

Space Applications For Environment

MODIS Image Processing using GRASS in GIS-Knoppix ver. 1.2



宇宙航空研究開発機構
Japan Aerospace Exploration Agency



Version 1.3

July. 28, 2009

The latest version of this tutorial is provided at
SAFE portal site:

<http://www.safe.iis.u-tokyo.ac.jp/>

Contents

Section 1. Preliminary Preparation

- 1. How to get GIS-Knoppix CD**
- 2. Copy data to FAT formatted storage device**
- 3. How to boot a PC from GIS-Knoppix CD**
- 4. Preparation for starting GRASS**

Section 2. Data Processing in Latitude-Longitude Coordinate System

- 1. Starting and Terminating GRASS**
- 2. Defining coordinate system and region**
- 3. Import raster data**
- 4. Display raster data**

Contents

- 5. Import vector data**
- 6. Display vector data**
- 7. Overlay raster and vector data**
- 8. Create RGB image**
- 9. Raster calculations**
- 10. Raster mosaic**
- 11. Raster statistics**
- 12. Land cover classification**
- 13. Principal Component Analysis**
- 14. 3-D display**
- 15. Export raster**

Section 1. Preliminary Preparation

1. How to get GIS-Knoppix CD

1.1 What is GIS-Knoppix ?

- GIS-Knoppix is a bootable Linux CD with pre-installed GIS software.
- GIS-Knoppix is based on Knoppix.

- What is Knoppix?

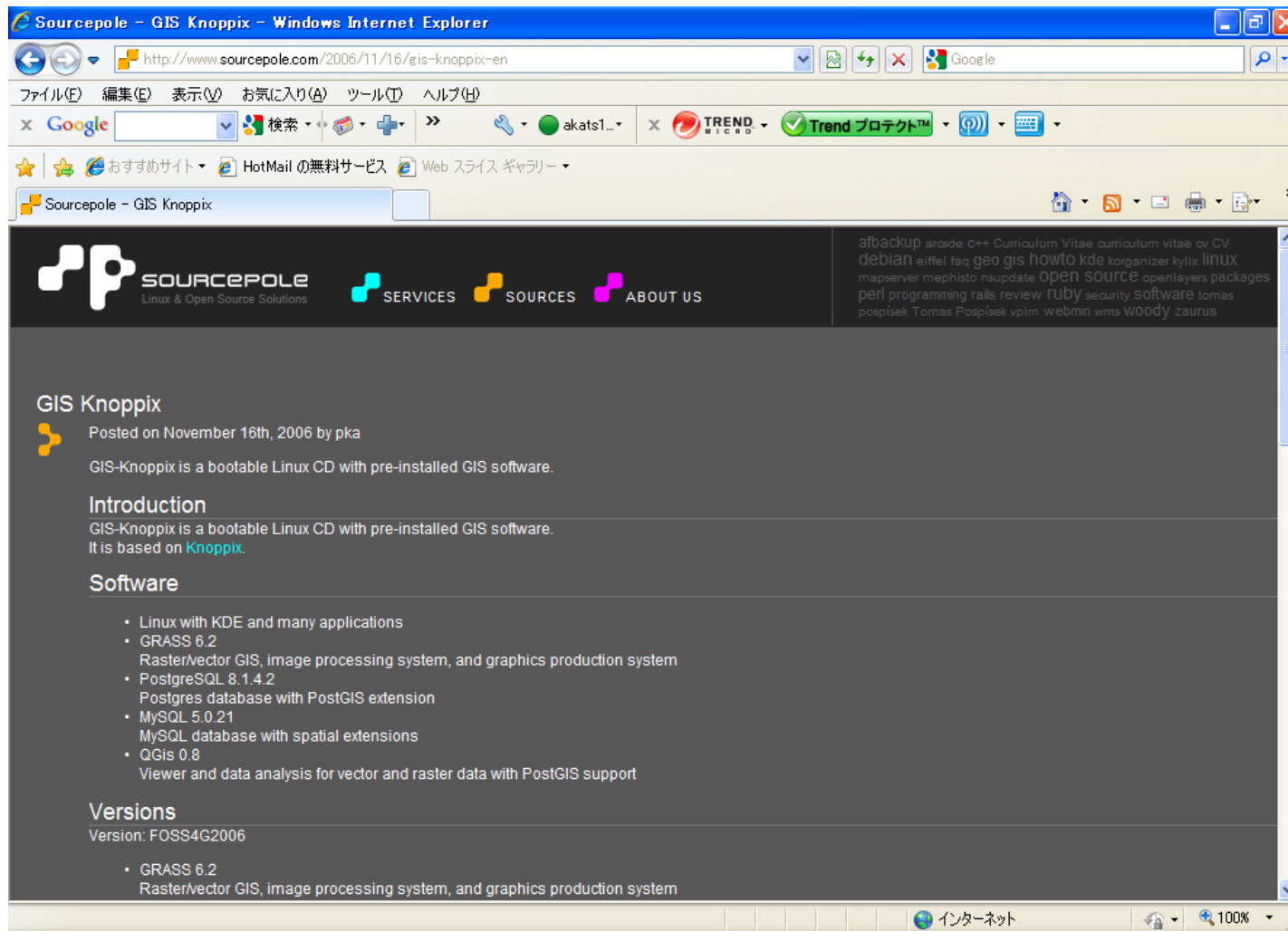
KNOPPIX is a bootable Live system on CD or DVD, consisting of a representative collection of GNU/Linux software, automatic hardware detection, and support for many graphics cards, sound cards, SCSI and USB devices and other peripherals. KNOPPIX can be used as a productive Linux system for the desktop, educational CD, rescue system, or adapted and used as a platform for commercial software product demos. It is not necessary to install anything on a hard disk.

(<http://www.knopper.net/knoppix/index-en.html>)

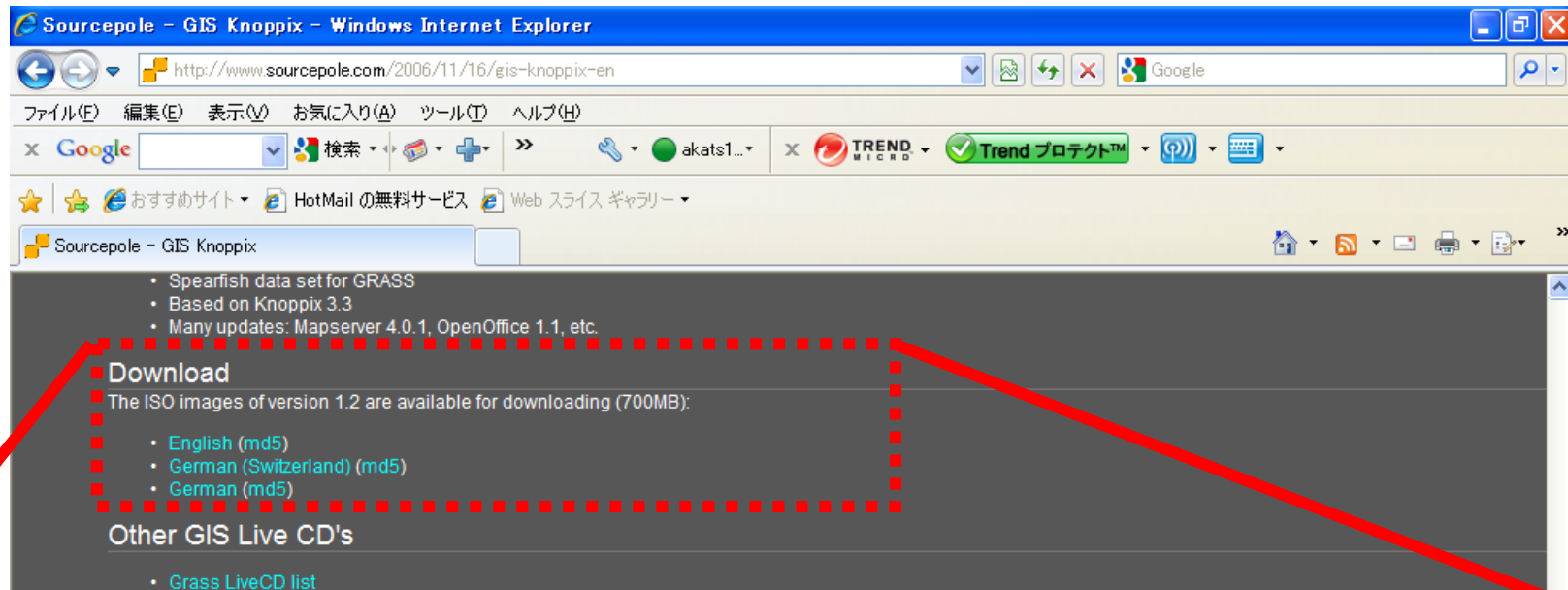
1.2 How to get GIS-Knoppix CD

Step 1: Go to

“ <http://www.sourcepole.com/2006/11/16/gis-knoppix-en> ”



Step 2: Download the ISO images of version 1.2

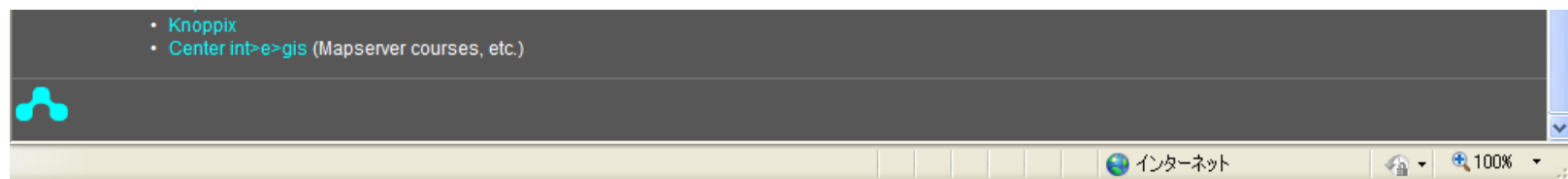


Download

The ISO images of version 1.2 are available for downloading (700MB):

- English (md5)
- German (Switzerland) (md5)
- German (md5)

Click and Download



Step 3: Burn the downloaded ISO image file to a CD

- Burning the ISO image file to a CD or DVD is a little different than just burning the file. To do it, you'll need to choose the "burn image" or "write image" option in your burning software and then choose the file.
- If your CD/DVD burning software does not have this option, download free CD/DVD burning software.
 - ⇒ see “<http://pcsupport.about.com/od/toolsoftthetrade/ht/burnisofile.htm>”

Section 1. Preliminary Preparation

2. Copy data to FAT formatted storage device

Using your windows PC,

Step 0: Insert data DVD into disc drive

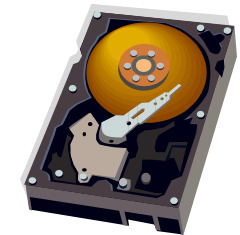
**Step 1: Make “gis_knoppix” directory
in your FAT formatted storage
device**

**Step 2: Make “data” directory
under the “gis_knoppix”
directory**

**Step 3: Copy data into “data”
directory from data DVD**

see next slide

FAT formatted
storage device



Data list in “Data” directory (minimum)

- MOD_Vietnam20070401-10_HKM_CH1.raw
- MOD_Vietnam20070401-10_HKM_CH2.raw
- MOD_Vietnam20070401-10_HKM_CH3.raw
- MOD_Vietnam20070401-10_HKM_CH4.raw
- MOD_Vietnam20070401-10_HKM_CH5.raw
- MOD_Vietnam20070401-10_HKM_CH6.raw
- MOD_Vietnam20070401-10_HKM_CH7.raw

from DVD

“ Vietnam/data/MODIS ”

- BU_LC_Vietnam_1KM.raw
- LSM_Vietnam_HKM
- ROI_Vietnam

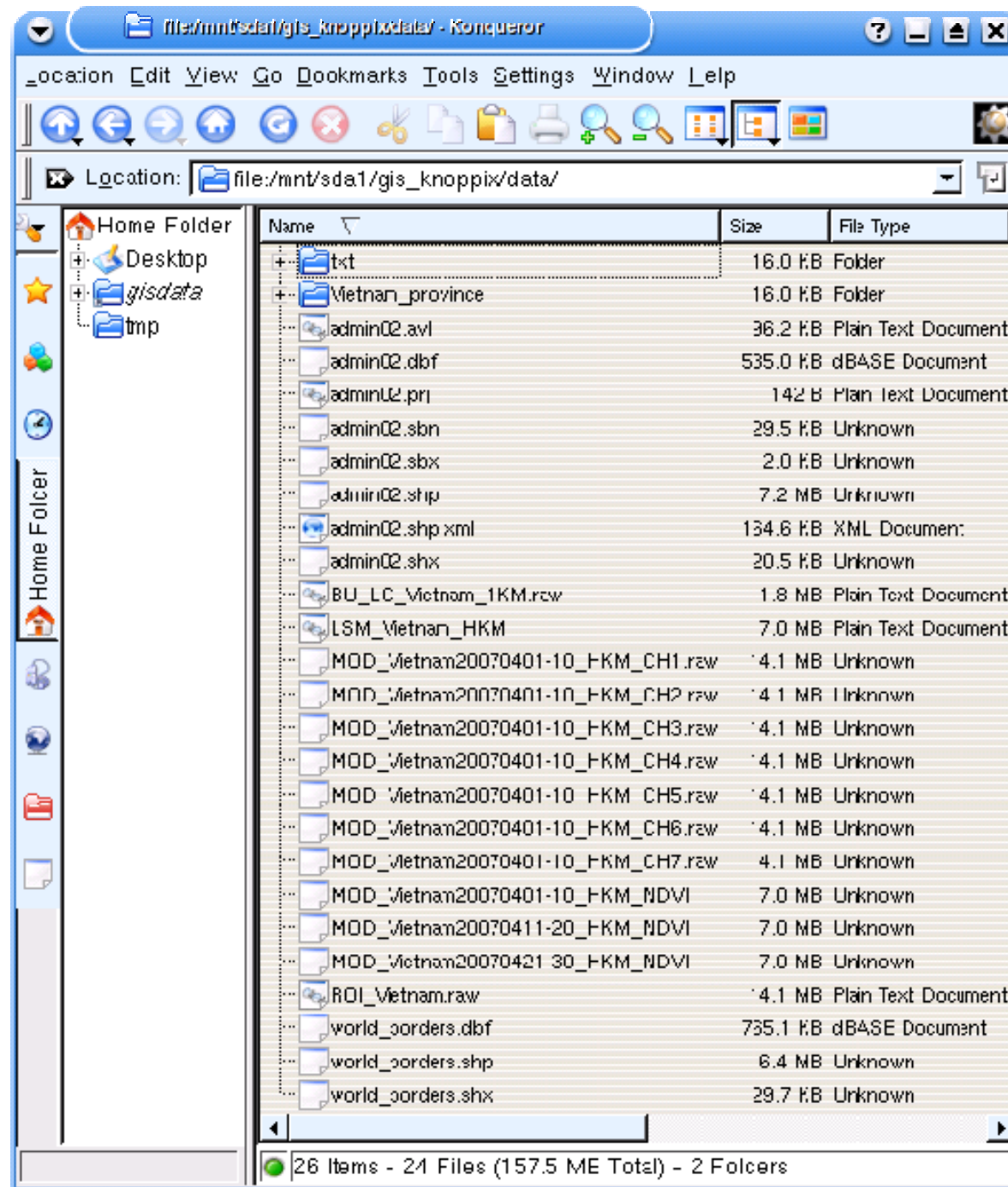
from DVD “ Vietnam/data/rast ”

- world_borders.shp , world_borders.shx , world_borders.dbf

from DVD “ Vietnam/data/vect/World_border ”

- “Vietnam_province” folder

from DVD “ Vietnam/data/vect/Vietnam ”



Section 1. Preliminary Preparation

3. How to boot a PC from the GIS-Knoppix CD

Step 1: Change the BIOS boot order so the CD drive is listed first. Some computers are already configured this way but many are not.

If the CD drive is not first in the boot order, your PC will start "normally" (i.e. boot from your hard drive) without even looking at what might be in your disc drive.

Note: After setting your optical drive as the first boot device in BIOS, your computer will check that drive for a bootable CD each time your computer starts. Leaving your PC configured this way shouldn't cause problems unless you plan on leaving a disc in the drive all the time.

Step 2: Insert your GIS-Knoppix CD in your disc drive. a

Step 3: Restart your computer.

Step 4: Watch for a “*Press any key to boot from CD...*” message.

To boot from the CD, you'll need to press any key on your keyboard (like the space bar) within the few seconds that the message is on the screen.

If you do nothing, your computer will check for boot information on the next boot device in the list in BIOS (see Step 1) which will probably be your hard drive.

Step 5: Your computer should now boot from the CD disc.

If you tried the above steps but your computer did not boot from the CD or DVD properly, check out some of the tips below.

Tips :

1. Recheck the boot order in BIOS (Step 1).

The number one reason a bootable disc won't boot is because BIOS is not configured to check the CD/DVD drive first.

2. Do you have more than one CD or DVD drive?

Your computer probably only allows for one of your disc drives to be booted from. Insert the GIS-Knoppix CD in the other drive and restart your computer.

3. Burn a new CD.

The disc may have errors on it that reburning could correct.

Reference

- How To Boot your Computer from a Bootable CD or DVD
by Tim Fisher, About.com

(<http://pcsupport.about.com/od/tipstricks/ht/bootcddvd.htm>)

Section 1. Preliminary Preparation

4. Preparation for starting GRASS

- **First time to work in GIS-koppix**

- 4.1 Boot PC from the GIS-knoppix CD

- 4.2 Change Keyboard layout

- 4.3 Make FAT formatted storage device writable

- 4.4a Make working directory

- **Second time or later to work in GIS-koppix**

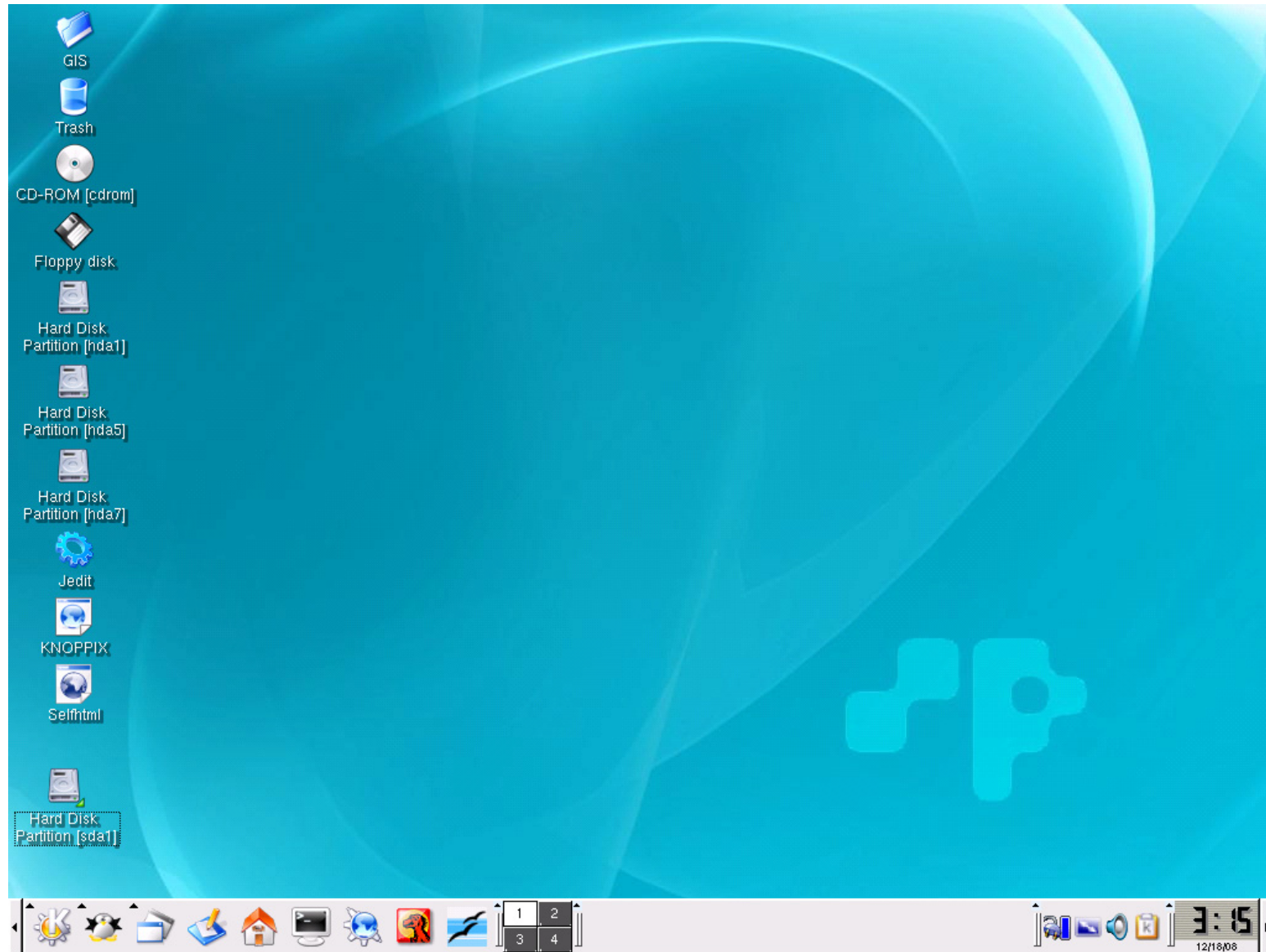
- 4.1 Boot PC from the GIS-knoppix CD

- 4.2 Change Keyboard layout

- 4.3 Make FAT formatted storage device writable

- 4.4b Move to “work” directory

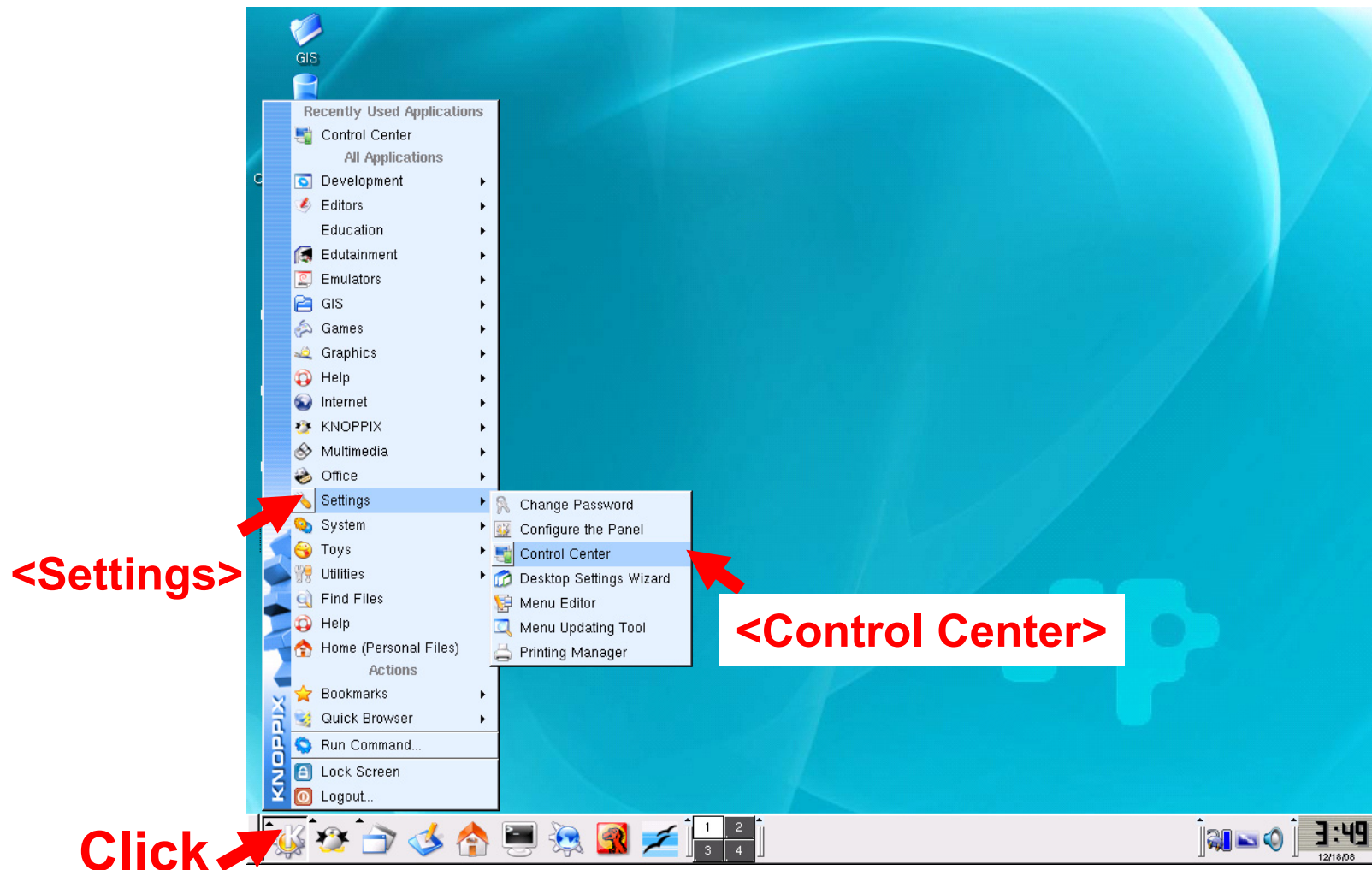
4.1 Boot PC from the GIS-knoppix CD



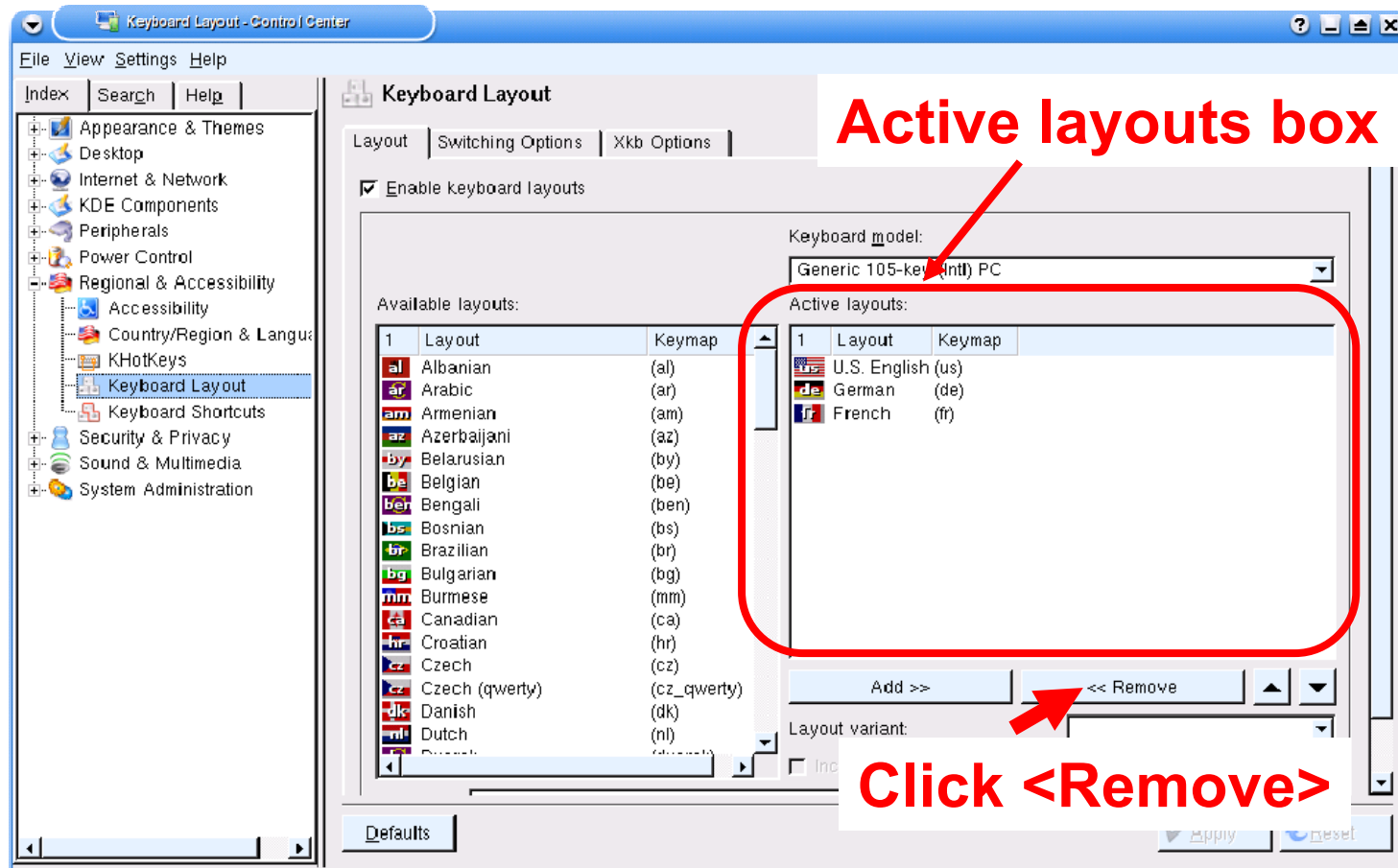
4.2 Change Keyboard layout

Step1 : Click icon at bottom-left corner

→ <Settings> → <Control Center>

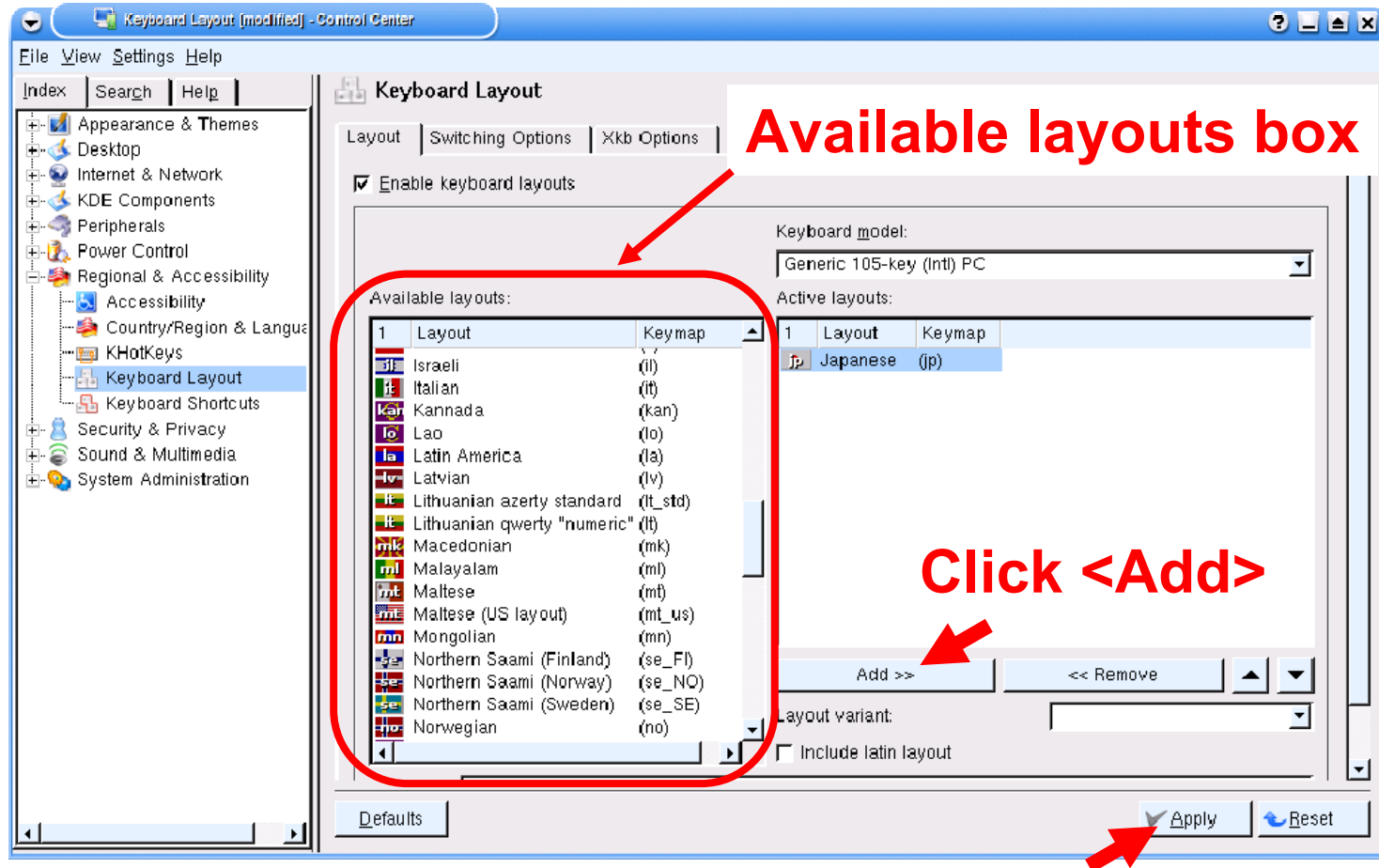


Step2 : Click <Regional & Accessibility >
→ Click <Keyboard Layout >



Step3 : Select languages from Active layouts box
→ Click <Remove>

Step4 : Select your language from Available layouts box → Click <Add>

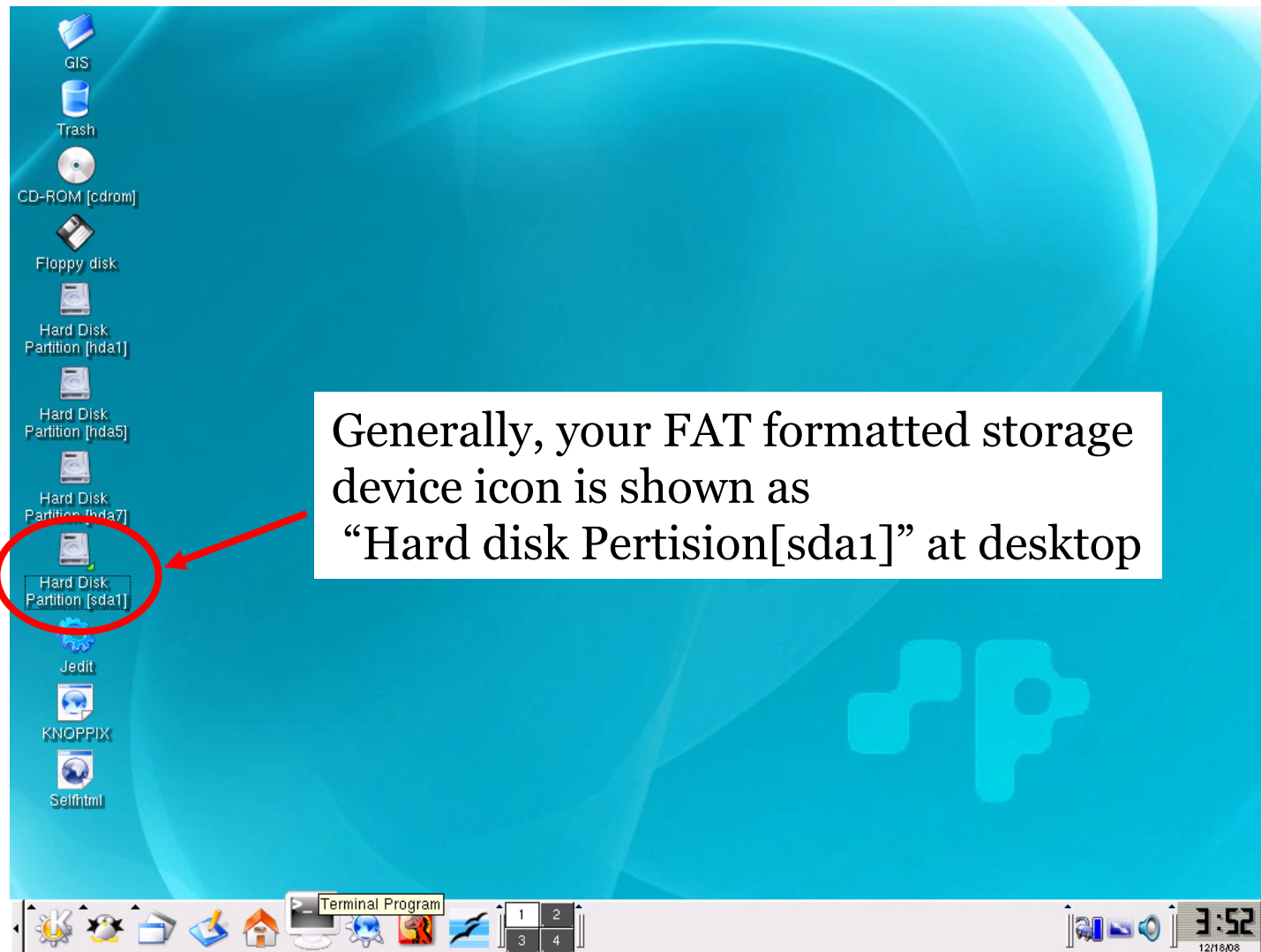


Step5 : Click <Apply >

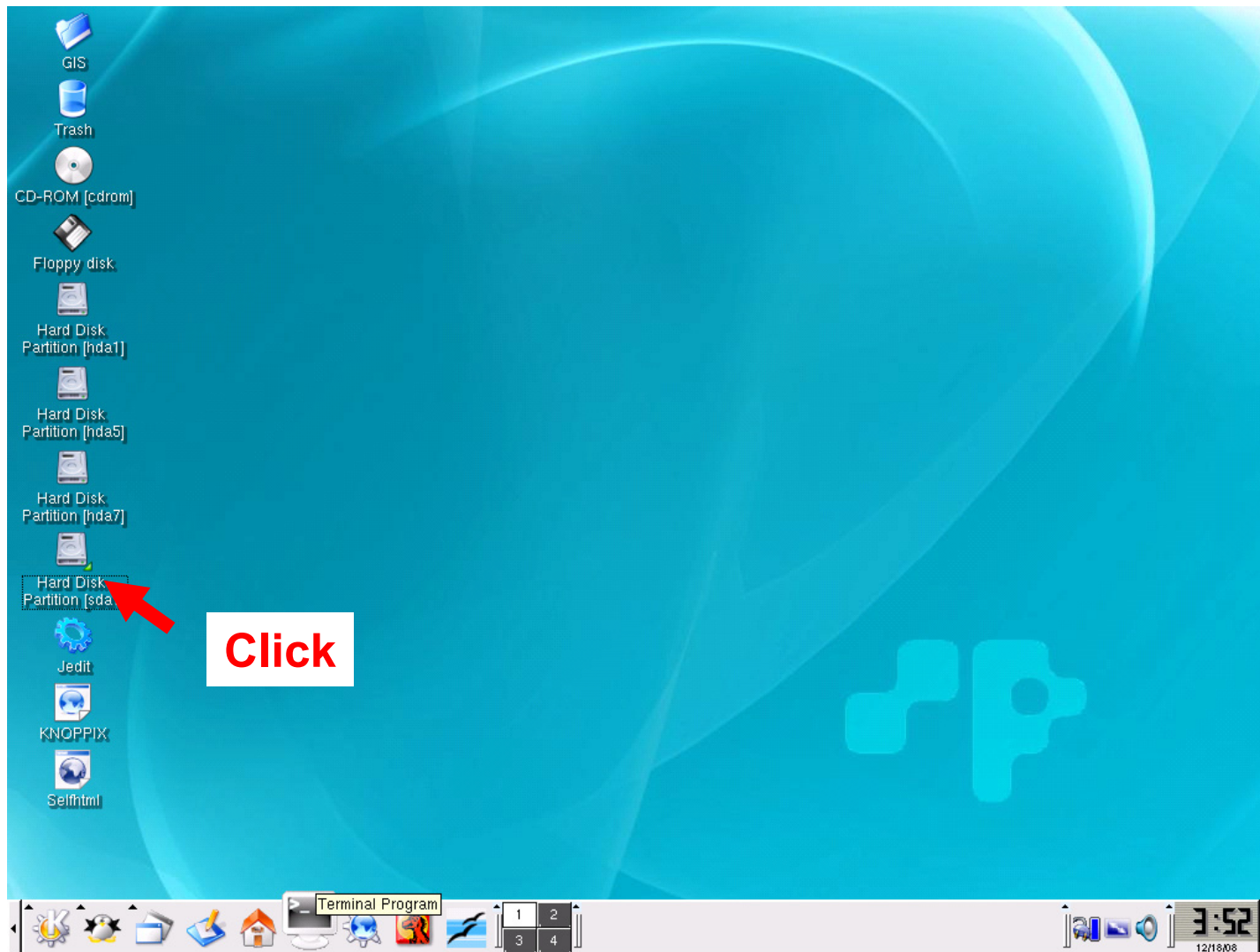
Click <Apply>

4.3 Make FAT formatted storage device writable

Step1 : Connect FAT formatted storage device to PC

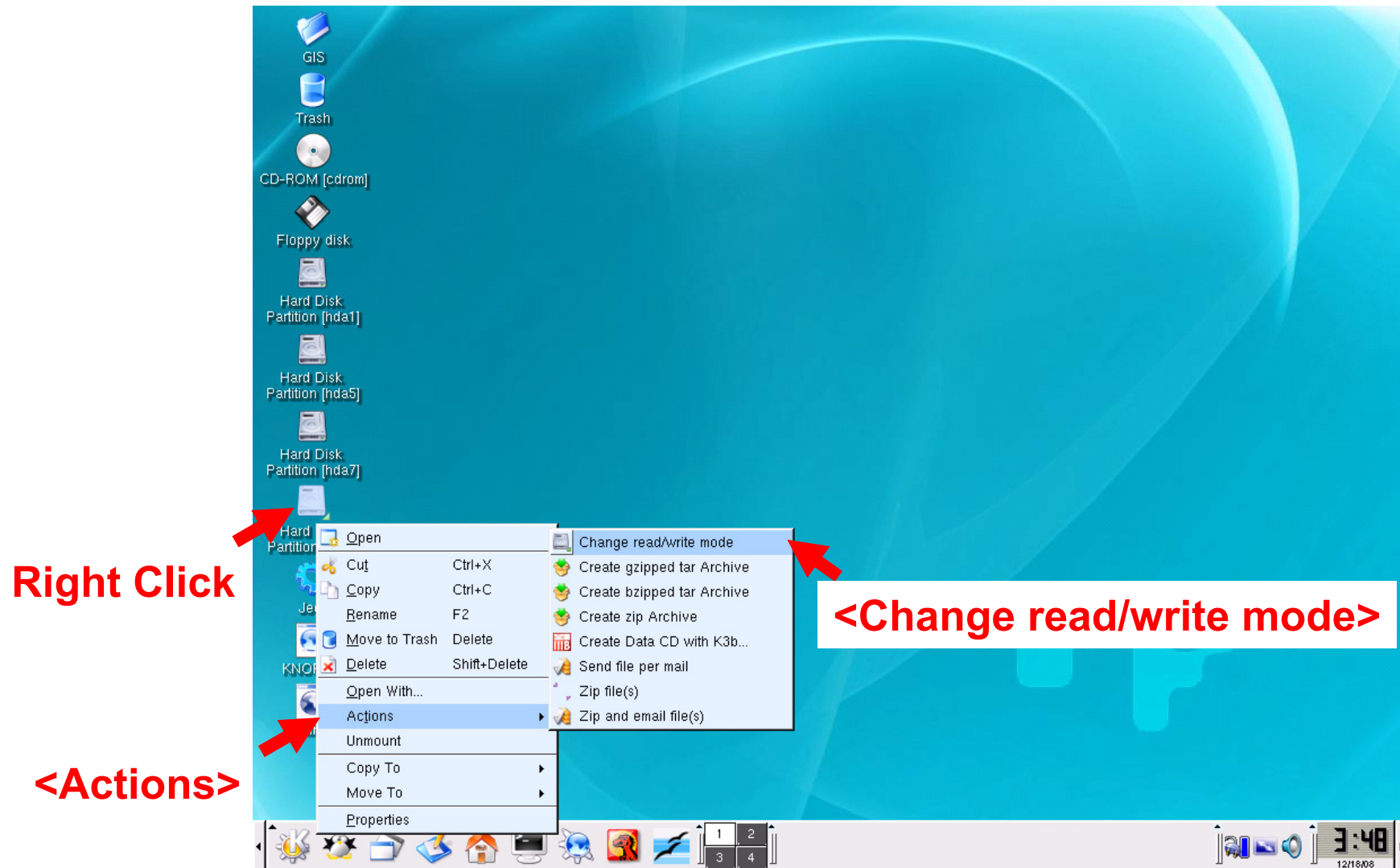


Step2 :Mount FAT formatted storage device [sda1] by clicking the [sda1] icon

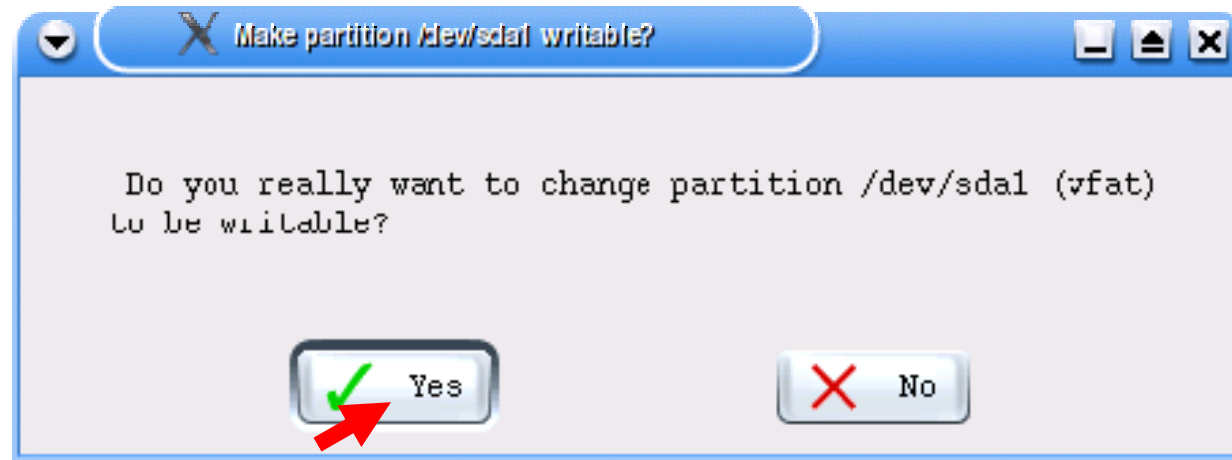


Step2 : Right click [sda1] icon on the desktop

→ <Actions> → <Change read/write mode>



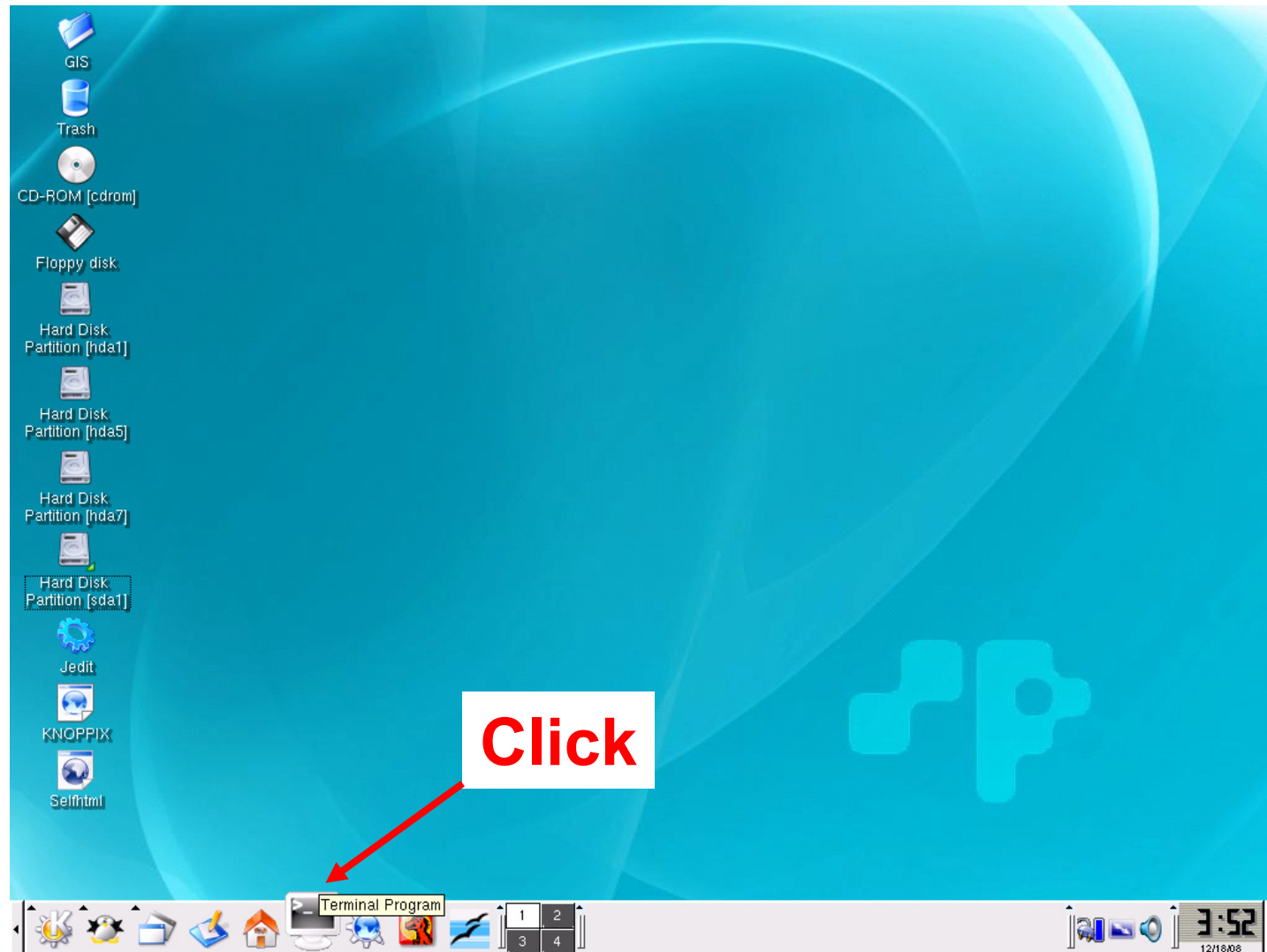
Step3 : Click <Yes >



Click

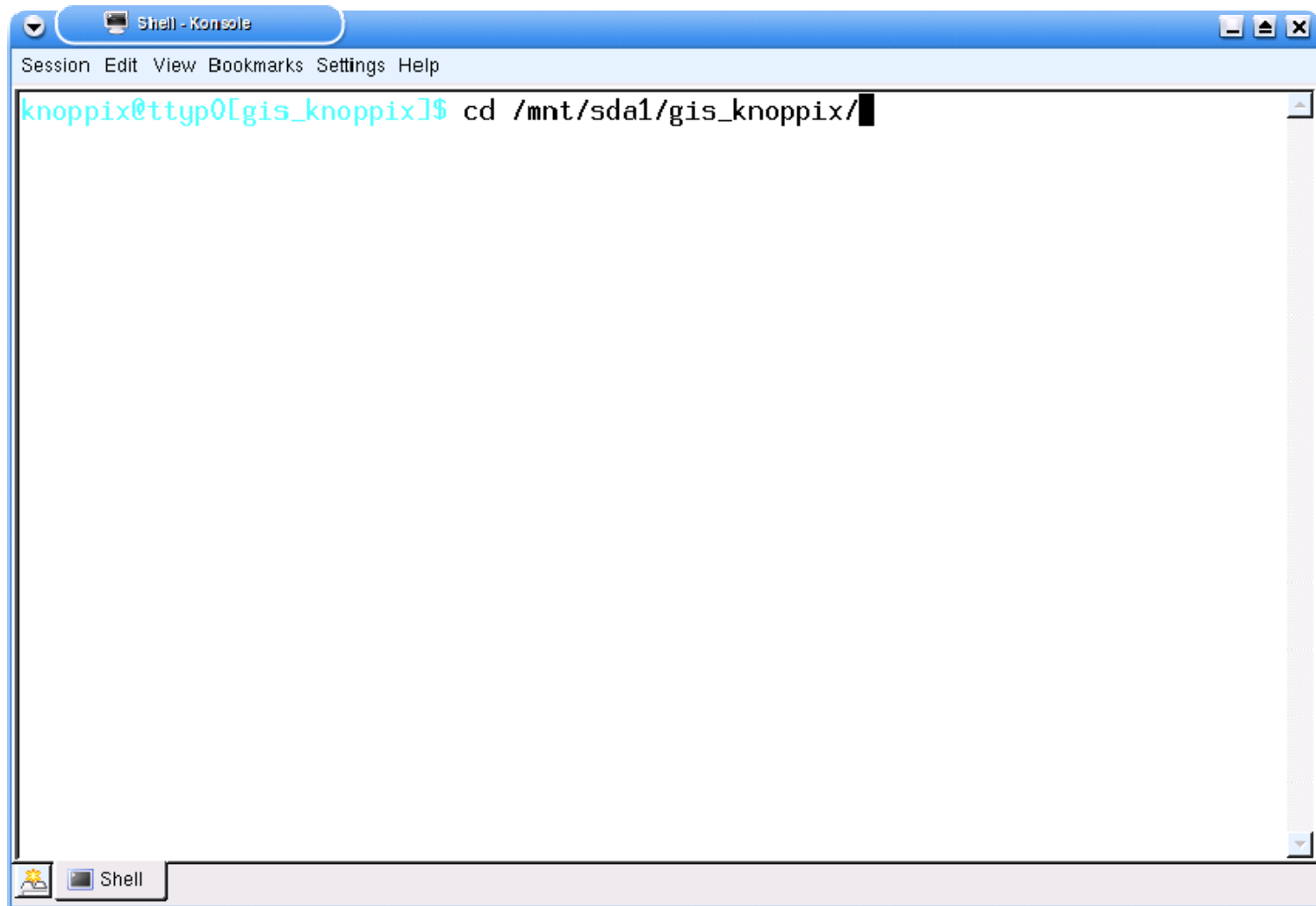
4.4a Make working directory

Step1 : Open “ Terminal Program ”



Step2 : On the Terminal window, write following command and hit <enter(return)> key

> cd /mnt/sda1/gis_knoppix : space



Step3 : Check directories in the “gis_knoppix” directory using following command and hit <enter(return)> key
> **ls**

```
knoppix@tty0[gis_knoppix]$ cd /mnt/sda1/gis_knoppix/
knoppix@tty0[gis_knoppix]$ ls
```

