett régiója, de az aktuális számítási régiók már eltérhetnek egymástól.

### 2.4.2.6. Névvvel ellátott régiók tárolása a GRASS-ban

Bármely régió elnevezhető és e néven elmenthető a GRASS-ban. Az így elmentett régiókat bármikor számítási régióvá tehetjük, mert a nevükkel könnyen hivatkozhatunk rájuk. A régió mentésekor egy, az adott névvvel ellátott állomány jön létre az adott mapset *windows* (ablakok) könyvtárában. Természetesen tetszőleges számú régiót menthetünk el, hiszen mindegyik csak egy fájl, ilyenből pedig gyakorlatilag akármennyi lehet egy könyvtárban. Ezeknek a fájloknak a felépítése mindenben egyezik a DEFAULT\_WIND-ével és a WIND-ével, értelmezésük is ugyanaz. Amikor egy névvvel ellátott régiót választunk ki számítási régiónak, mondjuk a *Map Display* ablak *Zoom to...* menüjének *Zoom display to saved region* pontjával, akkor tulajdonképpen csak annyi történik, hogy a GRASS az adott névv állomány tartalmával felülírja a WIND állomány tartalmát. Ezért fog az aktuális számítási régió a továbbiakban megegyezni a kiválasztott régióval. Ha valamelyik, névvvel ellátott régiót módosítani akarjuk, akkor azt ugyanúgy, egy egyszerű szövegszerkesztővel is megtehetjük, mint az a *DEFAULT\_WIND megváltoztatása* című részben olvasható.

## 2.5. Digitalizálás a GRASS-ban

A GRASS fejlett digitalizáló eszköztárral rendelkezik, melynek kezelése ugyan kissé eltér az általában megszokottól, mindazonáltal igen jól használható, ezért mindenképpen érdemes megismerni. A digitalizálás történhet raszteres és vektorosan, mindkettő támogatott a GRASS-ban.

Mivel egyszerűbb a használata, először ismerkedjünk meg a raszteres digitalizálással!

### 2.5.1. Raszteres digitalizálás a GRASS-ban

Mielőtt nekilátnánk a digitalizálásnak, mindenképpen győződjünk meg róla, hogy a régióbeállítások megfelelőek-e! A digitalizálás során létrehozott raszteres térkép felbontás ugyanis olyan lesz, mint az éppen aktuális régió felbontása. Vagyis, ha mondjuk a régióra érvényes aktuális beállításban 100×100 m felbontás van megadva, akkor a digitalizálás eredménye olyan térkép lesz, amelynek a felbontása 100×100 m, függetlenül attól, hogy egyébként a régió alapértelmezett felbontása – és így valószínűleg a benne levő térképeké is – esetleg sokkal jobb, mondjuk 30×30 m.

Ugyanilyen fontos tudnunk azt is, hogy a digitalizálás a számítási területen történik. Vagyis, az éppen aktuális régió területén. Ha tehát mondjuk az aktuális régió (vagyis a számítási terület) a térképünknek csak egy kis részét foglalja magában, akkor a digitalizálás is csak ezen a területen valóul meg, függetlenül attól, hogy előtte esetleg az egész alapértelmezett régiót megjelenítettük a *Monitor* ablakban. Mindenesetre száz szónak is egy a vége: a digitalizálás megkezdése előtt ellenőrizzük a régióbeállításokat és ha szükséges úgy módosítsuk azokat, hogy az aktuális régió (számítási terület) kiterjedése és felbontása megfeleljen a kívánalmainknak!

Ezek után állhatunk neki a tulajdonképpeni digitalizálásnak (2.11. ábra).

Először válasszuk a *Raster* menü *Develop map* almenüjének *Digitize raster* pontját! Ennek hatására megjelenik egy fehér háttérű ablak, neve (alapértelmezésben) *Monitor: x0*, valamint egy terminál, melynek neve most *r.digit*. Ez utóbbiban adjuk meg a szükséges paramétereket, a monitor ablakban pedig a térképet rajzoljuk. A terminál használata megegyezik az *Új location egyénileg megadott értékek alapján* című részben bemutatottal.

- Az első teendőnk, hogy az *r.digit* ablakban megadjuk a létrehozni kívánt raszteres térkép nevét. Segítség, hogy lehetőség van a névadás előtt az adott mapsetben már meglévő raszteres térképek neveinek listázására is. Ehhez annyit kell tenni, hogy nem a létrehozni kívánt állomány nevét adjuk meg, hanem azt írjuk az ablakba, hogy *list*, aztán *entert* ütünk.
- Megjelenik a kérdéses mapsetben levő raszteres térképek neve és újra lehetőségünk van a létrehozni kívánt állomány nevének megadására. Tegyük ezt meg és üssünk *entert*! Ha olyan nevet találunk, amelyik már foglalt, akkor hibaizenetet kapunk és ismét megpróbálkozhatunk a névadással.
- Ha létrejött az állomány, rögtön egy újabb kérdést kell megválaszolni: kívánunk-e valamilyen térképet a háttérben megjeleníteni (*Display commands to be used for canvas backdrop (separated by ';')*) vagyis: a háttér meghatározó megjelenítési parancsok, ;-vel elválasztva egymástól). Ha igen, akkor meg kell adni a megjelenítési parancsot és annak értékeként a megjeleníteni kívánt térképet. Adjuk meg gyakorlásként azt, hogy

*d.rast map=elevation.dem@PERMANENT*

Ez azt jelenti, hogy jelenítsük meg a PERMANENT mapsetben levő *elevation.dem* nevű állományt (A GRASS parancsairól részletesebben a GRASS nagyon haladóknak részben a *Parancsok használata a GRASS-ban* című fejezetben olvashatunk!). Ezután megerősítést kér a program, hogy valóban a megadott parancsot kívánjuk-e használni. Ha igen, üssünk *entert*, ha nem, írjuk be, hogy „n” és azután üssünk *entert*! Ha semmit sem kívánunk a háttérbe, akkor erre a kérdésre ne válaszoljunk, egyszerűen menjünk tovább az *enter* lenyomásával.

- Ezek után megkezdhetjük a digitalizálást. A monitoron megjelenik a betöltött térkép, vagy – ha semmit sem kértünk a háttérbe – üresen marad. A terminálban a választható eljárások listája tűnik fel: A, C, L, X, Q, vagyis *area* (terület), *circle* (kör), *line* (vonal), *exit* (kilépés és mentés), *quit* (kilépés mentés nélkül). Ha vonalat kívánunk rajzolni, akkor írjunk egy „a” betűt (mindegy, hogy kicsi vagy nagy) és üssünk *entert*! A terminálban megjelenik, hogy melyik egérgomb mire használható és nekiláthatunk vonalat rajzolni.
  - *Bal gomb*: hol vagyok? A kurzor koordinátáját adja vissza.
  - *Középső gomb*: pont megjelölése. Ezzel adjuk meg az egyenes egy pontját.
  - *Jobb gomb*: a vonalrajzolás befejezése.

Első lépésként meg kell adni az új raszteres állomány nevét. Ha már meglevő térképet kívánunk javítani, akkor listából is kiválasztható a neve.

Ha térképről digitalizálunk, akkor szükség lesz háttérképére is. Ezt adjuk meg a második lépésben, parancssoros formában.

A harmadik lépésben meg kell erősíteni a választásunkat.

Ezután kell kiválasztani, hogy milyen objektumok szereténk létrehozni. Ha például vonalat, akkor írjuk be, hogy „l” és nyomjunk *enter*t

Megkezdhető a digitalizálás. A terminálban mindig láthatjuk, hogy melyik egérgomb mire szolgál.

A digitalizálás során a megjelenítőablakban dolgozunk, a terminálban pedig az egyes, digitalizált pontokat és koordinátájukat láthatjuk. Ha a jobb egérgombbal befejeztük a vonal rajzolását, akkor

- először adnunk kell neki szakadatot (*category*), majd az *enter* lenyomása után
- címkét is adhatunk, újabb *enter* után pedig
- jóváhagyhatjuk az adatokat.
- *Enter* után visszatérünk a digitalizálandó objektumok kiválasztásához.

2.11. ábra  
Raszteres digitalizálás menete a GRASS-ban

Egy vonal természetesen több töréspontot is tartalmazhat, nem kell feltétlenül két pont közötti egyenesnek lennie.

A vonal befejezésekor (vagyis a jobb egérgomb megnyomását követően) a terminálban meg kell adnunk a vonal azonosítójaként egy számot. Ez lehet mindegyik vonal esetében különböző, de természetesen több vonalnak is lehet ugyanaz az azonosítója (mert mondjuk mindegyik vonal egy-egy másodrendű főutat jelent, ezért mindegyiknek az azonosítója 2). Egy vonalnak azonban csak egy numerikus azonosítója lehet. Ha megadjuk a kívánt számot, üssünk *entert*!

A numerikus azonosító után szöveges azonosítót (*label*, címke) is rendelhetünk a vonalunkhoz. Ennek megadása nem kötelező. A címke akármilyen karaktert tartalmazhat, így az összes magyar ékezetes betűt is. Az enter lenyomása után a program megerősítést kér, rákérdez, hogy egyetértünk-e a megadott numerikus és szöveges azonosítóval. Ha hibásnak látjuk valamelyiket, az „n”-t választva újra megadhatjuk őket. Ha egyetértünk, üssünk *entert*!

Visszatérünk a választható eljárások listájához, újabb elemet digitalizálhatunk, vagy kiléphetünk az alkalmazásból. Kilépéskor figyeljünk arra, hogy az X és a Q nagyon mást jelent!

Ha területet digitalizálunk, ugyanúgy kell eljárni, mint a vonal digitalizálásakor, ügyelve természetesen arra, hogy a határvonal zárt legyen. Ha a határvonal nem zárt, akkor a digitalizáló zárja, összeköti egy egyenessel az első és az utolsó pontot.

Kör digitalizálásakor az első kattintás a középső gombbal megadja a kör középpontjának a helyét, aztán az egér mozgatásával állítható be a kör sugara. A második kattintás a középső gombbal jóváhagyja az éppen aktuális sugarat és megjelenik a kör. A továbbiakban ugyanaz az eljárás, mint a vonal és a terület megadásakor.

### 2.5.2. Vektoros digitalizálás

A vektoros digitalizálás lényegesen finomabb felbontású lehet, mint raszteres társa, ráadásul sokkal összetettebb formákat állíthatunk elő így, amelyek mindegyikéhez gyakorlatilag tetszőleges mennyiségű adatot rendelhetünk. Cserében persze sokkal bonyolultabb is. Vektoros digitalizálással lehetőségünk van teljesen új térkép készítésére, akár új adatbázissal is, új térképet készíthetünk régi adatbázissal és akár már meglévő vektoros térképünket is átszerkeszthetjük vagy módosíthatjuk az egyes objektumaihoz rendelt szakadatokat.

A legbonyolultabb persze a teljesen új térkép készítése, különösen, ha új adattáblát is kell hozzá csinálni. De azért ez sem valami nagyon nehéz feladat.

A vektoros digitalizálás menete a legegyszerűbb esetben az alábbi:

1. A digitalizáló indítása, ezzel új – egyelőre üres – vektoros térkép létrehozása.
2. A tulajdonképpeni digitalizálás megkezdése előtt a digitalizálóban adattábla létrehozása, mely azonnal automatikusan csatolódik is az adott térképhez (lásd: *A digitalizáló beállításai* című részben).
3. A digitalizálás elvégzése.
4. Ha kell, a szükséges utóműveleteket elvégzése (pl. topológia építése, hibák automatikus javítása).
5. Az eredmény ellenőrzése.

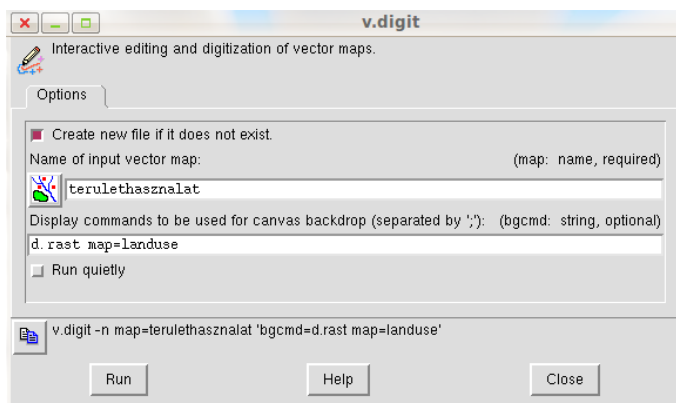
Ha olyan adattáblát kell létrehoznunk, amelyet a digitalizálóban nem lehet elkészíteni, akkor a munka menete így alakul:

1. Létre kell hozni egy új üres adatbázist, amelyben a térkép elemeihez tartozó szakadatokat lehet tárolni. (lásd: adatbázisok kezelése a GRASS-ban)
2. Új üres vektoros térképet kell készíteni.
3. Az előbb megalkotott két elemet össze kell kapcsolni egymással. (lásd: vektoros térképek kezelése részt, vagy adatbázisok kezelése a GRASS-ban részt)
4. El kell végezni a digitalizálást.
5. Ha kell, végrehajtani a szükséges utóműveleteket (pl. topológia építése, hibák automatikus javítása)
6. Az eredmény ellenőrzése.

Új üres vektoros térképet a *Vector* menü *Develop map* almenüjének *Digitize* pontjával készíthetünk (v.digit).

- A felbukkanó ablakban két üres mezőt és egy opciót találunk. Először is adjuk meg a létrehozandó térkép nevét a *Name of input vector map* mezőben és jelöljük be a *Create new file if it does not exist* opciót. Ezzel engedélyezzük, hogy elkészüljön a megadott nevű térkép, ha még nem lenne ilyen az aktuális mapsetben.
- A második mezőben azt adjuk meg, hogy mi legyen a digitalizálás során a háttérben. Hagyhatjuk üresen is, ekkor a digitalizálás háttere alapértelmezésben fehér lesz. Ha adunk meg háttérrel, akkor az lehet raszteres és vektoros térkép is.

A mező felirata tájékoztat arról, hogy miként kell eljárni: *Display commands to be used for canvas backdrop (separated by ';')*, vagyis, adjuk meg azt a parancsot, amellyel meghatározzuk, hogy mely térkép kerüljön a háttérbe (2.12. ábra), ha több parancsot is adunk, akkor pedig azokat pontosvesszővel kell elválasztani (a parancsok használatát bővebben lásd a *Parancsok használata a GRASS-ban* című részben!).



**2.12. ábra**  
A GRASS vektoros digitalizálójának indítása

Két egyszerű példa segít megérteni a parancsok megadását.

```
d.vect map=roads@PERMANENT
```

A fenti parancs hatására a háttérben a PERMANENT mapsetben levő *roads* nevű vektoros térkép jelenik meg. Természetesen az egyenlőségjel után bármely más vektoros térkép neve is szerepelhetne.

Ha raszteres térképet akarunk a háttérbe tenni, akkor írjuk azt, hogy

```
d.rast map=landuse@PERMANENT
```

Így a háttérbe a PERMANENT mapset *landuse* nevű raszteres térképe kerül, de az egyenlőségjel után itt is megadhattunk volna más térképet.

- Kattintsunk a Run gombra és ha minden rendben van, a digitalizáló tovább lép.
- Két új ablak jelenik meg: az egyik a digitalizáló eszközeiket tartalmazza (*v.digit.toolbox*), a másikban pedig a digitalizálást lehet elvégezni (*v.digit*).
- Készítsük el a digitalizálandó vektoros térkép adattábláját! Ehhez kattintsunk az eszköztárban (*v.digit.toolbox*) az *Open settings* (beállítások megnyitása) gombra és a megnyíló ablakból válasszuk ki a *Table* (táblázat) részt. Tegyük meg itt a szükséges beállításokat úgy, ahogyan az Hiba: A hivatkozás forrása nem található. részben, a Hiba: A hivatkozás forrása nem található. oldalon olvasható.
- Végezzük el a digitalizálást a *Digitalizálás menete a GRASS-ban* című részben (46. oldal) foglaltak szerint.

Végül, csak a teljesség kedvéért nézzük meg, hogy milyen a digitalizálás indítása parancssorból:

```
v.digit -n map=a_terkep {bgcmd=d.rast -o elevation.dem@PERMANENT}
```

**v.digit** a parancs neve

**-n** új térkép létrehozása, ha az adott mapsetben még nincs ilyen néven

**map=** a digitalizálandó térkép neve

**{}**= opciók

**bgcmd=** background command, vagyis „háttérkép parancs”

Ha mindent rendben megadtunk, akkor kezdődhet a digitalizálás.

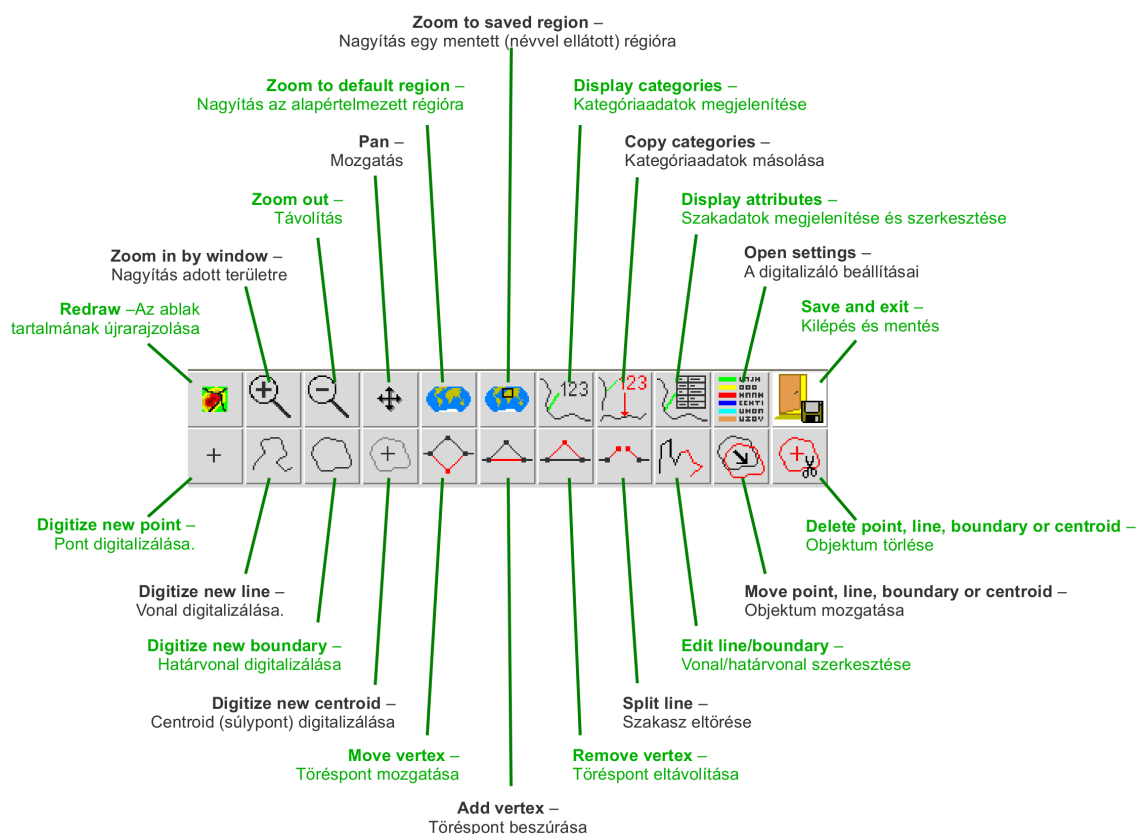
Az egyik ablak – mint az fentebb már említett volt – a digitalizáló eszközök ablaka, a másik pedig a digitalizáló ablak. Ha választottunk háttértérkép-et, akkor az ebben fog megjelenni és természetesen ebben jelennek majd meg az általunk rajzolt elemek is.

Tekintsük át először a digitalizáló eszközöket (2.13.ábra)!

Az eszközök egy jó része nyilvánvalóan érthető, semmiféle különösebb magyarázatot nem igényel a használatuk, más részük viszont esetleg kevésbé ismert. Figyeljük meg, hogy az eszköztár valamely elemét kiválasztva a gombsor alatt olvasható, hogy az aktuális eszköz esetében a bal-, a jobb- és a középső egérgombbal való kattintás milyen műveletet eredményez. Például a nagyítót választva a bal egérgombbal adjuk meg a nagyítandó tartomány egyik sarkát, a középső gombbal a szemben levő sarkot, jobb egérgombbal pedig kilépünk a nagyításból.

- **Redraw** – Az *ablak tartalmának újrarajzolása*. A gombra kattintva az ablak tartalma frissül, újrarajzolja azt a GRASS.

- **Zoom in by window** – Nagyítás adott területre. A térképek megjelenítésekor is használt nagyító eszközzel megegyező a használata.
- **Zoom out** – *Távolítás*. A térképek megjelenítésekor is használt távolító eszközzel megegyező a használata.
- **Pan** – *Mozgatás*. Használata kissé eltér a szokásostól. A gomb kiválasztása után a térképnek arra a részére kell kattintanunk, amelyiket a megjelenítőablak közepén szeretnénk látni. Vagyis nem tudjuk az egérrel húzni a térképet (és a háttértérképet), hanem azt adjuk meg, hogy melyik pontja legyen középen. Célszerűen az éppen megjelenített terület szélére érdemes kattintani vele. Egy- más után többször is használható, így az egész térkép bejárható a segítségével.
- **Zoom to default region** – *Nagyítás az alapértelmezett régióra*. Használata megegyezik a térképek megjelenítésénél használt eszközével. Ha kiválasztjuk, az alapértelmezett régió (vagyis a PERMENET mapsetben levő DEFAULT\_WIND állományban definiált terület) jelenik meg az ablakban.



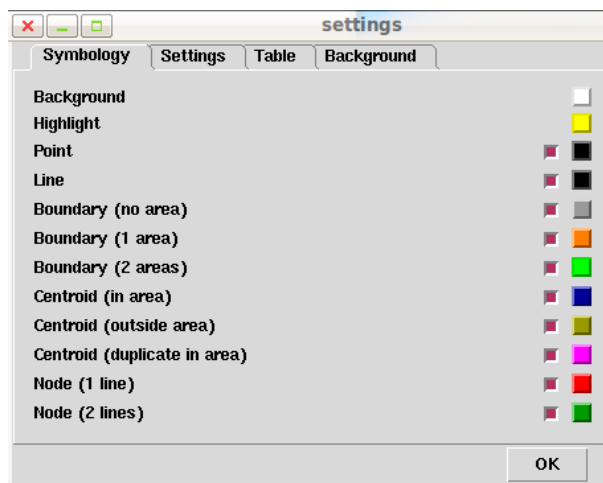
2.13. ábra  
A GRASS vektoros digitalizálójának eszköztára

- **Zoom to saved region** – *Nagyítás egy mentett (névvel ellátott) régióra.* Használata megegyezik a térképek megjelenítésénél használt eszközével. Ha kiválasztjuk, egy listából választhatjuk ki, hogy mely, korábban definiált és névvel ellátott régióbeállítás lépjen érvénybe.
- **Display categories** – *Kategóriaadatok megjelenítése.* Megmutatja, hogy a kiválasztott elem melyik rétegen van és mi az azonosítója (kategória). Ha szeretnénk, itt egyúttal meg is lehet változtatni a kiválasztott elem ezen tulajdonságait.
- **Copy categories** – *Kategóriaadatok másolása.* Valamely objektum kategória (azonosító) adatát másolja át egy másik objektumra, így annak is ugyanaz lesz a megfelelő adata. A gomb megnyomása után először a forrásobjektumot (amiről másolni akarunk) kell kijelölni, majd a célobjektumot (aminek meg akarjuk változtatni a megfelelő adatát). Természetesen az egérgombok funkciója most is olvasható a gombsor alatt.
- **Display attributes** – *Szakadatok megjelenítése és szerkesztése.* A kiválasztott objektumhoz rendelt szakadatokat tekinthetjük meg, illetve szerkeszthetjük is azokat. Persze, csak akkor, ha a digitalizálás alatt álló vektoros állományhoz van csatolva adattábla.
- **Open settings** – A digitalizáló beállításai.
- **Save and exit** – *Kilépés és mentés.* A gombot megnyomva a digitalizáló befejezi a futását, kilép és menti a változásokat
- **Digitize new point** – *Pont digitalizálása.* Egy pontot digitalizál.
- **Digitize new line** – *Vonal digitalizálása.* Általános vonalat rajzolhatunk vele, tetszőleges számú törésponttal.
- **Digitize new boundary** – *Határvonal digitalizálása.* Valamely poligon határvonalát adhatjuk meg vele, tetszőleges számú törésponttal.
- **Digitize new centroid** – *Centroid (súlypont) digitalizálása.* Súlypontot adhatunk meg.
- **Move vertex** – *Töréspont mozgatása.* Vonala és határvonal belső töréspontját (vertex) mozgathatjuk.
- **Add vertex** – *Töréspont beszúrása.* Vonalba és határvonalba illeszthetünk be új töréspontot (vertexet).
- **Remove vertex** – *Töréspont eltávolítása.* Vonalból és határvonalból távolítja el a kijelölt töréspontot (vertexet).
- **Split line** – *Szakasz eltörése.* A kijelölt szakaszt a megadott helyen két különálló szakaszra bontja, a végpontokat pedig csomóponttá (node) alakítja.
- **Edit line/boundary** – *Vonal/határvonal szerkesztése.* Vonala és határvonal szerkesztésének folytatása.
- **Move point, line, boundary or centroid** – *Objektum mozgatása.* A kijelölt objektumot helyezhetjük át másik helyre.
- **Delete point, line, boundary or centroid** – *Objektum törlése.* A kijelölt objektumot teljes egészében törli.

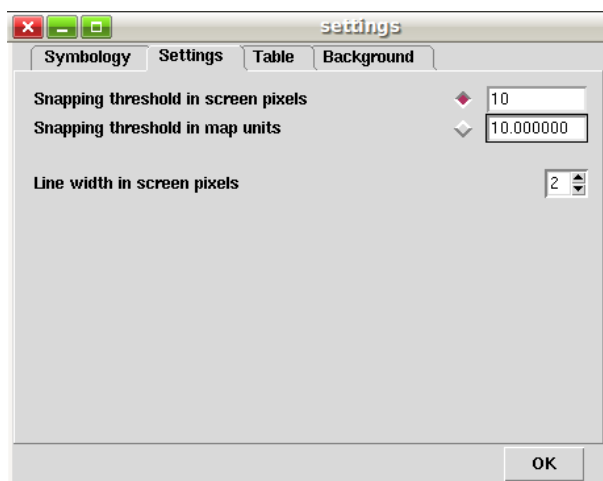
### 2.5.3. A digitalizáló beállításai

A digitalizáló alapbeállításai általában megfelelőek, azonban egyes esetekben szükséges lehet, hogy új értékeket állítsunk be. Ugyancsak itt van lehetőség arra, hogy az újonnan létrehozott vektoros térképhez adattáblát készítsünk.

- **Symbology** – az egyes elemek színének megadása. Ez arra szolgál, hogy a digitalizálás során az egyes elemek (vonal, pont stb.) olyan színnel jelenjenek meg, amely elegendő mértékben elüt a háttérként szolgáló térkép színétől (2.14. ábra).
- *Background* – háttér
- *Highlight* – kiemelés, az éppen kijelölt elem színe
- *Point* – pont színe
- *Line* – vonal színe
- *Boundary (no area)* – határvonal színe, ha nem fog körbe területet
- *Boundary (1 area)* – határvonal színe, ha egyetlen területet fog körbe
- *Boundary (2 areas)* – zárt határvonal színe, ha hozzá nem zárt határvonal csatlakozik
- *Centroid (in area)* – elsőként megadott súlypont, terület belsejében
- *Centroid (outside area)* – súlypont, területen kívül
- *Centroid (duplicate in area)* – ugyanazon területen belüli második és további súlypontok
- *Node (1 line)* – egy vonal csomópontja
- *Node (2 lines)* – két vonal közös csomópontja
- **Settings** – A digitalizálás során vonalak és pontokat hozunk létre. A vonalak gyakran nem különálló elemek, hanem össze kell azokat kapcsolnunk egymással. Ekkor az a probléma merül fel, hogy vajon csak akkor kapcsolódjék-e össze két vonal egymással, ha a



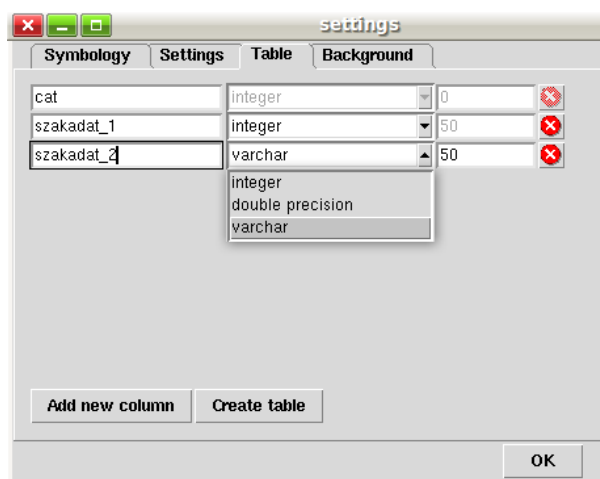
2.14. ábra  
A digitalizáló beállításai: *Symbology*



2.15. ábra  
A digitalizáló beállításai: *Settings*

két vonal egy-egy végpontja csakugyan egymáshoz ér, vagy már akkor is, ha elég közel vannak egymáshoz? A gyakorlatban az a tapasztalat, hogy az utóbbi megoldás a jobb. Mivel a munka során általában nem nagyítunk bele annyira a képbe, hogy pontosan egymáshoz illeszthessük a vonalakat, ezért célszerű egy olyan küszöbértéket megadni. Az egymáshoz ennél közelebb kerülő vonalvégeket a GRASS automatikusan összekapcsolja. Gyakran előfordul, hogy ezt az értéket a munka során változtatni kell: a túl kicsiny érték csak nagy nagyításnál jó, kis nagyításnál nagyon zavaró a használata, míg a túl nagy érték a nagy nagyítás során teszi lehetetlenné a munkát (2.15. ábra).

