



GIS - GRASS

Wprowadzenie do systemu

Paweł Netzel



Wrocław 2000

1. Wstęp

Celem tego tekstu jest zapoznanie czytelnika z techniką GIS na przykładzie systemu GRASS. Dobierając zakres prezentowanych zagadnień przyjąłem założenie, że czytelnik posiada elementarne umiejętności z zakresu obsługi komputera, nie boi się go włączyć oraz (w przypadku pracy w systemie wielodostępnym) ma swoje konto i umie zarejestrować się w systemie podając nazwę użytkownika i hasło. Oczywiście zakładam, że czytelnik ma dostęp do komputera z pracującym systemem GRASS.

Cała praca podzielona jest na następujące części:

- wprowadzenie (które jest na szczęście krótkie);
- opis podstawowych czynności wykonywanych w trakcie jednej, pełnej sesji w systemie GRASS;
- opis wybranych poleceń systemu GRASS.

Ponadto w dodatkach czytelnik znajdzie: listę poleceń systemu GRASS, informacje o programie „awk”, opis łączenia się z serwerem przy użyciu X-Terminala.

Mam nadzieję, że po zapoznaniu się z tym tekstem będzie w stanie samodzielnie poruszać się po systemie GRASS. Będzie też w stanie, nie wdając się w zawłościami teorii GIS, posługiwać się takim systemem przy własnych analizach przestrzeni geograficznej.

Aby zapewnić dostęp do oprogramowania osobom, które pierwszy raz spotykają się z programem GRASS i nie mają doświadczenia z instalowaniem oprogramowania w systemie UNIX, ani nie mają kolegów, krewnych, znajomych, którzy mogliby im pomóc, przygotowałem CD-ROM zawierający prostą dystrybucję systemu Linux z wstępnie zainstalowanym programem GRASS.

Może pojawić się tu pewien kłopot, gdyż w zależności od posiadanej karty graficznej będzie możliwe wykorzystanie okien do prezentacji graficznej lub też nie. Zależy to od tego czy uda się uruchomić system X-Windows (polecenie „startx”).

System X-Windows można skonfigurować z linii poleceń za pomocą komendy „xf86config”.

Życzę wszystkim powodzenia przy pracy nad własnymi analizami przestrzennymi.

2. Systemy informacji geograficznej - GIS

2.1. GIS - co to jest

Istnieje wiele definicji systemu GIS (*geographical information system* – system informacji geograficznej). Począwszy od określania nazwą GIS programu do pracy nad mapami, poprzez definiowanie systemu GIS jako zespołu program-komputer-człowiek ukierunkowanego na gromadzenie i przetwarzanie danych o przestrzeni geograficznej, aż do określania mianem GIS abstrakcyjnego modelu bazy danych przestrzennych.

Tu posłużymy się definicją określającą GIS jako: system do gromadzenia i przetwarzania przestrzennej informacji o środowisku geograficznym w zadanym układzie odniesienia. Sprowadza to GIS do systemu gromadzącego informacje o abstrakcyjnej przestrzeni dwuwymiarowej.

2.2. Informacje gromadzone w systemie GIS

W systemie GIS są gromadzone informacje o szeroko rozumianej przestrzeni geograficznej. Będą to zarówno informacje o zjawiskach występujących w przyrodzie w sposób ciągły (pole ciśnienia, wysokości punktów w terenie nad poziom morza), informacje o zadanych obszarach (własności działek, geologia) jak i informacje o obiektach rozmieszczonych w terenie. Wszystkie te informacje mogą być przechowywane w różnych postaciach (w postaci rastrowej, wektorowej, obiektowej itp.).

2.2.1. Systemy wektorowe

Wśród systemów GIS można wyróżnić systemy ukierunkowane na gromadzenie informacji w postaci wektorowej. Baza danych takiego systemu zawiera informacje o kształcie danego elementu oraz jego atrybuty. Systemy takie są jak gdyby naturalnym przeniesieniem mapy w świat komputera. Niestety ta forma gromadzenia danych ma też swoje minusy. Prowadzenie zaawansowanych analiz przestrzeni geograficznej, nakładanie na siebie poszczególnych pól i obliczanie buforów czy też podobne czynności prowadzą do bardzo złożonych obliczeń.

2.2.2. Systemy rastrowe

Innym sposobem gromadzenia danych jest ich przechowywanie w systemach rastrowych. Tego typu systemy przechowują informacje zebrane z regularnej siatki punktów (najczęściej kwadratowej choć są też systemy oparte na trójkątach lub siatce heksagonalnej). Ta forma przechowywania informacji pozwala na bardzo szybki dostęp do danych oraz ich przetwarzanie. Problemem jest tu rozdzielczość, a dokładniej mówiąc rozmiar pola elementarnego. Im siatka gęstsza tym lepsze odwzorowanie przestrzeni geograficznej w bazie danych (ale tym większe zbiory).

Ciekawą sprawą jest to, że jak wykazują eksperymenty, systemy rastrowe jak i wektorowe są sobie równoważne pod względem objętości baz danych i szybkości dostępu. Różni je jedynie czas i pracochłonność przygotowania danych do wprowadzenia do bazy.

2.2.3. Systemy warstwowe

Zarówno systemy rastrowe jak i wektorowe najczęściej są też systemami warstwowymi. Co to znaczy? Znaczy to, że poszczególne aspekty przestrzeni geograficznej przechowywane są w oddzielnych warstwach tematycznych (warstwa zawierająca użytkowanie, ciekі, model wysokościowy itp.). Poszczególne warstwy można na siebie nakładać, liczyć pokrywanie się odpowiednich obszarów, ich przekroje, dopełnienia różnice.

2.2.4. Systemy obiektowe

Przeciwieństwem systemu warstwowego jest system obiektowy. Nie rozróżniamy w nim informacji tematycznej. W takim systemie informacja przechowywana jest w postaci obiektów. Obiekt zawiera nie tylko informacje o lokalizacji przestrzennej (w postaci wektorowej lub rastrowej) i atrybutach, ale także informacje o swoim zachowaniu, oddziaływaniach itp.

Prawie każdy system GIS zawiera w sobie trochę każdego z powyższych aspektów. Są systemy bardziej ukierunkowane na pracę w oparciu o bazę wektorową, są też takie

które wykorzystują przede wszystkim rastry. Są też takie, które rozwijają się w kierunku wykorzystania obiektów.

2.3. System GIS a systemy generowania map

Nie należy mylić systemów GIS z systemami do generowania map. Nie należy też zawężać ich wykorzystania tylko do zastosowania ściśle kartograficznego. Takie traktowanie systemu GIS sprowadza go jedynie do (bardziej lub mniej) inteligentnej deski kreślarskiej.

Wydaje mi się, że główną zaletą systemów GIS jest (i powinna być) możliwość prowadzenia szeroko rozumianych analiz i modelowań przestrzennych. Jeżeli system dodatkowo pozwala na wyprowadzenie informacji w postaci tradycyjnej mapy, to i bardzo dobrze. Jednak najważniejszą sprawą jest łatwość operowania danymi o przestrzeni.

3. System GRASS

3.1. System GRASS - co to jest

Czym jest system GRASS? Na to pytanie można odpowiedzieć następująco: jest to system GIS pozwalający na operowanie danymi rastrowymi, wektorowymi, na przetwarzanie obrazów, a w efekcie końcowym pozwala na wyprowadzanie rezultatów pracy w postaci graficznej (np. mapy). System ten pracuje w środowisku UNIX i pozwala na wykorzystanie graficznego interfejsu X-Windows.

Sama nazwa GRASS (po polsku trawa lub trawka ;-)) jest akronimem pełnej nazwy angielskiej: *Geographical Resource Analysis Support System*, czyli system obsługujący analizę zasobów geograficznych.

GRASS jest systemem publicznie dostępnym. Oznacza to, że można go używać do własnych zastosowań bez konieczności ponoszenia kosztów zakupu systemu komercyjnego GIS. Jednak pod względem możliwości wcale takim systemom nie ustępuje. Znany jest tekst źródłowy oprogramowania GRASS. Pozwala to na rozwijanie, modyfikowanie i przenoszenie do innych systemów programu GRASS przez szerokie rzesze programistów z całego świata. Rezultatem takich prac jest wersja binarna systemu GRASS pracująca pod kontrolą systemu Linux. Jeżeli wziąć pod uwagę, że Linux jest darmową wersją systemu UNIX, można przy ZEROWYCH nakładach finansowych stworzyć sobie na biurku wydajną stację roboczą GIS.

Wydaje się, że jedynym ograniczeniem systemu GRASS jest to, że program ten gromadząc dane w warstwach rastrowych pamięta je jako liczby całkowite. W praktycznych zastosowaniach okazuje się, że nie jest to żadna przeszkoda. Model wysokościowy DEM (digital elevation model) nie jest przecież pamiętany z dokładnością do tysięcznych części milimetra, nachylenia są pamiętane w pełnych stopniach, a opady w milimetrach. W miarę potrzeby można przecież uzyskiwać większą precyzję przemnażając wartości przez odpowiedni współczynnik (najczęściej 10 lub 100). Wiele poleceń GRASS uwzględnia takie postępowanie, pytając o przelicznik do metrów (lub innej wykorzystywanej w danej operacji jednostki

podstawowej). Równocześnie należy mieć świadomość, że operacje na liczbach całkowitych są najszybszymi operacjami wykonywanymi przez procesor. Daje to niezwykle szybkość systemowi GRASS

3.2. Historia systemu GRASS

Prace nad systemem GRASS rozpoczęły się 1982 roku. Powstawał on przy udziale wielu amerykańskich agencji rządowych, uniwersytetów i firm prywatnych. Bazą dla tego systemu były prace USA-CERL (U.S Army - Construction Engineering Research Laboratory). USA-CERL zakończył swoje prace na wersji systemu o numerze 4.1. Nastąpiło to w roku 1992. Do roku 1995 pojawiło się jeszcze pięć łatek i uaktualnień systemu.

Obecnie prace nad systemem GRASS są prowadzone przez GRASS Research Group z Baylor University. Całość prac nad systemem została przeniesiona na ten uniwersytet i obecnie właśnie tam znajduje się „centrala” GRASS.

Prace nad wersją 4 GRASS obecnie miały zostać zakończone. Ostatnia „czwórka” powinna nosić numer 4.2.1 v 27. Jednak prace nad tą wersją (całkowitoliczbową) są nadal kontynuowane. Pojawiła się obecnie nowa wersja o numerze 4.3.

W fazie testów jest obecnie wersja piąta systemu GRASS (zmiennoprzecinkowa). W chwili pisania tego tekstu można było się zapoznać i przetestować jego wersję 5 beta 6.

3.3. Wymagania systemu GRASS

GRASS dostępny jest zarówno w wersji źródłowej jak i skompilowanej. Z założenia przeznaczony jest on do pracy w systemie UNIX. Istnieją jednak wersje systemu GRASS dla Windows 9x/NT. Są to jednak pakiety komercyjne. Wersje skompilowane i gotowe do natychmiastowego użytku można znaleźć dla systemów takich jak SUN Solaris, DEC Alfa OSF/1, HP UX, SCO UNIX, SGI Irix oraz dla systemu Linux.

Wymagania wersji systemu GRASS dla środowiska Linux nie odbiegają znacząco od wymagań samego systemu operacyjnego. Jako sprzęt pracujący wydajnie można

wykorzystać komputer PC z procesorem 486 i 32 MB pamięci RAM (minimalne wymagania to 386SX i 8MB pamięci) oraz 50 MB wolnego miejsca na dysku.

3.4. Koncepcja i struktura systemu GRASS

GRASS należy do grupy programów GIS pracujących w oparciu o strukturę warstw. Nie jest on pojedynczym programem, ani nawet złożoną lecz odrębną aplikacją. Należy go raczej postrzegać jako rozszerzenie systemu operacyjnego pozwalające na pracę nad specyficznymi strukturami danych.

3.4.1. Organizacja systemu

GRASS jest zbiorem odrębnych programów uruchamianych bezpośrednio z poziomu systemu operacyjnego (z linii komend).

Tu uwaga dla osób które nie umieją się biegle posługiwać komputerem: pomimo tego, że linia komend systemu UNIX jest „dla prawdziwych mężczyzn”, polecenia systemu GRASS są bardzo przyjazne; wyświetlają podpowiedzi, w większości przypadków wręcz odpytują o niezbędne parametry; ponadto powstało już kilka rozszerzeń systemu o interfejs graficzny (przykładem jest TclTkGRASS).

Wszystkie programy wchodzące w skład GRASS są uruchamiane po wstępnym ustawieniu odpowiedniego środowiska, pewnych zmiennych systemowych itp. Na szczęście użytkownik nie musi się tymi szczegółami interesować. Ustawienia te są wykonywane przez programy narzędziowe w sposób dla niego niedostrzegalny.

3.4.2. Grupy komend

Każdy program wchodzący w skład systemu GRASS, będziemy go nazywać komendą, ma nazwę tworzoną w pewien charakterystyczny sposób. Nazwa składa się z litery, kropki oraz, najczęściej jednego, słowa. Oto przykłady: d.colormode, r.support, v.digit, g.manual. Pierwsza litera oznacza przynależność danej komendy do grupy komend. Mamy tu, na przykład, grupę komend ogólnych wyróżnionych literą „g” (general). Grupę komend dotyczących wyświetlania - litera „d” (display). Grupy komend obsługi

danych rastrowych, wektorowych oraz obrazów (np. dane satelitarne) - odpowiednio litery „r” (raster), „v” (vector), „i” (imaging).

Ogólnie wszystkie polecenia GRASS są podzielone na następujące grupy:

- wyświetlania obrazów na ekranie i przenoszenia ich na papier;
- operacje na warstwach rastrowych;
- operacje na warstwach wektorowych;
- operacje na danych punktowych;
- operacje na obrazach wielospektralnych;
- obsługi bazy danych;
- polecenia w fazie testów alfa;
- programy dołączone, ale nie stanowiące elementów bazowego systemu;
- programy dodatkowe.

Listę wszystkich komend dostępnych w danej dystrybucji GRASS można uzyskać wywołując komendę `g.manual`, a następnie wybierając opcję „list”.

Jest jeszcze inny sposób grupowania komend. Komendy GRASS są podzielone na komendy wchodzące w skład podstawowego pakietu GRASS, komendy aktualnie testowane pod względem ich przydatności i poprawności działania przed przyłączeniem „na stałe” do systemu GRASS, komendy dołączone przez użytkowników, skrypty systemowe i inne. O przyłączeniu danego programu do grupy podstawowych komend GRASS decyduje GRASS Reserach Group.

3.4.3. Opis komend - pomoc

Każde polecenie GRASS można uruchomić z parametrem „--help” (dwie kreski nie są błędem, tak ma być!). Wyświetla ono wtedy krótki opis, możliwych do wykorzystania,

parametrów. Jeżeli nie jesteśmy przygotowani, lub nie mamy ochoty na wpisywanie jednej długiej linii, zawierającej nazwę komendy i niezbędne parametry, możemy uruchomić polecenie bez podania parametrów. Spowoduje to przejście danej komendy w tryb interaktywny. Oznacza to, że program będzie odpytywał nas kolejno o wszystkie parametry, informując co to za parametr i czy musi być on koniecznie podany, czy też można go pominąć. Bardzo często proponowane są też wartości domyślne.

Może się zdarzyć, że pomimo opisanych ułatwień, nie można zrozumieć, jak działa dana komenda oraz jaką rolę pełnią jej parametry. Jest to sytuacja, z jaką spotkamy się przy pierwszych kontaktach z systemem GRASS. Na ratunek przychodzi nam wtedy polecenie `g.manual`. Polecenie to, jak sama nazwa mówi (manual - podręcznik), pozwala na wyświetlenie pełnego opisu każdego polecenia. `g.manual` operuje łącznie setkami (jak nie tysiącami) stron opisu poleceń.

3.4.4. Organizacja danych

Wszystkie dane, którymi operuje GRASS, zebrane są w bazę danych GRASS. Odpowiada to wskazaniu katalogu bazowego, w którym (lub też w którego podkatalogach) są przechowywane wszystkie informacje. Baza danych GRASS podzielona jest na lokacje (location). Każda lokacja ma swój własny układ odniesienia, odwzorowanie, położenie. Oznacza to, że lokacja LOC1 może być zdefiniowana w oparciu o współrzędne kartezjańskie (X-Y) użytkownika, zmieniające się w przedziałach 0-10000, 0-15000, z podstawowym krokiem co 100, a lokacja LOC2 może zostać określona w oparciu o współrzędne geograficzne w siatce co 30”.

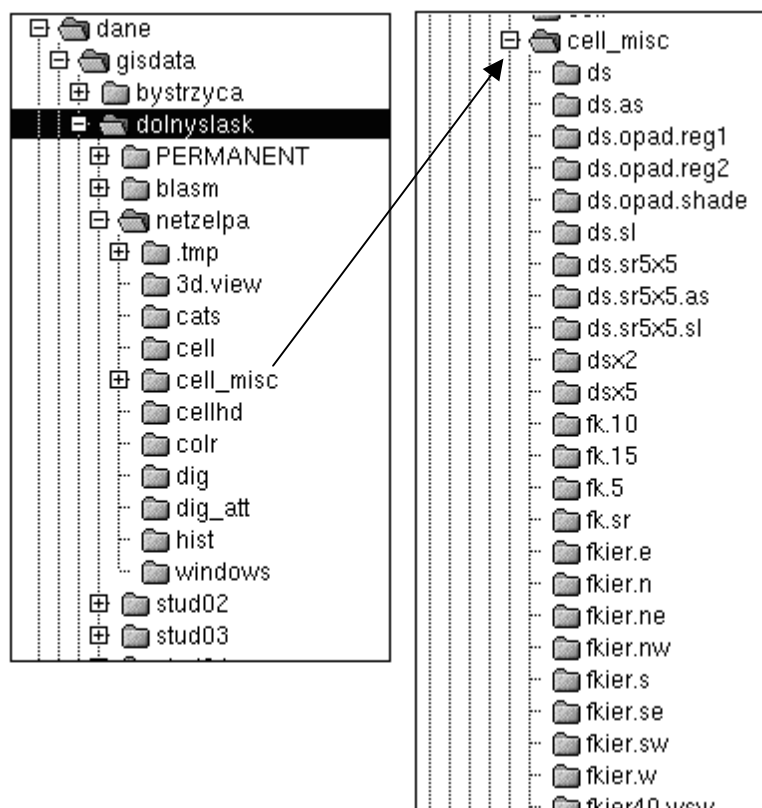
W ramach danej lokacji każdy użytkownik ma przydzielany swój zbiór map (mapset). Wyjątkiem jest mapset PERMANENT. Jest to mapset zawierający podstawowe informacje o danej lokacji. Administrator systemu może w nim umieścić podstawowe warstwy tematyczne (np. model wysokościowy, drogi, rzeki itp.).