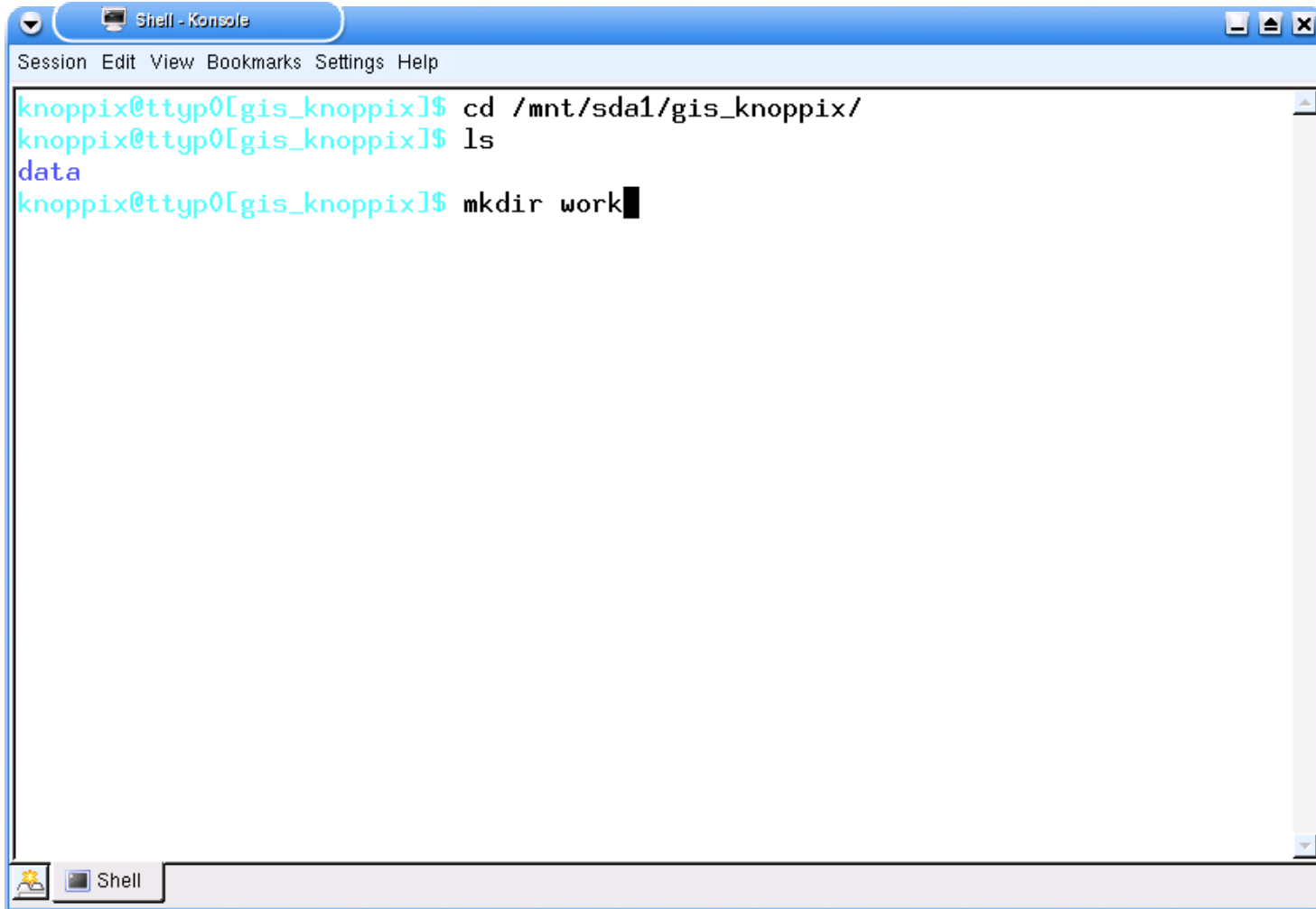
```
knoppix@tty0[gis_knoppix]$ cd /mnt/sda1/gis_knoppix/
knoppix@tty0[gis_knoppix]$ ls
data
knoppix@tty0[gis_knoppix]$
```

There is only “data” directory in the “gis_knoppix” directory

Step4 : Make “work” directory in the “gis_knoppix” directory using following command and hit <enter(return)> key

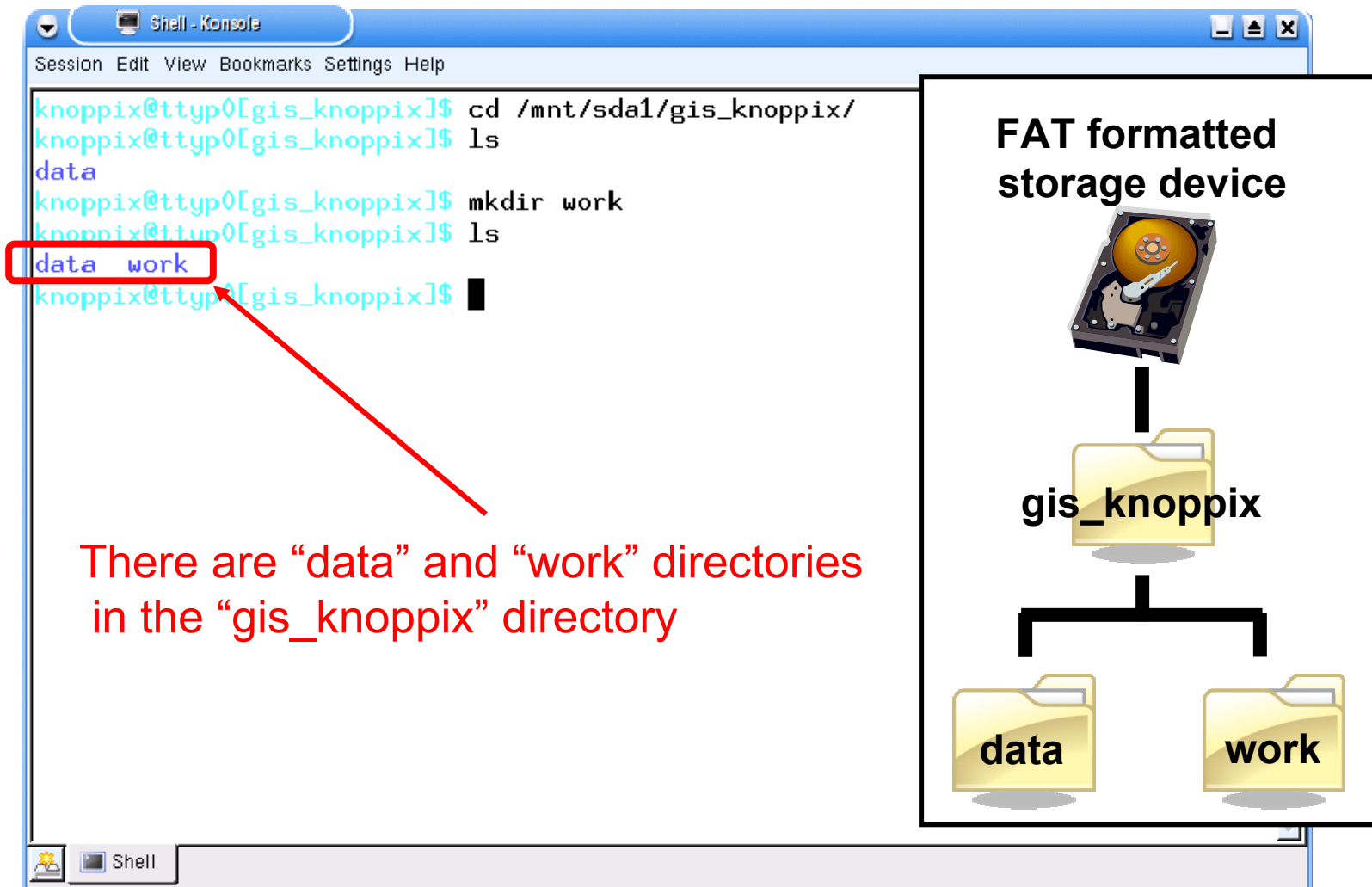
> mkdir work

: space



```
knoppix@tty0[gis_knoppix]$ cd /mnt/sda1/gis_knoppix/
knoppix@tty0[gis_knoppix]$ ls
data
knoppix@tty0[gis_knoppix]$ mkdir work
```

Step5 : Check directories in the “gis_knoppix” directory using following command and hit <enter(return)> key
> ls



The image shows a terminal window titled "Shell - Konsole" with the following commands and output:

```
knoppix@tty0[gis_knoppix]$ cd /mnt/sda1/gis_knoppix/  
knoppix@tty0[gis_knoppix]$ ls  
data  
knoppix@tty0[gis_knoppix]$ mkdir work  
knoppix@tty0[gis_knoppix]$ ls  
data work  
knoppix@tty0[gis_knoppix]$
```

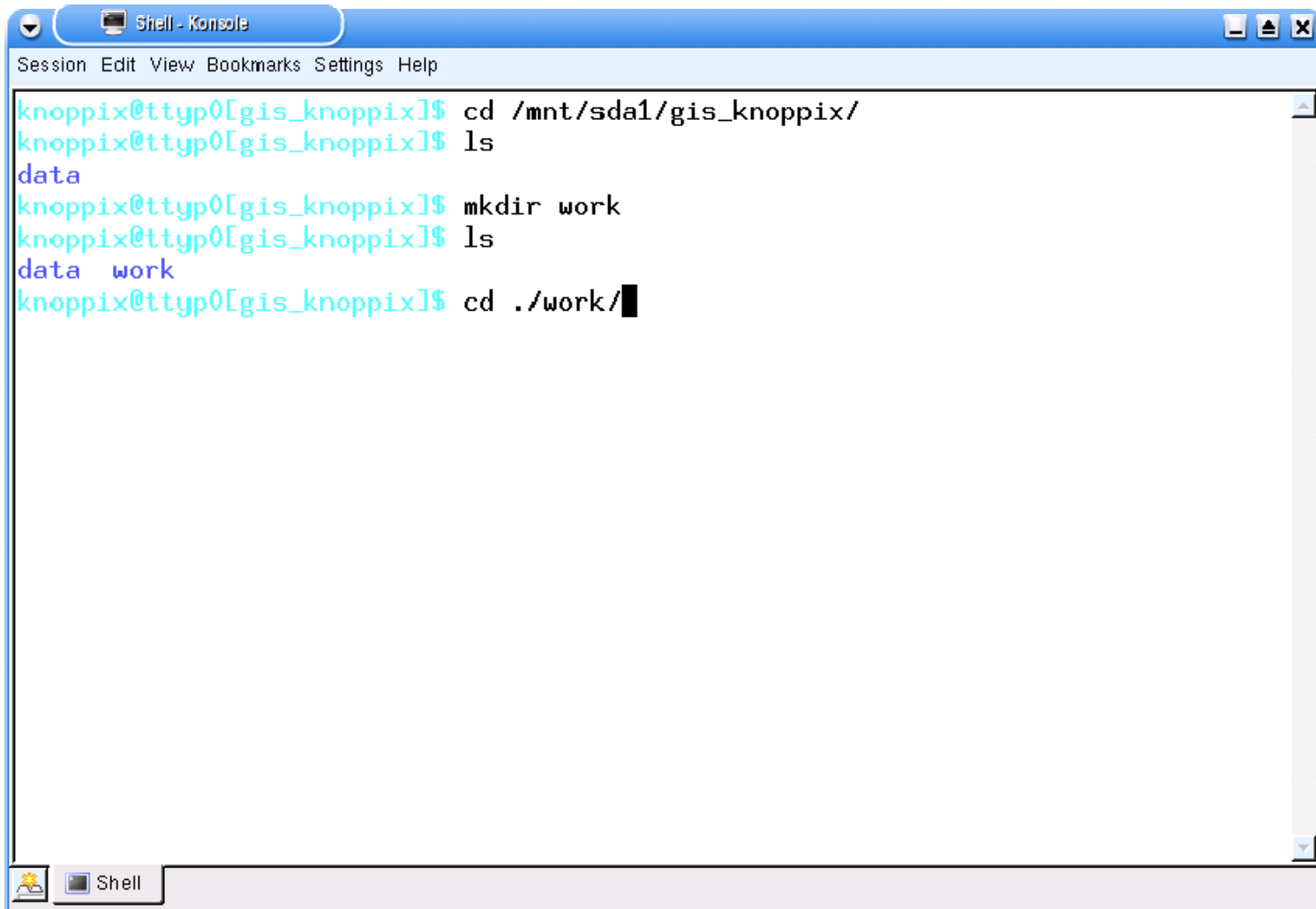
The output "data work" is highlighted with a red box. A red arrow points from this box to a text annotation: "There are “data” and “work” directories in the “gis_knoppix” directory".

To the right of the terminal window is a diagram titled "FAT formatted storage device". It shows a hard drive icon connected to a folder icon labeled "gis_knoppix". Below "gis_knoppix", there are two folder icons labeled "data" and "work", indicating the directory structure.

Step6 : Move to “work” directory using following command and hit <enter(return)> key

> cd ./work

 : space

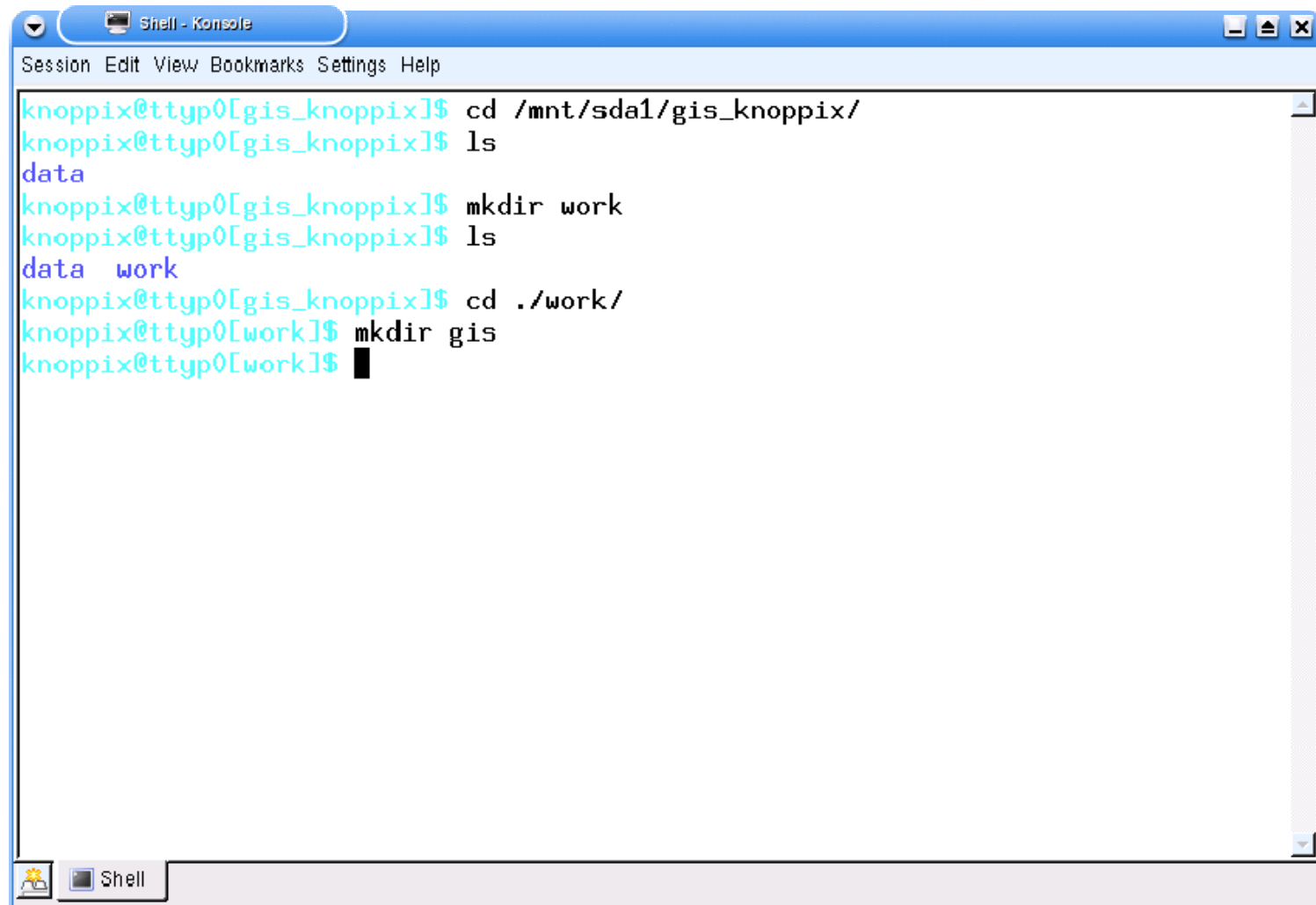


```
knoppix@tty0[gis_knoppix]$ cd /mnt/sda1/gis_knoppix/
knoppix@tty0[gis_knoppix]$ ls
data
knoppix@tty0[gis_knoppix]$ mkdir work
knoppix@tty0[gis_knoppix]$ ls
data work
knoppix@tty0[gis_knoppix]$ cd ./work/
```

Step7 : Make “gis” directory in the “work” directory using following command and hit <enter(return)> key

> mkdir gis

 : space



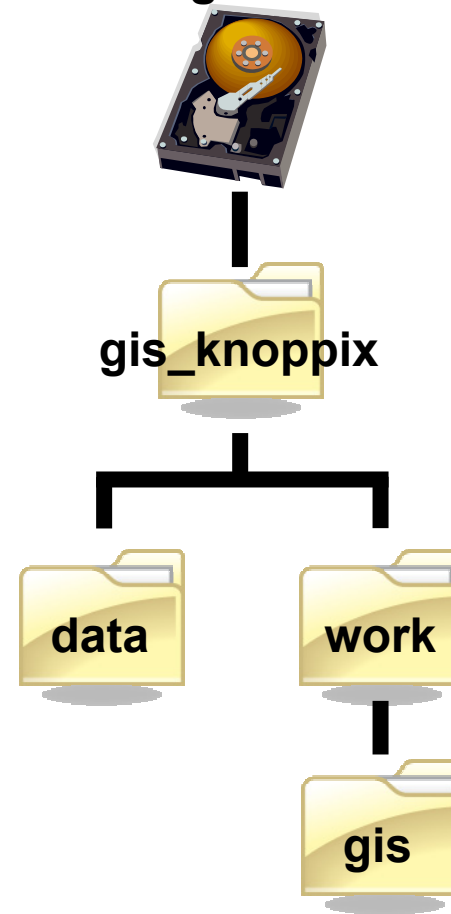
```
knoppix@tty0[gis_knoppix]$ cd /mnt/sda1/gis_knoppix/
knoppix@tty0[gis_knoppix]$ ls
data
knoppix@tty0[gis_knoppix]$ mkdir work
knoppix@tty0[gis_knoppix]$ ls
data  work
knoppix@tty0[gis_knoppix]$ cd ./work/
knoppix@tty0[work]$ mkdir gis
knoppix@tty0[work]$
```

Step8 : Check directories in the “work” directory using following command and hit <enter(return)> key
> ls

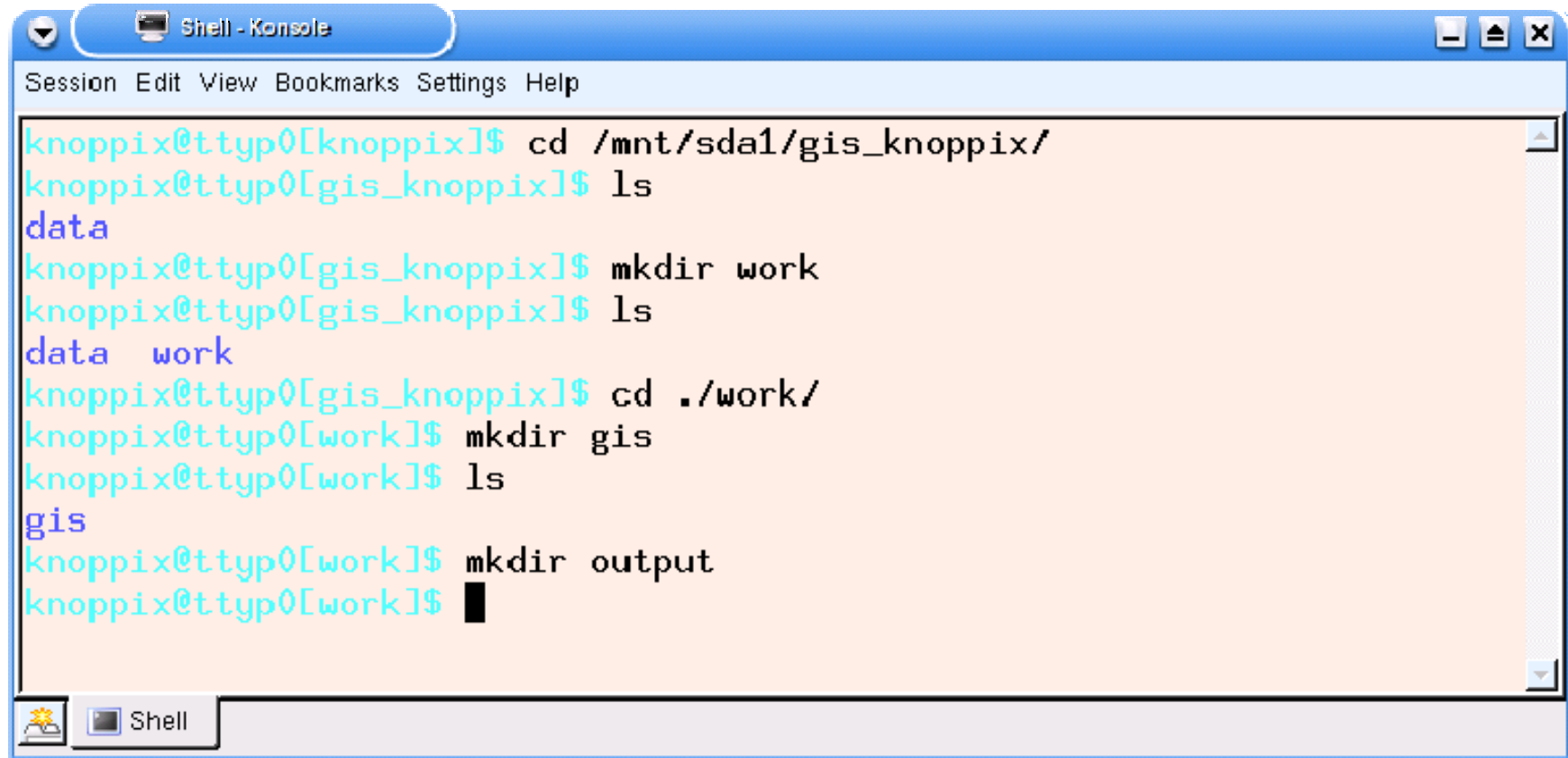
```
knoppix@tty0[gis_knoppix]$ cd /mnt/sda1/gis_knoppix/
knoppix@tty0[gis_knoppix]$ ls
data
knoppix@tty0[gis_knoppix]$ mkdir work
knoppix@tty0[gis_knoppix]$ ls
data work
knoppix@tty0[gis_knoppix]$ cd ./work/
knoppix@tty0[work]$ mkdir gis
knoppix@tty0[work]$ ls
gis
knoppix@tty0[work]$
```

There is “gis” directory in the “work” directory

**FAT formatted
storage device**

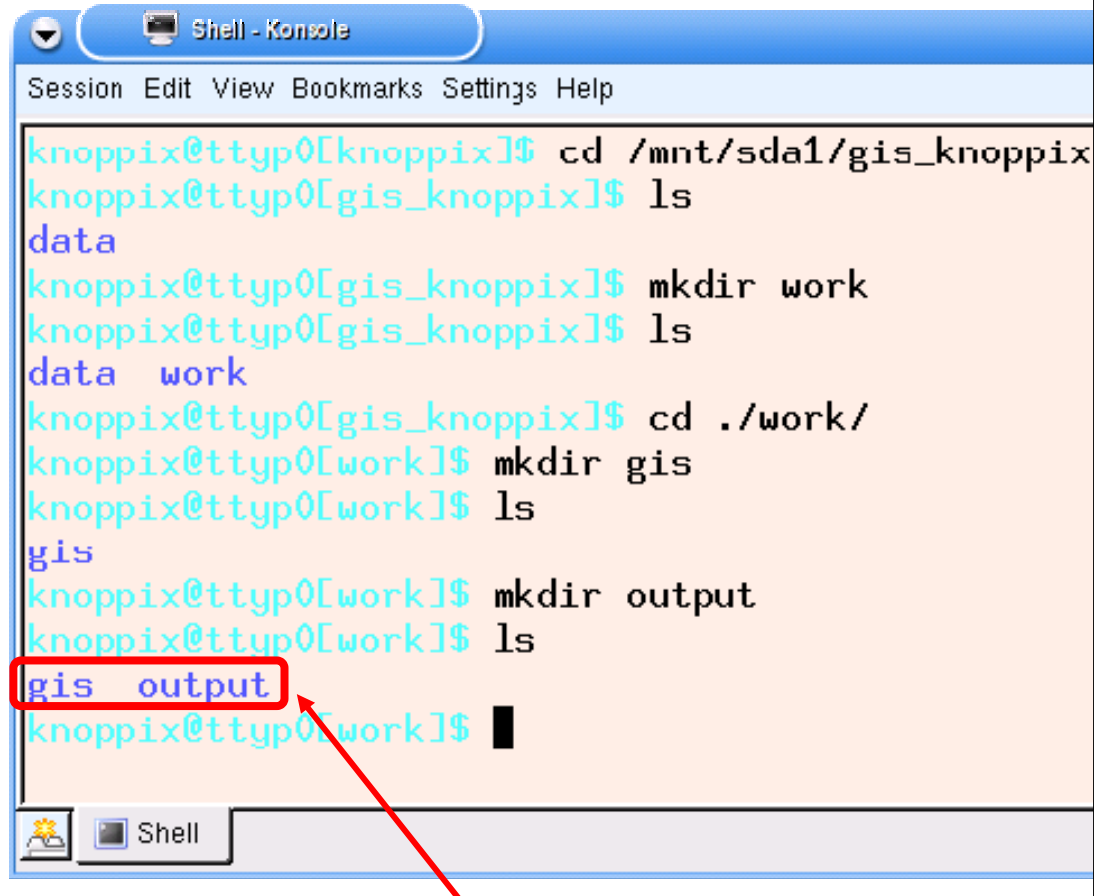


Step9 : Make “output” directory in the “work” directory
using following command and hit <enter(return)> key
> **mkdir** **output** : space



```
knoppix@tttyp0[knoppix]$ cd /mnt/sda1/gis_knoppix/
knoppix@tttyp0[gis_knoppix]$ ls
data
knoppix@tttyp0[gis_knoppix]$ mkdir work
knoppix@tttyp0[gis_knoppix]$ ls
data  work
knoppix@tttyp0[gis_knoppix]$ cd ./work/
knoppix@tttyp0[work]$ mkdir gis
knoppix@tttyp0[work]$ ls
gis
knoppix@tttyp0[work]$ mkdir output
knoppix@tttyp0[work]$
```

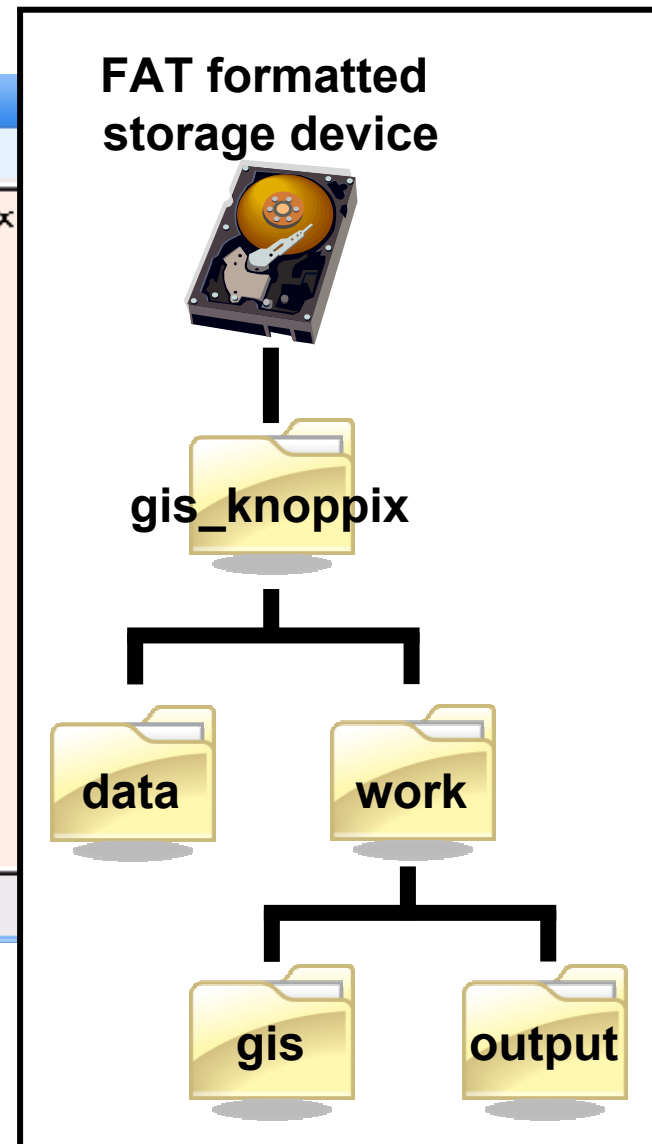
Step10 : Check directories in the “work” directory using following command and hit <enter(return)> key
> ls



```
knoppix@tty0[knoppix]$ cd /mnt/sda1/gis_knoppix
knoppix@tty0[gis_knoppix]$ ls
data
knoppix@tty0[gis_knoppix]$ mkdir work
knoppix@tty0[gis_knoppix]$ ls
data work
knoppix@tty0[gis_knoppix]$ cd ./work/
knoppix@tty0[work]$ mkdir gis
knoppix@tty0[work]$ ls
gis
knoppix@tty0[work]$ mkdir output
knoppix@tty0[work]$ ls
gis output
knoppix@tty0[work]$
```

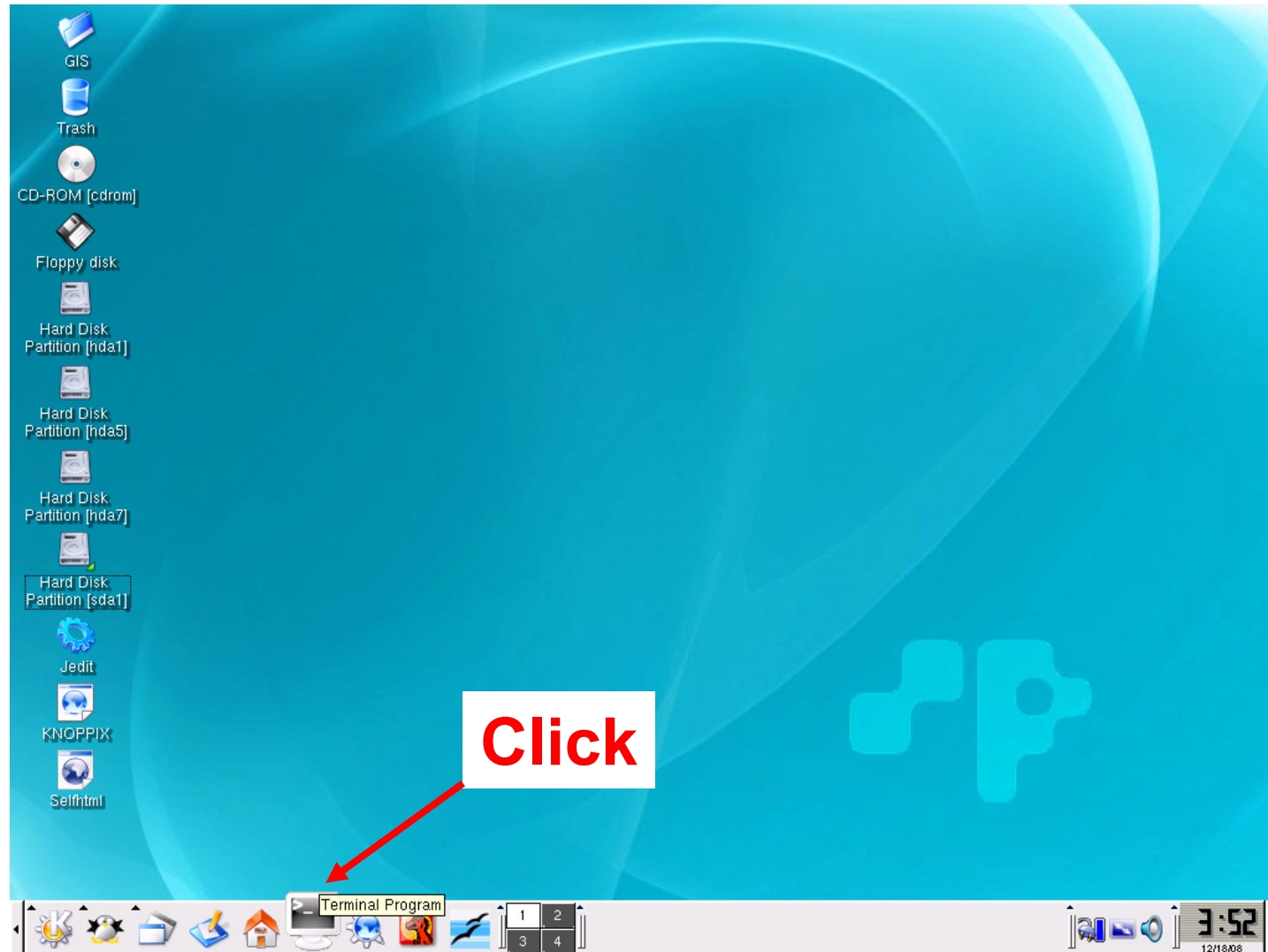
The screenshot shows a terminal window titled "Shell - Konsole". The user navigates to the directory /mnt/sda1/gis_knoppix and lists its contents, showing a "data" directory. Then, a "work" directory is created. The user enters the "work" directory and creates a "gis" directory. Finally, an "output" directory is created, and the "ls" command is used to list the contents of the "work" directory, which shows "gis" and "output". A red box highlights the output "gis output", and a red arrow points from this box to the text below.

There are “gis” and “output” directories in the “work” directory

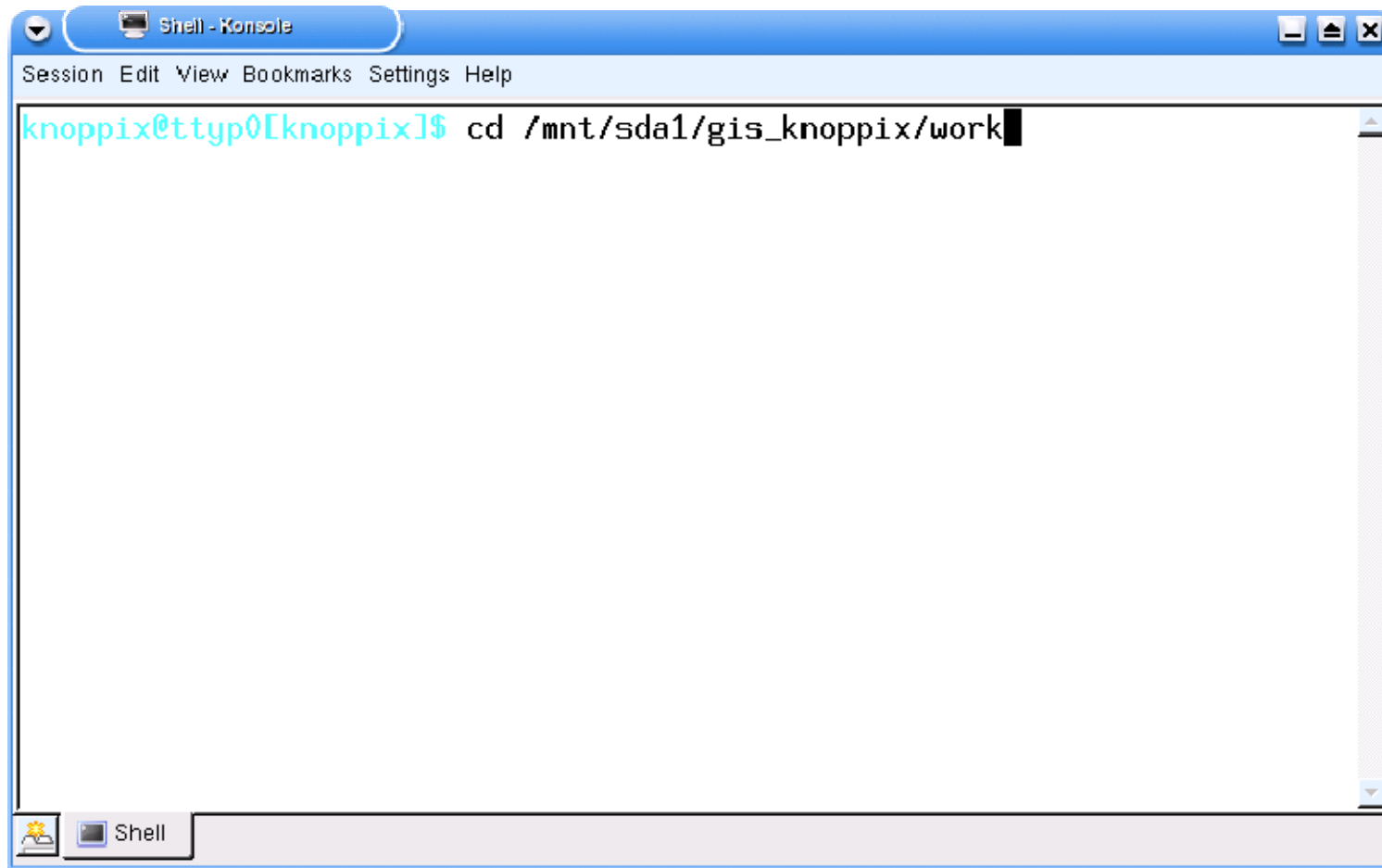


4.4b Move to “work” directory

Step1 : Open “Terminal Program”



Step2 : On the Terminal window, write following command and hit <enter(return)> key
> **cd /mnt/sda1/gis_knoppix/work** : space



Section 2. Data Processing in Latitude-Longitude Coordinate System

1. Starting and Terminating GRASS

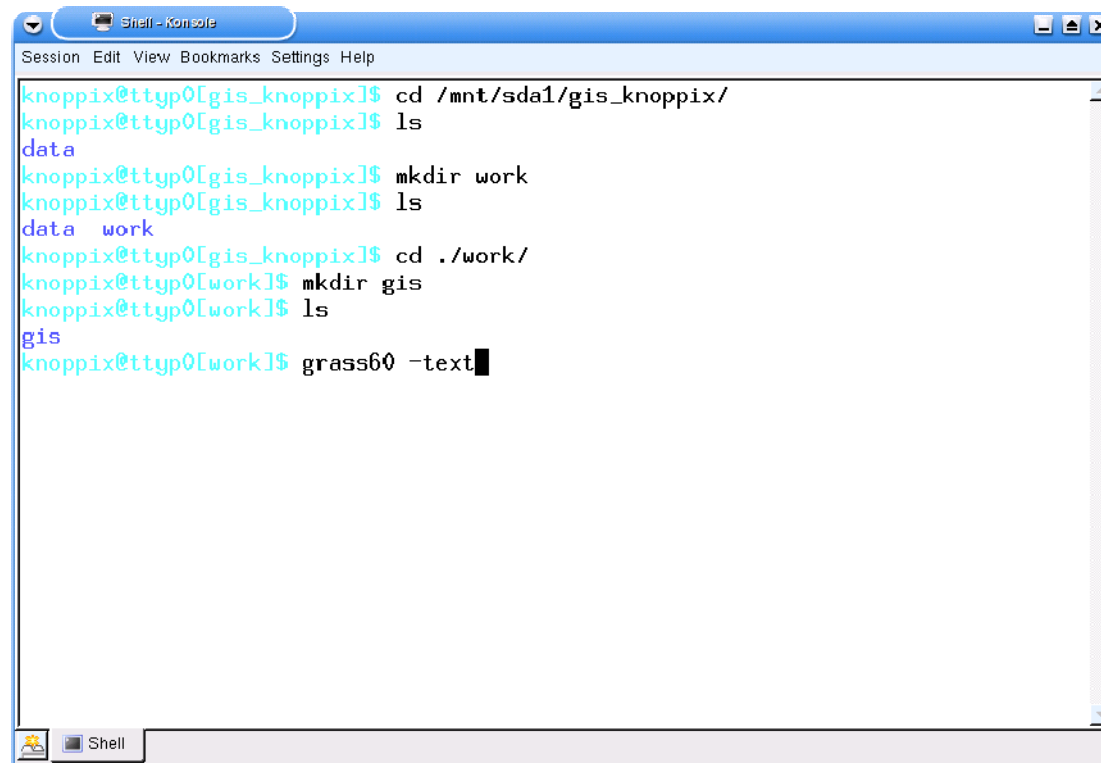
1.1 Starting GRASS

> grass60 -text

: space

This program is used to start GRASS. It will parse the command line arguments and then initialize GRASS for the user.

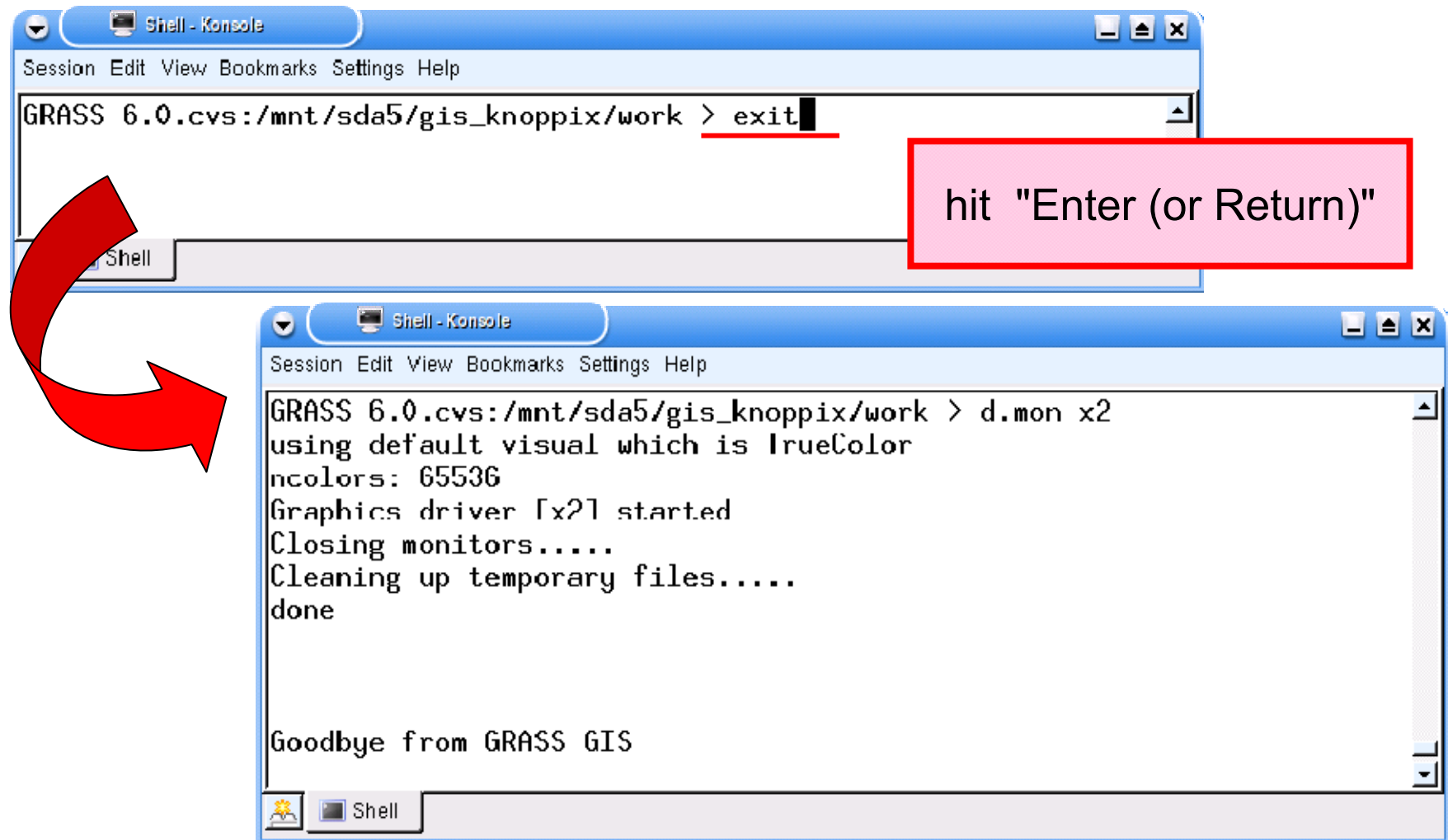
-text : Indicates that the text based user interface should be used



```
knoppix@tty0[gis_knoppix]$ cd /mnt/sda1/gis_knoppix/
knoppix@tty0[gis_knoppix]$ ls
data
knoppix@tty0[gis_knoppix]$ mkdir work
knoppix@tty0[gis_knoppix]$ ls
data  work
knoppix@tty0[gis_knoppix]$ cd ./work/
knoppix@tty0[work]$ mkdir gis
knoppix@tty0[work]$ ls
gis
knoppix@tty0[work]$ grass60 -text
```

1.2 Terminate GRASS

> exit



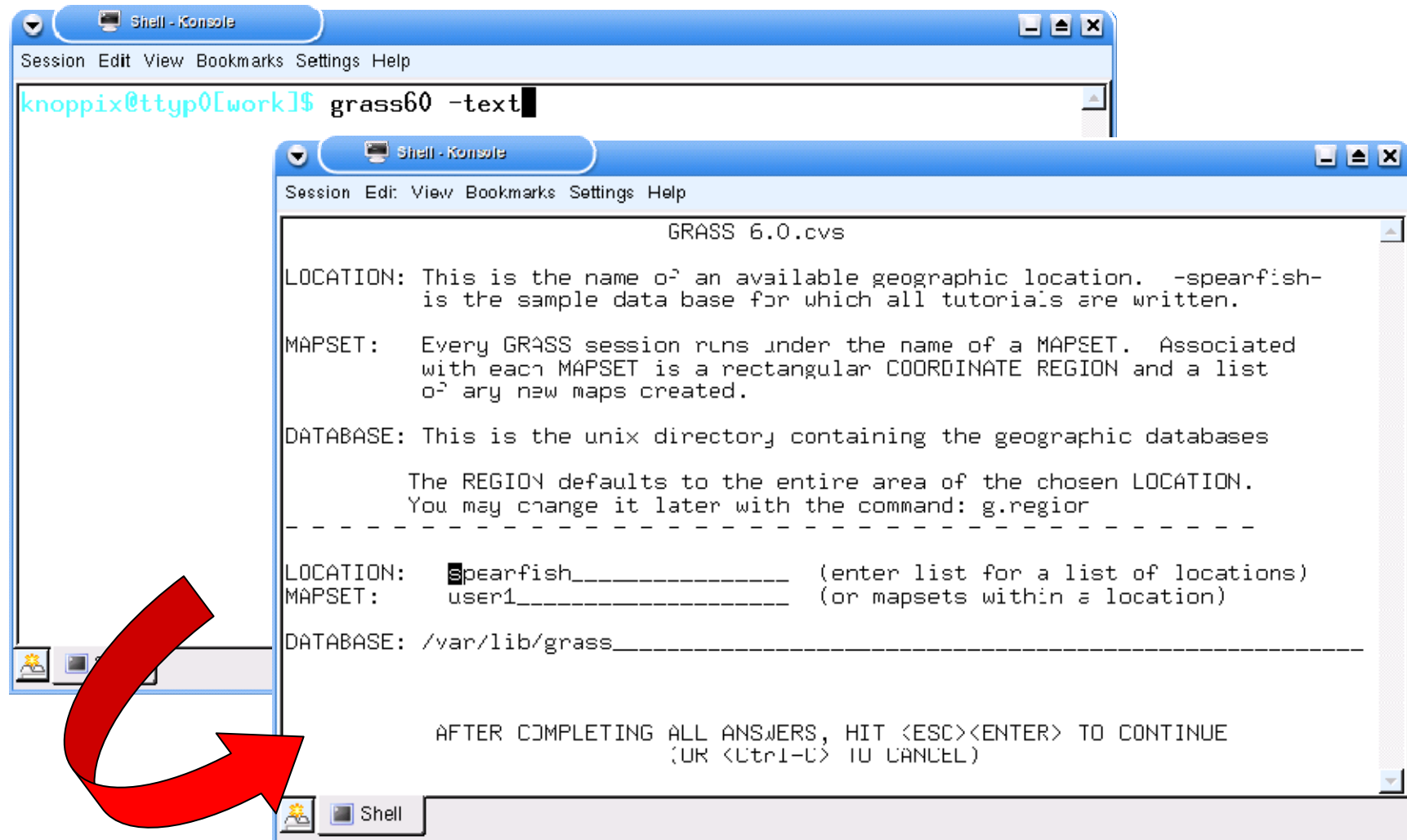
Section 2. Data Processing in Latitude-Longitude Coordinate System

2. Defining coordinate system and region

2.1 Starting GRASS

> grass60 -text

: space



```
knoppix@tty0[work]$ grass60 -text
```

```
GRASS 6.0.cvs
```

```
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: spearfish_____ (enter list for a list of locations)
MAPSET: user1_____ (or mapsets within a location)
DATABASE: /var/lib/grass_____
```

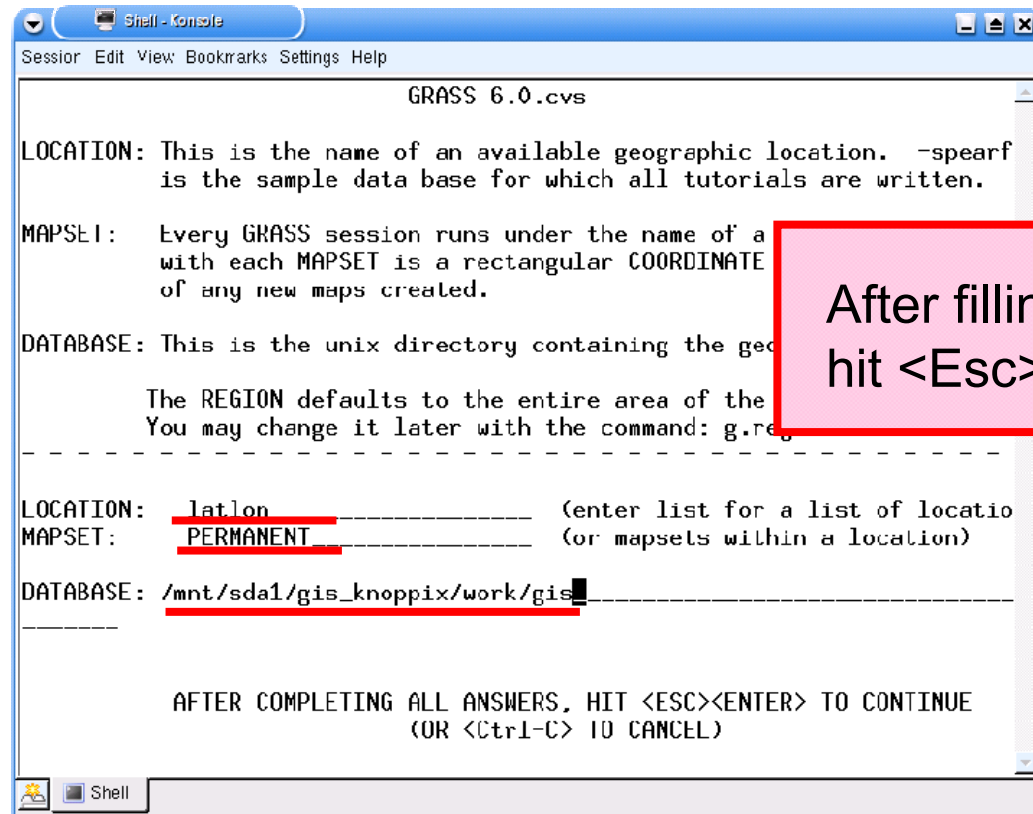
```
-----
AFTER COMPLETING ALL ANSWERS, HIT <ESC><ENTER> TO CONTINUE
(OR <Ctrl-C> TO CANCEL)
```