- *Snapping threshold in screen pixels* – Összekapcsolódási küszöb értéke képernyő-pixelekben
- *Snapping threshold in map units* – Összekapcsolódási küszöb értéke a térkép mértékegységében
- *Line width in screen pixels* – Vonalvastagság képernyő-pixelekben
- **Table** – Amikor új térképet hozunk létre, akkor ahhoz általában adattáblát kell csatolni. Ha ez az adattábla még nem létezik, akkor itt létrehozható. Ebben a részben már meglévő adattábla nem módosítható, csak az új térképhez készíthetünk benne egyetlen új táblát. Az elkészült táblát a GRASS automatikusan csatolja az új térképhez. Figyeljünk arra, hogy a *Create table* gomb lenyomása után már nem módosítható a tábla (2.16. ábra).



**2.16. ábra**  
A digitalizáló beállításai: *Table*

Balról jobbra az első mezőben adjuk meg az új adattábla-mező nevét, a másodikban pedig a típusát. A típus *integer* (egész szám), *double precision* (dupla pontosságú lebegőpontos szám, valamint *varchar* (változó hosszúságú szöveg) lehet. A harmadik mező a használható karakterlán/szám hosszát adja meg. Figyeljünk arra, hogy itt csak *.dbf* típusú adattábla hozható létre, vagyis tartsuk be a *.dbf* adattáblákra vonatkozó korlátozásokat! Ennek megfelelően a név

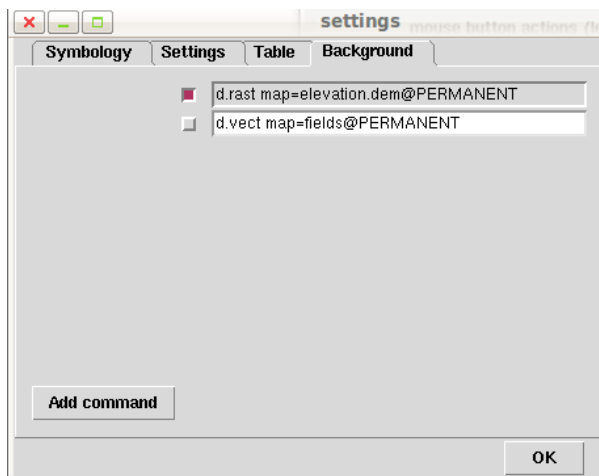
- ne legyen hosszabb 10 karakternél,
- ne tartalmazzon szóközt és ékezetes betűket,
- ne kezdődjék számmal,
- ne tartalmazzon kivonás-, összeadás-, szorzás- és osztásjelet, valamint egyéb műveleti jeleket sem.

Két gomb van az ablak alsó részén:

- *Add new column* – új adattábla-mező hozzáadása. Csak új adattábla készítésekor használható.
- *Create table* – Az új adattábla létrehozása.

- **Background** – Egyazon digitalizálás során több, különböző háttér-térképet is használhatunk, melyek közül mindig azt jelenítjük meg, amelyikre éppen szükségünk van. Egyszerre több térkép is megjeleníthető, de természetesen a felül levő kitakarhatja a lentebbi egyes részeit (2.17. ábra).

Háttér-térképet úgy adunk meg, hogy az *Add command* gombra kattintás után megjelenő mezőben megadjuk a megfelelő megjelenítő parancsot (erről részletesen lásd a 40. oldalon a háttér-térkép megadásáról szóló részt, valamint a Parancsok használata a GRASS-ban című részt). A háttér-térképek megjelenítését ugyanúgy engedélyezhetjük vagy tilthatjuk, mint a rétegek megjelenítését a *GIS Manager* ablakban.



2.17. ábra  
A digitalizáló beállításai: *Background*

## 2.5.4. A digitalizálás menete a GRASS-ban

A digitalizálás menetét legknyebben úgy sajátíthatjuk el, ha az egyszerűtől haladunk a bonyolultabb eljárások felé.

### 2.5.4.1. Pontok digitalizálása

A digitalizálás megkezdése előtt a nagyító eszközök használatával állítsuk be úgy a nagyítást, hogy a digitalizálandó objektumot világosan elkülöníthessük a környezetétől és elég nagyra látszódjék ahhoz, hogy biztosan jó helyre tehesük a digitalizálás során a pontot. Ezután kell kiválasztani a pont digitalizálása eszközt. Használata egyszerű: bal egérgombbal rakható le új pont, a jobb gombbal pedig az eszközről lehet kilépni (2.18. ábra). Ha elhelyezünk egy pontot, akkor nyomban megjelenik a szakadat-beviteli ablak és megadhatjuk a frissen digitalizált ponthoz tartozó szakadatokat. Ezek a térképhez csatolt adatbázisba kerülnek. Ha ilyen adatbázis nincs az éppen digitalizálás alatt álló térképhez csatolva, akkor figyelmeztető üzenetben kapunk e hiányosságról értesítést. Ilyenkor szakadatot rögzíteni nem lehet. Ha a szakadatot bevittük, akkor az ablak alján található *Resume* gomb megnyomásával rögzíthetjük az adatbázisban. Ha a művelet sikerrel lezajlik, akkor erről is tájékoztatást kapunk, mégpedig zölddel írva: *Record succesfully updated*.

Ha a digitalizálás előrehaladtával olyan helyre kell pontot felvinni, amelyik az adott nagyításban nem látható a digitalizáló balakban, két módon járhatunk el:

- a nagyítást csökkentve tesszük láthatóvá a keresett pontot (ez azonban ronthatja a digitalizálás pontosságát, ezért kerülendő),

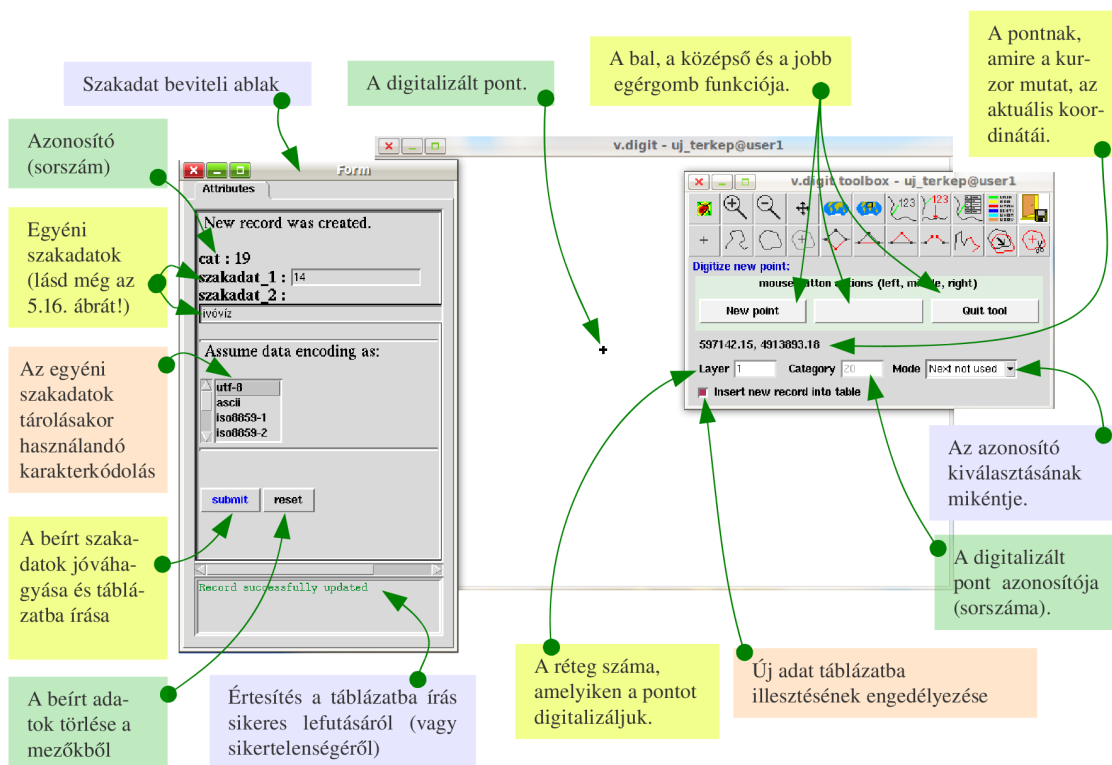
- a mozgatás (pan) eszköz használatával csúsztatjuk el a háttér-térképet és a róla digitalizált vektoros térképet, mindaddig, míg a keresett pontot meg nem találjuk.

A mozgatást úgy végezzük, hogy ha a mozgatás gomb lenyomása után a térkép egy pontjára kattintunk vele, akkor a térkép úgy modul el, hogy a kijelölt pont kerüljön a megjelenítő ablak középpontjába. Vagyis, ha például a megjelenítőablak széle felé eső részére kattintunk a térképnek, akkor az nyomban az ablak közepére kerül és így helyette a térkép további részei lesznek az ablak szélén. A művelet a *bal egérgomb* lenyomásával korlátlanul ismételhető. Kilépes a *jobb egérgomb*bal lehetséges.

Ha szükséges, a térképünk több réteget is tartalmazhat, vagyis eldönthetjük, hogy a digitalizálandó pont melyik rétegre kerüljön. Ehhez csak annyit kell tenni, hogy miután kiválasztottuk a digitalizálandó objektum típusát (jelen esetben a pontot), a digitalizáló eszköztár ablakában a *Layer* mezőben megadjuk, hogy melyik rétegre kívánjuk elhelyezni a digitalizálandó pontot. Ha a megadott réteg nem létezik, akkor azt a GRASS létrehozza. Ez azért nagyon hasznos, mert bár egyazon réteg tartalmazhat többféle objektumot is (például pontot és határvonalat), de egyes esetekben az a célszerű, ha a különböző típusú objektumokat különböző rétegekben tároljuk.

A digitalizáló eszköztár ablakában, a *Layer* mező mellett további kettőt is találunk: a *Category* és a *Mode* mezőket. A *Mode*-ban állíthatjuk be, hogy hogyan rendeljünk egyedi azonosítót az éppen digitalizált elemhez. Három lehetőséget kínál fel:

- *No category* – Nem adunk meg azonosítót és a GRASS sem ad meg *category* értéket
- *Manual entry* – Kézzel, saját magunk adunk meg azonosítót



2.18. ábra  
Pont digitalizálása

- *Next not used* – A GRASS hozza létre az azonosítót automatikusan, úgy, hogy a következő, még nem foglalt, egész számot rendeli a digitalizált obejkumhoz

Az alapértelmezett beállítás a *Next not used*. Ekkor, új térkép esetében, a GRASS az elsőként digitalizált elemhez az 1-es számot rendeli, következőhöz a 2-t és így tovább. Ha nem új, még üres térképen dolgozunk, akkor az első olyan egész számot rendeli a digitalizált elemhez, amelyik még nem foglalt (mondjuk, 20).

Ha a *Next not used* beállítás van érvényben, akkor a *Category* mező mutatja meg, hogy az éppen digitalizált objektumnak mi lesz az azonosítója. Ha a *Manual entry*-t választottuk, akkor itt adható meg az álatunk kívánt azonosító, ez azonban ekkor is csak pozitív egész szám lehet.

Fontos megértenünk, hogy a digitalizálás során létrehozott elemek összessége egy geometriai adatbázist képez, amelyben akkor lehet hivatkozni az egyes elemekre, ha mindegyiknek van valamilyen egyedi azonosítója. Ezt az egyedi azonosítót a GRASS-ban *Id*-nak (*identifier*) nevezik és nem ugyanaz, mint a *category*, vagyis a felhasználó által megadható azonosító. A digitalizálás során a GRASS minden objektumhoz hozzárendel egy-egy *Id*-t, mindegyikhez egyet és egy azonosító csak egyszer szerepelhet ugyanazon térképen belül. Ez teszi lehetővé, hogy maga a program képes legyen azonosítani az egyes rajzi elemeket. A *category* (kategória) arra szolgál, hogy a felhasználó szempontjai szerinti azonosítás is lehetséges legyen. Azért is lényeges ez a különbségtétel, mert általában a *category* szerepel kulcsmezőként az adatbázissal való összekapcsolásban.

### 2.5.4.2. Vonal digitalizálása

Vonal digitalizálása alapvetően a pont digitalizálásánál leírtakhoz hasonlóan lehetséges, bizonyos különbségekkel, természetesen.

- A bal egérgombbal adjuk meg a vonal kezdőpontját és minden további töréspontját.
- A középső gombbal vonhatjuk vissza a vonal kezdőpontjának vagy töréspontjának létrehozását. A visszavonás korlátlanul ismételhető, de csak addig, amíg a vonalat le nem zárjuk.
- A vonal lezárása a jobb egérgomb megnyomásával lehetséges. Ekkor az utolsóként megadott pont lesz a vonal végpontja. A vonal lezárása művelet nem vonható vissza.
- A vonal lezárása után (a pont digitalizálásakor már látott módon) megjelenik az attribútum táblázat és a frissen digitalizált vonalhoz rögzíthetjük a szükséges szakadatokat is.
- Ha a vonalnak olyannak kell lennie, hogy kezdő- és végpontja egybeessen (vagyis zárt vonal legyen az eredmény), akkor a töréspont mozgatása gombot választva lehetőségünk van arra, hogy a vonal kezdő- és végpontját összekapcsoljuk. A töréspont mozgatása gomb megnyomása után válasszuk ki a bal egérgombbal az elmozdítandó végpontot, vigyük a kurzort a másik végpontra, majd ismét kattintsuk egyet a bal egérgombbal. Ha a művelet sikeres volt, akkor a végpontokat jelző piros keresztek helyett egyetlen zöld kereszt jelenik meg. Ekkor a kérdéses vonal kezdő- és végpontja pontosan egybeesik. A vonal színe ennek során nem változik.

Fontos tudnunk, hogy ettől ez még vonal marad, nem pedig határvonal! Ha területet kell digitalizálnunk, használjuk a határvonal digitalizálása eszközt! (Vonal kezdő- és végpontjának egybeesése csak azt jelenti, hogy a vonal eleje és vége ugyanoda esik, de nem jelenti azt, hogy az így bezárt területet különál-

ló entitásként kellene kezelni, vagy, hogy ez a vonal bármiféle határt képezni. Ilyen lehet például egy versenypálya nyomvonala, vagy zárt hurokként felépülő vezetékhalózati is.)

### 2.5.4.3. Határvonal digitalizálása, terület digitalizálása

Létrehozása megegyezik a vonal digitalizálásával. Egyetlen különbség csak az, hogy a határvonal bezárásakor (vagyis amikor a kezdő- és a végpont egybeesik) nemcsak a végpontokat jelző keresztek színe változik pirosról zöldre, hanem maga a vonal is ugyanilyen színű lesz. Határvonalat akkor használunk, ha a vonal valóban mint elválasztó elem működik, a bal- és a jobb oldalán különböző területek (poligonok) definiálhatók.

*Területet (area)* a GRASS-ban határvonal segítségével digitalizálhatunk.

Ha a digitalizálandó területek egymástól teljesen különállóak, közös határuk nincsen, akkor egyszerűen csak körberajzoljuk a határukat, ügyelve arra, hogy a határvonal kezdő- és végpontja egybeessen.

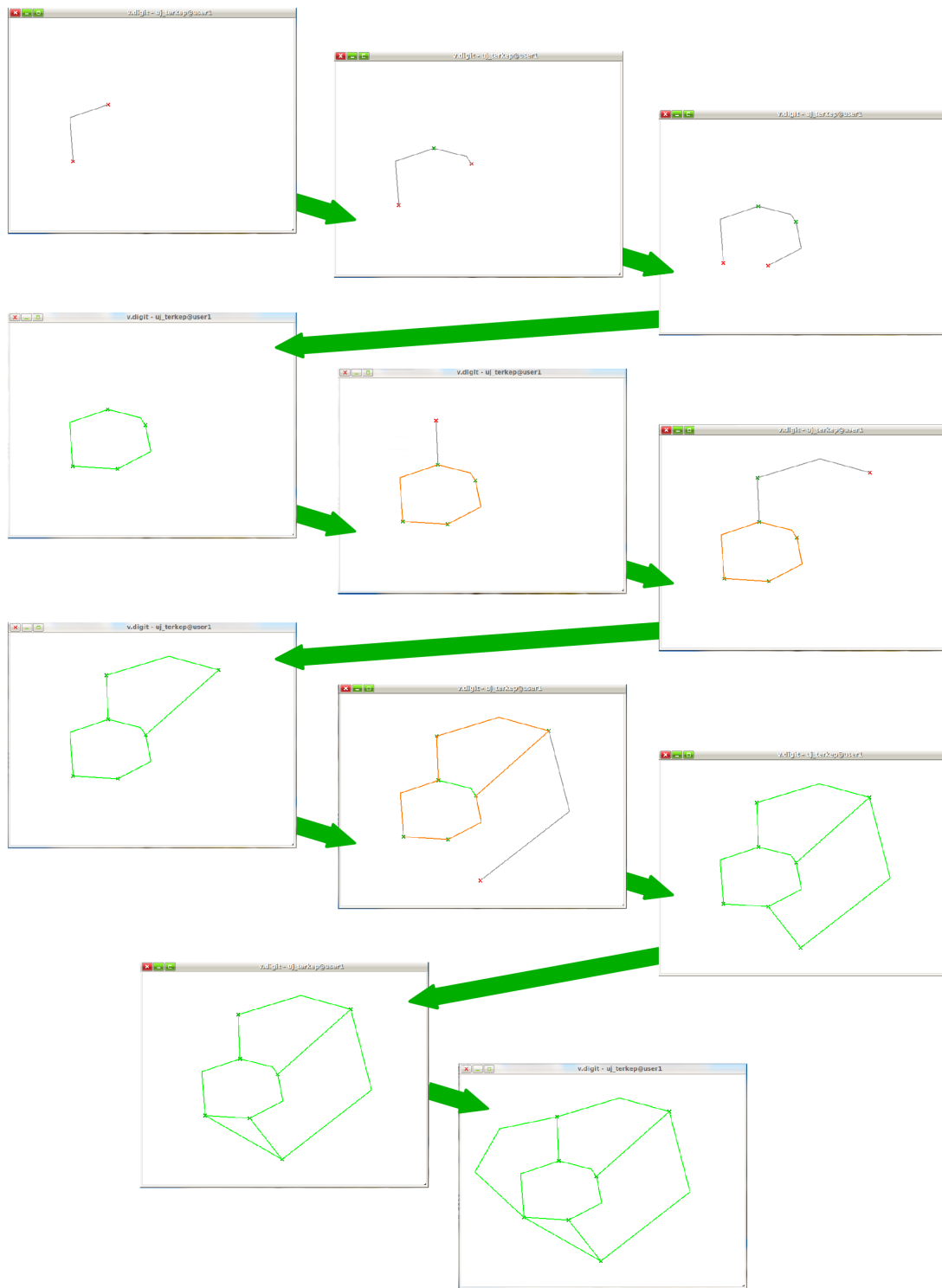
Ha azonban egy vagy több közös határuk van, akkor másképp kell eljárni, ebben az esetben ugyanis minden egyes határszakaszt külön-külön kell digitalizálni. Határszakasz az a szakasz, amelynek a két oldalán egy-egy terület definiálható. Az egyes, egymás után következő határszakaszoknak kapcsolódniuk kell egymáshoz, vagyis az egyik a végpontjának a másik kezdőpontjával kell egybeesnie. Fontos sajátosság, hogy egyszerre több határszakasz végpontja és kezdőpontja is eshet ugyanarra a pontra.

Végül jegyezzük meg, hogy csak az ugyanazon rétegen levő határvonalak alkalmasak terület definiálására. Vagyis, két különböző rétegen levő határvonalak nem alkalmasak arra, hogy közösen határozzanak meg egy területet. Ettől függetlenül természetesen akárhány rétegen lehetnek különféle határvonalak, de mindig csak az ugyanazon rétegben levők kapcsolhatók össze egy-egy terület határaivá.

Érdekes az 2.19. ábrán a digitalizálás folyamatát követni. A végül egy-egy területtől álló poligonok határai nem egyszerre, egy vonallal készültek el, hanem olyan különálló darabokból, melyek mindegyike ugyanazt a két területet választja el egymástól, vagy egy területet a külvilágtól. A még bezáratlan határvonal-szakaszok szürkék, a már bezárt, tehát területet közrefogó, vagy két területet elválasztó vonalak pedig zöldek. Ha egy zárt, területet közrefogó határvonalhoz még bezáratlan (tehát szürke) határvonal csatlakozik, akkor narancssárga lesz. Az egymáshoz illesztett végpontok zöldek, az egyedülállóak pedig (amelyek tehát csak egyetlen vonalnak a végpontjai) pirosak. A munka során a határszakaszok digitalizálásának sorrendje közömbös, de persze érdemes valamiféle rendszer szerint végezni a munkát. Figyeljünk arra, hogy a munka befejeztével minden határszakasz zöld legyen (tehát ne legyen nyitott határvonal) és minden csomópont is zöld legyen (tehát illeszkedjenek egymáshoz a megfelelő kezdő- és végpontok)!

Terület digitalizálásakor ne feledkezzünk meg a legszélső vonalak (a digitalizálandó terület külső határai) digitalizálásáról sem! Ha kell, használjuk a *mozgatás* eszközt, illetve váltsunk bátran nagyítást, különösen az aprólékos részletek pontos követéséhez! Figyeljünk arra is, hogy szükséges lehet a vonalak automatikus összekapcsolásának küszöbértékét is változtatni (lásd *A digitalizáló beállításai* részt).

A területek végleges digitalizálásához általában szükséges centroid (súlypont) elhelyezése is (lásd a következő részben).

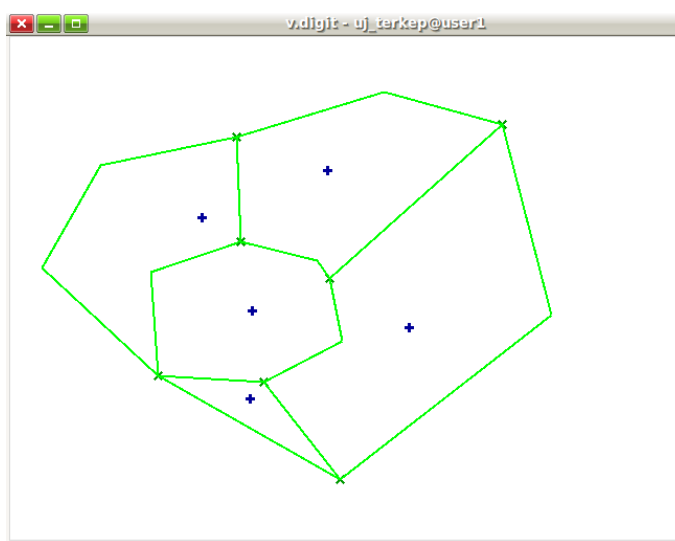


**2.19. ábra**  
Terület digitalizálásának folyamata. Figyeljük meg, hogyan változik a határvonalak és a csomópontok színe (lásd még *A digitalizáló beállításai* című részben a *Symbology-t*)

#### 2.5.4.4. Súlypont (centroid) digitalizálása

A területek digitalizálásának csak egyik része a határvonalak elkészítése. Ahhoz, hogy a határvonalak által bezárt területeket önálló entitásként értelmezhessük, azokhoz szakadatokat rendelhessünk, szükséges súlypontot is elhelyezni a poligon területén belül (2.20. ábra). A súlypont kissé szerencsétlen elnevezés, mert valójában nem szükségképpen a terület geometriai értelemben vett súlypontja. Pusztán csak arról van szó, hogy a határvonalak által bezárt területre csak akkor lehet önállóan hivatkozni, ha van valamifajta azonosítója. A centroid nem más, mint egy olyan pont, amely alkalmas arra, hogy azonosítsa egy és csakis egy területet. Létrehozása tehát pontosan úgy történik, mint a pont létrehozása, azzal a megköttéssel, hogy ugyanazon a rétegen (layer) kell lennie, mint azok a határvonalak, amelyek a kérdéses területet határolják. Nyilvánvalóan, ha a térképünk olyan, hogy több rétegben is vannak határvonalak, akkor mindegyik rétegben definiálhatunk centroidokat. A centroid az adott területen belül bárhol elhelyezhető, megkötések nélkül. Általában arra a pontra rakjuk, amely pont a legjobban jellemzi a kérdéses területet (például a város központi tere).

Ugyanazon területen belül több centroid is elhelyezhető, ebben az esetben az előbb részletezett szerepet általában az elsőként elhelyezett pont játssza. Az elsőként letett centroidot a GRASS alapértelmezésben más színnel jelzi, mint a többit (lásd *A digitalizáló beállításai* című részben a *Symbology*-t).



2.20. ábra  
Területek és centroidok

#### 2.5.4.5. Már digitalizált elemek módosítása

Bármely, már digitalizált elem összes tulajdonság megváltoztatható később is. Egyrészt megváltoztathatók a szakadatok, másrészt a réteg- és kategóriaadatok, harmadrészt pedig a geometriai adatok, vagyis az objektum kinézete (például egy vonal alakja).

##### Szakadatok módosítása

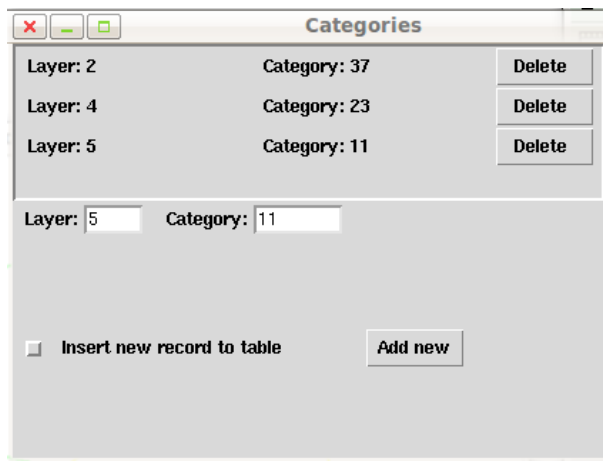
A szakadatok megváltoztatásához a *Display attributes* (Szakadatok megjelenítése és szerkesztése) gombot használhatjuk. Lenyomása után egyszerűen csak ki kell választanunk a kívánt objektumot és bal egérgombbal kattintani rajta. Megjelenik a digitalizálás során már megismert adatbeviteli ablak és ebben szükség szerint módosíthatjuk az objektumhoz rendelt szakadatokat. A változtatások elmentéséhez a *Resume* gombra kell kattintani.

A réteg- és kategóriaadatok megváltoztatásához válasszuk a *Display categories* (Kategóriaadat megjelenítése) gombot! A gomb megnyomása után kell kijelölni bal egérgombbal a változtatandó elemet. Ha kijelöltük a kívánt elemet, akkor a megjelenő ablakban egyrészt leolvasható, hogy melyik rétegen van jelenleg az adott elem, valamint az aktuális kategóriaértéke is. A változtatáshoz töröljük a jelenlegi beállítást a jobb oldalon levő *Delete* gombbal, és adjuk meg az új réteg, illetve kategória értéket a megfelelő mezőkben, majd nyomjuk meg az *Add new* gombot! A törlés és az új érték definiálásának sorrendje egyébként közömbös.

Ebben az ablakban arra is lehetőség van, hogy egyazon elemhez több réteg- és kategóriadatot csatoljunk. Ezt úgy lehet megtenni, hogy a *Layer* (réteg) és a *Category* (szakadat) mezőkbe újabb adatot kell beírni és ismét meg kell nyomni az *Add new* (új hozzáadása) gombot. A nem kívánt adatok természetesen a megfelelő sornak az egérrel való kiválasztása után, a sor mellett levő *Delete* gombbal törölhetők. Így például, ha egy vonalhoz az 2.21. ábrán látható adatok tartoznak, az azt jelenti, hogy a kérdéses vonal a 2-es, a 4-es és az 5-ös rétegen is ugyanott található, de a kategóriaadata mindegyiken más.

Vagyis, ha ebben az ablakban azt adjuk meg egy elemhez, hogy az több rétegben is előfordul, akkor ezzel tulajdonképpen az adott elemet másoljuk a megadott rétegekbe úgy, hogy mindegyik rétegben az ott megadott kategóriaadat lesz rá érvényes.

Ez az eljárás akkor például akkor hasznos, ha különféle vonalas létesítmények, vagy határvonalak esnek térbelileg egybe, de funkciójuk különböző. Ilyen például az, ha ugyanazon közműalagútban, ugyanazon nyomvonalon futnak eltérő vezeték (mondjuk a kábeltévéé és az optikai gerinchálózaté), vagy ha egy közigazgatási határ egybeesik valamilyen ellátási határral.



2.21. ábra  
Digitalizált elemek módosítása: Display categories

### Geometriai tulajdonságok megváltoztatása

A geometriai sajátosságok megváltoztatása több módon lehetséges. Pont és centroid esetében csak két változtatási lehetőség van, az áthelyezés és a törlés. Vonal és határvonal esetében ezeken kívül még megváltoztatható a töréspontok száma és helye is, új csomópontokat lehet létrehozni (vagyis kettévágható a vonal) és lehet folytatni a szerkesztést is az egyik csomóponttól indulva.

- **Objektum áthelyezése** – A *Move point, line, boundary or centroid* gomb megnyomása után lehetséges. Először bal egérgombbal ki kell jelölni a mozgatni kívánt elemet (ekkor az sárga lesz), aztán a kurzort az új helyre kell vinni és ismét kell egyet kattintanmi bal egérgombbal: ekkor a kijelölt objektum erre az új helyre kerül. Ha rossz lemmet választottunk ki, a középső egérgombbal vonható vissza a kiválasztás és újból lehet próbálkozni. Kilépés a jobb gombbal.
- **Objektum törlése** – A *Move point, line, boundary or centroid* gomb megnyomása után végezhető el. Először bal egérgombbal ki kell jelölni a törendő objektumot (a kijeölt elem sárgára vált), aztán a bal egérgom ismételt lenyomásával megerősíteni a törlést. A középső egérgomb vissza-

vonja a kijelölést és újra lehet próbálkozni, a jobb egérgombbal pedig kilépni lehet a törlés menüből.