Ważne jest aby pamiętać, że użytkownik może zapisywać wynik pracy jedynie w swoim własnym zbiorze map. Każdy użytkownik ma, z założenia, dostęp do zbioru map PERMANENT. Ponadto używając komendy „`g.mapsets`” możemy uzyskiwać dostęp do pracy innych użytkowników. System dba o to, aby dostęp ten był realizowany jedynie na

poziomie odczytu. Zabezpiecza to pracę innych przed nieumyślnym (lub umyślnym) uszkodzeniem.

3.4.5. Struktura katalogów

Każdy mapset, fizycznie na dysku, jest reprezentowany przez drzewo katalogów. Jeżeli musimy przenieść dane z komputera na komputer, informacja o tym gdzie szukać danych, może okazać się niezbędna.



Struktura przykładowej bazy danych systemu GRASS pokazana jest na rysunku.

Oto opis niektórych ważniejszych podkatalogów:

„cell” Znajdują się w nim pliki zawierające dane rastrowe; pliki te są w formacie binarnym; są to macierze liczb całkowitych 1, 2 lub 4 bajtowych; każdy plik zawiera dane dotyczące jednej warstwy; pliki

te są identyfikowane po nazwie

- „cellhd” Zawiera pliki nagłówkowe dla danych rastrowych; są to pliki tekstowe; pliki te zawierają informacje o rozmiarach danej mapy rastrowej (liczba kolumn, liczba wierszy, rozmiar rastra), format danych (1,2,4 bajty), opis mapy; nazwa pliku nagłówkowego odpowiada nazwie pliku z danymi rastrowymi.
- „colr” Tu znajdują się pliki zawierające definicje kolorów dla mapy rastrowej; nazwa pliku definiującego kolory odpowiada nazwie pliku z danymi.
- „cats” Zawiera pliki z opisem kategorii dla wartości rastra; nazwa pliku z kategoriami odpowiada nazwie pliku z danymi.
- „cell_misc” Zawiera kartoteki z informacjami dodatkowymi, takimi jak zakres zmienności wartości w danej warstwie, histogram wartości danej warstwy; nazwa kartoteki z informacjami dodatkowymi odpowiada nazwie pliku z danymi.
- „hist” Kartoteka zawiera historię powstania poszczególnych warstw rastrowych; kto i kiedy daną warstwę utworzył oraz w jaki sposób to wykonał; nazwa pliku z historią odpowiada nazwie pliku z danymi.
- „dig” Kartoteka z plikami zawierającymi dane wektorowe
- „dig_att” W tej kartotece przechowywane są pliki z atrybutami każdego łańcucha wektorów dla danej warstwy; nazwa pliku z atrybutami odpowiada nazwie pliku z danymi.
- „dig_cats” Tu znajdują się pliki z definicjami kategorii dla danej warstwy wektorowej; nazwa pliku z definicjami kategorii odpowiada nazwie pliku z danymi.

3.4.6. Przenoszenie danych

Ważne jest, aby się orientować które pliki/kartoteki należy przenieść, jeżeli chcemy przenieść daną warstwę z jednego komputera na drugi. Ta wiedza nie jest konieczna, gdy przenosimy cały mapset/location. Problemem jest tu zazwyczaj duża ilość danych. Z tego powodu często należy ograniczyć się do przesłania tylko wybranych plików.

Tak więc, dla warstw rastrowych wystarczy skopiować odpowiednie pliki z kartotek „cell” i „cellhd”. W pewnych uzasadnionych przypadkach (warstwa zawiera opisy kategorii np. geologia) – dodatkowo z kartoteki „cats”. Dla warstw wektorowych będą to odpowiednie pliki z kartotek: „dig” i „dig_att”. Tak samo w tym przypadku w pewnych przypadkach należy przekopiować pliki z kartoteki „dig_cats”.

Zwróćmy uwagę, że wszystkie pliki opisujące daną warstwę noszą taką samą nazwę. Oznacza to, że musimy przenieść pliki zachowując strukturę drzewa katalogów (inaczej pliki o takich samych nazwach zostały by zamazane).

Uwaga. Przy przenoszeniu warstw utworzonych przy pomocy polecenia „r.reclass” należy przenieść dodatkowo warstwy, które były argumentami tej komendy. Ponadto na komputerze docelowym nazwy lokacji jak i zbioru map muszą być takie same jak na komputerze źródłowym.

Do spakowania przenoszonych danych system UNIX udostępnia nam polecenia „tar” oraz „gzip”. Pełny opis składni tych poleceń możemy uzyskać pisząc (w linii komend): „man tar” lub „man gzip”.

4. Sesja w systemie GRASS

4.1. Uruchomienie programu

System GRASS uruchamiamy z poziomu linii poleceń systemu UNIX pisząc „grass42”. Oczywiście nazwa pliku startowego może być inna, jeżeli użytkownik ją zmieni, jednak w czasie instalacji plik startowy nosi właśnie taką nazwę. Plik „grass42” nie jest programem binarnym, jest to skrypt systemowy, ustalający odpowiednie zmienne pozwalające na poprawną pracę systemu GRASS oraz wywołujący właściwy program.

W momencie uruchomienia system pyta nas o nazwy lokacji, zbioru map oraz kartoteki zawierającej bazę danych GRASS. Informacje te są zapamiętywane, tak że przy ponownym uruchamianiu systemu przez tego samego użytkownika, pola będą już wstępnie wypełnione.

Informacje te są przechowywane w pliku „grassrc” w kartotece użytkownika.

Uwaga. Poza plikiem „grassrc” system tworzy w kartotece użytkownika plik „gislock”. Plik ten informuje GRASS o tym, że dany użytkownik ma otwartą sesję w systemie. W przypadku gdy, w trakcie uruchamiania systemu GRASS, pojawi się komunikat sygnalizujący, że GRASS został już wcześniej uruchomiony, należy usunąć ten plik ręcznie. Taka sytuacja może zaistnieć wtedy, gdy nie zakończyliśmy prawidłowo poprzedniej sesji w systemie GRASS

Informacje wprowadzamy wpisując kolejne znaki w trybie zastępowania. Pomędzy polami przechodzimy naciskając klawisz „ENTER”, a wprowadzone informacje zatwierdzamy naciskając klawisz „ESC”. Jeżeli chcemy usunąć wprowadzone już znaki, należy je nadpisać. Jeżeli znajdzie konieczność przzerwania startu systemu GRASS, należy równocześnie nacisnąć klawisze „Ctrl” i „C”

Przykładowy ekran startowy systemu pokazany jest na rysunku.



```
Terminal
File Edit Settings Help

GRASS 4.2.1

LOCATION: This is the name of an available geographic location. -spearfish-
is the sample data base for which all tutorials are written.

MAPSET: Every GRASS session runs under the name of a MAPSET. Associated
with each MAPSET is a rectangular COORDINATE REGION and a list
of any new maps created.

DATABASE: This is the unix directory containing the geographic databases

The REGION defaults to the entire area of the chosen LOCATION.
You may change it later with the command: g.region
-----

LOCATION:  płnyslask____          (enter list for a list of locations)
MAPSET:   netzelpa____          (or mapsets within a location)
DATABASE: /dane/gisdata_____

AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
(OR <Ctrl-C> TO CANCEL)
```

Ekran powitalny systemu GRASS

W przypadku, gdy chcemy rozpocząć pracę z nową lokacją, musimy być przygotowani na podanie dodatkowych informacji. Program sam rozpozna, że podana lokacja nie istnieje we wskazanej bazie danych i zapyta o układ współrzędnych, zasięg i rozmiar rastra oraz o opis lokacji. W przypadku, gdy posiadamy żądane informacje, program utworzy nową lokację, utworzy w niej mapset PERMANENT oraz mapset o nazwie zgodnej z nazwą zarejestrowanego użytkownika.

Przykładowe ekrany, pojawiające się w trakcie tworzenia nowej lokacji, widać na rysunkach kolejnych rysunkach.

```

README.1st  bystrzyca dolnyslask kwisaorto szrenica
README.fire dane fire-locat sudety v_in_dxf3d_testdata,dxf
LOCATION <sniezka> - doesn't exist

Available locations:
-----
Would you like to create location <sniezka> ? (y/n) █

```

To pytanie pojawi się w momencie próby otwarcia lokacji która jeszcze nie istnieje.

```

Terminal
File Edit Settings Help
To create a new LOCATION, you will need the following information:
1. The coordinate system for the database
   x,y (for imagery and other unreferenced data)
   UTM
   State Plane
   Latitude-Longitude
   other projection
2. The zone for the database
   (except for x,y and Latitude-Longitude databases)
3. The coordinates of the area to become the default region
   and the grid resolution of this region
4. A short, one-line description or title for the location
Do you have all this information for location <sniezka> ? (y/n) █

```

Następnie system upewnia się czy posiadamy informacje wystarczające do utworzenia nowej lokacji

```

Terminal
File Edit Settings Help
Please specify the coordinate system for location <sniezka>
0 x,y
1 UTM
2 State Plane
3 Latitude-Longitude
99 Other
RETURN to cancel
> █

```

Możemy wybrać typ układu odniesienia

```

Terminal
File Edit Settings Help
DEFINE THE DEFAULT REGION

===== DEFAULT REGION =====
| NORTH EDGE:0-----|
| WEST EDGE          | EAST EDGE          |
| 0-----|         | 0-----|
| SOUTH EDGE:0-----|
|-----|
PROJECTION: 0 (x,y)                                ZONE: 0

GRID RESOLUTION
East-West:    0-----
North-South:  0-----

AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
(OR <Ctrl-C> TO CANCEL)

```

Kolejną czynnością jest określenie rozmiaru lokacji oraz podstawowej rozdzielczości danych rastrowych

```

projection:  0 (x,y)
zone:        0
north:       10000
south:       0
east:        15000
west:        0

e-w res:     12
n-s res:     12,00480192 (Changed to conform to grid)

total rows:      833
total cols:      1250
total cells:     1,041,250

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

```

Na koniec GRASS upewnia się czy prawidłowo określiliśmy współrzędne

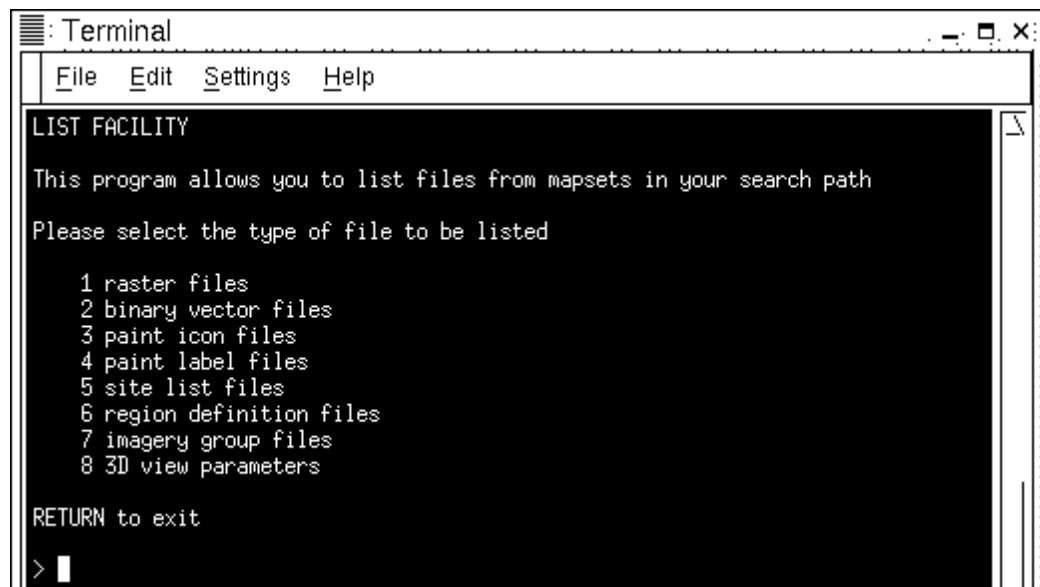
Nie jest dobrą praktyką, w czasie pracy z wielodostępem, zmiana nazwy mapsetu na inną niż nazwa użytkownika. Nie prowadzi to do błędów, lecz zaburza logiczne powiązanie wyników pracy z konkretną osobą.

4.2. Przeglądanie zgromadzonych danych

Po prawidłowym uruchomieniu systemu w oknie programu X-Terminal pojawi się komunikat o uruchomieniu systemu GRASS oraz znak zachęty. Możemy teraz wywoływać komendy systemu.

4.2.1. Spis warstw

Pierwszą czynnością, jaką wykonamy, będzie sprawdzenie, jakimi warstwami dysponujemy. W tym celu posłużymy się poleceniem „g.list”. Polecenie to możemy wywołać z parametrem „rast” lub z parametrem „vect”. W pierwszym przypadku zostaną wyświetlone jedynie warstwy rastrowe, w drugim warstwy wektorowe. Wywołanie komendy „g.list” bez parametru spowoduje wyświetlenie wszystkich dostępnych typów plików w systemie GRASS.



```
Terminal
File Edit Settings Help
LIST FACILITY
This program allows you to list files from mapsets in your search path
Please select the type of file to be listed

1 raster files
2 binary vector files
3 paint icon files
4 paint label files
5 site list files
6 region definition files
7 imagery group files
8 3D view parameters

RETURN to exit
> |
```

Rezultat komendy „g.list” wywołanej bez parametru

```

Terminal
File Edit Settings Help

RETURN to exit
>
[netzelpa@stratus netzelpa]$ g.list rast
-----
raster files available in mapset netzelpa:
ds          dsx2      fkier,nw   fkolo,5    wynik,r     zk,ne
ds,as       dsx5      fkier,s    od15       xcoord      zk,nw
ds,opad,reg1  fk,10    fkier,se   shad,col   xwsp        zk,s
ds,opad,reg2  fk,15    fkier,sw   shade      ycoord      zk,se
ds,opad,shade fk,5      fkier,w    shade,col  ywsp        zk,sw
ds,sl        fk,sr     fkier40,ws  shade_dshis  zaslkie,ssww,50km  zk,w
ds,sr5x5     fkier,e   fkier50,ssww  test        zk,40km
ds,sr5x5,as  fkier,n   fkolo,10    wspN        zk,e
ds,sr5x5,sl  fkier,ne  fkolo,15    wspS        zk,n

raster files available in mapset blasm:
ds1         ds_mod   fme         fmin9a     foga      max_ds      rob2      teren1
ds2_10000   ds_sr5   fmediana9   fmo        fogb      max_dsds_min_1000  rob3
ds_med      fmax9    fmi         fmoda9     fogc      mm          sr5_ds
ds_min      fmax9a   fmin9       fog         fsr5     rob1        teren

[netzelpa@stratus netzelpa]$

```

Rezultat komendy „g.list” wywołanej z parametrem „rast”

4.2.2. Uruchomienie okna graficznego

Wiedząc już czym dysponujemy, możemy przygotować się do oglądania poszczególnych warstw. System GRASS pozwala na otwieranie okien graficznych zwanych x-monitorami. W wersji podstawowej system pozwala na równoczesne otwarcie do sześciu monitorów. Administrator systemu może jednak zwiększyć ich ilość. Monitory noszą kolejne numery. Numeracja zaczyna się od wartości 0. W środowisku wielodostępnym, w danej chwili, tylko jedna osoba może korzystać z monitora o danym numerze.

Do otwierania oraz zarządzania oknami monitorów służy polecenie „d.mon”.

Otwarcie monitora następuje po wywołaniu komendy „d.mon” z parametrem „x{numer_monitora}”. Przykładowo otwarcie monitora o numerze 1 nastąpi po wpisaniu: „d.mon x1”.

Uruchomienie komendy „d.mon” bez żadnych parametrów spowoduje rozpoczęcie pracy z poleceniem w trybie interaktywnym. Pojawi się lista pozwalająca na wybór czynności (otwarcie, zamknięcie, zmiana monitora), którą chcemy wykonać.

4.2.3. Wyświetlenie warstwy

Mając już otwarte okno graficzne, możemy przystąpić do przeglądania danych.

Uwaga: Przed wyświetleniem okna graficznego należy wybrać jego rozmiar i położenie. Zmiana wymiarów okna graficznego spowoduje jego wyczyszczenie.

Do wyświetlania informacji z warstwy rastrowej używamy polecenia „d.rast”. Po komendzie „d.rast” należy podać nazwę warstwy rastrowej, którą chcemy zobaczyć. Program zapyta nas jedynie o to, czy wartość „0” występująca w węźle rastra (oznacza ona brak danych) ma być pomijana (przezroczysta), czy też nie. W przypadku, gdy nie pamiętamy dokładnej nazwy warstwy, możemy wywołać „d.rast” bez parametrów. W tym przypadku „d.rast” zapyta nas o nazwę warstwy. Jeżeli zamiast nazwy warstwy wpiszemy słowo „list”, program wyświetli listę warstw rastrowych, a następnie ponownie poprosi o podanie nazwy warstwy. Wybranie właściwej nazwy nie stanowi już problemu.

```

Terminal
File Edit Settings Help
[netzelpa@stratus netzelpa]$ d.rast
OPTION: Raster map to be displayed
key: map
required: YES

Enter the name of an existing raster file
Enter 'list' for a list of existing raster files
Hit RETURN to cancel request
> list
<list>

-----

raster files available in mapset netzelpa:
ds          dsx2      fkier.nw   fkolo.5    wynik,r     zk.ne
ds.as       dsx5      fkier.s    odl5       xcoord      zk.nw
ds.opad.reg1  fk.10    fkier.se   shad.col   xwsp        zk.s
ds.opad.reg2  fk.15    fkier.sw   shade      ycoord      zk.se
ds.opad.shade fk.5      fkier.w    shade.col   ywsp        zk.sw
ds.sl       fk.sr     fkier40.wsw shade_dshis zaslkie.r.ssw.50km zk.w
ds.sr5x5     fkier.e   fkier50.ssw test        zk.40km
ds.sr5x5.as  fkier.n   fkolo.10   wspN       zk.e
ds.sr5x5.sl  fkier.ne  fkolo.15   wspS       zk.n

raster files available in mapset blasz:
ds1         ds_mod    fme        fmin9a     foga      max_ds      rob2      teren1
ds2_10000   ds_sr5    fmediana9  fmo        fogb      max_dsds_min_1000 rob3
ds_med      fmax9     fmi        fmoda9     fogc      mm          sr5_ds
ds_min      fmax9a    fmin9      fog         fsr5     rob1        teren

-----

Enter the name of an existing raster file
Enter 'list' for a list of existing raster files
Hit RETURN to cancel request
> ds
<ds>

FLAG: Set the following flag?
Overlay (non-zero values only)?(y/n) [n] █

```

Komenda „d.rast’ - praca w trybie interaktywnym

W celu wyświetlenia warstwy wektorowej można wykorzystać polecenie „d.vect”. Polecenie to ma takie same opcje jak „d.rast”.