2.2 Determination of LOCATION, MAPSET, DATABASE

LOCATION: **latlon**

MAPSET: **PERMANENT**

DATABASE: **/mnt/sda1/gis_knoppix/work/gis**



The screenshot shows a terminal window titled "Shell - Konsole" with a menu bar (Session, Edit, View, Bookmarks, Settings, Help). The main text reads "GRASS 6.0.cvs". Below this, there are three sections of text explaining the configuration options: LOCATION, MAPSET, and DATABASE. Each section is followed by a dashed line and a prompt for user input. The input for each prompt is highlighted with a red box. At the bottom, there is a message: "AFTER COMPLETING ALL ANSWERS, HIT <ESC><ENTER> TO CONTINUE (OR <Ctrl-C> TO CANCEL)".

```
GRASS 6.0.cvs

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

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

DATABASE: This is the unix directory containing the geo

        The REGION defaults to the entire area of the
        You may change it later with the command: g.re

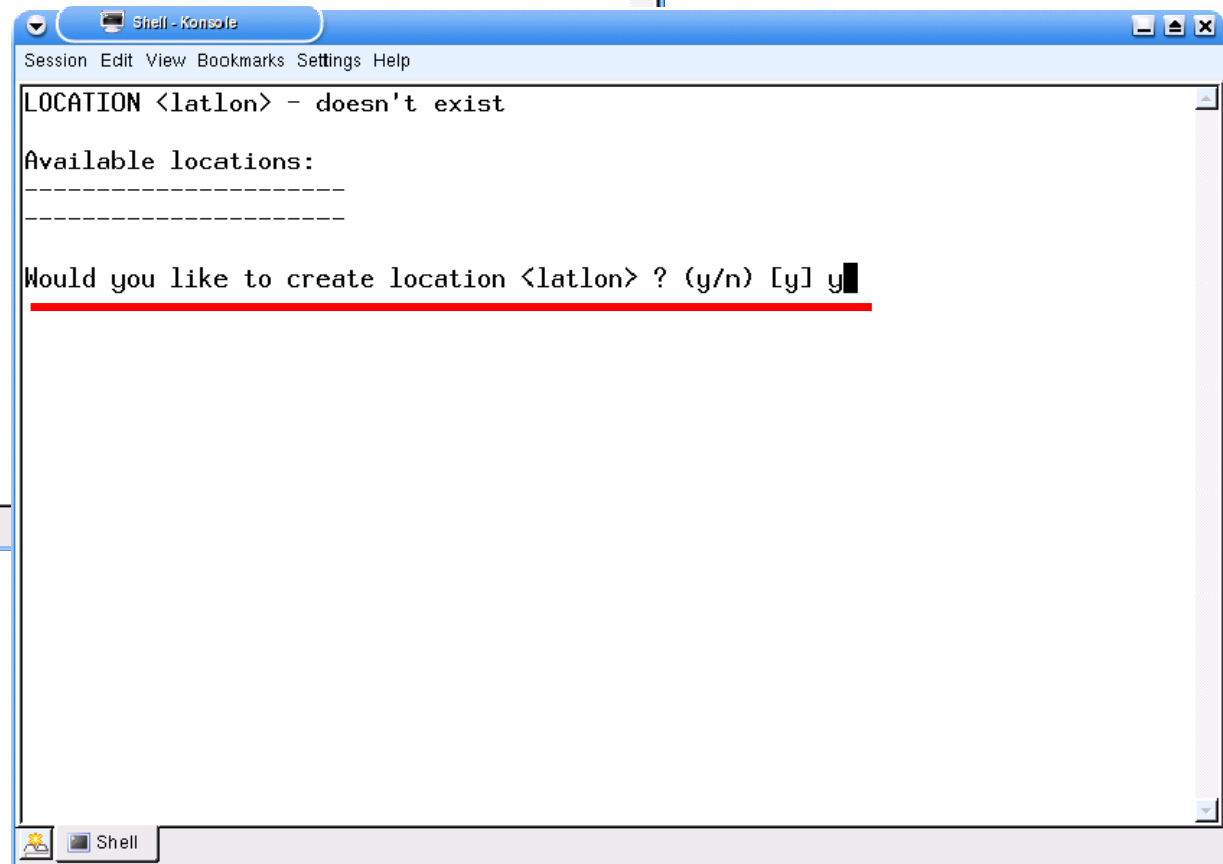
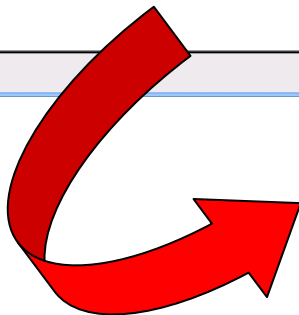
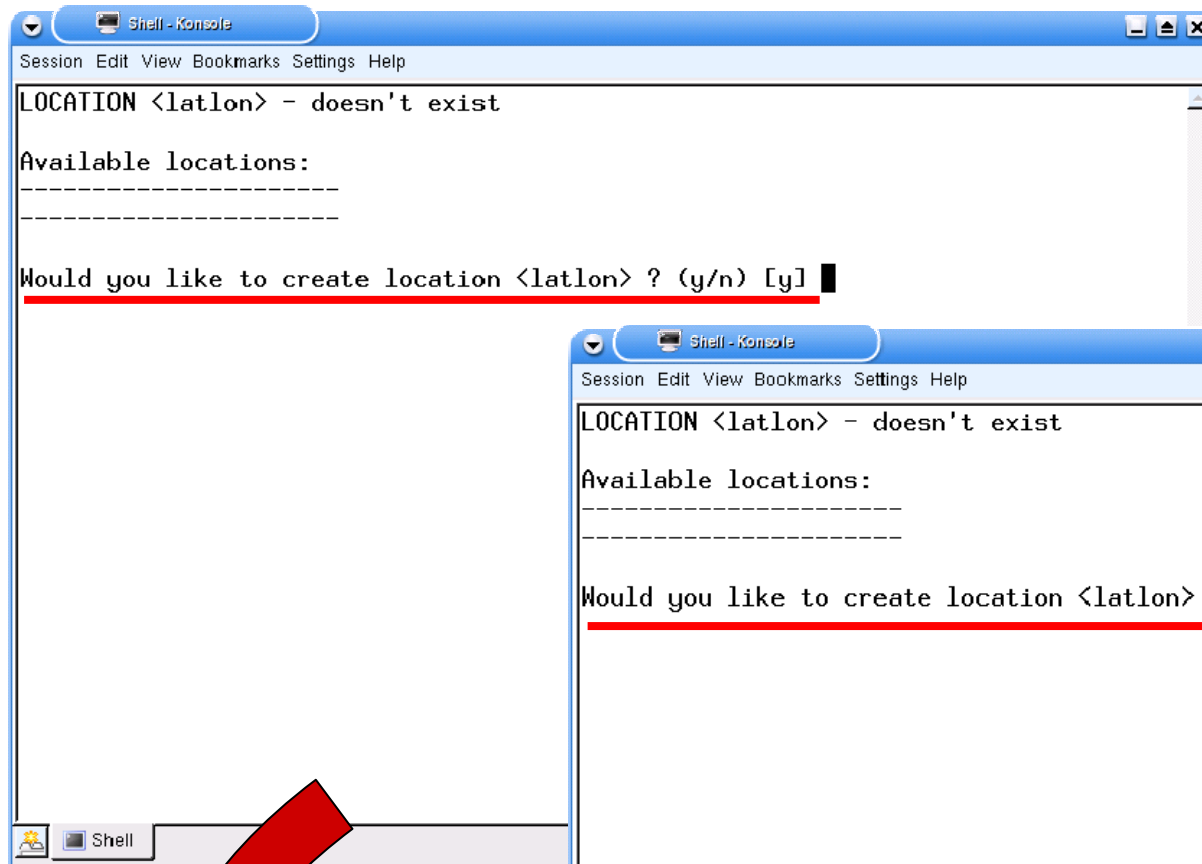
-----
LOCATION:  latlon _____ (enter list for a list of locatio
MAPSET:   PERMANENT _____ (or mapsets within a location)
DATABASE: /mnt/sda1/gis_knoppix/work/gis _____

-----

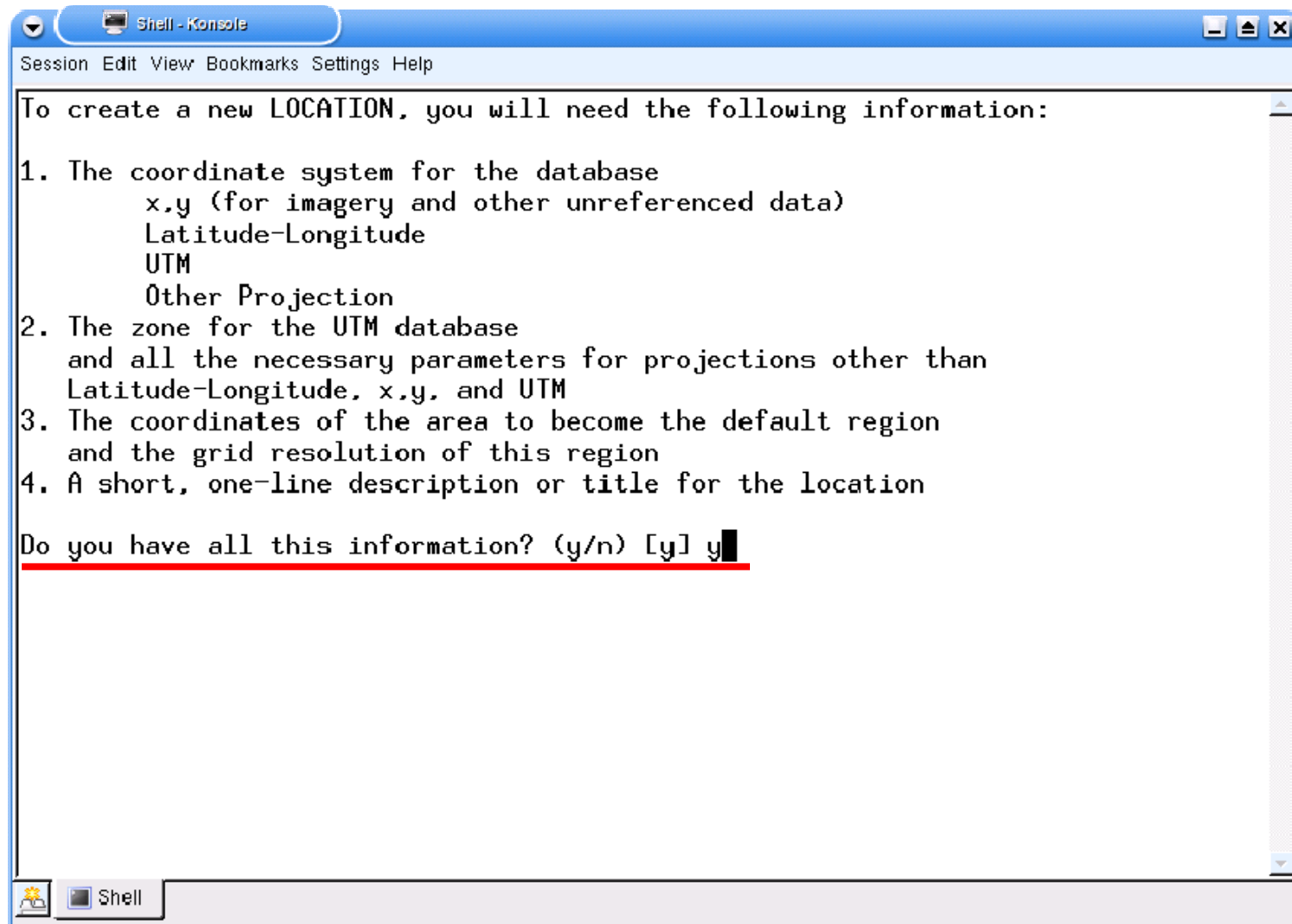
AFTER COMPLETING ALL ANSWERS, HIT <ESC><ENTER> TO CONTINUE
        (OR <Ctrl-C> TO CANCEL)
```

After filling the above information,
hit <Esc> + <Enter (or Return) >

enter < y >, then hit < Enter (or Return) >



enter < y >, then hit < Enter (or Return) >



```
Shell - Konsole
Session Edit View Bookmarks Settings Help

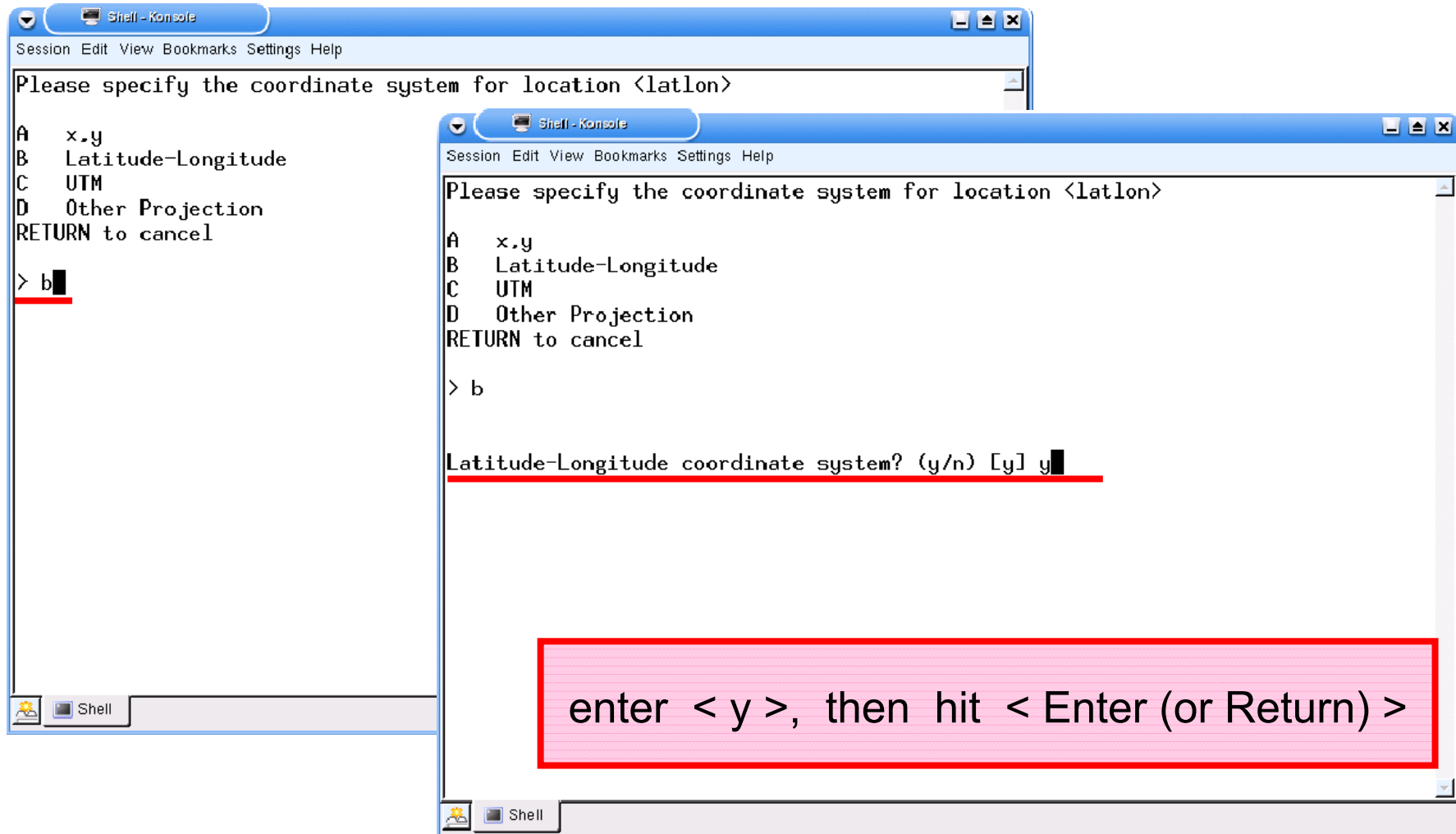
To create a new LOCATION, you will need the following information:

1. The coordinate system for the database
    x,y (for imagery and other unreferenced data)
    Latitude-Longitude
    UTM
    Other Projection
2. The zone for the UTM database
    and all the necessary parameters for projections other than
    Latitude-Longitude, x,y, and UTM
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

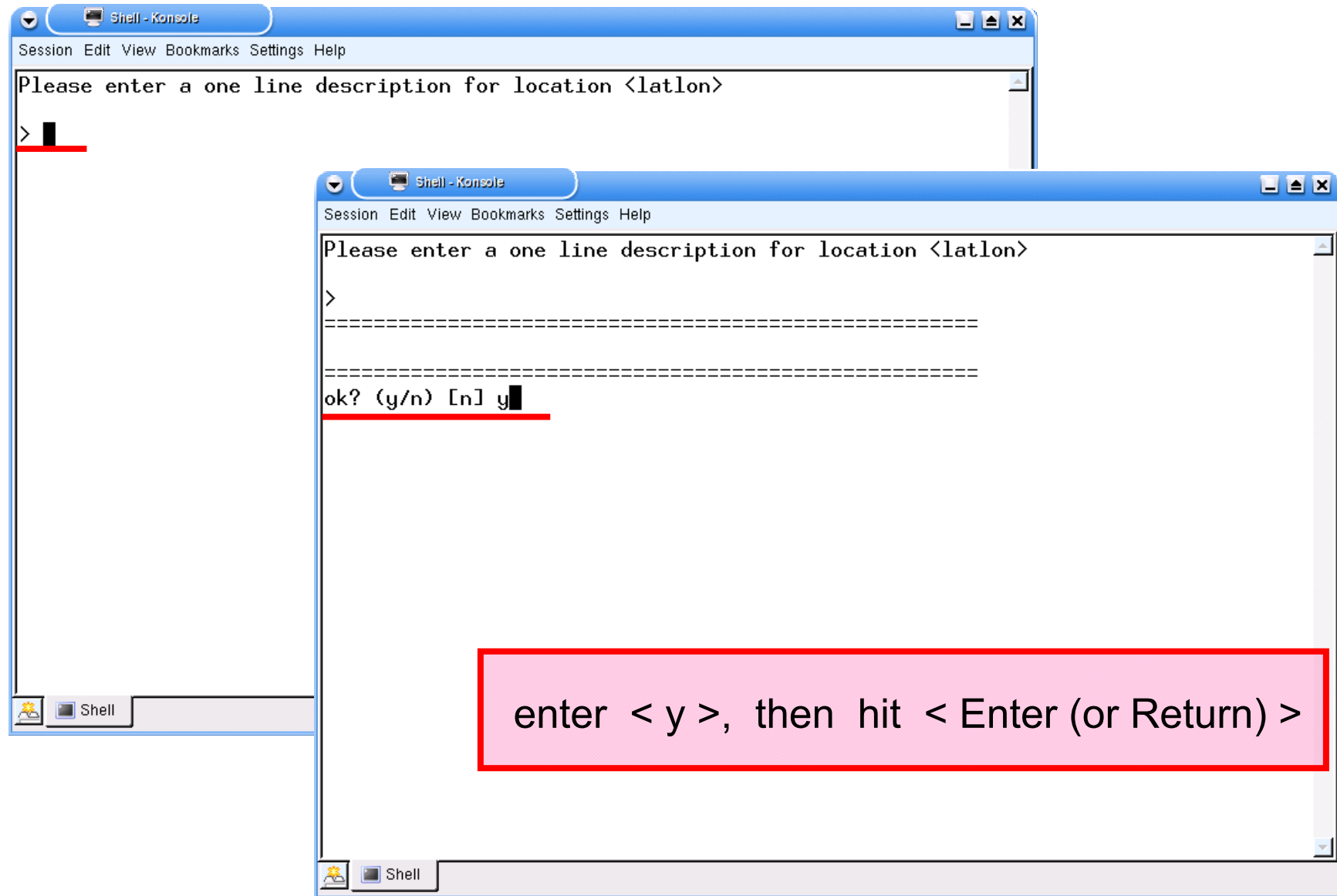
Do you have all this information? (y/n) [y] y
```

2.3 Selection of projection : Latitude-longitude coordinate

enter < b >, then hit < Enter (or Return) >

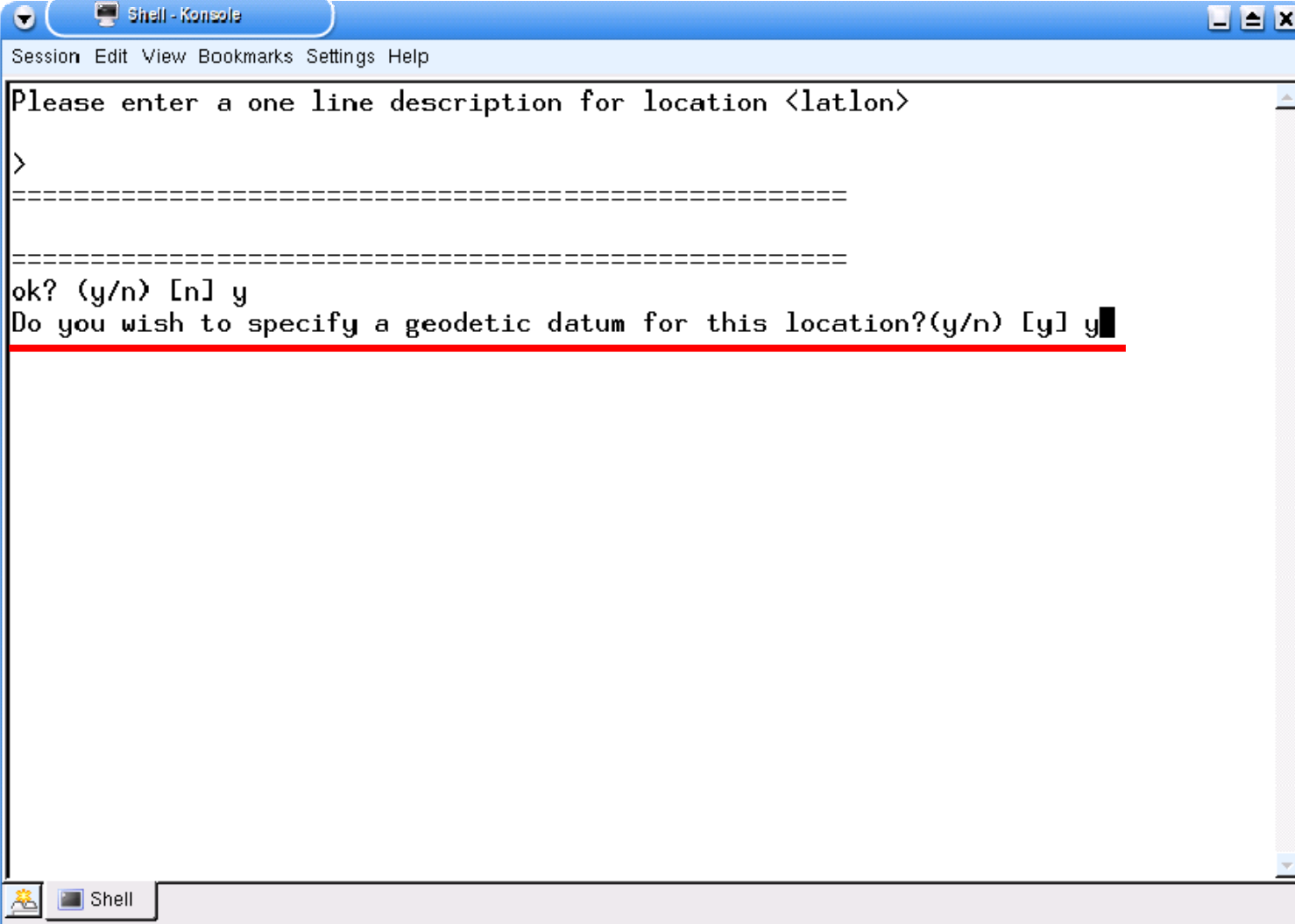


just hit < Enter (or Return) >



enter < y >, then hit < Enter (or Return) >

enter " y ", then hit " Enter (or Return) "

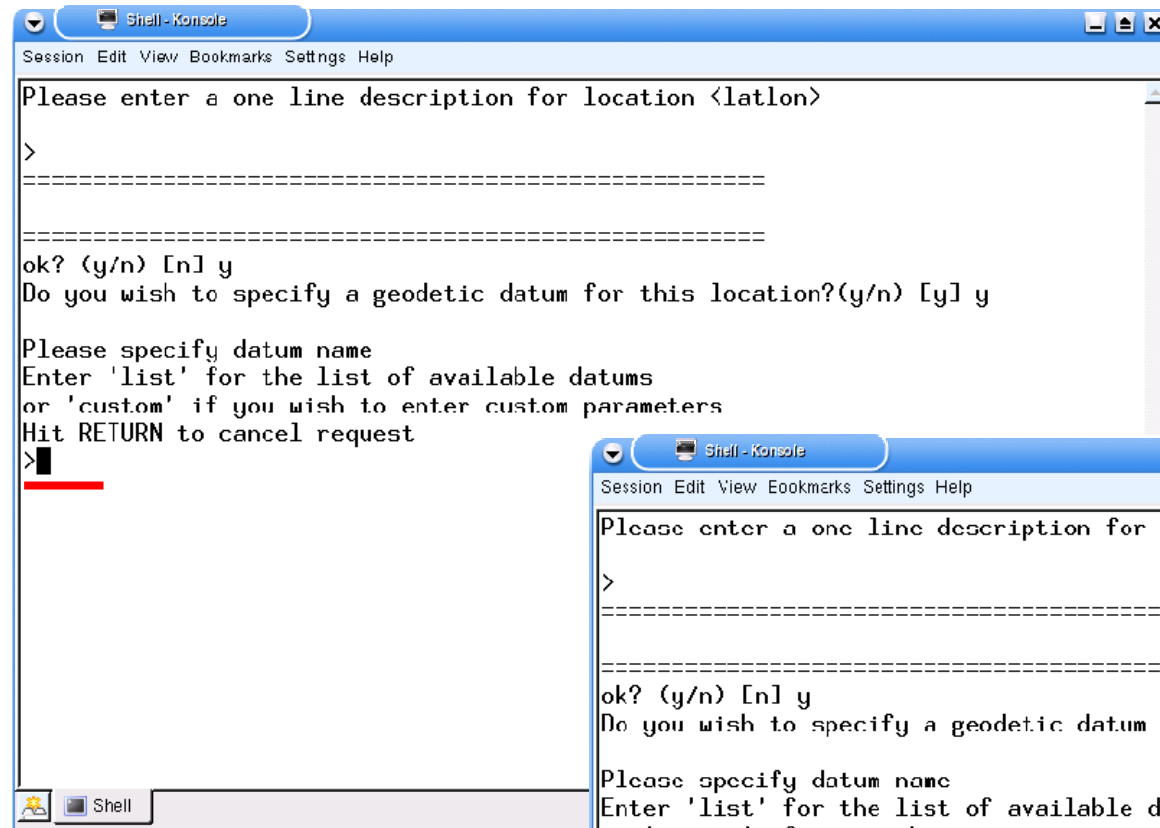


```
Shell - Konsole
Session Edit View Bookmarks Settings Help

Please enter a one line description for location <latlon>
>
=====
=====
ok? (y/n) [n] y
Do you wish to specify a geodetic datum for this location?(y/n) [y] y

```

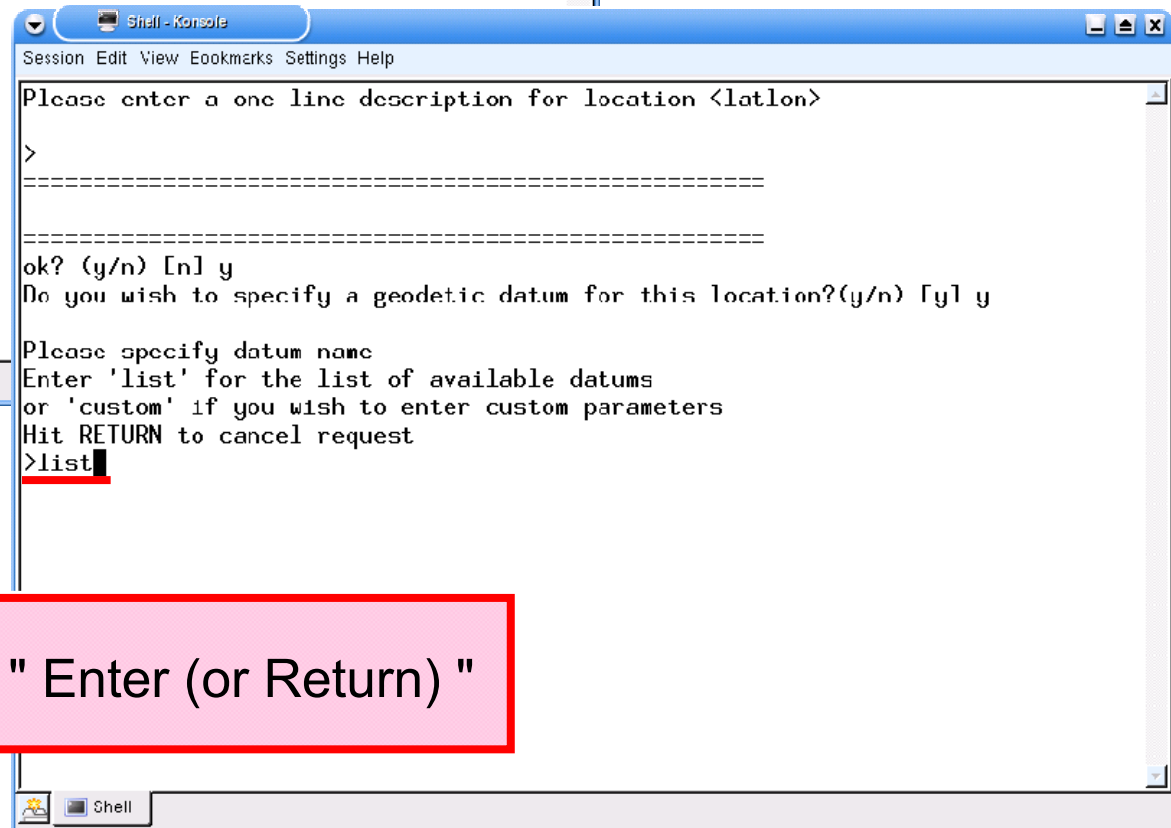
2.4 Selection of datum



```
Shell - Konsole
Session Edit View Bookmarks Settings Help

Please enter a one line description for location <latlon>
>
=====
=====
ok? (y/n) [n] y
Do you wish to specify a geodetic datum for this location?(y/n) [y] y

Please specify datum name
Enter 'list' for the list of available datums
or 'custom' if you wish to enter custom parameters
Hit RETURN to cancel request
>
```



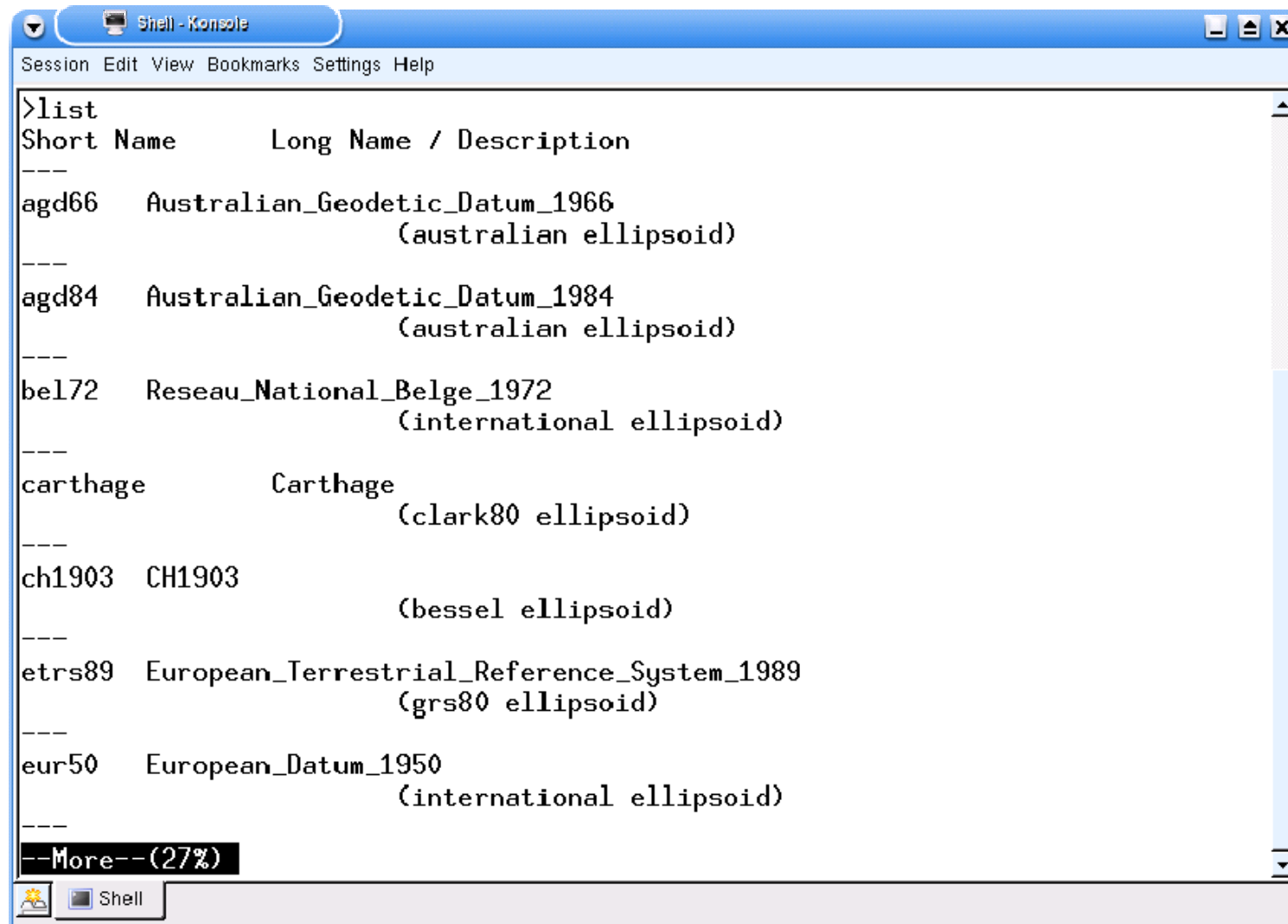
```
Shell - Konsole
Session Edit View Bookmarks Settings Help

Please enter a one line description for location <latlon>
>
=====
=====
ok? (y/n) [n] y
Do you wish to specify a geodetic datum for this location?(y/n) [y] y

Please specify datum name
Enter 'list' for the list of available datums
or 'custom' if you wish to enter custom parameters
Hit RETURN to cancel request
>list
```

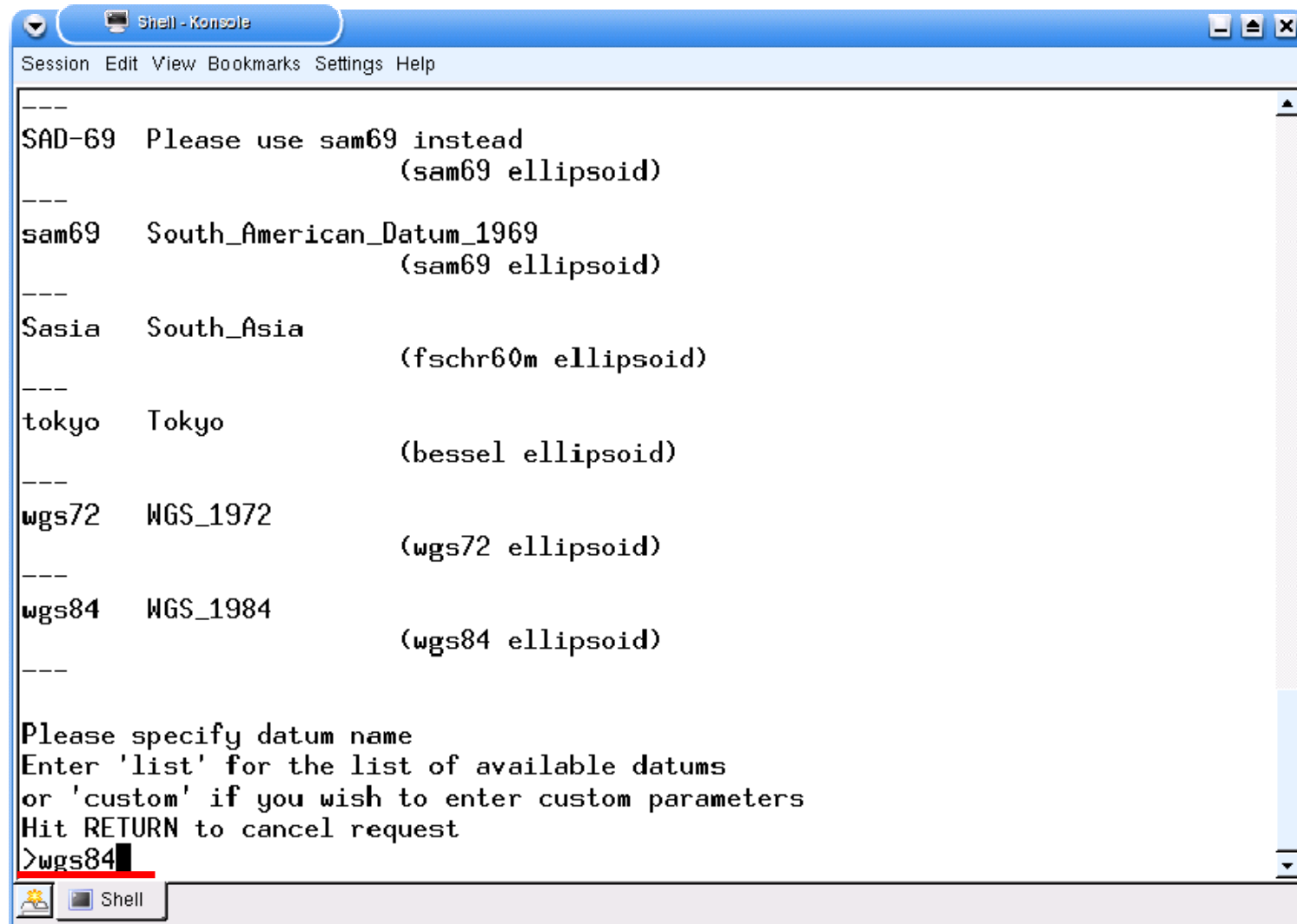
enter "list", then hit "Enter (or Return)"

hit "Space" key for page down



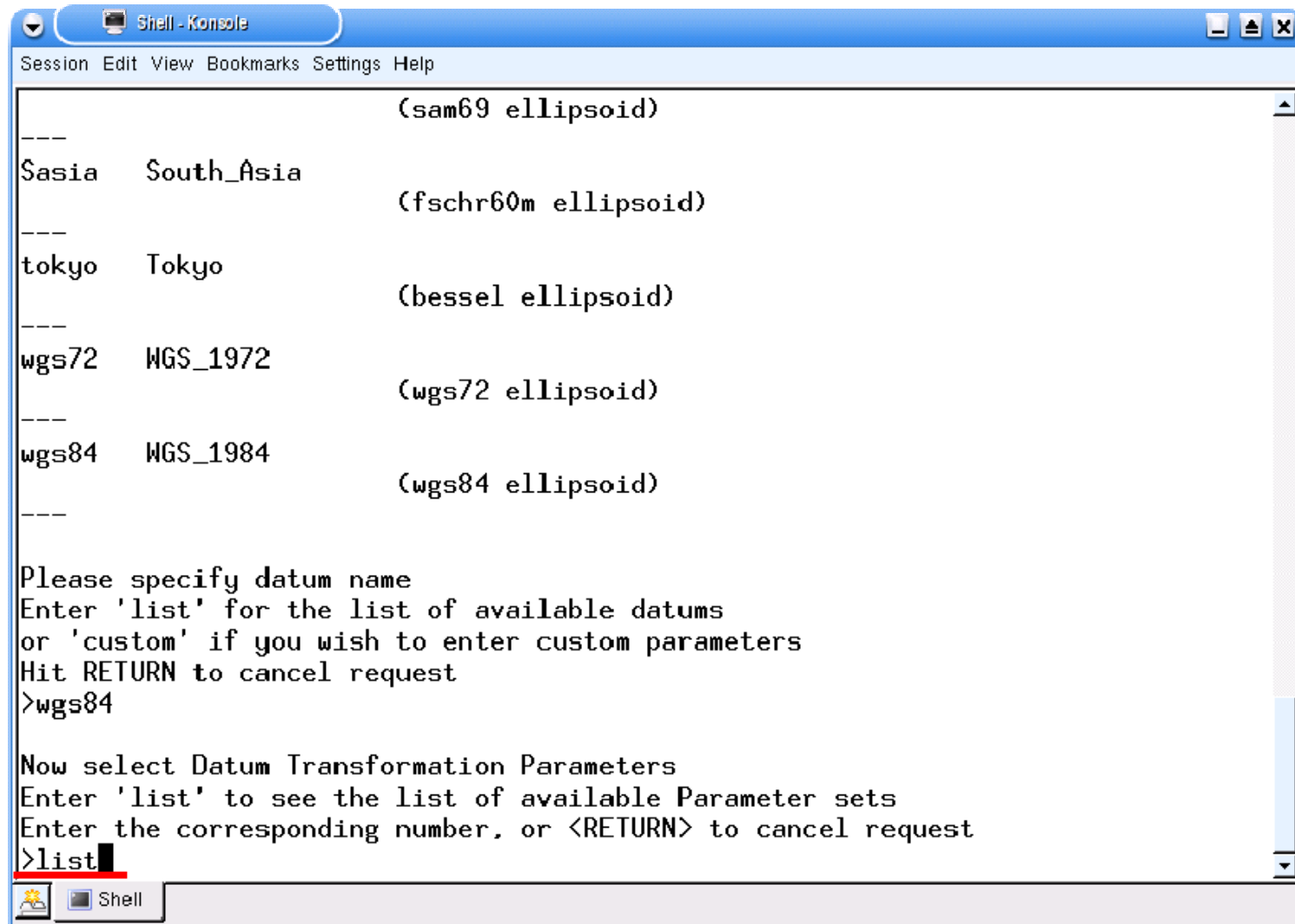
```
>list
Short Name      Long Name / Description
---
agd66   Australian_Geodetic_Datum_1966
              (australian ellipsoid)
---
agd84   Australian_Geodetic_Datum_1984
              (australian ellipsoid)
---
bel72   Reseau_National_Belge_1972
              (international ellipsoid)
---
carthage      Carthage
              (clark80 ellipsoid)
---
ch1903  CH1903
              (bessel ellipsoid)
---
etrs89   European_Terrestrial_Reference_System_1989
              (grs80 ellipsoid)
---
eur50    European_Datum_1950
              (international ellipsoid)
---
--More--(27%)
```

enter "wgs84", then hit "Enter (or Return)"



```
---  
SAD-69 Please use sam69 instead  
              (sam69 ellipsoid)  
---  
sam69 South_American_Datum_1969  
              (sam69 ellipsoid)  
---  
Sasia South_Asia  
              (fschr60m ellipsoid)  
---  
tokyo Tokyo  
              (bessel ellipsoid)  
---  
wgs72 WGS_1972  
              (wgs72 ellipsoid)  
---  
wgs84 WGS_1984  
              (wgs84 ellipsoid)  
---  
  
Please specify datum name  
Enter 'list' for the list of available datums  
or 'custom' if you wish to enter custom parameters  
Hit RETURN to cancel request  
>wgs84
```

enter "list", then hit "Enter (or Return)"



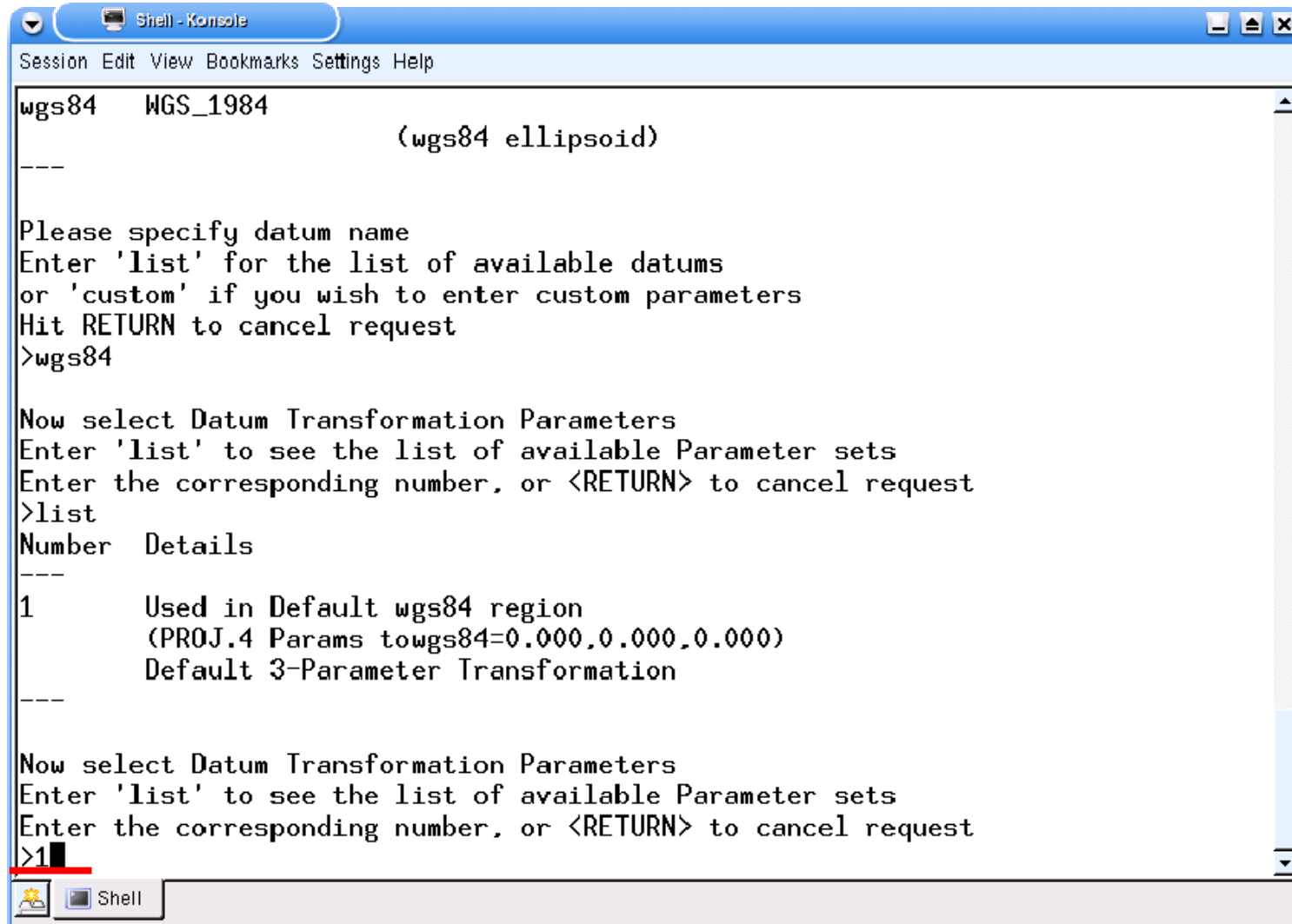
```
Shell - Konsole
Session Edit View Bookmarks Settings Help

----
(sam69 ellipsoid)
----
Sasia    South_Asia
         (fschr60m ellipsoid)
----
tokyo    Tokyo
         (bessel ellipsoid)
----
wgs72    WGS_1972
         (wgs72 ellipsoid)
----
wgs84    WGS_1984
         (wgs84 ellipsoid)
----

Please specify datum name
Enter 'list' for the list of available datums
or 'custom' if you wish to enter custom parameters
Hit RETURN to cancel request
>wgs84

Now select Datum Transformation Parameters
Enter 'list' to see the list of available Parameter sets
Enter the corresponding number, or <RETURN> to cancel request
>list
```

enter " 1 ", then hit "Enter (or Return)"



```
Shell - Konsole
Session Edit View Bookmarks Settings Help

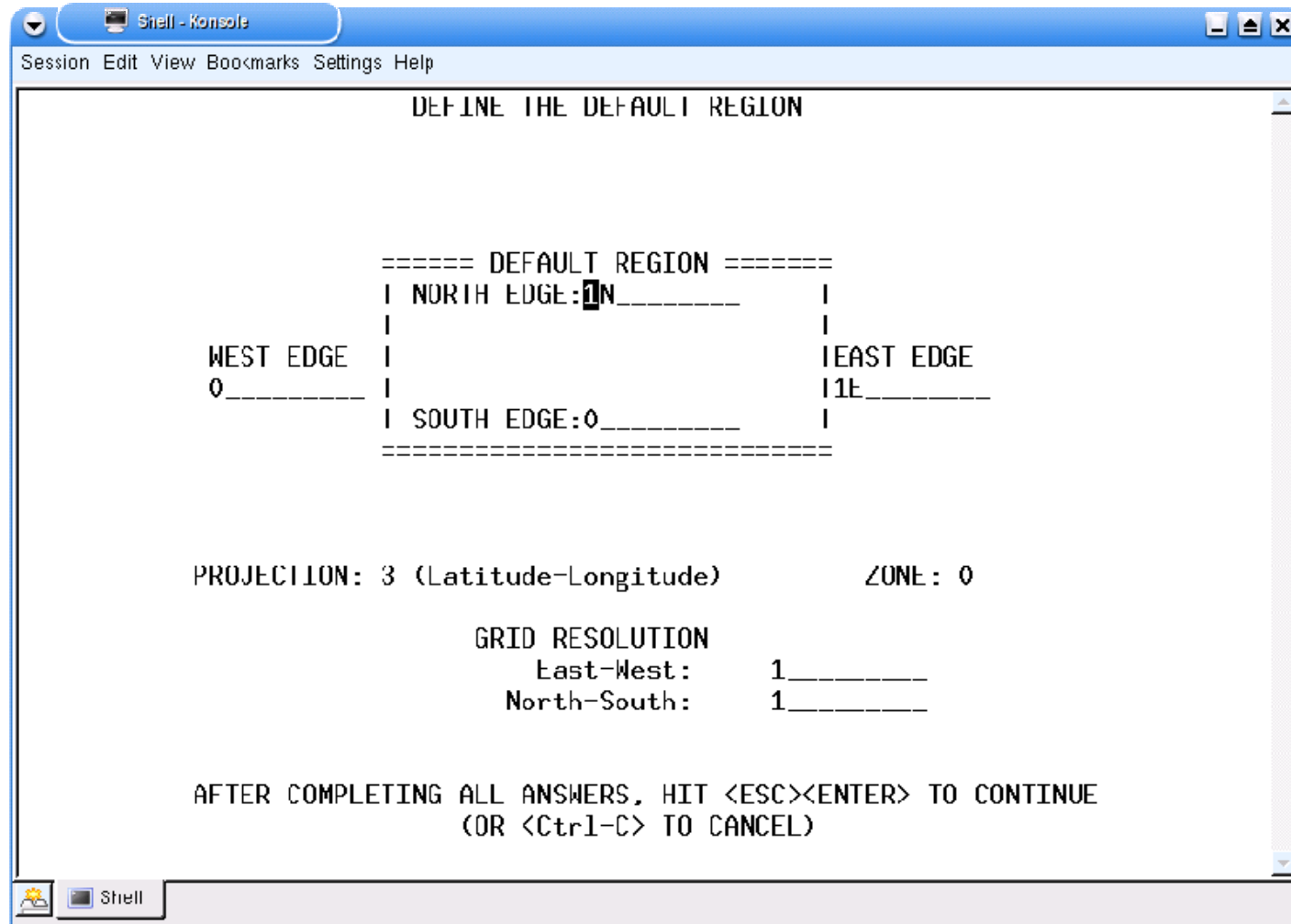
wgs84  WGS_1984
                                     (wgs84 ellipsoid)
---

Please specify datum name
Enter 'list' for the list of available datums
or 'custom' if you wish to enter custom parameters
Hit RETURN to cancel request
>wgs84

Now select Datum Transformation Parameters
Enter 'list' to see the list of available Parameter sets
Enter the corresponding number, or <RETURN> to cancel request
>list
Number  Details
---
1       Used in Default wgs84 region
        (PROJ.4 Params towgs84=0.000,0.000,0.000)
        Default 3-Parameter Transformation
---

Now select Datum Transformation Parameters
Enter 'list' to see the list of available Parameter sets
Enter the corresponding number, or <RETURN> to cancel request
>1
```

2.5 Definition of region



You can use “space” key to erase letters.

You can use “↓” key to move.

Define the region which includes entire Vietnam

NORTH EDGE : **24N**

SOUTH EDGE : **8N**

WEST EDGE : **102E**

EAST EDGE : **110E**

GRID RESOLUTION East-West : **00:00:15**

North-South : **00:00:15**

<= 1deg. : 240pixels
(500m resolution)

For latitude-longitude coordinate system, you have to define the GRID RESOLUTION in “degree:minutes:seconds (DMS)”.

Spatial resolution of image		GRID RESOLUTION
1 km	→	00:00:30
500 m	→	00:00:15
100 m	→	00:00:03

Shell - Konsole

Session Edit View Bookmarks Settings Help

DEFINE THE DEFAULT REGION

```
===== DFFAUI T REGTON =====
| NORTH EDGE:24N_____ |
|                               |
WEST EDGE |                               | EAST EDGE
102E_____ | 110E_____
| SOUTH EDGE:8N_____ |
=====
```

PROJECTION: 3 (Latitude-Longitude) ZONE: 0

GRID RESOLUTION

Fast-West: 00:00:15

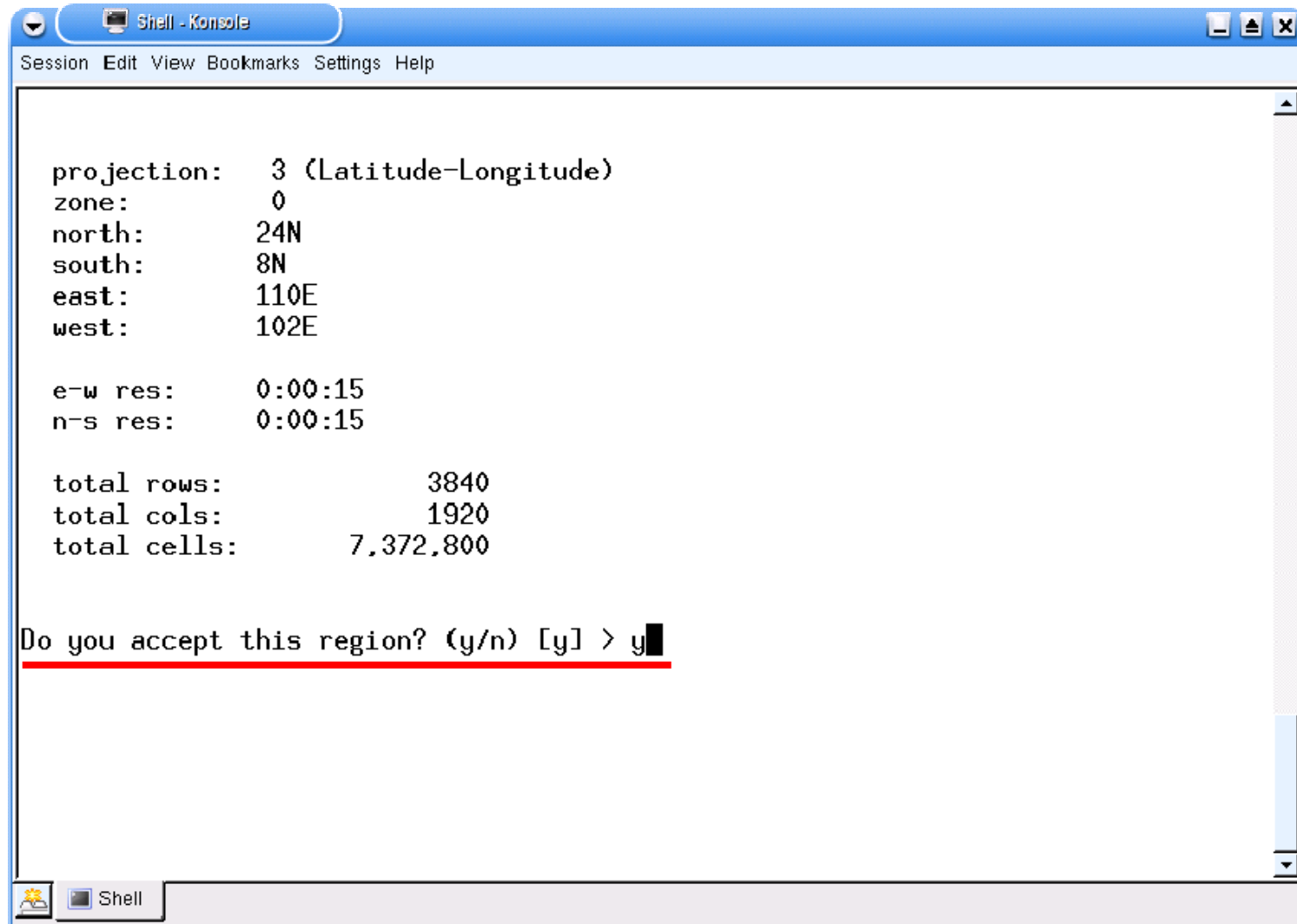
North-South: 00:00:15

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

Shell

After filling the above information,
hit "Esc" + "Enter (or Return)"

enter "y", then hit "Enter (or Return)"



The screenshot shows a 'Shell - Konsole' window with a menu bar (Session, Edit, View, Bookmarks, Settings, Help) and a terminal area. The terminal displays the following text:

```
projection: 3 (Latitude-Longitude)
zone: 0
north: 24N
south: 8N
east: 110E
west: 102E

e-w res: 0:00:15
n-s res: 0:00:15

total rows: 3840
total cols: 1920
total cells: 7,372,800

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

The prompt line is underlined in red. At the bottom of the window, there is a taskbar with a 'Shell' icon and a 'Shell' label.

The screenshot shows a terminal window titled "Shell - Konsole". The window has a menu bar with "Session", "Edit", "View", "Bookmarks", "Settings", and "Help". The main content area displays the following text:

```
projection: 3 (Latitude-Longitude)
zone: 0
north: 24N
south: 8N
east: 110E
west: 102E

e-w res: 0:00:15
n-s res: 0:00:15

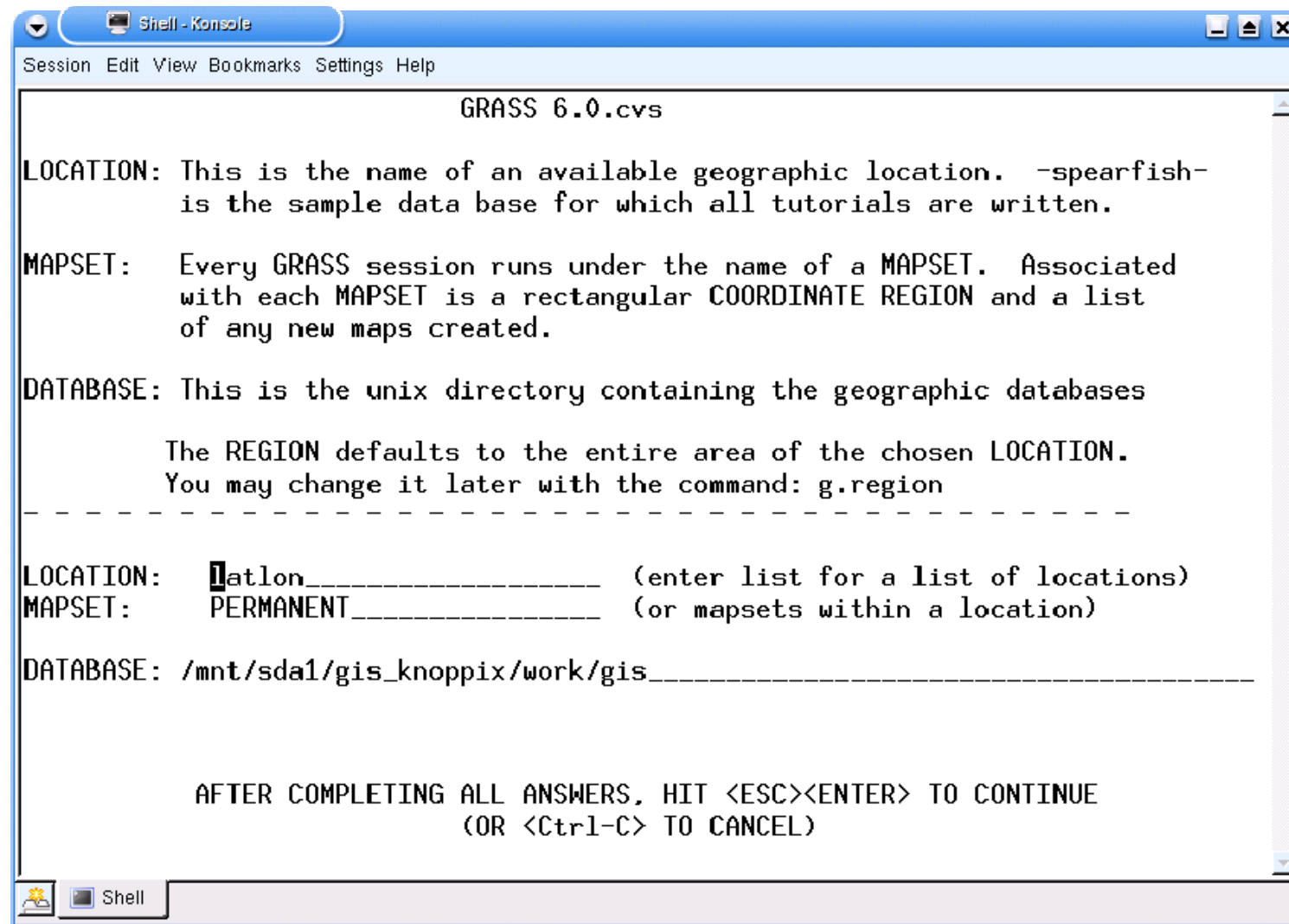
total rows: 3840
total cols: 1920
total cells: 7,372,800

Do you accept this region? (y/n) [y] > y
LOCATION <latlon> created!