- **Vonal egy csomópontjának vagy töréspontjának mozgatása** – A *Move vertex* gomb megnyomása után tehetjük meg. Használata mindenben megegyezik az objektum áthelyezésénél elmondottakkal.
- **Töréspont (vertex) beszúrása** – Az *Add vertex* gomb megnyomása után először bal egérgombbal jelöljük ki a leendő töréspont helyét a vonalon vagy határvonalon. Ha a kijelölés sikerült (oda kattintottunk, ahová valóban szerettünk volna), akkor az egér elmozdítása nélkül, bal egérgomb ismételt lenyomásával megerősíthetjük az új töréspont létrehozását. Ha az új töréspont segítségével rögtön meg is akarjuk változtatni a vonal futását, akkor az első kattintás után húzzuk a kurzort a kívánt helyre és ott kattintsunk ismét a bal gombbal. Természetesen a középső gomb ebben az esetben is elveszi a kijelölést és újra lehet próbálkozni az új töréspont helyének kijelölésével. Kilépés most is a jobb egérgombbal lehetséges.
- **Töréspont eltávolítása** – Ha túl sok töréspontot tartalmaz egy vonal vagy határvonal, akkor a *Remove vertex* gomb megnyomása után bal gombbal kell kijelölni az eltávolításra szánt vertexet és a gomb ismételt megnyomásával lehet eltávolítani. Ha a kijelölés hibás volt, a középső gomb visszavonja azt és újból lehet próbálkozni. Kilépés a jobb gombbal.
- **Szakasz (vonallal) eltörése** – A *Split line* gomb megnyomása után tehetjük meg azt, hogy a kiválasztott vonalat vagy határvonalat a bal egérgombbal kijelölt helyen eltörjük. Ilyenkor tulajdonképpen az történik, hogy két új csomópont keletkezik, az eddigi egyetlen szakasz helyett kettőt kapunk eredményül. A két szakasz egymástól teljesen független objektum lesz, de szakadatuk (ha volt) ugyanaz marad. Figyeljük meg, hogy a törés után a csomópontokat jelző kereszt színe zöld, vagyis a két csomópont egybeesik. A csomópontok elmozgatása esetén természetesen a jelölés pirosra változik. A kijelölés helye – az eddigiekkel egyezően – bal egérgombbal hagyható jóvá, középső gombbal vonható vissza és a jobb egérgomb itt is a kilépést eredményezi. Érdemes figyelni arra, hogy szakasz törése után ellenőrizzük az eredményül kapott szakaszok kategóriaadatát és szakadatát is! Ha a szakasz eltörése után kapott új csomópontok valamelyikét új helyre akarjuk pozicionálni, akkor használjuk a *Move vertex* gombot (lásd fentebb).
- **Vonal/határvonal szerkesztése** – Az *Edit line/boundary* gomb megnyomása után, bal egérgombbal kattintva jelölhető ki a szerkesztendő vonalat vagy határvonalat. A kijelölés után a vonal szerkesztése pontosan ugyanúgy folytatható, mint ahogyan az a vonal digitalizálása részben olvasható. Figyeljünk arra, hogy a vonal szerkesztését a program a vonalnak attól a végétől folytatja, amelyikhez közelebb esett a kattintás a vonal kijelölésekor!

## 2.6. Adattáblák és adatbázisok kezelése a GRASS-ban

A GRASS-ban az adattáblák kezelése alapvetően vektoros állományokhoz kötötten történhet, bár sok művelet azoktól függetlenül is elvégezhető.

### 2.6.1. Adatbázis és adattábla

A kötetben csak nagyon kevés alapfogalom tisztázására térünk ki, azokra is csak akkor, ha a tapasztalat szerint szükséges. Az adatbázis és az adattábla fogalma ebbe a körbe tartozik. Hétköznapi értelemben az adatbázis valami olyasmit jelent, hogy adatok előre meghatározott rendszerbe foglalt halmaza. A számítástechnikában ez azonban nem így van. Érdemes tehát megjegyezni:

*Az adatbázis az a könyvtár, amely az adattáblákat tartalmazza. Adattábla az a táblázat, amelyben előre megadott szabályok szerint adatokat helyezünk el. Egyazon adatbázis több adattáblát is tartalmazhat.*

Vagyis, a hétköznapi fogalmaink szerinti adatbázisnak az adattábla felel meg, míg az adatbázis a számítógép számára nem más mint egy könyvtár, amelyben adattáblák találhatóak.

### 2.6.2. Vektoros térkép és adattábla csatolása

A vektor alapú térképkezelés előnyeit csak akkor lehet kihasználni, ha a térképhez adattáblát is csatolunk. Lehetőség van arra is, hogy minden réteghez más-más adattábla tartozzon, de ebben az esetben is igaz, hogy egyszerre csak egy tábla kapcsolható egy-egy réteghez.

Az adattábla-csatolást a *Databases* menü *Vector-database connections* almenü *Set vector-database connection* pontjával kezdeményezhetjük. A megjelenő ablak (*v.db.connect*) első mezőjében adjuk meg annak a vektoros térképnek a nevét, amelyhez adattáblát akarunk csatolni. Figyeljünk arra, hogy ez a művelet csak az aktuálisan használt mapsetben levő térképeken hajtható végre! Ezután ki kell választani a használandó adattábla típusát, pontosabban azt, hogy a GRASS milyen meghajtóprogrammal kapcsolódjon az adattáblához. Az alapértelmezés a *dbf*, de lehetőség van még *mysql*, *odbc*, *ogr*, *pg* és *sqlite* használatára is.

- A *Database name* mezőbe írjuk be a használandó adatbázis nevét és elérési útját (ne feledjük az adattábla és az adatbázis közötti különbséget!). A GRASS alapesetben azt tételezi fel, hogy a használni kívánt adatbázis nem más, mint az éppen használt mapsetben levő *dbf* könyvtár. Ezt fejezi ki az alapértelmezésben olvasható sor:

```
$GISDBASE/$LOCATION_NAME/$MAPSET/dbf/
```

Tekintve, hogy a \$ (dollár) jel a számítástechnikában értékadást jelent, a fenti sort valahogy így lehetne magyarrá fordítani:

„Az éppen használt GRASS adatbázis (pl.: */home/titusz/grass*) éppen használt location-je (pl.: *spearfish60*) aktuálisan megnyitott mapsetjében (pl.: PERMANENT) levő *dbf* könyvtár.” Ennek a megadási módnak tehát az az előnye, hogy mindig éppen arra a környezetre vonatkozik, amelyikben vagyunk. Természetesen ez felülbírálható azzal, ha egyszerűen beírjuk a használni kívánt adatbázis (tehát egy könyvtár!) nevét, mondjuk azt, hogy */home/user/adatbazis*.

Jó ha tudjuk, hogy az alapértelmezett adatbázis megváltoztatható, lásd a 2.6.3.3. részben a 58. oldalon.

Az adattábla nevének megadásakor figyeljünk arra, hogy nem kell kiterjesztést megadni! Tehát, ha a csatolni kívánt adattábla fájlneve mondjuk az, hogy *baranya.dbf*, akkor csak annyit írjunk az adattábla mezőbe, hogy „*baranya*”!

- A kulcsmező megadása (*key column*) mezőben kell megadni a kulcsmező nevét. Figyeljünk rá, hogy kulcsmező csak egész szám (integer) típusú mező lehet!
- A *Layer number* (réteg száma) mezőbe írjuk be annak a rétegnek a számát, amelyhez az adattáblát csatolni szeretnénk.
- Ha minden szükséges adatot megadtunk, akkor már csak az „*Overwrite connection parameter for certain layer*” opciót kell engedélyezni és a *Run* gombot megnyomni.

Ez a paraméter (a csatolási paraméterek felülírása a megadott rétegre vonatkozóan) engedélyezi, hogy a megadott paramétereknek megfelelően valóban létrejöhessen a csatolás. Ha ezt az opciót nem engedélyezzük, akkor a GRASS nem hajtja végre a kívánt összerendelést.

- Ellentétes hatású a „*Delete connection for certain layer (not the table)*” opció engedélyezése. Ez ugyanis törli az eddig meglévő kapcsolatot a térkép és az adattábla között, de csak a összekapcsolást szünteti meg, magát az adattáblát természetesen nem törli.

A v. db.connect ablak opcióinak egy része a Print fülben, másik része pedig az Options fülben érhető el.

- **Print all map connections parameters and exit** – Az output ablakban megjeleníti a *Name of input vector map* mezőben megadott vektoros térkép összes adattábla kapcsolatát és azok paramétereit. Egyéb művelet nem történik, így az összekapcsolás sem megy végbe.
- **Print all map connections parameters and exit in shell script style** – Megegyezik az előzővel, de a kimenet shell script-ben használható formában jelenik meg (ennek a munka automatizálásakor és a parancsok használatakor van értelme, részletesen erről ott lehet olvasni). Egyéb művelet nem történik, így az összekapcsolás sem megy végbe.
- **Print types/names of table columns for specified layer and exit** – Csak a megadott vektoros térkép megadott rétegéhez csatolt adattábla oszlopainak (mezőinek) nevét és típusát írja ki. Egyéb művelet nem történik, így az összekapcsolás sem megy végbe.
- **Field separator for shell script style output** – Ebben a mezőben azt adhatjuk meg, hogy mi legyen a mezőelválasztó karakter, ha a *Print all map connections parameters and exit in shell script style* opció van engedélyezve.

- **When printing, limit to layer specified by the layer option** – Megegyezik a *Print all map connections parameters and exit* opcióval, de csakis a megadott réteghez tartozó adatokat jeleníti meg.
- **Overwrite connection parameter for certain layer** – Engedélyezi, hogy a megadott paramétereknek megfelelően valóban létrejöhessen a vektoros térkép és az adattábla összekapcsolása. Ha ezt az opciót nem engedélyezzük, akkor a GRASS nem hajtja végre a kívánt összerendelést.
- **Delete connection parameter certain layer (not table)** – Törli az eddig meglévő kapcsolatot a térkép és az adattábla között, de csak a csatolást szünteti meg, magát az adattáblát nem törli.

### 2.6.3. Adatbázis-műveletek a GRASS-ban

Az adatbázisokkal kapcsolatos műveleteket két nagy csoportban tárgyalhatjuk: az egyik a lekérdezések és információkérések, a másik pedig az adattáblák módosításai. Az adatbázis műveletek legtöbbjét a *Databases* menüből érhetjük el. Egyetlen kivétel a *Vector* menü *Reports and statistics* almenüjének *Upload or report topology* pontja, melyen keresztül adatokat tölthetünk a vektoros térképből a hozzá csatolt adattáblába (lásd később).

#### 2.6.3.1. Általános információk kérése adattáblákról

A *Databases* menü *Database information* almenüjéből az alábbiakban felsorolt általános információkat kérhetjük le.

- **Describe table** (Adattábla részletes leírása) – A *Database name* mezőben megadott adatbázis, *Table name* mezőben megadott nevű adattáblájának mezőit mutatja meg és azok minden tulajdonságát. A *Run* gomb megnyomása után az információk az *Output* ablakban jelennek meg. Ha be van kapcsolva a *Print column names instead of full column descriptions* opció, akkor ugyanaz, mint a következő menüpont. (A parancs neve: *db.describe*)
- **List columns** (Adattábla oszlopainak [mezőinek] listázása) – Használata megegyezik az előzővel, de csak a mezők nevét adja meg. Ugyanaz, mint az előző, ha be van kapcsolva a *Print column names instead of full column descriptions* opció. (A parancs neve: *db.columns*)
- **List drivers** (Az elérhető [használható] adatbázis-meghajtók [drivereket] listája) – Kिलistázza a használható adatbázis-típusokat, vagyis azokat, amelyekhez van a kapcsolódást biztosító segédprogram. (A parancs neve: *db.drivers*)
- **List tables** (Megadott adatbázisban levő, megadott típusú adattáblák listája) – Kिलistázza a *Database name* mezőben megadott adatbázis mindazon adattábláját, melynek típusa egyezik a *Driver name* mezőben kiválasztottal. (A parancs neve: *db.tables*).

Ez az információkérés kissé következetlennek tűnhet az ablakozós rendszereken szocializálódott felhasználó számára. Mert mi értelme van annak, hogy az egyik menü pont ugyanazt tudja, mint egy másik, ha abban egy bizonyos opció engedélyezve van. A megoldás az, hogy ezek nem feltétlenül ugyan-

azok a programok (szkriptek). Ennek az az értelme, hogy a feladatok automatizáláskor (lásd a megfelelő részben) így a megfelelő parancs futtatható, ha például a cél csak az, hogy egy bizonyos adattábla mezőinek nevét megtudjuk.

### 2.6.3.2. Adattábla lekérdezése

A *Database* menü *Query* (lekérdezés) almenüjének pontjaival lehetséges.

- **Query any table** (Bármely tábla lekérdezése) – Egy kiválasztott adatbázis, megadott választott táblájának összes mezőjét és rekordját kilistázza az *Output* ablakban, feltéve, hogy a tábla a *Driver name* mezőben kiválasztott meghajtóprogram által olvasható.

Lehetőség van arra is, hogy csak azokat a rekordokat listázza a program, melyek az *SQL select statement* mezőben megadott SQL kifejezésnek megfelelnek. Ha a használni kívánt SQL kifejezés egy fájlban van leírva, akkor ezt az állományt kell kiválasztani a *Name of file with sql statement mezőben*. A használni kívánt állománynak ASCII típusúnak – vagyis egyszerű szöveges állománynak – kell lennie.

A lekérdezés eredménye úgy jelenik meg, hogy az *Output field separator* mezőben megadott karakter (vagy karaktersorozat) lesz a mezőelválasztó és az *Output vertical record separator* mezőben megadott karakter (vagy karaktersorozat) lesz a rekord-elválasztó. Ha a lekérdezés eredménye NULL-értéket is tartalmaz, akkor a *NULL value indicator* mezőben megadott karakter (vagy karaktersorozat) fogja ezt reprezentálni.

Természetesen most is, mint a GRASS-ban általában, arról van szó, hogy ezek az opciók a GRASS parancs-soros használatát, a feladatok megoldásának automatizálását könnyítik meg nagyon jelentős mértékben. A használandó parancs neve egyébként: *db.select*.

A *db.select* parancs és az ablak opciói:

- *Do not include column names output* – A lekérdezés eredménye ne tartalmazza az adattábla oszlopainak (mezőinek) nevét (vagyis elhagyja a fejléc megjelenítését).
- *Vertical output (instead of horizontal)* – Minden adat új sorban jelenik meg, szemben a szokással, ahol az ugyanabba a rekordba tartozó adatok ugyanabban a sorban jelennek meg. Ha nincs bekapcsolva a *Do not include column names output opció*, akkor minden sor elején kiírja az adott mező nevét is.
- *Describe query only (don't run it)* – Csak leírja a lekérdezést, de nem futtata.
- *Only test query, do not execute* – Nem fut le a lekérdezés, csak a parancssoros kimenete jelenik meg az *Output* ablakban
- **Query vector attribute table** (Vektoros térképhez csatolt adattábla lekérdezése) – Ezt a menüpontot akkor célszerű használni, ha egy adott vektoros térképhez tartozó adattáblát akarunk lekérdezni. Ennek megfelelően az első mezőben a kérdéses vektoros térkép nevét kell megadni, illetve a kérdéses réteg számát a következő mezőben. Egyebekben ennek a menünek a használata megegyezik a *Query any table* menüével. A használandó parancs: *v.db.select*.

A *v.db.select* paranacs és az ablak opciói:

- *Print minimal region extent of selected vector features instead of attributes* – Annak a legkisebb területnek az É-i, D-i, K-i és Ny-i határát adja meg, amelyen belül az összes olyan elem megtalálható, mely megfelel a megadott paramétereknek.
- *Do not include column names output* – A lekérdezés eredménye ne tartalmazza az adattábla oszlopainak (mezőinek) nevét (vagyis elhagyja a fejléc megjelenítését).
- *Vertical output (instead of horizontal)* – Minden adat új sorban jelenik meg, szemben a szokással, ahol az ugyanabba a rekordba tartozó adatok ugyanabban a sorban jelennek meg. Ha nincs bekapcsolva a *Do not include column names output opció*, akkor minden sor elején kiírja az adott mező nevét is.
- **SQL statement** (SQL kifejezés) – A megjelenő ablakban választható ki az a szöveges állomány, amelyikben a futtatni kívánt SQL kifejezés található. Meg kell még adni a használandó adatbázis-meghajtóprogram típusát is, valamint a lekérdezendő adatbázist. A parancssoros megfelelője: *db.execute*. Egyetlen opciója van:
  - *Ignore SQL errors and continue* – A futtatás során hagyja figyelmen kívül az esetleges SQL hibákat és folytatja a kérés végrehajtását.

### 2.6.3.3. Adatbázis műveletek

A *Databases* menü *Manage database* almenüjének pontjai szolgálnak arra, hogy adatbázisokon műveleteket végezzünk. Ezen kívül a *Vector* menü *Reports and statistics* almenüjének *Upload or report topology* pontja és a *digitalizáló felület* beállításainak *Table* része is végez adatbázis műveleteket.

- **Connect to database** (Csatlakozás adatbázishoz) – A megadott paraméterekkel adatbázishoz csatlakozik a GRASS. Az itt megadott paraméterek és adatbázis lesz alapértelmezett mindaddig, míg – ugyanilyen módon – felül nem bíráljuk. Másképpen fogalmazva: az itt beállított adatbázis jelenik meg alapértelmezettként, ha például vektor-adatbázis csatlakozást kívánunk létesíteni.
- *Print current connection parameters and exit* – Megjeleníti az *Output* ablakban az éppen aktuális csatlakozási paramétereket
- *Check connection parameters, set if uninitialized, and exit* – Ellenőrzi, hogy vannak-e megadva csatlakozási paraméterek és ha nincsenek, akkor a GRASS alapértelmezett értékeit használja. Ez az opció scriptek alkalmazásakor hasznos, új adattábla létrehozása előtt.
- **Login to database** (bejelentkezés adatbázisba) – Ezen az ablakon keresztül adatbázisba jelentkezhetünk be, a *Username* mezőben megadott felhasználónév és a *Password* mezőben megadott jelszó segítségével. Figyeljük meg, hogy a *Database* mezőben alapértelmezésben annak az adatbázisnak a neve jelenik meg, amit a *Connect to database* ablakban beállítottunk. Persze, ha ez most nem lenne megfelelő, akkor minden további nélkül átírható itt is.
- **Copy table** (Tábla másolása) – Az *input* mezőkben megadott paraméterű táblát másolja az *output* mezőkben megadott paramétereknek megfelelően. Lehetőség van arra is, hogy ne a teljes

táblát másolja át, hanem csak azokat a rekordokat, amelyek megfelelnek az utolsó két mező valamelyikében megadott SQL kifejezésnek.

- **New table** (Új tábla létrehozása) – Létrehoz a *Name of new attribute table* mezőben megadott néven egy új adattáblát és ezt a táblát rögtön csatolja is a *Vector map for wich to add new attribute table* beviteli mezőben megadott vektoros térképnek a *Layer where to add new attribute table* beviteli mezőben megadott számú rétegéhez. Az új adattáblának legalább egy mezőt, a kulcsmezőt tartalmaznia kell már a létrehozáskor. Hogy mi legyen a kulcsmező neve, azt adhatjuk meg az utolsó beviteli mezőben. Alapértelmezésként felkínálja a *cat* (ez a *category* kifejezés rövidítése) szót a kulcsmező nevéül, de ez természetesen felülbírálható. Az adattábla létrehozásakor rögtön létrehozhatjuk a szükséges mezőket (oszlopokat) is, ugyanitt, az utolsó beviteli mezőben. Ennek módja az, hogy megadjuk a kívánt mező nevét, valamint szóközzel elválasztva a típusát (*varchar*, vagyis szöveges, *int*, vagyis egész szám, *double*, vagyis dupla pontosságú lebegőpontos szám, *date*, vagyis dátum). Ha több mezőt is meg akarunk adni, akkor az egyes mezőket vesszővel elválasztva kell felsorolni, például így:

*cat integer,adat1 double, adat2 varchar. Ekkor létrejön a cat nevű, egész szám típusú mező (ez lesz a kulcsmező, az adat1 nevű lebegőpontos-szám típusú mező és az adat2 nevű mező, melyben szöveges adatok tárolhatók.*

Fontos megjegyeznünk, hogy az új tábla csak valamely, az adott mapsetben levő vektoros térképhez csatoltan hozható létre, és csak *dbf* típusú lehet!

- **Remove table** (Tábla törlése) – A megadott vektoros térkép megadott rétegéhez csatolt adattáblát törli. Csak olyan tábla törölhető, amely az adott mapsetben levő vektoros térképhez csatolva van. Egy opciója van:
  - *Force removal (required for actual deletion of table)* – Erőltetett törlés. Ha tényleg törölni akarjuk a táblát, akkor kell engedélyezni, egyébként a törlés nem valósul meg.
- **Add columns** (Oszlop [mező] hozzáadása létező táblához) – A megadott vektoros térkép adott rétegéhez csatolt adattáblához adhatunk hozzá újabb oszlopot (mezőt). A mező hozzáadása megegyezik az új tábla hozzáadásánál leírtakkal.
- **Change values** (Értékek megváltoztatása) – Egy megadott vektoros térkép megadott rétegéhez csatolt attribútumtábla adatainak megváltoztatására szolgál. Értelemszerűen: a *Vector map to...* mezőben kell megadni a vektoros térkép nevét, a *Layer to which...* mezőben a réteg számát, a *Column to update* mezőben pedig annak az oszlopnak(mezőnek) a nevét, amelyet meg akarunk változtatni. Három módon járhatunk el.
  - Az egyik lehetőség, hogy a *value update to column with* mezőbe beírjuk azt az értéket, amit a mező rekordjaiba kívánunk bejegyezni. Ebben az esetben az adott oszlop összes rekordjába az itt megadott érték kerül. Ha a mező karakter típusú, akkor a beírt karakterláncot egyszeres idézőjelek között kell elhelyezni (például így: 'adat').
  - A másik lehetőség az, hogy ugyanazon tábla egy már meglévő mezőjének (oszlopának) adatait másoljuk be a megváltoztatandó oszlopba. Ekkor is a *Column to update* mezőben adjuk meg a változtatásra kijelölt oszlop nevét, de most a *Column to query* mezőben azt adjuk meg, hogy melyik, már meglévő oszlop adatait másolja át a GRASS. Arra is lehetőség van, hogy a máso-

lás úgy menjen végbe, hogy közben az eredeti adatokon valamilyen matematikai műveletet végzünk. Ha például a *Column to update* mezőbe ezt írjuk:

*adat2,*

a *Column to query*-be pedig ezt:

*adat1\*3*

akkor az *adat1* oszlop adatait a program úgy másolja át az *adat2* oszlopba, hogy közben megszorozza őket 3-mal.

- Harmadik lehetőségként úgy is eljárhatunk, hogy SQL kifejezést használunk. Ekkor a *Value to update...* sorban megadott értéket csak a *Column to update* sorban megadott oszlop azon rekordjai veszik fel, melyek kielégítik a *WHERE conditions...* sorban megadott SQL kifejezésben foglalt feltételeket.
- **Drop column** (Oszlop [mező] törlése) – Csak olyan adattáblából lehet oszlopot törölni, amelyik csatolva van valamelyik vektoros térképhez, pontosabban: annak egy rétegéhez. Ekkor a *Vector map...* mezőben kell megadni ennek a vektoros térképnek a nevét, a *Layer where...* mezőben pedig a réteg számát. Végül a *Name of the column* mezőben adjuk meg az adattáblából törlendő oszlop nevét is. A *Run* megnyomása után a művelet végbemegy.
- **Rename column** (Oszlop [mező] átnevezése)– Használata megegyezik a *Drop column*-ével. Különbség csak annyi, hogy az utolsó sorban (*Old name...*) meg kell adni az átnevezendő oszlop nevét, majd vesszővel elválasztva ugyanazt az oszlop új nevét, valahogy így: *regi\_nev,uj\_nev*. A vessző előtt- és után nem kell szóköz!
- **Test database** (Adatbázis tesztelése) – A *Connect to database* pontban megadott adatbázist teszteli több SQL-kéréssel. A tesztelés eredménye az *Output* ablakban tekinthető meg. Egyetlen paraméterként csak azt kell kiválasztani a legördülő menüből, hogy *test1* és a *Run* gombra kattintva indítható a művelet.

## 2.7. Exportálás és importálás a GRASS-ban

*Fontos! Tartsuk szem előtt, hogy egyes sajátosságok nem exportálhatók maradéktalanul, vagy egyáltalán nem exportálhatók. Az exportálás során ugyanis – hiszen pont erről szól – a GRASS-ban létrehozott tartalmat más, külső formátumba alakítva kíséreljük meg eltárolni (mondjuk geotiff állományba). Ha a választott állomány nem képes a kérdéses raszteres térkép minden tulajdonságának tárolására (mondjuk nem tud megfelelő számú szint tárolni), akkor az eredményül kapott fájl akár hibás is lehet, vagy az adatok egy része elveszhet az exportálás során. Egyes esetekben az exportálás nem is történik meg, a kimeneti állomány nem jön létre, hibüzenettel leáll a művelet. Fontos megértenünk, hogy ez nem azért történik, mert a GRASS elrontott valamit, hanem általában arról van szó, hogy a kérdéses térkép a megadott formátumba nem menthető, vagy csak adatvesztéssel teljesíthető a kérés.*

*További fontos sajátosság, hogy az exportálás csak azokat a területeket érinti, melyek az aktuális régióbeállítás szerint a számítási régióba esnek. Tehát nem azon elemeket exportálja a GRASS, melyek a*

*Map Display* ablakban éppen láthatóak, hanem amelyek a korábban megadott számítási régióon belül helyezkednek el (lásd a Régiók kezelése a GRASS-ban részt). A GRASS ezen tulajdonságának a figyelembevétele lényeges, mert könnyen előfordulhat, hogy elfeledkezve a korábbi beállításokról, exportálunk valamely állományt, de az eredmény nem egyezik meg a várakozásainkkal. Ilyenkor általában a régióbeállítás ellenőrzésével kell a hibakeresést kezdeni.

Az exportálás és az importálás ismerete nagyon fontos része a térinformatikai munkának, mert ez teszi lehetővé, hogy más programokkal is kommunikálhassunk, illetve más programokkal dolgozók adatait is használhassuk, ha szükséges. Ez azonban csak akkor lehetséges, ha a szükséges minimális mértékben ismerjük az importáló/exportáló alkalmazások használatát, valamint a használt fájlformátumok fontosabb sajátosságait.

A továbbiakban az importáló/exportáló alkalmazások használatáról lesz szó, de csak annyi, amennyi a használatuk megértéséhez minimálisan szükséges. Akinek további segítségre van szüksége, az az adott ablak alján található *Help* gomb megnyomásával kérhet segítséget. Ekkor a rendszer alapértelmezett böngészőjében olvashatja az adott alkalmazás használatának részletes, általában több oldalas, angol nyelvű leírását. Ebben nem csak azt találja meg, hogy melyik gomb, melyik mező mire való, hanem a program sajátosságai, esetleges ismert hibái is olvashatók.

Ugyanezért lehet ajánlatos a *Help*-ek böngészése a nagyobb tapasztalattal rendelkező felhasználóknak. Ha az exportálás vagy importálás során hibát tapasztalnak, vagy csak egyszerűen nem az történt, amit vártak, akkor célszerű lehet a *Help* megnyitása és a vonatkozó anyag áttanulmányozása. Sokszor derül ki, hogy tulajdonképpen nincs is semmiféle hiba, csak egyszerűen vagy a választott fájlformátum, vagy az export/import alkalmazás korlátai miatt nem az történik, amit a felhasználó szeretett volna. Az olvasottak felhasználásával a tapasztaltabbak ilyenkor esetleg találhatnak valamilyen kerülőutat, hogy mégiscsak elérjék a céljukat.

A különféle fájlformátumok sajátosságait nem fogjuk tárgyalni, mert az nem célja ennek a könyvnek, és a terjedelmet is igencsak megnövelné. Mit tegyen akkor az, aki nem ismeri még a fontosabb fájlformátumok legfontosabb sajátosságait sem? A válasz egyszerű: használja az internetet és olvasson után a témának. Ez persze így egy kicsit talán nyersen hangzik, de márt nem igazán lehet ajánlani.

A továbbiakban olvasható leírások is csak bemutatják, hogy egy-egy export/import ablak mezőit hogyan kell kitölteni és az egyes opciók mit jelentenek, de ezek jó része csak azoknak lesz érthető, akik ismerik az adott formátum sajátosságait. Például a *Vektoros állományok exportálása* fejezet *POV-Ray* részében az egyik opcióról ez olvasható:

„*Object modifier* – A *POV-Ray* eredeti dokumentációja által tartalmazott objektum-módosítók (*OBJECT\_MODIFIER*) alkalmazhatók, a megadott példa szerint.”

Nyilvánvaló, hogy a hétköznapi felhasználónak ez semmit sem mond, annak azonban, aki ismeri a *POV-Ray*-t, teljesen elegendő információt tartalmaz. Aki használ *POV-Ray*-t, (nincs belőlük túl sok) ugyanis tudja, hogy mi az az *object modifier*, aki pedig nem használja, annak meg úgyis mindegy. Viszont nagyon hosszadalmas lenne elmondani, hogy mi is az az *object modifier* és felesleges is.

Vagyis, ha valaki nem érti, hogy mire is való egyik vagy másik opció, az valószínűleg azért van, mert nem tudja, hogy az adott állománytípusnak milyen sajátosságai vannak. Ha azonban az adott fájl típust mégiscsak használnia kell, akkor ne legyen rest, olvasson utána. Ne feledjük: az interneten a számítógéppel kapcsolatos MINDEN információ elérhető.

Egy kivétel azonban lesz. Az ASCII állományok használatára (viszonylag) részletesen kitérünk.

## 2.7.1. ASCII állományok és a GRASS

Az ASCII állományok nagyon fontos sajátossága, hogy gyakorlatilag operációs rendszertől függetlenek, és az összes szövegfeldolgozó és táblázatkezelő program képes használni őket. Szerkesztésük, kezelése, használatuk rendkívül egyszerű. A fentiek miatt ideális átjáró formátumok a különféle rendszerek, programok és környezetek között. Az ASCII állományok használatát a GRASS messzemenően támogatja, mind export, mind import oldalon.