Jeżeli X-Terminal jest w trybie grafiki ośmiobitowej, może on wyświetlić jedynie 256 kolorów. Część kolorów jest zarezerwowana przez system na barwy ramek, ikon systemowych itp. W tym momencie do wyświetlenia barwnego rastra pozostaje niewiele kolorów. Prezentacja barwna, od której oczekujemy płynnej zmiany barw, barwnych półtonów, nie spełni naszych oczekiwań. Możemy temu zaradzić. Należy po prostu udostępnić x-monitorowi wszystkie 256 kolorów. Wykonujemy to z pomocą komendy „d.colormode”. Ta komenda posiada dwa parametry: „float” i „fixed”. Uruchomienie

„d.colormode” z parametrem „float” udostępnia bieżącemu x-monitorowi wszystkie dostępne kolory kosztem kolorów systemowych. Uruchomienie z opcją „fixed” przywraca poprzednią sytuację. Po udostępnieniu oknu x-monitora wszystkich kolorów, trzeba ponownie wyświetlić warstwę („d.rast”). To co widzimy, nie jest ciekawe. Pożądany wygląd okna uzyska się dopiero po najechaniu na okno wskaźnikiem myszy (na pozostałej części ekranu widać, jakim kosztem).

4.2.4. Zamknięcie okna graficznego

Jeżeli już napatrzeliśmy się na obrazki i chcemy zakończyć pracę, należy zamknąć okno(a) graficzne.

Uwaga: Okna graficzne MUSIMY zamykać z poziomu linii komend. Nie należy kończyć pracy z systemem GRASS bez zamknięcia WSZYSTKICH okien graficznych. Nie można także zamykać okien poprzez przycisk zamykania na ramce okna. Zamknięcie okna w niewłaściwy sposób ograniczy możliwość korzystania z monitora o tym numerze innym użytkownikom.

Aby zamknąć okno ponownie używamy komendy „d.mon”. Tym razem wywołujemy polecenie z parametrem „stop=x {numer_monitora}”. Oto przykład: „d.mon stop=x1”. Polecenie to zamknie okno x-monitora o numerze 1.

Podobnie, jak w przy otwieraniu okna graficznego, można też wywołać komendę „d.mon” w trybie interaktywnym. Uzyskamy to uruchamiając „d.mon” bez podania parametrów. Jedną z proponowanych opcji jest wtedy zamykanie aktywnego okna. Osoba cierpliwa może w ten sposób kolejno uaktywniać okna i je zamykać.

4.3. Wykonywanie podstawowych obliczeń na warstwach rastrowych

Samo przeglądanie warstw tematycznych nie stanowi już problemu. Zajmiemy się teraz odpowiedzią na pytanie: jak utworzyć nową warstwę zawierającą wyniki naszych obliczeń? Oczywiście, większość specjalizowanych poleceń automatycznie tworzy

warstwy wynikowe. Przykładem niech będzie polecenie „r.slope.aspect”. Generuje ono automatycznie, na podstawie warstwy zawierającej DEM, mapy nachyleń i ekspozycji.

W tym momencie interesuje nas jednak nie jak automatycznie utworzyć warstwę (tym zajmiemy się przy omawianiu poszczególnych poleceń), lecz jak swoje wzory, algorytmy obliczeń wprowadzić do systemu GRASS i otrzymać wyniki w układzie przestrzennym.

4.3.1. Wykorzystywanie polecenia r.mapcalc

Do wykonywania obliczeń na warstwach rastrowych w systemie GRASS przeznaczone jest polecenie „r.mapcalc”. Program „r.mapcalc” jest swego rodzaju ‘kalkulatorem’. Kalkulator ten przyjmuje warstwy rastrowe jako argumenty. Oto kilka przykładów wyrażeń przyjmowanych przez program:

1 - „> wyn = w1 + w3”

2 - „> wyn = if(nachylenie>20,2,1)”

3 - „> wyn = sin(w3)+cos(w2)”

Znak „>” jest znakiem zachęty programu „r.mapcalc”. Jak większość programów systemu GRASS, „r.mapcalc” można uruchomić w trybie interaktywnym. Pozwala to na wykonanie kilku/kilkunastu kolejnych obliczeń bez konieczności ponownego wywoływania polecenia „r.mapcalc”.

Wróćmy do przykładów.

Przykład 1 utworzy nową warstwę „wyn”. Każdy element rastra „wyn” powstanie z sumy, odpowiadających pod względem współrzędnych X-Y, elementów warstw „w1” i „w3”.

Przykład 2 utworzy nową warstwę „wyn”. Każdy element rastra „wyn” przyjmie wartość 2 gdy, odpowiadający pod względem współrzędnych X-Y, element warstwy „nachylenie” będzie miał wartość większą niż 20. W przeciwnym wypadku element ten przyjmie wartość 1.

Przykład 3 utworzy nową warstwę „wyn”. Każdy element rastra „wyn” powstanie z sumy sinusa i cosinusa, odpowiadających pod względem współrzędnych X-Y, elementów warstw „w3” i „w2”.

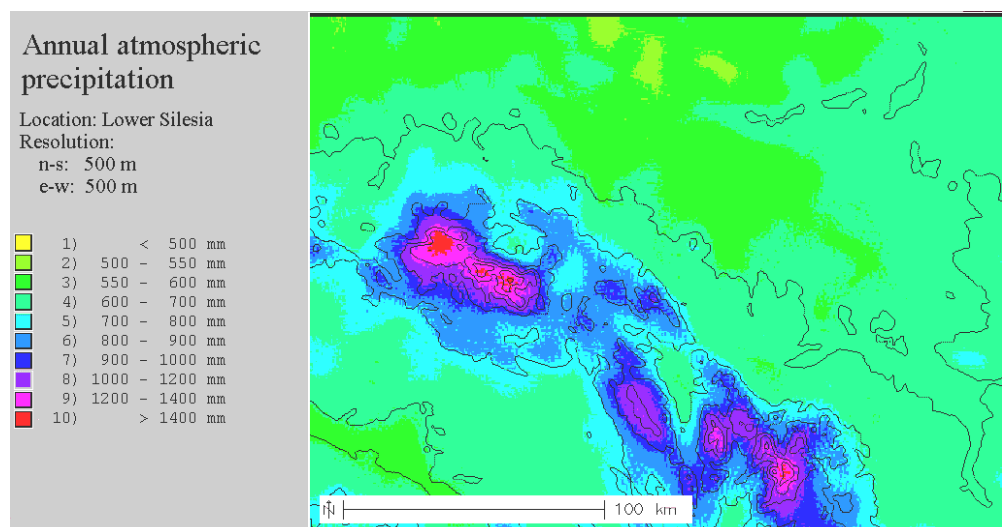
Oczywiście, powyższe przykłady są bardzo proste. Program „r.mapcalc” nie ogranicza nas tylko do tak elementarnych operacji, ani do tak małej liczby wykorzystywanych warstw. W linii komend programu „r.mapcalc” można wprowadzić bardzo złożone formuły, np.:

$$\begin{aligned} opad = & 433.37 + 2.32 \cdot x \cdot 10^{-4} + 5.77 \cdot y \cdot 10^{-5} + 0.652899 \cdot \bar{z} - 5.09876 \cdot s + \\ & 0.31033 \cdot zk_{40} + 1.183 \cdot S \cdot 10^{-3} - 1.59566 \cdot N - 1.72335 \cdot fk_2 + 0.512477 \cdot fk_5 + \\ & 0.481684 \cdot fk_{10} + \frac{2.344087 \cdot 10^6}{S_d} \end{aligned}$$

W programie „r.mapcalc” formuła ta wygląda następująco (wszystko wprowadzamy w jednej linii):

```
ds.opad.reg2 = 433.37 + 0.000232 * xwsp + 0.0000577 * ywsp + 0.652899 * ds.sr5x5 -
5.09876 * ds.sl + 0.31033 * zk.40km + 0.001183 * wspS - 1.59566 *
wspN - 1.72335 * fk.sr + 0.512477 * fk.5 + 0.481684 * fk.10 + 234.4087
* 10000.0 / odIS
```

W wyniku tego działania otrzymujemy warstwę przedstawioną na rysunku.



We wszystkich powyższych przykładach program „r.mapcalc” był wykorzystywany do wykonywania obliczeń „w poprzek” warstw. To znaczy element rastra wynikowego był wyliczany na podstawie elementów rastra innych warstw, które to elementy miały takie same współrzędne X-Y, jak element wynikowy. Często można spotkać się z sytuacją, gdzie wynik musi zależeć nie tylko od wartości parametru w danym punkcie, lecz także od otoczenia tego punktu (wartości parametru w sąsiednich elementach rastra). Program „r.mapcalc” umożliwia też takie operacje. Oto przykład:

„> wyn = wys(0,0) - (wys(-1,0) + wys(1,0) + wys(0,-1) + wys(0,1))/4”

Przyjmując, że „wys” jest nazwą warstwy zawierającej DEM, warstwa „wyn”, która powstanie po wykonaniu powyższego polecenia, zawierała będzie pewnego rodzaju współczynnik opisujący wklęsłość/wypukłość terenu w danym punkcie.

Zasada wskazywania sąsiednich elementów rastra jest zilustrowana na poniższym rysunku.

| | | | | |
|---------|---------|--------|--------|--------|
| (-2,2) | (-1,2) | (0,2) | (1,2) | (2,2) |
| (-2,1) | (-1,1) | (0,1) | (1,1) | (2,1) |
| (-2,0) | (-1,0) | (0,0) | (1,0) | (1,1) |
| (-2,-1) | (-1,-1) | (0,-1) | (1,-1) | (2,-1) |
| (-2,-2) | (-1,-2) | (0,-2) | (1,-2) | (2,-2) |

Siatka z naniesionymi przesunięciami względnymi

Wpisując „nazwa_warstwy(0,0)” otrzymamy bazowy element rastra „nazwa_warstwy”. Odpowiednio operując liczbami w nawiasach można otrzymać element rastra przesunięty względem elementu bazowego o zadaną wartość.

Pomimo, że wartości elementów rastra są przechowywane w postaci liczb całkowitych, wszystkie obliczenia są prowadzone z pełną precyzją na liczbach zmiennoprzecinkowych.

4.3.2. Wykorzystywanie polecenia `r.reclass`

Wszystkie polecenie programu „`r.mapcalc`” mają pewną cechę wspólną, tworzą nową warstwę. Z jednej strony jest to zaleta. Wynik jest kolejną warstwą i może być wykorzystywany do dalszych obliczeń. Z drugiej jednak strony, nowa warstwa to nowy zestaw plików (w katalogach `cell`, `cellhd`, `colr` itp). Pliki te często mają łączną objętość rzędu megabajtów. W wielu przypadkach, gdy wyniki zależą tylko od danych zawartych w jednej warstwie, można wykorzystać polecenie „`r.reclass`”. Program, jak sama nazwa mówi, dokonuje reklasyfikacji danych wejściowych..

Polecenie to nie tworzy warstwy rastrowej w pełnym tego słowa znaczeniu. Tworzy ono jedynie „receptę”, jak ta warstwa ma być generowana. To właśnie ta „recepta” (plik tekstowy wielkości kilkuset bajtów), a nie cała warstwa (plik binarny wielkości megabajtów), jest zapamiętywana.

We wszystkich przypadkach, gdy jest to możliwe, należy stosować polecenie „`r.reclass`”, a nie „`r.mapcalc`”.

4.4. Wyprowadzanie rezultatów obliczeń w postaci mapy

Po wykonaniu określonych obliczeń, utworzeniu nowych warstw, opracowaniu warstw stanowiących podkład, przejrzaniu wyników pracy, przychodzi kolej na utrwalenie ich nie tylko w pamięci komputera, ale przeniesienie ich na papier. GRASS oferuje nam wiele sposobów na utrwalenie swoich prac w formie mapy. Począwszy od przesłania wyniku pracy do pliku w formacie graficznym, poprzez wydruk do formatu `postscript` aż po wielkoformatowe wydruki na papierze.

4.4.1. Sterownik CELL

Sterownik CELL został wprowadzony do systemu GRASS w celu kierowania wyświetlania graficznych wyników do pliku zamiast na monitor.

Można go wykorzystać do:

- kierowania wyjścia graficznego do pliku z większą rozdzielczością niż można uzyskać przechwytyjąc grafikę prosto z ekranu;
- zapisywania obrazów 3d z programu d.3d do pliku;
- zapisywania nakładek (raster/raster/wektory/...) do pliku w celu wyeksportowania ich np. przy pomocy „r.out.tiff”.

Po użyciu sterownika CELL w naszej bazie danych GRASS pojawi się plik rastrowy „D_cell”. Plik ten może być następnie wyeksportowany przy pomocy komendy „r.out.tiff” lub też może być wykorzystany w poleceniu „ps.map”.

Uwaga. Należy pamiętać, że plik „D_cell” ma inny układ współrzędnych i odwzorowanie niż dana lokalacja. W związku z tym można go wyświetlić jedynie przy pomocy pewnej sztuczki.

Po pierwsze, aby skutecznie używać sterownik CELL należy ustawić rozmiar generowanej mapy bitowej. Można to zrobić wykorzystując zmienne systemowe „GRASS_WIDTH” oraz „GRASS_HEIGHT”. Mają one wartość domyślną odpowiednio 641 i 481. Zmieniamy ich wartość przy pomocy poleceń:

```
export GRASS_WIDTH=xxx
```

```
export GRASS_HEIGHT=xxx
```

Uwaga. Należy dodawać 1 do xxx aby ustawić zadaną rozdzielczość.

Przykład:

Chcemy ustawić rozdzielczość 800x600:

```
export GRASS_WIDTH=801
```

```
export GRASS_HEIGHT=601
```

Jeżeli nasz system odmówi wykonania polecenia „export”, można posłużyć się poleceniem „setenv”. Wtedy nasz przykład będzie wyglądał następująco:

```
setenv GRASS_WIDTH=801
```

```
setenv GRASS_HEIGHT=601
```

Teraz, gdy już mamy przygotowane środowisko pracy, możemy zainicjować a następnie wykorzystać sterownik CELL. Typowa sekwencja poleceń wykonywana przy wykorzystywaniu sterownika wygląda następująco:

```
d.mon start=CELL
```

Czekamy na komunikat o gotowości sterownika. Oczekiwanie to może trochę potrwać. Czas oczekiwania zależy od zadeklarowanego rozmiaru pliku wynikowego (zmiennne „GRASS_WIDTH” oraz „GRASS_HEIGHT”).

```
d.mon select=CELL
```

```
d.yyyy      np. d.rast, d.vect, d.3d
```

Używamy poleceń wyświetlania. Na ekranie nie zobaczymy rezultatów gdyż wyniki komend wyświetlania są kierowane bezpośrednio do pliku „D_cell”. Należy używać parametru „-o” przy nakładaniu kolejnych warstw rastrowych.

```
d.mon stop=CELL
```

W tym momencie plik „D_cell” jest zapisywany. To także może trwać dłuższą chwilę. Mamy teraz gotowy plik wynikowy. Aby go obejrzeć na ekranie, posłużymy się pewna sztuczką:

```
g.region raster=D_cell      Ustawiamy region zgodnie z parametrami pliku „D_cell”
```

```
d.mon start=x0              Uruchamiamy x-monitor
```

```
d.rast D_cell               Wyświetlamy plik
```

```
d.mon stop=x0              Po obejrzeniu pliku zamykamy monitor
```



```
p.map input=definicja_mapy
```

Kolejnym poleceniem obsługującym wydruki jest „p.map.new”. Ma on nieco większe możliwości niż „p.map”. Można go wywoływać w podobny sposób jak komendę „p.map”.

Wśród poleceń drukowania szczególną rolę odgrywa komenda „p.select”. Polecenie to umożliwia wybranie urządzenia drukującego. Pracę nad wydrukiem powinniśmy rozpocząć właśnie od tego polecenia.

Przykład:

```
p.select painter=ppm
```

Wybranie wydruku do pliku typu PPM

```
p.select -l
```

Wyświetlenie dostępnych sterowników

```
p.select -p
```

Wyświetlenie aktualnie wybranego sterownika

4.4.3. Wydruk w formacie Postscript

System GRASS pozwala na wyprowadzanie wyników pracy w formacie postscript. Jest to szeroko stosowany język opisu strony. Jego cechą szczególną jest to, że informacje wektorowe są pamiętane w formacie wektorowym (to samo dotyczy napisów). W związku z tym obraz w formacie postscript można skalować bez utraty jakości. Ponadto format ten jest wykorzystywany przez większość profesjonalnych programów do obróbki i drukowania grafiki.

Polecenia pozwalające na obsługę wydruków w formacie postscript:

```
„ps.icon”, „ps.map”, „ps.select”.
```

Ich funkcje i sposób obsługi jest zbliżony do obsługi, odpowiadających im, programów z grupy „p.*”. Oczywiście język definiowania mapy w poleceniu „ps.map” jest nieco zmieniony w stosunku do języka polecenia „p.map” i pozwala na wykorzystanie specyfiki formatu postscript. Składnia wywołań, z linii komend, programów z grupy „ps.*” nie odbiega od składni odpowiadających im programów z grupy „p.*”.

4.4.4. Inne formy wyprowadzania wyników

Prace nad systemem GRASS odbywają się w sposób ciągły, przez cały czas. Dzięki temu GRASS rozwija się bardzo dynamicznie. Większość nowych koncepcji znajduje w nim, niemal natychmiast, swoje odbicie.

Jedną z takich koncepcji jest rzeczywistość wirtualna. Związany z nią jest VRML (*virtual reality modeling language*). Jak sama nazwa mówi, jest to język służący do opisu rzeczywistości wirtualnej. GRASS posiada funkcję pozwalającą na wyprowadzanie wyników pracy z wykorzystaniem VRML. Nosi ona nazwę „p.vrml”.

Polecenie to obsługuje wersję 1.0 VRML. Pozwala ono na wyprowadzenie modelu wysokościowego. Przykład wywołania:

```
p.vrml input=nazwa_rastra output=nazwa_pliku_vrml
```

Parametr „input” określa nazwę warstwy rastrowej zawierającej informację o kształcie terenu (DEM). Poza podstawowymi parametrami („input”, „output”) program pozwala na podanie nazwy warstwy rastrowej zawierającej barwy nakładane na generowaną powierzchnie oraz współczynnik skali wysokości.

Uwaga. Ze względu na ograniczenia przeglądarek vrml nie należy tworzyć plików wynikowych z obszarów większych niż 75x75 elementów rastra.