Hit RETURN -->
```

The text "Hit RETURN -->" is underlined with a red line. At the bottom of the window, there is a taskbar with a "Shell" icon and the label "Shell".

hit "Enter (or Return)"



```
GRASS 6.0.cvs

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

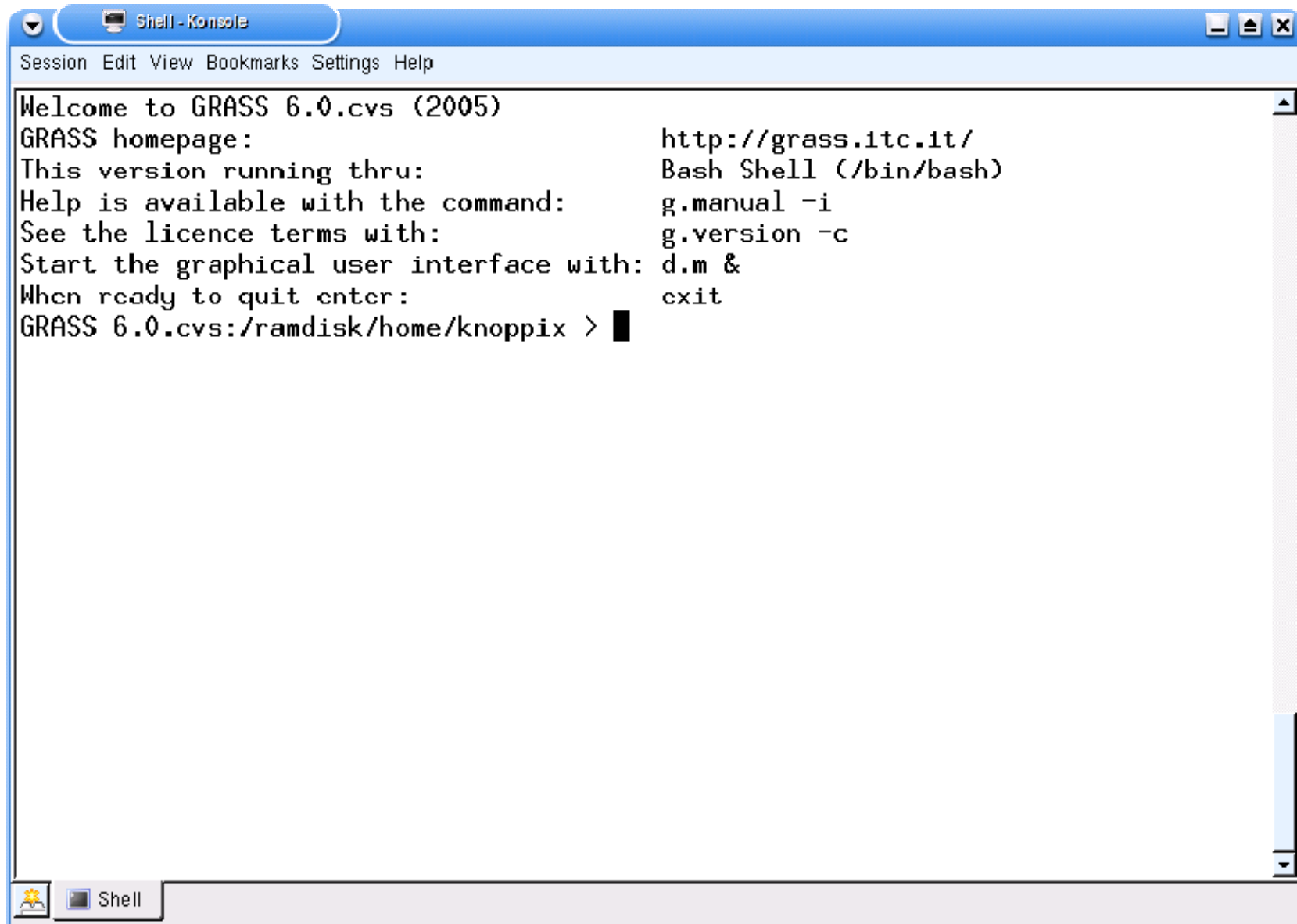
        The REGION defaults to the entire area of the chosen LOCATION.
        You may change it later with the command: g.region
        -----

LOCATION:  latlon_____ (enter list for a list of locations)
MAPSET:  PERMANENT_____ (or mapsets within a location)

DATABASE: /mnt/sda1/gis_knoppix/work/gis_____

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

enter “ y “, then hit “Esc” + "Enter (or Return)"



```
Session Edit View Bookmarks Settings Help

Welcome to GRASS 6.0.cvs (2005)
GRASS homepage:                http://grass.itc.it/
This version running thru:      Bash Shell (/bin/bash)
Help is available with the command: g.manual -i
See the licence terms with:      g.version -c
Start the graphical user interface with: d.m &
When ready to quit enter:        exit
GRASS 6.0.cvs:/ramdisk/home/knoppix > █
```

Section 2. Data Processing in Latitude-Longitude Coordinate System

3. Import raster data

GRASS <i>Module command</i>	<i>Import</i> <i>Raster format</i>
r.in.ascii	GRASS ASCII
r.in.bin	BIL, GMT binary files, LANDSAT TM5
r.in.gdal	ARC/INFO ASCII/Binary GRID, BIL, ERDAS (LAN, IMG), USGS DOQ, JPEG, SAR CEOS, EOSAT, GeoTIFF, PPM/PNM, SDTS DEM, GIF, PNG (see also http://www.gdal.org/formats_list.html)

Image format: The individual rasters always have positive, integral values in the known pixel-based image formats such as PPM, PNG, JPEG, and GIF.

ASCII format: The individual rasters of the ASCII format can contain positive and negative, integral values as well as floating point values. The ASCII-GRID of Arcinfo is an example of this format.

Binary format: In the binary raster format the individual pixels with positive and negative, integral values or floating point values can also be saved in different channels with different resolutions. (Geo)TIFF or ERDAS/IMG are examples for this format.

3.1 Import MODIS 10-days composite image

```
> r.in.bin input=A output=B bytes=C north=D south=E east=F west=G rows=H cols=I
```

A = /mnt/sda1/gis_knoppix/data/(input file name)

B = (output file name)

C = 2 (bytes)

D = 24 (north)

E = 8 (south)

F = 110 (east)

G = 102 (west)

H = 3840 (rows)

I = 1920 (cols)

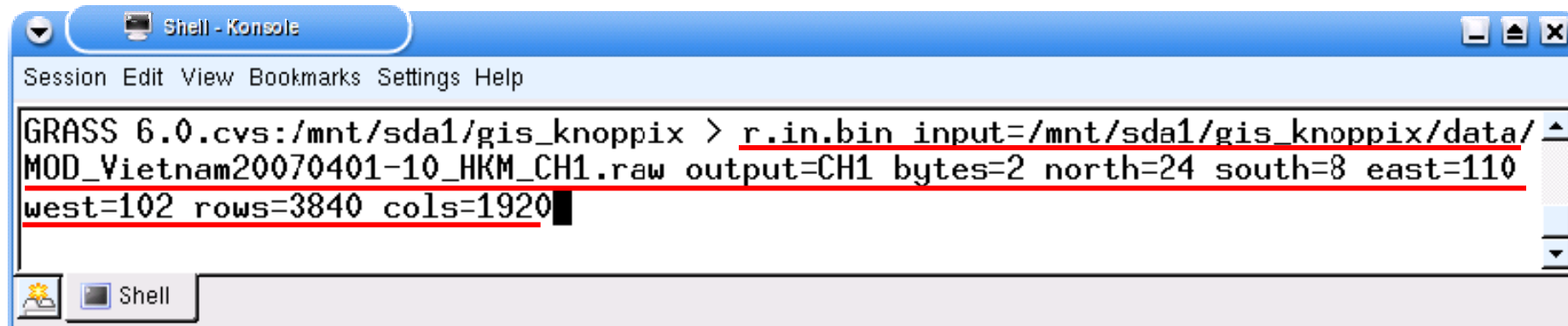
For Vietnam

$\leq (24 - 8) * 240 = 3840$ 1deg. : 240pixels (500m resolution)

$\leq (110 - 102) * 240 = 3840$ 1deg. : 240pixels (500m resolution)

: space

r.in.bin allows a user to create a (binary) GRASS raster map layer from a variety of binary raster data formats.

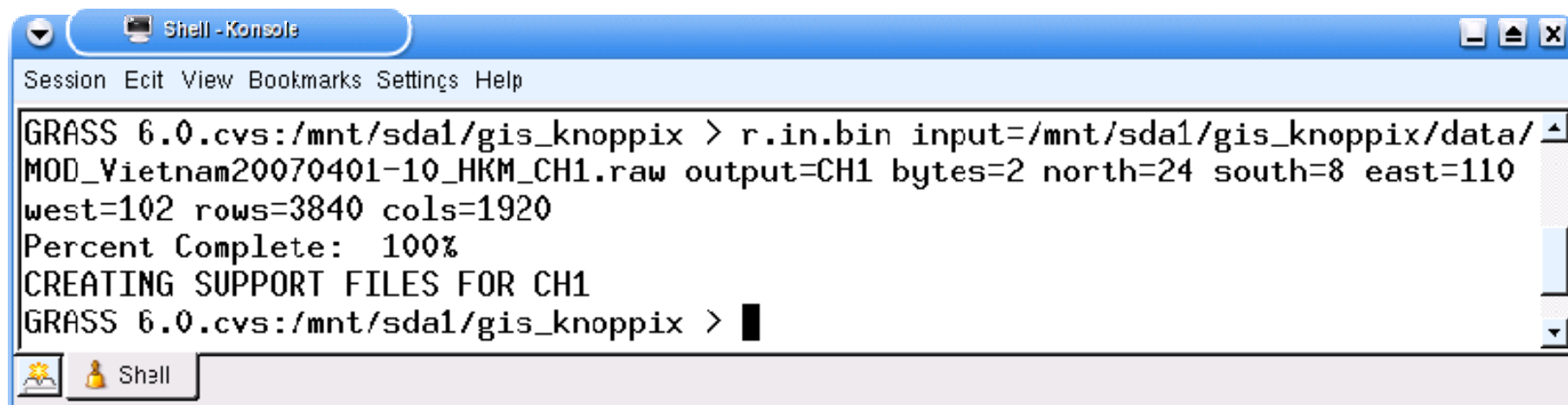


```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix > r.in.bin input=/mnt/sda1/gis_knoppix/data/  
MOD_Vietnam20070401-10_HKM_CH1.raw output=CH1 bytes=2 north=24 south=8 east=110  
west=102 rows=3840 cols=1920
```

If you have stored MODIS images in “data” directory:

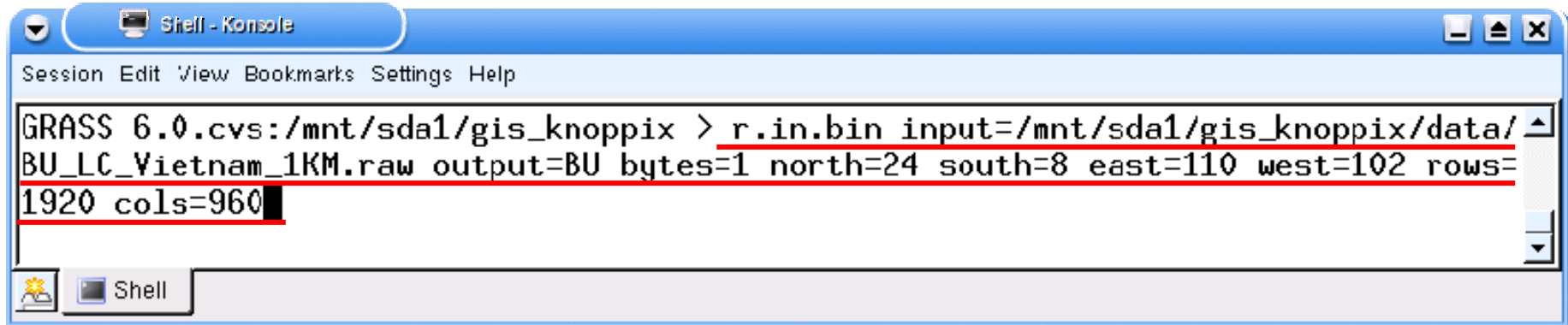
```
> r.in.bin input=/mnt/sda1/gis_knoppix/data/MOD_Vietnam2007040110_  
HKM_CH1.raw output=CH1 bytes=2 north=24 south=8 east=110  
west=102 rows=3840 cols=1920
```

␣ : space



```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix > r.in.bin input=/mnt/sda1/gis_knoppix/data/  
MOD_Vietnam20070401-10_HKM_CH1.raw output=CH1 bytes=2 north=24 south=8 east=110  
west=102 rows=3840 cols=1920  
Percent Complete: 100%  
CREATING SUPPORT FILES FOR CH1  
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix >
```

3.2 Import land cover map



```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix > r.in.bin input=/mnt/sda1/gis_knoppix/data/  
BU_LC_Vietnam_1KM.raw output=BU bytes=1 north=24 south=8 east=110 west=102 rows=  
1920 cols=960
```

> r.in.bin **input=A** **output=B** **bytes=C** **north=D** **south=E** **east=F** **west=G** **rows=H** **cols=I**

A = /mnt/sda1/gis_knoppix/data/(input file name)

B = (output file name)

C = 1 (bytes)

D = 24 (north)

E = 8 (south)

F = 110 (east)

G = 102 (west)

H = 1920 (rows) $\leq (24 - 8) * 120 = 1920$ 1deg. : 120pixels (1km resolution)

I = 960 (cols) $\leq (110 - 102) * 120 = 960$ 1deg. : 120pixels (1km resolution)

: space

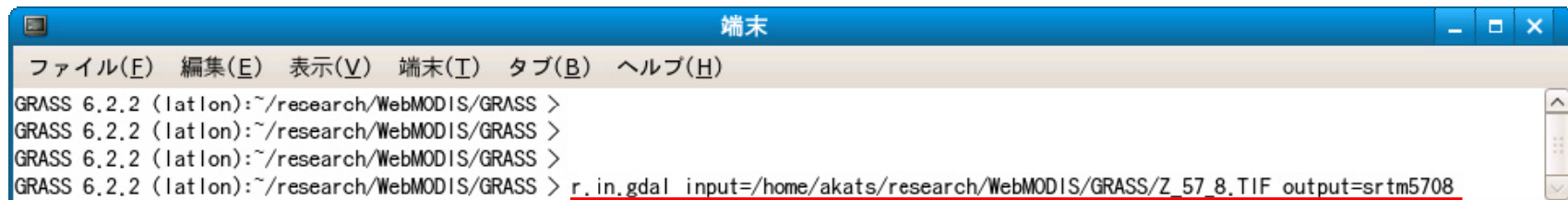
3.3 Import SRTM image

```
> r.in.gdal input = A output = B
```

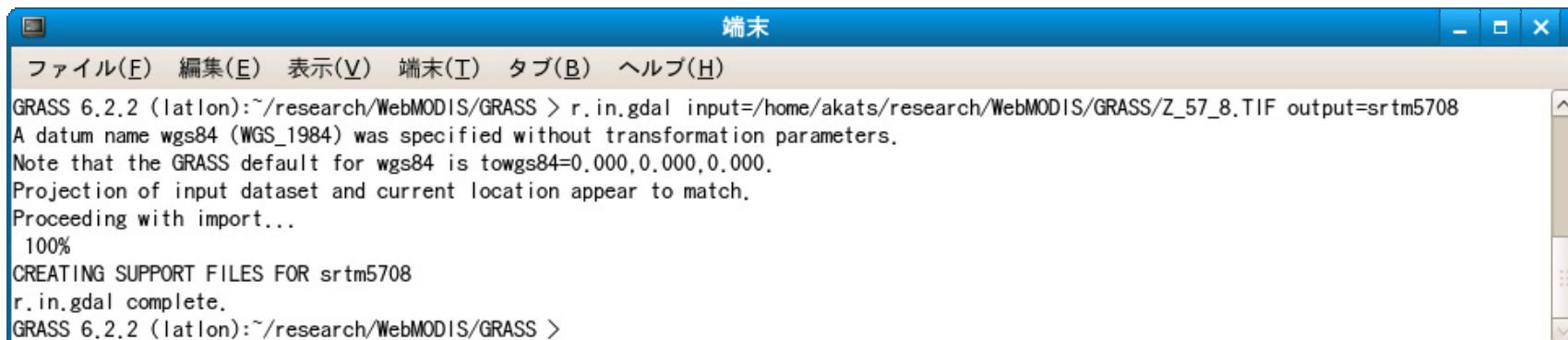
A = /mnt/sda1/gis_knoppix/data/(input file name)

B = (output file name) : space

r.in.gdal allows a user to create a (binary) GRASS raster map layer, or imagery group, from any GDAL supported raster map format, with an optional title.



```
端末
ファイル(E) 編集(E) 表示(V) 端末(T) タブ(B) ヘルプ(H)
GRASS 6.2.2 (latlon):~/research/WebMODIS/GRASS >
GRASS 6.2.2 (latlon):~/research/WebMODIS/GRASS >
GRASS 6.2.2 (latlon):~/research/WebMODIS/GRASS >
GRASS 6.2.2 (latlon):~/research/WebMODIS/GRASS > r.in.gdal input=/home/akats/research/WebMODIS/GRASS/Z_57_8.TIF output=srtm5708
```

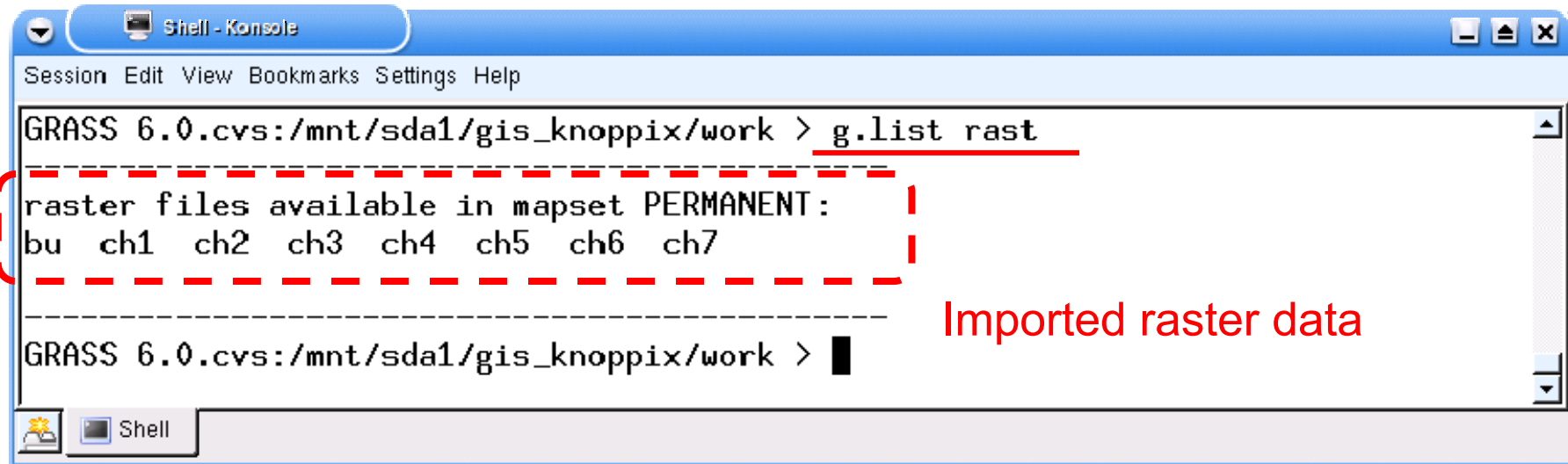


```
端末
ファイル(E) 編集(E) 表示(V) 端末(T) タブ(B) ヘルプ(H)
GRASS 6.2.2 (latlon):~/research/WebMODIS/GRASS > r.in.gdal input=/home/akats/research/WebMODIS/GRASS/Z_57_8.TIF output=srtm5708
A datum name wgs84 (WGS_1984) was specified without transformation parameters.
Note that the GRASS default for wgs84 is towgs84=0.000,0.000,0.000.
Projection of input dataset and current location appear to match.
Proceeding with import...
100%
CREATING SUPPORT FILES FOR srtm5708
r.in.gdal complete.
GRASS 6.2.2 (latlon):~/research/WebMODIS/GRASS >
```

3.4 List imported raster data

```
> g.list  type = rast
```

 : space



The screenshot shows a terminal window titled "Shell - Konsole". The prompt is "GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work >". The command g.list rast has been entered. The output is "raster files available in mapset PERMANENT:" followed by a list of files: "bu ch1 ch2 ch3 ch4 ch5 ch6 ch7". This output is enclosed in a red dashed box. Below the output, there is a red text label "Imported raster data". The prompt "GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work >" is followed by a cursor.

```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > g.list rast  
-----  
raster files available in mapset PERMANENT:  
bu ch1 ch2 ch3 ch4 ch5 ch6 ch7  
-----  
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > Imported raster data
```

g.list allows the user to list user-specified, available and accessible files from *mapsets* under the user's current location.

Section 2. Data Processing in Latitude-Longitude Coordinate System

4. Display raster data

Step 0. Import raster data

⇒ See “3. Import raster data”

Step 1. Launch monitor

```
> d.mon x#
```

= 0, 1, 2, ..., 6 : space

d.mon allows the user to start, select, list, query the status of, release control of, stop, and unlock control of, available graphics monitors.

Step 2. Display raster

```
> d.rast map = A
```

A = (input file name) : space

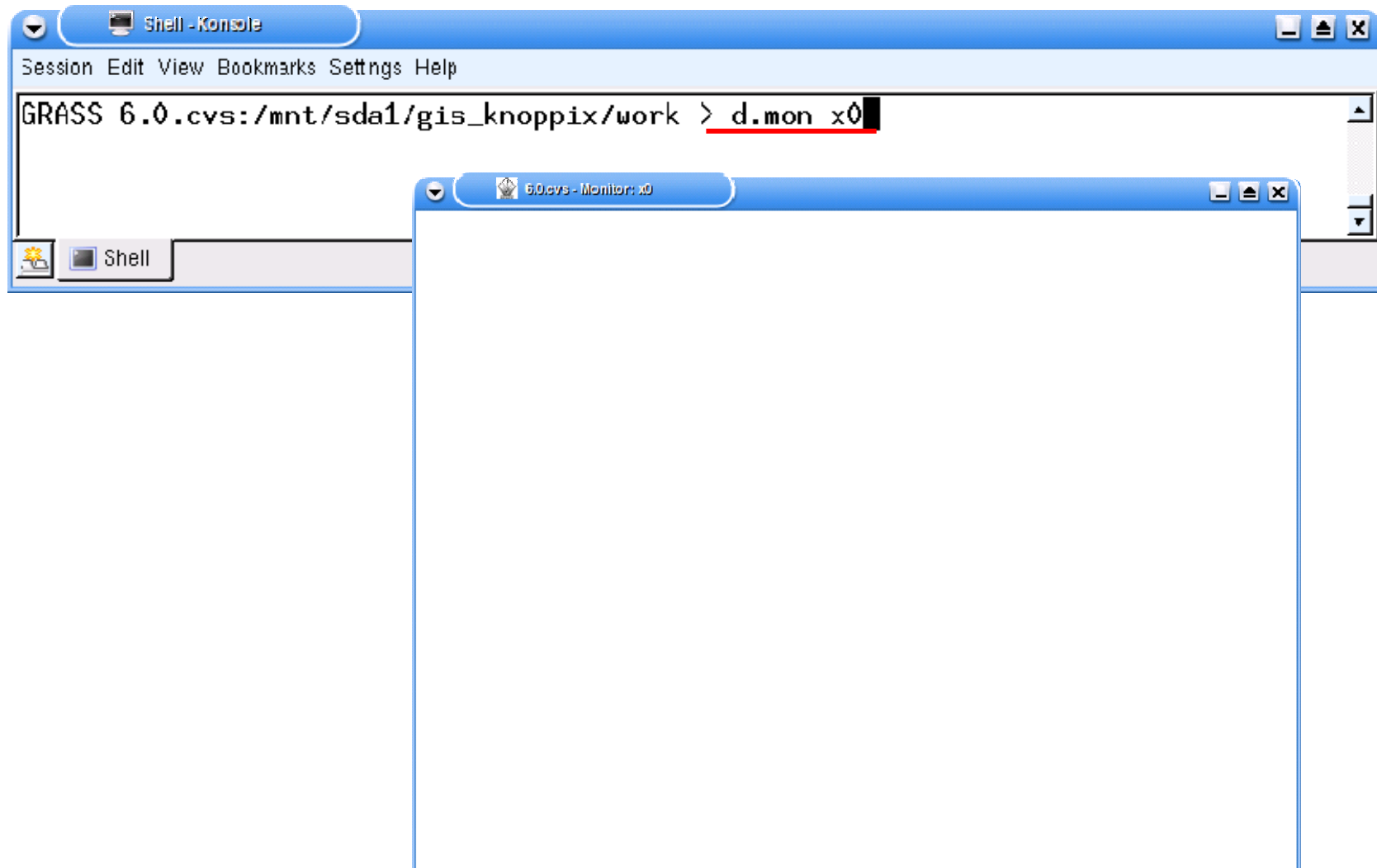
d.rast displays raster map layer(s) *name* in the active display frame on the graphics monitor.

4.1 Launch Monitor

> d.mon x#

= 0, 1, 2, ..., 6

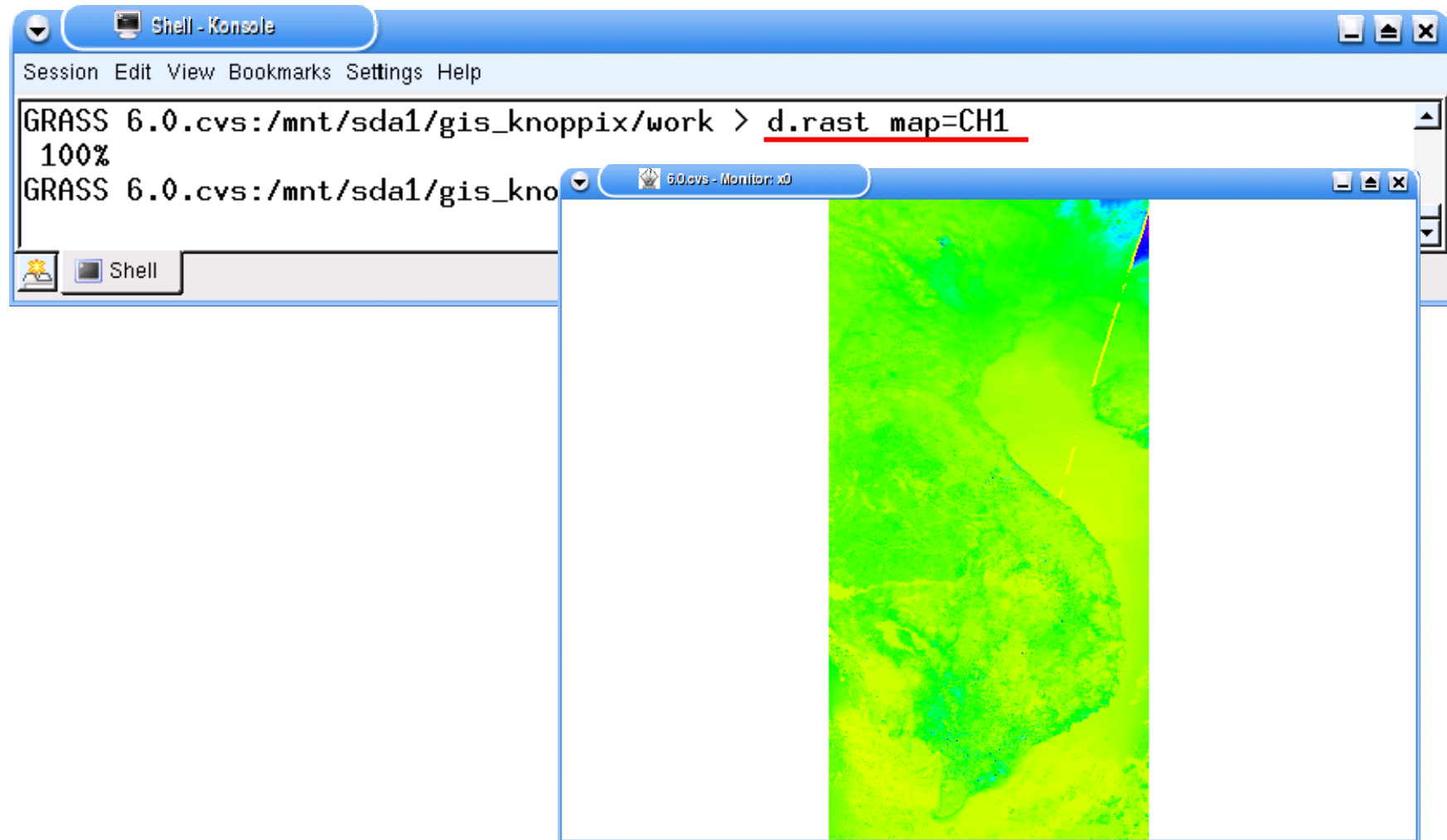
: space



4.2 Display raster image

> d.rast map = A

A = (input file name) : space



4.3 Change image color

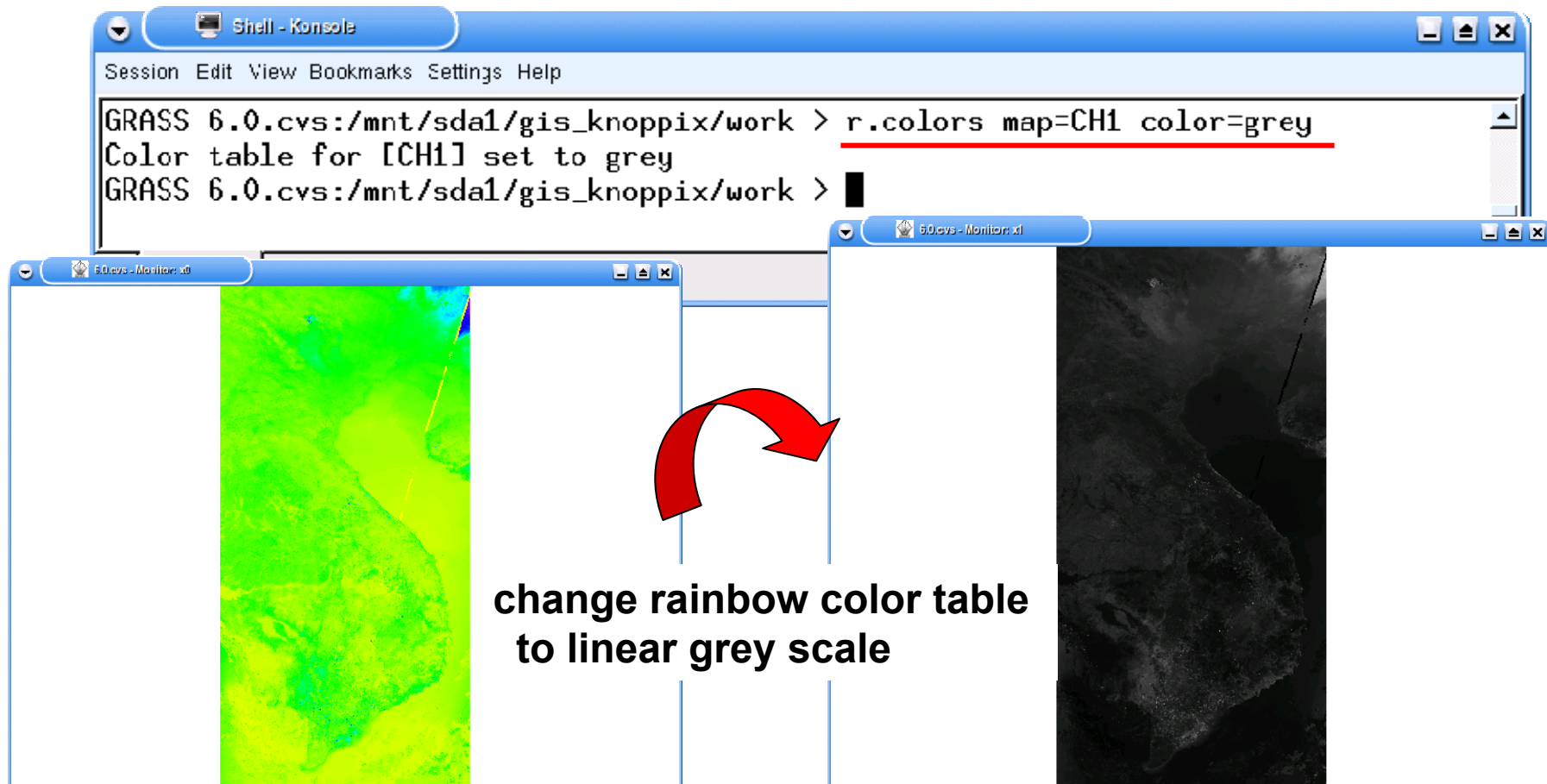
□ : space

```
> r.colors □map = A □color = type
```

A = (input file name)

B = (type of color table)

r.colors allows the user to create and/or modify the color table for a raster map layer.



4.3.1 Change image color using new color table

< Style of Color table >

“raster value” □ “R” □ “G” □ “B”

0 0 0 0

1 255 0 0

2 0 255 0

3 0 0 255

4 100 255 0

5 0 130 200

6 30 90 160

7 255 255 255

“raster value” □ “color”

-1 black

0 blue

200 green

400 yellow

800 brown

1500 red

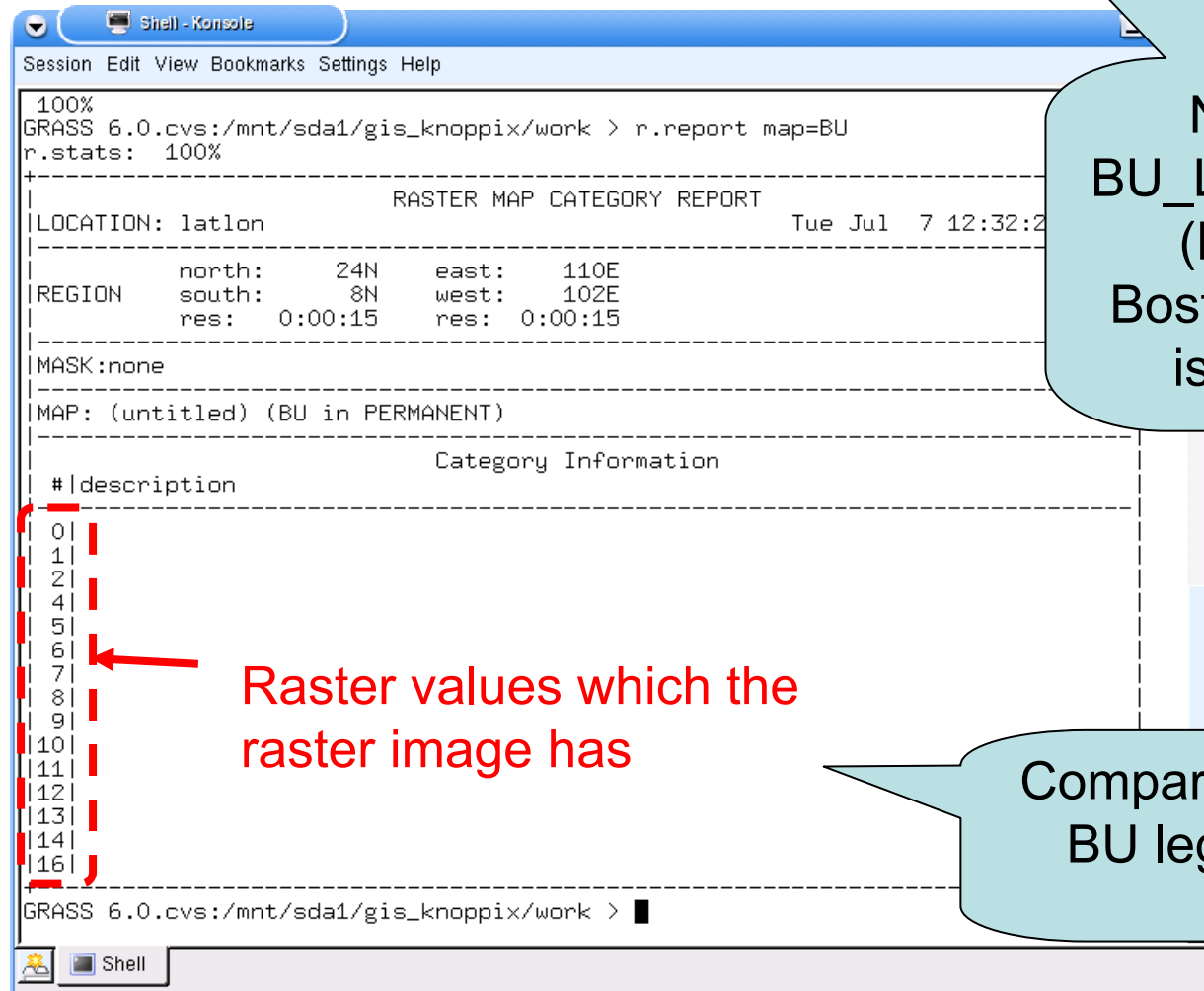
3000 white

nv black

< procedures to create new color table on the Terminal window >

Step 1. Reports statistics for raster map layers.

```
> r.report map=(file name)
```



```
100%
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > r.report map=BU
r.stats: 100%
-----
LOCATION: latlon                                Tue Jul  7 12:32:2
-----
REGION      north:    24N    east:    110E
            south:    8N     west:    102E
            res:     0:00:15  res:     0:00:15
-----
MASK:none
-----
MAP: (untitled) (BU in PERMANENT)
-----
Category Information
-----
#|description
0|
1|
2|
4|
5|
6|
7|
8|
9|
10|
11|
12|
13|
14|
16|
-----
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work >
```

Now, we are using
BU_LC_Vietnam_1KM.raw
(Land cover map of
Boston University) which
is imported as “BU”.

Raster values which the
raster image has

Compare these values with
BU legend of next slide

BU Legend

0 : Water (and Goode's interrupted space)
1 : Evergreen Needleleaf Forest
2 : Evergreen Broadleaf Forest
3 : Deciduous Needleleaf Forest
4 : Deciduous Broadleaf Forest
5 : Mixed Forest
6 : Closed Shrublands
7 : Open Shrubland
8 : Woody Savannas
9 : Savannas
10 : Grassland
11 : Permanent Wetlands
12 : Croplands
13 : Urban and Built-up
14 : Cropland/Natural Vegetation Mosaic
15 : Snow and Ice
16 : Barren or Sparsely Vegetated

BU Color Table

0 : 134 201 226
1 : 33 138 33
2 : 49 205 49
3 : 154 205 49
4 : 151 250 151
5 : 143 187 143
6 : 187 143 143
7 : 245 222 179
8 : 218 235 157
9 : 255 214 0
10 : 239 184 102
11 : 70 130 180
12 : 250 238 115
13 : 255 0 0
14 : 153 147 85
15 : 255 255 255
16 : 190 190 189

Step 2. Create color table using *r.colors* command on Terminal window.

```
> r.colors map=(file name) color=rules << EOF  : space  
> 0 134 201 226  
> 1 33 138 33  
  .  
  .  
  .  
  .  
  .  
> 16 190 190 189  
> EOF
```

enter RGB color corresponding raster
value line by line
and then hit "Enter" key

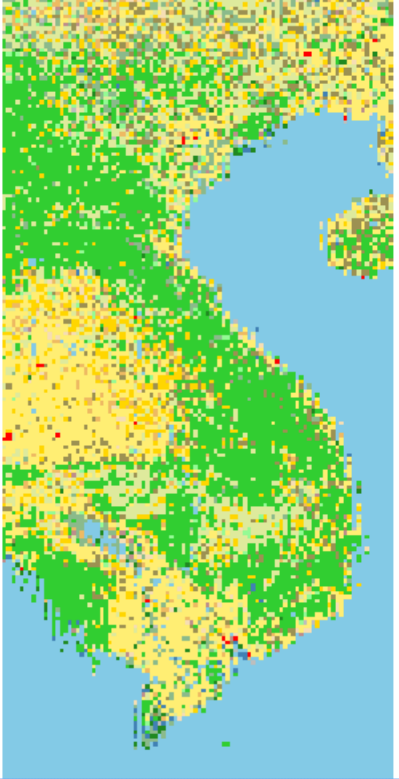
: hit "Enter" key

Shell - Konsole

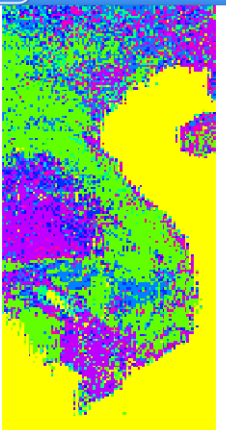
Session Edit View Bookmarks Settings Help

```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > r.colors map=BU color=rules<<EOF  
> 0 134 201 226  
> 1 33 138 33  
> 2 49 205 49  
> 3 154 205 49  
> 4 151 250 151  
> 5 143 187 143  
> 6 187 143 143  
> 7 245 222 179  
> 8 218 235 157  
> 9 255 214 0  
> 10 239 184 102  
> 11 70 130 180  
> 12 250 238 115  
> 13 255 0 0  
> 14 153 147 85  
> 15 255 255 255  
> 16 190 190 189  
> EOF  
Color table for [BU] set to rules  
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work >
```

6.0.cvs - Monitor: x1



6.0.cvs - Monitor: x1

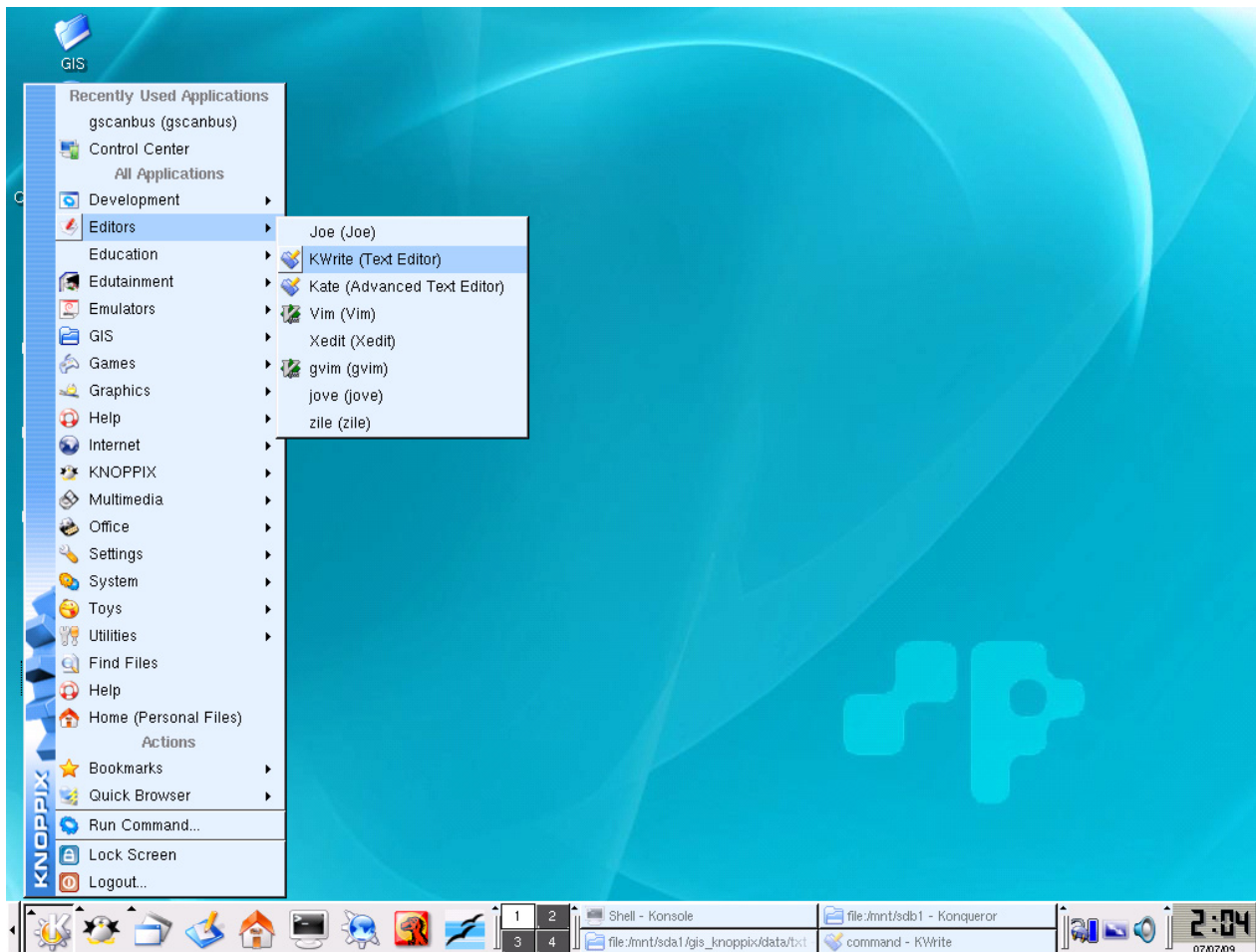


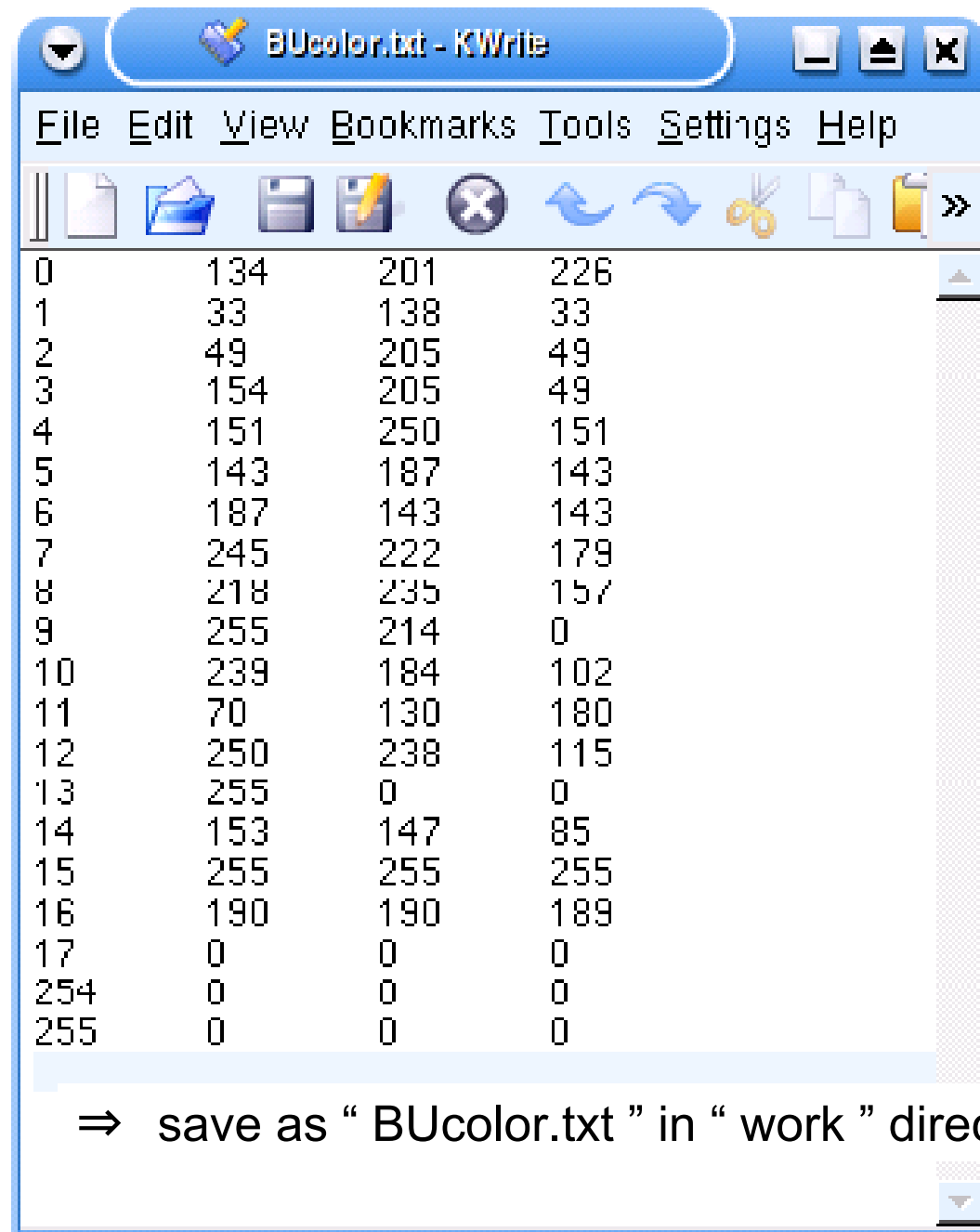
< procedures to create new color table using text editor >

Step 1. Reports statistics for raster map layers.

Step 2. Create color table using text editor.

open text editor <Editors> → <KWrite>



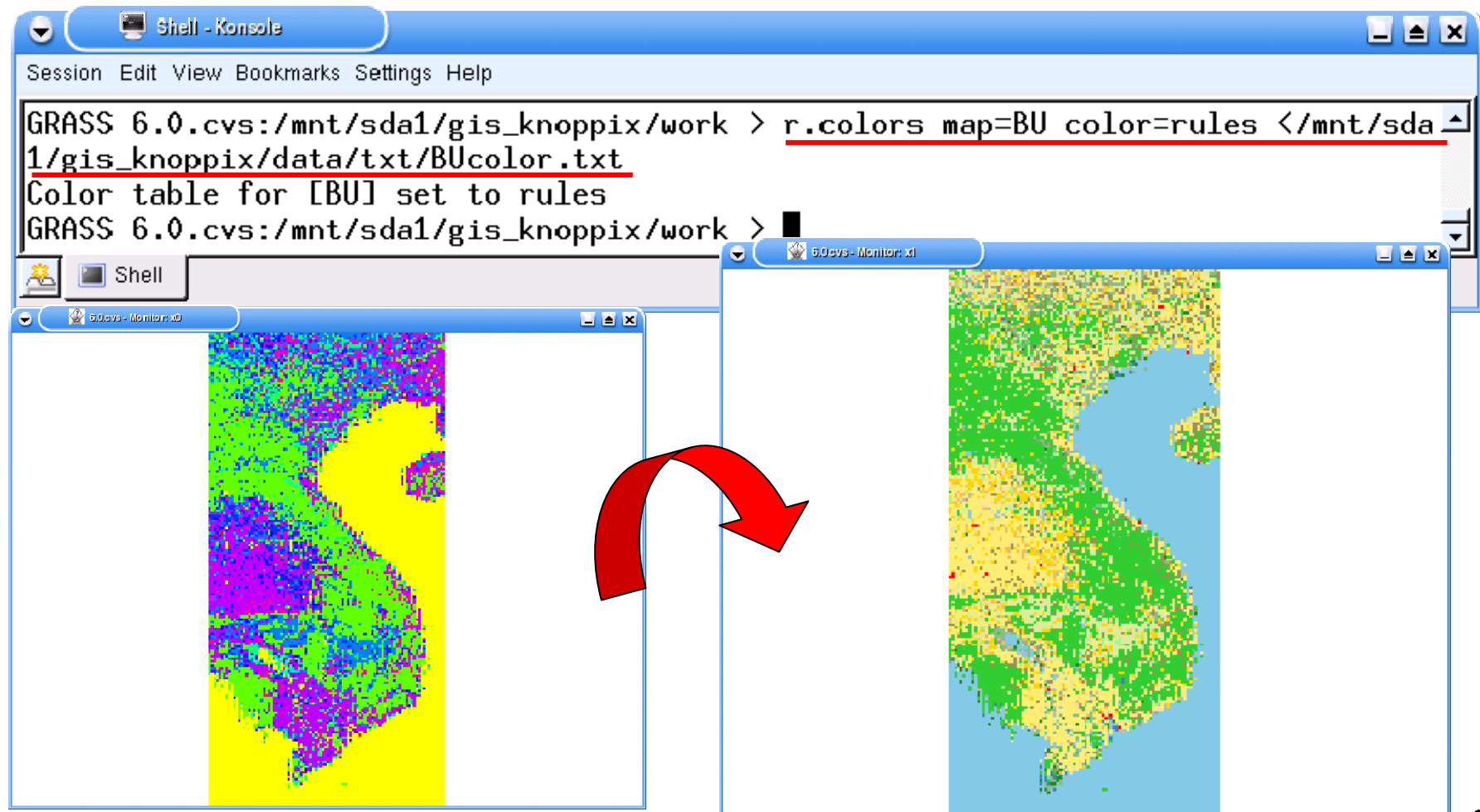


Step 3. Change color table using ***r.colors*** command and color table text file .