ASCII állományba raszteres és vektoros térképet is exportálhatunk, és természetesen ezeket importálni is lehet a GRASS-ba. De ASCII térképet nem csak térinformatikai alkalmazással lehet előállítani, hanem akár a legegyszerűbb szövegszerkesztővel vagy táblázatkezelővel is, és aztán ezek is importálhatóak a GRASS-ba, ahol teljes értékű térképekként kezelhetők.

### 2.7.1.1. Raszteres ASCII

*ASCII grid* (ASCII rács) állományban minden egyes rasztert egy-egy szám jelöl, mert a raszteres állományban minden raszterhez kell szakadatnak is tartoznia. Ha valamely raszter értéke nem értelmezett, akkor azt NULL-értékűnek mondjuk. Az egyes számokat – vagyis az egyes rasztereket – valamilyen mezőelválasztó karakter választja el egymástól, ez általában egyetlen *szóköz*, de egyes esetekben más is lehet.

Az állomány fejléccel kezdődik, mely az állomány georeferálását teszi lehetővé (figyeljünk a kisbetű-nagybetű különbségre!).

Lássunk egy egyszerű, ESRI ASCII grid típusú állományt!

```
ncols 11
nrows 11
xllcorner 0
yllcorner 0
cellsize 100
NODATA_value -9999
0 2 2 4 5 6 6 5 5 4 -9999
0 2 3 4 5 6 7 6 6 5 -9999
0 3 3 4 5 7 7 6 6 5 -9999
2 3 3 5 6 7 8 7 7 6 -9999
4 4 5 5 6 6 7 8 7 6 -9999
4 5 5 6 7 7 8 8 7 6 -9999
3 4 4 5 6 6 8 7 6 6 -9999
3 3 2 4 4 5 6 6 5 5 -9999
2 2 1 3 3 4 5 5 4 3 -9999
1 1 1 2 2 3 4 4 3 2 -9999
-9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999
```

Értelmezése a következő:

- **ncols** – a térkép *oszlopainak* a száma. Ez jelen esetben 11, vagyis egy sorban 11 szám kell, hogy legyen. Az is baj ha ennél több van és az is, ha kevesebb. Az egyes számokat egymástól egy darab szóközzel kell elválasztani.
- **nrows** – a térkép *sorainak* a száma. Ez jelen esetben 11, vagyis egy oszlopban 11 szám kell, hogy legyen. Az is baj ha ennél több van és az is, ha kevesebb. Az egyes sorokat sor vége karakterrel kell egymástól elválasztani. Erről még szó lesz, mert nem egyértelmű a dolog.
- **xllcorner** – a térkép bal alsó sarkának az  $x$  koordinátája, jelen esetben tehát a legelső oszlop első száma által reprezentált raszter bal alsó sarkának az  $x$  koordinátája.
- **yllcorner** – a térkép bal alsó sarkának az  $y$  koordinátája, jelen esetben tehát a legelső oszlop első száma által reprezentált raszter bal alsó sarkának az  $y$  koordinátája.
- **cellsize** – azt mutatja meg, hogy egy raszter a valóságban hányszor hány egység méterű. Jelen esetben ez 100, vagyis azt jelenti, hogy minden egyes szám egy darab  $100 \times 100$  egység méretű területet fed le. A raszterek folytonosan, tehát hézagmentesen és átfedés mentesen következnek egymás után. Azt, hogy az egységet milyen mértékegységben kell értelmezni, az adott location beállítása dönti el. Ha mondjuk, az adott location vetületi rendszere méterben mér, akkor a raszterméret  $100 \times 100$  m, vagyis egy hektár lesz.
- **NODATA\_value** – az ide írt szám a NULL értéket reprezentálja. Mivel a térkép csak számot tartalmazhat, ezért figyeljünk arra, hogy olyan érték reprezentálja a NULL értékű rasztereket, amely a térképen egyébként semmiképpen sem fordulhat elő. Itt jelen esetben értéke -9999. Ha a térképünk a Föld teljes domborzatát ábrázoló térkép lenne és méterben értelmeznénk az adatait, akkor ez nem lenne jó, mert -9999 m-nél mélyebb árkok is vannak. Ebben az esetben érdemes lenne mondjuk -99999-et választani NULL-értéknek.

Vagyis, ez a térkép olyan, hogy  $11 \times 11$  raszterre bontva ábrázol egy területet. Minden raszter  $100 \times 100$  egységnyi méretű, ami, ha az adott location vetületi rendszere méter alapú, azt jelenti, hogy egy raszter  $100 \times 100$  m kiterjedésű, a teljes térkép tehát  $1100 \text{ m} \times 1100 \text{ m}$ . A térkép bal alsó sarkának koordinátája  $x=0, y=0$ . Ebből, és a raszterméretből tudható bármelyik pont, illetve raszter koordinátája. Például a jobb felső sarok koordinátái:  $x=1100, y=1100$ . Ezen kívül az is látható, hogy a térkép alsó részéről és jobb oldaláról (vagyis a déli és a keleti szegélyéről) nem áll rendelkezésre adat, ezt jelzik a NULL-értékű raszterek.

A bemutatott állomány teljes értékű térkép, importálás után a GRASS-ban a szokásos módon használható. Ugyanilyen típusú állományt kapunk bármelyik raszteres térképünkből, ha a *File* menü *Export raster map* almenüjének *ESRI ASCII grid* pontjával exportáljuk. Érdemes kipróbálni!

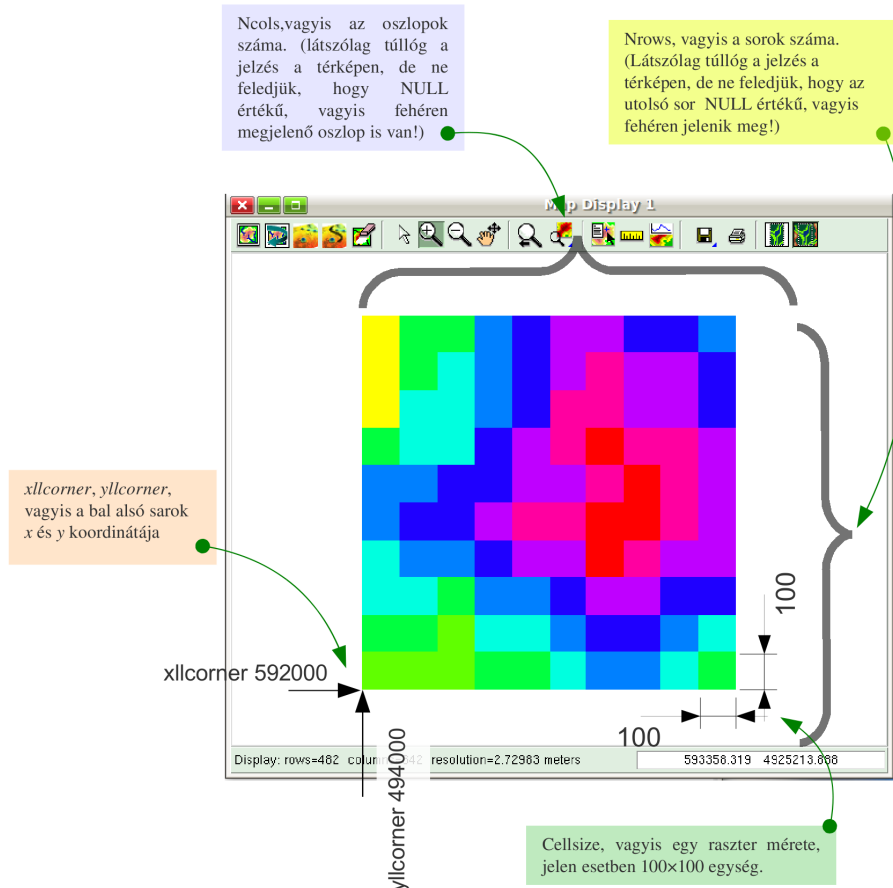
Az alábbi példa ugyanezt a térképet most a spearfish60 location területén jeleníti meg, mert most az *xllcorner* és az *yllcorner* értékei olyanok, hogy a location területére helyezik a térképünket:

```
ncols 11
nrows 11
xllcorner 592000
yllcorner 4924000
cellsize 100
NODATA_value -9999
0 2 2 4 5 6 6 5 5 4 -9999
```

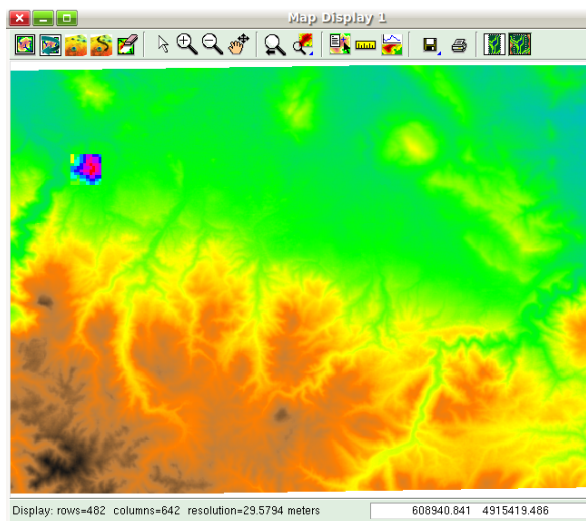
```

0 2 3 4 5 6 7 6 6 5 -9999
0 3 3 4 5 7 7 6 6 5 -9999
2 3 3 5 6 7 8 7 7 6 -9999
4 4 5 5 6 6 7 8 7 6 -9999
4 5 5 6 7 7 8 8 7 6 -9999
3 4 4 5 6 6 8 7 6 6 -9999
3 3 2 4 4 5 6 6 5 5 -9999
2 2 1 3 3 4 5 5 4 3 -9999
1 1 1 2 2 3 4 4 3 2 -9999
-9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999
    
```

Az eddigiek értelmezése a 2.22. ábrán tanulmányozható. Ez az ábra a bemutatott állományból készült. Ha ezt az állományt importálni akarjuk, akkor a *File* menü *Import Raster map* almenüjének *ESRI grid* pontját kell választanunk (használatát lásd lentebb a részletes leírásoknál). Végül az importált állomány megjeleníthető térképként is, mondjuk az *elevation.dem*-mel együtt. Jól látható, hogy a megadott koordináták a valóságban mely területet fedik le (2.23. ábra). Érdeemes összevetni a 2.22. ábrával!



**2.22. ábra**  
**ESRI ASCII grid fejléc adatainak értelmezése**



2.23. ábra

**A 2.22. ábra térképe az elevation.dem-mel együtt megjelenítve.  
A térkép ASCII forrását lásd a szövegben!**

Lássunk egy további példát a raszteres ASCII állományok használatára!

Készítsünk területhasználati térképet szabadon hozzáférhető műholdképek alapján (2.24. ábra)! A mintaterület jelen esetben Pécs egy része és nyugati szomszédsága legyen; a kiválasztott terület képe egyszerűen lementhető a Google Maps-ről. Érdekes rögtön lekérdezni a terület „bal alsó”, vagyis délnyugati sarkának a koordinátáit is és feljegyezni!

A munka menete egyszerű. Először a lementett képre, a képen levő aránymérték segítségével 1 km-es rácshálót illesztünk egy tetszőleges, a feladatra alkalmas rajzolóprogramban.

Ezután az előre meghatározott területhasználati típusok valamelyikébe sorolunk minden egyes 1x1 km-es rasztert.

A eredményül kapott számok úgy rendezzük el, hogy az eredeti rácshálónak megfelelően helyezkedjenek el a számok, úgy hogy minden számot egy-egy szóköz választ el a szomszédjától, a sort pedig *enter* zárja.

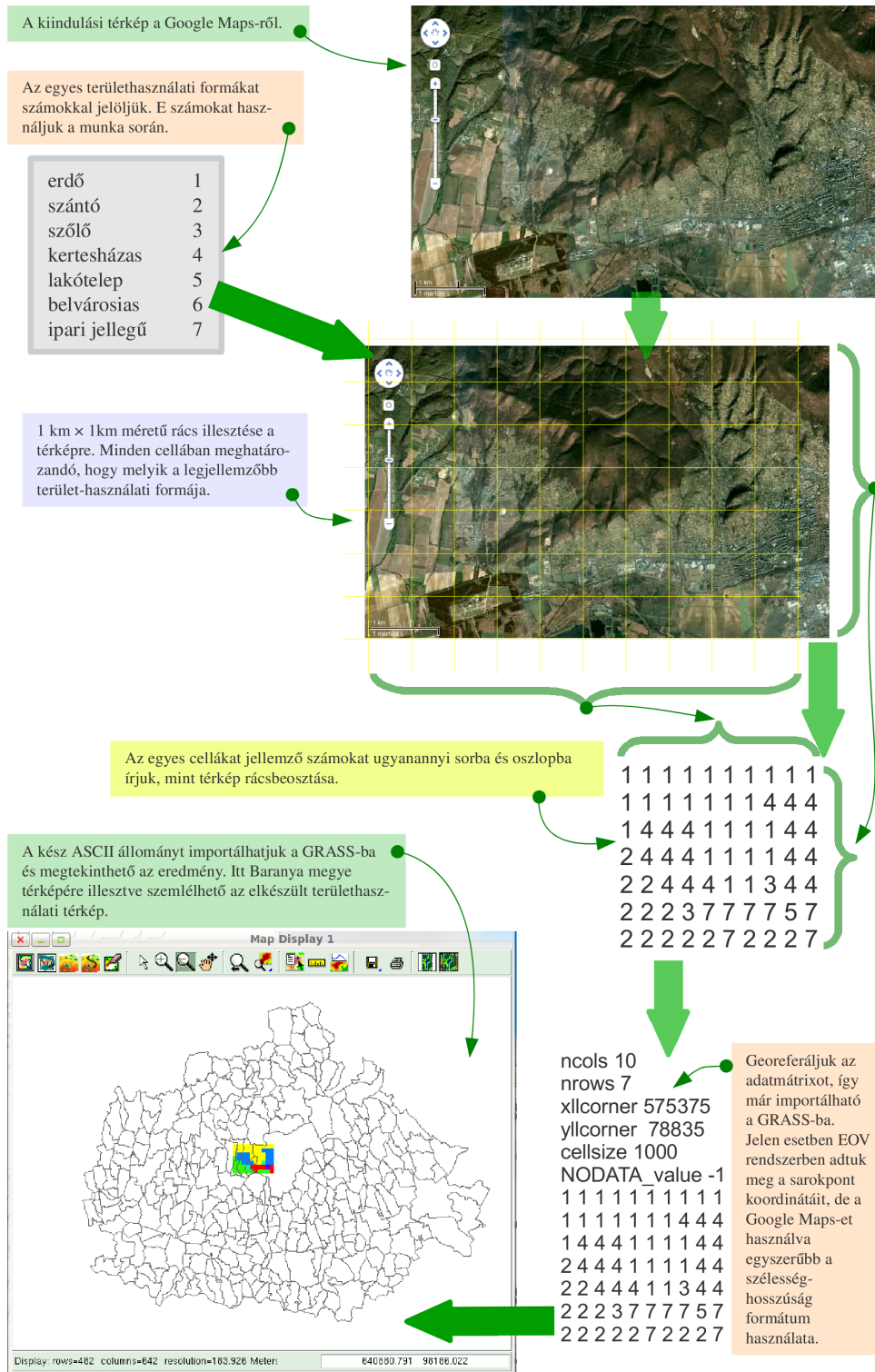
Ezután töltsük ki a fejléctet a fentebb leírtak szerint, ügyelve arra, hogy helyes értékeket adjunk meg, és ha tizedestörtet kell használnunk, akkor tizedespontot alkalmazzunk.

Végül az elkészült állományt importáljuk GRASS-ba.

A 2.24. ábrán látható a munka menete, az eredménnyel együtt. Érdekes lehet önállóan is kipróbálni!

Felmerülhet rögtön a kérdés, mire jó ez a módszer? Nos, előnye az egyszerűség: nem kell digitalizálóprogramot indítani, nem kell importálni a képet és georeferálni. Csak a szükséges adatokat importáljuk, ráadásul az ilyen jellegű munkák egyébként sem automatizálhatók minden esetben. Bár a példában műholdképet használtunk, ugyanez a követendő eljárás, ha mondjuk régi térképeket kell feldolgozni, vagy éppen nincs is térkép, csak valamifajta vázlat.

További előny, hogy az ilyen típusú adatokat táblázatkezelőben is használhatjuk és különféle műveleteket végezhetünk velük, ha szükséges.



2.24. ábra  
Egy példa raszteres ASCII állomány (ESRI ASCII grid) előállítására

### 2.7.1.2. Vektoros ASCII

ASCII vektoros állományt két változatban készíthetünk: *standard* és *point* módban (részletesebben lásd a *vektoros állományok exportálása* című részben). Mindkét állomány szabványos ASCII, csak a belső adat struktúrájuk tér el egymástól. A *standard*-ban pont, vonal és terület is tárolható, a *point* típusúban pedig csak különálló pontokat lehet megadni.

A **point** mód használata nagyon egyszerű. Minden pontot egy-egy sorban definiálunk, úgy, hogy megadjuk a pont egyik koordinátáját aztán a másikat, végül pedig a pont azonosító számát. Az egyes számokat egymástól valamilyen karakterrel választjuk el, például egy szóközzel. Hogy milyen sorrendben adjuk meg a koordinátákat, az tulajdonképpen mindegy, mert az állomány importálásánál kell megadnunk, hogy melyik oszlop melyik koordinátát tartalmazza.

Egy példa ASCII pont állományra:

```
591800 4919340 1
592800 4919340 2
591800 4920340 3
592800 4920340 4
```

Ez az állomány egy 1 km oldalhosszúságú négyzet négy sarokpontját tartalmazza.

Ennek az eljárásnak az ismerete különösen akkor hasznos, ha nagyobb mennyiségű ponttal kell dolgoznunk, és a koordinátáik adottak, de még nincsenek térképre rögzítve. Ilyenkor a mutatott módon gyorsan térkép készíthető belőlük. Az ASCII állományt a *File* menü *Import vector map* almenüjének *ASCII points or GRASS ASCII vector* pontjával végezhetjük (részletesen lásd lentebb).

A **standard** mód már összetettebb felépítésű állományt igényel. Ennek az állománynak már fejléce is lehet, mely az alábbiakat tartalmazhatja (de nem kötelező mindegyiket tartalmaznia):

ORGANIZATION – A vektoros digitális térképet kibocsátó szervezet neve.  
DIGIT DATE – A digitalizálás időpontja.  
DIGIT NAME – A digitalizálást végző neve.  
MAP NAME – Az eredeti térkép, melyről a digitalizálás készült, neve (címe).  
MAP DATE – Az eredeti térkép, melyről a digitalizálás készült, kibocsátási ideje.  
MAP SCALE – Az eredeti térkép, melyről a digitalizálás készült, méretaránya.  
OTHER INFO – Megjegyzések.  
ZONE – A zóna száma, ha a térkép vetületi rendszer zónákra osztott (pl: UTM).  
WEST EDGE – A térkép nyugati szélének kelet-nyugati koordinátája.  
EAST EDGE – A térkép keleti szélének kelet-nyugati koordinátája.  
SOUTH EDGE – A térkép déli szélének észak-déli koordinátája.  
NORTH EDGE – A térkép északi szélének észak-déli koordinátája.  
MAP THRESH – A térkép hibája.

E kulcsszavakat kettőspontnak kell követni, mely után tetszőleges mezőelválasztó (például egy tab) után adjuk meg a vonatkozó adatot.

Magukat az adatokat a VERTI: kulcsszó után kezdhetjük megadni, új sorban. Minden egyes objektumot (mondjuk egy utat vagy egy épületet) egy adattömbbel írunk le. Az adattömb elején meg kell adni, hogy milyen típusú objektumról van szó:

**P** – pont,

**L** – vonal,

**B** – határvonal,

**C** – centroid (súlypont),

**F** – arculat (3D felület),

**K** – mag (3D súlypont),

**A** – terület, megegyezik a B-vel, de használata nem ajánlott, csak a régebbi állományokkal való kompatibilitás miatt használható, helyette tehát B-t adjunk meg.

A típust jelző betű után adjuk meg, hogy az objektumot hány törésponttal írjuk le, vagyis, hogy az objektumot hány pont határozza meg. Mivel az objektum minden egyes pontját új sorban kell megadni, ezért egyúttal a az adattömb sorainak számát is jelenti. A következő szám a rétegekre utal, általában érdemes 1-et használni.

A következő sorokban írjuk le a kezdő- és végpontok, illetve az egyes töréspontok koordinátáit. Mivel pontokról van szó, minden sorba csak két szám kerül, az adott pont  $x$  és  $y$  koordinátája.

Ha az objektum összes töréspontját megadtuk, akkor az adattömböt azzal zárjuk, hogy megadjuk, melyik rétegen helyezkedik el ez az objektum és mi az azonosítója.

Ezután folytathatjuk a munkát a következő objektum adattömbjének beírásával.

A példa kedvéért készítsünk most térképet az előző példa négy pontjából!

VERTI:

B 5 1

591800 4919340

591800 4920340

592800 4920340

592800 4919340

591800 4919340

1 1

C 1 1

592300 4919840

1 1

$B$  azonosítóval indul az adattömb, ami azt jelenti, hogy a következő koordináták határvonalat jelentenek. Mivel a határvonalnak önmagába kell záródnia, ezért a kezdőpont és a végpont koordinátái megegyeznek, másképpen fogalmazva, a kezdőpontot meg kell ismételni a végén is. Ezért van az, hogy az előző példa négy pontja helyett itt öt pont van megadva, bár valójában tényleg csak négy koordinátapár szerepel. Elhagytuk az előző példa harmadik oszlopát, az azonosítókat is, mert most e négy pont együtt alkot egy objektumot, amelyet az adattömb végén elhelyezett azonosító jellemez. A koordináták sorrendje is változott, úgy, hogy a koordináták sorrendje a határvonal tényleges futását tükrözze, vagyis a koordináták úgy következnek egymás után, ahogyan a határvonalon haladva egymás után következnek. Az adattömböt lezáró sor azt mutatja, hogy a határvonal az 1-es rétegen van és az azonosítója is 1.

A határvonalon belüli területet súlypont (centroid) jellemzi. Ezt a súlypontot definiálja a következő, C-vel kezdődő adattömb.

A példa kedvéért álljon itt egy kissé összetettebb vektoros ASCII állomány is, térképi megjelenésével együtt (2.25. ábra). A háttérben a `spearfis60 location`-ból az `elevation.dem` térkép látható.

DIGIT DATE: 2010. 08. 30.

OTHER INFO: Példaállomány, a vektoros ASCII állományok megértéséhez.

VERTI:

B 5 1

591800 4919340

591800 4920340

592800 4920340

592800 4919340

591800 4919340

1 1

B 5 1

593800 4921340

592800 4920340

592800 4919340

593800 4918340

593800 4921340

1 2

C 1 1

592300 4919840

1 1

C 1 1

593300 4919840

1 2

L 10 1

593006 4917529

592539 4918276

591965 4918516

591391 4919090

591297 4919678

591431 4920572

591978 4920679

592365 4921133

593526 4922454

594381 4922281

1 3

L 5 1

591044 4920305

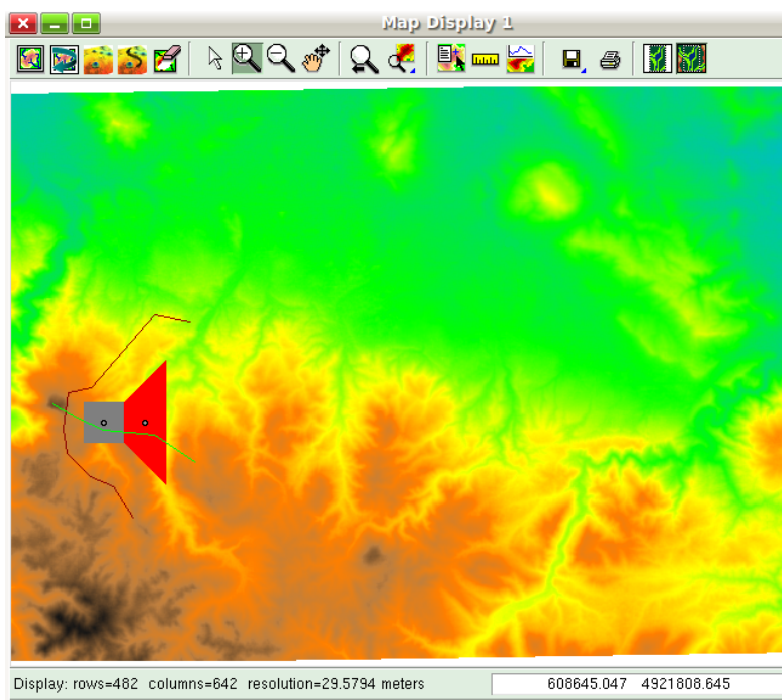
591698 4919905

592298 4919664

593540 4919518

594514 4918890

1 4



2.25. ábra

A szövegben levő vektoros ASCII példa térképi megjelenése,  
hátterben az elevation.dem-mel

## 2.7.2. Raszteres állományok exportálása

A *File* menü *Export raster map* almenüjének pontjai szolgálnak raszteres állományok exportálására és megfigyelhető, hogy meglehetősen sok lehetőségből választhatunk. A menü öt részre van osztva, ez megkönnyíti a választást.

### 2.7.2.1. Multiple formats using GDAL

Ezzel a ponttal általános térinformatikai formátumokba, valamint általános képformátumokba exportálhatunk. Erre a célra a GRASS a *Geospatial Data Abstraction Library* nevű fordító könyvtárat használja, erre utal a GDAL rövidítés a menüpontban. Ezzel az eszközzel az alább felsorolt formátumokba exportálhatunk.

AAIGrid: Arc/Info ASCII Grid  
ADRG: ARC Digitized Raster Graphics  
BMP: MS Windows Device Independent Bitmap  
BT: VTP .bt (Binary Terrain) 1.3 Format  
DTED: DTED Elevation Raster  
EHdr: ESRI .hdr Labelled  
ELAS: ELAS  
ENVI: ENVI .hdr Labelled  
ERS: ERMapper .ers Labelled  
FIT: FIT Image  
GIF: Graphics Interchange Format (.gif)  
GMT: GMT NetCDF Grid Format  
GSAG: Golden Software ASCII Grid (.grd)  
GSBG: Golden Software Binary Grid (.grd)  
GTiff: GeoTIFF  
HDF4Image: HDF4 Dataset  
HFA: Erdas Imagine Images (.img)  
IDA: Image Data and Analysis  
ILWIS: ILWIS Raster Map  
INGR: Intergraph Raster  
JPEG: JPEG JFIF  
JPEG2000: JPEG-2000 part 1 (ISO/IEC 15444-1)  
Leveller: Leveller heightfield  
MEM: In Memory Raster  
MFF: Vexcel MFF Raster  
MFF2: Vexcel MFF2 (HKV) Raster  
netCDF: Network Common Data Format  
NITF: National Imagery Transmission Format  
PAux: PCI .aux Labelled  
PCIDSK: PCIDSK Database File  
PCRaster: PCRaster Raster File

PNG: Portable Network Graphics  
 PNM: Portable Pixmap Format (netpbm)  
 RMF: Raster Matrix Format  
 RST: Idrisi Raster A.1  
 SRTMHGT: SRTMHGT File Format  
 Terragen: Terragen heightfield  
 USGSDEM: USGS Optional ASCII DEM (and CDED)  
 VRT: Virtual Raster  
 XPM: X11 PixMap Format

Mint a felsorolásból látható, meglehetősen sok (összesen 40) állományformátumból választhatunk. Használata egyszerű. A megjelenő ablakban a *Required* részben meg kell adni az exportálandó raszteres állomány nevét a *Name of Raster map (or group) to export* mezőben, valamint az exportálás eredményeképpen létrejövő állománynak kell nevet adni a *Name of output raster file* mezőben. A *Print* fülben kérhető a támogatott formátumok kilistázása (a lista, szokás szerint, az *Output* ablakban jelenik meg). Az *Options* részben adhatjuk meg az exportálás további beállításait.

A *GIS format to write* nevű legördülő menüből választható ki a használni kívánt formátum (lásd a listát), alatta pedig a *File type* menüből a használni kívánt fájlformátum. Ez utóbbi az alábbiak egyike lehet: *Byte*, *Int16*, *UInt16*, *Int32*, *UInt32*, *Float32*, *Float64*, *Cint16*, *Cint32*, *Cfloat32*, *Cfloat64*. Ezek mindegyike egy-egy adattípus, mely meghatározza, hogy mekkora számokat lehet tárolni az adott fájlban.

Adattípus	minimum	maximum
Byte	0	255
UInt16	0	65 535
Int16, CInt16	-32 768	32 767
UInt32	0	4 294 967 295
Int32, CInt32	-2 147 483 648	2 147 483 647
Float32, Cfloat32	$-3,4 \cdot 10^{38}$	$3,4 \cdot 10^{38}$
Float64, Cfloat64	$-1,79 \cdot 10^{308}$	$1,79 \cdot 10^{308}$

A továbbiakban három, opcionális, vagyis nem kötelezően kiteltendő mezőt találunk.

A *Creation option...* nevet viselőben arra vonatkozóan adhatunk utasításokat, hogy az exportálás során az állományt létrehozó meghajtóprogram miként járjon el. Ha például úgy szeretnénk egy állományt exportálni, hogy az *ESRI tfw* (world fájl) szabványnak megfeleljen, akkor ide azt kell írunk, hogy *TFW=YES*. Ebben az esetben létrejön a megadott néven egy tiff állomány, valamint egy ugyanolyan nevű, de *tfw* kiterjesztésű másik fájl is. Ez utóbbi fogja tartalmazni a kép értelmezéséhez szükséges meta-adatokat. Ha több opciót kell megadnunk, akkor azokat vesszővel elválasztva tehetjük meg, de a vesszők előtt- és után nem kell szóközt tenni. Vagyis, ha mondjuk az előző példát szeretnénk azzal is kiegészíteni, hogy az exportálás LZW tömörítéssel történjen, akkor ezt kell ide írni:

*TFW=YES,COMPRESS=LZW*

A *Metadata keys...* mezőben azt lehet megadni, hogy milyen meta-tag-hez milyen érték tartozzon. Ennek általános formája:

*META-TAG=VALUE*

Ebben az esetben is igaz, hogy ha több opciót akarunk megadni, akkor azokat vesszővel elválasztva tehetjük meg, de a vesszők előtt- és után nem kell szóközt tenni.