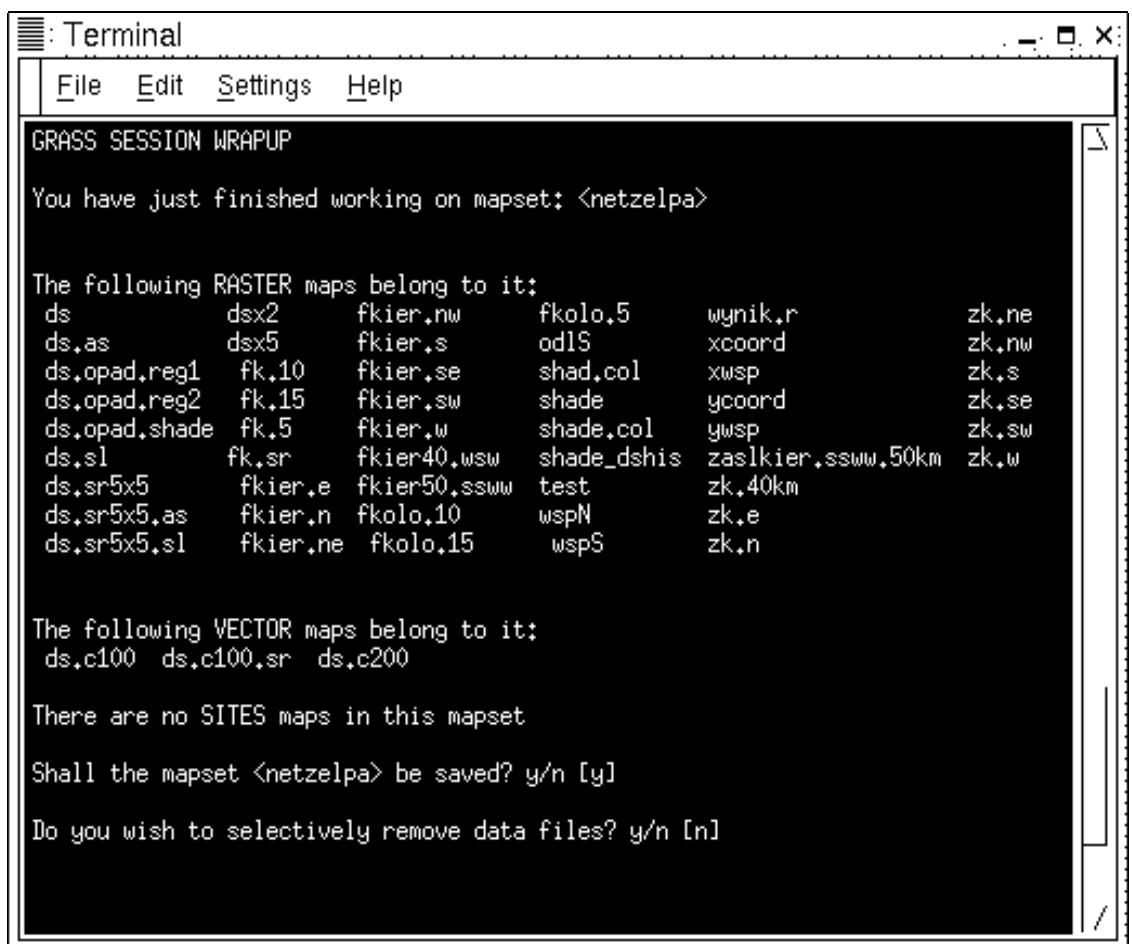
4.5. Zakończenie pracy z programem

Po wykonaniu wszystkich prac, jakie mieliśmy zaplanowane, gdy (na pewno) jesteśmy już zadowoleni z ich wyników, możemy zamknąć system.

Pierwszym krokiem jest zamknięcie wszystkich otwartych okien graficznych - x-monitorów. Ta czynność została już opisana dokładnie w rozdziale dotyczącym okien graficznych („Zamknięcie okna graficznego”). Pamiętajmy jedynie, że zgodnie z uwagą tam zamieszczoną, niezbędne jest zachowanie procedury zamykania okna.

Po zamknięciu okien graficznych system jest gotowy do zakończenia pracy. Wykonujemy to za pomocą komendy „exit”.

W trakcie wykonywania komendy „exit” system zapyta nas o to, czy chcemy usunąć niektóre warstwy. Pozwala to na usunięcie wszystkich warstw, które tworzyliśmy do przechowywania tymczasowych, pośrednich wyników obliczeń. Możemy to teraz wykonać, lub też jeżeli nie odczuwamy potrzeby dodatkowych porządków, zatwierdzić domyślną odpowiedź pozwalającą pominąć te czynności.



```

Terminal
File Edit Settings Help
GRASS SESSION WRAPUP

You have just finished working on mapset: <netzelpa>

The following RASTER maps belong to it:
ds          dsx2      fkier.nw   fkolo.5    wynik.r     zk.ne
ds.as       dsx5      fkier.s    od1S       xcoord      zk.nw
ds.opad.reg1  fk.10    fkier.se   shad.col   xwsp        zk.s
ds.opad.reg2  fk.15    fkier.sw   shade      ycoord      zk.se
ds.opad.shade fk.5      fkier.w    shade.col  ywsp        zk.sw
ds.sl       fk.sr     fkier40.wsw shade_dshis zaslkier.ssww.50km zk.w
ds.sr5x5     fkier.e  fkier50.ssww test        zk.40km
ds.sr5x5.as  fkier.n  fkolo.10   wspN       zk.e
ds.sr5x5.sl  fkier.ne fkolo.15   wspS       zk.n

The following VECTOR maps belong to it:
ds.c100 ds.c100.sr ds.c200

There are no SITES maps in this mapset

Shall the mapset <netzelpa> be saved? y/n [y]

Do you wish to selectively remove data files? y/n [n]

```

Zakończenie pracy z systemem GRASS

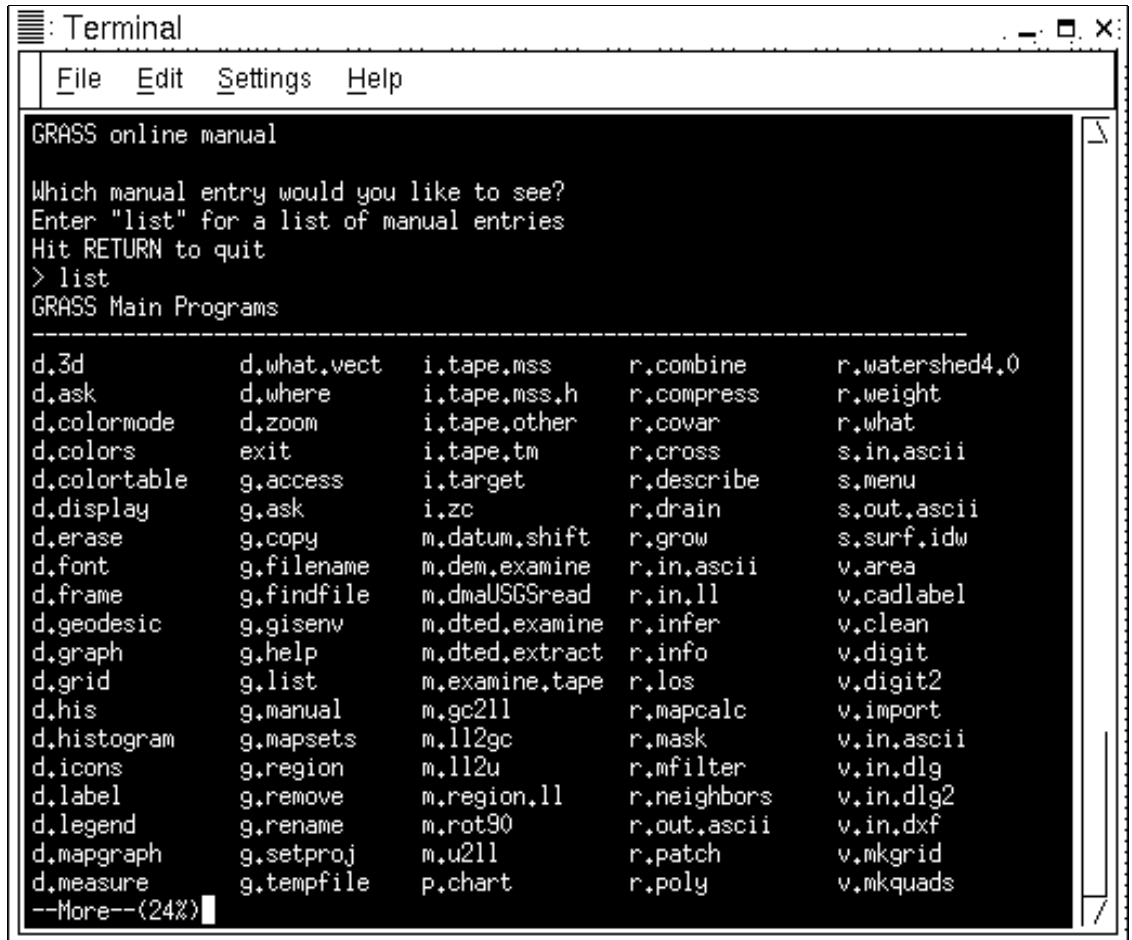
Po tym etapie zamykania programu GRASS powinniśmy zobaczyć znak zachęty systemu operacyjnego.

Jeżeli jesteśmy podłączeni zdalnie do serwera UNIX możemy używać magicznego słowa „exit” tak długo, aż zamkniemy wszystkie okna. Gdy już odłączyliśmy się od

serwera możemy rozpocząć procedurę wyłączenia naszego komputera. W systemie Windows procedurę wyłączenia zaczyna się od wybrania menu „Start”, a następnie „Zamknij system”.

5. Wybrane polecenia sytemu GRASS

W tym rozdziale zapoznamy się z tłumaczeniem opisów wybranych poleceń systemu GRASS. Są to polecenia wykorzystane w prezentowanych przykładach. Pełne ich omówienie można znaleźć w systemie pomocy osiągalnej komendą „g.manual”.



```

Terminal
File Edit Settings Help
GRASS online manual
Which manual entry would you like to see?
Enter "list" for a list of manual entries
Hit RETURN to quit
> list
GRASS Main Programs
-----
d.3d          d.what,vect  i.tape,mss   r.combine    r.watershed4.0
d.ask        d.where     i.tape,mss.h r.compress   r.weight
d.colormode  d.zoom     i.tape,other r.covar     r.what
d.colors     exit        i.tape,tm    r.cross     s.in.ascii
d.colortable g.access    i.target     r.describe  s.menu
d.display   g.ask      i.zc        r.drain     s.out.ascii
d.erase     g.copy     m.datum,shift r.grow      s.surf,idw
d.font      g.filename m.dem,examine r.in.ascii  v.area
d.frame     g.findfile m.dmaUSGSread r.in,ll     v.cadlabel
d.geodesic  g.gisenv   m.dted,examine r.infer     v.clean
d.graph     g.help     m.dted,extract r.info     v.digit
d.grid      g.list     m.examine,tape r.los     v.digit2
d.his      g.manual   m.gc211     r.mapcalc  v.import
d.histogram g.mapsets  m.ll2gc    r.mask     v.in.ascii
d.icons     g.region   m.ll2u     r.mfilter  v.in,dlg
d.label     g.remove   m.region,ll r.neighbors v.in,dlg2
d.legend    g.rename   m.rot90    r.out.ascii v.in,dxf
d.mapgraph  g.setproj  m.u211     r.patch    v.mkgrid
d.measure   g.tempfile p.chart    r.poly     v.mkquads
--More--(24%)

```

Część spisu poleceń wywołanego przy pomocy komendy „g.manual”

5.1. Polecenia ogólne

5.1.1. g.list

g.list – wypisuje, na standardowe wyjście, dostępne pliki danych systemu GRASS, które są typu podanego przez użytkownika.

Składnia

g.list

g.list help

g.list [-f] type=typ_danych [mapset=nazwa]

Opis

g.list wypisuje dostępne pliki ze zbioru map (*mapset*) zgodne z wyspecyfikowanym przez użytkownika typem.

Opcje

W przypadku podania prostej instrukcji *g.list*, program żąda od użytkownika podania typu danych, które mają być wypisane ze wszystkich, dostępnych aktualnie, zbiorów. Użytkownik może wypisać pliki ze zbioru map, nie ujętego w aktualnej ścieżce dostępu, przez uruchomienie programu w trybie nieinteraktywnym, określając (opcjonalnie) ustawienia znacznika i wartości parametru w linii komend. Znaczniki i parametry programu są opisane poniżej.

Znaczniki:

-f

Zwraca rozszerzoną listę plików, zawierającą tytuły map.

Parametr:

type=typ_danych

Typ danych do wypisania

Opcje:

rast - pliki rastrowe

vect - binarne pliki wektorowe

icon - pliki ikon

labels - pliki etykiet

sites - pliki z informacją o punktach

region - pliki z definicjami regionów

group - pliki grup obrazowych

mapset=nazwa

Nazwa zbioru map to przeszukiwania dla plików o wyspecyfikowanym typie. Dowolna nazwa zbioru map zgodnie z aktualną lokalizacją, która może być lub nie wypisana w aktualnej ścieżce dostępu podanej przez użytkownika.

Standardowo: Jeśli nie jest to określone, na standardowym wyjściu będą wypisane pliki o wyspecyfikowanym typie ze wszystkich zbiorów map w aktualnie podanej ścieżce dostępu.

Uwagi

Jeżeli użytkownik zażąda, żeby zostały wypisane pliki ze zbioru map, do których dostęp był ograniczony, żaden plik nie będzie wypisany z tego zbioru map.

5.1.2. g.region

g.region – program zarządza definicjami granic dla regionów geograficznych.

Składnia

g.region

g.region

help

g.region [-dgpu] [region=nazwa] [raster=nazwa] [vector=nazwa] [sites=nazwa]
 [3dview=nazwa] [n=wartość] [s=wartość] [e=wartość] [w=wartość] [res=wartość]
 [nsres=wartość] [ewres=wartość] [zoom=nazwa] [align=nazwa] [save=nazwa]

Opis

Program *g.region* pozwala użytkownikowi na zarządzanie ustawianiem aktualnych regionów geograficznych. Te granice mogą być ustawiane bezpośrednio i/lub z pliku definicji regionów (zapamiętanym w kartotece **windows** w aktualnym zbiorze map użytkownika). Użytkownik może, dla dowolnego zbioru map, tworzyć, modyfikować i zapamiętywać tak wiele definicji regionów geograficznych jak sobie życzy. Jednak tylko jedna definicja regionu geograficznego będzie aktywna w dowolnie podanym momencie dla wyspecyfikowanego zbioru map; tj. program GRASS który respektuje ustawienia regionów geograficznych będzie używać ustawienia aktualnego regionu geograficznego.

INTERAKTYWNE UŻYWANIE PROGRAMU: MENU GŁÓWNE

Menu główne zawiera listę sekcji informacji aktualnej bazy danych systemu GRASS LOCATION, MAPSET i CURRENT REGION wynikającą z opcji użytkownika.

```

                                REGION FACILITY
LOCATION: sample                                MAPSET: grass
CURRENT REGION: N=5167600 S=5156755 RES=50 ROWS=216
                  E=729314 W=705924 RES=50 COLS=467
PROJECTION: 1 (UTM)
ZONE: 13

Please select one of the following options

      Current Region                Region Database
1 Modify current region            6 Save current region in
  directly                          the database
2 Set from default region          7 Create a new region
3 Set from a database              8 Modify an existing region
  region
4 Set from a raster map
5 Set from a vector map

RETURN to quit

```

DEFINICJE

Region:

Tutaj region odnosi się do geograficznego obszaru ze zdefiniowanymi granicami, opartego na systemie współrzędnych i projekcji mapy. Każdy region może być połączony ze specyficzną rozdzielczością wschód-zachód i północ-południe jego jednostek (prostokątna jednostka nazywana „komórką”).

Granice regionu są wysunięte jak najdalej na północ, południe, wschód i zachód punkty, które definiują jego obszar. Granice północna i południowa zwykle są nazywane *northings*, podczas gdy wschodnie i zachodnie granice są nazywane *eastings*.

Rozdzielczość komórki regionu definiuje rozmiar najmniejszego kawałka danych rozpoznawanego (importowanego, wyświetlanego, przechowywanego) przez

program GRASS. Rozdzielczość północ-południe i wschód-zachód nie musi być taka sama, stąd komórki danych nie muszą być kwadratami.

Default Region:

Każdy `LOCATION_NAME` ma ustalony region geograficzny, nazywany standardowym regionem geograficznym (zapamiętanym w pliku regionu `DEFAULT_WIND` w specjalnym zbiorze map `PERMANENT`). Jest to punkt wyjściowy do definiowania nowego regionu geograficznego. Region zdefiniowany przez użytkownika nie musi się zawierać wewnątrz tego regionu.

Current Region:

Każdy zbiór map ma aktualny region geograficzny. Ten region definiuje obszar geograficzny w który będzie wyświetla i będą w nim wykonywane wszystkie analizy. Dane będą ponownie spróbkowane, jeśli to konieczne, aby odpowiadały rozdzielczości komórek z ustawienia aktualnego regionu geograficznego.

Region Data Base:

Każdy zbiorów map (*mapset*) może zawierać dowolną liczbę predefiniowanych i nazwanych regionów geograficznych. Te definicje regionów są zapamiętane w kartotece **windows** aktualnego zbioru map użytkownika.

EDYCJA REGIONU

Najwięcej opcji wymaga od użytkownika edycja regionu geograficznego, czy to będzie aktualny region geograficzny czy też jeden z zapamiętanych definicji regionów (kartoteka **windows**). Parametry regionu standardowego są wykorzystywane w charakterze podpowiedzi. Przykład poniżej:

GIS - GRASS - wprowadzenie do systemu

```

IDENTIFY REGION

===== DEFAULT REGION =====
Default North: 3402025.00

===YOUR REGION===
NORTH EDGE
3402025.00_
WEST EDGE          EAST EDGE
233975.00_        236025.00_
SOUTH EDGE
3399975.00_
=====

Def West: 233975.00
Def East: 236025.00
Default South: 3399975.00

Default  GRID RESOLUTION  Region
50.00   --- East-West ---  50.00__
50.00   -- North-South --  50.00__

AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE

```

Pola NORTH EDGE, SOUTH EDGE, WEST EDGE i EAST EDGE są granicami regionu geograficznego, które użytkownik może zmieniać. Pola Default North, Default South, Def West and Def East są granicami standardowego regionu geograficznego, które są wyświetlane jako informacja i nie mogą być zmieniane. Dwa pola GRID RESOLUTION Region (east-west i north-south) określają rozdzielczość komórek regionu geograficznego, które użytkownik może zmieniać. Dwa pola GRID RESOLUTION Default podają rozdzielczość standardowego regionu geograficznego; są one wyświetlone jako informacja i nie mogą być zmieniane przez użytkownika.

NIEINTERAKTYWNE UŻYWANIE PROGRAMU

Alternatywnie, użytkownik może modyfikować ustawienia aktualnego regionu geograficznego przez specyfikację wszystkich potrzebnych parametrów w linii komend. Użytkownik wprowadza polecenie *g.region parametry*, gdzie *parametry* są następującymi po sobie parametrami i/lub znacznikami.

Znaczniki:**-d**

Nadaje ustawieniom aktualnego regionu wartości równe ustawieniom regionu standardowego.

-g

Drukuje ustawienia aktualnego regionu w formacie, który może być wykorzystany w linii komend polecenia *g.region*.

-p

Drukuje ustawienia aktualnego regionu.

-u

Nie aktualizuje ustawień aktualnego regionu. Pozwala użytkownikowi chwilowo używać innych ustawień, bez ich zapisywania.