> r.colors map= A color = rules < B.txt

A = (input file name)

B = (color table text file)



4.4 Display images in RGB composite

```
> d.rgb red=A green=B blue=C
```

A = (file name1)

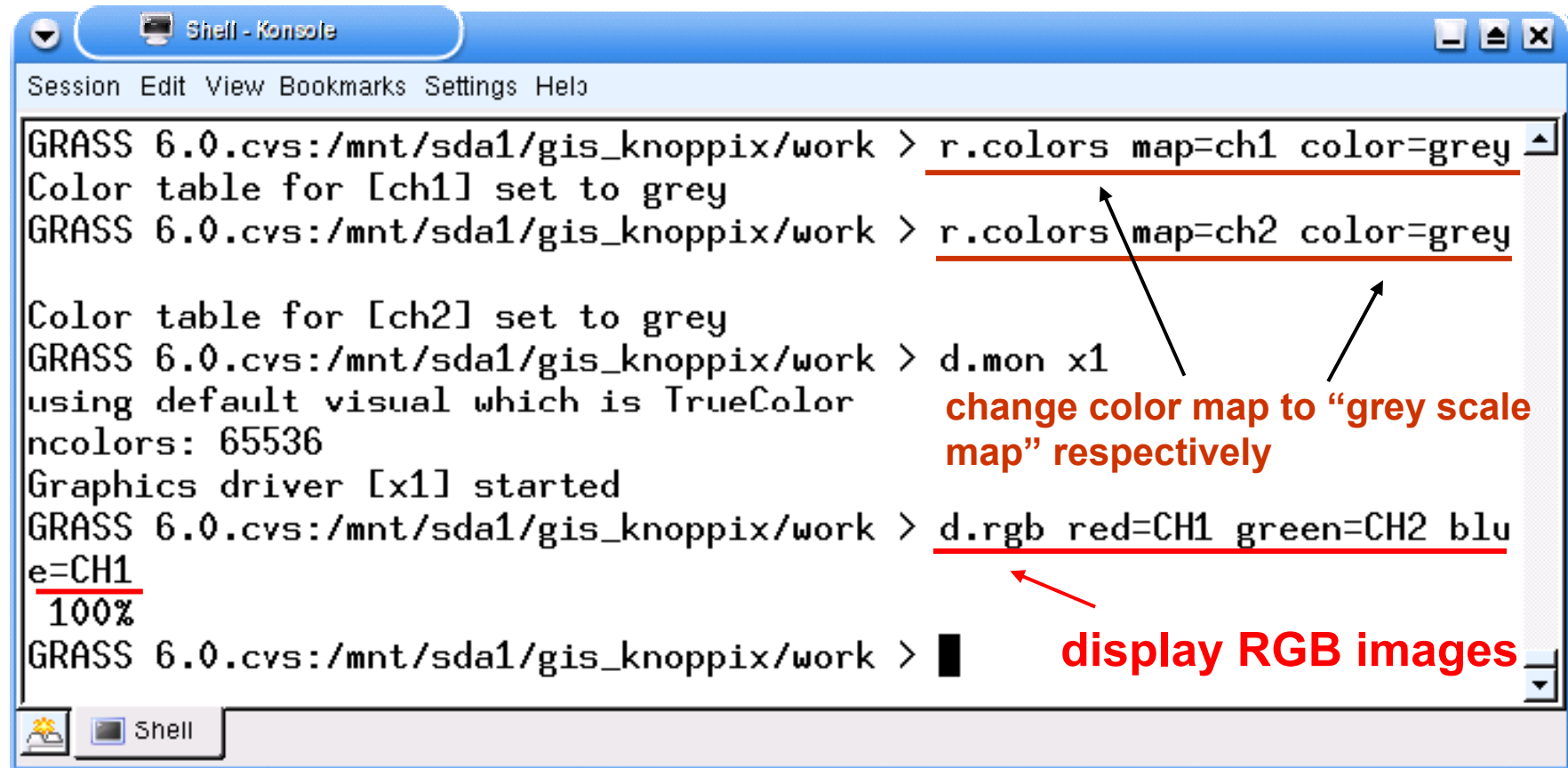
B = (file name2)

C = (file name3)

: space

d.rgb - Displays three user-specified raster map layers as red, green, and blue overlays in the active graphics frame. **This command does not create new image file.**

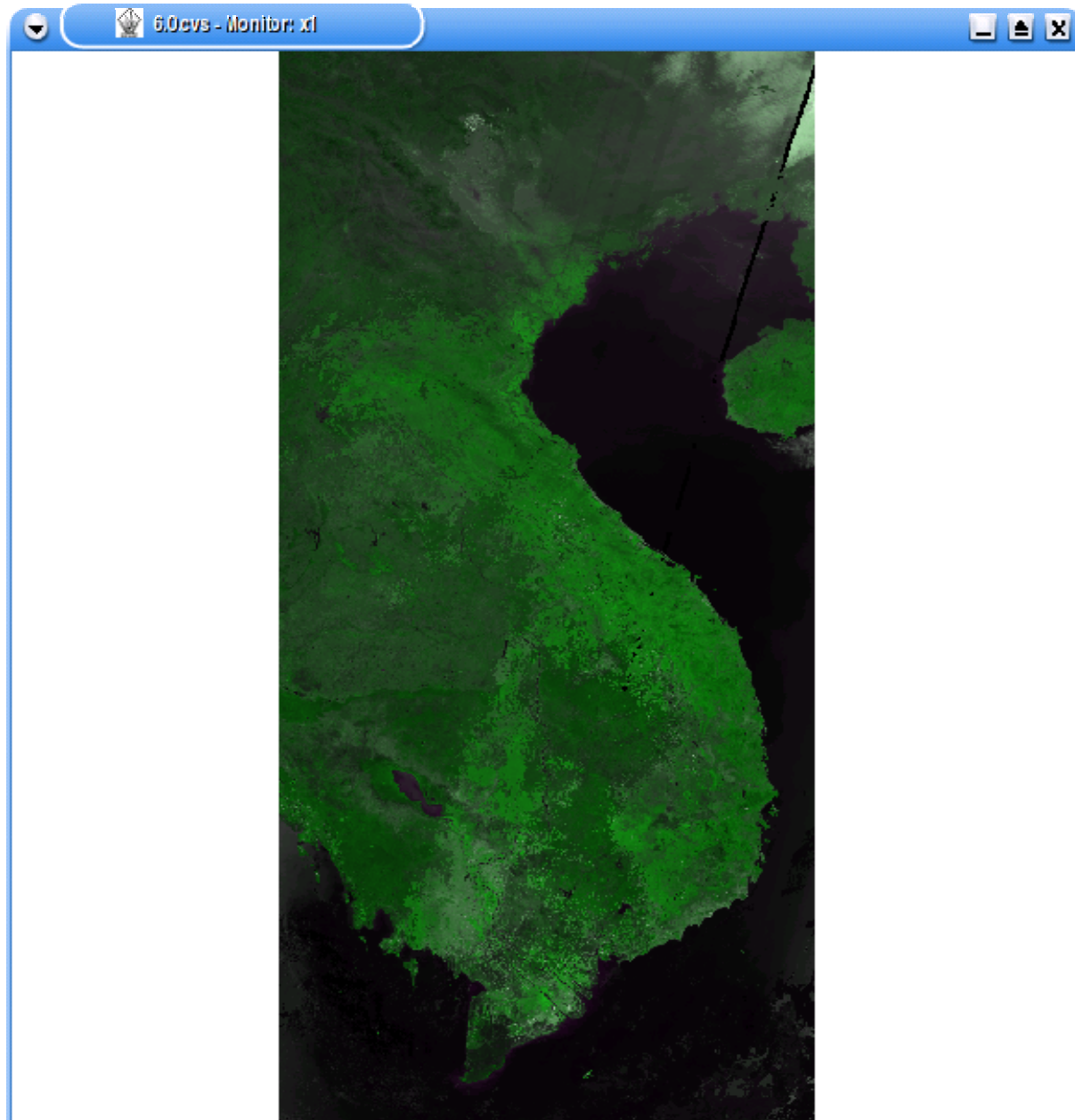
The color table of each image must be “grey-scale”.



The screenshot shows a terminal window titled "Shell - Konsole" with a menu bar (Session, Edit, View, Bookmarks, Settings, Help). The terminal output shows GRASS 6.0 commands and their results. Three red annotations with arrows point to specific parts of the command lines:

- An arrow points from the text "change color map to 'grey scale map' respectively" to the `color=grey` parameter in the first `r.colors` command.
- Another arrow points from the same text to the `color=grey` parameter in the second `r.colors` command.
- A red arrow points from the text "display RGB images" to the `d.rgb` command.

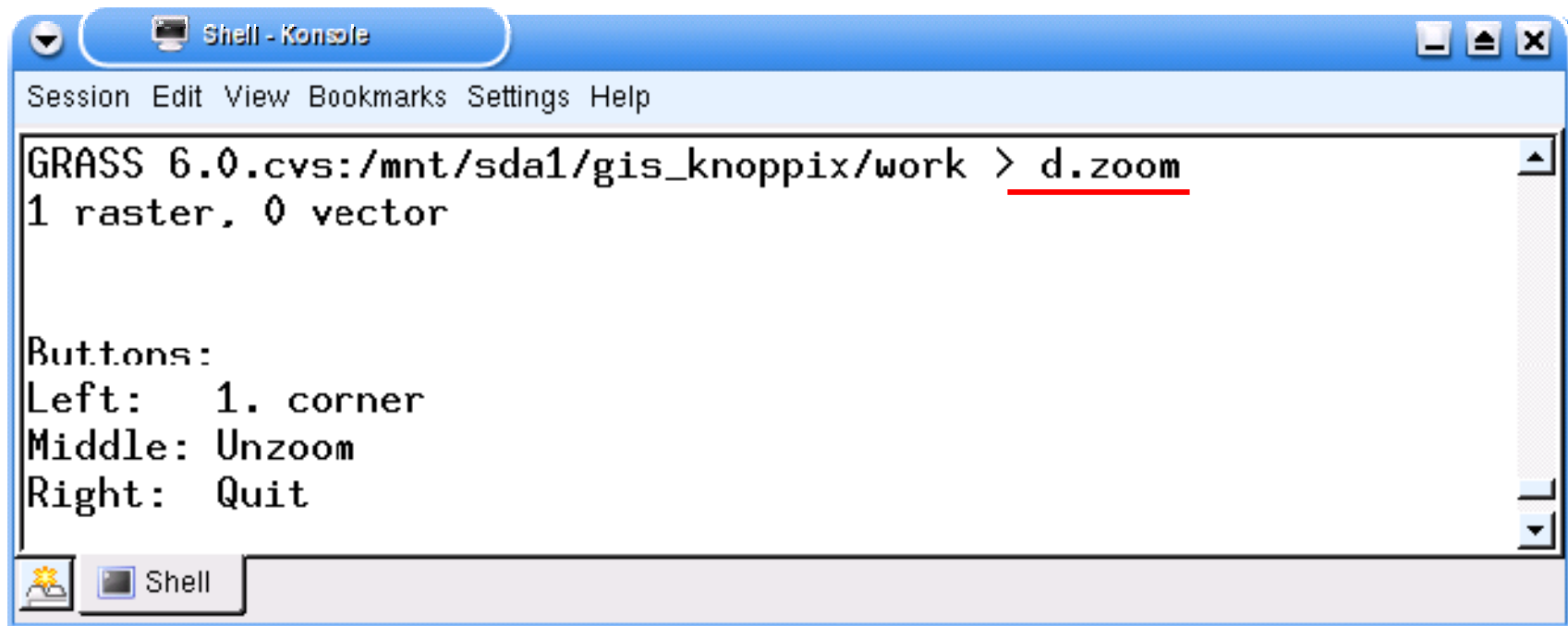
```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > r.colors map=ch1 color=grey
Color table for [ch1] set to grey
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > r.colors map=ch2 color=grey
Color table for [ch2] set to grey
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.mon x1
using default visual which is TrueColor
ncolors: 65536
Graphics driver [x1] started
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.rgb red=CH1 green=CH2 blue=CH1
100%
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > █
```



4.5 Zoom in the image

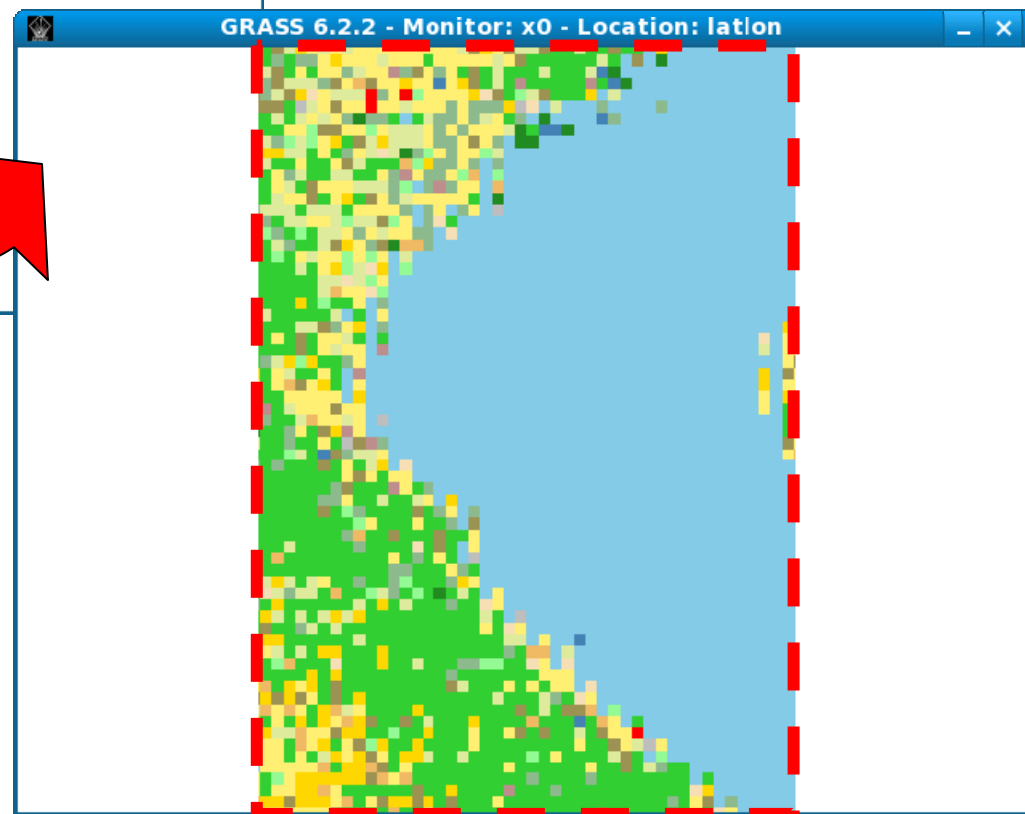
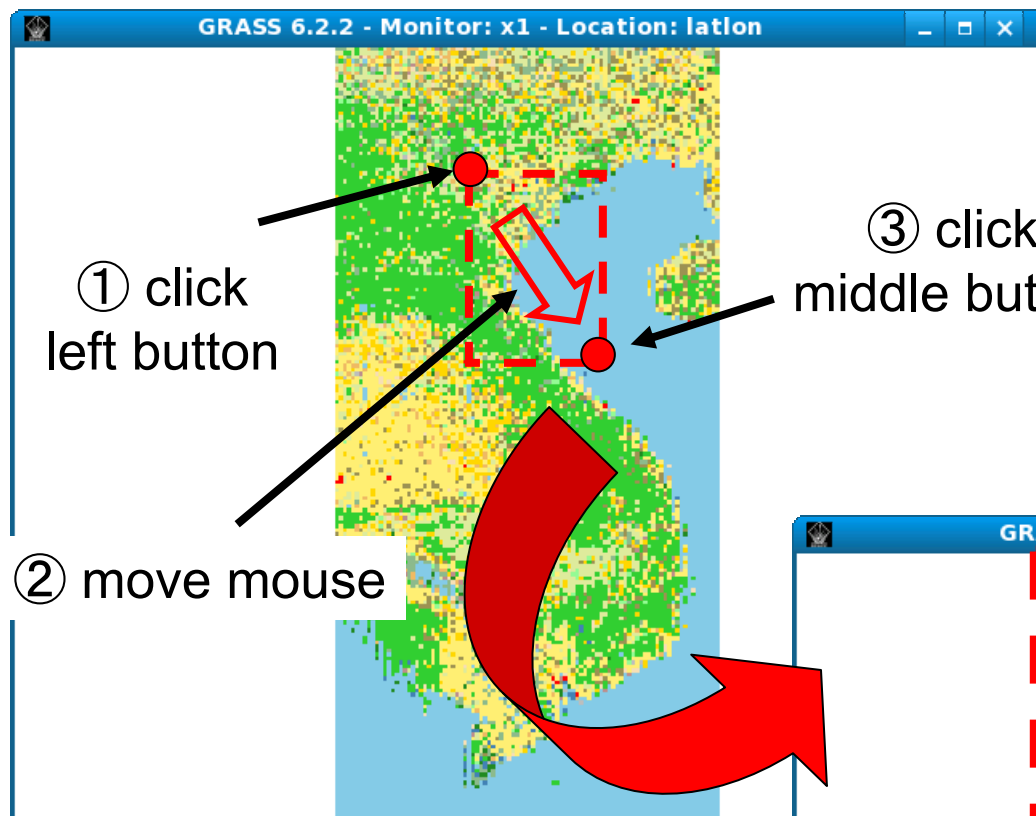
```
> d.zoom
```

d.zoom allows the user to change the current geographic region settings interactively, with a mouse.



```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.zoom
1 raster, 0 vector

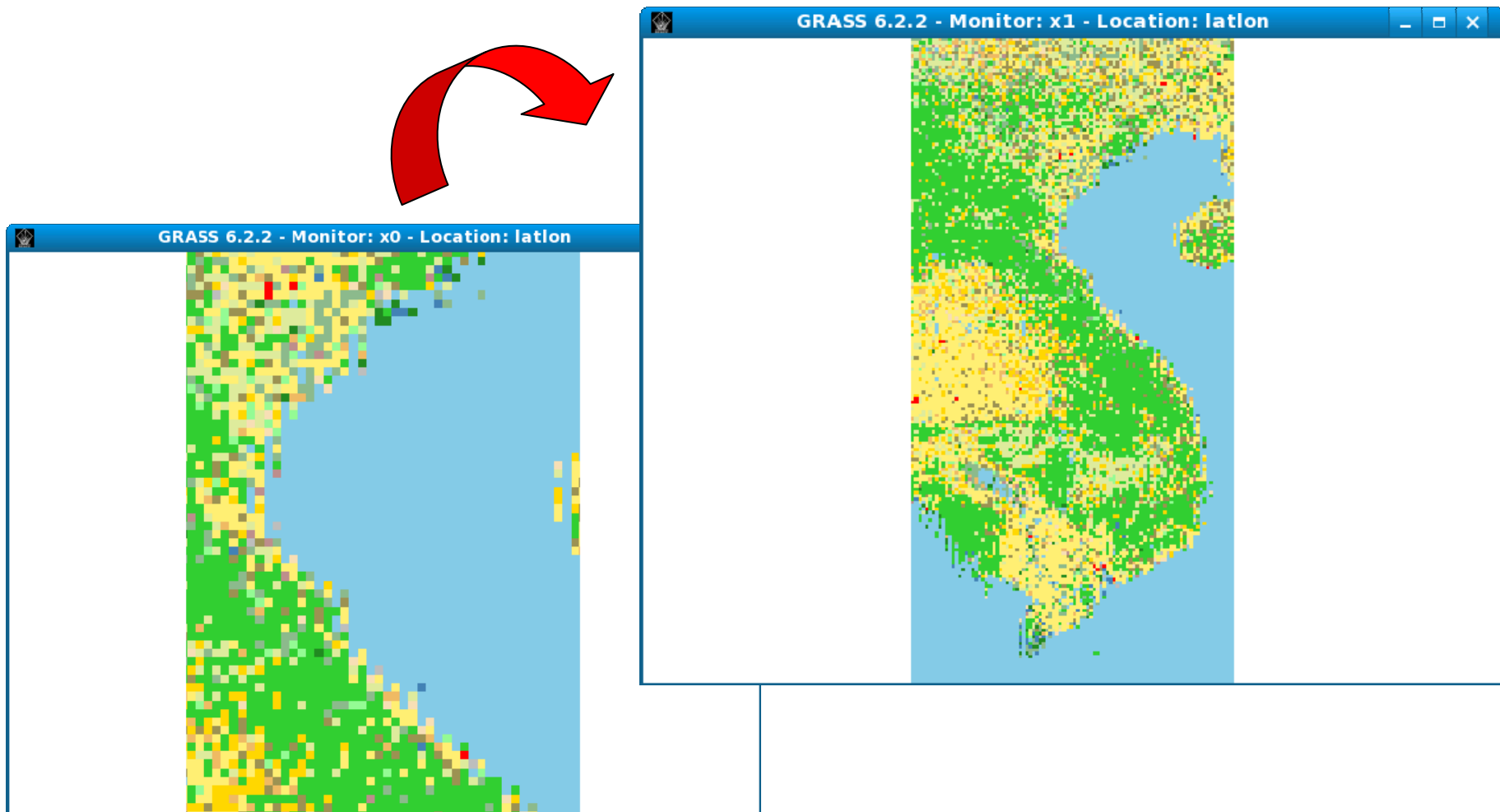
Buttons:
Left: 1. corner
Middle: Unzoom
Right: Quit
```



- Return to the previous zoom

```
> d.zoom -r
```

`-r` : space



- Reset the region

```
> g.region  $\square$ w = P  $\square$ e = Q  $\square$ n = R  $\square$ s = S
```

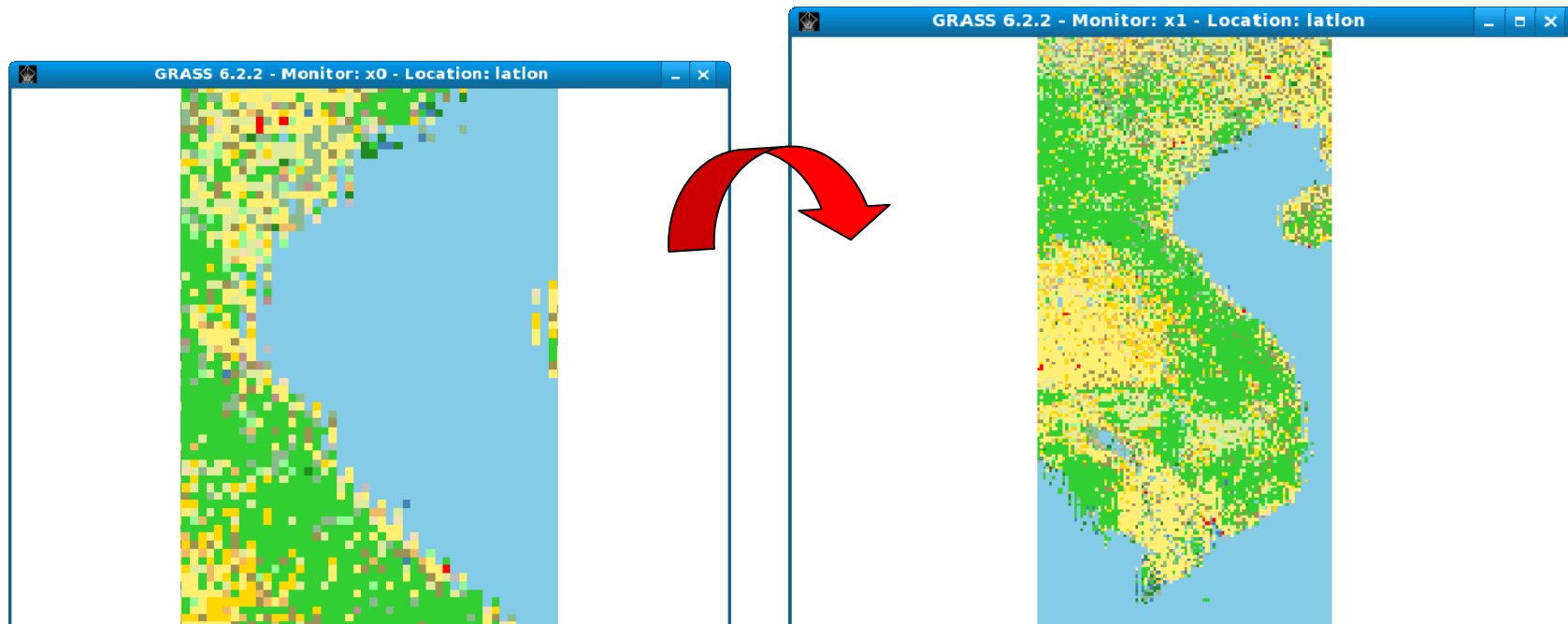
\square : space

P = 102 (value of western edge)

Q = 110 (value of eastern edge)

R = 24 (value of northern edge)

S = 8 (value of southern edge)

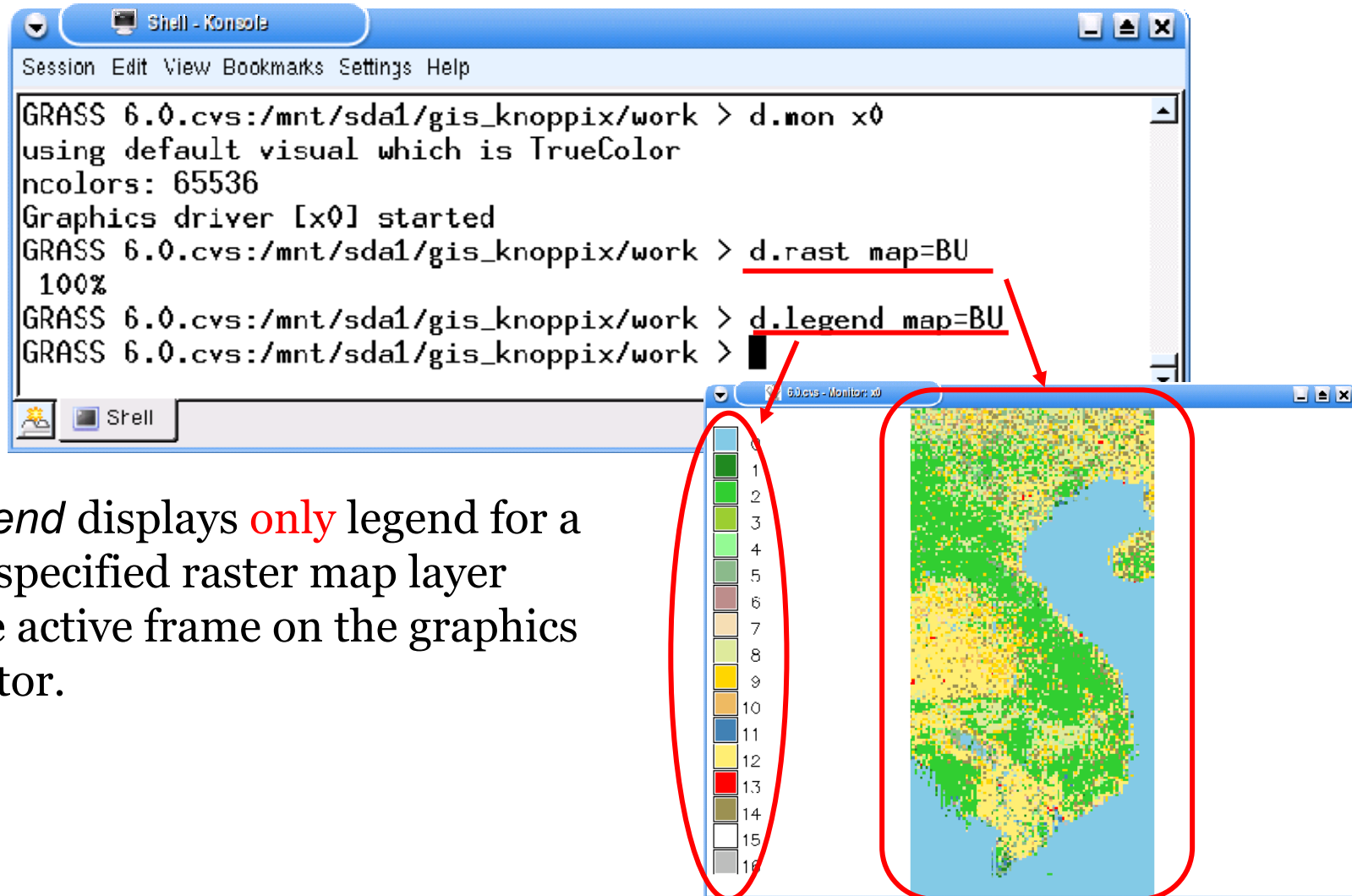


4.6 Display legend

> d.legend map = A

A = (file name)

: space

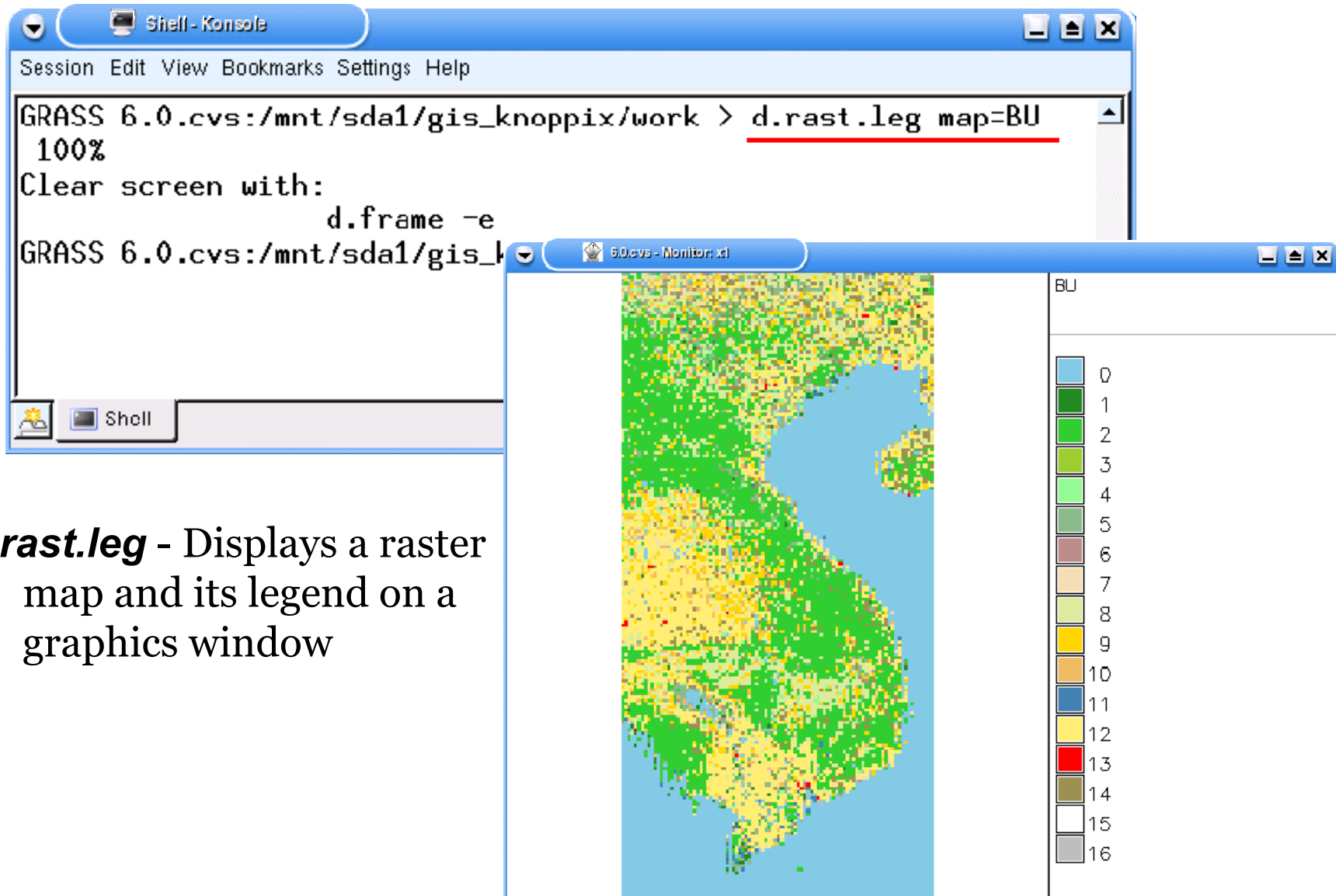


d.legend displays **only** legend for a user-specified raster map layer in the active frame on the graphics monitor.

> d.rast.legend map = A

A = (file name)

: space



d.rast.legend - Displays a raster map and its legend on a graphics window

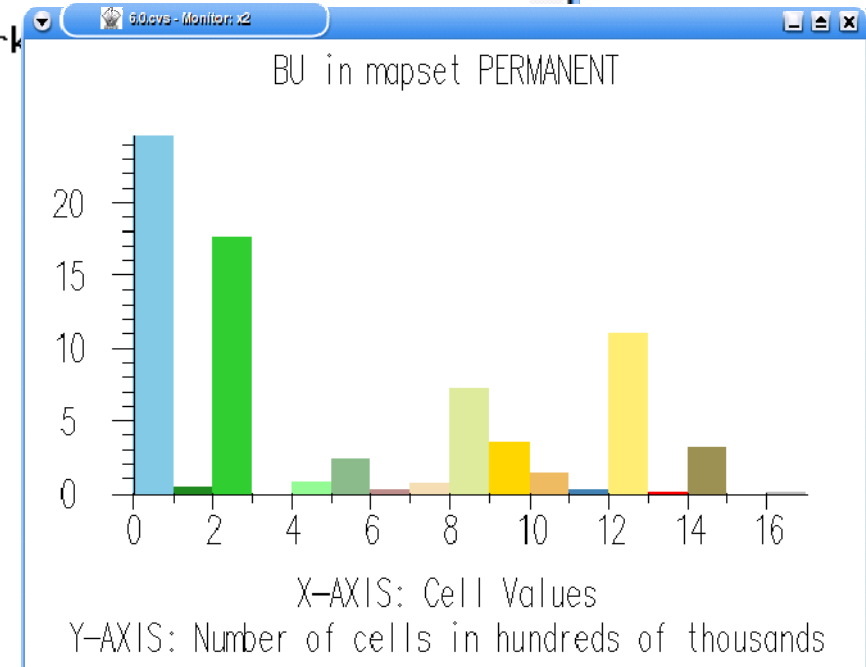
4.7 Display histogram

> d.histogram map = A A = (file name) : space

```
Shell - Konsole
Session Edit View Bookmarks Settings Help
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.mon x2
using default visual which is TrueColor
ncolors: 65536
Graphics driver [x2] started
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.histogram map=BU
r.stats: 100%
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work
```

d.histogram displays the category-value distribution for a user-specified raster map layer, in the form of a bar chart or a pie chart.

The display will be displayed in the active display frame on the graphics monitor, using the colors in the raster map layer's color table. The program determines the raster file's category value distribution by counting cells.

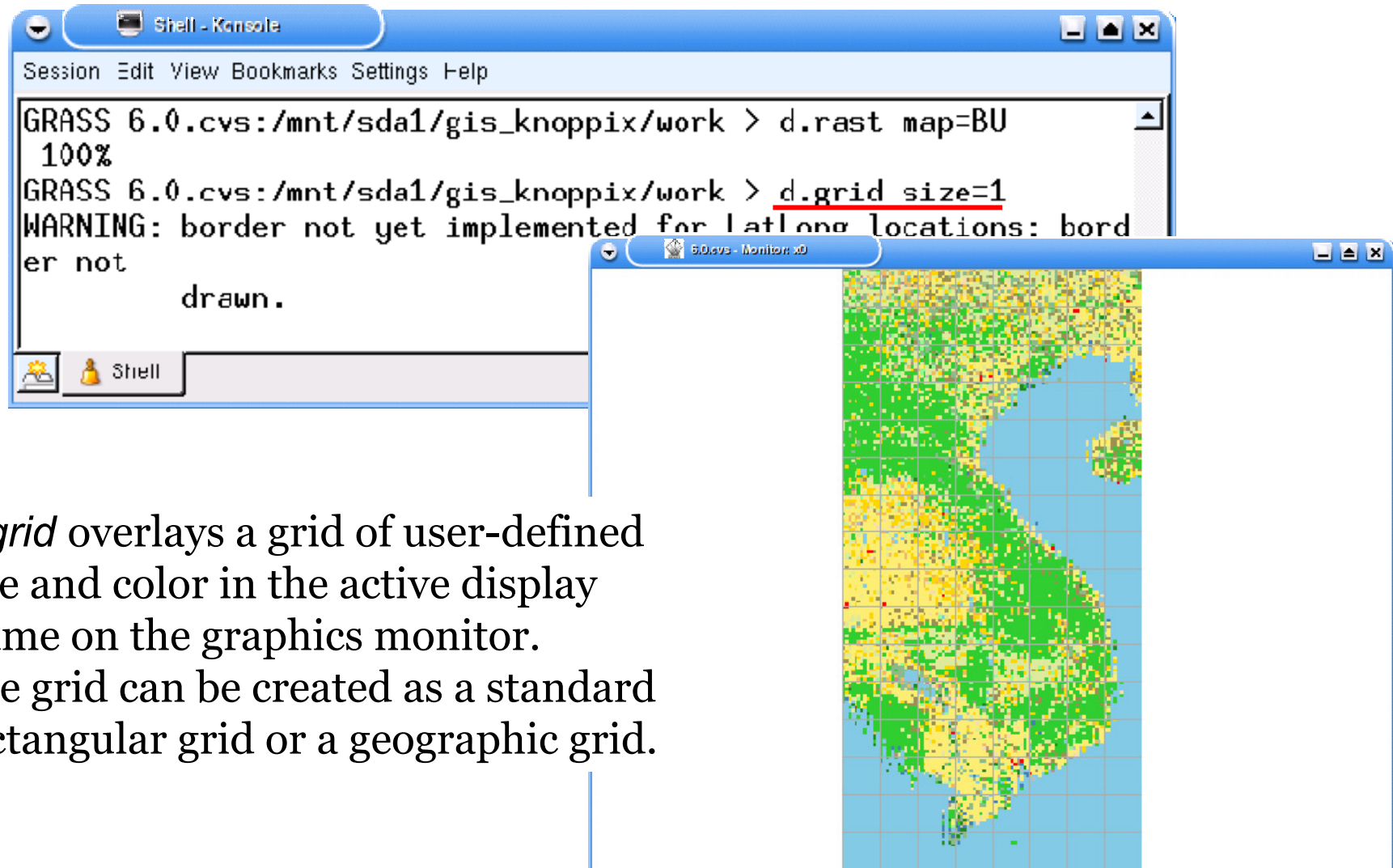


4.8 Display grid line

```
> d.grid  $\square$  size = A
```

A = (size of grid)

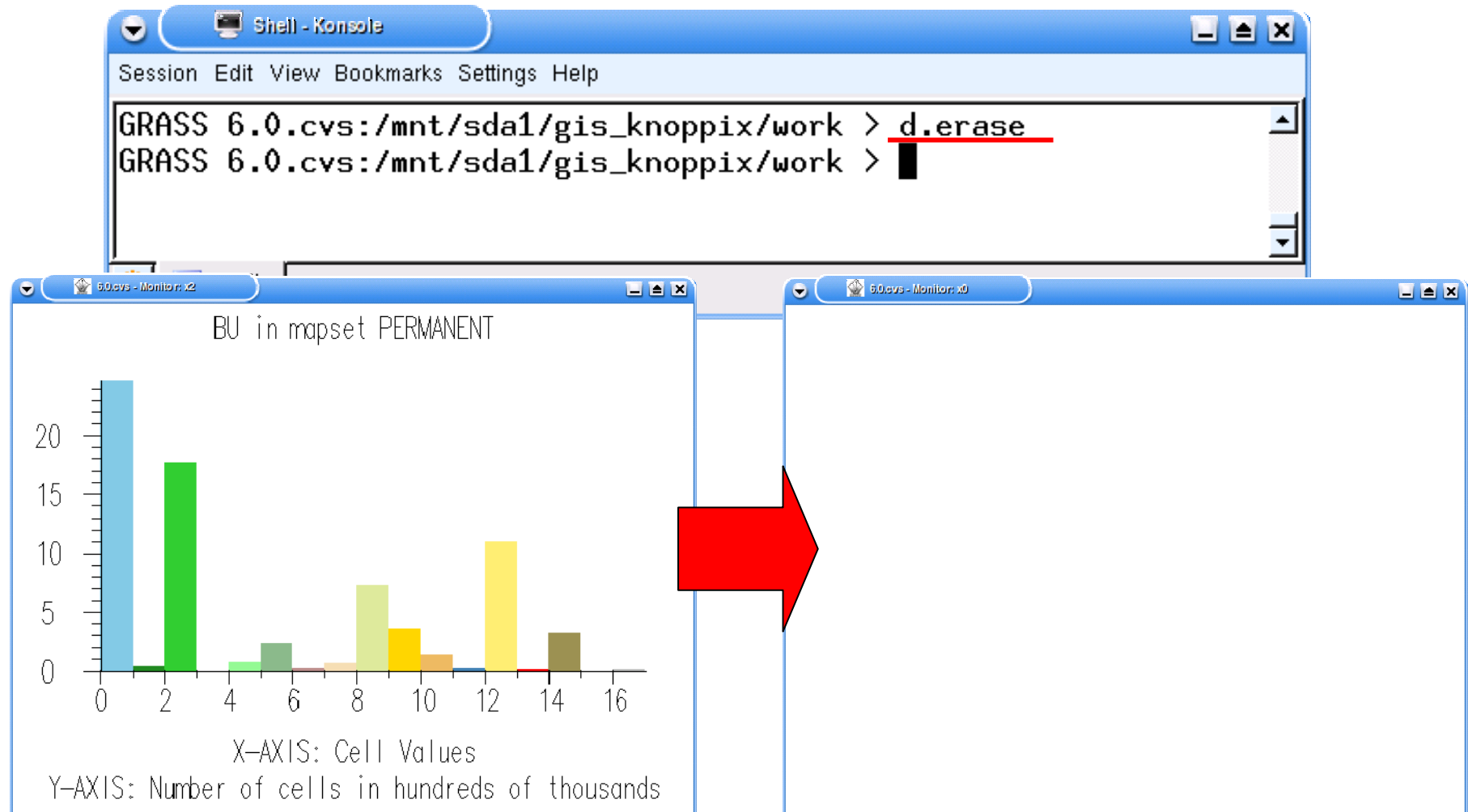
\square : space



d.grid overlays a grid of user-defined size and color in the active display frame on the graphics monitor. The grid can be created as a standard rectangular grid or a geographic grid.

4.9 Erase the contents from active display

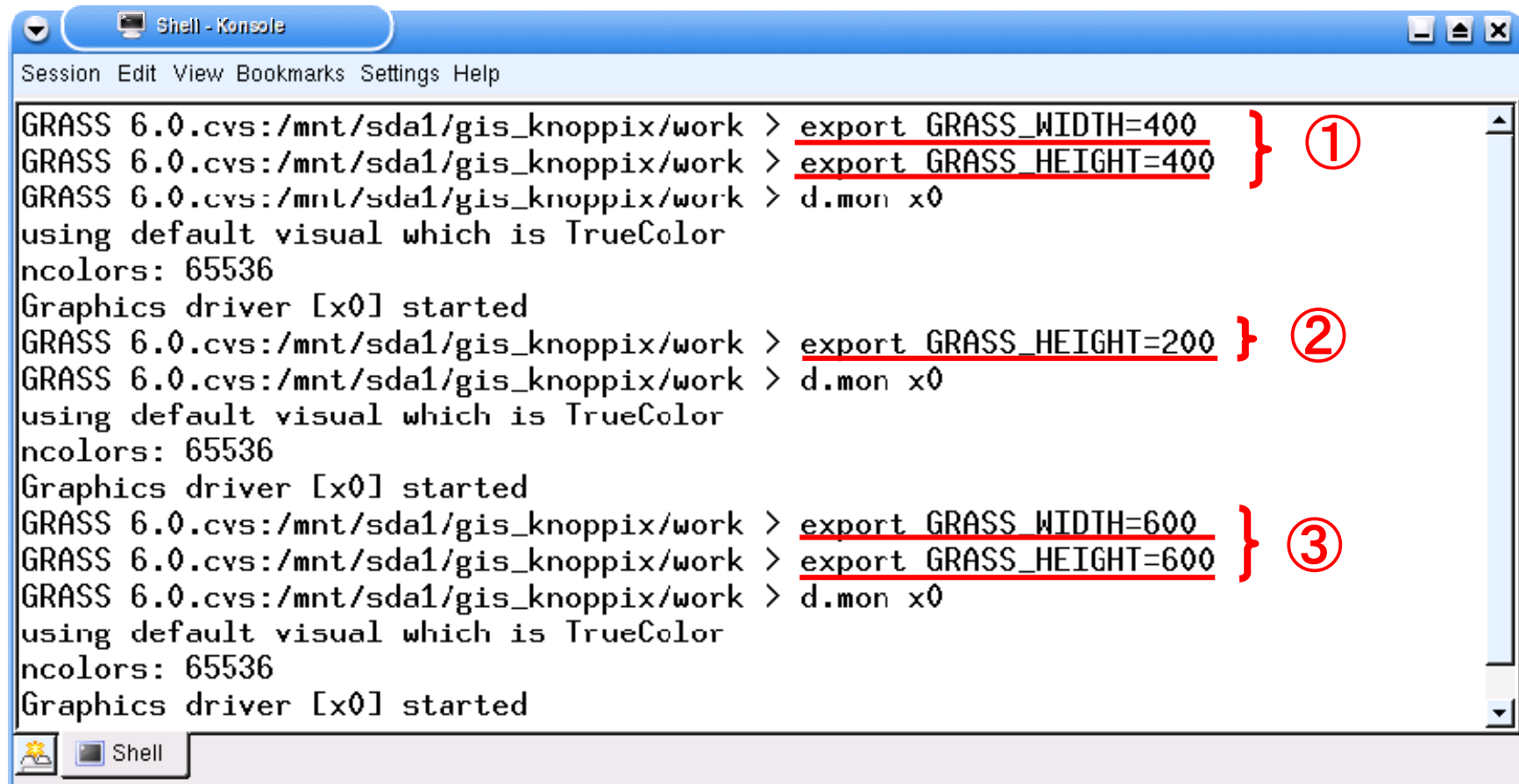
> d.erase



4.10 Change monitor size

```
> export GRASS_WIDTH = "###"
> export GRASS_HEIGHT = "###"
```

: space



The screenshot shows a terminal window titled "Shell - Konsole" with a menu bar (Session, Edit, View, Bookmarks, Settings, Help). The terminal output shows three sequential steps to change the GRASS monitor size, each starting from the prompt "GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work".

```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > export GRASS_WIDTH=400
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > export GRASS_HEIGHT=400
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.mon x0
using default visual which is TrueColor
ncolors: 65536
Graphics driver [x0] started

GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > export GRASS_HEIGHT=200
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.mon x0
using default visual which is TrueColor
ncolors: 65536
Graphics driver [x0] started

GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > export GRASS_WIDTH=600
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > export GRASS_HEIGHT=600
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.mon x0
using default visual which is TrueColor
ncolors: 65536
Graphics driver [x0] started
```

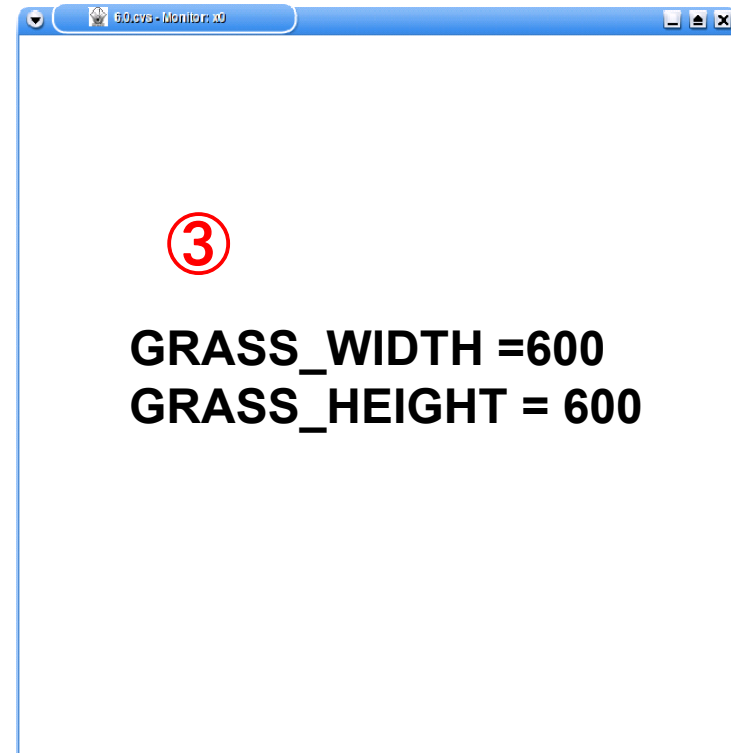
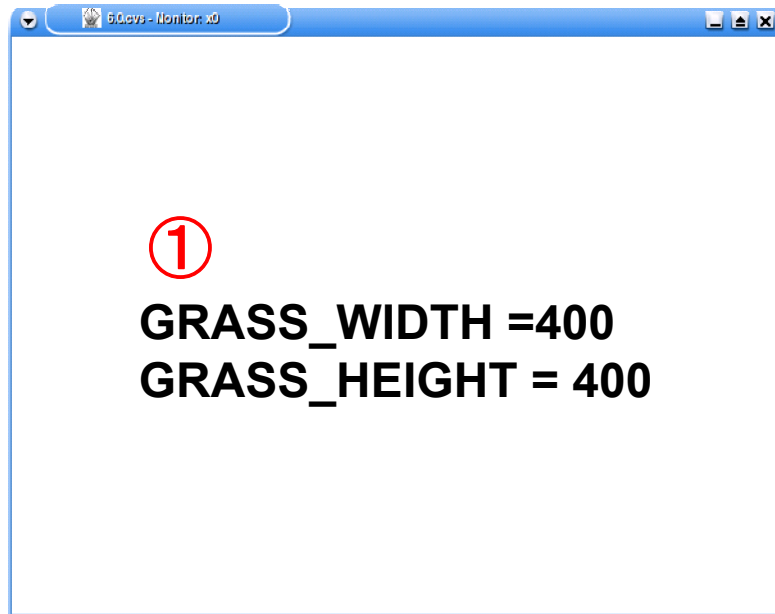
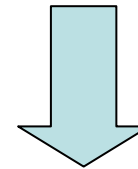
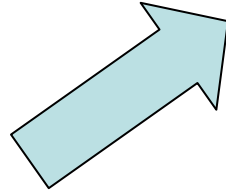
Red annotations are present in the terminal window:

- A red closing curly brace "}" and a circled number "1" are next to the first two export commands.
- A red closing curly brace "}" and a circled number "2" are next to the second two export commands.
- A red closing curly brace "}" and a circled number "3" are next to the third two export commands.

The terminal window has a taskbar at the bottom with icons for a desktop, a shell, and a taskbar.