Az utolsó mezőben azt adhatjuk meg, hogy milyen érték jelölje a kimeneti állományban azokat a rasztereket, amelyek a GRASS-ban NULL-értékűek voltak. Ennek az az értelme, hogy ha a kimeneti állomány olyan, hogy nem kezeli a NULL-értéket, akkor egy alkalmasan megválasztott értékkel (mondjuk: -9999) jelezhessük, hogy a vonatkozó cella nem értelmezett értékű.

Az ablakban van egy olyan opció is, melyet engedélyezve a GRASS nem exportálja a hosszú színtáblázatokat (*Do not export long colortable*). Ez arra való, hogy az exportálás során a lehető legnagyobb kompatibilitást biztosíthassuk más rendszerek felé. Általában nem kell bekapcsolni, de egyes esetekben – ha az exportálás során hibát észlelünk – hasznos lehet az engedélyezése.

### 2.7.2.2. További raszteres exportálási lehetőségek

#### ASCII grid

A *Name of an existing raster map* mezőben megadott raszteres térképet exportálja a *Name for output ASCII grid map* mezőben megadott állományba. Az exportálás eredménye szabványos ASCII karaktereket tartalmazó – és ezért elvileg bármely számítógépen feldolgozható – állomány. Ha a kimeneti állomány neve helyett „-” karaktert írunk, akkor az exportálás eredménye nem fájlba íródik, hanem az Output ablakba írja ki a GRASS (pontosabban a szabványos kimenetre).

Ha pontosan meg kell szabnunk az egy-egy rasztert leíró, az adott raszterre vonatkozó szakadatot hordozó szignifikáns számjegyek számát, akkor azt a *Number of significant digits* mezőben tehetjük meg. Ez az opció természetesen csak lebegőpontos számokat tartalmazó raszteres állomány exportálásakor érhető el, hiszen egész számokra vonatkozóan nem lenne értelmezhető a használata.

A *Number of values printed...* mező kitöltésének akkor van értelme, ha előbb engedélyezzük a *Write SURFER (Golden Software) ASCII grid* (írás Surfer ASCII formátumba) vagy a *Write MODFLOW (USGS) ASCII array* (írás MODFLOW ASCII tömbbe) opciókat a megfelelő választónégyzetre való kattintással. Ekkor a mezőben azt adhatjuk meg, hogy legfeljebb milyen hosszú számokat tartalmazzon a kimenet.

Az utolsó mezőben azt adhatjuk meg, hogy milyen karakter vagy karaktrelánc reprezentálja a NULL-értékeket a kimeneti állományban. Alapértelmezésben ez a csillag (\*) karakter, de ez szabadon felülbírálnak. Az ablak opciói:

- **Suppres printing of header information** – tiltjuk a fejléc létrehozását a kimeneti állományban, vagyis nem jelennek meg a térképben foglalt adatok helyére vonatkozó információk, csak maguk a térképi adatok.
- **Write SURFER (Golden Software) ASCII grid** – A kimeneti ASCII állomány a Surfer számára értelmezhető szerkezetben készül el. Ekkor a NULL értékeket a kimeneti állományban a 1.70141e+038 szám reprezentálja (1,70141·10<sup>38</sup>).
- **Write MODFLOW (USGS) ASCII array** – A kimeneti ASCII állomány MODFLOW ASCII tömb típusú lesz.

- **Force output of integer values** – Csak egész számokat fog tartalmazni a kimeneti ASCII állomány, így engedélyezésekor az exportálás során a lebegőpontos számábrázolást használó típusú térképekből egész-szám típusú ASCII állományok lesznek.

### ASCII x, y, z

Egyszerű ASCII tömbbe írja ki a térkép adatait. Ebben az állományban minden egyes raszter egyetlen sorban lesz leírva, úgy, hogy a raszter *x*, *y*, *z* koordinátái és szakadata egymás után következnek, a *Filed separator* (mezőelválasztó) mezőben megadott karakterrel elválasztva. Az első mezőben az exportálandó állomány nevét, a másodikban pedig a kimeneti állomány nevét kell megadni. A *Filed separator* alapértelmezésben a „|” karakter, de ez szabadon felülbírálható. Példaként álljon itt az *elevation.dem* állomány eleje ezen exportálás után:

```
601205.918815125|4927981.73419182|1110
601305.911483535|4927981.73419182|1113
601405.904151945|4927981.73419182|1111
601505.896820355|4927981.73419182|1110
```

### ESRI ASCII grid

Használata hasonló az előző pontéhoz. Az első mezőben az exportálandó állományt, a másodikban pedig a kimeneti állományt kell megadni. A *Number of decimal places* mezőben adjuk meg figyelembe veendő tizedesértékek számát, ennek természetesen csak lebegőpontos számábrázolású térképek esetében van értelme.

Két fontos opciója van.

- **Suppress printing header information** – bekapcsolásával tiltjuk a fejléc létrehozását a kimeneti állományban, vagyis nem jelennek meg a térképben foglalt adatok helyére vonatkozó információk, csak maguk a térképi szakadatok.
- **List one entry per line instead of full row** – engedélyezésével a kimeneti állományban az egyes raszterekre vonatkozó szakadatok úgy jelennek meg, hogy soronként csak egy szakadat lesz található. Egyébként (és ez az alapértelmezés) minden sorban annyi szakadat lesz, amennyi raszter van. Vagyis, ha mondjuk az exportálandó raszteres állományuk 400 sorból és 600 oszlopból áll, akkor exportálás után alapértelmezésben 400 sorban lesznek a szakadatok és soronként, szóközzel elválasztva egymástól, 600 adatot fogunk találni, bekezdés jel után pedig a következő sor leírása kezdődik. Ezen opció bekapcsolásával viszont 240 000 sorban lesz a 240 000 raszter szakadata és mindegyik sort bekezdés jel zár le.

### GRIDATB.FOR

Csak a bemeneti és a kiemeneti állomány nevét kell megadni, a *Run*-ra kattintás után pedig lezajlik az exportálás.

### **MAT-File (v.4)**

Használata megegyezik a GRIDATB.FOR-éval.

### **Binary**

Bináris tömbként exportálja a *Name of output...* mezőben megadott néven megadott állományba a *Name of input raster map* mezőben megadott állomány tartalmát. A *Value to write out for null* mezőben az adható meg, hogy a NULL értékek helyett az exportálásnál milyen érték szerepeljen. Figyeljünk arra, hogy ide csak szám kerülhet.

A választható opciók:

- **Output integer category values not cell values** – A kimenet a (egész szám típusú) kategória-értékeket fogja tartalmazni, nem pedig a raszteres szakadatokat.
- **Export array with GMT compatible header** – A raszteres állományt (tömböt) GMT kompatibilis fejléccel ellátva exportálja.
- **Generate BIL world and header files** – Az exportálás során BIL típusú fejléceket és világleíró állományt készít.
- **Byte swap output** – A kimenet a bájtokat fordított sorrendben fogja tartalmazni.

### **MPEG-1**

Mozgóképpé fűzi össze a megadott állományokat.

Fontos sajátossága, hogy egyszerre legfeljebb négy nézetet (view) képes kezelni. A nézet azt jelenti, hogy az eredményül kapott filmen egyszerre, egymástól fekete vonallal elválasztva, annyi film látható, ahány nézetet definiáltunk. Ha csak egy nézetet akarunk használni, akkor csak a *Raster file(s) for view 1* mezőt töltjük ki. Ha több nézetre is szükség van, akkor töltjük ki másik nézet mezőt vagy mezőket is! Természetesen az egyes nézet mezőkben nem kell ugyanazokat a térképeket megadni. Ha mondjuk a *Raster file(s) for view 2*-t is kitöltjük, akkor a film olyan lesz, hogy a bal oldalon és a jobb oldalon egyszerre fut egy-egy film. Mondjuk az egyik oldalon a csapadékviszonyok időbeli változása látható, a másik oldalon pedig ugyanazon a területen és ugyanabban az időben a növényzeté.

- **Quiet – suppress progress report** – A munka során nem küld üzeneteket az *Output* ablakba.
- **Convert on the fly, use less disk space** – Ennek az opciónak a bekapcsolása azt eredményezi, hogy kevesebb helyet foglal a *művelet közben* a GRASS. Figyelni kell azonban arra, hogy csak akkor használható, ha egy csupán egy nézetet (view) kérünk! A mai, nagy merevlemezzel szerelt számítógépeken nincs különösebb jelentősége.
- *Raster file(s) for view 1* – Azoknak a raszteres térképeknek a listája, amelyeket filmmé szeretnénk összefűzni. A fájlnevek megadhatók egyenként is, de ennek nincs sok értelme, hiszen egy-két térkép egymás utáni megjelenése még nem igazán tekinthető filmnek. Éppen ezért lehetőség van arra, hogy a megszokott helyettesítő karaktereket alkalmazzuk a fájlnevek megadásakor. Ha például azt szeretnénk, hogy az összes meglévő térképet belevegye a filmbe, akkor írjuk be azt, hogy \* (csillag). Ha mondjuk, csak azokat a térképeket akarjuk használni, melyek neve úgy kez-

dődik, hogy elev, akkor írjuk ezt: elev\*. További tippeket találunk a *Help* gomb megnyomása után.

- *Name for output file* – A kimeneti állomány neve.
- *Quality factor* – A kimeneti állomány minősége, 1-től 5-ig állítható. 1 a legjobb, de legnagyobb fájlméretet produkáló, 5 a legrosszabb, de a fájlméret is sokkal kisebb lesz.

### PNG

Png (portable network graphics) formátumba exportál. Ha szeretnénk, hogy a létrejövő állománynak legyen kiterjesztése is, akkor azt a névvel együtt meg kell adni! Ha a kiterjesztést nem adjuk meg, akkor a GRASS kiterjesztés nélkül hozza létre az állományt, de ez a tartalmát természetesen nem befolyásolja.

### PPM

Ppm formátumba exportál (a ppm állományokról részletesebben lásd: <http://netpbm.sourceforge.net/doc/ppm.html> vagy [http://en.wikipedia.org/wiki/Netpbm\\_format](http://en.wikipedia.org/wiki/Netpbm_format)) Ha szeretnénk, hogy a létrejövő állománynak legyen kiterjesztése is, akkor azt a névvel együtt meg kell adni, a png-nél olvasottakhoz hasonlóan! Egyetlen opciója van:

- **Output greyscale instead of color** – A kimenet mindenképpen szürkeskálás legyen, akkor is, ha az exportálandó térkép színes.

### PPM for RGB

Három, külön-külön *szürkeskálás* raszteres térképet egyesít egyetlen, színes ppm formátumú képpé. A három kép egyike legyen a vörös színkivonat (red), a másik a zöld (green), a harmadik pedig a kék (blue). A kimeneti állomány nevét a *Name for new PPM file* mezőben adjuk meg. Opciója:

- **Add comments to describe the region** – A régiót leíró kommentet fűz a kimeneti állományhoz.

### POV-Ray

A POV-Ray számára értelmezhető magasságmező-állományt készít. (A POV-Ray sugrákövetéses eljárással működő grafikai környezet.) Ebből rögtön következik, hogy bár akármilyen térképet exportálhatunk, de csak olyat van értelme, amelyet aztán mint domborzati térképet használunk a POV-Ray-ben. Ha szeretnénk, hogy a létrejövő állománynak legyen kiterjesztése is, akkor azt a névvel együtt meg kell adni!

Az exportálandó raszteres térkép nevét adjuk meg a *Name of an existing raster map mezőben*, a kimeneti, tga formátumú, állomány nevét pedig a *Name of output povray file (TGA height field file)* mezőben. A *Height-field type* (0=actual heights 1=normalized) beviteli mezőben azt adhatjuk meg, hogy a létrejövő fájl hogyan tartalmazza a magasságokat: 0=a valós magasságok, 1=normalizált értékek. Ha szükséges, akkor az *Elevation bias* (magassági korrekció) mezőben megadhatjuk azt az értéket is, melyet a GRASS hozzáad a kimeneti állományhoz. Ha például azt akarjuk, hogy 100 m-rel alacsonyabbak legyenek az egyes raszterek, akkor ide írjuk azt, hogy „-100”. Látható tehát, hogy negatív értéket is adhatunk, de egyébként tizedestörtet is használhatunk. Végül a *Vertical scaling factor* mezőben azt adhatjuk meg, hogy

végezzen-e pozitív vagy negatív túlmagasítást a exportáló és ha igen, mennyit. Ha ide, mondjuk, azt írjuk, hogy 0.5, akkor az eredeti magasságokat felére csökkenti a GRASS az exportálás során.

### TIFF

Tiff állományt hoz létre a *Name for new tiff file* mezőben megadott néven, a *Name of input raster map* mezőben megadott állományból. Megadható a tömörítés típusa is a *TIFF file compression* legördülő menüből. Ha szeretnénk, hogy a létrejövő állománynak legyen kiterjesztése is, akkor azt a névvel együtt meg kell adni! A választható opciók az alábbiak:

- **TIFF Palette output (8 bit instead of 24 bit)** – A kimenet 8 bites (legfeljebb 256 színt tartalmazó) tiff állomány lesz, nem pedig az alapértelmezett 24 bites (vagyis legfeljebb 16 777 216 színű).
- **Output TIFF world file** – A tiff mellet létrehoz egy *tfw* állományt is, mely az exportált terület határait és felbontását is tartalmazza, így az eredményül kapott tiff fájlt georeferáltként lehet kezelni.
- **Output Tiled TIFF** – A bementi állományt úgy exportálja, hogy a kimenet 128×128 raszter méretű, különálló tiff képekből fog állni.

### VRML

A VRML fejlett virtualizációt lehetővé tevő nyelv, melynek használatát a GRASS ezen az exportálón keresztül támogatja. Fontos azonban, hogy csak domborzati térkép exportálásának van értelme, mert az elkészült VRML állományt, mint domborzatot használja majd a modellező.

- *Name of input elevation map* – az exportálandó térkép nevét kell megadni.
- *Name of color file* – annak a térképnek a neve, mely a domborzati térkép színeit adja meg. Ez lehet maga a domborzati térkép is, de bármelyik másik, ugyanazt a területet lefedő raszteres térkép is. Hasonló ez az NVIZ-nél alkalmazott megoldáshoz, lásd ott.
- *Vertical exaggeration* – a függőleges túlmagasítás értéke. 0-nál nagyobb lebegőpontos szám (tizedes tört) lehet.
- *Name for new VRML file* – a kimenet eredménye, az új VRML fájl neve.

### VTK

A VTK struktúrájú ASCII állományt készítő exportáló felület számos elemet tartalmaz. Például eleve kétféleképpen adható meg az exportálandó állomány is. Mivel VTK állományokat alighanem csak az használ, akinek ilyesmre szüksége van és ért is hozzá, az *Advanced options (bővített opciók)* részt nem tárgyaljuk. Az itt megtehető beállítások és megadható opciók az általános térinformatika-felhasználó számára úgysem mondanak sokat, aki pedig járatos a VTK formátumban, úgyis tudja, hogy mi mit jelent és mire van szüksége.

- *Name of input raster map(s)* – Az exportálandó raszteres térkép neve. Ha több térképet is exportálni szeretnénk, akkor a neveket vesszővel elválasztva kell megadni. Ha terepmodellt akarunk exportálni, akkor érdemes lehet az *Elevation raster map* mezőt használni.
- *Name for VTK-ASCII output file* – A kimeneti állomány neve.

- *Elevation raster map* – Ha az exportálandó raszteres térképet, mint terepmodellt kívánjuk használni, akkor érdemes itt megadni a nevét, nem pedig a *Name of input raster map*-ban.
- *Value to represent no data cell* – Adjuk meg azt a számot, mely a NULL-értékeket jelenti majd az exportált fájlban (ha szükséges)
- *Elevation (if no elevation map is specified)* – Magassági érték rendelhető a raszteres térképhez, ha az *Elevation raster map* mezőben nem adtunk meg semmit.
- *Create VTK point data instead of VTK cell data (if no elevation map is given)* – VTK pont adat típusú lesz a kimenet, nem pedig az alapértelmezett VTK raszteres formátum.

### 2.7.3. Vektoros állományok exportálása

A vektoros állományok exportálása alapvetően a raszteres exportálásnál leírtak szerint történik, a két állománytípus különbségei miatt, természetesen számos különbséggel. Az *Export vector map* menüi az alábbiak:

- **Multiple formats using OGR** – Különböztet, az OGR által ismert, vektoros formátumokba exportál. Az OGR a GDAL (*Geospatial Data Abstraction Library*) része, szó volt róla a raszteres exportálás kapcsán.
- **ASCII points or GRASS ASCII vector** – Valsztható, hogy pontokat exportáljon egyszerű ASCII állományba, vagy ASCII formátumú, de a GRASS saját struktúrája szerinti állományba. Ez utóbbi esetben vonal és terület típusú objektumokat is képes exportálni.
- **DXF** – DXF-be exportál. Ez a CAD rendszerek által elterjedten használt fájlformátum.
- **Multiple GPS format using GPSBabel** – A GPSBabel nevű, GPS konvertáló program használatával általános GPS formátumokba exportál.
- **POV-Ray** – A POV-Ray világleíró nyelvére fordítja az exportálandó állományt.
- **SVG** – *Scalable Vector Graphics*, általános, programfüggetlen vektoros rajzi állomány.
- **VTK** – VTK struktúrájú ASCII állományba exportál.

#### 2.7.3.1. Multiple formats using OGR

Az alábbi vektoros formátumokba exportálhatunk a segítségével:

ESRI Shapefile  
Mapinfo\_file  
TIGER  
S57  
DGN  
Memory  
BNA

CSV  
GML  
GPX  
KML  
GeoJSON  
Interlis\_1  
Interlis\_2  
GMT  
SQLite  
ODBC  
PostgreSQL  
MySQL  
DXF  
Geoconcept  
GeoRSS  
GPSTrackMaker

Az exportálás első lépéseként adjuk meg az exportálandó vektoros térkép nevét az *Options* fülben levő *Name of input vector map* mezőben, aztán a kimeneti állomány nevét az *OGR output datasource name* mezőben.

Az *Input* fülben válasszuk ki, hogy a vektoros térkép mely összetevőit (pont, vonal stb.) és melyik rétegét szeretnénk exportálni. Ez utóbbit a *Layer number* mezőbe kell beírni.

A *Creation* fülben állíthatjuk be az új állomány létrehozásának további paramétereit.

- *OGR Layer name. If not specified input name is used* – Megadható, hogy mi legyen a neve a kimeneti állományban annak a rétegnek, amelyiket az *Input* menüben adtunk meg, mint exportálandót. Ha ide nem írunk semmit, akkor az eredeti nevével exportálja a réteget.
- *OGR format* – Itt kell kiválasztanunk a létrehozandó állomány formátumát.
- *OGR dataset creation option (format specific, NAME=VALUE)* – Az új állomány elkészítésére vonatkozó opciókat adhatunk meg, az adott formátum saját opciói közül választva. A megadás módja az, hogy megadjuk a használandó opció nevét, majd egyenlőségjel után az értékét is. Ha például kml (a Google Earth formátuma) formátumba exportálunk, akkor ha ebben a mezőben az erre a formátumra vonatkozó opciót adhatunk meg. Mondjuk ezt:

*AltitudeMode=absolute*

- *OGR layer creation option (format specific, NAME=VALUE)* – Hasonló az előzőhöz, de itt a rétegre vonatkozó opciókat adhatunk meg. Ha például ESRI shapefile-ba exportálunk, akkor itt lehet olyat mondani, hogy

*SHPT=ARCZ*

Ennek hatására a GRASS-ban levő 3D vonalakat (*face*), 3D vonalakként exportálja (ilyenek például egy épület tömbjének az élei)

Az ablakban az *Options* fülben három opciót találunk:

- **Export features with category (labeled only). Otherwise all features are exported** – Ha engedélyezzük, akkor az elemeket úgy exportálja, hogy csak a kategória adatok maradnak meg. Egyébként mindent exportál. Fontos tudnunk, hogy ha olyan, terület típusú alakzatokat akarunk exportálni, melyekben lyuk van, és szeretnénk, hogy az exportált állomány is ugyanígy tartalmazza ezeket a területeket, akkor ezt az opciót engedélyeznünk kell.
- **Use ESRI-style .prj file format (applies to Shapefile output only)** – Csak shapefile-ba exportáláskor használható opció. Hatására elkészül egy ESRI szabványú .prj állomány is, mely a vetületi információkat tartalmazza.
- **Export lines as poligons** – A vonalakat nem vonalakként, hanem poligonként exportálja.

### 2.7.3.2. ASCII points or GRASS ASCII vector

A *Name of input vector map* mezőben megadott térképet exportálja a *Path to resulting ASCII file or ASCII vector name if '-o' is defined* mezőben megadott állományba. Ha az állomány nem létezik, akkor létrehozza. Két kimeneti formátum közül kell választanunk:

- *point* – csak pontokat exportál. Ha egyéb elemeket is tartalmaz a térkép, azokat figyelmen kívül hagyja.
- *Standard* – a térkép minden elemét exportálja.

Egyetlen opciója van:

- **Create old (version 4) ASCII file** – A GRASS 4 verziója szerinti ASCII állományt készítését engedélyezi.

Ha *point* kimeneti formátumot választottunk, akkor a *Points* fülben állíthatjuk be a rá vonatkozó paramétereket.

- *A Filed separator* mezőben adjuk meg, hogy az eredményül kapott ASCII állományban mi legyen a mezőelválasztó karakter, alapértelmezése a „|” (függőleges vonal), de bármilyen más karakterlánc megadható.
- *A Number of significant digits (floating point only)* mezőben azt adjuk meg, hogy hány tizedes pontosságig exportáljon adatokat a GRASS a készülő ASCII fájlba. Legkisebb értéke 0, a leghosszabb szám pedig 32 tizedesjegy hosszú lehet. Csak lebegőpontos térképek esetében van értelme, egész szám típusú adatok esetében nem használható.
- *Name of attribute column(s) to be exported (point mode)* – itt adható meg, hogy a térképhez csatolt adattábla mely mezőit (oszlopait) exportálja a pontokkal együtt a program.

Egyetlen opciója van:

- **Only export points falling within current 3D region (points mode)** – Engedélyezése esetén csak azokat a pontokat exportálja, melyek az aktuális 3D régió belül vannak. (Alapértelmezésben az aktuális 3D régió felületi kiterjedése megegyezik a 2D régióéval.)

A *Selection* fül mind a *point*, mind a *standard* formátum választásakor használható.

- *Layer number* – Itt adható meg, hogy az eredeti vektoros térkép melyik rétegét exportálja a GRASS
- *WHERE conditions of SQL statement without 'where' keyword* – Megadható, hogy csak azokat az elemeket exportálja a GRASS, amelyek egy megadott SQL kifejezésnek megfelelnek. Ekkor az SQL kifejezést úgy kell megadni, hogy csak a „where” utáni részt írjuk be.

### 2.7.3.3. DXF

Csak az exportálandó vektoros térkép nevét kell megadni a *Name of input vector map* mezőben és a kimeneti állomány elérési útját és nevét a *DXF output file* mezőben. Ha nem adunk elérési utat, csak fájlnévet, akkor a fájl a saját home könyvtárunkban jön létre.

### 2.7.3.4. Multiple GPS format using GPSTools

GPS vevőből nem csak adatok importálhatók, hanem arra is szükség lehet, hogy oda adatokat exportáljunk. Erre szolgál ez a pont, illetve a *v.out.gpsbabel* nevű program, melynek ez az ablak a grafikus felülete.

#### Options

- **Export as waypoints** – A térkép exportálása útpontokként.
- **Export as routes** – Exportálás mint út.
- **Export as tracks** – Exportálás, mint nyomvonal.
- *Name of input vector map* – Az exportálandó térkép neve.
- *Feature type(s)* – Az exportálandó objektumtípusok kiválasztása. Amelyik objektumtípusok választónégyzete aktív, azok lesznek exportálva.
- *Name of output file or GPS device* – A kimeneti állomány neve. Lehetőség van arra is, hogy ne egyszerűen fájlba mentjük a térképet, hanem rögtön a GPS eszközre töltjük fel. Ekkor annak az eszközt magát kell megadni. Ez Linux és Unix rendszerek alatt jellemzően ugyancsak fájlként jelenik meg, mégpedig alapesetben */dev/gps* néven, de ettől eltérő is lehet. Mivel az eszközök közvetlen kezelése igényel némi háttérismeretet, ez az eljárást csak azoknak javasolható, akik rendelkeznek a szükséges felkészültséggel. Egyébként érdemes csak egyszerűen egy fájlnévet megadni.
- *GpsBabel supported output format* – A kimeneti állomány formátuma. Csak a *GpsBabel* által támogatott formátum választható. Alapértelmezése: *gpx*.

#### Subset

- *Layer number* – Az exportálandó réteg száma
- *WHERE conditions of SQL statement without 'where' keyword* – Csak az itt megadott, SQL nyelven megfogalmazott feltételt kielégítő objektumok lesznek exportálva.

### 2.7.3.5. POV-Ray

Alapvetően különbözik a raszteres exportálásnál látott POV-Ray exportálótól.

- *Name of input vector map* – Az exportálandó vektoros térkép neve.
- *Feature type* – Az exportálandó elemek kijelölése.
- *Output file* – A kimeneti állomány neve.
- *Radius of sphere for points and tube for lines. May be also variable, e.g. grass\_r* – A pontokat a POV-Ray számára reprezentáló gömb sugara, illetve a vonalat jelentő cső keresztmetszetének sugara.
- *Modifier for z coordinates, this string is appended to each z coordinate* – A magassági (z) koordináta módosítható a kimeneti állományban, a megadott példák szerint.
- *Object modifier* – A POV-Ray eredeti dokumentációja által tartalmazott objektum-módosítók (OBJECT\_MODIFIER) alkalmazhatók, a megadott példa szerint.

### 2.7.3.6. SVG

*Scalable vector graphics*, általános vektoros rajzformátum, a legtöbb vektoros rajzoló program írja és olvassa ezt a formátumot, ennek megfelelően nagyon alkalmas arra, hogy adatokat cseréljünk a térinformatikai program és egyéb alkalmazások között.

- *Name of input map* – Az exportálandó térkép neve.
- *Name for SVG output file* – A kimeneti SVG állomány neve.
- *Output type* – A kimeneti állomány típusa. Azt adja meg, hogy hogyan exportálja az egyes alakzatokat a GRASS
  - *poly* – poligon
  - *line* – vonal
  - *point* – pont
- *Coordinate precision* – A koordináták pontossága az exportált állományban. Itt csak egész szám adható meg.
- *Attribute(s) to include in output SVG* – Azt adhatjuk meg, hogy az eredeti vektoros térképhez csatolt adattábla mely mezőinek adatait csatolja az egyes rajzi elemekhez az SVG létrehozásakor.
- *Layer number* – Az exportálandó réteg száma.

### 2.7.3.7. VTK

VTK szerkezetű ASCII állományt készít.

- *Name of input vector map* – Az exportálandó állomány neve.
- *Path to resulting VTK file* – Akimeneti VTK fájl elérési útja és neve. Ha csak a név van megadva, akkor a fájl a felhasználó home könyvtárában jön létre.
- *Feature type* – Az exportálandó elemtípusok kijelölése
- *Number of significant digits (floating points only)* – A szignifikáns számjegyek száma, vagyis tulajdonképpen a tizedestört hossza. Csak egész szám lehet és csak lebegőpontos szakadatokat tartalmazó térkép exportálásakor van értelme.
- *Layer number* – Az exportálandó réteg száma.

Opciók:

- **Correct the coordinates to fit the VTK-OpenGL precision** – Úgy változtatja meg az összetevők koordinátáit, hogy pontosságuk legfeljebb az OpenGL-ben engedélyezett pontossággal egyezzen meg.
- **Export numeric attribute table fields as VTK scalar variable** – Az attribútumtáblát VTK skálár változóként exportálja.

### 2.7.4. Raszteres állományok importálása

A GRASS meglehetősen sokféle raszteres képi- és térinformatikai állományt képes importálni, mindezenre elmondható, hogy a szokásosnál többet. Az importálás alapvetően nagyon hasonló az exportáláshoz, csak éppen fordított irányban. Ennek megfelelően az input map az importálandó raszteres fájl, az output pedig az eredményül kapott GRASS bináris, raszteres térkép. Figyeljünk arra, hogy az importálás abba mapsetbe történik, amelyet a GRASS indításakor kiválasztottunk!