Parametry:**region=nazwa**

Tworzy ustawienia aktualnego regionu takie jak ustawienia regionu o nazwie *nazwa*.

raster=nazwa

Tworzy ustawienia aktualnego regionu takie same jak te w nagłówku mapy rastrowej o nazwie *nazwa*. Ale patrz opcja: **zoom=nazwa**, poniżej.

vector=nazwa

Tworzy ustawienia aktualnego regionu takie same jak w mapie wektorowej o nazwie *nazwa*.

sites=nazwa

Ustawia aktualny region do najmniejszego regionu zawierającego wszystkie współrzędne w pliku **site_list** o nazwie *nazwa*.

3dview=nazwa

Tworzy ustawienia aktualnego regionu takie same jak w pliku *3dview* o nazwie *nazwa*. Plik ten przechowuje region, który był aktualny, gdy *3dview* był zapamiętywany.

n=wartość

Ustawia wartość współrzędnej mapy odpowiadającą północnemu wierzchołkowi.

s=wartość

Ustawia wartość współrzędnej mapy odpowiadającą południowemu wierzchołkowi.

e=wartość

Ustawia wartość współrzędnej mapy odpowiadającą wschodniemu wierzchołkowi.

w=wartość

Ustawia wartość współrzędnej mapy odpowiadającą zachodniemu wierzchołkowi.

res=wartość

Ustawia rozdzielczość siatki (zarówno *noth-south* jak i *east-west*).

nsres=wartość

Ustawia rozdzielczość *north-south* siatki.

ewres=wartość

Ustawia rozdzielczość *east-west* siatki.

zoom=nazwa

Ustawia parametry aktualnego regionu do najmniejszego regionu zawierającego niezerowe dane w warstwie *nazwa* mapy rastrowej, które znajdują się wewnątrz aktualnego regionu użytkownika.

Jeśli użytkownik dołącza również opcję **raster**=*nazwa* w linii komend, opcja **zoom**=*nazwa* będzie ustawiała ustawienia aktualnego regionu do najmniejszego regionu zawierającego niezerowe dane w mapie **zoom**, która znajduje się wewnątrz regionu wyrażanego w nagłówku komórki dla mapy rastrowej o nazwie *nazwa*.

align=*nazwa*

Ustawia aktualną rozdzielczość równą tej, która jest w mapie rastrowej o nazwie *nazwa* i wyrównuje aktualny region do krawędzi wiersza i kolumny w tej mapie. Wyrównanie zmienia tylko krawędzie istniejącego regionu do krawędzi następnej najbliższej komórki w mapie rastrowej *nazwa* – a nie do krawędzi mapy. Aby wykonać tą drugą funkcję użyj opcji **raster**=*nazwa*.

save=*nazwa*

Zapamiętuje ustawienia aktualnego regionu w pliku regionu o nazwie *nazwa*.

Uwagi

Mimo wszystkich aktualizacji, które były stosowane, południowe i zachodnie granice aktualnego regionu są dostosowywane tak, aby odległość północ-południe była wielokrotnością rozdzielczości północ-południe, a odległość wschód-zachód jest wielokrotnością rozdzielczości wschód-zachód.

5.2. Polecenia wyświetlania na ekranie

5.2.1. d.mon

d.mon – uruchamia i steruje monitorem graficznym.

Składnia

d.mon

d.mon help

d.mon [-lLprs] [start=*nazwa*] [stop=*nazwa*] [select=*nazwa*] [unlock=*nazwa*]

Opis

Program d.mon pozwala użytkownikowi uruchamiać, wybierać, wylistować, odpytywać stan, przenieść kontrolę, zatrzymać i odblokować kontrolę nad dostępnym monitorem graficznym. Użytkownik może uruchomić ten program zarówno interaktywnie (poprzez menu) lub nieinteraktywnie przez podanie nazwy monitora i innych znaczników w linii komend .

Znaczniki:

- l Lista wszystkich monitorów
- L Lista wszystkich monitorów (z aktualnym statusem)
- p Drukuje nazwę aktualnie wybranego monitora
- r Opuszcza aktualnie wybrany monitor
- s Nie wybieraj automatycznie podczas startu

Parametr:

start=*nazwa*

Nazwa monitora graficznego do uruchomienia

stop=nazwa

Nazwa monitora graficznego do zatrzymania

select=nazwa

Nazwa monitora graficznego do wyboru.

unlock=nazwa

Nazwa monitora graficznego do odblokowania.

Kiedy użytkownik wprowadza komendę `d.mon` bez określonej wartości parametru, na ekranie pojawia się poniższe menu:

MONITOR MENU

Making sure that the graphics monitor is running:

- 1 - Start a graphics monitor
(also automatically selects this monitor)
- 2 - Stop a graphics monitor

Choosing a graphics monitor for your graphics:

- 3 - Select a graphics device for output
(currently selected monitor: name)
- 4 - Release control of the graphics driver
(let someone else use it)
(option 4 appears only after selection of a monitor)

RETURN quit

Te parametry wykonują następujące funkcje:

1. Uruchomienie monitora.

Żeby wyświetlać na ekranie graficznym programu GRASS, użytkownik musi uruchomić i wybrać monitor graficzny. Standardowo komenda `start` faktycznie uruchamia dwie komendy, zarówno `start` i wybór monitora wybranego przez użytkownika. (Użytkownik może otrzymać listę dostępnych monitorów poprzez ustawienie znacznika `-i` lub `-L` na linii komend).

Kiedy monitor jest uruchamiany, jest on też automatycznie wybierany jako wyjście, chyba że jest ustawiony znacznik `-s`; użytkownik może także wyraźnie wskazać monitor, który będzie uruchamiany (patrz pkt.3 poniżej). Po starcie monitora, powinna pojawić się czysta ramka graficzna na terminalu, którego użytkownik używa do wyświetlania grafiki.

Pożądaný monitor może być wybrany raz i nie musi być restartowany chyba że został zatrzymany z jakiegoś powodu (opcja 2). Monitor może kontynuować swoją pracę dowolnie długo, nawet gdy sesja GRASS została zakończona. Program monitora pracuje w tle.

2. Zatrzymanie monitora.

Czasem program monitora musi zostać zatrzymany (zakończony). Wybór opcji 2 pozwala na zakończenie wybranego przez użytkownika programu monitora. Monitor graficzny może przyjmować dwa stany: program monitora nie pracuje i monitor pracuje. Monitor, który został uruchomiony i/lub wybrany będzie wylistowany jako pracujący, monitor, który został zatrzymany (lub nie wystartowany) będzie wylistowany jako nie pracujący. Znacznik `-L` będzie wypisywać stan każdego monitora połączonego do systemu.

3. Wybór monitora.

Kiedy użytkownik uruchamia monitor, jest on też automatycznie wybierany jako wyjście graficzne chyba, że użytkownik ustawił znacznik `-s`. Aby skorzystać z monitora (bezpośrednie wyjście grafiki), użytkownik musi wskazać, poprzez start monitora bez znacznika `-s` lub wybór monitora jako wyjście używając opcji 3, monitor który ma być używany. Tylko pracujące monitory mogą być wybrane jako graficzne wyjście. Jeśli jeden użytkownik wybrał monitor jako wyjście graficzne, żaden użytkownik nie może używać tego monitora jako wyjście graficzne dopóki sterownik monitora nie zostanie zwolniony (przez użytkownika) lub odblokowany (przez innego użytkownika lub system).

Użytkownik może uruchomić wiele monitorów graficznych.

4. Zwalnianie monitora.

Jeśli użytkownik wybrał monitor jako wyjście graficzne, jest on zablokowany do użytku przez niego aż do momentu gdy (1) użytkownik dobrowolnie zwalnia kontrolę nad monitorem (opcja 4) lub (2) inny użytkownik programu GRASS odblokowuje kontrolę nad monitorem. Opcja 4 menu pojawia się tylko dla osoby, która wybrała dany monitor (stąd tylko taki użytkownik może zwolnić kontrolę wybranego monitora graficznego). Jeśli inny użytkownik pragnie odblokować kontrolę nad monitorem, musi uruchomić *d.mon* z linii komend i ustawić parametr **unlock**=*nazwa*.

Możesz wybierać wielokrotnie opcje w programie *d.mon*. Opuszczając menu programu *d.mon*, naciśnij klawisz <ENTER>.

Uwagi

Program *d.mon* może ustalać zasady kontroli monitorów graficznych zarówno w systemie używającym wielu monitorów jak i w systemie zawierającym pojedynczy monitor graficzny.

Zobacz także

d.colormode

5.2.2. d.rast

d.rast – wyświetla i nakłada na siebie warstwy map rastrowych w aktywnej ramce graficznej na monitorze graficznym.

Składnia

d.rast

d.rast help

d.rast [-o] map=*nazwa*

Opis

d.rast wyświetla warstwę mapy rastrowej o nazwie *nazwa* w aktywnej ramce graficznej na monitorze graficznym.

Program może być uruchomiony zarówno w sposób nieinteraktywny jak i interaktywny. Jeżeli użytkownik poda nazwę warstwy mapy rastrowej (**map=*nazwa***) lub poda opcję **-o** w linii komend, program będzie uruchomiony w sposób nieinteraktywny. Alternatywnie użytkownik może podać proste polecenie *d.rast* w linii komend. W tym przypadku program będzie wymagał od użytkownika określenia znacznika ustawień i wartości parametrów zgodnie ze standardem GRASS.

Znacznik:

-o

Nakłada nazwaną warstwę mapy rastrowej na dowolną, która już jest wyświetlana w aktywnej ramce graficznej. Podanie dowolnej zerowej wartości w obszarze danych w nowej mapie rastrowej czyni ją przezroczystą.

Jeśli jest ustawiony znacznik **-o**, tylko komórki zawierające niezerowe wartości są wyświetlane z nakładanej mapy rastrowej i pokrywają obraz poprzednio wyświetlany w ramce graficznej. Wszystkie inne obszary (czyli części nakładanej

mapy, które wartość kategorii mają ustawioną na zero) będą pozostawać nietknięte przy wyświetlaniu. Jeśli znacznik `-o` nie jest ustawiony przez użytkownika `d.rast` będzie dokładnie nadpisywał cokolwiek pojawia się w aktywnej ramce graficznej.

Parametr:

map=nazwa

Nazwa wyświetlanej mapy rastrowej. Jeśli aktywna ramka graficzna zawiera już tekst lub grafikę i użytkownik nie życzy sobie korzystać z opcji `-o`, należy wykorzystać program `d.erase`, aby wyczyścić aktywną ramkę graficzną przed uruchomieniem `d.rast`. Po uruchomieniu `d.rast` inne programy takie jak `d.vect` i `d.grid` mogą być używane do wzbogacania rysunku.

Uwagi

`d.rast` pracuje w opcji nakładania mapy rastrowej (`-o`) tylko wtedy, gdy tablica przeglądowa kolorów dla monitora graficznego jest ustawiona (`d.colormode`).

Zobacz także

`d.colormode`

5.2.3. d.colormode

d.colormode – pozwala użytkownikowi na ustawienie, czy mapa będzie wyświetlana przy użyciu jej własnej tablicy kolorów czy przy pomocy ustalonej tablicy kolorów monitora graficznego.

Składnia

d.colormode

d.colormode help

d.colormode mode=*nazwa*

Opis

d.colormode ustawia jaka tablica kolorów będzie używana do wyświetlania map na monitorze graficznym.

Parametry:

mode=*nazwa*

Opcje: fixed lub float

Zawsze, gdy nowa warstwa mapy rastrowej jest rysowana na ekranie, połączona do mapy tablica kolorów jest ładowana do monitora graficznego. Komenda **d.colormode mode=fixed** żąda, aby tablica kolorów monitora była ustalona (tzn. stała). Efekt tego jest taki, że kolory rysowane na ekranie podczas wywołań poleceń graficznych nie będą się zmieniać, gdy kolejna mapa jest rysowana na ekranie. Kiedy jest używana opcja *fixed*, kolory w tablicy kolorów mapy są odwzorowywane na najbliższy kolor dostępny w ustalonej tablicy kolorów monitora. Kiedy używana jest opcja *float*, tablica kolorów mapy jest bezpośrednio ładowana do tablicy kolorów monitora.

Każda z opcji ustawiania sposobu wyświetlania kolorów ma swoje zalety. Programy *d.display* i *d.colors* pozwalają użytkownikowi interakcyjnie modyfikować tablicę

kolorów mapy wyświetlanej na monitorze graficznym. W ten sposób użytkownik może bezpośrednio modyfikować tablicę kolorów monitora. Stąd, aby móc używać programów `d.display` i `d.colors` należy wybrać opcję `float`. Jednak opcja ta ma czasem niepożądany skutek zmieniając kolory w następnych mapach wyświetlanych na monitorze graficznym (choć faktycznie tablice kolorów tych map nie zostały zmienione). Dlatego tryb `float` najlepiej gdy jest używany wtedy, kiedy użytkownik życzy sobie interakcyjnych zmian w tablicy kolorów mapy lub gdy użytkownik wyświetla jedną lub więcej map mających tą samą tablicę kolorów. Opcja `fixed` pozwala wyświetlać dowolną liczbę map na ekranie, gdzie każda mapa używa innych kolorów, ale wszystkie używają tej samej ustalonej tablicy przeglądowej kolorów. Nie możesz zmienić tablicy kolorów mapy, kiedy pracujesz w trybie `fixed`.

Program może być uruchomiony nieinteraktywnie, jeśli użytkownik określi tryb wyświetlania koloru w linii komend (przez podanie `d.colormode mode=fixed` lub `d.colormode mode=float`). W przeciwnym razie, jeśli użytkownik poda proste polecenie `d.colormode` w linii komend, program będzie wymagał od użytkownika określenia trybu wyświetlania kolorów zgodnie ze standardem interfejsu GRASS.

Uwagi

Czasem urządzenia (np. drukarka atramentowa, ploter) wspierają tryb ustalony koloru. Liczba kategorii koloru możliwych do wyświetlenia jest zależna od urządzenia. Jeśli twoje kolory leżą poza tymi granicami w trybie `float`, nie mogą być wyświetlane w kolorach wskazywanych w pliku tablicy kolorów mapy.

Pliki tablicy kolorów połączone z warstwami map rastrowych są zapisywane w aktualnym mapsecie użytkownika do kartotek `colr` i `colr2`.

Jest bardzo polecane, aby użytkownik czyścił ekran monitora graficznego (przez uruchomienie programu `d.erase`) natychmiast zmianie trybu między `fixed` i `float`.

Zobacz także

d.rast

5.3. Polecenia dotyczące warstw rastrowych

5.3.1. r.mapcalc

r.mapcalc – przelicza dane mapy rastrowej.

Składnia

r.mapcalc

r.mapcalc [wynik=wyrażenie]

Opis

r.mapcalc wykonuje działania na warstwach mapy rastrowej. Dzięki niemu mogą być tworzone nowe warstwy mapy rastrowej, które powstają na podstawie wyrażeń zawierających istniejące warstwy map rastrowych, stałe typu całkowitego lub zmiennoprzecinkowego oraz funkcje.

UŻYWANIE PROGRAMU

Jeżeli program *r.mapcalc* jest używany bez argumentów w linii komend, będzie on czytał dane wejściowe, jedna linię na raz, z wejścia standardowego (którym jest klawiatura). Program *r.mapcalc* oczekuje, że linia wyrażenia wejściowego ma formę:

wynik=wyrażenie

gdzie *wynik* jest nazwą mapy rastrowej zawierającą wyniki obliczeń a *wyrażenie* jest dowolnym prawidłowym wyrażeniem arytmetycznym zawierającym istniejące warstwy mapy rastrowej, stałe typu całkowitego lub zmiennoprzecinkowego, i znane funkcje. Nawiasy są dozwolone w wyrażeniu i mogą być zagnieżdżane dowolnie głęboko. Wynik będzie tworzony w aktualnym zbiorze map użytkownika.

Formuła wprowadzana do *r.mapcalc* przez użytkownika jest zapisywana zarówno w tytule wynikowej mapy (który pojawia się w pliku kategorii) jak i w pliku historii.

Kilka znaków ma specjalne znaczenie dla programu *r.mapcalc*. Jeśli użytkownik wprowadza wyrażenia do *r.mapcalc* z linii komend, wyrażenia powinny być zawarte wewnątrz pojedynczych cudzysłowów. Zobacz UWAGI poniżej.

OPERATORY I KOLEJNOŚĆ WYKONYWANIA DZIAŁAŃ

| Operator | Znaczenie | Typ | Priorytet |
|----------|-----------------------------|--------------|-----------|
| % | modulo (reszta z dzielenia) | arytmetyczny | 4 |
| / | dzielenie | arytmetyczny | 4 |
| * | mnożenie | arytmetyczny | 4 |
| + | dodawanie | arytmetyczny | 3 |
| - | odejmowanie | arytmetyczny | 3 |
| == | równy | logiczny | 2 |
| != | różny | logiczny | 2 |
| > | większy | logiczny | 2 |
| >= | większy lub równy | logiczny | 2 |
| < | mniej | logiczny | 2 |
| <= | mniej lub równy | logiczny | 2 |
| && | koniunkcja (logiczne i) | logiczny | 1 |
| | alternatywa (logiczne lub) | logiczny | 1 |

Operatory są wykonywane od lewej do prawej. Te, które mają wyższy priorytet są wykonywane przez te, które mają niższy priorytet. Dzielenie przez 0 i modulo 0 są dopuszczalne i dają wynik 0. Logiczne operacje dają wynik 1 jeśli porównanie jest prawdziwe i 0 w przeciwnym wypadku.

NAZWY MAP RASTROWYCH

Wszystkie części wyrażenia, które nie są liczbami, operatorami lub nazwami funkcji są przyjmowane jako nazwy map rastrowych. Przykład:

elevation x3 3d.his

Wiele map ma powyższą konwencję nazewnictwa. Jednak, jeśli warstwa mapy rastrowej ma nazwę, która koliduje z powyższymi regułami, powinno to być zaznaczone. Na przykład, wyrażenie:

$x = a - b$

byłoby zinterpretowane jako: x równa się a minus b, podczas gdy

$x = „a-b”$

byłoby zinterpretowane jak: x jest równe warstwie mapy rastrowej o nazwie a-b.

Także

$x = 3107$

utworzyliby warstwę x wypełnioną liczbą 3107, gdy

$x = „3107”$

skopiowałoby warstwę 3107 do warstwy x.

Znaki cudzysłowia nie są wymagane, jeżeli nazwa warstwy mapy rastrowej nie wygląda jak liczba lub nie zawiera operatorów oraz program jest uruchomiony w trybie interaktywnym. Przykłady tu podane przyjmują, że program jest uruchomiony w trybie interaktywnym. Patrz UWAGI poniżej.

r.mapcalc będzie szukać warstw mapy rastrowej zgodnie ze ścieżką dostępu do aktualnego zbioru map podaną przez użytkownika. Można opuścić ścieżkę dostępu i podać bezpośrednio zbiór map zawierający mapę rastrową. To jest możliwe przez podanie nazwy w formie:

nazwa@mapset

Na przykład, następujące wyrażenie jest poprawne:

$wynik = x@PERMANENT / y@GLEBY$

Ta metoda opuszczania ścieżki dostępu do zbioru map jest wspólna dla wszystkich poleceń GRASS, nie tylko dla *r.mapcalc*.