**GRASS_WIDTH =400
GRASS_HEIGHT = 200**

②



Section 2. Data Processing in Latitude-Longitude Coordinate System

5. Import vector data

5.1 Import Vector data (ESRI shapefile)

```
> v.external dsn = " A " layer = " B " output = " C "      : space
```

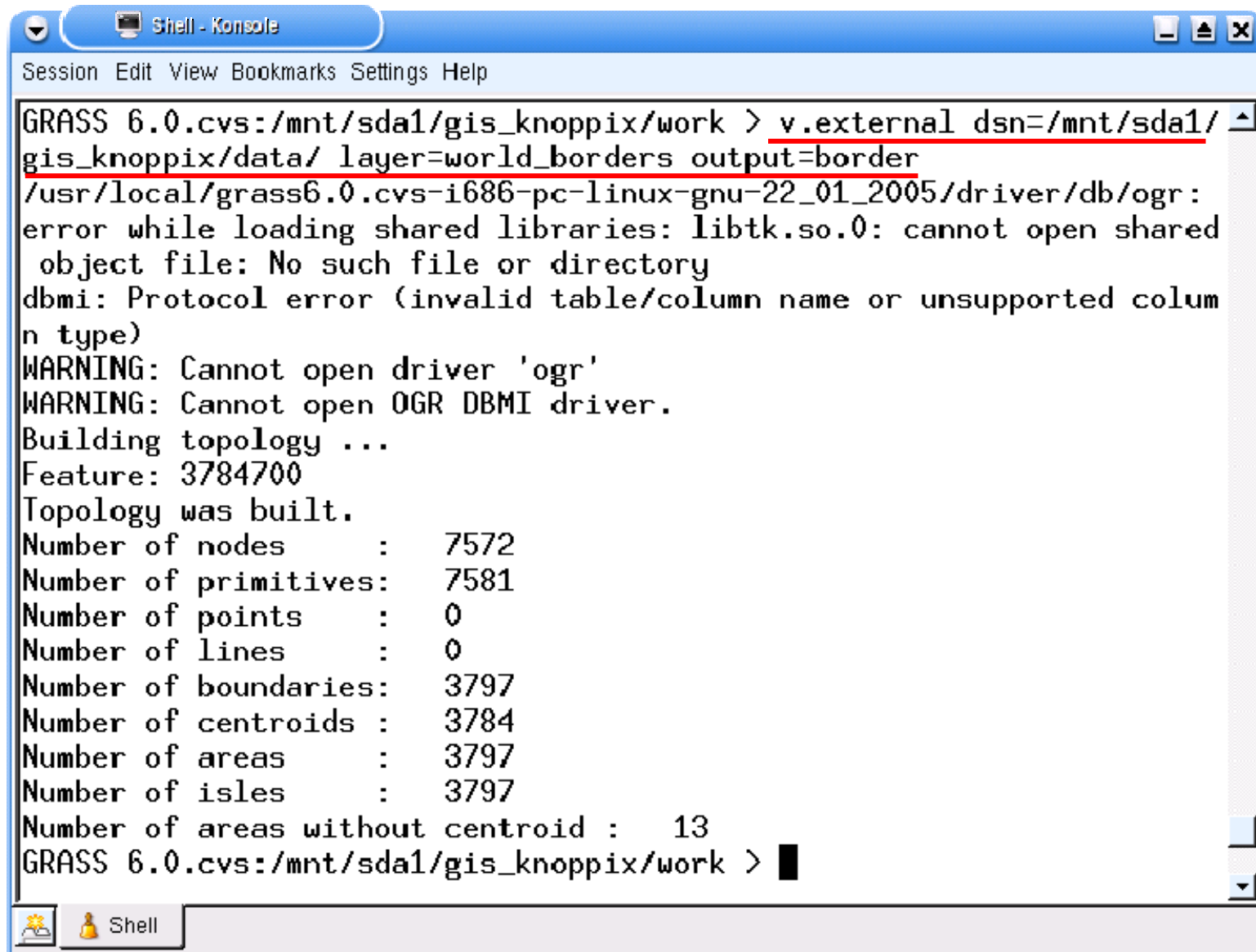
A = directory containing shapefiles(/mnt/sda1/gis_knoppix/data)

B = shapefile name

C = output vector name

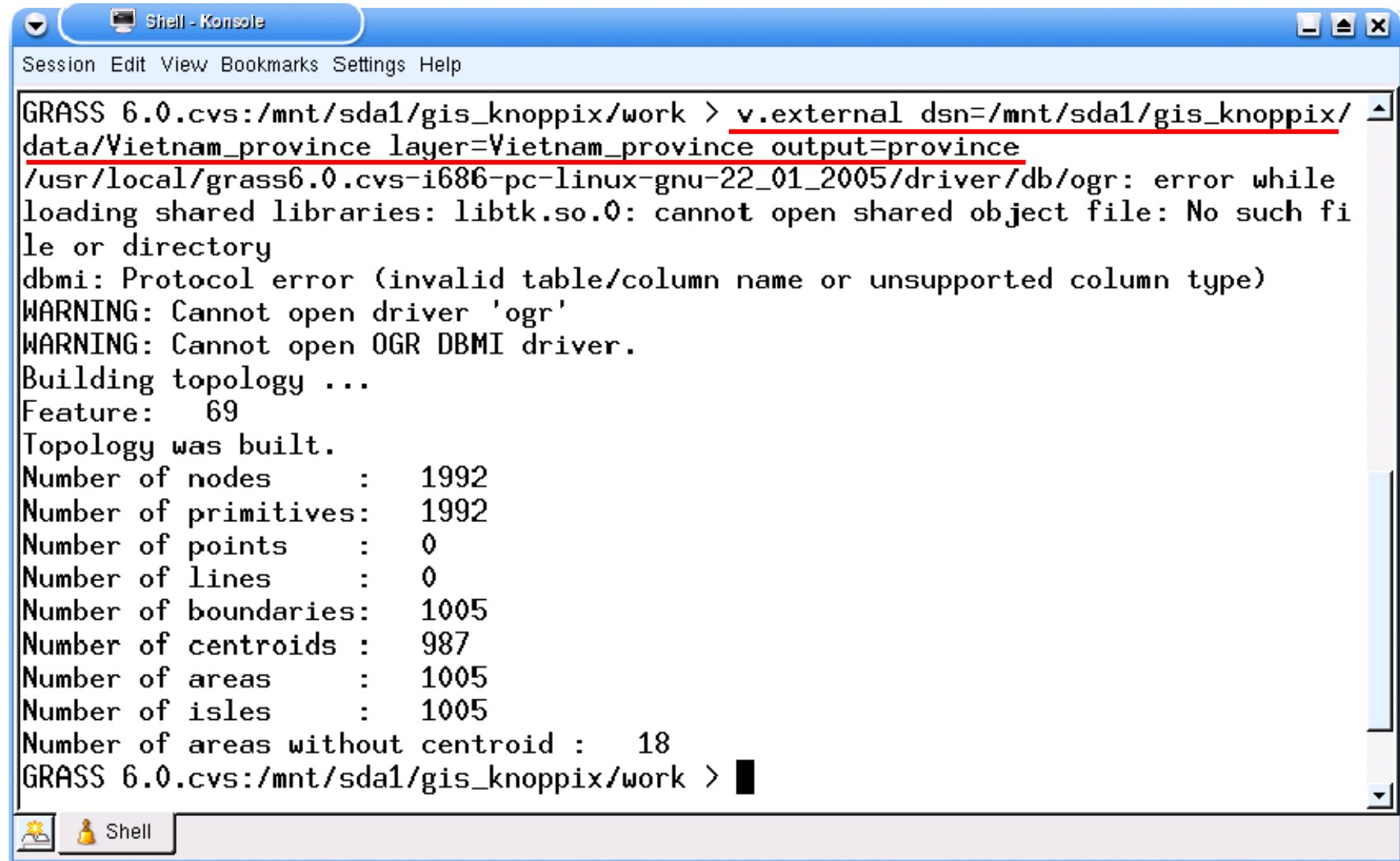
v.external creates new vector as a read only layer

Import vector data of national borders



```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > v.external dsn=/mnt/sda1/  
gis_knoppix/data/ layer=world_borders output=border  
/usr/local/grass6.0.cvs-i686-pc-linux-gnu-22_01_2005/driver/db/ogr:  
error while loading shared libraries: libtk.so.0: cannot open shared  
object file: No such file or directory  
dbmi: Protocol error (invalid table/column name or unsupported colum  
n type)  
WARNING: Cannot open driver 'ogr'  
WARNING: Cannot open OGR DBMI driver.  
Building topology ...  
Feature: 3784700  
Topology was built.  
Number of nodes      :    7572  
Number of primitives:    7581  
Number of points     :      0  
Number of lines      :      0  
Number of boundaries:    3797  
Number of centroids  :    3784  
Number of areas      :    3797  
Number of isles      :    3797  
Number of areas without centroid :    13  
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > █
```

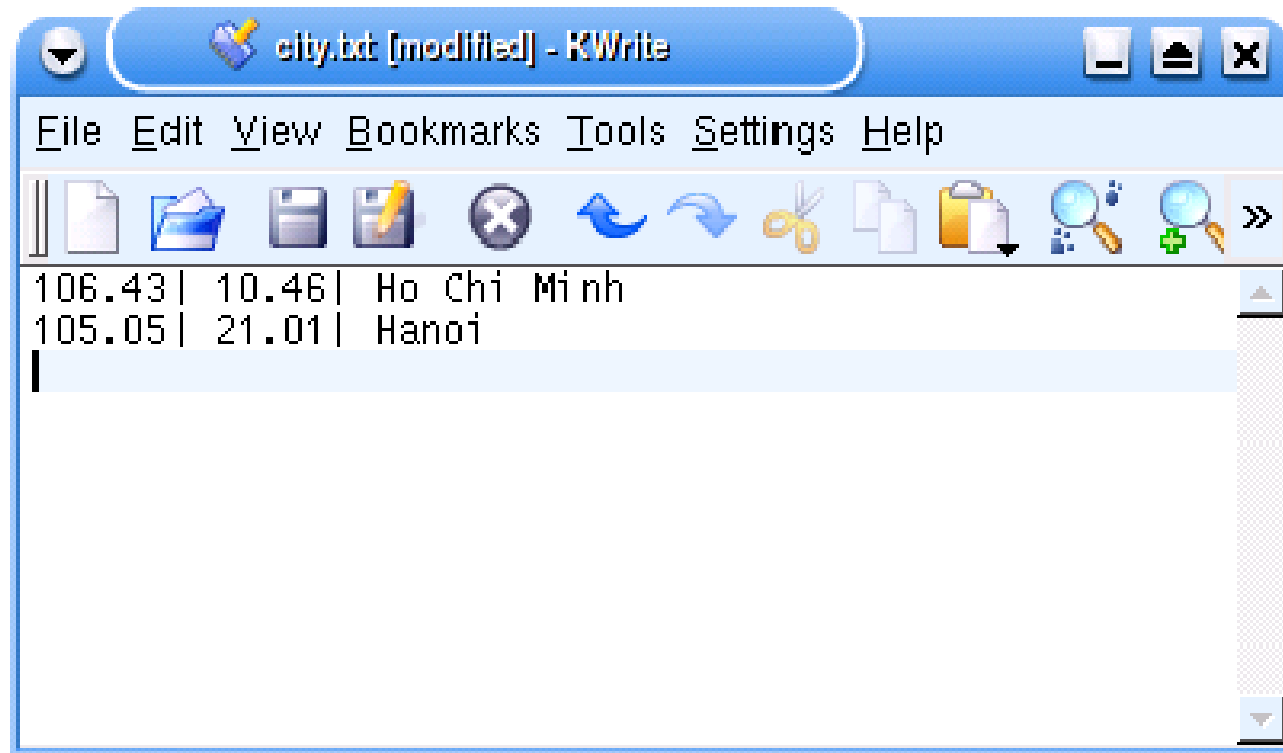

Import vector data of administrative district borders



```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > v.external dsn=/mnt/sda1/gis_knoppix/  
data/Vietnam_province layer=Vietnam_province output=province  
/usr/local/grass6.0.cvs-i686-pc-linux-gnu-22_01_2005/driver/db/ogr: error while  
loading shared libraries: libtk.so.0: cannot open shared object file: No such fi  
le or directory  
dbmi: Protocol error (invalid table/column name or unsupported column type)  
WARNING: Cannot open driver 'ogr'  
WARNING: Cannot open OGR DBMI driver.  
Building topology ...  
Feature: 69  
Topology was built.  
Number of nodes      : 1992  
Number of primitives: 1992  
Number of points     : 0  
Number of lines      : 0  
Number of boundaries: 1005  
Number of centroids  : 987  
Number of areas      : 1005  
Number of isles      : 1005  
Number of areas without centroid : 18  
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > █
```

5.2 Import point data

5.2.1 Create a vector data in ASCII format using text editor



use “ | ” as separator

⇒ save as “ city.txt ” in “ work ” directory

5.2.2 Convert vector data in ASCII format to vector data in binary format

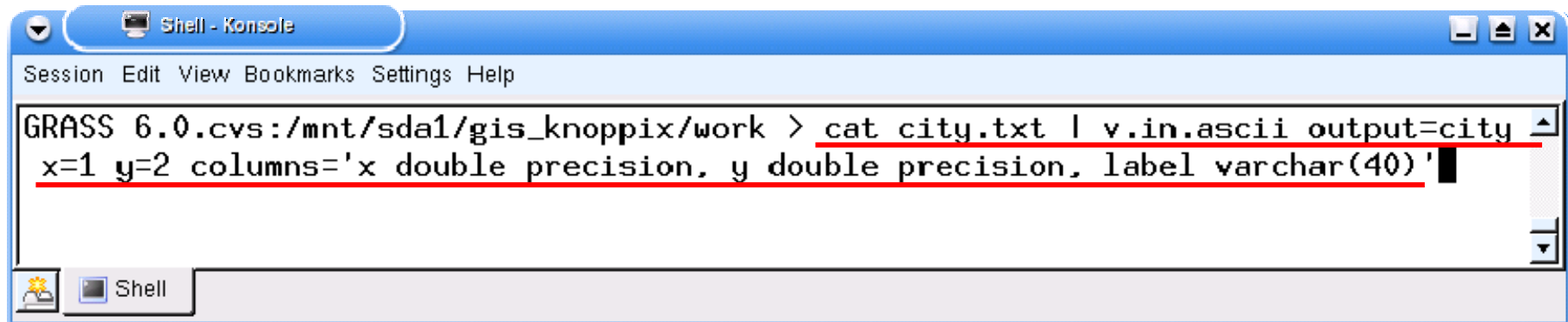
```
> cat " A " | v.in.ascii output = " B " x=1 y=2 columns = 'x double precision,  
y double precision, label varchar( # )'
```

A = city.txt

= 40 (in this case)

: space

v.in.ascii converts a vector map in ASCII format to a vector map in binary format.

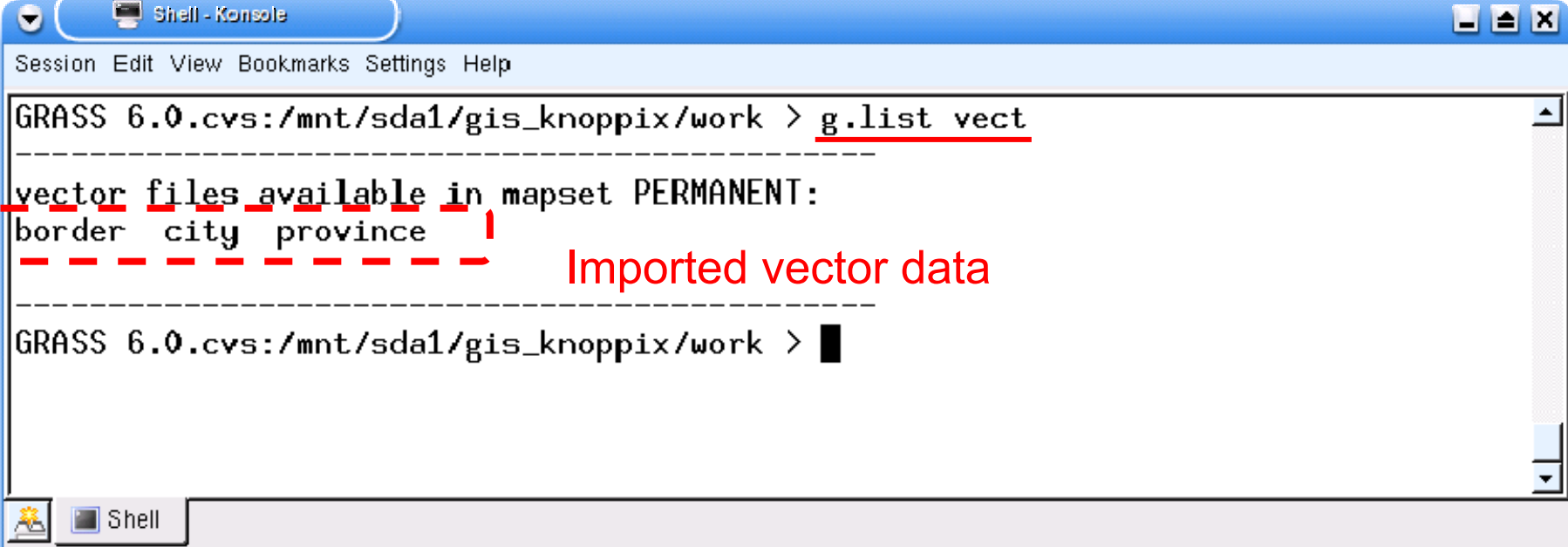


```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > cat city.txt | v.in.ascii output=city  
x=1 y=2 columns='x double precision, y double precision, label varchar(40)'
```

5.3 List imported vector data

> g.list type = vect

: space



The screenshot shows a terminal window titled "Shell - Konsole" with a menu bar (Session, Edit, View, Bookmarks, Settings, Help). The command prompt shows the user is in the directory "/mnt/sda1/gis_knoppix/work". The command g.list vect has been entered. The output is "vector files available in mapset PERMANENT:" followed by a list of files: "border", "city", and "province". These three file names are enclosed in a red dashed box. To the right of this box, the text "Imported vector data" is written in red. Below the list, the command prompt is shown again with a cursor.

```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > g.list vect
-----
vector files available in mapset PERMANENT:
border  city  province
-----
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > █
```

Section 2. Data Processing in Latitude-Longitude Coordinate System

6. Display vector data

Step 0. Import raster data

Step 1. Launch monitor

```
> d.mon  x#
```

= 0, 1, 2, ..., 6

: space

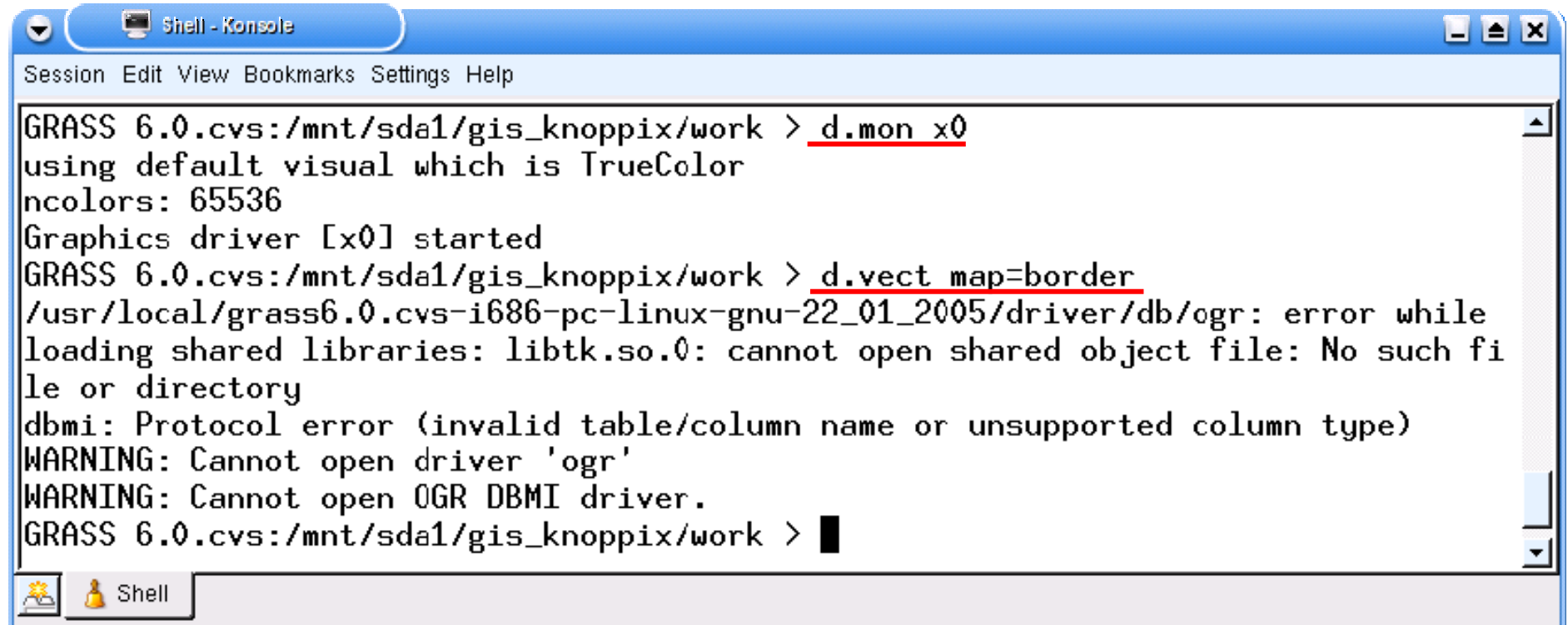
Step 2. Display vector

```
> d.vect  map = A
```

A = (input file name)

: space

6.1 Display vector data (1)



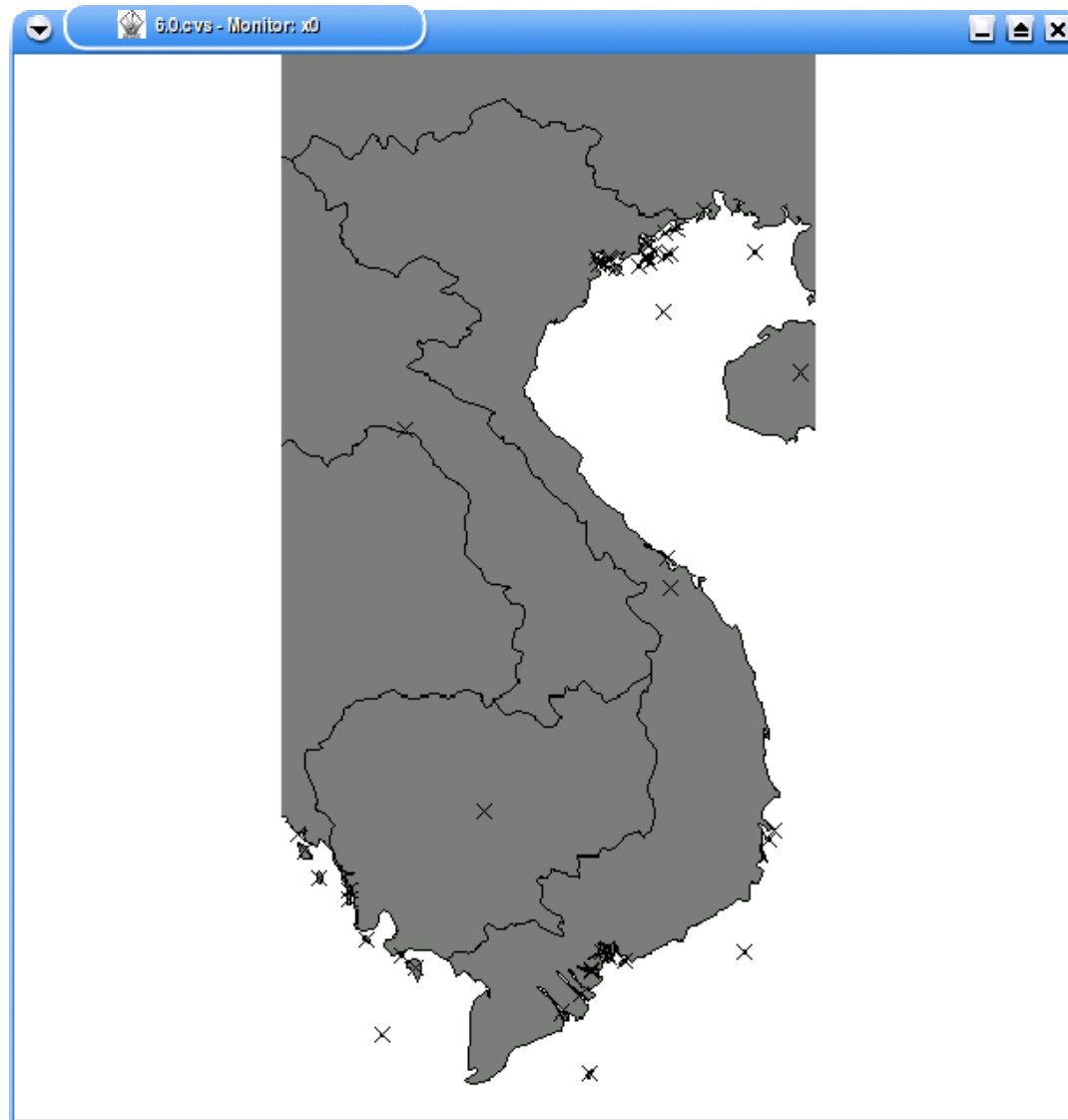
The screenshot shows a terminal window titled "Shell - Konsole". The window has a menu bar with "Session", "Edit", "View", "Bookmarks", "Settings", and "Help". The terminal content shows the following sequence of commands and output:

```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.mon x0
using default visual which is TrueColor
ncolors: 65536
Graphics driver [x0] started
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.vect map=border
/usr/local/grass6.0.cvs-i686-pc-linux-gnu-22_01_2005/driver/db/ogr: error while
loading shared libraries: libtk.so.0: cannot open shared object file: No such fi
le or directory
dbmi: Protocol error (invalid table/column name or unsupported column type)
WARNING: Cannot open driver 'ogr'
WARNING: Cannot open OGR DBMI driver.
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > █
```

At the bottom of the window, there is a taskbar with two icons: a sun icon and a person icon labeled "Shell".

> d.vect map = A

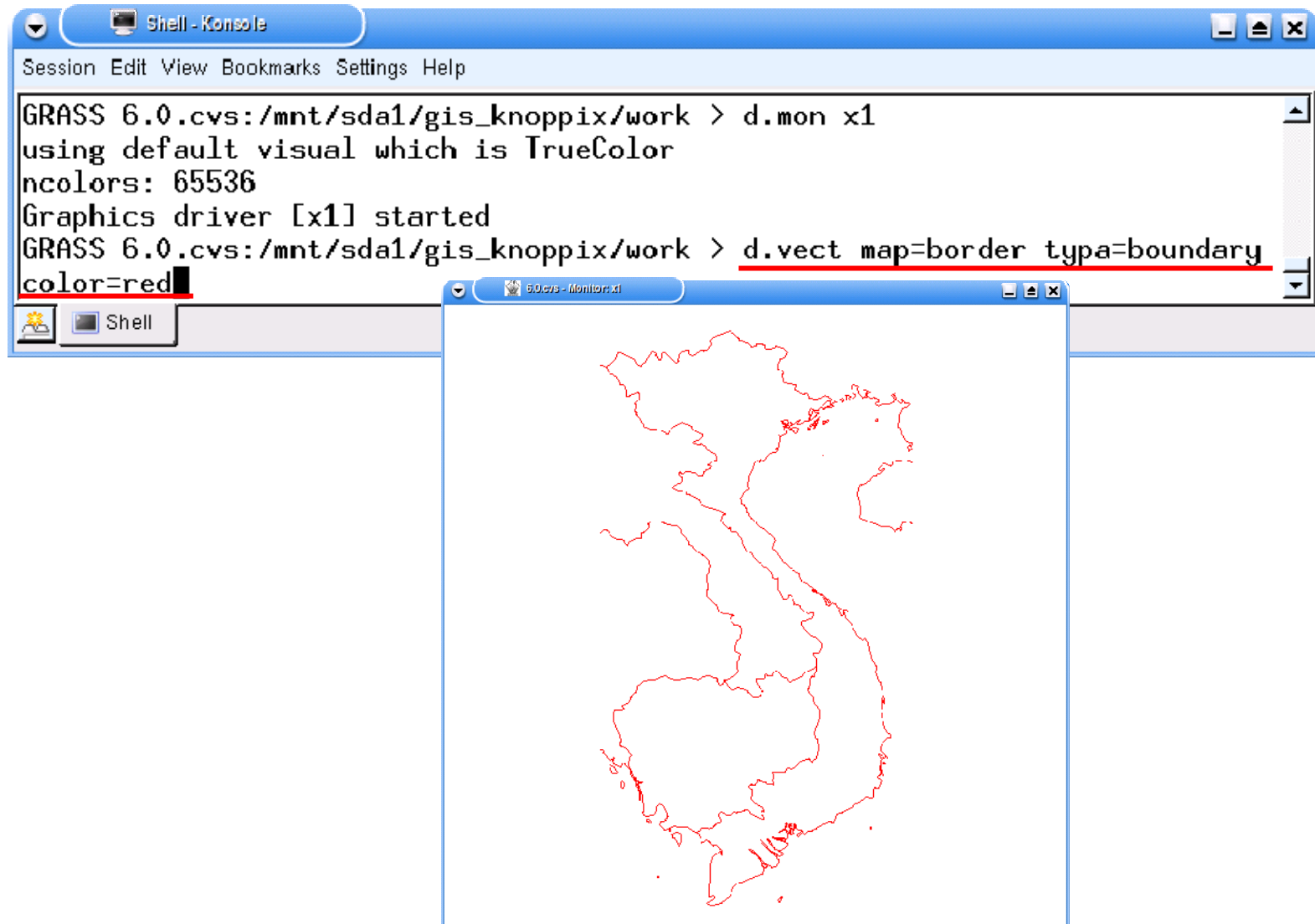
A = (input file name) : space



> d.vect map = A type = boundary color = B
 : space

A = (input file name)

B = (line color)

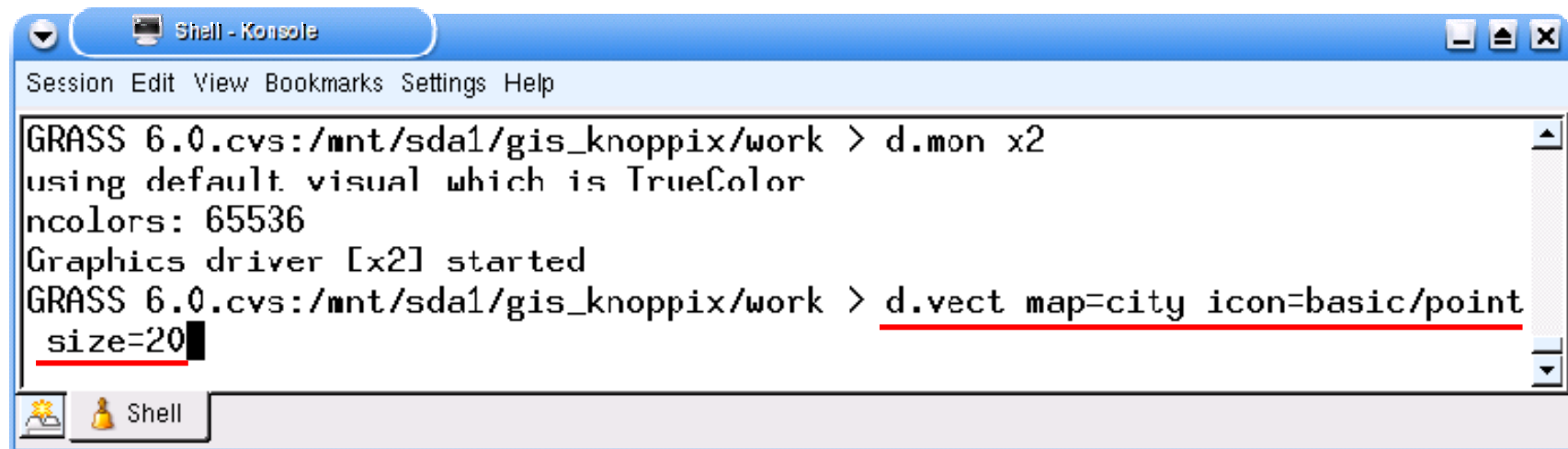


6.2 Display vector data (2) - point data -

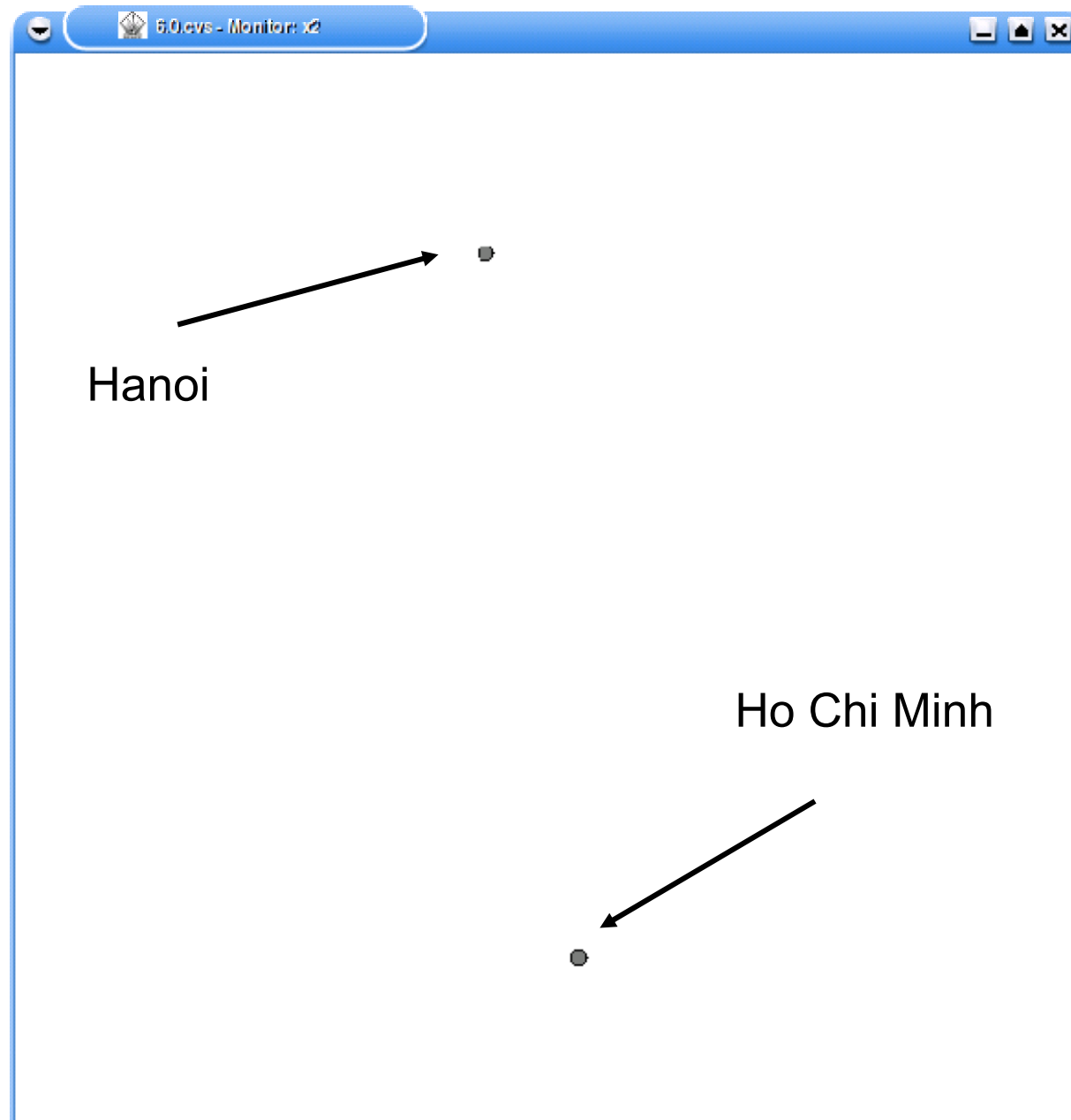
< display icon >

> d.vect `map = A icon = B size = C`

A = (input file name)
B = (point symbol option)
C = (icon size) : space



```
Shell - Konsole
Session Edit View Bookmarks Settings Help
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.mon x2
using default visual which is TrueColor
ncolors: 65536
Graphics driver [x2] started
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.vect map=city icon=basic/point
size=20
```



< display city name >

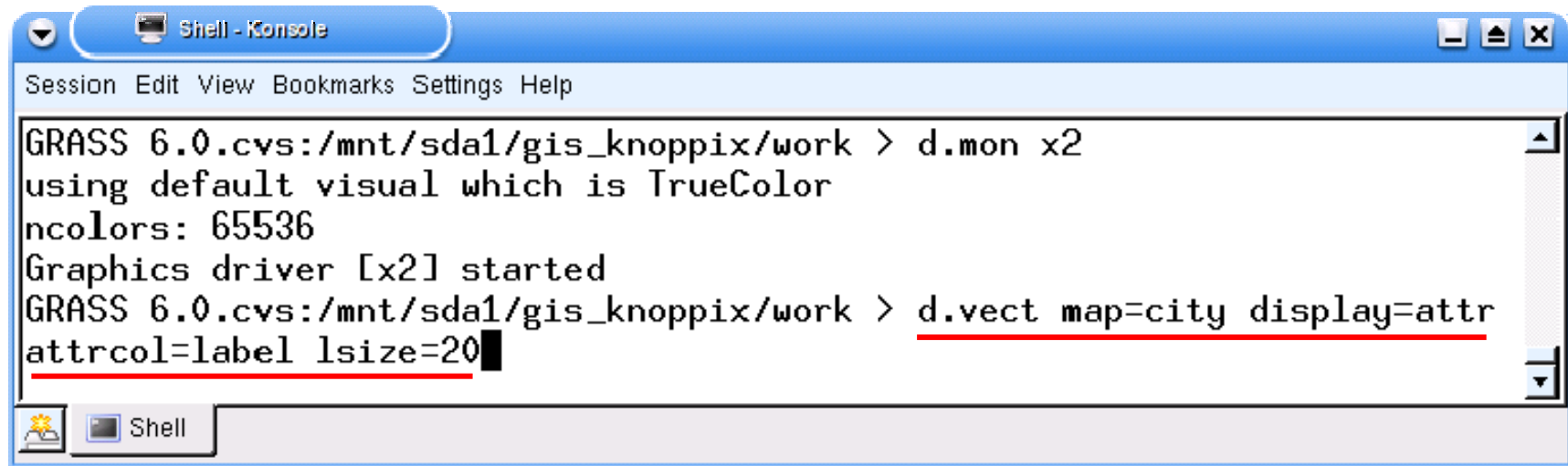
> d.vect map = A display= B attrcol = C lsize = D : space

A = (input file name)

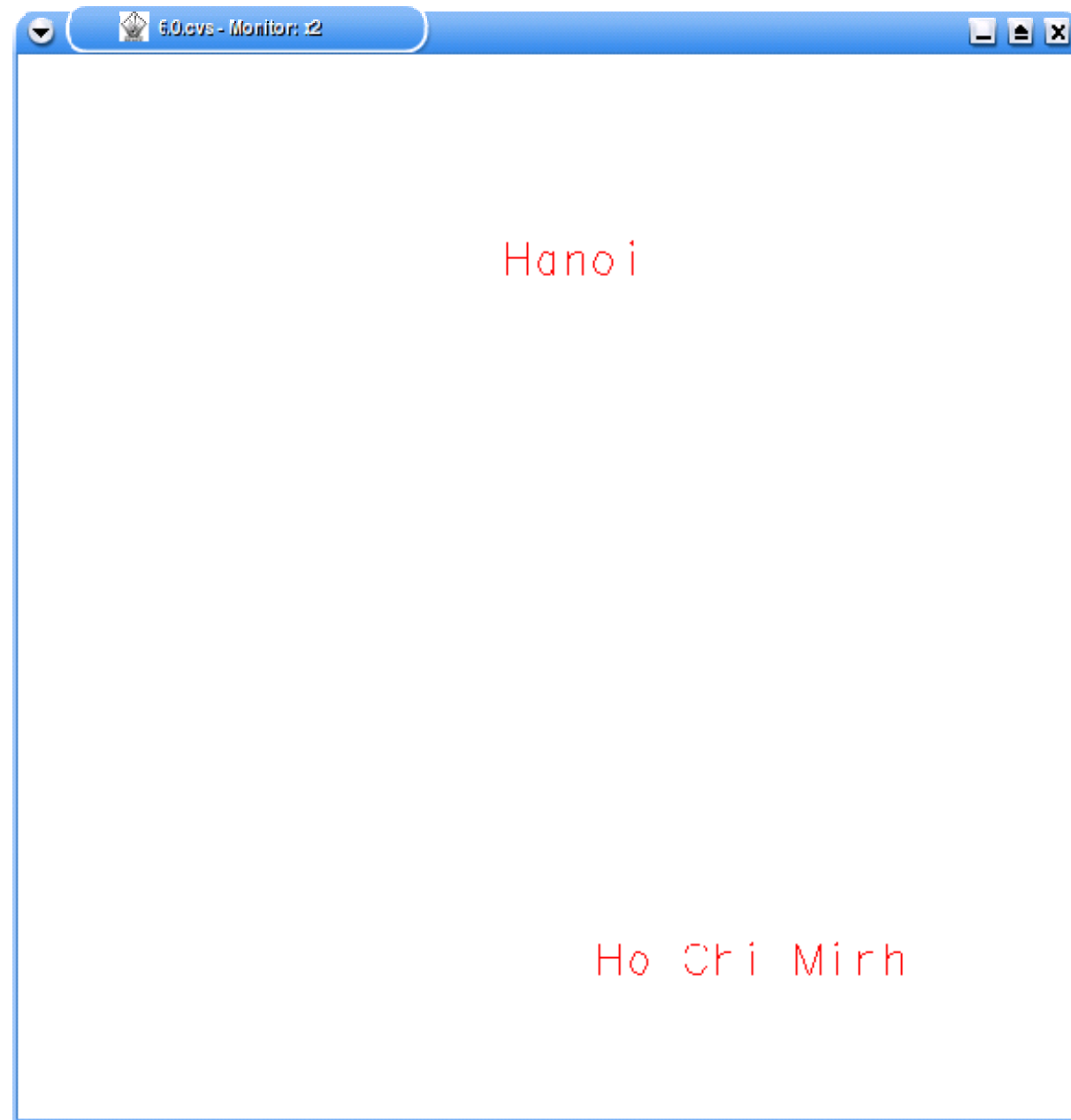
B = (display option)

C = (Name of column to be display)

D = (label size (pixels))



```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.mon x2
using default visual which is TrueColor
ncolors: 65536
Graphics driver [x2] started
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.vect map=city display=attr
attrcol=label lsize=20
```



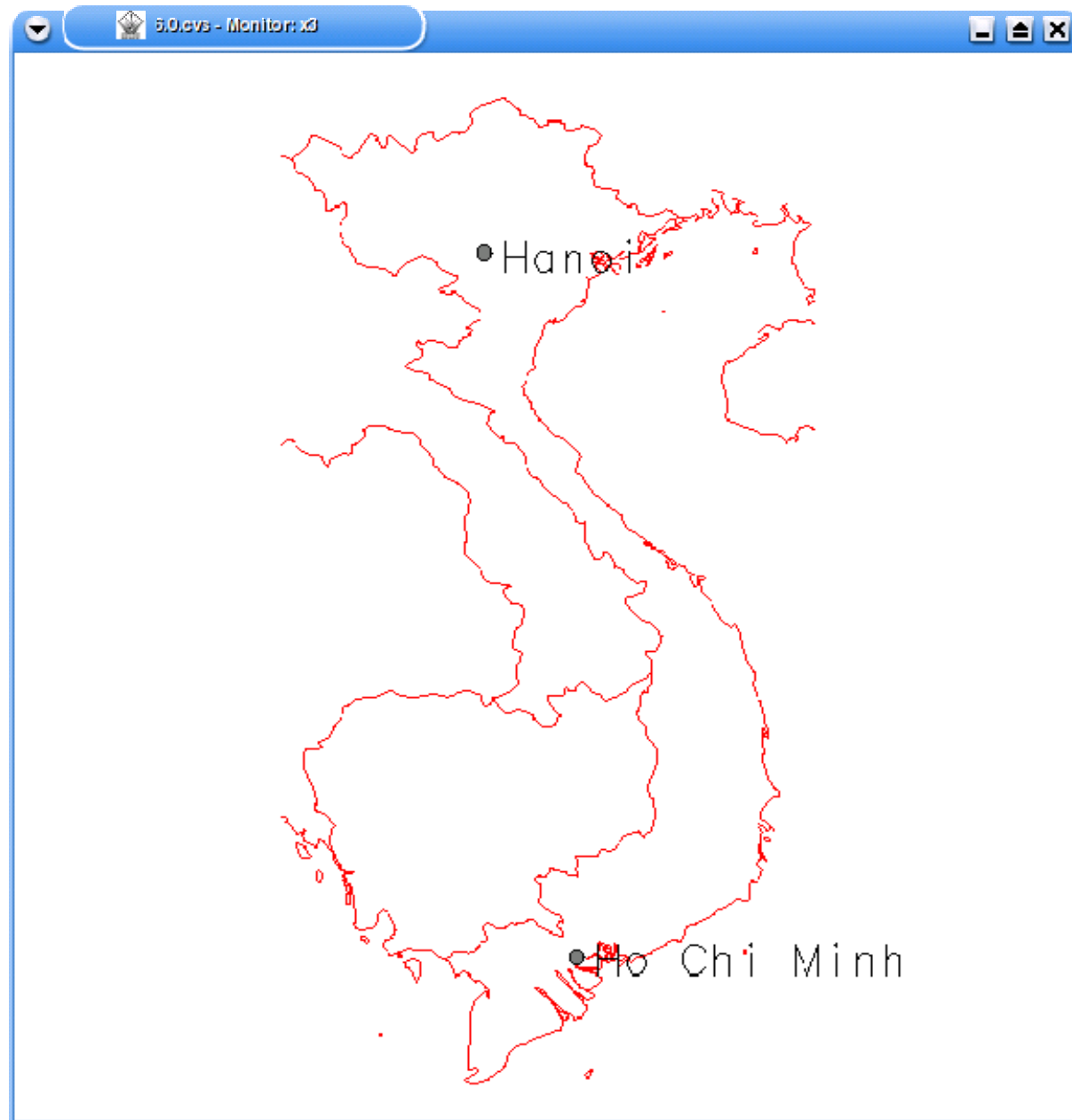
The screenshot shows a terminal window titled "Shell - Konsole" with a menu bar (Session, Edit, View, Bookmarks, Settings, Help). The terminal output is as follows:

```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.mon x3
using default visual which is TrueColor
ncolors: 65536
Graphics driver [x3] started
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.vect map=border type=boundary
color=red
/usr/local/grass6.0.cvs-i686-pc-linux-gnu-22_01_2005/driver/db/ogr: error
while loading shared libraries: libtk.so.0: cannot open shared object file
: No such file or directory
dbmi: Protocol error (invalid table/column name or unsupported column type
)
WARNING: Cannot open driver 'ogr'
WARNING: Cannot open OGR DBMI driver.
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.vect map=city icon=basic/poin
t_size=20
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.vect map=city display=attr at
trcol=label lsize=20 lcolor=black
```

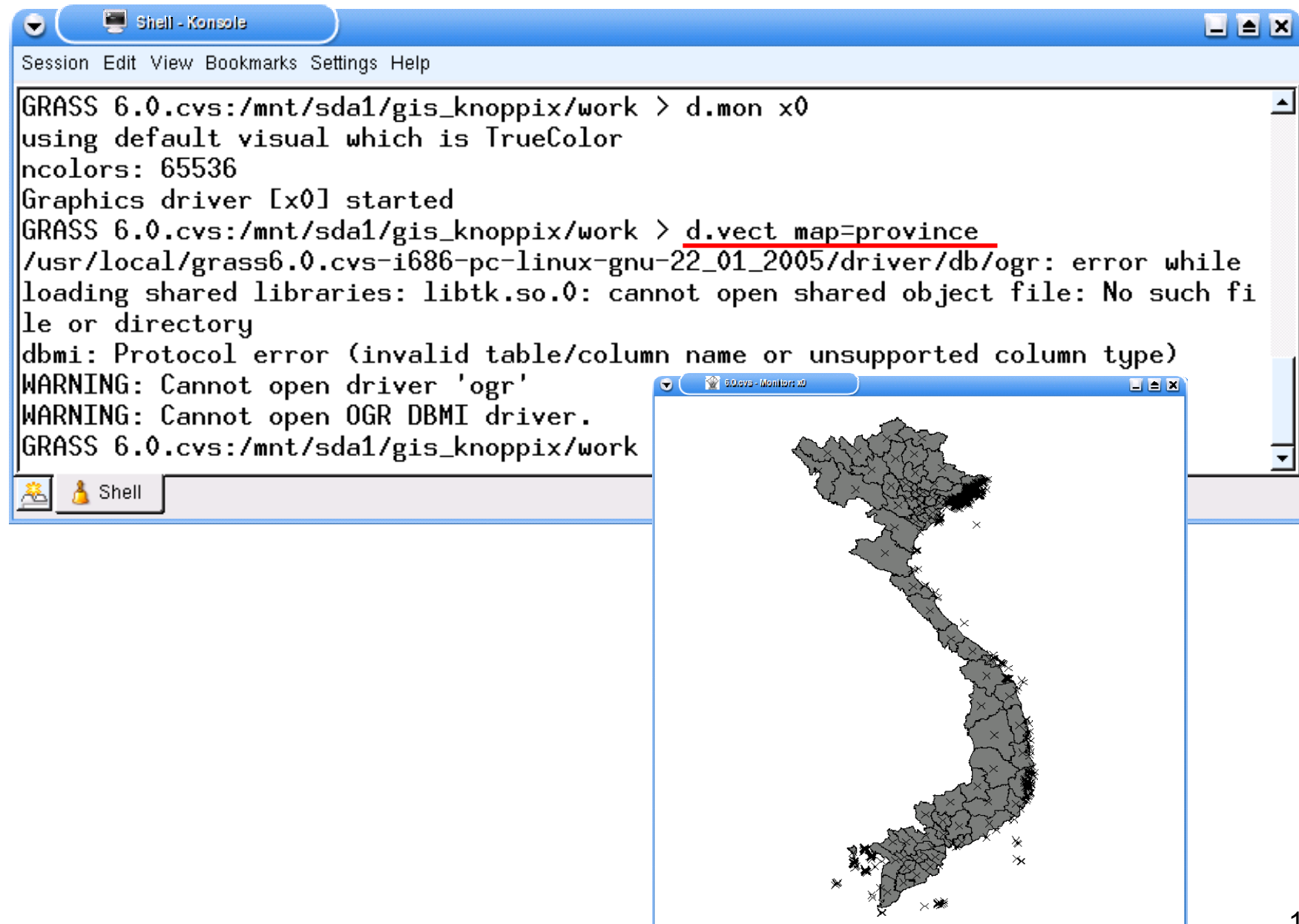
Red arrows point from the following text labels to specific parts of the commands in the terminal:

- display border** points to `d.vect map=border type=boundary color=red`.
- display symbol** points to `d.vect map=city icon=basic/point_size=20`.
- display label** points to `d.vect map=city display=attr atrcol=label lsize=20 lcolor=black`.

- ① display border
- ② display symbol
- ③ display label



6.3 Convert vector data to raster data




```
> v.to.rast [input = A [output = B [use = C
```

A = (vector file name)

B = (output raster file name)

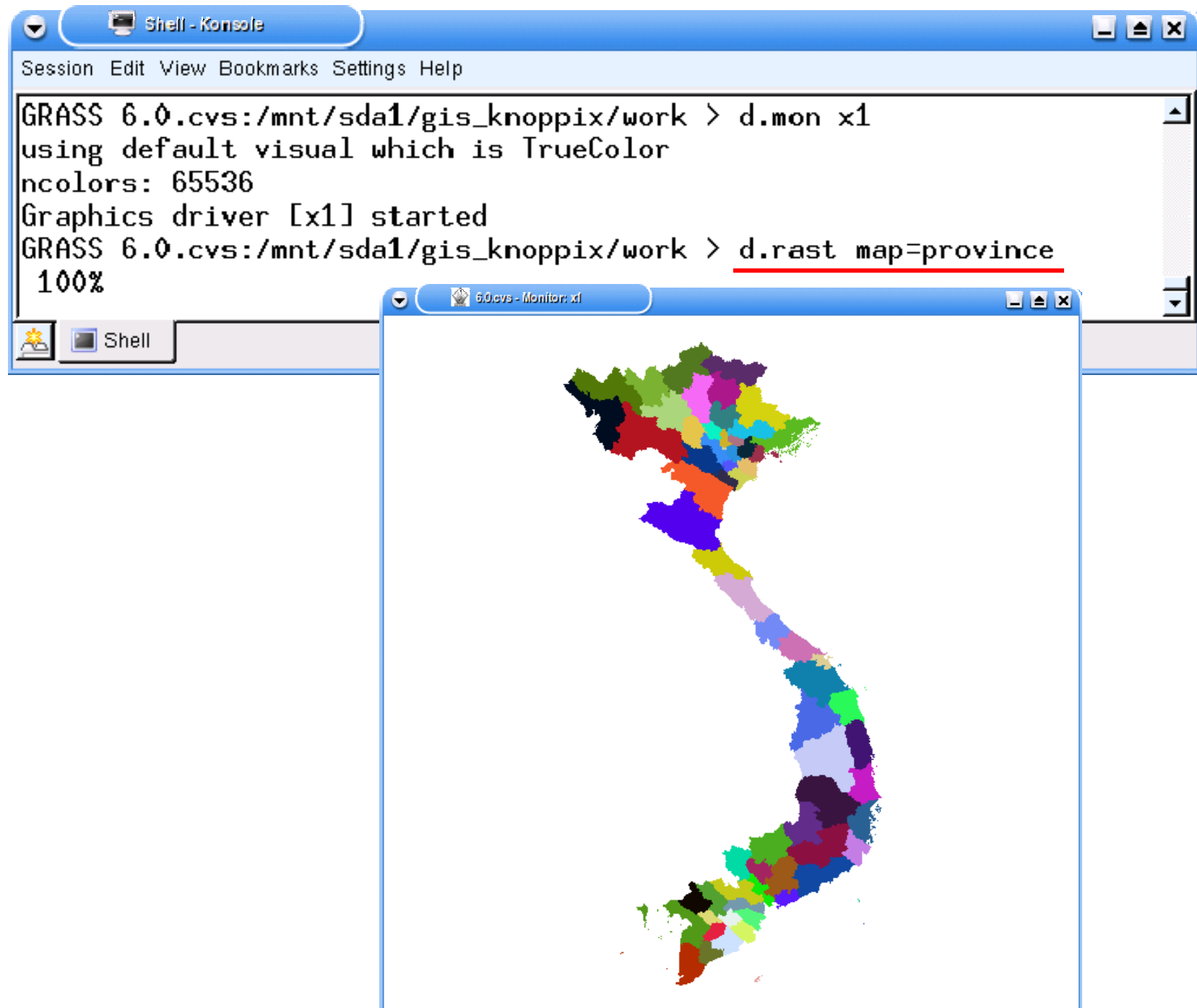
C = (Source of raster values)

[: space

v.to.rast will only affect data in areas lying inside the boundaries of the current geographic region.

```
Shell - Konsole
Session Edit View Bookmarks Settings Help

GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > v.to.rast input=province output
=province use=cat
Loading vector information .../usr/local/grass6.0.cvs-i686-pc-linux-gnu-22
_01_2005/driver/db/ogr: error while loading shared libraries: libtk.so.0:
cannot open shared object file: No such file or directory
dbmi: Protocol error (invalid table/column name or unsupported column type
)
WARNING: Cannot open driver 'ogr'
WARNING: Cannot open OGR DBMI driver.
0 mins 00 secs
Sorting areas by size ...WARNING: Area without centroid (may be OK for isl
and)
WARNING: Area without centroid (may be OK for island)
WARNING: Area without centroid (may be OK for island)
WARNING: Area without centroid (may be OK for island)
WARNING: Area without centroid (may be OK for island)
WARNING: Area without centroid (may be OK for island)
WARNING: Area without centroid (may be OK for island)
WARNING: Area without centroid (may be OK for island)
WARNING: Area without centroid (may be OK for island)
WARNING: Area without centroid (may be OK for island)
WARNING: Area without centroid (may be OK for island)
WARNING: Area without centroid (may be OK for island)
WARNING: Area without centroid (may be OK for island)
WARNING: Area without centroid (may be OK for island)
WARNING: Area without centroid (may be OK for island)
WARNING: Area without centroid (may be OK for island)
WARNING: Area without centroid (may be OK for island)
WARNING: Area without centroid (may be OK for island)
1005 areas 0 mins 00 secs
Processing areas ... 1005 areas 0 mins 02 secs
Processing lines ... 1992 lines 0 mins 00 secs
Writing raster map ... 0 mins 02 secs
Creating support files for raster map ... 0 mins 00 secs
```



Section 2. Data Processing in Latitude-Longitude Coordinate System

7. Overlay raster and vector data

Step 0. Import raster and vector data

Step 1. Launch monitor

```
> d.mon x#
```

= 0, 1, 2, ..., 6 : space

Step 2. Display raster

```
> d.rast  map = A
```

A = (input file name)

Step 3. Display vector

```
> d.vect  map = A  type = boundary  color = B
```

A = (input file name)
B = (line color)

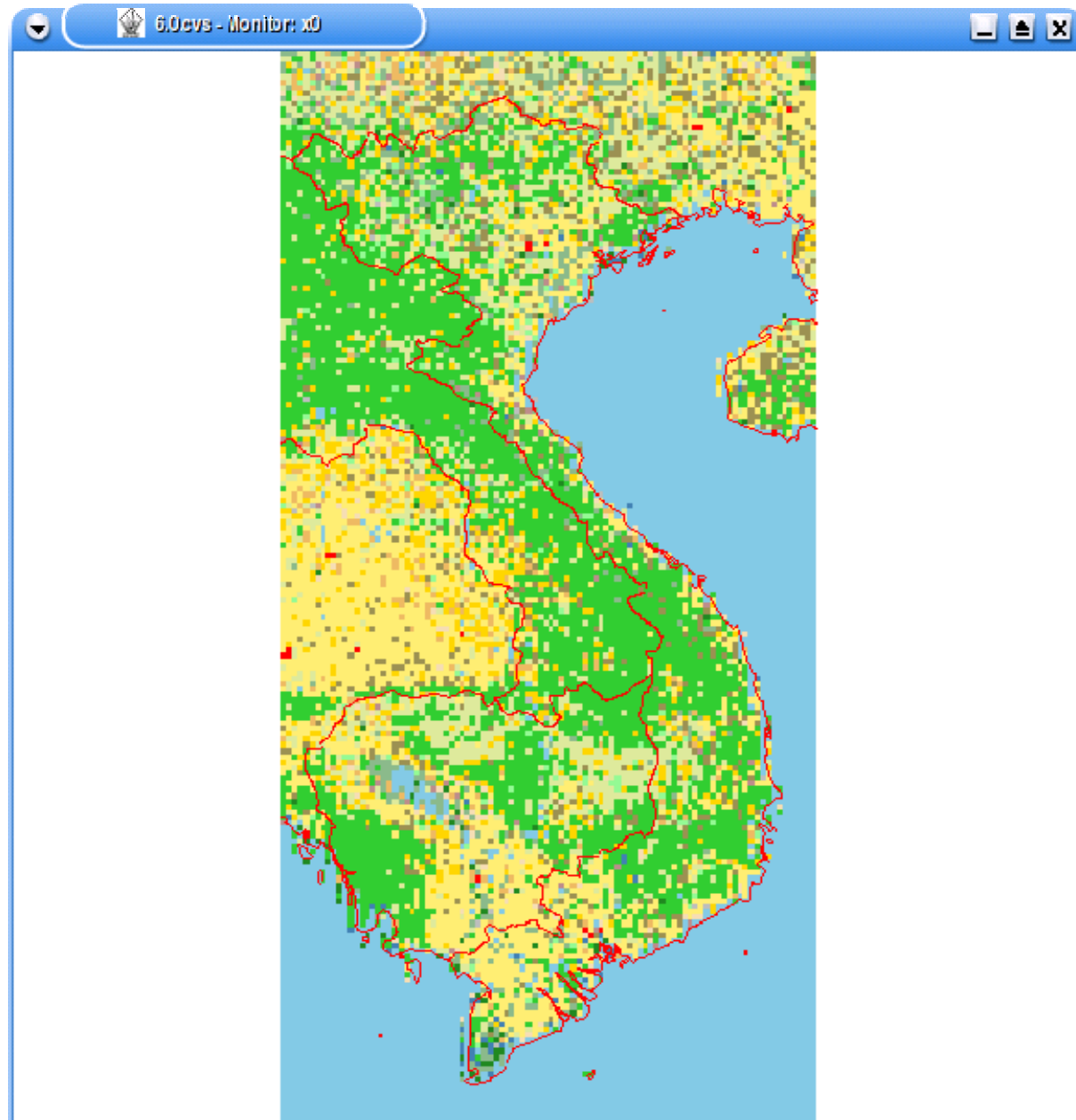
Launch monitor

Display raster

Display vector

```
Shell - Konsole
Session Edit View Bookmarks Settings Help

GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.mon x0
using default visual which is TrueColor
ncolors: 65536
Graphics driver [x0] started
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.rast map=BU
100%
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.vect map=border type=bo
undary color=red
/usr/local/grass6.0.cvs-i686-pc-linux-gnu-22_01_2005/driver/db/ogr:
error while loading shared libraries: libtk.so.0: cannot open shared
object file: No such file or directory
dbmi: Protocol error (invalid table/column name or unsupported colum
n type)
WARNING: Cannot open driver 'ogr'
WARNING: Cannot open OGR DBMI driver.
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > █
```