#### 2.7.4.1. Multiple formats using GDAL

Importálásra is használja a GRASS a GDAL-t, hasonlóan a raszteres exportáláshoz (lásd ott), és itt is igen sok formátumot támogat, még többet is, mint az exportálásnál:

AAIGrid: Arc/Info ASCII Grid  
ADRG: ARC Digitized Raster Graphics  
AIG: Arc/Info Binary Grid  
AirSAR: AirSAR Polarimetric Image  
BAG: Bathymetry Attributed Grid  
BIGGIF: Graphics Interchange Format (.gif)  
BLX: Magellan topo (.blx)  
BMP: MS Windows Device Independent Bitmap  
BSB: Maptech BSB Nautical Charts  
BT: VTP .bt (Binary Terrain) 1.3 Format  
CEOS: CEOS Image  
COASP: DRDC COASP SAR Processor Raster

COSAR: COSAR Annotated Binary Matrix (TerraSAR-X)  
CPG: Convair PoIGASP  
DIMAP: SPOT DIMAP  
DIPEX: DIPEX  
DOQ1: USGS DOQ (Old Style)  
DOQ2: USGS DOQ (New Style)  
DTED: DTED Elevation Raster  
EHdr: ESRI .hdr Labelled  
EIR: Erdas Imagine Raw  
ELAS: ELAS  
ENVI: ENVI .hdr Labelled  
ERS: ERMapper .ers Labelled  
ESAT: Envisat Image Format  
FAST: EOSAT FAST Format  
FIT: FIT Image  
FujiBAS: Fuji BAS Scanner Image  
GenBin: Generic Binary (.hdr Labelled)  
GFF: Ground-based SAR Applications Testbed File Format (.gff)  
GIF: Graphics Interchange Format (.gif)  
GMT: GMT NetCDF Grid Format  
GRASS: GRASS Database Rasters (5.7+)  
GRIB: GRIdded Binary (.grb)  
GS7BG: Golden Software 7 Binary Grid (.grd)  
GSAG: Golden Software ASCII Grid (.grd)  
GSBG: Golden Software Binary Grid (.grd)  
GSC: GSC Geogrid  
GTiff: GeoTIFF  
GXF: GeoSoft Grid Exchange Format  
HDF4: Hierarchical Data Format Release 4  
HDF4Image: HDF4 Dataset  
HDF5: Hierarchical Data Format Release 5  
HDF5Image: HDF5 Dataset  
HFA: Erdas Imagine Images (.img)  
HTTP: HTTP Fetching Wrapper  
IDA: Image Data and Analysis  
ILWIS: ILWIS Raster Map  
INGR: Intergraph Raster  
ISIS2: USGS Astrogeology ISIS cube (Version 2)  
ISIS3: USGS Astrogeology ISIS cube (Version 3)  
JAXAPALSAR: JAXA PALSAR Product Reader (Level 1.1/1.5)  
JDEM: Japanese DEM (.mem)  
JPEG: JPEG JFIF  
JPEG2000: JPEG-2000 part 1 (ISO/IEC 15444-1)  
LIB: NOAA Polar Orbiter Level 1b Data Set

LAN: Erdas .LAN/GIS  
LCP: FARSITE v.4 Landscape File (.lcp)  
Leveller: Leveller heightfield  
MEM: In Memory Raster  
MFF: Vexcel MFF Raster  
MFF2: Vexcel MFF2 (HKV) Raster  
MSGN: EUMETSAT Archive native (.nat)  
NDF: NLAPS Data Format  
netCDF: Network Common Data Format  
NITF: National Imagery Transmission Format  
NWT\_GRC: Northwood Classified Grid Format .grc/.tab  
NWT\_GRD: Northwood Numeric Grid Format .grd/.tab  
OGDI: OGDI Bridge  
PAux: PCI .aux Labelled  
PCIDSK: PCIDSK Database File  
PCRaster: PCRaster Raster File  
PDS: NASA Planetary Data System  
PNG: Portable Network Graphics  
PNM: Portable Pixmap Format (netpbm)  
R: R Object Data Store  
Rasterlite: Rasterlite  
RIK: Swedish Grid RIK (.rik)  
RMF: Raster Matrix Format  
RPFTOC: Raster Product Format TOC format  
RS2: RadarSat 2 XML Product  
RST: Idrisi Raster A.1  
SAGA: SAGA GIS Binary Grid (.sdat)  
SAR\_CEOS: CEOS SAR Image  
SDTS: SDTS Raster  
SGI: SGI Image File Format 1.0  
SRP: Standard Raster Product (ASRP/USRP)  
SRTMHGT: SRTMHGT File Format  
Terragen: Terragen heightfield  
TIL: EarthWatch .TIL  
TSX: TerraSAR-X Product  
USGSDEM: USGS Optional ASCII DEM (and CDED)  
VRT: Virtual Raster  
WCS: OGC Web Coverage Service  
WKT Raster: PostGIS WKT Raster driver  
WMS: OGC Web Map Service  
XPM: X11 PixMap Format

A menü kiválasztása után megjelenő ablak négy fülben tartalmazza a kitöltendő mezőket és a választható opciókat.

### Required

- *Raster file to be imported* – Az importálandó raszteres állomány neve.
- *Name for output raster map* – Az eredményül kapott GRASS raszteres térkép neve.

### Options

- *Band to select* – Az importálandó sáv száma. Ha az importálandó raszteres állományhoz tartozik színtábla, akkor ebből tudható meg. Alapesetben a vörös (red), a zöld (green) és a kék (blue) összetevőt szokták tartalmazni a képek, de persze ettől eltérő is lehet. Ha nem adunk meg semmit sem, a GRASS mindegyik sávot importálja. Ha például egy 24 bit színmélységű, vagyis szokásos *jpg* állományt importálunk akkor a GRASS azt a három összetevőre bontja és három raszteres térképet hoz létre. Mindegyik térkép neve utal arra, hogy melyik színkivonatot tartalmazza. Ha mondjuk a *Name for output raster map* mezőben azt adtuk meg a kimeneti állomány nevéként, hogy *terkep*, és nem választottunk ki importálandó sávot, akkor mindhárom sávot importálja, és a három térkép neve ez lesz: *terkep.red*, *terkep.green*, *terkep.blue*.

Az importált sáv szürkeskálás (grayscale) lesz és így is jelenik meg. A GRASS későbbi verzióiban változás lesz, hogy ha nem adunk meg színsávot, akkor az egész képet importálja a GRASS, de szürkeskálásan.

- *Name of location to read projection from for GCPs transformation* – Megadható egy létező location, amelyből a GRASS a vetületi adatokat veszi és azok szerint hajtja végre a vetületi konverziót.

Ez például akkor hasznos, ha egy olyan térképet akarunk importálni, amely ugyanazt a területet fedi le, mint az a location, amelybe importálni akarjuk, de más vetületű. Ekkor, ha van olyan locationunk, amelynek a vetülete egyezik az importálandó térképével, megadjuk annak a nevét. Ebből a GRASS tudni fogja, hogy miképpen alakítsa át az importálandó térkép, illetve a vetületét, így a térkép az importálás után használható lesz az adott locationban (mapsetben).

- *Name for new location to create* – Ha itt megadunk egy nevet, akkor a GRASS ezen a néven létrehoz egy új locationt és abba importálja be a raszteres állományt (egyébként abba a mapsetbe importálja, amelyikben éppen dolgozunk). Az új location vetületi adatait az importálandó térképből veszi.
- **Override projection (use location's projection)** – Felülbírálja az importálandó térkép vetületi adatait (ha egyáltalán vannak) és annak a locationnak a vetületét veszi figyelembe, amelyikben éppen dolgozunk.
- **Extend location extents based on new dataset** – Az adott location kiterjedését (az alapértelmezett régiót) kiterjeszti ha szükséges, hogy az importált térkép is teljes egészében beleférjen. Használatkor engedélyezzük az *Override projection (use location's projection)* opciót is!
- **Keep band numbers instead of using band color names** – Az importálás után nem az importált sáv nevét, hanem a számát kapja az importált térkép (lásd a *Band to select* mezőnél, fentebb!)

### Print

- *List supported formats and exit* – Kilitázza az Output-GIS.m ablakba a használható formátumokat, és semmilyen más műveletet nem hajt végre.

### Metadata

- *Title for resultant raster map* – Az importált térképnek adhatunk címet.

## 2.7.4.2. Multiple formats using GDAL (link into instead of import)

Használata alapvetően egyezik az előző pontéval (*Multiple formats using GDAL*), de lényeges különbség, hogy valójában nem importálja a megadott raszteres térképet, hanem csak egy rá mutató linket hoz létre.

## 2.7.4.3. Aggregate ASCII xyz

ASCII tömböt importál, az adatszerkezetet lásd a 73. oldalon.

- **Scan data file for extent then exit** – végignézi az adatállományt, hogy talál-e benne a területi kiterjedésre (határokra) vonatkozó adatokat és megjeleníti azokat..
- **In scan mode, print using shell script style** – Ha *scan* módban van, vagyis az előző opció be van kapcsolva, akkor a kimeneten az eredményt shell script szerkezetben jeleníti meg.
- **Ignore broken line** – Figyelmen kívül hagyja a törött sorokat, vagyis azokat, melyek hamarabb fejeződnek be, mint ahogy kellene.
- *ASCII file containing input data* – Az importálandó ASCII állomány neve.
- *Name for output raster map* – A kimeneti raszteres térkép neve.
- *Statistic to use for raster values* – A kimeneti térkép szakadatainak előállítására használandó eljárás:

- n – számosság
- min – legkisebb érték
- max – legnagyobb érték
- range – terjedelem
- sum – összeg
- mean – számtani közép (ez az alapértelmezett)
- stddev – szórás
- variance – szórásnégyzet
- coeff\_var –
- median – medián (középső érték)
- percentile – percentilis
- skewness – ferdeség (eltérés a normál eloszlástól)
- trimmean – trimmelt közép

- *Storage type for resultant raster map* – a kimeneti raszteres térkép típusa:
  - CELL – egész szám típusú,
  - FCELL – lebegőpontos típusú (ez az alapértelmezett),
  - DCELL – dupla pontosságú lebegőpontos szám típusú.
- *Field separator* – mezőelválasztó karakter (alapértelmezett: |)
- *Column number of x coordinates in input file (first column is 1)* – Megadható, hogy az ASCII állomány hányadik oszlopa tartalmazza a térképi  $x$  koordinátákat. Balról az első oszlop az 1-es. Alapértelmezése 1. Akkor használható, ha az *Input file format* legördülő menüben a *point* formátumot választottuk.
- *Column number of y coordinates in input file* – Megadható, hogy az ASCII állomány hányadik oszlopa tartalmazza a térképi  $y$  koordinátákat. Balról az első oszlop az 1-es. Alapértelmezése 2. Akkor használható, ha az *Input file format* legördülő menüben a *point* formátumot választottuk.
- *Column number of data values in input file* – Megadható, hogy az ASCII állomány hányadik oszlopa tartalmazza a térkép elemeihez tartozó azonosítókat (kategória). Az első oszlop az 1-es. Alapértelmezése 0, vagyis alapértelmezésben nem importál ilyen adatokat a GRASS. Akkor használható, ha az *Input file format* legördülő menüben a *point* formátumot választottuk.
- *Filter range for z data (min,max)* – Csak azokat a rasztereket importálja, amelyek magassági értéke az itt megadott határok közötti. Az importálandó legkisebb és legnagyobb értéket veszővel elválasztva kell megadni, például így: *1200,1650*.
- *Scale to apply to z data* – Az importálás során az itt megadott értékkel szorozza meg a magassági adatokat. Ha mondjuk a magassági adatok az ASCII fájlban centiméterben vannak megadva, ám mi méterben kívánjuk importálni, akkor írjuk ide azt, hogy *0.1*. Alapértelmezése 1.
- *Percent of map to keep in memory* – A térképnek a memóriában tartandó része. 0-tól 100-ig terjedő számmal lehet megadni, hogy hány százalékát tartsa a memóriában. Alapértelmezése 100, de ne feledjük, hogy ez az importálómodul egészen nagy, akár több száz gigabájtos állományokat is képes importálni, és ilyenkor már problémás lehet, ha a térkép egészét a memóriában kell tárolnia.
- *pth percentile of the values* – Az adható meg itt, hogy a térképi értékek hányadik percentiliséit importálja. 1 és 100 közötti egész számmal adható meg a kívánt érték.
- *Discard <trim> percent of the smallest and <trim> percent of the largest observations* – Az itt megadott százaléknál kevesebb adatot hagyja el az adatsor aljáról és tetejéről is. Ha mondjuk ide azt írjuk, hogy 10, akkor az adatsor aljáról nem importálja a legalacsonyabb 10 százalékot, az adatsor tetejéről pedig a legmagasabb 10 százalékot. Értéke 0 és 50 közötti lebegőpontos szám lehet.

### 2.7.4.4. ASCII grid

Egyszerű ASCII állományú adatot importál a GRASS-ba.

- **Integer values are imported** – Egész számként importálja az értékeket.
- **Floating points values are imported** – Lebegőpontos számként importálja az értékeket.

- **Double floating points values are imported** – Dupla pontosságú lebegőpontos számként importálja az értékeket.
- **SURFER (golden Software) files will be imported** – SURFER állományként kezeli az importálandó fájlt.
- *ASCII raster file to be imported. If not given reads from standard input* – Az itt megadott állományt importálja.
- *Name of output raster map* – Az eredményül kapott raszteres térkép neve.
- *Title for resultant raster map* – Az eredményül kapott raszteres térkép címe.
- *Multiplier for ASCII data* – Szorzófaktor. Az itt megadott értékkel szorozza az importált adatokat. Ennek az az értelme, hogy így már az importálásnál elvégezhető valamilyen szükséges konverzió és az importált térkép rögtön használható is. Ez a helyzet például akkor, ha az eredeti állomány magassági adatai láb-ban vannak megadva, de méterbe kell átszámolnunk. A szorzófaktor bármilyen tizedestört lehet.
- *String representing NULL value data cell* – Az itt megadott karakterláncot az importálás során NULL értékűvé alakítja a GRASS. Mivel az ASCII állományokban csak úgy tárolható NULL érték, hogy azt valamilyen karaktersorozat reprezentálja, ezért ha ismerjük ezt a karakterláncot, az importálásakor azt megadhatjuk a GRASS-nak. Hogy mi reprezentálja a NULL értékeket, azt egyszerűen megtudjuk, ha az az ASCII állományt egy egyszerű szövegszerkesztővel megnyitjuk. Alapesetben egyébként az ASCII állomány fejléce tartalmazza, hogy mit kell NULL-nak tekinteni.

#### 2.7.4.5. ASCII polygons and lines

ASCII állományban tárolt poligonokat (területeket) és vonalakat importál.

- *Name of input file* – Az importálandó állomány neve
- *Name of output raster map* – Az eredményül kapott raszteres térkép neve.
- *Title for resultant raster map* – Az eredményül kapott raszteres térkép címe.
- *Number of rows to hold in memory* – A memóriában tartandó sorok száma. Az alapértelmezés általában jól működik, de ha kell, felülbíráható ez az érték. Ha valami hibát észlelünk, elsőként érdemes lehet ezt az értéket csökkenteni, akár néhány százig is. A kisebb érték lassítja a művelet sebességét, a nagyobb gyorsítja.

#### 2.7.4.6. Binary

Bináris raszteres állományt importál.

##### Options

- **Import as Floating Point Data** – Az adatokat lebegőpontos számokként importálja, akkor is, ha egyébként egész számok.

- **Import as Double Precision Data** – Az adatokat dupla pontosságú lebegőpontos számokként importálja.
- *Binary raster file to be imported* – Az importálandó bináris állomány neve.
- *Name for output raster map* – A kimeneti állomány neve.
- *Title for resultant raster map* – A kimeneti állomány (térkép) címe.

### Settings

- **Byte Swap the Data During Import** – A bájtok sorrendjének felcserélése az importálás során.
- *Number of bytes per cell (1, 2, 4)* – A bájtok száma raszterenként a kimeneti állományban. Értéke 1, 2 vagy 4 lehet. Alapértelmezése 1.
- *Set Value to NULL* – Az itt megadott, lebegőpontos típusú, számot az importálás során NULL értékűvé alakítja.

### Bounds

- **Get region info from GMT style header** – Az importálás során az importálandó térkép által lefedett terület határait a GMT típusú fejlécből olvassa ki a GRASS. Ha egyénileg kívánjuk megadni, hogy a térkép mely részét importálja, akkor az, az alábbi mezők kitöltésével lehetséges. Ekkor ezt az opciót nem kell engedélyeznünk!
- *Northern limit of geographic region (outer edge)* – Az importálandó terület északi szegélyének észak-déli koordinátája.
- *Southern limit of geographic region (outer edge)* – Az importálandó terület déli szegélyének észak-déli koordinátája.
- *Eastern limit of geographic region (outer edge)* – Az importálandó terület keleti szegélyének kelet-nyugati koordinátája.
- *Western limit of geographic region (outer edge)* – Az importálandó terület nyugati szegélyének kelet-nyugati koordinátája.
- *Number of rows* – Sorok száma az importálandó területben (vagyis az észak-déli irányú felbontás).
- *Number of columns* – Oszlopok száma az importálandó területben (vagyis a kelet-nyugati irányú felbontás).

#### 2.7.4.7. ESRI grid

ESRI ARC/info struktúrájú ASCII állományokat importál, az adatszerkezetet és az általános tudnivalókat részletesen lásd a 62. oldalon.

- *ARC/INFO ASCII raster file to be imported* – Az importálandó fájl neve.
- *Name for output raster map* – Az eredményül kapott raszteres térkép neve.

- *Storage type for resultant raster map* – Az eredményül kapott raszteres állomány típusa:
  - CELL – egész szám,
  - FCELL – lebegőpontos szám,
  - DCELL – dupla pontosságú lebegőpontos szám.
- *Title for resultant raster map* – Az eredményül kapott térkép címe.
- *Multiplier for ASCII data* – Szorzófaktor. Az itt megadott értékkel szorozza az importált adatokat. Ennek az az értelme, hogy így már az importálásnál elvégezhető valamilyen szükséges konverzió és az importált térkép rögtön használható is. Ez a helyzet például akkor, ha az eredeti állomány magassági adatai centiméterben vannak megadva, de méterbe kell átszámolnunk. A szorzófaktor bármilyen tizedestört lehet.

#### 2.7.4.8. GRIDATB.FOR

Csak a bemeneti és a kimeneti állomány nevét kell megadni.

#### 2.7.4.9. MAT-File (v.4)

Csak a bemeneti és a kimeneti állomány nevét kell megadni.

#### 2.7.4.10. SPOT NDVI

1 km felbontású *SPOT vegetation* (növényborítás) NDVI-ket importál. A képek az importálás után az *ndvi* színtábla színeit kapják.

Csak a bemeneti és a kimeneti állomány nevét kell megadni. Egyetlen opciója van:

- **Also import quality map (SM status map layer) and filter NDVI map** – Szűrt kép, általában jobb minőségű, mint a nyers. Érdemes lehet az így importált és az e nélkül importált képet összevetni.

#### 2.7.4.11. SRTM hgt

Csak a bemeneti és a kimeneti állomány nevét kell megadni. Egyetlen opciója:

- **Input is a 1 arcset tile (default 3 arcsec)** – Ha az importálandó állomány 1 szögmásodperc felbontású, akkor az ennek az opciónak a bekapcsolásával kell jelezni. E nélkül alapértelmezésben 3 szögmásodperc felbontásúnak tekinti az importálandó állományt.

### 2.7.4.12. Terra ASTER

- **Input ASTER image to be georeferenced & rectified** – Az importálandó állomány neve.
- **ASTER imagery processing type** – Választanunk kell a legördülő menüből, hogy milyen ASTER típusúállományt akarunk importálni.
- **L1A or L1B band to translate** – Értéke csak 1, 2, 3n, 3b, valamint 4-14 közötti egész szám lehet.
- **Name for output raster map** – Az eredményül kapott raszteres térkép neve.

### 2.7.4.13. Web Mapping Server

Arra való, hogy egy megadott szerverről térképeket szerezzünk be.

#### Request

- **List available layers and exit** – Az elérhető rétegek listázása az Output-GIS.m ablakba és kilépés, minden egyéb művelet elvégzése nélkül
- **Don't request transparent data** – Átlátszó adatok figyelmen kívül hagyása
- **Use GET method instead of POST data method** – Ha azt a szerver (illetve a csatlakozás) lehetővé teszi, akkor ne GET, hanem POST eljárással szerezzük be az adatokat.
- *Mapserver to request data from* – A használandó térképszerver neve.
- *Layers to request from mapserver* – A letöltendő réteg(ek) neve.
- *Styles to request from mapserver* – A kérés teljesítésére vonatkozó opciók megadása (például, hogy méterben legyenek a magassági adatok). Ezekről az opciókról általában a használni kívánt térképszerver honlapján kaphatunk tájékoztatást.
- *Source projection to request from server* – A kért térkép vetülete. Alapértelmezésben ez a 4326 EPSG kódú vetületi rendszer, ami nem más, mint a WGS84. Ha mást akarunk, akkor adjuk meg az EPSG kódját.
- *Image format requested from the server* – A kért állomány képformátuma. Választható formátumok:
  - geotiff,
  - tiff,
  - jpeg,
  - gif,
  - pngAz alapértelmezés geotiff
- *Additional query options for server* – További lekérdezési paraméterek (pl. SQL) adhatók át a szervernek. Alapértelmezése version=1.1.1

- *Maximum columns to request at a time* – Az egyszerre letöltendő oszlopok száma legfeljebb (a képméretre utal). Csak egész szám adható meg, alapértelmezése 1024.
- *Maximum rows to request at a time* – Az egyszerre letöltendő sorok száma legfeljebb (a képméretre utal). Csak egész szám adható meg, alapértelmezése 1024.
- *Named region to request data for* – Névvvel ellátott régió adható meg, amelyre vonatkozóan az adatokat kérjük, Ha üresen hagyjuk, akkor az aktuális számítási régiót veszi figyelembe.
- *Filename to save capabilities XML file to* – Megadhatunk egy fájlt, amelybe – *xml* formátumban – elmenti a kérés körülményeit és eredményét. Használatához a *Requires list available layers* opciót engedélyezni kell.

### Download

- **Skip to downloading (to resume downloads faster)** – A letöltés felfüggesztése (hogya aztán később, ha gyorsabb a kapcsolat, folytathassuk)
- **Clean existing data out of download directory** – A letöltési könyvtáron kívüli adatok törlése a letöltés után.
- *Folder to save downloaded data to* – A letöltött adatok elhelyezésére szolgáló könyvtár elérési útja és neve.
- *Additional options for wget* – a letöltést végző, *wget* nevű, programnak átadható további opciók. Alapértelmezése: `-c -t 5 -nv`.
- *Additional options for curl* – A *curl* nevű programnak átadható további opciók. Alapértelmezése: `-C --retry 5 -s -S`

### Import

- **Keep band numbers instead of using band color names** – A színsávok számát használja, ne a nevüket.
- **Don't reproject the data, just patch it** – A letöltött adatokon ne végezzen vetületi transzformációt, csak illessze be azokat.
- *Name for output raster map* – Az eredményül kapott raszteres térkép neve
- *Reprojection method to use* – A vetületi transzszformáció során esetleg felmerülő interpolációs eljárás megadása.
  - nearest,
  - bilinear,
  - cubic,
  - cubicsplinealapértelmezett a nearest.

### Options

- *Additional options for r.tileset* – A GRASS *r.tileset* nevű részprogramjának szóló további opciók adhatók meg. További információért a GRASS-ban adjuk meg: `man. r.tileset`.

## 2.7.5. Vektoros állományok importálása

### 2.7.5.1. Multiple formats using OGR

A vektoros exportáláshoz hasonlóan ezzel a menüponttal többféle állományt is kezelhetünk, jelen esetben importálhatunk. A támogatott fájlformátumok az alábbiak:

- AVCbin
- AVCE00
- BNA
- CSV
- DGN
- DXF
- ESRI Shapefile
- Geoconcept
- GeoJSON
- GeoRSS
- GML
- GMT
- GPSTrackMaker
- GPX
- GRASS
- Interlis 1
- Interlis 2
- KML
- MapInfo File
- Memory
- MySQL
- ODBC
- OGDI
- PCIDSK
- PGeo
- PostgreSQL
- REC
- S57
- SDTS
- SQLite
- TIGER
- UK .NTF
- VFK
- VRT
- XPlane

### Required

Kötelezően megadandó adatok.

- *OGR data source name* – Az importálandó vektoros állomány neve. ESRI shapefile és Mapinfo vektoros térképek importálásakor annak a könyvtárnak a nevét kell megadni, amely az importálandó térkép összes állományát tartalmazza. Éppen ezért, importálás előtt célszerű az egyazon térképhez tartozó fájlokat egyazon alkönyvtárba másolni/áthelyezni, ha eredetileg nem így voltak.
- *Name for output vector map* – Az eredményül kapott vektoros térkép neve

### Options

Választható opciók, kiegészítő lehetőségek.

- **List available layers in data source and exit** – Csak kilistázza az Output-GIS.m ablakba az importálandó állományban található rétegek nevét, de semmilyen más műveletet nem hajt végre.
- **List supported formats and exit** – Csak kilistázza az Output-GIS.m ablakba az importálandó fájl típusokat, de semmilyen más műveletet nem hajt végre.
- **Do not clean polygons (not recommended)** – Nem hajt végre automatikus topológia ellenőrzést és javítást. Nem ajánlott a használata, bár nem is tilos.
- **Create 3D output** – A kimenet 3D vektoros térkép lesz, de csak akkor, ha az importálandó állomány tartalmaz olyan adatokat (mondjuk magsságiakat), amelyek erre alkalmassá teszik.
- **Override dataset projection (use location's projection)** – Felülbírálja az importálandó térkép vetületi adatait (ha egyáltalán vannak) és annak a locationnak a vetületét veszi figyelembe, amelyekben éppen dolgozunk.
- **Extend location extents based on new dataset** – Az adott location kiterjedését (az alapértelmezett régiót) kiterjeszti ha szükséges, hogy az importált térkép is teljes egészében beleférjen. Használatakor engedélyezzük az *Override projection (use location's projection)* opciót is!
- *Name for new location to create* – Ha itt megadunk egy nevet, akkor a GRASS ezen a néven létrehoz egy új locationt és abba importálja be a vektoros állományt (egyébként abba a mapsetbe importálja, amelyikben éppen dolgozunk). Az új location vetületi adatait az importálandó térképből veszi.

### Attributes

Szakadatokra vonatkozó beállítások.

- **Do not create attribute table** – Nem hoz létre szakadat táblát az importált állományhoz és nem is csatol hozzá semmilyen adattáblát.
- **Change column names to lowercase characters** – Engedélyezésével az importált térkép adattáblája úgy változik, hogy az egyes oszlopok (mezők) nevében minden nagybetűt kisbetűre cserél a program.
- *List of column names to be used instead of original names, first is used for category column* – A vektoros térképhez csatolt adattábla oszlopainak nevét nem importálja, hanem azzal helyettesíti, amit megadunk. A neveket egymás után, vesszővel elválasztva kell megadni, de sem a vesszők

előtt, sem utánuk nem kell szóközt tenni. Az elsőnek megadott név lesz a kategória mező neve. A megadott nevek nem kezdődhetnek számmal, nem tartalmazhatnak szóközt és ékezetes karaktereket, továbbá nem lehetnek tíz karakternél hosszabbak.

### Subregion

Az importálandó területre vonatkozó beállítások.

- **Limit import to the current region** – Az importálandó térképnek csak azt a részét importálja, amelyik az aktuális számítási régió területére esik.
- *Import subregion only* – Csak a megadott területet importálja. A terület megadása úgy történik, hogy megadjuk a határait az adott térkép vetületi rendszere (geodéziai dátuma) szerinti mértékegységben. A határokat ebben a sorrendben kell megadni: *legkisebb x, legkisebb y, legnagyobb x, legnagyobb y*. A számokat vesszővel kell elválasztani, de sem a vesszők elé, sem a vesszők mögé ne tegyünk szóközt!

### Selection

- *OGR layer name. If not given, all available layers are imported* – Az importálandó réteg neve. Ha nem adunk meg nevet, minden réteget importál.
- *Where conditions of SQL statement whitout 'where' keyword* – Csak azokat az elemeket importálja a bementi állományból, amelyek megfelelnek az itt megadott SQL kérésnek. Figyeljünk rá, hogy az SQL kérésnek csak a „where” utáni részét kell megadni!
- *Optionally change default input type* – Megadható, hogy milyen módon importáljon a GRASS egyes elemtípusokat:
  - *point* – minden súlypontot (centroidot) pontként importál,
  - *line* – minden határvonalat vonalként importál,
  - *boundary* – minden vonalat határvonalként importál,
  - *centroid* – minden pontot súlypontként (centroidként) importál.

### Min-area & Snap

- *Minimum size of area to be imported (square units)* – Megadható, hogy mekkora legyen az a legkisebb terület, amelyet még importálni akarunk. A területet az importálandó térkép mértékegységében kell megadni és négyzetegységben értelmezendő. Ha például azt adjuk meg, hogy 1000 és a térkép mértékegysége méter, akkor ez azt eredményezi, hogy az 1000 m<sup>2</sup>-nél kisebb területeket a GRASS nem importálja.
- *Snapping threshold for boundaries* – A határvonalak automatikus összeillesztésének küszöbértéke. Ha két határvonal csomópontjai az itt megadott értéknél közelebb vannak egymáshoz, akkor azokat a GRASS összekapcsolja a csomópontok elmozdításával. Ha nem kívánjuk ezt a lehetőséget használni, akkor adjunk meg -1 (mínusz egy) értéket. Egyébként ez az alapértelmezés is.

### 2.7.5.2. ASCII points or GRASS ASCII vector

#### Required

- ASCII fájl name to be imported – Az importálandó ASCII állomány neve.

### Options

- **Create 3D vector map** – Az importálandó állományt 3D-ként értelmezi és 3D vektoros állományt hoz létre belőle, ha lehet.
- **Create a new empty vector map and exit. Nothing is read from the input** – Új, üres vektoros térképet készít, de ez kívül semmit nem csinál, és nem is olvas be semmit.
- **Don't expect a header when reading in standard format** – Akkor használható, ha az *Input file format* legördülő menüben a *standard* formátumot választottuk. Ekkor ennek az opciónak az engedélyezésekor nem vizsgálja az importálandó állomány fejlécét, hanem mindenképpen megkísérel az importálást.
- **Do not build topology in points mode** – Akkor használható, ha az *Input file format* legördülő menüben a *point* formátumot választottuk. Ennek az opciónak az engedélyezésével nem épít topológiát az importálás után.
- **Only import points falling within current region (points mode)** – Akkor használható, ha az *Input file format* legördülő menüben a *point* formátumot választottuk. Ilyenkor, ha ezt az opciót engedélyezzük, csak azt a részét importálja az ASCII állománynak, amelyik az éppen aktuális számítási régió területére esik.
- *Name for output vector map* – Az eredményül kapott vektoros térkép neve.
- *Input file format* – Az importálandó ASCII állomány adatstruktúráját választhatjuk ki.
  - *point* – Az állomány csak pontokat tartalmaz.
  - *standard* – Az állomány pontokat, vonalakat és területeket (poligonokat) is tartalmazhat.
- *Filed separator* – Adjuk meg, hogy az importálandó ASCII állományban mi a mezőválasztó karakter. Alapértelmezése a „|” karakter, de ez természetesen megváltoztatható. Ha például az eredeti ASCII állományban a mezőválasztó szóköz, akkor egyszerűen csak töröljük a | karaktert és írjuk a mezőbe egy szóközt”
- *Number of the header lines to skip at top of input file (points mode)* – Akkor használható, ha az *Input file format* legördülő menüben a *point* formátumot választottuk. Ekkor megadhatjuk, hogy az importálandó ASCII állomány első hány sorát hagyja ki.