MODYFIKATOR SĄSIEDZTWA

Mapy i obrazy są plikami danych zapamiętanymi w formacie rastrowym tzn. jako dwuwymiarowe macierze o wartościach całkowitych. W *r.mapcalc* mapy mogą powstawać także na podstawie sąsiedztwa elementów. Modyfikator sąsiedztwa

specyfikuje względne przesunięcie względem aktualnej komórki która jest wyliczana. Forma to: mapa[r,c], gdzie r jest przesunięciem wiersza a c jest przesunięciem kolumny. Na przykład, mapa[1,2] odnosi się do komórki w wierszu poniżej i dwie kolumny na prawo od aktualnej komórki, mapa[-2,-1] odnosi się do komórki dwa wiersze powyżej i jedna kolumna na lewo od aktualnej komórki, a mapa[0,1] odnosi się do komórki w kolumnie na lewo od aktualnej komórki. Ta składnia dopuszcza rozwój filtrów opartych o sąsiedztwo wewnątrz pojedynczych map lub w przekroju wielu map.

WARTOŚCI WARSTW MAP RASTROWYCH Z PLIKÓW KATEGORII

Czasem warto używać wartości skojarzonych z etykietami kategorii zamiast wartości kategorii. Jeśli nazwa mapy rastrowej jest poprzedzona operatorem @, wtedy w wyrażeniu zamiast wartości kategorii jest używana etykieta.

Na przykład, przypuśćmy, że warstwa *ph.ziemi* (reprezentująca wartość pH ziemi) ma plik etykiet kategorii jak następuje:

| cat | label |
|-----|---------|
| 0 | no data |
| 1 | 1.4 |
| 2 | 2.4 |
| 3 | 3.5 |
| 4 | 5.8 |
| 5 | 7.2 |
| 6 | 8.8 |
| 7 | 9.4 |

Wtedy wyrażenie

wynik = @ph.ziemi*10

dałoby wynik z wartościami kategorii 0, 14, 24, 35, 58, 72, 88, 94.

Zauważmy, że ten operator może być tylko stosowany do warstwy mapy rastrowej i daje wartości zmiennoprzecinkowe. Więc etykieta kategorii musi zaczynać się poprawną liczbą. Brakujące etykiety lub etykiety, które nie rozpoczynają się liczbą otrzymają wartość 0.

ZAMIENNIKI W SKALI SZAROŚCI LUB SKŁADNIKI RGB

Często jest pomocne operowanie kolorami przypisanymi do kategorii map. To jest szczególnie użyteczne, gdy własności komórki mają znaczenie (jako dane obrazowe) lub kiedy wartości kategorii map przedstawiają prawdziwe ilości (wartości kategorii będą odbijać prawdziwe wartości wysokości). Operowanie kolorami map może pomagać w rozpoznawaniu wzrokowym i drukowaniu map.

Operator # może być używany albo do przekształcania wartości kategorii map do ich zamienników w skali szarości albo do składników RGB.

wynik = #mapa

przekształca każdą wartość kategorii w mapie do wartości z zakresu 0-255, która reprezentuje poziomy skali szarości oznaczających kolor dla kategorii. Jeśli mapa ma tablicę kolorów w skali szarości, wtedy jest z niej wyliczany poziom szarości. W przeciwnym wypadku oblicza się tak:

*.18 * red + .81 * green + .01 * blue*

Operator # ma trzy inne postacie: *r#mapa*, *g#mapa*, *b#mapa*. Te polecenia wyodrębniają odpowiednio składniki czerwony, zielony lub niebieski ze wskazanej mapy rastrowej. Na przykład, aby wyodrębnić czerwony składnik z mapy i zapamiętać w nowej warstwie o zakresie 0-255, użytkownik powinien podać:

czerwony = r#mapa

Aby przypisać tej mapie szare kolory:

r.colors map=red color=rules

black

white

Aby przypisać tej mapie czerwone kolory:

r.colors map=red color=rules

black

red

FUNKCJE

Aktualnie rozpoznawane funkcje są wypisane w tabeli poniżej. Typ wyniku jest wskazywany w ostatniej kolumnie. F oznacza, że funkcja zawsze zwraca wynik jako wartość typu zmiennoprzecinkowego, I oznacza, że funkcje dają wynik całkowity, a * wskazuje że wynik jest typu zmiennoprzecinkowego, jeśli którykolwiek z argumentów funkcji jest typu zmiennoprzecinkowego a typu całkowitego, jeśli wszystkie argumenty są liczbami całkowitymi.

| funkcja | opis | typ |
|------------------------|---|-----|
| abs(x) | wartość bezwzględna z x | * |
| atan(x) | arcus tangens x (wynik w stopniach) | F |
| cos(x) | cosinus z x (x jest w stopniach) | F |
| eval([x,y, ...], z) | oblicza wartości podanych wyrażeń, wyniki wstawia do z | * |
| exp(x) | e do potęgi x | F |
| exp(x,y) | x do potęgi y | F |
| float(x) | konwertuje x do zmiennej typu zmiennoprzecinkowego | F |
| opcje funkcji if | | * |
| if(x) | 1 jeśli x≠0, 0 jeśli x=0 | |
| if(x,a) | a jeśli x≠0, 0 jeśli x=0 | |
| if(x,a,b) | a jeśli x≠0, b jeśli x=0 | |
| if(x,a,b,c) | a jeśli x>0, b jeśli x=0, c jeśli x<0 | |
| int(x) | przekształca x do zmiennej typu całkowitego (obcina część ułamkową) | I |
| log(x) | logarytm naturalny z x | F |
| log(x,b) | logarytm z x przy podstawie b | F |
| max (x, y[, z ...]) | największa wartość z listy | * |
| median (x, y[, z ...]) | mediana elementów z listy | * |
| min (x, y[, z ...]) | najmniejsza wartość z listy | * |
| rand(x,y) | wartość losowa pomiędzy x i y | * |
| round(x) | liczba całkowita najbliższa x | I |
| sin(x) | sinus z x (x podany w stopniach) | F |
| sqrt(x) | pierwiastek kwadratowy z x | F |
| tan(x) | tangens z x (x podany w stopniach) | F |

WARTOŚCI ZMIENNOPRZECINKOWE W WYRAŻENIACH

Liczby typu zmiennoprzecinkowego są dozwolone w wyrażeniach. Liczba typu zmiennoprzecinkowego jest to liczba, która zawiera kropkę dziesiętną:

2.3 12. .81

Wartości typu zmiennoprzecinkowego w wyrażeniach są obsługiwane w specjalny sposób. Przy arytmetycznych lub logicznych operatorach, jeśli argument jest typu zmiennoprzecinkowego, wtedy inne argumenty są konwertowane do tego typu i wynik tej operacji jest także tego typu. To oznacza, w szczególności, że dzielenie wartości całkowitych daje w wyniku wartość całkowitą (przybliżoną), podczas gdy wynik dzielenia liczb typu zmiennoprzecinkowego jest dokładną wartością. Funkcje z typem wyniku * (tabela powyżej) dają w wyniku liczby typu zmiennoprzecinkowego, jeśli jakikolwiek argument jest tego typu, w przeciwnym wypadku ich wynikiem jest liczba całkowita.

PRZYKŁADY

Aby obliczyć średnią z dwu warstw a i b mapy rastrowej należy podać:

$$ave = (a + b)/2$$

Średnia ważona:

$$ave = (5*a + 3*b)/8.0$$

Aby otrzymać dwójkową reprezentację warstwy mapy rastrowej taką, że kategoria 0 pozostaje kategorią 0 a wszystkie inne kategorie otrzymują 1 należy podać:

$$mask = a/a$$

Tu można by też zastosować:

$$mask = if(a)$$

Maskowanie warstwy b przez warstwę a:

$$result = if(a,b)$$

Do przedstawienia wartości NULL używaj funkcji `isnull()`:

```
r.mapcalc "b=isnull (a)"
```

```
r.mapcalc "b=if (isnull (a),1,0)"
```

```
r.mapcalc "b=if (isnull (a), null (), a)"      #nic nie zostanie zmodyfikowane
```

Aby otrzymać mapę typu zmiennoprzecinkowego z istniejącej mapy typu całkowitego:

```
test_fp=1.0*test_int
```

REGION/MASKA

Użytkownik musi być świadomy aktualnego regionu geograficznego i aktualnych ustawień maski kiedy używa `r.mapcalc`. Wszystkie warstwy mapy rastrowej są czytane do aktualnego regionu geograficznego przez aktualną maskę. Jeśli życzymy sobie modyfikować istniejącą warstwę mapy rastrowej bez wklejania innych warstw, region geograficzny powinien być tak ustawiany aby zgadzał się z nagłówkiem warstwy mapy rastrowej. Przypuśćmy, że warstwa wysokości mapy rastrowej musi mieć każdą kategorię zwiększoną o 10 m. Następujące wyrażenie jest poprawne i będzie pracować:

```
new_elevation = elevation +10
```

Ponieważ wartość 0 jest używana w systemie GRASS dla elementów, które nie istnieją, nowa warstwa mapy rastrowej będzie zawierać wartość 10 w komórkach, które nie istnieją w oryginalnej mapie. Stąd, jest ważne, aby granice regionu geograficznego zgadzały się z nagłówkiem mapy.

Jednak jeśli jest ustawiona maska, wynikowa warstwa mapy rastrowej jest maskowana podczas zapisu, tzn. wartość 0 kategorii w masce wymusza wartość 0 na wyjściu.

Uwagi

Należy wyjątkowo uważać przy podawaniu wyrażeń w linii komend. Niektóre znaki mają w systemie UNIX specjalne znaczenie. Są to:

* () > & /

Należy brać wyrażenia w cudzysłów. Tj.

*result = 'elevation * 2'*

Bez cudzysłowia, znak *, który ma specjalne znaczenie w systemie UNIX, byłby zmieniany i *r.mapcalc* widziałby coś innego niż *.

Jeśli dane pochodzą bezpośrednio z klawiatury (tryb interaktywny) i istnieje wynikowa warstwa mapy rastrowej, użytkownik będzie pytany, czy wynik może być nadpisany. W przeciwnym razie, o ile istnieje, wynik będzie automatycznie nadpisany.

Ujmowanie w cudzysłów wyniku nie jest dozwolone. Jednakże, nie jest to konieczne, gdyż domyślnie wynik jest nazwą mapy rastrowej.

Dla formuł, które użytkownik wprowadza z wejścia standardowego (nie z linii komend), jest możliwa kontynuacja zapisu w nowej linii. Jeśli użytkownik doda \ na końcu linii, *r.mapcalc* przyjmuje, że formuła wprowadzana przez użytkownika jest kontynuowana w następnej linii. Stąd można wprowadzać nieograniczoną liczbę linii wejściowych i nie ma ograniczeń co do długości formuły.

Jeżeli formuła wprowadzana przez użytkownika jest bardzo długa, tytuł mapy będzie zawierać tylko jej część, a większość (jeśli nie wszystko) będzie zawierać plik historii dla mapy wynikowej.

Jeśli użytkownik będzie wprowadzał dane wejściowe do *r.mapcalc* w sposób nieinterakcyjny z linii komend, program nie będzie ostrzegał użytkownika przed nadpisaniem istniejących warstw mapy. Dlatego użytkownik powinien sam uważać, aby przypisywać nazwy plików rastrowych, które jeszcze nie istnieją w aktualnym zbiorze map.

Linie kontynuowane musi kończyć się literą a i nie mogą mieć ŻADNYCH pustych znaków na końcu (spacja, tabulator). Jeśli użytkownik puste znaki przy końcu linii, komunikat o błędzie wyświetlony przez *r.mapcalc* nie będzie miał żadnego znaczenia i wyrażenie nie będzie pracowało tak jak użytkownik zamierzał.

Komunikaty o błędach podawane przez *r.mapcalc* są najczęściej bezużyteczne. W przyszłości *r.mapcalc* powinien dokonywać próby określenia miejsca wystąpienia błędu, np.:

$x = a * b + + c$

ERROR: somewhere in line 1: ... b + + c ...

Zobacz także

g.region

5.3.2. r.reclass

r.reclass – Tworzy nową warstwę mapy, której wartości kategorii są oparte o kategorie pierwotnej mapy zgodnie z klasami zdefiniowanymi przez użytkownika.

Składnia

r.reclass

r.reclass help

r.reclass input=name output=name [title=name]

Opis

r.reclass tworzy warstwę mapy opartą na wejściowej warstwie rastrowej. Warstwa mapy wyjściowej będzie przeklasyfikowana z mapy wejściowej w oparciu o wejściowe reguły klasyfikacji i i może być traktowana w ten sam sposób jak traktowane są pliki rastrowe. Tytuł wyjściowej warstwy mapy może być (opcjonalnie) podany przez użytkownika.

Reguły klasyfikowania są czytane ze standardowego wejścia (tzn. z klawiatury, pliku lub innego programu).

Program może być uruchomiony w trybie nieinteraktywnym, jeśli użytkownik poda nazwę warstwy mapy rastrowej do przeklasyfikowania, nazwę warstwy wyjściowej i (opcjonalnie) tytuł dla mapy wyjściowej.

r.reclass input=name output=name [title=name]

Gdy użytkownik wprowadza informacje w linii komend, program będzie wymagał od niego podania reguł klasyfikacji do stosowania dla kategorii warstwy mapy rastrowej. Forma tych reguł jest opisana poniżej.

Użytkownik może też napisać prosto *r.reclass* w linii komend, bez podania argumentów. W tym przypadku użytkownik zostanie odpytany o potrzebne informacje.

Przed użyciem *r.reclass* należy znać:

1. Nowe kategorie i sposób przyporządkowania starych kategorii nowym kategoriom.
2. Nazwy nowych kategorii.

INTERAKTYWNE WYWOŁANIE PROGRAMU: PRZYKŁAD

Przypuśćmy, że chcemy przeklasyfikować warstwę dróg mapy rastrowej, składającą się z pięciu kategorii do trzech nowych kategorii: drogi brukowane, drogi niebrukowane i szlaki kolejowe. Użytkownik jest pytany, czy tablica przeklasyfikowań ma być zainicjowana początkową wartością ustawioną na 0 lub czy początkowa wartość każdej kategorii ma mieć swoją własną wartość. Pojawia się ekran, taki jak ten poniżej, wypisujący kategorie warstwy dróg mapy rastrowej do przeklasyfikowania i pytający użytkownika o nowe wartości, które mają być im przypisane.

```
ENTER NEW CATEGORY NUMBERS FOR THESE CATEGORIES
OLD CATEGORY NAME OLD NEW NUM NUM
no data 00__
Hard Surface, 2 lanes 10__
Loose Surface, 1 lane 20__
Improved Dirt 30__
Unimproved Dirt Trail 40__
Railroad, single track 50__
AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
(OR <Ctrl-C> TO CANCEL)
```

Na następnym ekranie nowe wartości kategorii będą wprowadzane obok stosownej nazwy starej kategorii. Komórkom, które miały przypisaną wartość kategorii 2, 3, 4 w starej warstwie mapy rastrowej teraz są przypisane nowa wartość kategorii 2 w mapie przeklasyfikowanej. Dana komórki dawniej przypisana do wartości 5 w starej mapie rastrowej teraz jest przypisana do nowej wartości kategorii 3.

```

ENTER NEW CATEGORY NUMBERS FOR THESE CATEGORIES
OLD CATEGORY NAME OLD NEW NUM NUM
no data 00__
Hard Surface, 2 lanes 11__
Loose Surface, 1 lane 22__
Improved Dirt 32__
Unimproved Dirt Trail 42__
Railroad, single track 53__
AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
(OR <Ctrl-C> TO CANCEL)

```

Po naciśnięciu klawisza <ESC> pokaże się następny ekran, w którym użytkownik jest pytany o nowy tytuł i etykiety kategorii dla ponownie przeklasyfikowanych kategorii.

```

ENTER NEW CATEGORY NAMES FOR THESE CATEGORIES
TITLE: Roads Reclassified
CATNEW CATEGORY NAME
NUM
0 no data
1 Paved Roads
2 Unpaved Roads
3 Railroad, single track
AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
(OR <Ctrl-C> TO CANCEL)

```

W oparciu o informacje otrzymane od użytkownika z powyższych ekranów, jest tworzona nowa mapa wyjściowa.

NIEINTERAKTYWNE WYWOŁANIE PROGRAMU: REGUŁY KLASYFIKOWANIA

Przy nieinterakcyjnym wykorzystaniu programu, nazwa mapy wejściowej, wyjściowej i tytuł mapy wyjściowej są podawane w linii komend. Jednak reguły przeklasyfikowania mogą być czytane z wejścia standardowego (tj. klawiatury, pliku, innego programu).

Użytkownik wyszczególnia wejściową warstwę mapy rastrowej, nazwę wyjściowej mapy rastrowej i (opcjonalnie) tytuł warstwy wyjściowej mapy przez wprowadzenie:

```
r.reclass input=name output=name [title=name]
```

Każda linijka wprowadzania musi mieć następujący format:

```
kategoria_wejsciowa=kategoria_wyjsciowa [etykieta]
```

gdzie każda linia specyfikuje wartości kategorii w warstwie wejściowej mapy rastrowej do przeklasyfikowania do wartości *kategoria wyjściowa*. Podanie etykiety połączonej z kategorią nowej warstwy mapy jest opcjonalne. Jeśli jest podana, to jest zapamiętana jako etykieta kategorii dla nowej wartości kategorii. Znak równości jest wymagany. Kategoria(e) wyjściowa może składać się z pojedynczych wartości kategorii lub zakresu takich wartości w formacie „od thru do”. Słowo „thru” musi być obecne.

Linia zawierająca tylko słowo end kończy wprowadzanie.

NIEINTERAKTYWNE WYWOŁANIE PROGRAMU: PRZYKŁADY

Następujące przykłady mogą pomóc zrozumieć reguły przeklasyfikowania.

1. Ten przykład przeklasyfikowuje kategorie 1, 3 i 5 w warstwie wejściowej mapy rastrowej do kategorii 1 z etykietką kategorii „poor quality” w warstwie wyjściowej mapy i przeklasyfikowuje kategorie 2, 4 i 6 warstwy wejściowej mapy rastrowej do kategorii 2 z etykietką „good quality” w warstwie wyjściowej mapy.

1 3 5 = 1 poor quality

2 4 6 = 2 good quality

- 2 Ten przykład przeklasyfikowuje kategorie 1 do 10 w wejściowej mapie do kategorii 1 w wyjściowej mapie, kategorie 11 do 20 w wejściowej mapie do kategorii 2 w wyjściowej mapie, kategorie 21 do 30 w wejściowej mapie do kategorii 3 w wyjściowej mapie.

1 thru 10 = 1

11 thru 20 = 2

21 thru 30 = 3

- 3 Późniejsze reguły przeklasyfikowania znoszą poprzednie. Stąd w poniższy przykład przeklasyfikowuje kategorie od 1 do 19 i od 51 do 100 w wejściowej mapie do kategorii 1 w wyjściowej mapie, kategorie 20 do 24 i 26 do 50 w wejściowej mapie do kategorii 2 w wyjściowej mapie i kategorię 25 w wejściowej mapie do kategorii 3 w wyjściowej mapie.