Section 2. Data Processing in Latitude-Longitude Coordinate System

8. Create RGB image

Step 0. Import images for R,G, B

⇒ *see “3. Import raster data”*

Step 1. change color map to “grey scale map” respectively

```
> r.colors[map=A] color= grey
```

A = (input file name)

[] : space

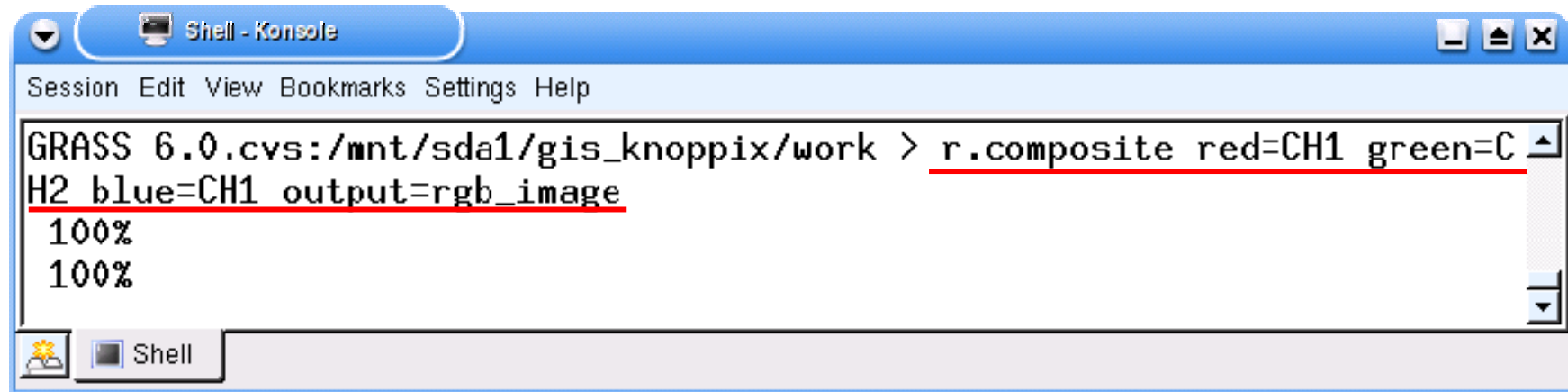
⇒ *see “4.3”*

Step 2. composite images

```
> r.composite red = A green = B blue = C output = D : space
```

A, B, C = (input file name) D = (output file name)

r.composite - Combines red, green and blue map layers into a single composite map layer. **This command creates new image file.**



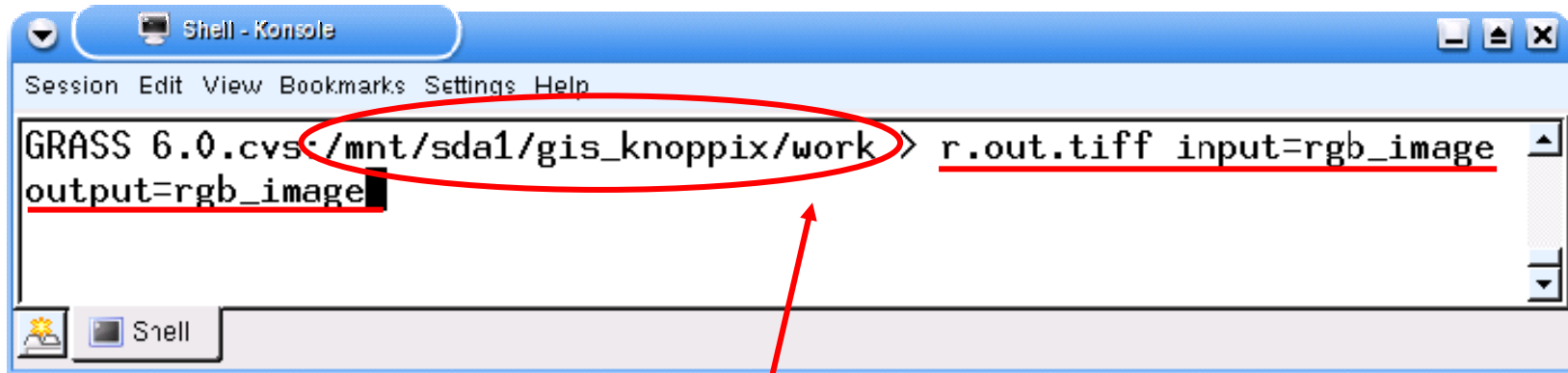
```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > r.composite red=CH1 green=C  
H2 blue=CH1 output=rgb_image  
100%  
100%
```

Step 3. save composite images as TIFF file

> r.out.tiff input = D output = E : space

D = (input file name)

E = (output file name)



```
GRASS 6.0.cvs /mnt/sda1/gis_knoppix/work> r.out.tiff input=rgb_image  
output=rgb_image
```

Output file is saved in this folder

Section 2. Data Processing in Latitude-Longitude Coordinate System

9. Raster calculations

9.1 Masking

9.2 Extraction of specific LC class from LC map

9.3 Calculation of NDVI

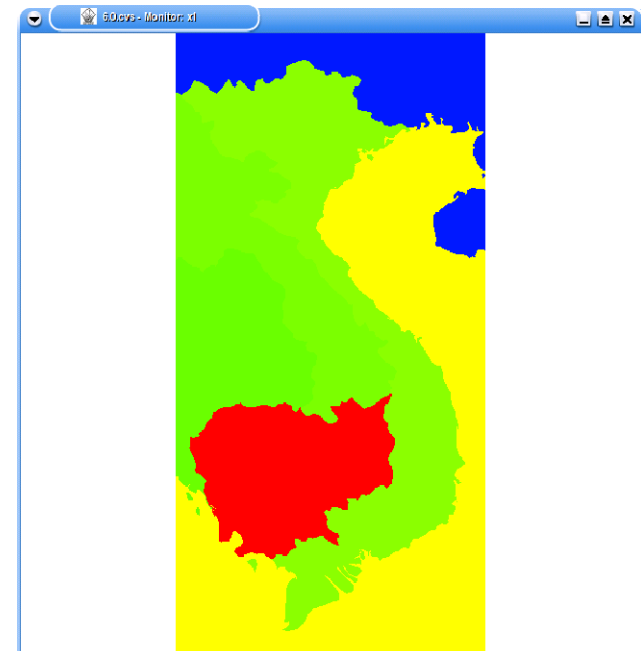
9.4 Histogram equalization

9.5 Histogram normalization

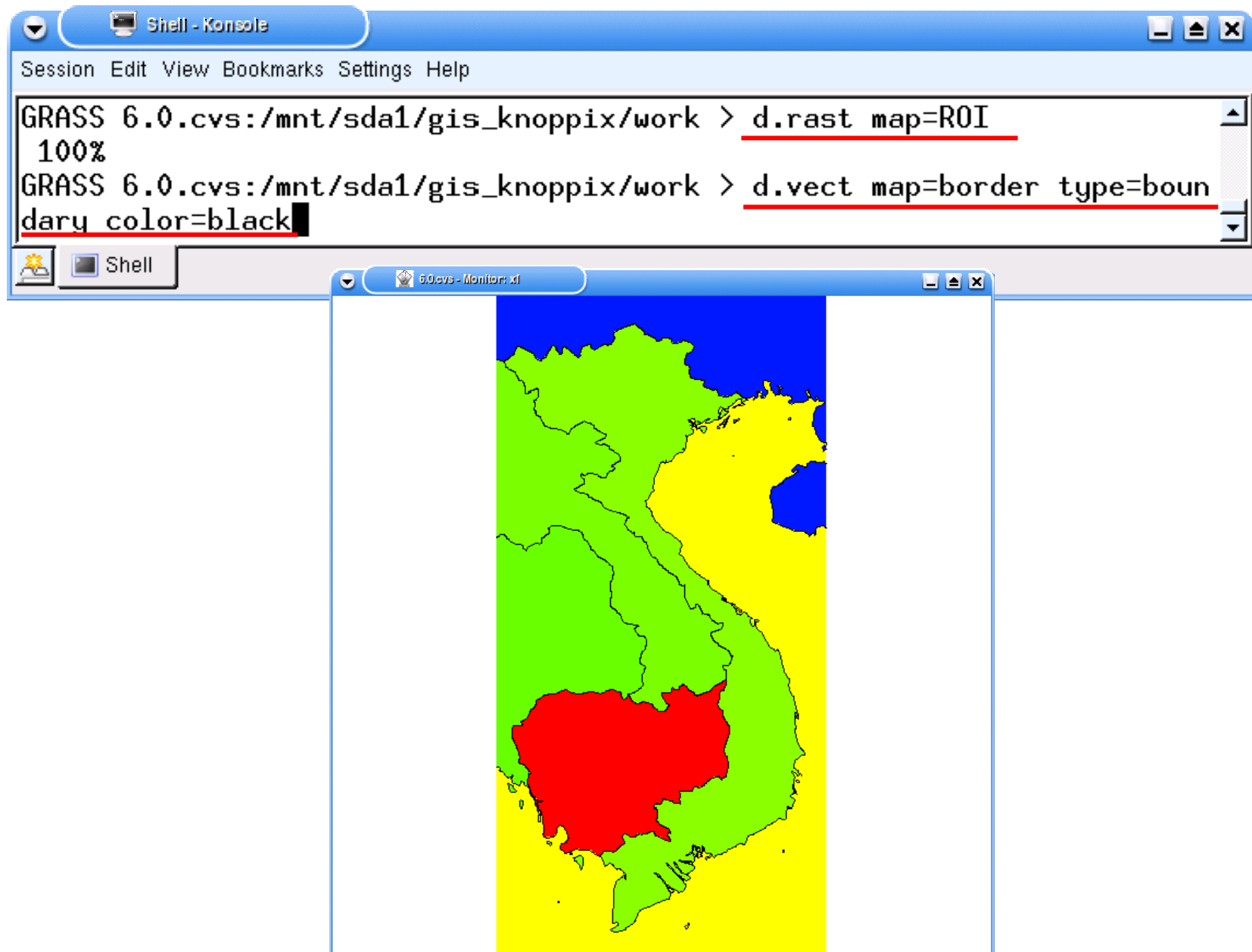
9.1 Masking

9.1.1 Import raster data

9.1.2 Display raster data

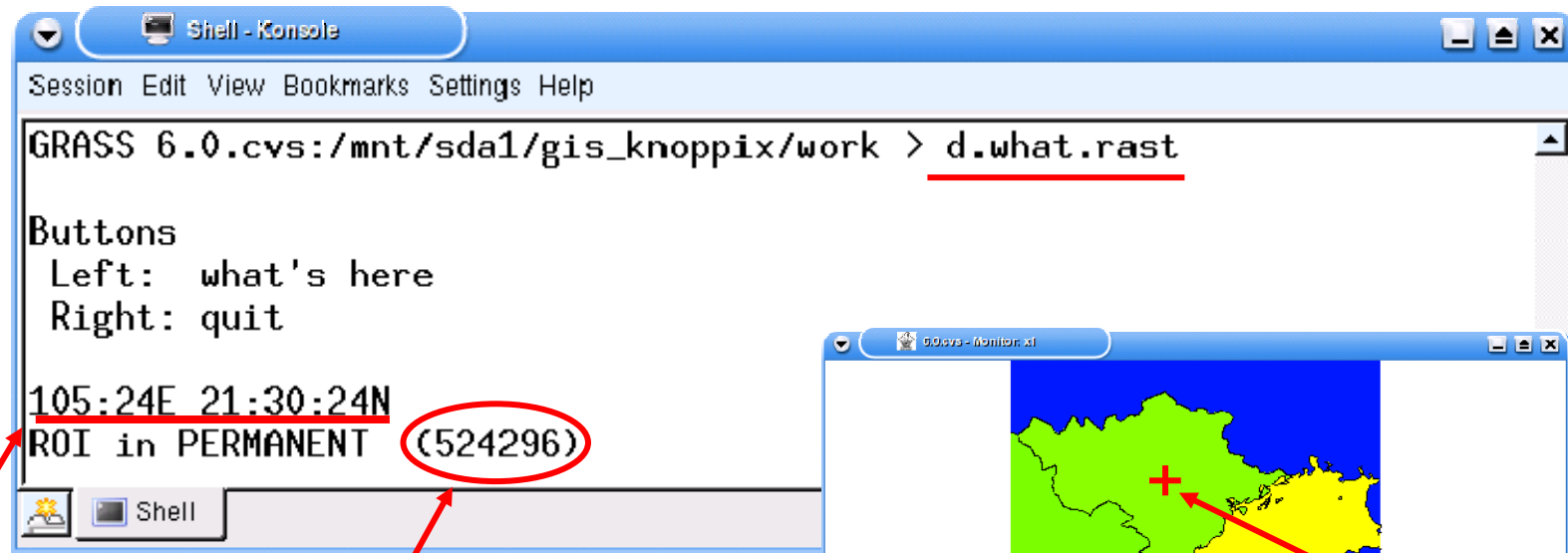


9.1.3 Overlay vector data



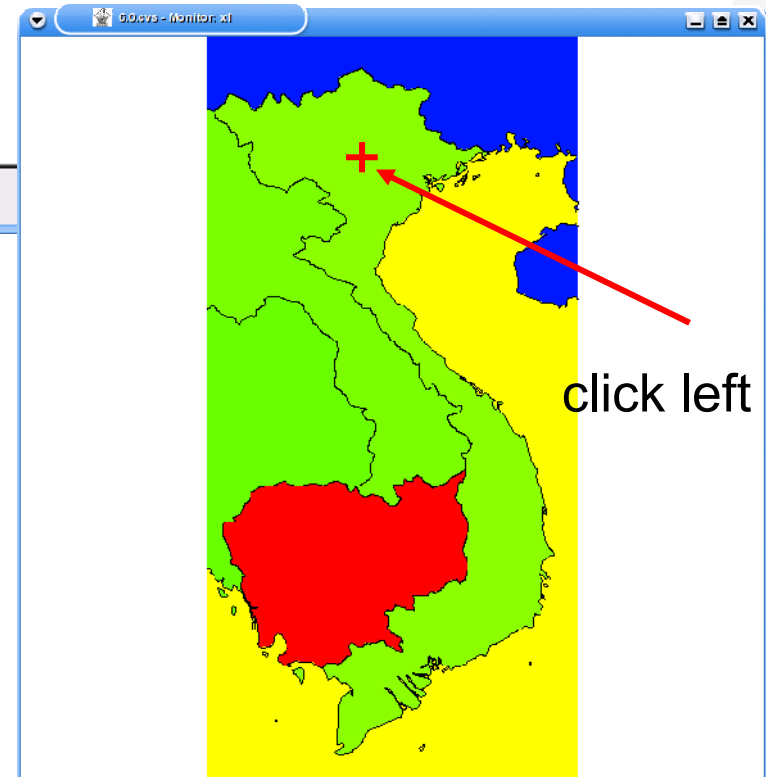
9.1.4 Search raster value

> d.what.rast



position of “+”

raster value below “+”



To quit this process,
click right button on
the display window


```
Shell - Konsole
Session Edit View Bookmarks Settings Help
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.what.rast

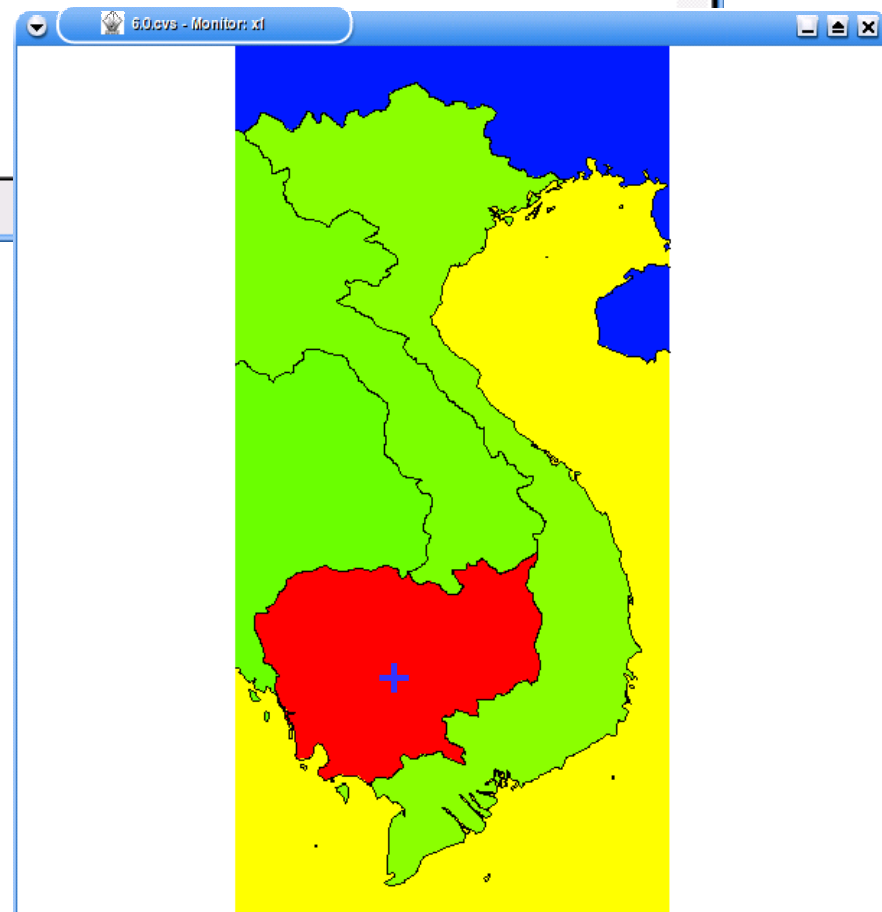
Buttons
Left:  what's here
Right: quit

105:24E 21:30:24N
ROI in PERMANENT (524296)

104:40:48E 12:15:12N
ROI in PERMANENT (5636182)
```

position of “+”

raster value below “+”



9.1.5 Create mask image

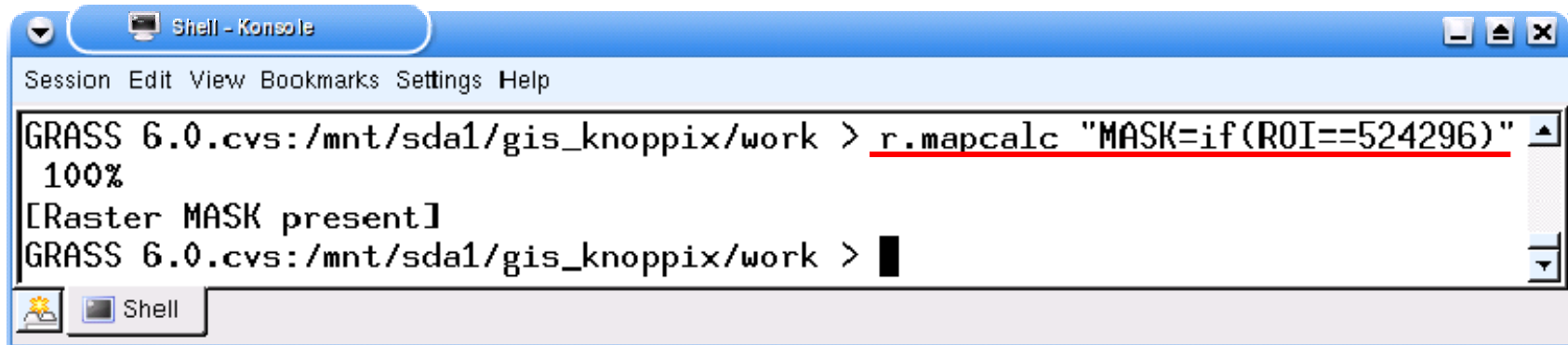
```
> r.mapcalc "MASK = if ( # * A )"
```

A = original file name

= value

* = operator (ex. ==, <=, >=)

␣ : space



```
Shell - Konsole
Session Edit View Bookmarks Settings Help
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > r.mapcalc "MASK=if(ROI==524296)"
100%
[Raster MASK present]
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > █
```

original image

1	4	4	1	1	5
9	2	4	6	1	1
9	6	1	8	1	7
3	3	5	8	2	1
3	1	8	1	5	9
3	1	1	7	7	1

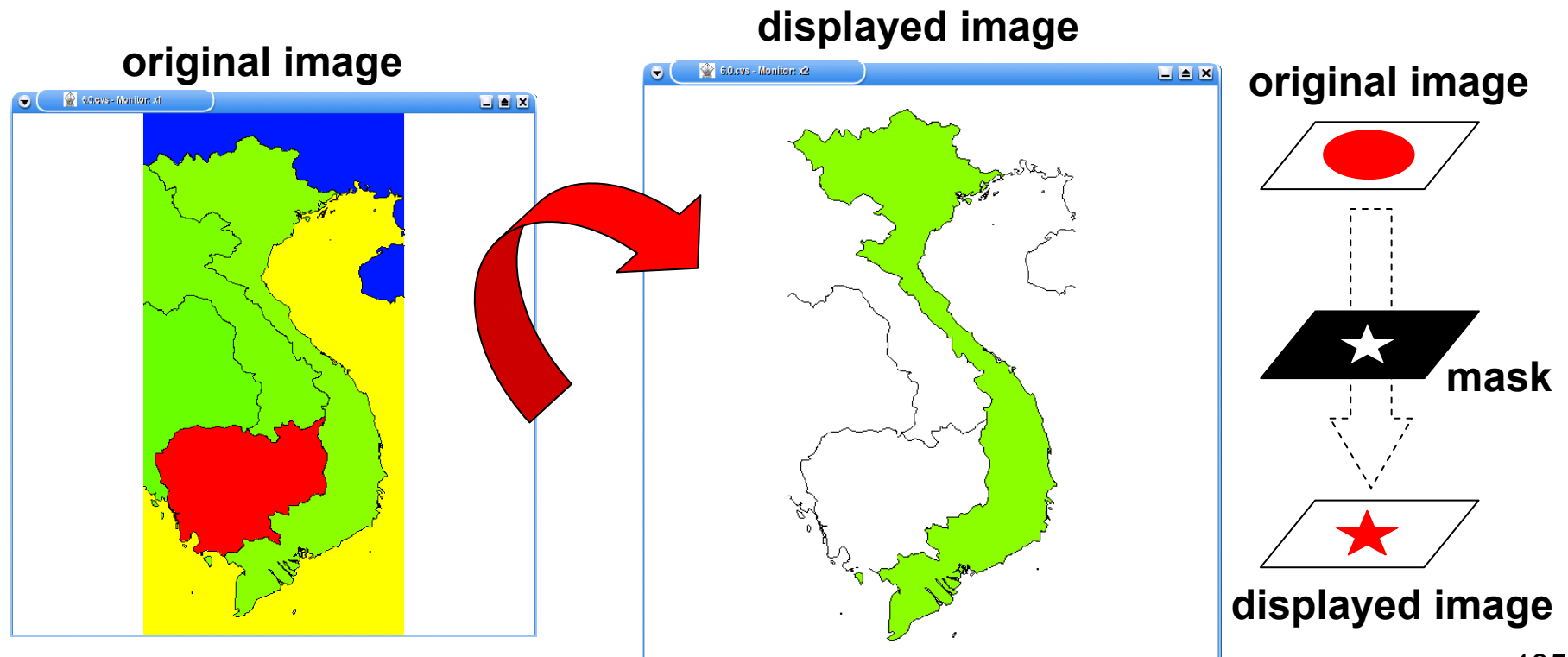


output image

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	1	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	0	0	0	0

9.1.6 Display raster image with mask

```
Shell - Konsole
Session Edit View Bookmarks Settings Help
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.rast map=R0I
100%
[Raster MASK present]
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.vect map=border type=boundary
color=black
```



9.1.7 Remove mask

```
> g.remove rast=MASK
```

: space

g.remove removes data base element files from the user's current mapset.

Once you create MASK, the following processes (commands) are affected by the MASK until you remove it.

The screenshot shows a terminal window titled "Shell - Konsole" with a menu bar (Session, Edit, View, Bookmarks, Settings, Help). The terminal output is as follows:

```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > r.mapcalc "MASK=if(ROI==524296)"
100%
[Raster MASK present]
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.mon x0
using default visual which is TrueColor
ncolors: 65536
Graphics driver [x0] started
[Raster MASK present]
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > g.remove rast=MASK
REMOVE [MASK]
raster
header
category
color    MISSING
history
misc
fcell    MISSING
g3dcell  MISSING
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.mon x1
using default visual which is TrueColor
ncolors: 65536
Graphics driver [x1] started
```

Annotations on the screenshot:

- Creation of MASK**: A red arrow points to the command `r.mapcalc "MASK=if(ROI==524296)"`.
- MASK is active**: A red arrow points to the status message `[Raster MASK present]` that appears after the first `d.mon x0` command.
- This command is affected by MASK.**: A blue arrow points to the `d.mon x0` command.
- MASK removal**: A red arrow points to the command `g.remove rast=MASK`.
- This command is not affected by MASK.**: A blue arrow points to the `d.mon x1` command.

At the bottom of the terminal window, a tab labeled "Shell" is visible. A large black arrow points from the text "[Raster MASK present] disappeared" below the terminal to this tab.

[Raster MASK present] disappeared

9.2 Extraction of specific LC class from LC map

9.2.1 Extract only “Grassland” area

BU Legend

0 : Water (and Goode's interrupted space)

1 : Evergreen Needleleaf Forest

2 : Evergreen Broadleaf Forest

3 : Deciduous Needleleaf Forest

4 : Deciduous Broadleaf Forest

5 : Mixed Forest

6 : Closed Shrublands

7 : Open Shrubland

8 : Woody Savannas

9 : Savannas

10 : Grassland

11 : Permanent Wetlands

12 : Croplands

13 : Urban and Built-up

14 : Cropland/Natural Vegetation Mosaic

15 : Snow and Ice

16 : Barren or Sparsely Vegetated

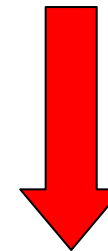
⇒ 0

⇒ 1

⇒ 0

1	4	4	10	10	5
9	2	4	6	10	10
9	6	10	8	1	7
3	3	5	8	2	1
3	1	8	1	5	9
3	1	1	7	7	1

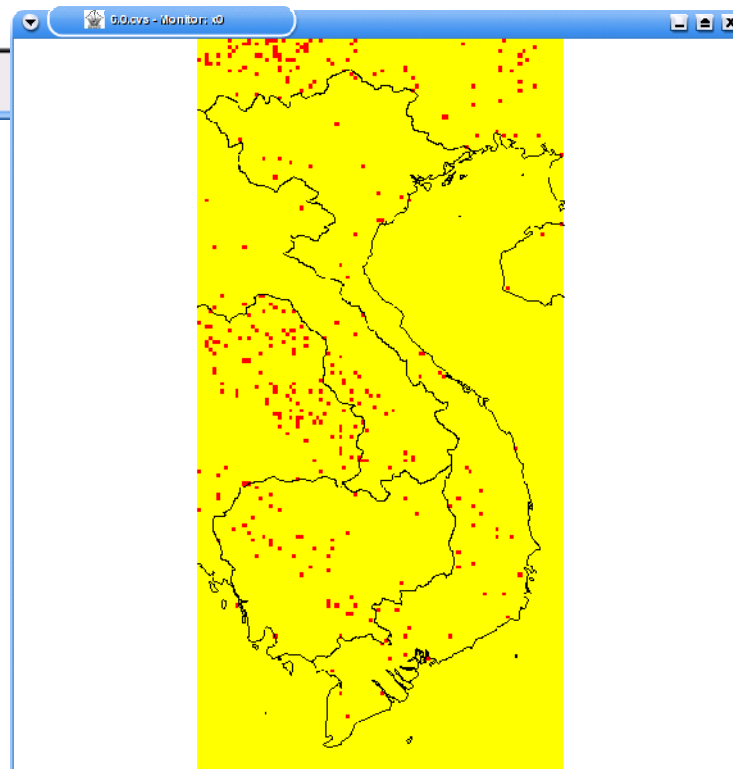
original image



0	0	0	1	1	0
0	0	0	0	1	1
0	0	1	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

output image

```
Shell - Konsole
Session Edit View Bookmarks Settings Help
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > r.mapcalc "grassland=if(BU==10)"
100%
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.mon x0
using default visual which is trueColor
ncolors: 65536
Graphics driver [x0] started
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.rast map=grassland
100%
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.vect map=border type=boundary
color=black
```



- Grassland
- Othders

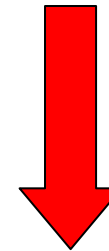
9.2.2 Extract “Forest” area

BU Legend

- | | |
|---|-------|
| 0 : Water (and Goode's interrupted space) | ⇒ 0 |
| 1 : Evergreen Needleleaf Forest | } ⇒ 1 |
| 2 : Evergreen Broadleaf Forest | |
| 3 : Deciduous Needleleaf Forest | |
| 4 : Deciduous Broadleaf Forest | |
| 5 : Mixed Forest | |
| 6 : Closed Shrublands | } ⇒ 0 |
| 7 : Open Shrubland | |
| 8 : Woody Savannas | |
| 9 : Savannas | |
| 10 : Grassland | |
| 11 : Permanent Wetlands | |
| 12 : Croplands | |
| 13 : Urban and Built-up | |
| 14 : Cropland/Natural Vegetation Mosaic | |
| 15 : Snow and Ice | |
| 16 : Barren or Sparsely Vegetated | |

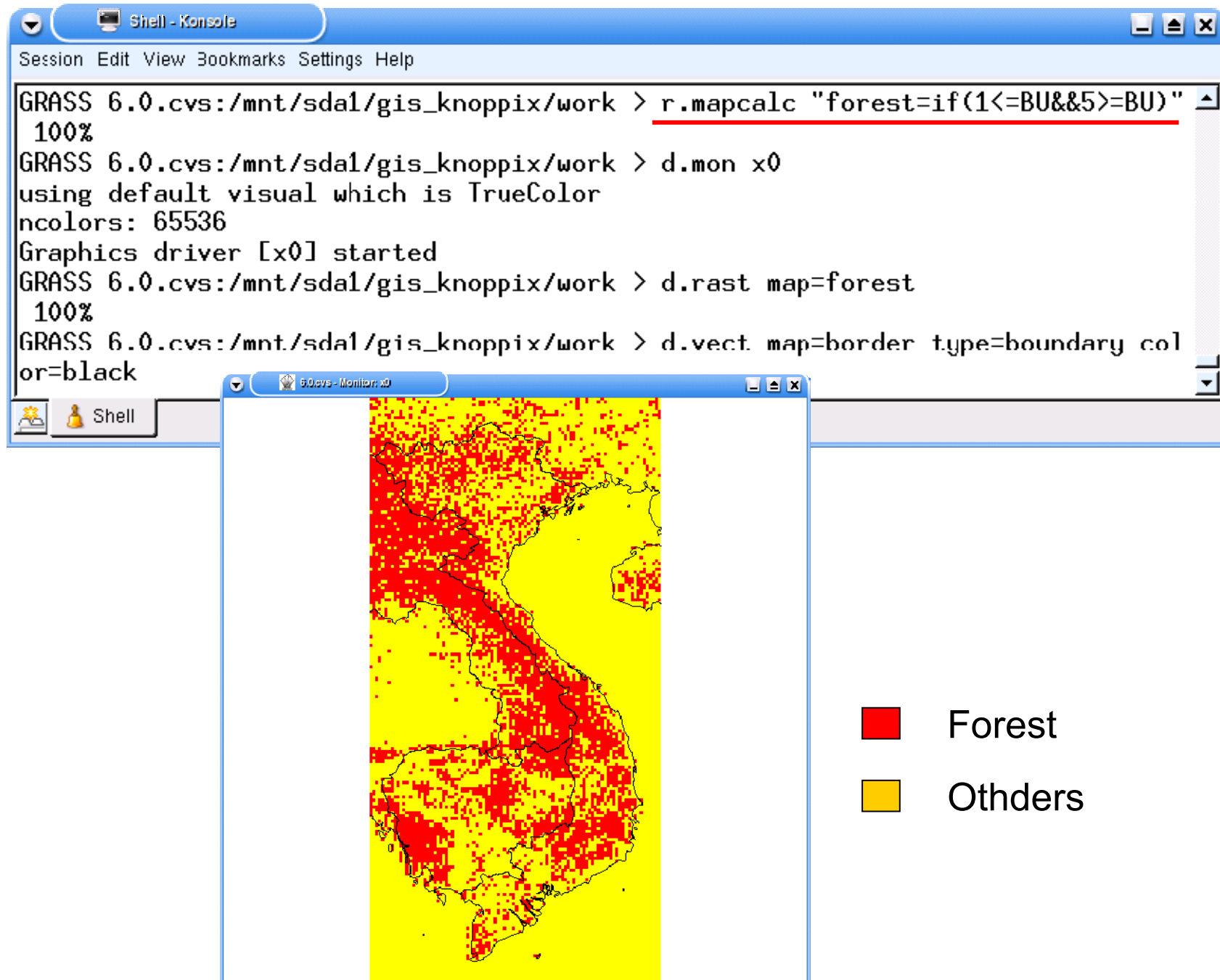
1	4	4	10	10	5
9	2	4	6	10	10
9	6	10	8	1	7
3	3	5	8	2	1
3	1	8	1	5	9
3	1	1	7	7	1

original image



1	1	1	0	0	1
0	1	1	0	0	0
0	1	0	0	1	0
1	1	1	0	1	1
1	1	0	1	1	0
1	1	1	0	0	1

output image



9.3 Calculation of NDVI

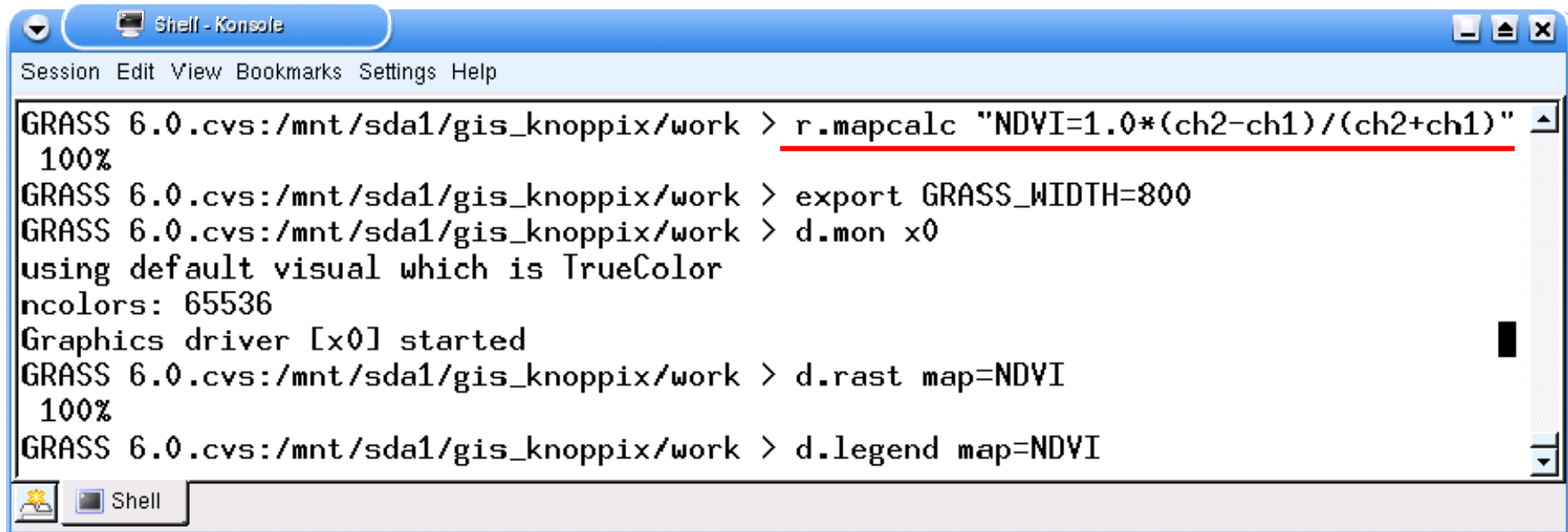
```
> r.mapcalc "NDVI = 1.0 * (A - B) / (A + B)"
```

A = (input file name1)

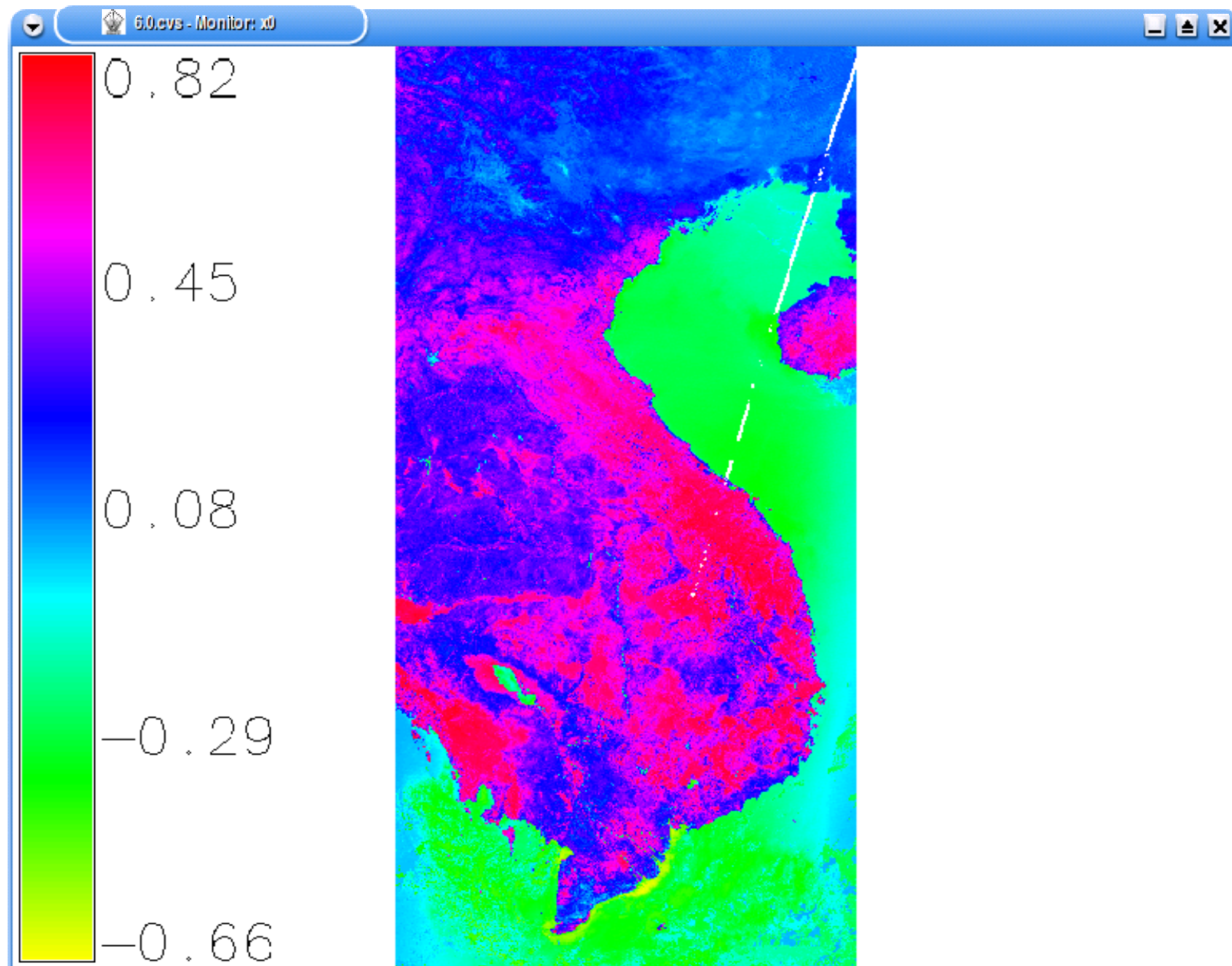
B = (input file name2)

: space

It is important to include a multiplier of **1.0** at the beginning of the map algebra expression when integer values are divided.



```
Shell - Konsole
Session Edit View Bookmarks Settings Help
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > r.mapcalc "NDVI=1.0*(ch2-ch1)/(ch2+ch1)"
100%
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > export GRASS_WIDTH=800
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.mon x0
using default visual which is TrueColor
ncolors: 65536
Graphics driver [x0] started
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.rast map=NDVI
100%
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.legend map=NDVI
```



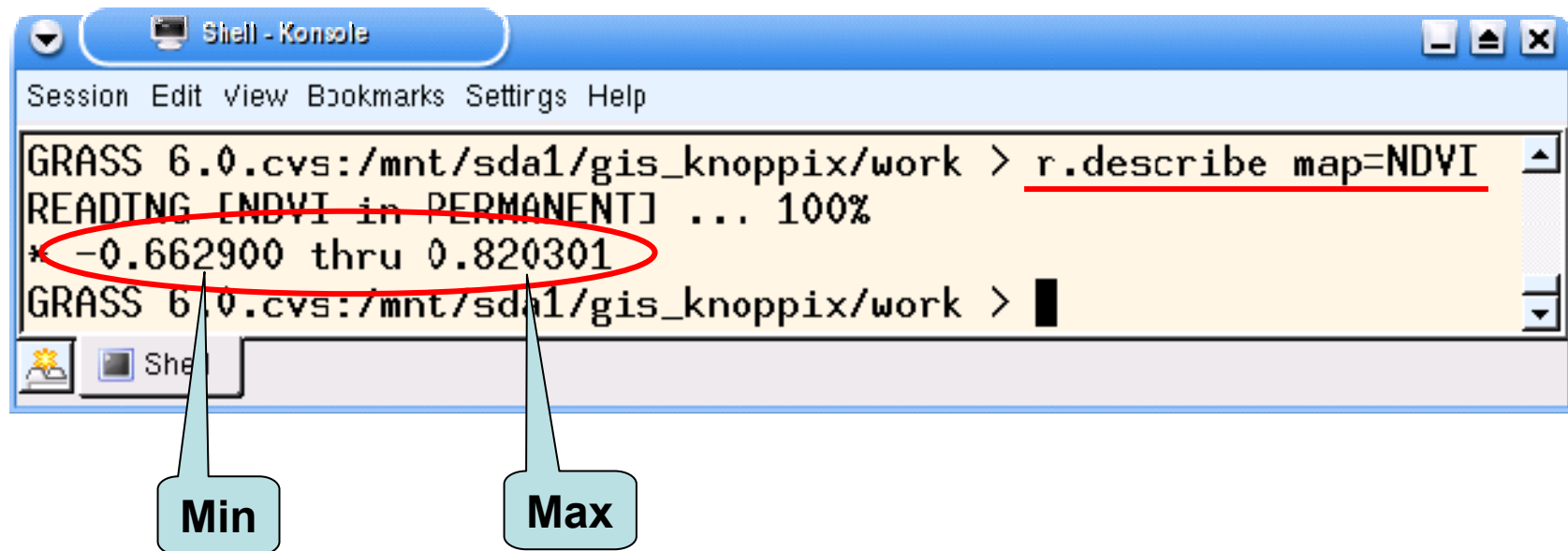
How to get max and min values of NDVI

> r.describe map = A

A = (input file name)

: space

r.describe prints terse list of category values found in a raster map layer.



```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > r.describe map=NDVI
READING [NDVI in PERMANENT] ... 100%
* -0.662900 thru 0.820301
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > 
```

Min

Max

9.4 Histogram equalization

> r.rescale.eq input = A output = B to = 0, 255

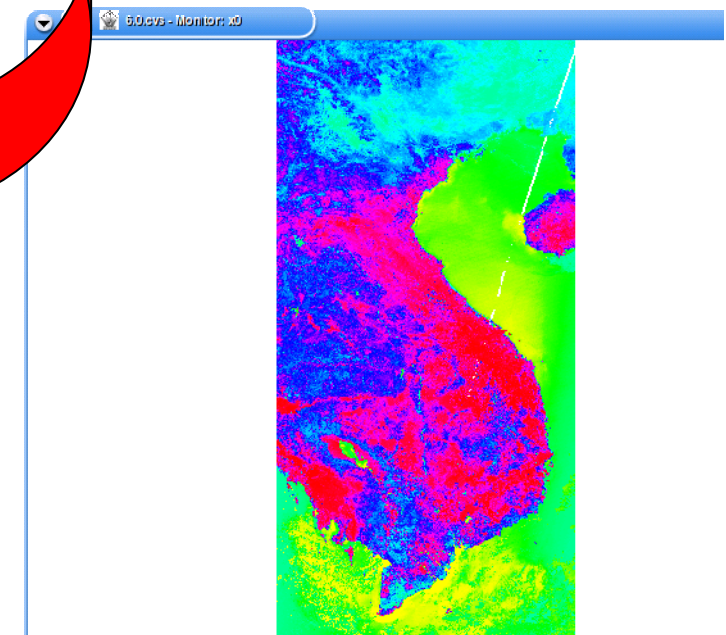
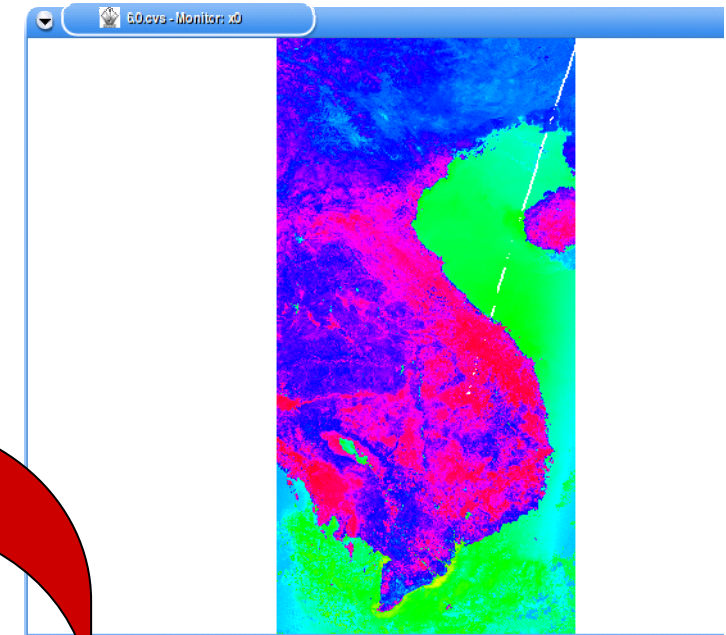
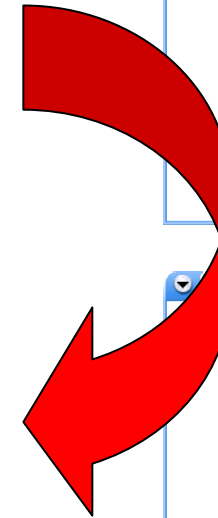
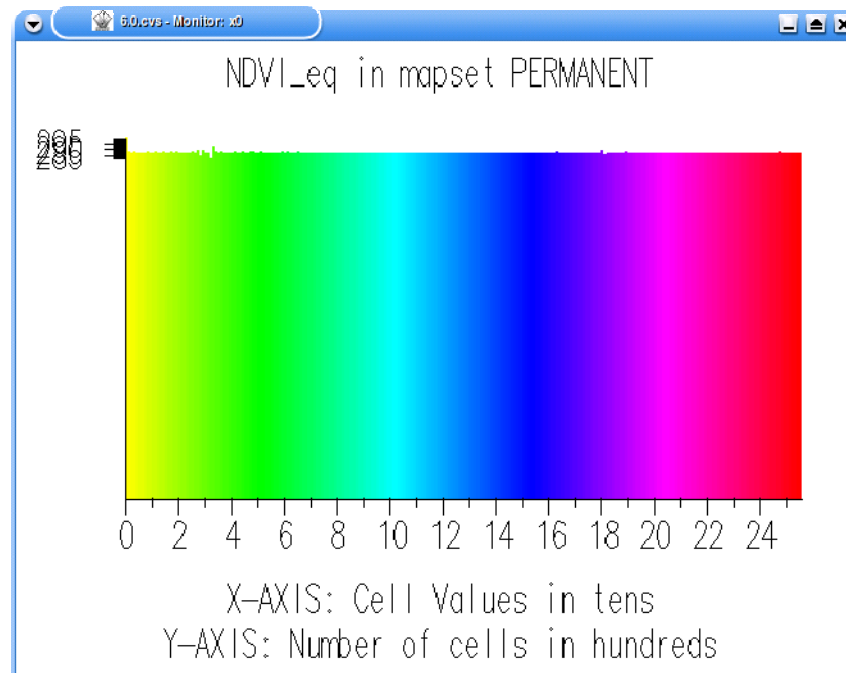
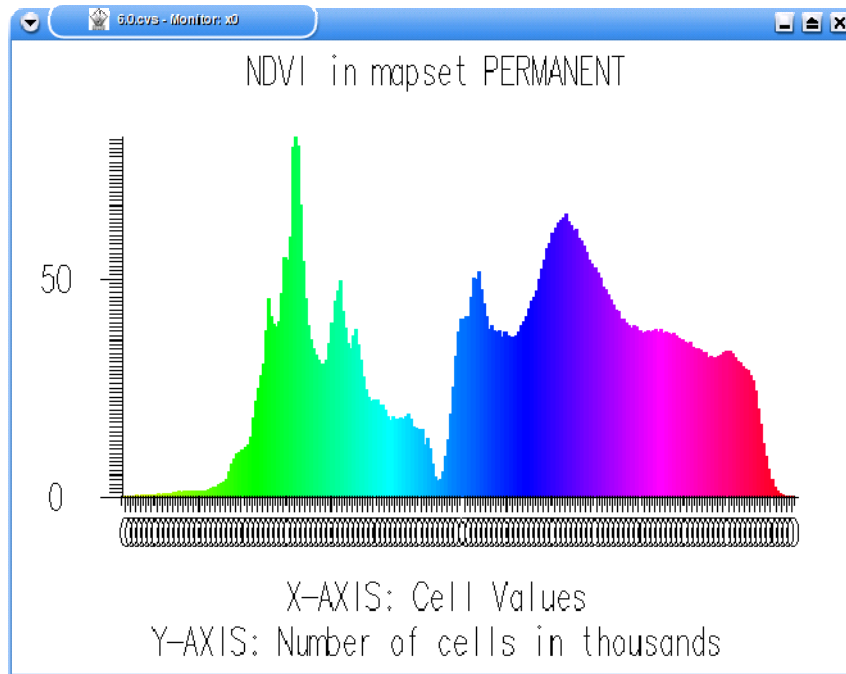
A = (input file name)
B = (output file name)

r.rescale.eq rescales histogram equalized the range of category values in a raster map layer.

: space

```
Shell - Konsole
Session Edit View Bookmarks Settings Help
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > r.describe map=NDVI
READING [NDVI in PERMANENT] ... 100%
* -0.662900 thru 0.820301
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > r.mapcalc "NDVI_int=1000000*NDVI"
100%
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > r.rescale.eq input=NDVI_int output=
NDVI_eq to=0,255
Reading NDVI_int ... 100%
Rescale NDVI_int[-662900,820301] to NDVI_eq[0,255]
```

If raster values are floating point, we need to change them to integer before rescaling.



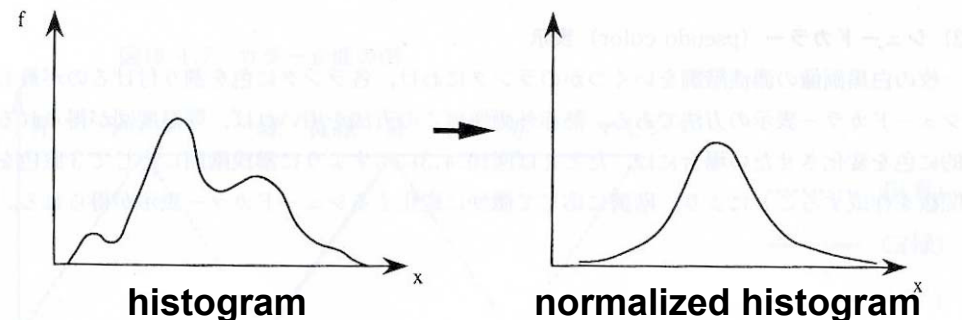
9.5 Histogram normalization

$$y = \frac{S_y}{S_x} (x - x_m) + y_m$$

where x_m ; mean of input image
 S_x ; standard deviation of input image
 y_m ; mean of output image
 S_y ; standard deviation of output image

Generally a normal distribution of density in an image would create an image that is natural for a human observation. In this sense the histogram of original image may be sometimes converted to the normalized histogram as shown in figure. However this conversion, pixels with same gray scale should be reallocated to other pixels with a different gray scale.

Histogram normalization may be applied to an unfocused image with a low dynamic range.



< Procedure of Histogram normalization >

Step 1. Calculate mean and standard deviation of input image by “*r.univar*”

□ : space

```
> r.univar □ input = A          A = (input file name)
```

r.univar calculates univariate statistics of a raster map.

This includes the number of cells counted, minimum and maximum cell values, range, arithmetic mean, population variance, standard deviation, and coefficient of variation.