### Columns

- **Do not create table in points mode** – Akkor használható, ha az *Input file format* legördülő menüben a *point* formátumot választottuk. Ha engedélyezzük, akkor nem hoz létre adattáblát az importált vektoros térképhez.
- *Column definition in SQL style (points mode)* – SQL stílusban adható meg a vektoros térképhez tartozó adattáblába új oszlop. Akkor használható, ha az *Input file format* legördülő menüben a *point* formátumot választottuk.
- *Number of column used as x coordinate (first column is 1) points mode* – Megadható, hogy az ASCII állomány hányadik oszlopa tartalmazza a térképi *x* koordinátákat. Balról az első oszlop az 1-es. Alapértelmezése 1. Akkor használható, ha az *Input file format* legördülő menüben a *point* formátumot választottuk.

- *Number of column used as y coordinate (first column is 1) points mode* – Megadható, hogy az ASCII állomány hányadik oszlopa tartalmazza a térképi y koordinátákat. Balról az első oszlop az 1-es. Alapértelmezése 2. Akkor használható, ha az *Input file format* legördülő menüben a *point* formátumot választottuk.
- *Number of column used as z coordinate (first column is 1) points mode* – Megadható, hogy az ASCII állomány hányadik oszlopa tartalmazza a térképi z (magassági) koordinátákat. Balról az első oszlop az 1-es. Alapértelmezése 0, vagyis alapértelmezésben nem is próbál meg magassági koordinátákat importálni (hiszen nincsen nulladik oszlop). Akkor használható, ha az *Input file format* legördülő menüben a *point* formátumot választottuk.
- *Number of column used as category (first column is 1) points mode* – Megadható, hogy az ASCII állomány hányadik oszlopa tartalmazza a térkép elemeihez tartozó azonosítókat (kategória). Az első oszlop az 1-es. Alapértelmezése 0, vagyis alapértelmezésben nem importál ilyen adatokat a GRASS. Akkor használható, ha az *Input file format* legördülő menüben a *point* formátumot választottuk.

### 2.7.5.3. Old GRAS vector

Régebbi, 6.0 verziónál korábbi, GRASS vektoros térképeket importál.

- *Name of input vector map* – Az importálandó térkép neve.
- *Name of output vector map* – Az eredményül kapott, importált, térkép neve.
- *Endian of input vector map* – A bementi térkép bájt-sorrendje. Alapvetően fontos az állomány helyes értelmezése szempontjából. Alapértelmezése a *little*, ez az Intel architektúrán használt (és így a legelterjedtebb) bájt-sorrend.

### 2.7.5.4. DXF

A CAD-rendszerek elterjedt formátumát importálja.

#### Options

- **Ignore the map extent of DXF file** – Figyelmen kívül hagyja a DXF állományban tárolt térkép kiterjedését.
- **Do not create attribute table** – Engedélyezése azt eredményezi, hogy az eredményül kapott vektoros térkép adattábla nélkül készül el, akkor is, ha eredetileg tartoztak szakadatok az egyes objektumokhoz.
- **Do not build topology** – Nem építi fel az importált térkép topológiáját.
- **Import polyface meshes as 3D wire frame** – Az importálandó DXF állományban levő, poligonokból álló, homlokzatokat, mint 3D drótvázakat importálja.
- *Name of input DXF file* – Az importálandó állomány neve.

- *Name of output vector map* – Az eredményül kapott térkép neve.

#### **DXF layers**

- **List available layers and exit** – Csak kilistázza az Output-GIS.m ablakba az importálandó állományban található rétegeket, de semmilyen más műveletet nem hajt végre.
- **Invert selecton by layers (don't import layers in list)** – Csak azokat a rétegeket importálja, amelyeket nem tartalmaz a *List of layers to import mezőben megadott lista*.
- **Import all objects into one layer** – Minden réteget egyetlen térképbe importál. Ha nem engedélyezzük ezt az opciót, akkor minden rétegből különálló térkép készül. Ez egyes esetekben (például szintvonalak) igen sok térképet eredményezhet.
- *List of layers to import* – Megadható, hogy mely rétegeket importálja a GRASS. Célszerű előbb a *List available layers and exit*-tel kilistáztatni a rétegeket és aztán választani közülük.

#### **2.7.5.5. ESRI e00**

- *E00 file* – Az importálandó fájl neve.
- *Input type, point, line or area* – Legördülő menüből választható ki, hogy milyen típusú a bemeneti állomány: pont, vonal, vagy terület.
- *Name of output vector map* – Az eredményül kapott vektoros térkép neve.

#### **2.7.5.6. Garmin GPS**

Útvonalpontokat, utakat és nyomvonalakat tölts le Garmin GPS készülékekről, vektoros térképpé alakítva azokat.

- **Verbose mode** – Bőbeszédű mód, részletesen tájékoztat az Output ablakban.
- **Download Waypoints from GPS** – Útpontok letöltése GPS-ről.
- **Download Routes from GPS** – Utak letöltése GPS-ről.
- **Download Track from GPS** – Nyomvonalak letöltése GPS-ről.
- **Force import of track or route data as points** – Track és route típusú adatokat is pontokként importálja GPS-ről.
- **Use gardump instead of gpstrans as the download program** – Letöltőprogramként inkább a *gardump*-ot használd a *gpstarns* helyett.
- **Import track in 3D (gardump only)** – A nyomvonalakat 3D állományként importálja (csak, ha az előző opció engedélyezve van!)
- **Do not attempt projection transform from WGS84** – Ne hajtson végre vetületi átszámítást WGS84-ről.

- **Allow output files to overwrite existing files** – Engedélyezi, hogy már létező állományt felülírjunk a *Name for output vector map (omit for display to stdout)* mezőben megadott néven.
- **Verbose module output** – Bőbeszédő kimenet engedélyezése.
- **Quiet module output** – „Csendes” kimenet engedélyezése, neincsenek üzenetek.
- *Name for output vector map (omit for display to stdout)* – A letöltés eredményeként kapott vektoros térkép neve.
- *Port Garmin receiver is connected to* – Itt mondhatjuk meg, hogy hova van csatlakoztatva a vevőkészülék. Ez Linux rendszer esetében a /dev könyvtár valamelyik fájl-ja kell, hogy legyen, alapértelmezése: /dev/gps.

### 2.7.5.7. GPSSbabel GPS

Adatok importálása GPS készülékekről a GPSSbabel programmal.

- **Verbose mode** – Bőbeszédű mód, több üzenet.
- **Import waypoints** – Útpontok letöltése GPS-ről.
- **Import routes** – Utak letöltése a GPS-ről
- **Import track** – Nyomvonalak letöltése a GPS-ről
- **Force vertices of track or route data as points** – Track és route típusú adatokat is pontokként importálja GPS-ről.
- **Do not attempt projection transform from WGS84** – Ne hajtson végre vetületi átszámítást WGS84-ről.
- **Allow output files to overwrite existing files** – Engedélyezi, hogy már létező állományt felülírjunk a *Name for output vector map (omit for display to stdout)* mezőben megadott néven.
- **Verbose module output** – Bőbeszédő kimenet engedélyezése.
- **Quiet module output** – „Csendes” kimenet engedélyezése, neincsenek üzenetek.
- *Device or file used to import data* – Itt mondhatjuk meg, hogy hova van csatlakoztatva a vevőkészülék. Ez Linux rendszer esetében a /dev könyvtár valamelyik fájl-ja kell, hogy legyen, alapértelmezése: /dev/gps.
- *Name for output vector map (omit for display to stdout)* – A kimeneti állomány neve. Ha nem adunk meg semmit, akkor nem menti, csak a képernyőn jeleníti meg a térképet.
- *Format of GPS input data (use gpssbabel supported formats)* – A GPS-ről bejövő adatok formátuma. (a GPSSbabel által támogatott formátumokból lehet választani). Alapértelmezése: garmin
- *Projection of input data (PROJ.4 style), if not set Lat/Lon WGS84 is assumed* – A GPS-ről bejövő adatok vetületét lehet megadni (PROJ.4 formátumban), ha nem szélesség-hosszúság formátumúak és WGS84 dátumúak.

### 2.7.5.8. Geonames

A geonames.org honlapról származó állományok importálására szolgál.

- *Uncompressed geonames file from (with .txt extension)* – A bemeneti *geonames* fájl neve. Figyeljünk rá, hogy csak a kitömörített és txt kiterjesztéssel ellátott állományokat fogadja el a program! (A *geonames*-ről letöltött állományok zip-pel tömörítettek.)
- *Name for output vector map* – Az eredményül kapott vektoros térkép neve.

### 2.7.5.9. GEOnet

Használata megegyezik a *Geonames*-ével

### 2.7.5.10. Matlab and MapGen

Csak vektoros vonalak – például partvonalak – importálására alkalmas.

- **Input map is Matlab format** – Az importálandó adatok Matlab formátumúak, nem pedig MapGen.
- **Create a 3D vector points map from 3 columns Matlab data** –A Matlab-állomány 3 adatából 3D pont térképet hozzon létre.
- *Name of input file in Matlab/MapGen format* – Az importálandó állomány neve.
- *Name of output vector map (omit for display to stdout)* – A kimeneti állomány neve. Ha nem adunk meg nevet, akkor az importált állományt csak a monitoron jeleníti meg a GRASS.

## 2.8. Parancsok használata a GRASS-ban

Ez az a rész, amitől a legtöbb felhasználó ódzkodik, mondván, hogy ez annyira bonyolult, annyira sok számítógépes tudást igénylő feladat, hogy erre ő bizonyosan nem képes, de ha képes is lenne véletlenül, megtanulni akkor sem tudná. Pedig nem ördöngösség ez sem, bizonyoság erre az, hogy igen sokan használják a számítógépük parancssoros lehetőségeit is.

Azt azonban be kell látnunk, hogy bár a parancssoros műveletvégzés nem különösebben bonyolult és nem is nehezen megtanulható eljárás, de a semminél azért több tanulást és gyakorlást igényel. Azoknak, akik számára a számítógép ilyen formában történő használata új, vagy legalábbis nem gyakorlottak benne, először meg kell szokniuk a parancsokkal való kommunikációt, és csak azután fogják megérezni és megérteni, hogy mennyire hatékony kommunikációs forma is ez. Biztatom tehát őket, hogy ha az elején még nem is megy a parancsok használata, vagy haszontalannak, unalmasnak, nehéznek találják a pa-

parancssor megértését, használatát akkor se adják fel a megtanulását. Egyebet sajnos nem tudok mondani, mint azt, hogy a parancsokat használók általános tapasztalata szerint megéri a befektetett munka. Motivációnak még egyet hadd tegyek hozzá: tapasztalt szerint a kívülállók azokat, akik képesek a parancssor használatára, sokkal hozzáértőbbnek gondolják, mint azokat, akik csak a grafikus felületen boldogulnak el. És tapasztalat szerint igazuk is van.

Azok számára, akik a Linux/Unix rendszerek parancssoros használatát jól ismerik, a GRASS parancsainak használata semmiféle nehézséget nem fog okozni. A GRASS parancssora ugyanis valójában nem egyéb, mint egy terminál, melyben a bash minden eszközét használhatjuk.

Akiknek ez nem megy, azok számára bizalomerosítõül lássunk egy példát! Indítsuk el a GRASS-t és írjuk be a terminálba az alábbi:

```
d.mon start=x1;d.rast map=roads
```

Azán üssünk *entert*!

Lám, ha mindent jól csináltunk a GRASS megértette a parancsot: elindított egy új monitorablakot és ebben kirajzolta a *roads* nevű raszteres térképet. Ugye, hogy nem fájt?

De most már térjünk át a parancsok szisztematikus tárgyalására!

A GRASS-ban parancsokat a *Terminal* ablakban adhatunk ki. A parancs beírása után *entert* kell ütünk, ha futtatni szeretnénk az adott parancsot. Egymás után, folytatólagosan több parancs is kiadható és természetesen futtatható is, ebben az esetben a parancsokat pontosvesszővel (;)(semicolon) kell elválasztanunk egymástól. Ebben az esetben a pontosvessző után nem kell szóköz.

A parancsok általános szerkezete az alábbi:

```
[parancs neve] [szóköz] [paraméter] [érték] .
```

Ha az érték az paraméter értéke, akkor közéjük egyenlőség jelet kell tenni (=), ha a parancs értéke, akkor szóközt.

Értelmezzük most az előző példánkat!

*A parancs neve:* d.mon

*paraméter:* start=

*érték:* x1

vagyis: *d.mon start=x1*

Magyarul:

A *d.mon* nevű parancs fusson le, úgy, hogy a számos paramétere közül a *start*-ot indítsa (ez indít új térkép megjelenítő ablakot), és az új ablak neve legyen *x1* (ez tehát a paraméter értéke).

A példa második része a pontosvessző után következik, tehát az voltaképpen egy önálló, új parancs, csak éppenséggel kényelmesebb volt így egybeírni őket, mint két külön parancsként megadni (egyébként úgy is sikerült volna a megjelenítés, ki lehet próbálni).

```
d.rast map=roads
```

A parancs neve *d.rast*, a szóköz után jön paraméterként a *map=*, majd ezt követi az értékadás, vagyis az, hogy *roads*. Itt azonban álljunk meg egy pillanatra!

Az első példában a *d.mon* parancs paramétere az volt, hogy *start*, vagyis elindított egy megjelenítő-ablakot. De könnyen kitalálható, hogy lehetett volna mondjuk olyan paramétere is, hogy *stop*, vagyis,

hogyan bezárjon egy ablakot! Valóban: ha azt írjuk be, hogy `d.mon stop=x1`, az azt jelenti, hogy zárja be az `x1` nevű ablakot (feltéve persze, hogy az ablak létezik). Ebben az esetben, tehát a parancsnak valóban több paramétere is lehet és mi választjuk ki, hogy melyiket kívánjuk igénybe venni (de van még két másik is). Más a helyzet a `d.rast`-tal. Ez a parancs arra szolgál, hogy raszteres térképet jelenítsünk meg. Használatának csak akkor van értelme, ha adunk is meg térképet, amit megjelenítsen, így aztán a `map=paraméter` megadása kötelező, de persze az egyenlőségjel után szabadon adhatjuk meg a térkép nevét. Vagyis, vannak olyan parancsok, amelyeknek kötelező paramétere van, anélkül a parancs értelmezhetetlen.

A parancs futása megszakítható a terminálban a `ctrl+c` billentyűkombinációval. Próbáljuk ki! Ha az eddigiek után a terminálban állva `ctrl+c`-t nyomunk, az `x1` ablak bezárul, mert megszakadt a program futása.

Ha egy olyan parancsot, amelyhez mindenképpen kell valamilyen paramétert is rendelni, paraméter nélkül adunk ki, akkor vagy kiírja a lehetséges paramétereket, vagy elindít egy ablakot, amelyben megadhatjuk a kívánt paramétereket. Ha például csak ennyit írunk a terminálba:

```
d.mon
```

és utána `enter`t ütünk, megjelenik a `d.mon` nevű ablak, amelyben immár grafikus környezetben adhatjuk meg a szükséges paramétereket és értékeket. Ebben az esetben az ablak bezárása egyenértékű a `ctrl+c` billentyűkombinációval.

A parancsok tartalmazhatnak opciókat (*flag*) is, bár ez szó szerint fordítva zászlót jelent. Az opciók a parancsot finomítják, árnyalják a jelentését. Az opciók előtt mindig egy vagy két mínuszjel áll, vagyis `-` vagy `--`. Az opciókat a parancs neve után kell megadni, úgy, hogy a parancs után szóköz legyen. A mínuszjel(ek) után sosem kell szóköz.

Lássunk egy példát az opciók hatására!

Írjuk be, a már ismert módon:

```
d.mon start=x2;d.rast map=elevation.dem
```

Vagyis, elindítjuk az `x2` nevű ablakot, benne megjelenítjük az `elevation.dem` térképet. Ha betöltődött, írjuk be a terminálba ezt:

```
d.rast -o map=roads
```

Azt láthatjuk, hogy a `roads` nevű térkép most úgy jelenik meg, hogy a háttérben látható marad az `elevation.dem` is. Ez a `-o` opció hatására történt így. Ez az opció ugyanis azt jelenti, hogy overlay, vagyis, hogy a megadott térkép minden olyan rasztere, amely NULL értékű, átlátszó legyen. Pontosan megegyezik a *GIS Manager*-ben megadható `overlay`-jel. Láthattuk tehát, hogy az opcióval valóban módosíthattuk a parancs eredeti jelentését. Ha egy parancshoz több opció is választható, akkor azokat együtt is meg lehet adni.

Adjuk ki az alábbi parancsot!

```
d.mon -l
```

Az `enter` leütése után a terminálban kilistázva olvashatjuk mindazon monitorokat, amelyeket megadhatunk, mint értéket (ezek közül az egyik az `x1`).

Most írjuk be ezt:

```
d.mon -p
```

Az *enter* után olvashatjuk, hogy melyik monitor az éppen aktív: *Currently selected monitor: x2*, vagyis, az éppen aktuális monitor az *x2* (legalábbis, ha követtük a fentieket).

De ez a két opció együtt is kiadható lett volna:

```
d.mon -lp
```

Az eredménye pedig az, hogy megjelenik a választható nevek listája, alatta pedig az éppen aktuálisan használt monitor neve. Figyeljük meg, hogy a két opció együttes megadásához nem kellett mindkettő előtt - jelet tenni, elég volt csak az első előtt és szóköz nélkül, rögtön egymás után megadni az opciókat. Ez a szabály általános érvényűnek tekinthető.

Persze, az eddigiek össze is vonhatók egy sorba:

```
d.mon stop=x1;d.mon stop=x2;d.mon start=x3;d.mon -lp;d.rast  
map=landuse;d.rast -o map=roads
```

Ezzel leállítjuk (ha esetleg még futna) az *x1* és *x2* monitorokat, elindítjuk az *x3* monitort, kilistázzuk az elérhető monitorok neveit, valamint az éppen aktuális nevét, majd megjelenítjük rajta a *landuse* (területhasználati) térképet, végül e felett a *roads* térképet, úgy, hogy alatta a *landuse* látható maradjon.

Az itt leírtak általános érvényű elvek, a GRASS többi parancsa is ezeknek megfelelően épül fel.

Felvetődik a kérdés: honnan lehet megismerni egy parancs lehetséges paramétereit és opcióit, illetve a parancs további sajátosságait? A válasz egyszerű, írjuk be a terminálba

```
man d.rast
```

Ennek hatására megjelenik a *d.rast* parancs kézikönyve, mely tartalmazza a parancs leírását, összes paraméterét, opcióját és sajátosságait, valamint a parancs szerzőjét és az esetleges hibák jegyzékét, valamint a megjegyzéseket. Általában egy-egy példát is találunk a parancs használatára vonatkozóak. Vagyis, ha egy parancs használatára vagyunk kíváncsiak, akkor a *man* után, szóközzel elválasztva adjuk meg a nevét! (Ha például a *man* használatára vagyunk kíváncsiak, írjuk be, hogy *man man*!) A kézikönyvoldalon a nyíl billentyűkkel mehetünk fel, illetve le. Kilépní a *q* billentyű megnyomásával lehet.

További kérdés lehet, hogy honnan lehet megtudni, egyáltalán milyen parancsok állnak rendelkezésre? Milyen parancsokat adhatunk ki?

A GRASS parancsai úgy vannak elnevezve, hogy a név minimum két részből áll, melyeket pont választ el és azután jön a tulajdonképpeni név. Több részből is állhat a név, de mindig az első pont előtti rész utal a parancs jellegére (vagyis, olyan mint a kiterjesztés a DOS-ban, csak nem hátul van, hanem elől).

A GRASS parancsainak kezdő elemei:

- d.* display, vagyis a megjelenítéssel összefüggő parancsok (pl. *d.mon*, *d.rast*)
- db.* adatbázisműveleteket végző parancsok
- g.* a GRASS állományaira, beállításaira vonatkozó parancsok (pl. *g.copy*, *g.region*)
- i.* képfeldolgozást végző parancsok
- m.* kiegészítő műveletek parancsai

- photo.* Fényképek feldolgozását végző parancsok
- ps.* postscript művelet
- r.* raszteres műveleteket végző parancsok (pl. *r.mapcalc*)
- r3.* 3D raszteres műveletek
- v.* vektoros műveleteket végző parancsok (pl. *v.buffer*)
- wxGUI* a GRASS új, a 7-es verzióhoz készített grafikus felhasználói felületét indítja.
- gis.m* A grafikus felületet indítja
- d.m* A GRASS egy régebbi, végül nem bevezetett grafikus felületét indítja. Nagyon hasonlít a *gis.m-re*.
- NVIZ* Az NVIZ-t indítja, a GRASS 3D megjelenítő felületét.
- xganim* Az *xganim* nevű programot indítja.

### 2.8.1. Parancssori segédeszközök

A parancsok alkalmazását a GRASS-ban segíti a parancskiegészítés és az előzmények tárolásának használata. A parancskiegészítés azt jelenti, hogy elegendő a parancs nevének az első, vagy az első néhány karakterét begépelni, mert utána a *tab* megnyomására a terminál automatikusan felajánlja mindazon parancsokat, amelyek a megadott karaktersorozattal kezdődnek. Vagyis, nem is kell megtanulni a parancsokat, elegendő megjegyezni a kezdőbetűjüket, vagy a legfeljebb még egy-két kezdő karakterüket.

Írjuk be a terminálba a következőt:

```
d.
```

majd nyomjuk le kétszer a *tab*-ot (tehát nem az *entert*!)

Ennek hatására a terminálban kilistázódnak mindazon parancsok, melyek a *grass*-ból elérhetőek és *d.*-tal kezdődik a nevük. Vagyis, láthatjuk az összes olyan parancsot, amelyik a megjelenítéssel kapcsolatos. Így, ha valamiért nem jutna eszünkbe, hogy a raszteres térképek megjelenítésére a *d.rast* szolgál, vagy az, hogy pontosan hogyan is kell írni ezt a parancsot, most már könnyen előkereshetővé vált.

A *d.rast* kiadásához sem kell begépelnünk végig a parancs nevét, mert a parancskiegészítés később is működik. Írjuk be, hogy

```
d.r
```

és nyomjuk meg egyszer a *tab*-ot!

Nem történik semmi. Nyomjuk meg még egyszer (tehát végül is most is két *tab*ot nyomtunk)! Ezután a rendszer felajánlja mindazon parancsok neveit, amelyek *d.r* karakterekkel kezdődnek. Ez már per-sze rövidebb lista lesz, mint az előző volt.

Töröljük ki az eddig beírt *d.r* karaktereket (a *backspace*-szel vagy *ctrl+c*-vel) és írjuk be, hogy

```
d.ra
```

aztán üssünk egy *tabot*!

Most a terminál nem adott listát, hanem kiegészítette az az általunk beírtakat arra, hogy *d.rast*.

Mi történt? Mikor kell egy *tab*, mikor kell kettő?

Ha az általunk beírt karaktersorozat csak egyféleképpen folytatható, akkor a *tab* billentyű lenyomására azt addig egészíti ki a terminál, ameddig olyan karakterhez nem ér, ami után már több választás lehetséges.

Vagyis, mivel jelen esetben csakis azok a parancsok kezdődnek *d.ra* karaktersorozattal, amelyek egyébként *d.rast*-tal kezdődnek, ezért a *d.ra* begépelése után a terminál egy *tab* lenyomásával *d.rast*-ra egészít ki, hiszen csak ez az egy lehetőség van. Mivel azonban a *d.rast* után (vagyis a *t* betű után) már megint több dolog jöhetne ezért itt meg is állt. Ha tehát most megint lenyomjuk a *tab*-ot, akkor megmutatja a választási lehetőségeket. Ebben a példában ezeket:

```
d.rast      d.rast.arrow      d.rast.edit d.rast.leg  d.rast.num
```

Láthatjuk tehát, hogy valóban több választás van, ugyanis a *d.rast* után (pontosabban a *t* után) vagy semmi jön, ebben az esetben kész is a parancs, vagy pont és utána valami.

Általában tehát elmondható, hogy ha az általunk beírt karaktersorozat után közvetlenül többféle karakter is állhat, akkor a *tab* egyszeri lenyomására nem történik semmi, mert nincs egyértelmű folytatás, és azért kell még egyszer megnyomni, hogy láthassuk a választási lehetőségeket.

A *d.* kezdetű parancsok nincsenek túl sokan, ezért a lista kényelmesen elfér a terminálban. Előfordulhat azonban, hogy a lista túl hosszú, nem fér ki a terminál ablakban. Írjuk be most a következőt:

```
r.
```

És nyomjunk két *tab*-ot. (Vagyis kérjük az összes rászteres parancs megjelenítését)

Az eddigiektől eltérően nem jelenik meg a lista, hanem az alábbi üzenetet olvashatjuk:

```
Display all 156 possibilities (y/n)?
```

Vagyis: megjelenítsem mind a 156 lehetőséget (y/n)?

A (y/n) azt jelenti, hogy igen/nem, vagyis választhatunk, hogy megjelenítse mind a 156 parancsot a terminál, vagy inkább mégse. Ha az *y* gombot nyomjuk meg (y=yes=igen), akkor megjeleníti, ha az *n* gombot (n=no=nem), akkor nem. Ez utóbbi esetben visszakapjuk a parancskiadás lehetőségét.

Most nyomjuk meg az *y* gombot!

Mivel a 156 elemű lista nem fér el a terminálban, ezért görgetni kell a képernyőt. Erre utal a terminál alján olvasható -- *More* -- (vagyis: tovább) üzenet. Két lehetőségünk van. Az *enter* lenyomásával egy sorral lép fentebb a lista, a *space* (szóköz) megnyomására egy egész oldallal. Próbáljuk ki előbb az *enter*, aztán a *szóköz* hatását! Ha meg kívánjuk szakítani a listázást, használjuk a *ctrl+c* billentyűkombinációt!

Végül jegyezzük meg, hogy önmagában, mindenféle karakterek megadása nélkül, a *tab* kétszeri lenyomására a terminál kilistázza az összes választható parancsot. (Ez egy átlagos Linux/Unix gépen jellemzően 1500-3000 parancsot jelent.)

A terminál nem csak kiegészíti a parancsokat, hanem meg is jegyzi az előzményeket. Az előzményekben a *fel* és a *le nyilakkal* tudunk böngészni. Próbáljuk ki: a *felfele nyíl* billentyűt nyomkodva a korábban kiadott parancsainkat érhetjük el. Ha túlhaladtunk egy parancson, akkor a *lefele nyíllal* lehet visszalépni benne. A parancsok megjegyzése nincs időhöz kötve és a megjegyzett parancsok a gép kikapcsolás

lását követően is elérhetőek. Azaz akár arra is lehetőségünk van, hogy az adott gépen valaha kiadott összes parancsunkat visszakeressük.

A kikeresett parancsok természetesen újra kiadhatók, ehhez egyszerűen csak az *enter*t kell megnyomni. Az előzmények közül előkeresett parancsok szerkeszthetőek is, vagyis minden további nélkül átíráhatóak és így is futtathatóak. A parancs kiadásához nem kell a kurzorral a sor végér állni, mert akárhol is állunk a parancsban, az *enter* megnyomásával az futtatható lesz.

Próbáljuk ki ezeket!

Keressük meg a fel és le nyilak használatával a korábban már kiadott

```
d.mon stop=x1;d.mon stop=x2;d.mon start=x3;d.mon -lp;d.rast
map=landuse;d.rast -o map=roads
```

parancsot!

Ha megvan, akkor írjuk át úgy, hogy csukja be az *x3* ablakot is (ha nyitva lenne), indítsa el az *x4*-et és a *landuse* helyett az utak mögött az *elevation.dem* látszódjék! A művelethez használjuk a bal és a jobb nyilat, a törlés/visszatörlés gombot, mintha csak egy szokásos szövegszerkesztőben dolgoznánk. Ha kész a szerkesztés, akkor ne menjünk a sor végére, hanem csak nyomjunk *enter*-t!

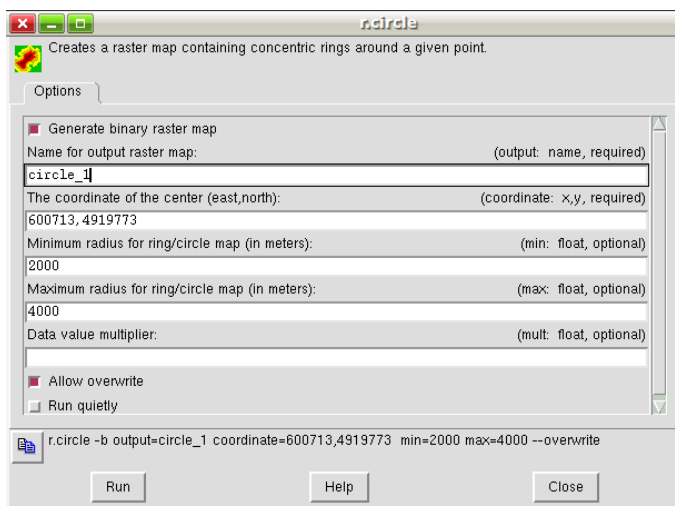
```
d.mon stop=x1;d.mon stop=x2;d.mon stop=x3;d.mon start=x4;d.mon
-lp;d.rast map=elevation.dem;d.rast -o map=roads
```

A parancsok használatának megtanulásához a GRASS két további segítséget is ad.

Az egyik az, hogy az Output ablakban mindenkor olvasható a végrehajtott művelet parancssoros megadása, akkor is, ha annak elvégzését a GUI-ból indítottuk, egérrel. A másik segítség pedig nem más, mint az, hogy minden olyan ablak alján, amelyből valamilyen műveletet tudunk indítani, megjelenik a parancssori kimenete annak, amit éppen beírtunk az egyes mezőkbe, vagy kiválasztottunk a különféle listákból és menükből. Lássunk a fentiekre példát is!