1 thru 100 = 1 poor quality
20 thru 50 = 2 medium quality
25 = 3 good quality

4 Poniższy przykład może być wprowadzony w sposób następujący:

1 thru 19 51 thru 100 = 1 poor quality

20 thru 24 26 thru 50 = 2 medium quality

25 = 3 good quality

lub :

1 thru 19 = 1 poor quality

51 thru 100 = 1

20 thru 24 = 2

26 thru 50 = 2 medium quality

25 = 3 good quality

Ostatni przykład był podany, aby pokazać jak posługiwać się etykietami. Jeśli wartość nowej kategorii pojawia się w więcej niż jednej regule (w tym przypadku są to wartości 1 i 2), ostatnia etykieta, która została podana staje się etykieta dla tej kategorii. W tym przypadku etykiety są przypisywane dokładnie tak jak w dwu poprzednich przykładach.

Uwagi

Faktycznie program *r.reclass* nie generuje żadnych nowych warstw map rastrowych (w interesie ochrony miejsca na dysku). Zamiast tego, jest zapamiętywana tablica przeklasyfikowań, która będzie używana do przeklasyfikowania warstwy oryginalnej mapy rastrowej zawsze, gdy nazwa nowej (przeklasyfikowanej) mapy jest żądana. Z punktu widzenia użytkownika (programisty) mapa rastrowa jest tworzona. Zauważmy też, że chociaż użytkownik może generować mapy *r.reclass*, które są oparte o inne mapy *r.reclass*, nowe mapy *r.reclass* będą pamiętane w programie GRASS jako przeklasyfikowania oryginalnych map rastrowych na której pierwsza przeklasyfikowana mapa była oparta. Dlatego, dopóki GRASS pozwala użytkownikowi na dostarczanie

informacji o warstwach map *r.reclass*, które są oparte na już przeklasyfikowanej mapie (dla wygody użytkownika) żadna warstwa mapy *r.reclass* nie będzie zapamiętywana jako *r.reclass* z *r.reclass*.

Aby zamienić mapę przeklasyfikowaną na warstwę regularnej mapy rastrowej, należy ustawić ustawienia regionu geograficznego odpowiednio do ustawień w nagłówku dla mapy przeklasyfikowanej i uruchomić *r.resample*.

Także program *r.mapcalc* może być używany do zamiany mapy przeklasyfikowanej na warstwę regularnej mapy rastrowej:

```
r.mapcalc raster_map=reclass_map
```

gdzie *raster_map* jest nazwą podaną nowej mapy rastrowej, a *reclass_map* jest istniejącą mapą przeklasyfikowaną.

Ponieważ *r.reclass* raczej generuje tablicę odpowiadającą jakiejś warstwie oryginalnej mapy rastrowej niż tworzy warstwę przeklasyfikowanej mapy rastrowej, warstwa mapy *r.reclass* już nie będzie dostępna jeśli warstwa oryginalnej mapy rastrowej na której jest oparta zostanie usunięta.

Wartości kategorii, które nie są w sposób wyraźny przeklasyfikowane do nowej wartości przez użytkownika, zostaną przeklasyfikowane do wartości 0.

5.3.3. r.slope.aspect

r.slope.aspect – generuje warstwy mapy rastrowej nachylenia, ekspozycji, krzywizny i pochodnych cząstkowych z mapy wysokości rzeczywistych.

Składnia

r.slope.aspect

r.slope.aspect help

```
r.slope.aspect [-aqz] elevation=name [slope=name] [aspect=name] [format=name]  
[zfactor=value] [prec=name] [pcurv=name] [tcurv=name] [dx=name] [dy=name]  
[dxx=name] [dyy=name][dxy=name] [min_slp_allowed=value]
```

Opis

r.slope.aspect generuje warstwy mapy rastrowej nachylenia, ekspozycji, krzywizny i pochodne cząstkowe pierwszego i drugiego rzędu z mapy wysokości rzeczywistych.

Opcje

Program może być uruchomiony nieinteraktywnie, jeżeli użytkownik poda potrzebne dla programu opcje w linii komend, używając formy:

```
r.slope.aspect [-aqz] elevation=name [slope=name] [aspect=name] [format=name]  
[zfactor=value] [prec=name] [pcurv=name] [tcurv=name] [dx=name] [dy=name]  
[dxx=name] [dyy=name][dxy=name] [min_slp_allowed=value]
```

Jeśli użytkownik uruchomi

r.slope.aspect

bez podania argumentów w linii komend, program będzie pytał użytkownika o znaczniki ustawień i wartości parametrów.

Znaczniki:**-a**

Nie wyrównuje ustawień aktualnego regionu geograficznego (do których wyjściowe warstwy nachylenia i ekspozycji mapy będą ustawiane) do tych z warstwy wysokości. Patrz: UWAGI.

-q

Uruchamia program w sposób cichy i ukrywa wydruk informacji o działaniu programu podczas wykonania.

-z

Przyjmuje, że zerowe wartości w mapach wysokości przedstawiają prawdziwe wartości wysokości, nie ma obszarów „bez danych”.

Parametry:**elevation=*name***

Nazwa warstwy mapy rastrowej prawdziwych wartości wysokości używana jako wejście.

slope=*name*

Nazwa warstwy mapy rastrowej wartości nachyleń tworzona z mapy wysokości.

aspect=*name*

Nazwa warstwy mapy rastrowej wartości ekspozycji tworzona z mapy wysokości.

format=*name*

Format raportu o nachyleniach.

Opcje: degrees, percent

Domyślnie: degrees

zfaktor=*value*

Współczynnik do zmiany jednostki wysokości na metry.

Domyślnie: 1.0

pcurv=*name*

Nazwa pliku wyjściowego profilu krzywizny.

tcurv=*name*

Nazwa pliku wyjściowych stycznych do krzywizny.

dx=*name*

Nazwa pliku wyjściowego nachylenia wschód-zachód.

dy=*name*

Nazwa pliku wyjściowego nachylenia północ-południe.

dxx=*name*

Nazwa pliku wyjściowej pochodnej cząstkowej dxx.

dyy=*name*

Nazwa pliku wyjściowej pochodnej cząstkowej dyy.

dxy=*name*

Nazwa pliku wyjściowej pochodnej cząstkowej dxy.

min_slp=value

Minimalna wartość nachylenia (w procentach) dla której jest obliczana ekspozycja.

Domyślnie: 0.0

prec=name

Typ map wyjściowych nachylenia i ekspozycji.

Opcje: default, double, float, int

Wynikowe warstwy nachyleń i ekspozycji map rastrowych są nazywane przez użytkownika i umieszczane w aktualnym zbiorze map (mapset).

MAPY RASTROWE WYSOKOŚCI

Warstwy map rastrowych wysokości podane przez użytkownika muszą zawierać prawdziwe wartości wysokości, nie dane przeskalowane lub sklasyfikowane.

MAPY EKSPOZYCJI

Warstwy map ekspozycji, które są tworzone wskazują kierunek nachylenia. Kategorie ekspozycji są reprezentowane przez liczbę stopni od kierunku wschód i rosną przeciwnie do kierunku ruchu wskazówek zegara: 90=północ, 180=zachód, 270=południe, 360=wschód. Pliki kategorii i tablic kolorów dla map ekspozycji są też generowane.

MAPY NACHYLENIA

Wynik warstwy mapy nachyleń będzie zawierać wartości nachyleń, wyrażane w stopniach pochylenia w stosunku do poziomu, jeśli została wybrana opcja format=degrees (która jest domyślna) lub w procentach, jeśli wybrano opcję format=percent. Pliki kategorii, ale nie pliki tablic kolorów są generowane przez *r.slope.aspect* dla warstw map nachylenia. Profile krzywizn i styczne do krzywizn są

krzywymi odpowiednio w kierunku najbardziej stromego nachylenia i w kierunku konturu styczego.

Dla wielu aplikacji, użytkownik może chcieć używać map przeklasyfikowanych nachylenia, których nachylenia mieszczą się w pewnym zakresie. To może być zrobione z użyciem **r.reclass**. Użyteczny przykład jest podany poniżej:

```
category range category labels
(in degrees) (in percent)
  1      0-1      0-2%
  2      2-3      3-5%
  3      4-5      6-10%
  4      6-8     11-15%
  5      9-11   16-20%
  6     12-14     21-25%
  7     15-90    26% and higher
```

Następująca tablica kolorów dobrze pracuje z powyższymi przeklasyfikowaniami.

| category | red | green | blue |
|----------|-----|-------|------|
| 0 | 179 | 179 | 179 |
| 1 | 0 | 102 | 0 |
| 2 | 0 | 153 | 0 |
| 3 | 128 | 153 | 0 |
| 4 | 204 | 179 | 0 |
| 5 | 128 | 51 | 51 |
| 6 | 255 | 0 | 0 |
| 7 | 0 | 0 | 0 |

Uwagi

Aby zapewnić, że warstwy map wysokości nie są nieodpowiednio próbkowane, ustawienia aktualnego regionu są nieznacznie modyfikowane (tylko podczas wykonywania programu): rozdzielczość jest dopasowana do rozdzielczości mapy wysokości i krawędzie regionu są przesuwane, jeżeli to konieczne, wzdłuż krawędzi najbliższych komórek w mapie wysokości. Jeżeli użytkownik pragnie powtórnego spróbkowania do rozdzielczości aktualnego regionu, powinien być ustawiony znacznik a.

Jeśli jest ustawiona aktualna maska, to jest ona ignorowana.

Algorytmy używane do określania nachylenia i ekspozycji używają 3x3 komórek sąsiadujących z komórką w pliku wysokości. Stąd nie jest możliwe określenie nachylenia i ekspozycji dla komórek przyległych do krawędzi w warstwie mapy wysokości. Te komórki mają przypisaną wartość „brak danych” (kategoria 0) w wartościach nachylenia i ekspozycji.

Ponieważ reguła Horns'a jest używana do znajdowania pochodnej w kierunku x i y, ekspozycja jest przesunięta w kierunku 0, 45, 90, 180, 225, 270, 315 i 360, tzn. rozkład kategorii ekspozycji jest bardzo nierówny, ze szczytami w kategoriach 0, 45, ... 360. Helena Mitasowa z USACERL zaobserwowała, że najczęściej komórek z bardzo małym nachyleniem ostatecznie otrzymuje kategorie 0, 45, ..., 360. Jest czasem możliwe, by zmniejszać błąd w tych kierunkach poprzez odfiltrowanie ekspozycji w obszarach, gdzie teren jest prawie płaski. Nowa opcja

`min_slp=value`

została dodana (minimalne nachylenie, dla którego jest obliczana ekspozycja). Ekspozycja dla wszystkich komórek z nachyleniem $< \text{min_slp}$ jest ustawiana na 0 (brak wartości).

Wysokość zero (jak również poniżej poziomu morza) są prawidłowe. To oznacza, że obszary zawierające wartości 0 mogą mieć dwa możliwe znaczenia: mogą być obszarami „bez danych” lub obszarami mającymi wysokość 0. Jeśli użytkownik pragnie używać `r.slope.aspect` przyjmując że komórki przypisane wartości 0 kategorii w mapach wysokości przedstawiają prawdziwe wartości wysokości, bez obszarów z brakiem danych, użytkownik powinien umieszczać znacznik `-z`.

Jeżeli znacznik `-z` nie jest ustawiana i mapa prawdziwych wysokości zawiera obszary „bez danych”, które są przypisywane do kategorii 0, zarówno przy jego krawędziach jak i w jego wnętrzu wynikiem są niepoprawne nachylenia (zwykle całkiem duże).

Zobacz także

`r.mapcalc`, `r.reclass`

5.4. Wydruki

5.4.1. p.select

p.select – wybiera urządzenie (drukarkę) jako urządzenie drukujące dla systemu GRASS.

Składnia

p.select

p.select help

p.select [-lpq] [painter=*nazwa*]

Opis

p.select pozwala użytkownikowi wybrać urządzenie drukujące dla GRASS. Użytkownik musi wybrać urządzenie zanim uruchomi inne funkcje GRASS dotyczące wydruku (np. *p.map*).

Opcje

Znaczniki:

-l

Wypisuje listę dostępnych drukarek.

-p

Drukuje nazwę aktualnie wybranej drukarki.

-q

Cicho wybiera drukarkę.

Parametr:

painter=nazwa

Nazwa wybieranej drukarki.

Opcje: zostanie wyświetlona lista dostępnych urządzeń drukujących.

TRYB INTERAKTYWNY

Jeśli użytkownik uruchomi *p.select* bez argumentów w linii komend, program będzie żądał od użytkownika nazwy urządzenia drukującego.

Uwagi

Opcje dostępne z programem *p.select* zmieniają się wraz z systemem i zależą od niego.

5.4.2. *p.map.new*

p.map.new – program użytkowy wyjścia kolorowych map.

Składnia:

p.map.new

p.map.new [**input**=nazwa] [**scale**=skala_mapy]

Opis

Polecenie *p.map.new* tworzy mapę dla wyjścia na kolorowym urządzeniu drukującym lub na monitorze graficznym. Wydruk może zawierać mapę rastrową, dowolną liczbę nakładek map wektorowych, dane punktowe, etykiety tekstowe i inne elementy mapy.

To polecenie ma trzy tryby działania. Tryb linii komend żąda uprzednio przygotowanego pliku z instrukcjami, opisujący elementy map do wydruku. Tryb interakcyjny (tzn. bez podawania argumentów w linii komend) pyta użytkownika o składniki do wydruku i nie wymaga od niego przygotowania pliku z instrukcjami. Tryb klawiaturowy wywołuje się przez wprowadzenia znaku „-„ przed parametrem **input**. Instrukcje *p.map.new* będą wtedy wprowadzane z klawiatury.

Parametry z linii komend:

input=*nazwa*

Plik zawierający instrukcje mapowania (lub wprowadź **input**= aby wprowadzić instrukcje z klawiatury).

scale=*skala_mapy*

Skala mapy wyjściowej np. 1:25000

Aktualny region geograficzny określa obszar, który jest drukowany przy użyciu *p.map.new*

INSTRUKCJE NIEINTERAKTYWNEGO GENEROWANIA MAPY

barscale, colormode, colortable, defpat, end, endpanel, grid, labels, legend, line, outline, point, raster, read, region, scale, setcolor, setpat, sites, startpanel, text, vector, verbose.

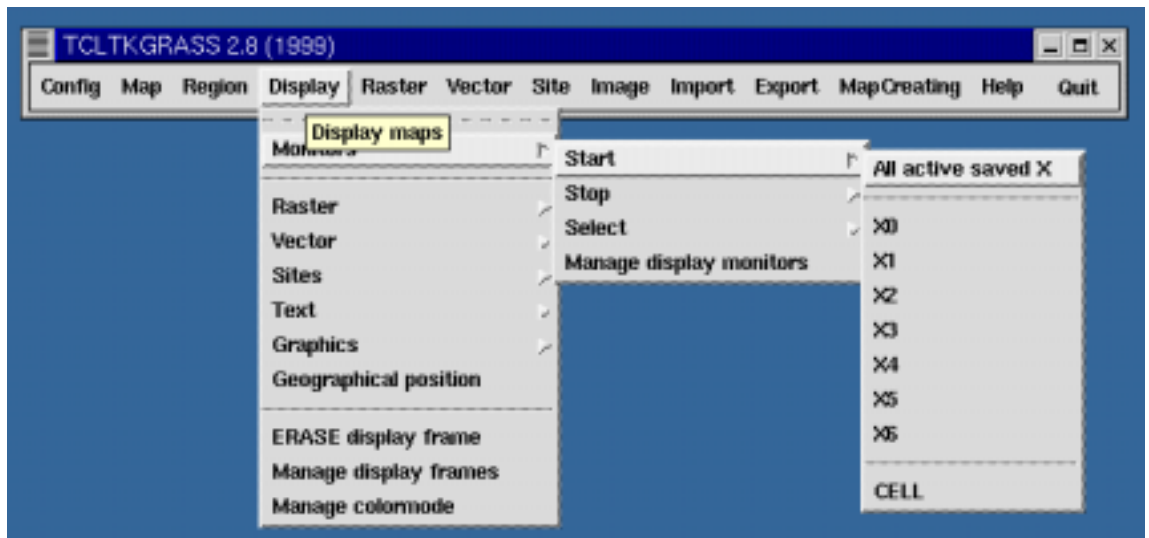
Zobacz także

p.select

6. Interfejs graficzny TcITkGRASS

Używając systemu GRASS najczęściej wpisujemy nazwy poleceń i podajemy ich parametry w oknie terminala. Taka praca nosi nazwę pracy w linii komend. Obecnie rozpowszechniły się systemy zapewniające użytkownikowi interfejs graficzny. Sprowadza to pracę z programami do odpowiedniego manewrowania myszą i (od czasu do czasu) podawania pewnych parametrów. Ten sposób pracy może nie jest szybki i wydajny, ale pozwala osobie mało zorientowanej w pracy z komputerem na osiągnięcie sensownych rezultatów pracy.

System operacyjny UNIX ma możliwość uruchomienia środowiska graficznego. Środowisko to nosi nazwę Xwindows. W związku z tym dziwne by było, gdyby sam program GRASS nie był wyposażony w interfejs graficzny, dający możliwość pracy w oparciu o menu i okienka. Interfejs ten nosi nazwę TcITkGRASS.

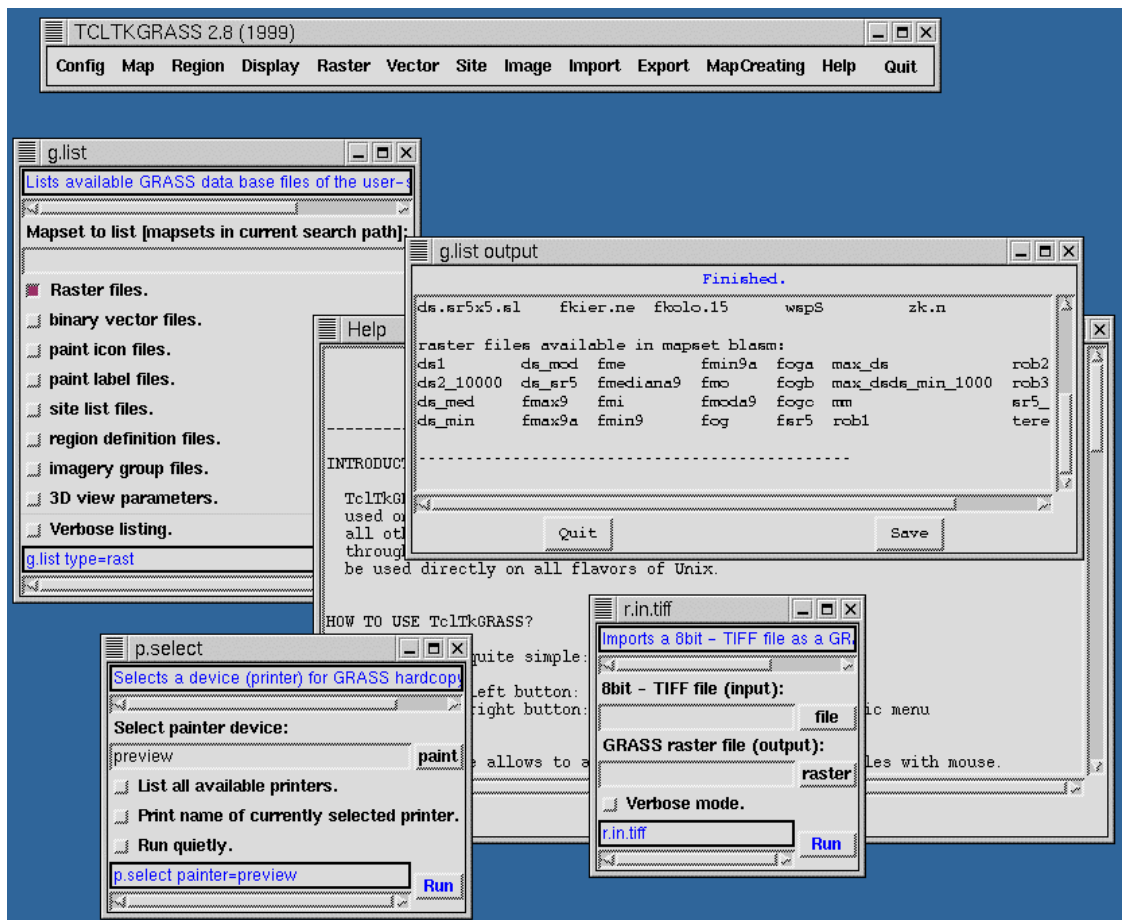


Otwieranie okna graficznego przy pomocy menu TcITkGrass

6.1. Koncepcja interfejsu

TclTkGRASS jest to system okien dający dostęp do wielu (ale nie wszystkich) komend systemu GRASS.

Poza oknem zawierającym główne menu programu, każde okno reprezentuje jedną wybraną komendę systemu.



6.2. Praca w TcITkGRASS

Aby móc skorzystać z możliwości TcITkGRASS, musimy najpierw uruchomić sam system GRASS. Przyjmuję, że komputer ma uruchomiony interfejs graficzny, a więc, że albo pracujemy korzystając z programu X-Terminal albo używamy X-Windows.

Po uruchomieniu systemu GRASS wydajemy polecenie „tcltkgrass&” (znak „&” nie jest błędem – oznacza on, że interfejs graficzny ma być uruchomiony w tle). Komenda ta powoduje wyświetlenie okna, zawierającego główne menu programu.

Teraz już możemy, wybierając odpowiednie pozycje z menu, otwierać okna graficzne, wyświetlać warstwy oraz wykonywać inne czynności.

Typowe okno TcITkGRASS w swoim nagłówku zawiera nazwę polecenia, które go dotyczy.

TcITkGrass jest rozwiązaniem dla osób posługujących się systemem GRASS od czasu do czasu. Prawdziwa potęga systemu ujawnia się bowiem w momencie wykorzystania możliwości zautomatyzowania prac poprzez użycie skryptów. Ty niemniej przy mniej skomplikowanych pracach można poprzestać na obsłudze menu za pomocą myszy. Dla wszystkich użytkowników wychowanych na środowisku okienkowym będzie to zdecydowanym ułatwieniem.

7. Dodatki

7.1. Lista podstawowych poleceń systemu GRASS

Podstawowe polecenia systemu GRASS

| | | | | |
|----------------|--------------|----------------|----------------|----------------|
| d.3d | d.what.vect | i.tape.mss | r.combine | r.watershed4.0 |
| d.ask | d.where | i.tape.mss.h | r.compress | r.weight |
| d.colormode | d.zoom | i.tape.other | r.covar | r.what |
| d.colors | exit | i.tape.tm | r.cross | s.in.ascii |
| d.colortable | g.access | i.target | r.describe | s.menu |
| d.display | g.ask | i.zc | r.drain | s.out.ascii |
| d.erase | g.copy | m.datum.shift | r.grow | s.surf.idw |
| d.font | g.filename | m.dem.examine | r.in.ascii | v.area |
| d.frame | g.findfile | m.dmaUSGSread | r.in.ll | v.cadlabel |
| d.geodesic | g.gisenv | m.dted.examine | r.infer | v.clean |
| d.graph | g.help | m.dted.extract | r.info | v.digit |
| d.grid | g.list | m.examine.tape | r.los | v.digit2 |
| d.his | g.manual | m.gc2ll | r.mapcalc | v.import |
| d.histogram | g.mapsets | m.ll2gc | r.mask | v.in.ascii |
| d.icons | g.region | m.ll2u | r.mfilter | v.in.dlg |
| d.label | g.remove | m.region.ll | r.neighbors | v.in.dlg2 |
| d.legend | g.rename | m.rot90 | r.out.ascii | v.in.dxf |
| d.mapgraph | g.setproj | m.u2ll | r.patch | v.mkgrid |
| d.measure | g.tempfile | p.chart | r.poly | v.mkquads |
| d.menu | g.version | p.colors | r.profile | v.out.ascii |
| d.mon | i.cca | p.icons | r.random | v.out.dlg |
| d.paint.labels | i.class | p.labels | r.reclass | v.out.dxf |
| d.points | i.colors | p.map | r.report | v.patch |
| d.profile | i.composite | p.ppm | r.resample | v.prune |
| d.rast | i.fft | p.select | r.rescale | v.spag |
| d.rgb | i.grey.scale | parser | r.slope.aspect | v.stats |

GIS - GRASS - wprowadzenie do systemu

| | | | | |
|-------------|-----------|---------------|----------------|--------------|
| d.rhumbline | i.group | r.average | r.stats | v.support |
| d.save | i.his.rgb | r.basins.fill | r.support | v.to.rast |
| d.scale | i.ifft | r.binfer | r.surf.idw | v.to.sites.b |
| d.sites | i.maxlik | r.buffer | r.surf.idw2 | v.transform |
| d.text | i.pca | r.cats | r.transect | v.trim |
| d.title | i.points | r.clump | r.volume | |
| d.vect | i.rectify | r.coin | r.water.outlet | |
| d.what.rast | i.rgb.his | r.colors | r.watershed | |

Polecenia systemu GRASS w wersji testowej alfa

| | | | | |
|----------------|----------------|----------------|----------------|----------------|
| d.labels | i.vpoints | ps.icon | r.nntool | v.cutter |
| d.rast.arrow | m.dem.extract | ps.map | r.out.mpeg | v.in.arc |
| d.rast.edit | m.flip | ps.select | r.out.ppm | v.in.dxf3d |
| d.rast.leg | m.in.pl94.db3 | r.bilinear | r.ros | v.in.tig.basic |
| d.rast.num | m.in.stf1.db3 | r.cn | r.spread | v.in.tig.lndmk |
| d.rast.zoom | m.in.stf1.tape | r.contour | r.spreadpath | v.in.tig.rim |
| i.cluster | m.lulc.USGS | r.cost | r.stage3 | v.in.transects |
| i.gensig | m.lulc.read | r.digit | r.sun | v.out.arc |
| i.gensigset | m.proj | r.flow | r.surf.contour | v.out.idrisi |
| i.in.erdas | m.tiger.region | r.hydro.CASC2D | r.thin | v.out.moss |
| i.ortho.photo | p.map.new | r.in.poly | r.water.fea | v.proj |
| i.quantize | photo.2image | r.in.sunrast | r.weight2 | v.reclass |
| i.rectify2 | photo.2target | r.line | s.surf.tps | v.to.sites |
| i.smap | photo.camera | r.mask.points | v.alabel | xganim |
| i.tape.spot | photo.init | r.median | v.apply.census | |
| i.tape.tm.fast | photo.rectify | r.mode | v.circle | |

Skrypty systemu GRASS

| | | | | |
|----------------|---------------|---------------|---------------|---------------|
| 3d.view.sh | grass.logo.sh | rgb.hsv.sh | slide.show.sh | tiger.info.sh |
| blend.sh | hsv.rgb.sh | shade.rel.sh | split.sh | v.in.dxf3d.sh |
| bug.report.sh | i.oif | show.color.sh | start.man.sh | |
| d.correlate.sh | old.cmd.sh | show.fonts.sh | tig.rim.sh | |

Programu dołączone do systemu GRASS

| | | | |
|---------------|-----------------------|----------------|----------------|
| DOS.delete | m.get.stp | r.tribs | v.in.tiger.scs |
| DOS.list | m.in.e00 | s.db.rim | v.make.subj |
| DOS.save | m.qcalc | s.delaunay | v.merge |
| DOS.show | m.regression | s.geom | v.mkgrid |
| ISMann2dig | m.sdts.read | s.in.ascii.dem | v.mkquads |
| ISMcell2grd | m.setproj | s.sample | v.out.dlg.scs |
| ISMcell2scat | m.stp.proj | s.surf.bissf | v.out.scsgef |
| ISMcont2dig | newman | s.surf.krig | v.out.sdts |
| ISMdig2ann | r.affinity | s.sv | v.plant |
| ISMgrd2cell | r.answers | s.sweep | v.proj |
| d.6386.delete | r.distance | s.to.vect | v.psu |
| d.6386.save | r.flow | s.voronoi | v.psu.subj |
| d.6386.show | r.in.erdas | s.windavg | v.random |
| d.igraph | r.in.miads | v.autocorr | v.reclass |
| d.linegraph | r.kappa | v.db.rim | v.reclass.inf |
| d.param.scale | r.linear.regression | v.export | v.report |
| d.rast.pg | r.ndvi.model | v.extract | v.rmdup |
| d.site.pg | r.out.ascii | v.geom | v.rmedge |
| d.to.sites | r.out.mpeg | v.import | v.scale.random |
| d.vect.cats | r.out.pov | v.in.dlg.scs | v.sdts.dq.cp |
| i.texture | r.param.scale | v.in.dxf3d | v.sdts.meta |
| m.bsplint | r.rational.regression | v.in.poly | v.sdts.meta.cp |
| m.eigensystem | r.reclass.scs | v.in.scsgef | v.what |
| m.geo | r.rescale.inf | v.in.sdts | v.zoom |
| m.get.fips | r.statistics | v.in.shape | xganim |

7.2. Program awk

Niektóre polecenia systemu GRASS w wyniku swojego działania tworzą pliki tekstowe. Są to na przykład pliki zawierające statystyki warstwy rastrowej, histogram częstości

wartości w rastrze, definicje kolorów itp. Niekiedy okazuje się przydatne wprowadzenie tych danych jako argumentu innej funkcji. W przypadku małych zbiorów ich przeredagowanie z pomocą dowolnego edytora tekstowego nie stanowi problemu. Inaczej sprawa się ma w przypadku, gdy zbiór tekstowy liczy dziesiątki/setki linii lub gdy chcemy zautomatyzować przetwarzanie warstw.

W takich przypadkach bardzo przydatny okazuje się program „awk”. Program ten jest częścią systemu UNIX i służy do automatycznego przetwarzania plików tekstowych. Pozwala on na pisanie specyficznych skryptów, pozwalających na przetwarzanie danych w pliku wejściowym i przesyłanie wyników pracy na ekran lub do pliku. Język, jakim posługuje się „awk”, zbliżony jest do języka C. Aby uzyskać więcej informacji na temat programu „awk”, polecam literaturę dotyczącą programowania w systemie UNIX.

7.3. Praca na serwerze

System GRASS może być wykorzystywany jako oprogramowanie stacji roboczej GIS. Może on być także wykorzystywany jako platforma do pracy grupowej. Jest to tym bardziej uzasadnione, że najczęściej całe zespoły opracowują dane, dotyczące danego terenu. W przypadku pracy grupowej zarówno dane jak i oprogramowanie przechowywane jest na jednym komputerze - serwerze.

Ponieważ w naszych warunkach najczęściej spotykanym systemem operacyjnym komputerów osobistych jest Windows, konieczne jest wykorzystanie specjalnego oprogramowania do łączenia się z serwerem (pracującym w systemie UNIX). Taki program nosi nazwę X-Terminal.

Istnieje kilka rozwiązań X-Terminali dla Windows. Są to programy zarówno komercyjne jak i darmowe. W naszym przypadku zajmiemy się darmowym programem X-Terminal firmy MicroImages o nazwie MIX95. Dodatkowym plusem korzystania z tego programu jest to, że procedura podłączania się do serwera jest dokładnie taka sama, jak procedura konieczna do zdalnej rejestracji wykonywanej z systemu UNIX.

Kolejne etapy uruchamiania programu:

- Uruchomienie systemu Windows

Ten punkt nie wymaga chyba komentarza. Procedura uruchamiania Windows następuje najczęściej natychmiast po włączeniu komputera. W trakcie startu Windows można, lub też nie, zarejestrować się w sieci. Rejestracja w sieci niezbędna jest jedynie, gdy chcemy korzystać z zasobów sieciowych bezpośrednio w systemie Windows.

- Uruchamianie programu X-Terminal

Aby uruchomić MIX95, należy przejść do kartoteki, w której jest on zainstalowany. Następnie uruchamiamy program `tnstart.exe`. W pewnych przypadkach uruchomienie programu `TNTstart` może doprowadzić do zawieszenia systemu. W takim przypadku trzeba kolejno uruchomić programy `xs.exe` oraz `twm.exe`. Czynności te najlepiej wykonać z poziomu linii komend.

- Przyłączanie się do serwera

W celu uzyskania dostępu do serwera uruchamiamy program „telnet”. Program ten jest częścią systemu Windows. Można go uruchomić używając pozycji „uruchom” menu „start”. Jako parametr można podać nazwę serwera, do którego chcemy uzyskać dostęp. Zamiast nazwy symbolicznej można podać także numer IP serwera.

Po uruchomieniu programu „telnet” na ekranie pojawi się okno terminala. W tym momencie pracujemy już bezpośrednio na serwerze.

Komunikaty wyświetlane w oknie terminala generowane są bezpośrednio przez serwer. Należy podać nazwę użytkownika oraz hasło. Jeżeli nazwa i hasło zostały podane poprawnie, pojawi się znak zachęty, zawierający nazwę użytkownika oraz serwera.

- Otwieranie okna graficznego w programie X-Terminal

W oknie terminala, w linii komend serwera trzeba teraz wpisać polecenie pozwalające na otwarcie okna graficznego i wyświetlenie go na naszym komputerze.

Polecenie, pozwalające na otwarcie okna w systemie X-Windows, nosi nazwę „xterm”.

Uwaga! System UNIX rozróżnia duże i małe litery.

Aby przekierować wyświetlanie okna na pulpit naszego X-Terminala, wykorzystamy parametr „display”.

Polecenie w linii komend powinno wyglądać następująco:

```
xterm -display nasz_komputer:0.0
```

Zamiast nasz_komputer podajemy adres IP naszego komputera (można go uzyskać przy pomocy programu „winipcfg”) lub, jeżeli komputer jest wpisany w serwerze DNS (jest rozpoznawany w sieci przez nazwę), możemy podać nazwę symboliczną (np. stratus.meteo.uni.wroc.pl). Po nazwie komputera występuje ciąg znaków „:0.0”. Mówi on serwerowi, na który ekran wirtualny naszego komputera ma być skierowane wyświetlanie.

Po wykonaniu tych czynności w oknie programu MIX95 pojawi się okienko z linią komend. Każdy program uruchamiany z tego okna będzie rezultaty swojej pracy wyświetlał już na naszym komputerze.

Spis treści

| | |
|--|-----------|
| 1. WSTĘP | 1 |
| 2. SYSTEMY INFORMACJI GEOGRAFICZNEJ - GIS | 3 |
| 2.1. GIS - co to jest | 3 |
| 2.2. Informacje gromadzone w systemie GIS | 3 |
| 2.2.1. Systemy wektorowe | 3 |
| 2.2.2. Systemy rastrowe | 4 |
| 2.2.3. Systemy warstwowe | 4 |
| 2.2.4. Systemy obiektowe | 4 |
| 2.3. System GIS a systemy generowania map | 5 |
| 3. SYSTEM GRASS | 6 |
| 3.1. System GRASS - co to jest | 6 |
| 3.2. Historia systemu GRASS | 7 |
| 3.3. Wymagania systemu GRASS | 7 |
| 3.4. Koncepcja i struktura systemu GRASS | 8 |
| 3.4.1. Organizacja systemu | 8 |
| 3.4.2. Grupy komend | 8 |
| 3.4.3. Opis komend - pomoc | 9 |
| 3.4.4. Organizacja danych | 10 |
| 3.4.5. Struktura katalogów | 11 |
| 3.4.6. Przenoszenie danych | 13 |
| 4. SESJA W SYSTEMIE GRASS | 14 |
| 4.1. Uruchomienie programu | 14 |
| 4.2. Przeglądanie zgromadzonych danych | 18 |
| 4.2.1. Spis warstw | 18 |
| 4.2.2. Uruchomienie okna graficznego | 19 |
| 4.2.3. Wyświetlenie warstwy | 20 |

| | | |
|-------------|--|-----------|
| 4.2.4. | Zamknięcie okna graficznego | 22 |
| 4.3. | Wykonywanie podstawowych obliczeń na warstwach rastrowych | 22 |
| 4.3.1. | Wykorzystywanie polecenia r.mapcalc | 23 |
| 4.3.2. | Wykorzystywanie polecenia r.reclass | 26 |
| 4.4. | Wyprowadzanie rezultatów obliczeń w postaci mapy | 26 |
| 4.4.1. | Sterownik CELL | 26 |
| 4.4.2. | Wydruk na drukarce | 29 |
| 4.4.3. | Wydruk w formacie Postscript | 30 |
| 4.4.4. | Inne formy wyprowadzania wyników | 31 |
| 4.5. | Zakończenie pracy z programem | 31 |
| 5. | WYBRANE POLECENIA SYTEMU GRASS | 34 |
| 5.1. | Polecenia ogólne | 35 |
| 5.1.1. | g.list | 35 |
| 5.1.2. | g.region | 37 |
| 5.2. | Polecenia wyświetlania na ekranie | 44 |
| 5.2.1. | d.mon | 44 |
| 5.2.2. | d.rast | 48 |
| 5.2.3. | d.colormode | 50 |
| 5.3. | Polecenia dotyczące warstw rastrowych | 52 |
| 5.3.1. | r.mapcalc | 52 |
| 5.3.2. | r.reclass | 62 |
| 5.3.3. | r.slope.aspect | 68 |
| 5.4. | Wydruki | 74 |
| 5.4.1. | p.select | 74 |
| 5.4.2. | p.map.new | 75 |
| 6. | INTERFEJS GRAFICZNY TCLTKGRASS | 77 |
| 6.1. | Koncepcja interfejsu | 78 |
| 6.2. | Praca w TclTkGRASS | 79 |
| 7. | DODATKI | 80 |

| | | |
|-------------|---|-----------|
| 7.1. | Lista podstawowych poleceń systemu GRASS | 80 |
| 7.2. | Program awk | 82 |
| 7.3. | Praca na serwerze | 83 |