Használjuk most a GRASS grafikus felületét és végezzünk el egy viszonylag egyszerű feladatot!

Készítsünk egy térképet, amely az E=600713 és N=4919773 koordinátájú pont körüli 2000 m és 4000 m sugarú körök közötti terület domborzatát ábrázolja (természetesen továbbra is a *spearfish60* locationról van szó)! Ehhez válasszuk ki a *raster* menü *Concetric circles* (koncentrikus körök) pontját. A megjelenő ablakba írjuk be a kimenetként kívánt állomány nevét, a koordinátákat és adjuk meg a létrehozandó kör belső (minimum) és külső (maximum) sugarát. Ne feledjük engedélyezni a *Generate binary map* opciót! A kitöltött ablakot lásd az 2.26. ábrán. Figyeljük meg, hogy a mezők kitöltése közben az ablak alján szépen alakul a parancssor is. Érdekes megpróbálni többféle sorrendben is kitölteni a mezőket: a parancssor szerkezete ennek elle-



2.26. ábra  
Területek és centroidok

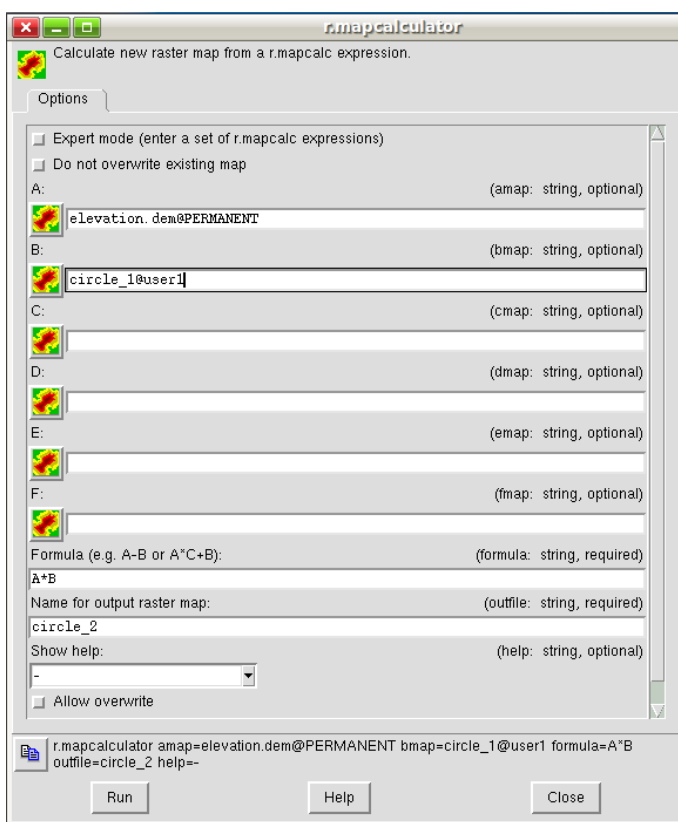
nére változatlan marad. Vagyis, nagyon egyszerűen tanulhatjuk meg, hogy ezt a parancsot hogyan kell használni, és ha valamiért nem jutna eszünkbe a használata, könnyen felfrissíthetjük a tanultakat. Mindenestre, ha kitöltöttük a mezőket, nyomjuk meg a *Run* gombot!

Ha elindítottuk a műveletet, váltsunk át az *Output – GIS.m* ablakra és figyeljük meg, hogy ott is megjelenik az elindított művelet parancssori alakja, szürkével kiemelve, rögtön a folyamatjelző csík fölött. Ha a művelet befejeződött, akár meg is jeleníthetjük az eredményt, bár a továbblépéshez ez nem szükséges. A következő lépésben összeszorozzuk az imént létrehozott térképet az *elevation.dem*-mel. Használjuk ehhez a *Raster* menü *mapcalculator* pontját! Válasszuk ki az *A* térképhez az *elevation.dem*-et, a *B* térképhez pedig az imént előállított térképünket. Gördítsük le az ablak tartalmát, hogy alul, a *Formula* (képlet) mezőbe beírassuk a műveletet, jelen esetben azt, hogy  $A*B$ . A *Name of output raster map* mezőben adjunk meg egy nevet a létrejövő új térkép számára (2.27. ábra)! Figyeljük meg, hogy a mezők kitöltögetése közben itt is folyamatosan olvasható a kiadandó parancs parancssori alakja és természetesen az előzőhöz hasonlóan, most is látható az *Output – GIS.m* ablakban. Ha a GRASS elvégezte a kért műveletet, meg is jeleníthető a kész mű (2.28. ábra).

Most már láthatjuk, hogyan lehet ezt a műveletet parancsokkal elvégezni, elegendő vagy a megfelelő ablakok aljából kiírni a parancsot, vagy az *Output – GIS.m* ablakból kimásolni. Ez utóbbi az egyszerűbb: kattintsunk a kimásolni kívánt parancsra, vagy bárhová ennek a sornak a szürke hátterén és az ablak alján rögtön megjelenik a parancs. Itt akár szerkeszthetjük is és ha kívánalmainknak megfelelően átalakítottuk, a *Run* gombbal ismét futtathatjuk.

Ha azonban azt akarjuk, akkor ki is másolhatjuk: jelöljük ki egérrel a kimásolandó részt, majd nyomjuk meg a *ctrl+c* (vagy *alma+c*) gombot. A leírtaknak megfelelően tehát másoljuk ki előbb a kört létrehozó *r.circle* parancsot teljes terjedelmében és illesszük be a terminálba, majd tegyünk is utána rögtön egy pontosvesszőt. Ezután illesszük mellé a második művelet parancsát ugyanígy:

```
r.circle -b output=circle_1 coordinate=600713,4919773 min=2000
max=4000;r.mapcalculator amap=circle_1@user1
bmap=elevation.dem@PERMANENT formula=A*B outfile=circle_2 help=-
```



**2.27. ábra**  
A szükséges paraméterek megadása a mapcalculator-ban  
(részletesen lásd a szövegben)

Fontos megfigyelni, hogy a *coordinate* változó két értékét vessző választja el egymástól! Ez a GRASS-ban általános érvényű szabály: ha több értéket adunk meg egymás után, azokat minden esetben vesszővel kell elválasztani egymástól, de sem a vessző előtt, sem utána nem állhat szóköz! Ebből következően tizedes törteket a GRASS-ban (az angolszász gyakorlatnak megfelelően) tizedesponnttal tagoljuk, nem vesszővel! Vagyis, ha jelen esetben a koordináták tizedes törtek lettek volna, akkor valami ilyesmit kellett volna írunk: *coordinate= 600713.5,4919773.35*

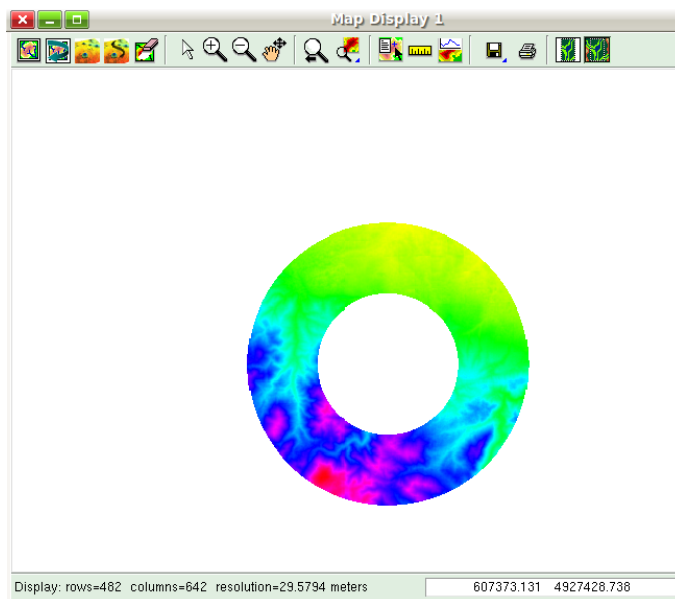
Ha ezt a viszonylag összetett utasítást futtatnánk az *enter* lenyomásával, akkor ugyanazt kapnánk, mint amit az előbb a grafikus felületen, az egér segítségével állítottunk elő. De csak így, feltehetően módban, mert ha valóban végre kívánnánk hajtani a megadott parancsokat, akkor hibüzenet lenne az eredmény. Arról van szó, hogy mivel a GUI-t használva egyszer már létrehoztuk a *circle\_1* és a *circle\_2* térképeket, ezért a parancsban nem adhatjuk meg ezek neveit még egyszer, mert ugyanazon a néven nem hozhatunk létre újból állományokat. Vagy azt kell tennünk, hogy megváltoztatjuk a parancsban a neveket (mondjuk lehetne *circle\_1b* és *circle\_2b*) vagy engedélyezzük a felülírásukat. A felülírás engedélyezése azt jelenti, hogy a parancsban foglaltakat a GRASS akkor is végrehajtja, ha a kimenetül adott neven már létezik térkép az adott mapsetben. Erre a *--overwrite* kapcsoló szolgál. Ez bármelyik olyan parancs után illeszthető, amelyikben kimeneti állománynevet is megadtunk (egyébként teljesen megegyezik a grafikus felület ablakaiban használható *Allow overwrite* opcióval). Használata egyszerű, csak annyit kell tenni, hogy a megadott parancs legvégére odaírjuk, hogy *--overwrite*. Természetesen figyeljünk rá, hogy előtte legyen egy szóköz! Jelen esetben tehát:

```
r.circle -b output=circle_1 coordinate=600713,4919773 min=2000
max=4000 --overwrite;r.mapcalculator amap=circle_1@user1
bmap=elevation.dem@PERMANENT formula=A*B outfile=circle_2 help=
--overwrite
```

Így kiadva a parancsot már biztosan lefut és létrejönnek a kívánt állományok.

### 2.8.1.1. Segítség a parancsok használatához

A parancsok használatának elsajátításához komoly segítséget ad, hogy minden parancsnak van úgynevezett *man* lapja, vagyis kézikönyv oldala (*man*=manual, használati utasítás). Ez tartalmazza az adott



2.28. ábra  
Az elkészült kör és az *elevation.dem* szorzata

parancs használatának részletes leírását, összes opcióját, paraméterét, kapcsolóját és általában példákat is tartalmaz a használatára vonatkozóan.

A *man* használata igen egyszerű, mert csak azt kell beírni a terminálba, hogy *man*, majd egy szóköz után annak a parancsnak a nevét, amelynek a használatára kíváncsiak vagyunk. Ha a GRASS termináljából indítjuk a *man*-t, akkor GRASS saját parancsain kívül az adott rendszeren levő összes parancs nevét is megadhatjuk. Ha a nem a GRASS termináljából indítjuk a *man*-t, hanem máshonnan, akkor csak a rendszer parancsainak *man* lapjait érjük el. Írjuk be a példa kedvéért a GRASS termináljába ezt:

```
man d.rast
```

Az *enter* leütése után a *d.rast* parancs *man* lapját olvashatjuk. A *man* lapon a le- és a fel nyílal közlekedhetünk, kilépni pedig a *q* gomb lenyomásával lehet. Ha például magának a *man*-nak a kezeléséről szeretnénk többet megtudni, akkor írjuk ezt:

```
man man (enter).
```

További segítség, hogy a GRASS parancsainak használata grafikus felületen is elérhető, ha az éppen használt ablak alján levő *Help* gombra kattintunk. Tehát, ha például a fent már mutatott NULL-érték kezelés ablakot megnyitjuk (*Raster/Develop map/Null values*), (figyeljük meg, hogy az ablak neve: *r.null*, vagyis magának a parancsnak a neve!) akkor az ablak alján levő *Help* gomb megnyomására megjelenik az *r.null* parancs leírása egy ablakban.

Természetesen a parancsok használatához további bőséges irodalom érhető el az interneten is. Célszerű a <http://grass.osgeo.org/> vagy a [grass.itc.it](http://grass.itc.it) oldalon kezdeni a keresést.

### 2.8.2. Automatizálási lehetőségek

Az egész fenti gondolatmenet fő célja nem egyéb, mint az, hogy végül a munkánkat automatizálhassuk, amennyire csak lehet, illetve, amennyire célszerű. Ennek mikéntjéről ugyan lesz még szó később, de az alapvető funkciók már most, ennyi ismeret birtokában is elérhetőek. Akik jártasak a *bash* scriptek világában, azok számára az automatizálás nagyon egyszerű lesz, mert a GRASS terminálban valójában ilyen scripteket futtathatunk.

Az automatizálás lényege az, hogy a parancsok formájában megfogalmazott utasítások elmenthetők egy egyszerű szöveges állományba (ASCII fájlba) és ha ezt a fájlt futtadjuk, akkor a benne tárolt utasításokat hajtja végre a GRASS. Ennek a megoldásnak több előnye is van:

- az elvégzett műveletek tárolva vannak, könnyen visszakereshető, hogyan készült a térkép;
- a műveletek akárhányszor megismételhetők más értékek megadásával, így nemcsak kísérletezni lehet gyorsan, hanem a sok ismétléssel járó munkák is egyszerűbben elvégezhetőek lesznek;
- mivel a parancsok egyetlen egyszerű ASCII állományban vannak tárolva, ezért azokat minden további nélkül átvihetjük más operációs rendszer alatt futtatott GRASS-ba is, és legfeljebb csak egészen kis változtatásokat kell bennük végrehajtani (bár általában semennyit sem).

Ilyen állomány létrehozása nagyon egyszerű. Nincs más teendőnk, mint elindítani egy egyszerű szövegszerkesztőt (Windows alatt a *jegyzetömböt*, Apple környezetben a *TextEdit*-et, Linux alatt pedig valamelyik szövegszerkesztőt, mondjuk a *gedit*-et, vagy a *vim*-et) és a parancsokat abba írni, vagy abba má-

solni az *Output-GIS.m* ablakból, az előzőekben bemutatottak szerint járva el. Ha ezzel készen vagyunk, akkor mentjük el ezt a a szöveges állományt és tegyük futtathatóvá (vagyis adjunk neki futtatási jogot)! Ezután nincs más dolgunk, mint a GRASS terminálban megadni ennek az állománynak a nevét, elérési úttal együtt, majd az *enterrel* elindítani a műveletek elvégzését.

Az alábbiakban látható egy példa az elmondottak illusztrálására. A script eredményeként előállított térképek a 2.29. ábrán találhatók.

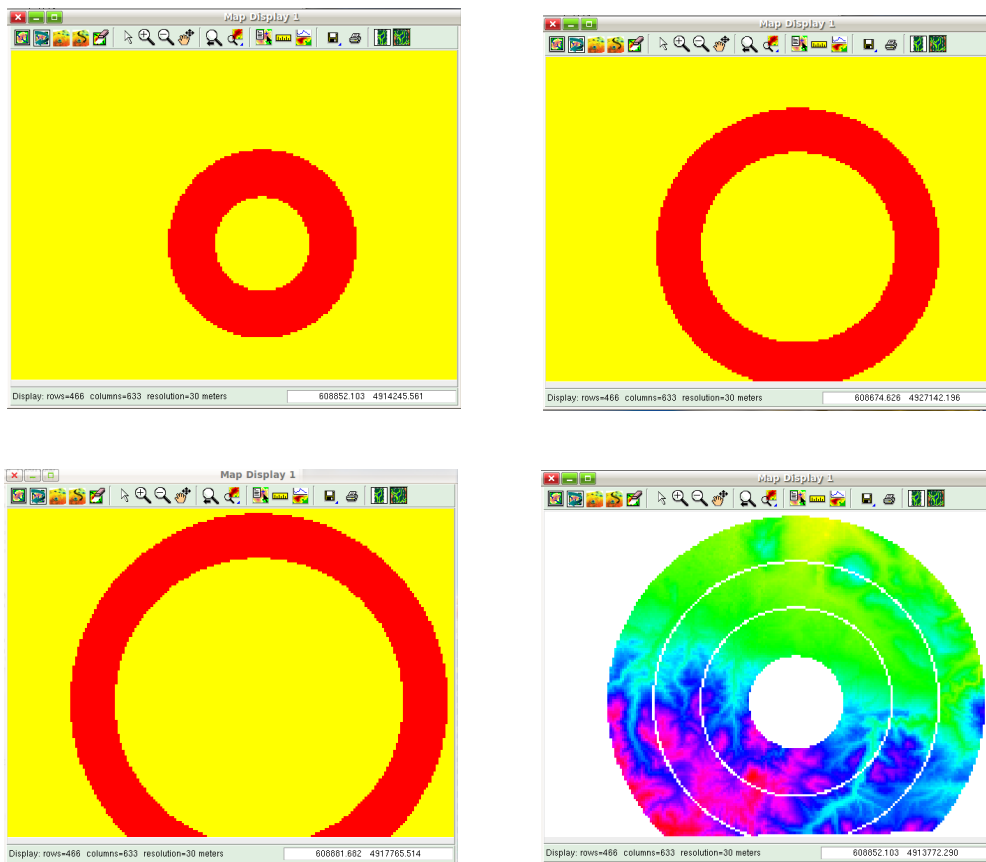
```
g.region rast=elevation.dem@PERMANENT nsres=100 ewres=100;
r.circle -b output=circle_1 coordinate=600713,4919773 min=2000 max=4000 --overwrite;
r.circle -b output=circle_2 coordinate=600713,4919773 min=4100 max=6000 --overwrite;
r.circle -b output=circle_3 coordinate=600713,4919773 min=6100 max=8000 --overwrite;
r.null map=circle_1@user1 null=0;
r.null map=circle_2@user1 null=0;
r.null map=circle_3@user1 null=0;
r.mapcalculator amap=circle_1@user1 bmap=circle_2@user1 cmap=circle_3@user1 dmap=elevation.dem@PERMANENT formula=A*D+B*D+C*D outfile=circle_4 --overwrite;
r.null map=circle_4 setnull=0;
r.out.png input=circle_4 output=circles.png --overwrite
```

Értelmezése a következő:

- A script először úgy állítja át az aktuális régióbeállítását, hogy az észak-déli és a kelet-nyugati felbontás is 100 m legyen. Erre azért van szükség, mert így sokkal gyorsabban elkészül a végeredmény, mint az eredeti, jobb felbontásban. Persze így a minősége is rosszabb lesz, de gyakorlásnak ez is teljesen megfelel. Természetesen az *nsres=* és az *ewres=* paramétereknek más értékek is adhatók, tetszés szerint.
- Ezután elkészít három kört, melyek középpontja egybeesik (vagyis koncentrikusak), és közöttük 100 méter távolságot hagy. E körök neve rendre *circle\_1*, *circle\_2* és *circle\_3*. E körök binárisak (ezt jelzi a *-b* kapcsoló), vagyis a körök területe 1, a többi terület pedig NULL értékű.
- A továbbiakban először a NULL értékeket változtatjuk 0-ra (nullára), mert ez szükséges a további műveletek elvégzéséhez. Erre szolgál az *r.null* parancs alkalmazása.
- Ezután e három kört összeadjuk és az eredményt összeszorozzuk a PERMANENT mapsetben található *elevation.dem*-mel, az eredményt pedig *circle\_4* néven mentjük.
- A továbbiakban visszaállítjuk a 0-t NULL-ra, az eredményül kapott térképet pedig exportáljuk egyszerű képként, png formátumban, *circles.png* néven.

Figyeljük meg, hogy ahol csak kellett, mindenütt *--overwrite*-tal végződnek a műveletek. Ennek az oka, hogy így a paraméterek megváltoztatásával egyszerűen lehet kísérletezni, anélkül, hogy újabb néven jönnének létre térképek. Egyébként erre a kísérletezésre bátorítom is az olvasót: gépelje be az állományt, futtassa és változtatgassa meg az egyes értékeket, aztán figyelje meg mi történik!

Fontos megjegyezni, hogy a script szövegszerkesztőbe illesztésekor (vagy gépeléskor) figyeljünk arra, hogy a sortörések a pontosvesszők *után* legyenek, ne pedig sor közben!



2.29. ábra

A bemutatott scrip-tel készített körök (bal fent: *circle\_1*, jobb fent: *circle\_2*, bal lent: *circle\_3*, jobb oldalt lent pedig a végeredmény látható, a *circle\_4*).

Ha futtatjuk az állományt, a terminálban folyamatosan nyomon követhetjük a munka állását, mert nem csak az látható, hogy az éppen folyó művelet hol tart, hanem az is, hogy eddig mi készült el (persze, egy kicsit is gyorsabb gépen erre nem nagyon lesz időnk).

Természetesen az eddigiekben tárgyaltak felhasználásával ennél sokkal összetettebb fájlok is készíthetők, vagy tetszés szerinti műveletek automatizálhatók.

Ne feledjük: ha egy általunk elkészített fájlt futtatunk, akkor a futtatás bármikor megszakítható a terminálban a *ctrl+c* billentyűkombinációval!

## 2.9. Adattárolás a GRASS-ban: könyvtárszerkezet és fájlok

A GRASS a felhasználói állományok kezelésére és tárolására összetett, viszonylag bonyolult könyvtárszerkezetet használ. Ennek számos előnye van, ugyanakkor kétségtelenül van legalább egy hátulütője is a dolognak: megfelelő ismeretek nélkül nehezebb benne eligazodni. Természetesen az eligazodás megtanulható, így a könyvtárszerkezet adta előnyök maradéktalanul kihasználhatóak, az esetleges problémák pedig elháríthatóak. Mindenek előtt ugyanis el kell mondani, hogy előfordulhatnak olyan problémák, amelyek elhárítása az e fejezetben foglaltak ismerete nélkül nem háríthatók el, ennek megfelelően javasolt legalább egyszeri áttanulmányozása.

De nem csak azért hasznos ismerni az adattárolás könyvtár- és állományszintű struktúráját, mert így könnyebb a hibaelhárítás. Vannak ugyanis olyan műveletek, amelyek akár a GRASS elindítása nélkül is elvégezhetők, illetve az alábbiak ismeretében a GRASS-ban egyébként szokásosnál könnyebben is elvégezhetők. Ilyenek például a vektor-adatbázis csatolások, vagy a vetületi információk megadása.

Előzetesen röviden az alábbiakat mondhatjuk el a GRASS állománytárolási struktúrájáról:

- viszonylag bonyolult, többszintű könyvtárszerkezet jellemzi, igen sok állománnyal;
- az összes beállítóállomány ASCII típusú;
- a vektoros és a raszteres állományok könyvtárszinten is világosan elkülönülnek;
- egyes fontos beállítások akár a GRASS elindítása nélkül is módosíthatók, illetve kiolvashatók.

A GRASS indításakor, mint azt már tudjuk, meg kell adni a használni kívánt térinformatikai adatbázist, a locationt és a mapsetet. Azt is tudjuk, hogy ezek valójában könyvtárak, mégpedig úgy, hogy legmagasabb szintű az adatbázis, ennek alkönyvtára a location és ennek alkönyvtára a mapset. Ha tehát a GRASS indításakor azt választjuk, hogy az adatbázis helye legyen a *home/user/grass*, akkor ez azt jelenti, hogy a továbbiakban a */home/user/grass* nevű könyvtárban levő locationok (alkönyvtárak) közül kell választanunk egyet. Ha mondjuk innen a *spearfish60* locationt választjuk, akkor valójában a *spearfish60* könyvtárat választottuk ki és az itt levő mapsetek egyikében fogunk dolgozni, mondjuk a PERMANENT-ben, vagyis a PERMANENT könyvtárban. Eddig nincs is semmi bonyolult a dologban, egyszerűen követhető mindez egy tetszőleges fájlkezelőben is. Az eddigiekből látható, a */home/user/grass/spearfish60/PERMANENT* könyvtár tekinthető úgy is, mint abszolút elérési út, mely a munkánk során voltaképpen használt állományokhoz vezet. És éppen erről van szó: az összes olyan állomány, amellyel egy adott mapsetben dolgozva összeakadhatunk, az adott mapsetben, vagyis a mapset nevét viselő alkönyvtárban helyezkedik el. Nézzük meg tehát, mi is van voltaképpen a *spearfish60* location PERMANENT alkönyvtárában! Figyelem: a továbbiakban foglaltak követéséhez érdemes lehet egy fájlkezelőben követni a leírtakat!

Fontos megértenünk, hogy mi a jelentősége a PERMANENT-nek. Ez a mapset tartalmazza az adott location egészére vonatkozó beállításokat, éppen ezért kötelező a megléte. Azt, hogy valamely locationban csak ugyanazon vetületi rendszerű térképek lehetnek, melyek alapértelmezett régiója is pontosan ugyanaz, az biztosítja, hogy a PERMANENT-ben megvannak a szükséges metaadatok. Éppen ezért ugyanazon location többi mapsetjében már nem kell megadni minden metaadatot, csak azokat, amelyek-

ben a mapset eltér a többitől. Érdekes lehet megfigyelni a spearfish60 location PERMANENT és user1 mapsetje közötti különbségeket!

A PERMANENT mapset tartalmát megtekintve azt látjuk, hogy abban további alkönyvtárak, valamint néhány egyéb fájl található, mégpedig ezek:

.tmp	PROJ_UNITS	colr
.bash_history	TIGER	colr2
.bashrc	TODO	dbf
COMBINE	VAR	dlg
Census	WIND	fcell
DEFAULT_WIND	bdlg	g3dcell
HISTORY	cats	hist
Informix	cell	paint
MYNAME	cell_misc	vector
PROJ_INFO	cellhd	windows

Az önálló fájlok mindegyike egyszerű ASCII (szöveges) állomány és ezek tartalmazzák az adott mapset alapvető környezeti beállításait. Meglétük és helyes belső felépítésük alapvető fontosságú a kérdéses mapset használhatósága szempontjából.

Jegyezzük meg, hogy a könyvtárak bármelyike előfordulhat bármelyik mapsetben, de a fájlok közül – melyek metadatokat, beállításokat tartalmaznak – csak néhánynak van helye a PERMANENTEN kívül más mapsetben is.

név	Csak a PERMANENT-ben	Más mapsetben is
.bash_history		*
.bashrc		*
DEFAULT_WIND	*	
HISTORY	*	
MYNAME	*	
PROJ_INFO	*	
PROJ_UNITS	*	
TODO	*	
VAR		*
WIND		*

- **.bash\_history** – Rejtett állomány, mivel pont-tal kezdődik a neve. Az adott mapsetben valaha kiadott parancsokat tárolja. Így egyrészt visszakereshető, hogy mely parancsot adtuk ki, milyen opciókkal és paraméterekkel, másrészt lehetővé teszi a parancselőzmények előhozását a nyíl billentyűvel (lásd parancssori segédeszközök című részben!)
- **.bashrc** – Rejtett állomány, a használható parancssori környezetre vonatkozó beállításokat tárolja.
- **DEFAULT\_WIND** – Ez az állomány tartalmazza az adott locationra vonatkozóan az alapértelmezett régió beállításait.
- **WIND** – Ez a fájl tartalmazza a mapset éppen aktuális régióbeállításának adatait (hiszen az aktuális régiónak nem kell ugyanannak lennie, mint aza alapértelmezettnek).
- **MYNAME** – A location nevét, leírását tartalmazó ASCII állomány.
- **HISTORY** – A location létrehozására vonatkozó metaadatokat tartalmazza.
- **PROJ\_INFO** – A location vetületét, illetve a vetületi információkat tartalmazza (pl. geodéziai dátumot).
- **PROJ\_UNITS** – A location vetületéhez tartozó mértékegység(ek)et adjuk meg ebben az állományban.
- **TODO** – A locationnal kapcsolatos további teendőket, a fejlesztés várható irányait adhatjuk meg itt.
- **VAR** – Azt mutatja meg, hogy a kérdéses mapsetre vonatkozóan mi az alapértelmezett adatbázis-meghajtó (pl. dbf), valamint mi az alapértelmezett adatbázis (vagyis az adattáblák alapértelmezett elérési útja).

### 2.9.1. Könyvtárak a mapsetben

- **cellhd/** – a raszteres térképek fejléceit tartalmazza, minden térképhez egyet;
- **cell/** – az egész szám (integer) típusú raszteres térképek adatmátrixai (vagyis a voltaképpeni raszteres térinformatikai modellek) vannak ebben a könyvtárban;
- **fcell/** – a lebegőpontos-szám (floating point) típusú raszteres térképek adatmátrixait tartalmazza;
- **grid3/** – a 3D típusú raszteres térképek adatmátrixait tartalmazza, függetlenül attól, hogy egész- vagy lebegőpontos számbázisúak;
- **hist/** – a raszteres állományok létrejöttével kapcsolatos metaadatokat találjuk itt, minden raszteres térképhez egy fájlt.
- **cats/** – a raszteres térképekhez csatolt szöveges vagy numerikus címkék (kategória állományok) helye;
- **colr/** – a raszteres állományokhoz opcionálisan rendelhető szín táblák helye;

- **cell\_misc/** – A raszteres állományokra vonatkozó kiegészítő információk helye: időbélyeg, érték-tartomány, kvantálási szabályok lebegőpontos állományokhoz, NULL (nincs adat) értékek;
- **vector/** – A vektoros térképekhez tartozó állományokat ebben a könyvtárban érhetjük el. Itt minden egyes vektoros térképnek saját alkönyvtára van és ebben vannak az alábbi állományok:
  - *head*: a vektoros térkép ASCII formátumú fejléce, mely tartalmazza a térkép létrehozásának idejét, továbbá a méretarányát és a pontosságát;
  - *coor*: bináris állomány, ez a tulajdonképpeni vektoros térkép, a grafikus elemek koordinátáit tartalmazza;
  - *topo*: bináris állomány, a grafikus elemek térbeli viszonyait leíró topológiai adatokat tartalmazza;
  - *hist*: az adott vektoros állomány létrejöttével kapcsolatos metaadatokat tartalmazza;
  - *cidx*: bináris állomány, a kategória indexeket tartalmazza, vagyis azt, hogy melyik vektoros elem melyik sorhoz kapcsolódik az adattáblában;
  - *dbln*: ASCII állomány, azt mutatja meg, hogy az adott vektoros térképhez melyik adattábla van csatolva.
- **COMBINE** –
- **Census** – Az USA népszámlálások térinformatikai formátumának támogatására szolgál, az ilyen formátumú állományok tárolási helye.
- **Informix** –
- **TIGER** – Tiger típusú térinformatikai állományok helye.