Step 2. Calculate normalized values by “*r.mapcalc*”

```
> r.mapcalc □ “ B = 1.0 * ( Sy / Sx ) / ( x - xm ) + ym ”
```

B = (output file name)

□ : space


```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > r.univar input=NDVI

Processing .. 100%

total null and non-null cells: 7372800
total null cells: 19258

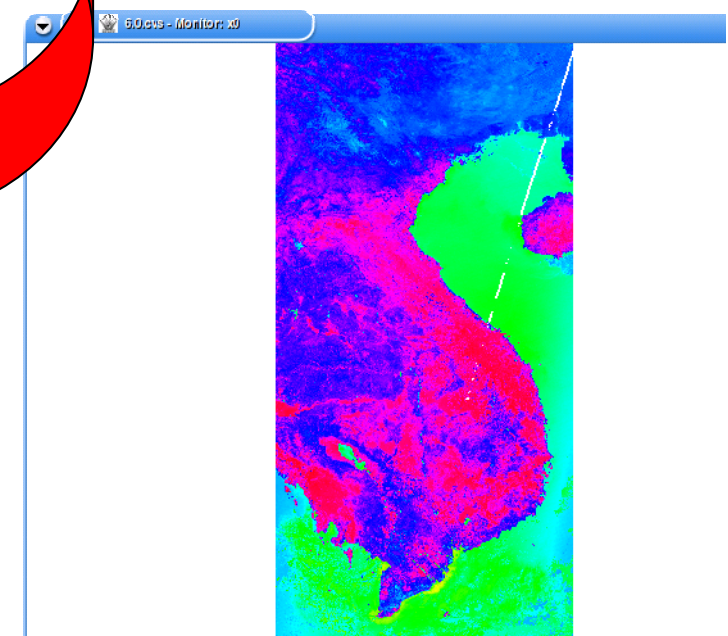
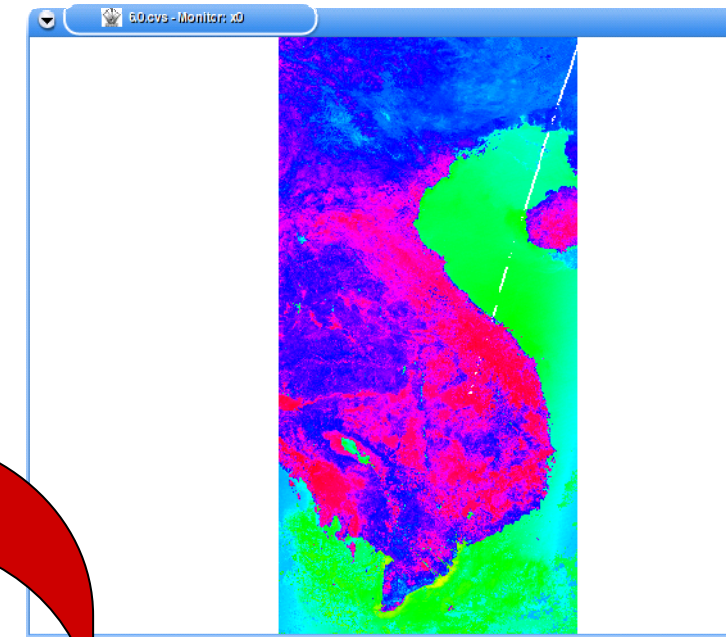
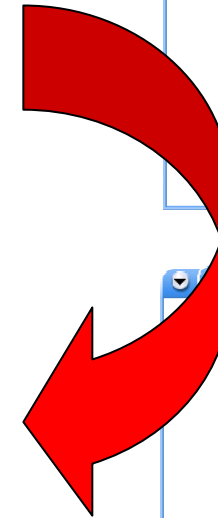
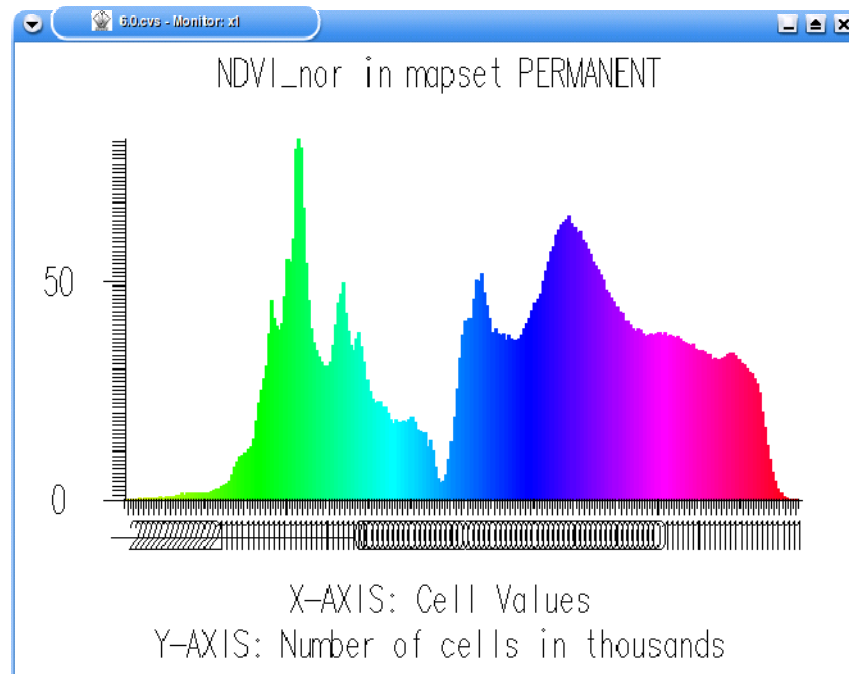
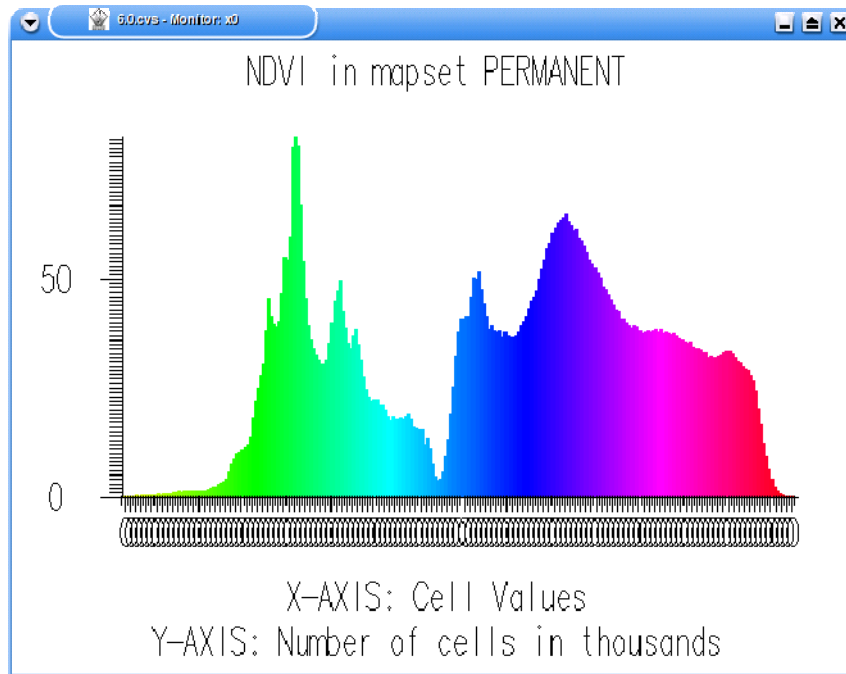
Of the non-null cells:
-----
n: 7353542
minimum: -0.6629
maximum: 0.820301
range: 1.4832
mean: 0.184511
standard deviation: 0.330028
variance: 0.108919
variation coefficient: 178.866 %
sum: 1.35681e+06

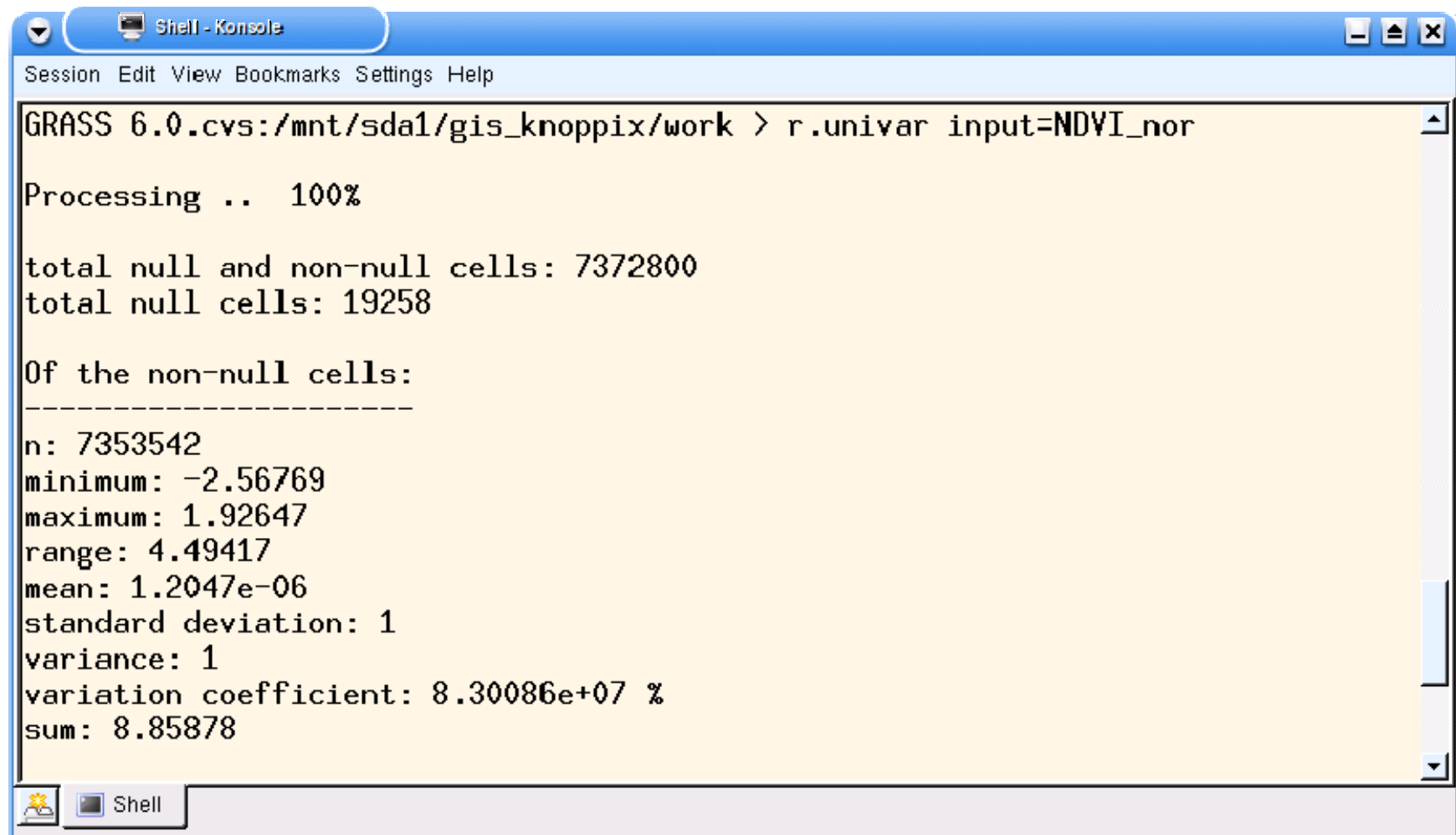
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > r.mapcalc "NDVI_nor=1.0*(1/0.330028)*
(ndvi-0.184511)"
100%
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > █
```

SD

Mean

Shell





The image shows a terminal window titled "Shell - Konsole". The window has a menu bar with "Session", "Edit", "View", "Bookmarks", "Settings", and "Help". The terminal content shows the execution of the command `r.univar input=NDVI_nor` in a GRASS GIS environment. The output indicates that processing is 100% complete. It provides statistics for the input map, including the total number of null and non-null cells, the number of non-null cells, and various statistical measures like minimum, maximum, range, mean, standard deviation, variance, variation coefficient, and sum.

```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > r.univar input=NDVI_nor

Processing .. 100%

total null and non-null cells: 7372800
total null cells: 19258

Of the non-null cells:
-----
n: 7353542
minimum: -2.56769
maximum: 1.92647
range: 4.49417
mean: 1.2047e-06
standard deviation: 1
variance: 1
variation coefficient: 8.30086e+07 %
sum: 8.85878
```

Section 2. Data Processing in Latitude-Longitude Coordinate System

10. Raster mosaic

10.1 Selection of the area

```
> g.region -w = P -e = Q -n = R -s = S -res = T
```

P = (value of western edge)

Q = (value of eastern edge)

R = (value of northern edge)

S = (value of southern edge)

T = (grid resolution)

- : space

10.2 Import of image files

```
> r.in.gdal -input=A -output=a
```

```
> r.in.gdal -input=B -output=b
```

```
> r.in.gdal -input=C -output=c
```

```
> r.in.gdal -input=D -output=d
```

```
> .....
```

A, B, C, D, ... = (input file name)

a, b, c, d, ... = (output file name)

- : space

10.3 Mosaicking

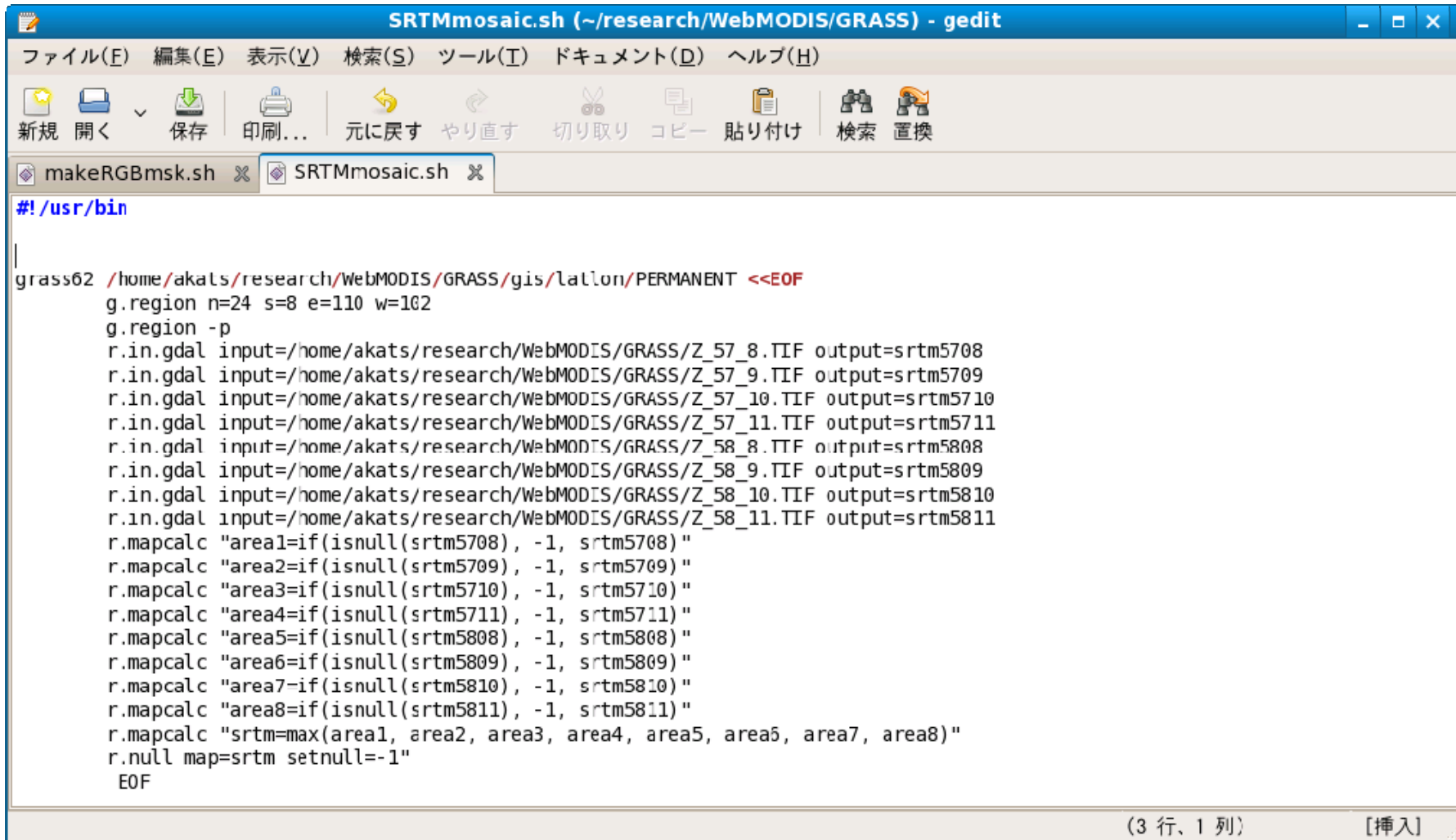
```
> r.mapcalc "area1=if(isnull(a), -1, a)"
> r.mapcalc "area2=if(isnull(b), -1, b)"
> r.mapcalc "area3=if(isnull(c), -1, c)"
> r.mapcalc "area4=if(isnull(d), -1, d)"
> -----

> r.mapcalc "output=max(area1, area2, area3, area4, -----)"

> r.null map=output setnull = -1"
```

␣: space

Shell script of SRTM mosaicking

A screenshot of a gedit text editor window titled "SRTMmosaic.sh (~/.research/WebMODIS/GRASS) - gedit". The window has a menu bar with options: ファイル(F), 編集(E), 表示(V), 検索(S), ツール(T), ドキュメント(D), ヘルプ(H). Below the menu is a toolbar with icons for new, open, save, print, undo, redo, cut, copy, paste, search, and replace. The tabs show "makeRGBmsk.sh" and "SRTMmosaic.sh". The script content is as follows:

```
#!/usr/bin

|
grass62 /home/akats/research/WebMODIS/GRASS/gis/lallon/PERMANENT <<EOF
  g.region n=24 s=8 e=110 w=162
  g.region -p
  r.in.gdal input=/home/akats/research/WebMODIS/GRASS/Z_57_8.TIF output=srtm5708
  r.in.gdal input=/home/akats/research/WebMODIS/GRASS/Z_57_9.TIF output=srtm5709
  r.in.gdal input=/home/akats/research/WebMODIS/GRASS/Z_57_10.TIF output=srtm5710
  r.in.gdal input=/home/akats/research/WebMODIS/GRASS/Z_57_11.TIF output=srtm5711
  r.in.gdal input=/home/akats/research/WebMODIS/GRASS/Z_58_8.TIF output=srtm5808
  r.in.gdal input=/home/akats/research/WebMODIS/GRASS/Z_58_9.TIF output=srtm5809
  r.in.gdal input=/home/akats/research/WebMODIS/GRASS/Z_58_10.TIF output=srtm5810
  r.in.gdal input=/home/akats/research/WebMODIS/GRASS/Z_58_11.TIF output=srtm5811
  r.mapcalc "area1=if(isnull(srtm5708), -1, srtm5708)"
  r.mapcalc "area2=if(isnull(srtm5709), -1, srtm5709)"
  r.mapcalc "area3=if(isnull(srtm5710), -1, srtm5710)"
  r.mapcalc "area4=if(isnull(srtm5711), -1, srtm5711)"
  r.mapcalc "area5=if(isnull(srtm5808), -1, srtm5808)"
  r.mapcalc "area6=if(isnull(srtm5809), -1, srtm5809)"
  r.mapcalc "area7=if(isnull(srtm5810), -1, srtm5810)"
  r.mapcalc "area8=if(isnull(srtm5811), -1, srtm5811)"
  r.mapcalc "srtm=max(area1, area2, area3, area4, area5, area6, area7, area8)"
  r.null map=srtm setnull=-1"
EOF
```

The status bar at the bottom right shows "(3 行、1 列)" and "[挿入]".

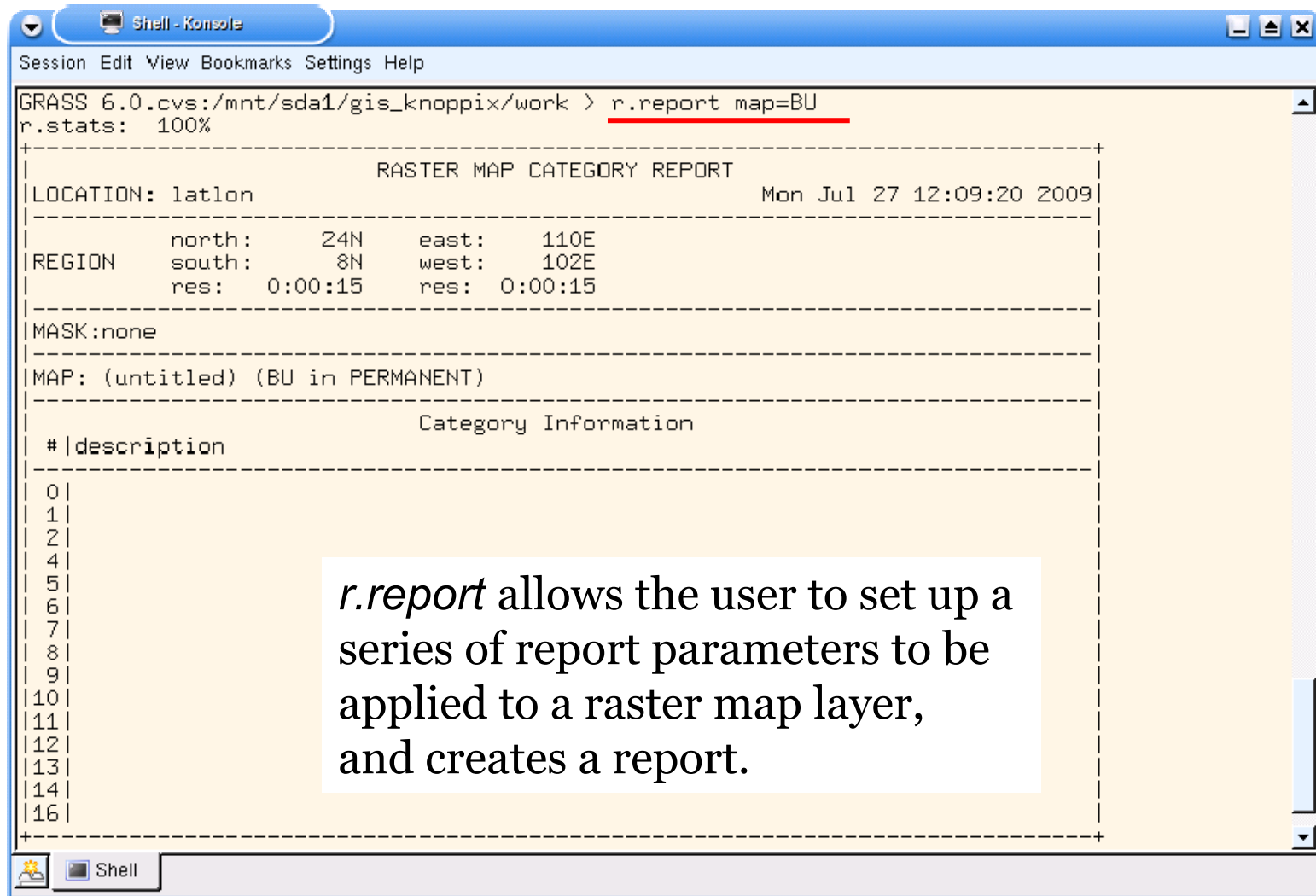
Section 2. Data Processing in Latitude-Longitude Coordinate System

11. Raster statistics

11.1 Search raster data information

> r.report map= A

A = (input file name) : space



The screenshot shows a terminal window titled "Shell - Konsole" with a menu bar (Session, Edit, View, Bookmarks, Settings, Help). The command prompt shows the user is in the directory `/mnt/sda1/gis_knoppix/work` and has executed `r.report map=BU`. The output shows the command completed successfully with 100% progress. Below this, a "RASTER MAP CATEGORY REPORT" is generated, dated "Mon Jul 27 12:09:20 2009". The report includes the location "latlon", region coordinates (north: 24N, east: 110E, south: 8N, west: 102E, resolution: 0:00:15), and mask "none". The map is titled "(untitled) (BU in PERMANENT)". A section for "Category Information" follows, with a table header "#|description" and a list of categories from 0 to 16, each with a vertical bar next to its number.

```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > r.report map=BU
r.stats: 100%
-----
LOCATION: latlon                               Mon Jul 27 12:09:20 2009
-----
REGION      north:    24N    east:    110E
            south:    8N     west:    102E
            res:      0:00:15  res:      0:00:15
-----
MASK:none
-----
MAP: (untitled) (BU in PERMANENT)
-----
Category Information
#|description
-----
0|
1|
2|
4|
5|
6|
7|
8|
9|
10|
11|
12|
13|
14|
16|
```

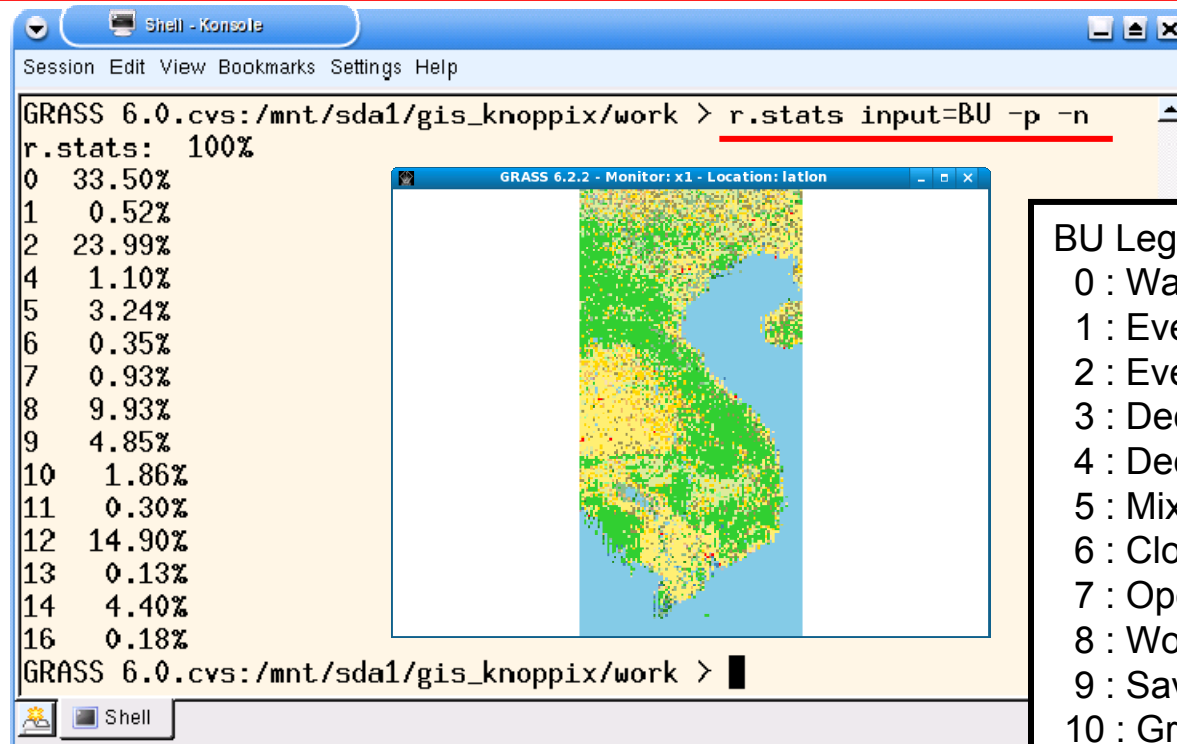
r.report allows the user to set up a series of report parameters to be applied to a raster map layer, and creates a report.

11.2 Generates area statistics for raster data

> r.stats input = A -p -n

A = (input file name)

: space



BU Legend

- 0 : Water (and Goode's interrupted space)
- 1 : Evergreen Needleleaf Forest
- 2 : Evergreen Broadleaf Forest
- 3 : Deciduous Needleleaf Forest
- 4 : Deciduous Broadleaf Forest
- 5 : Mixed Forest
- 6 : Closed Shrublands
- 7 : Open Shrubland
- 8 : Woody Savannas
- 9 : Savannas
- 10 : Grassland
- 11 : Permanent Wetlands
- 12 : Croplands
- 13 : Urban and Built-up
- 14 : Cropland/Natural Vegetation Mosaic
- 15 : Snow and Ice
- 16 : Barren or Sparsely Vegetated

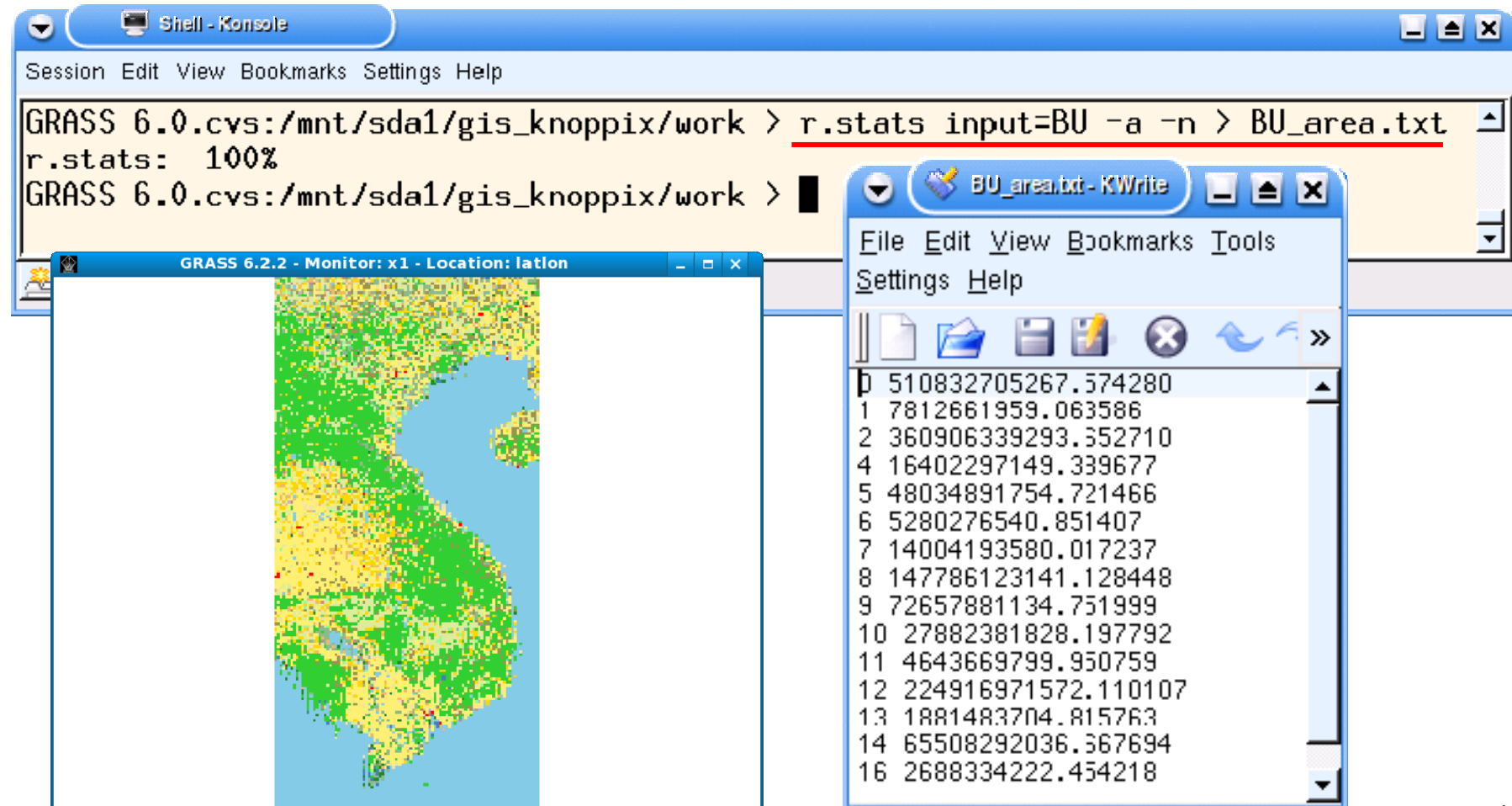
- A Print averaged values
- a Print area totals
- c Print cell counts
- p Print approximate percents
- l Print category labels
- n Suppress reporting of any NULLs

11.3 Export area statistics for raster data in text format

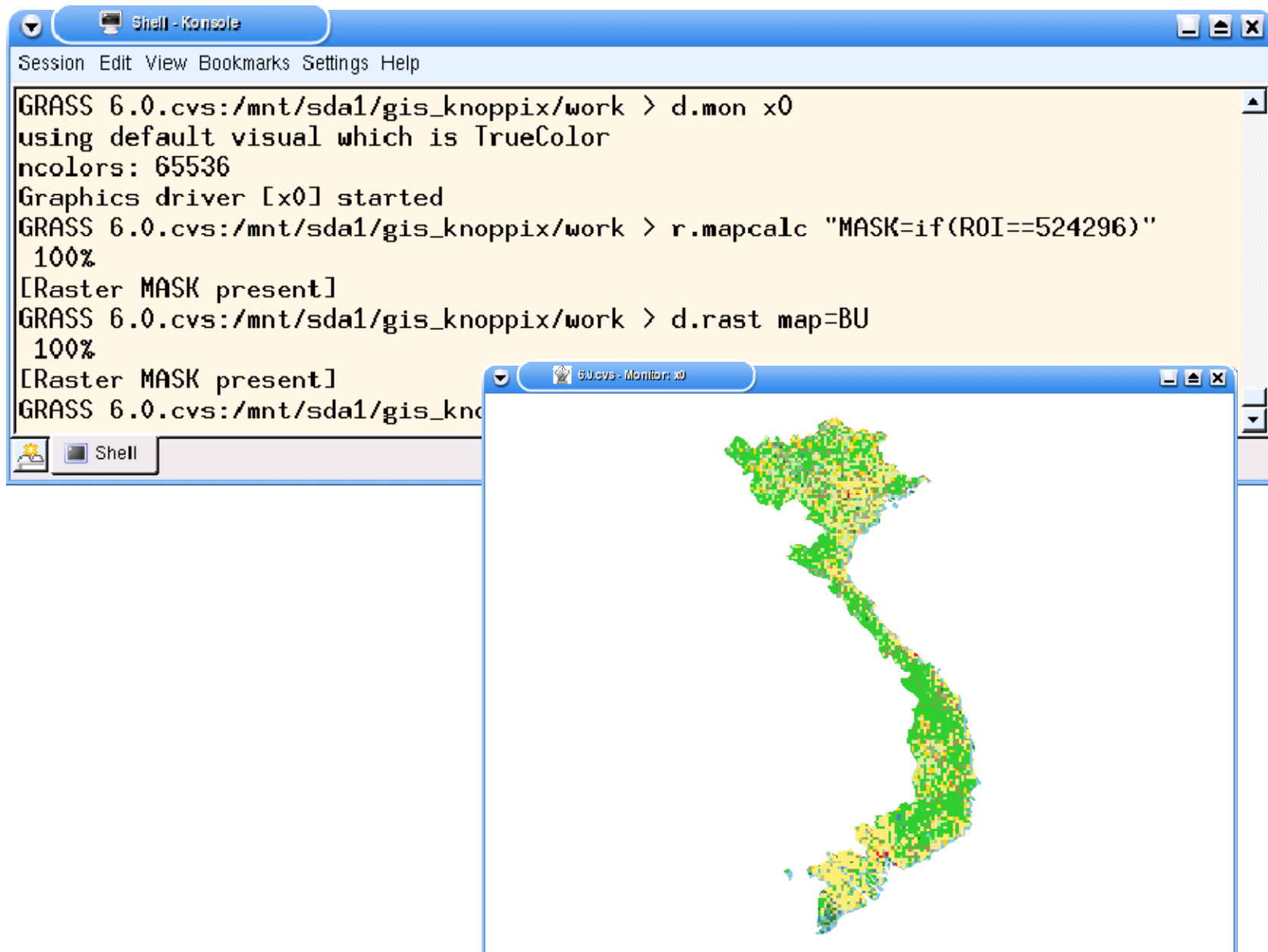
```
> r.stats input = A -a -n > B.txt
```

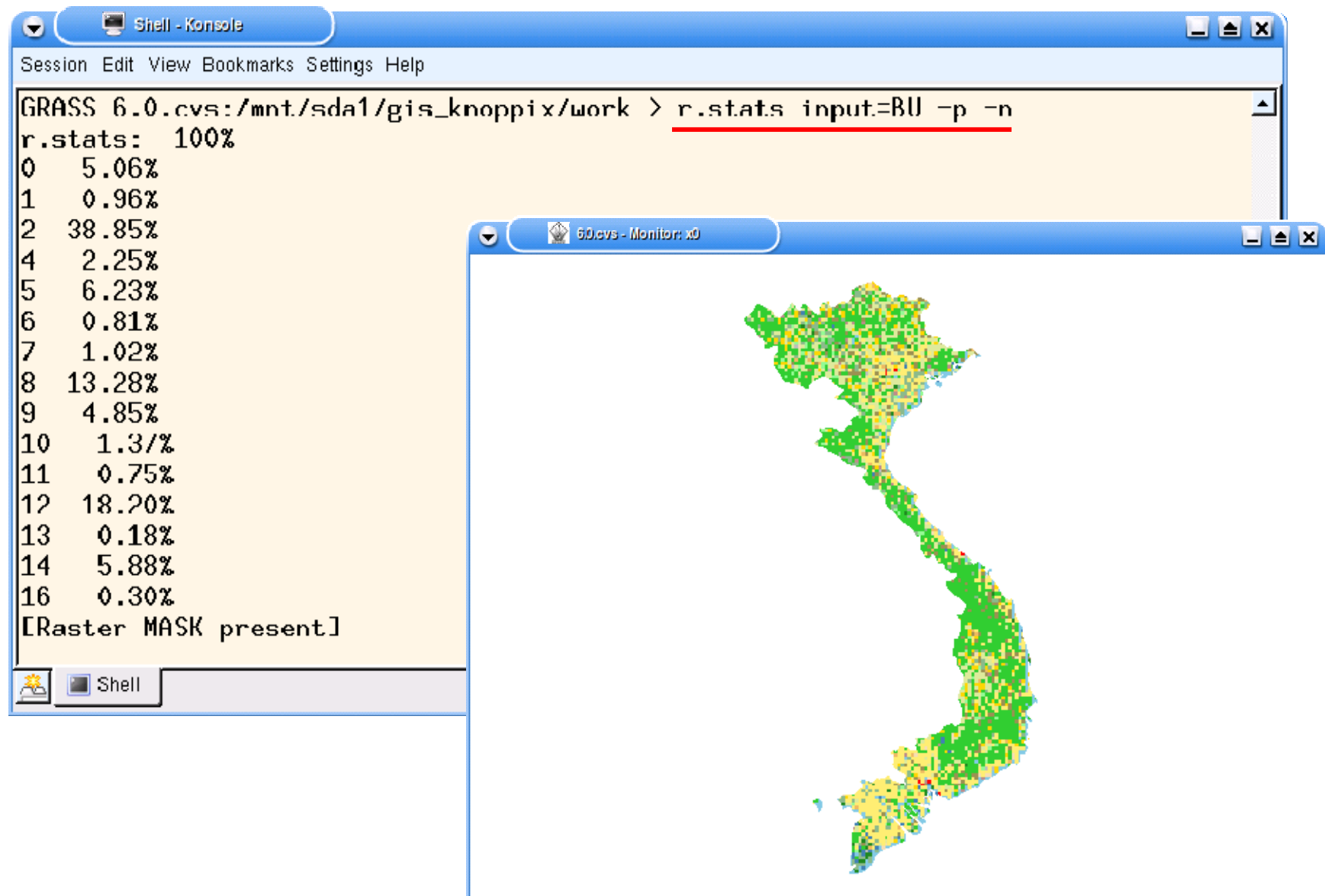
A = (input file name)

B = (output file name)



11.4 Generates area statistics for raster data with mask

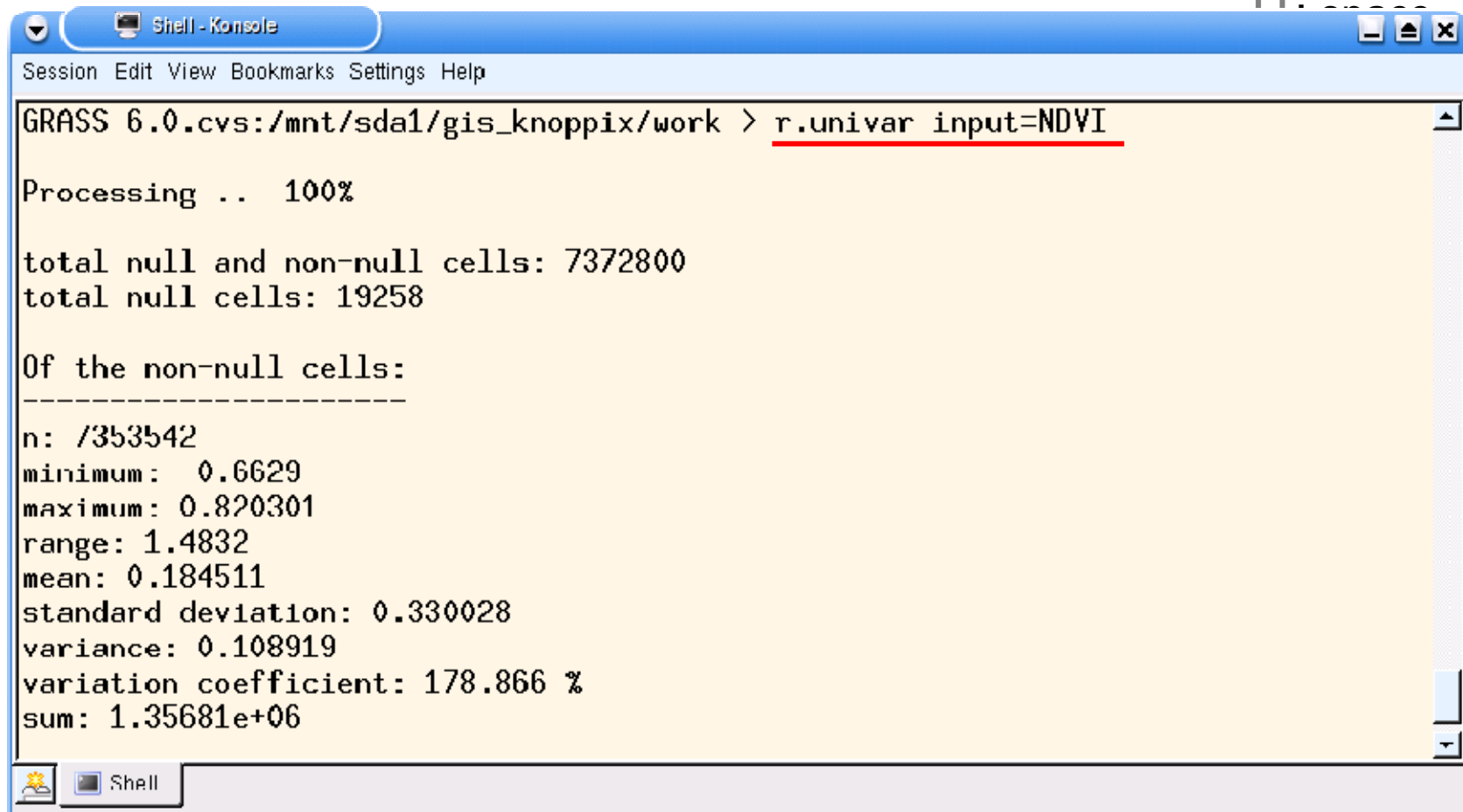




11.5 Calculation of univariate statistics from a raster map

> r.univar input = A

A = (input file name)



```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > r.univar input=NDVI

Processing .. 100%

total null and non-null cells: 7372800
total null cells: 19258

Of the non-null cells:
-----
n: 7353542
minimum: 0.6629
maximum: 0.820301
range: 1.4832
mean: 0.184511
standard deviation: 0.330028
variance: 0.108919
variation coefficient: 178.866 %
sum: 1.35681e+06
```

Section 2. Data Processing in Latitude-Longitude Coordinate System

12. Land cover classification

12.1 Supervised Classification

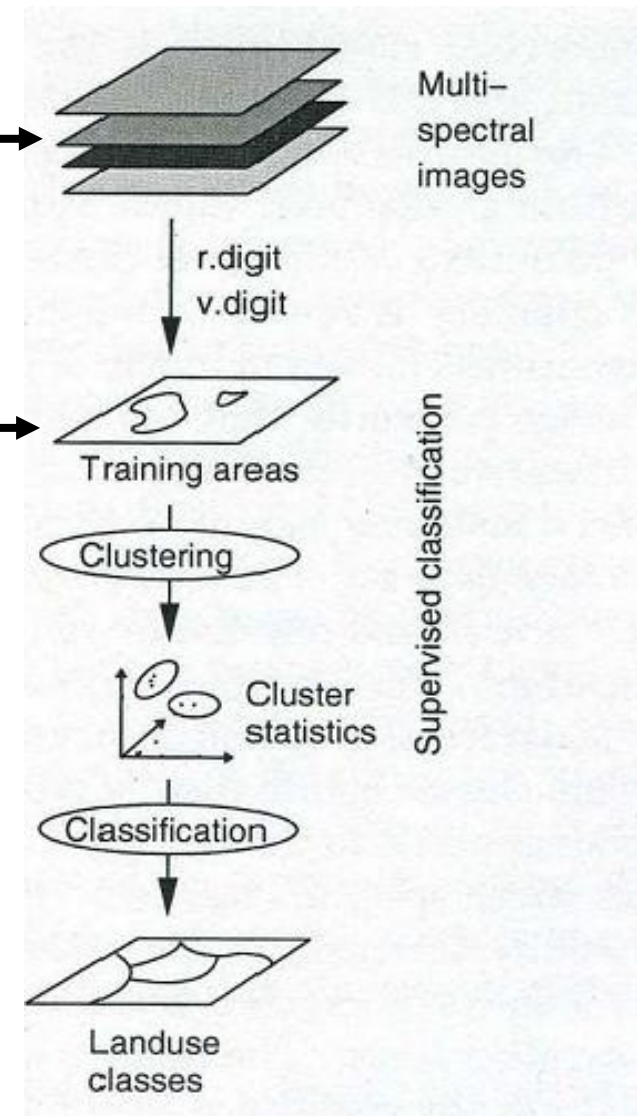
< Data List >

Images for classification

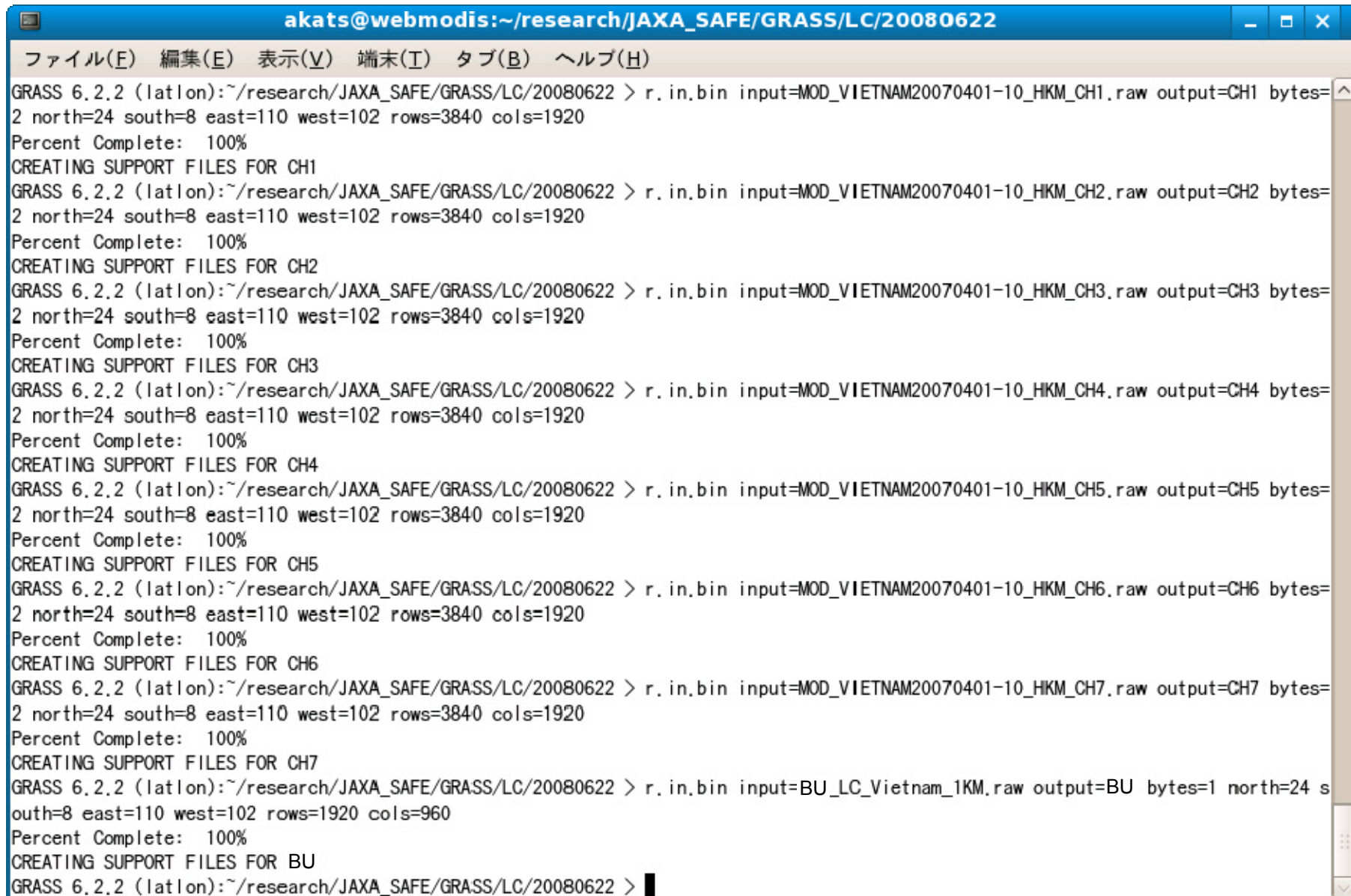
MODIS 10-days composite images
(CH1, CH2, CH3, CH4, CH5, CH6, CH7)

Training data

Land cover map of Boston University



12.1.0 Import raster data (imagery files)



The screenshot shows a terminal window titled "akats@webmodis:~/research/JAXA_SAFE/GRASS/LC/20080622". The window contains a series of GRASS GIS commands and their outputs. The commands are executed in a loop, importing seven different raster files (CH1 through CH7) and then a final file (BU). Each command specifies the input file, the output name, and the byte size. The output for each command shows the progress (Percent Complete: 100%) and the creation of support files. The final command shows the creation of support files for the BU file.

```
akats@webmodis:~/research/JAXA_SAFE/GRASS/LC/20080622
ファイル(E) 編集(E) 表示(V) 端末(T) タブ(B) ヘルプ(H)
GRASS 6.2.2 (latlon):~/research/JAXA_SAFE/GRASS/LC/20080622 > r.in.bin input=MOD_VIETNAM20070401-10_HKM_CH1.raw output=CH1 bytes=
2 north=24 south=8 east=110 west=102 rows=3840 cols=1920
Percent Complete: 100%
CREATING SUPPORT FILES FOR CH1
GRASS 6.2.2 (latlon):~/research/JAXA_SAFE/GRASS/LC/20080622 > r.in.bin input=MOD_VIETNAM20070401-10_HKM_CH2.raw output=CH2 bytes=
2 north=24 south=8 east=110 west=102 rows=3840 cols=1920
Percent Complete: 100%
CREATING SUPPORT FILES FOR CH2
GRASS 6.2.2 (latlon):~/research/JAXA_SAFE/GRASS/LC/20080622 > r.in.bin input=MOD_VIETNAM20070401-10_HKM_CH3.raw output=CH3 bytes=
2 north=24 south=8 east=110 west=102 rows=3840 cols=1920
Percent Complete: 100%
CREATING SUPPORT FILES FOR CH3
GRASS 6.2.2 (latlon):~/research/JAXA_SAFE/GRASS/LC/20080622 > r.in.bin input=MOD_VIETNAM20070401-10_HKM_CH4.raw output=CH4 bytes=
2 north=24 south=8 east=110 west=102 rows=3840 cols=1920
Percent Complete: 100%
CREATING SUPPORT FILES FOR CH4
GRASS 6.2.2 (latlon):~/research/JAXA_SAFE/GRASS/LC/20080622 > r.in.bin input=MOD_VIETNAM20070401-10_HKM_CH5.raw output=CH5 bytes=
2 north=24 south=8 east=110 west=102 rows=3840 cols=1920
Percent Complete: 100%
CREATING SUPPORT FILES FOR CH5
GRASS 6.2.2 (latlon):~/research/JAXA_SAFE/GRASS/LC/20080622 > r.in.bin input=MOD_VIETNAM20070401-10_HKM_CH6.raw output=CH6 bytes=
2 north=24 south=8 east=110 west=102 rows=3840 cols=1920
Percent Complete: 100%
CREATING SUPPORT FILES FOR CH6
GRASS 6.2.2 (latlon):~/research/JAXA_SAFE/GRASS/LC/20080622 > r.in.bin input=MOD_VIETNAM20070401-10_HKM_CH7.raw output=CH7 bytes=
2 north=24 south=8 east=110 west=102 rows=3840 cols=1920
Percent Complete: 100%
CREATING SUPPORT FILES FOR CH7
GRASS 6.2.2 (latlon):~/research/JAXA_SAFE/GRASS/LC/20080622 > r.in.bin input=BU_LC_Vietnam_1KM.raw output=BU bytes=1 north=24 s
outh=8 east=110 west=102 rows=1920 cols=960
Percent Complete: 100%
CREATING SUPPORT FILES FOR BU
GRASS 6.2.2 (latlon):~/research/JAXA_SAFE/GRASS/LC/20080622 > █
```

12.1.1 Create group and subgroup of imagery files

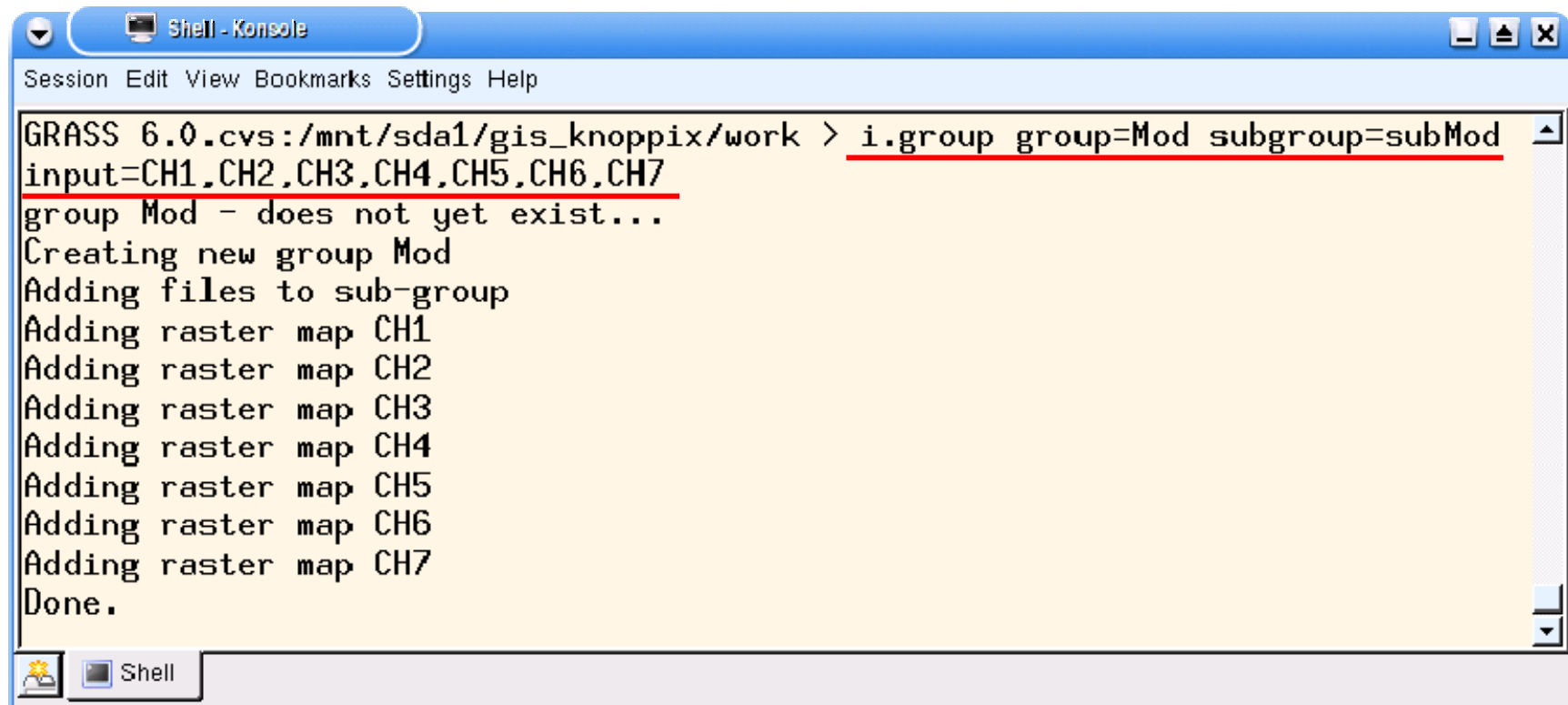
```
> i.group [group=A [subgroup=B [input=C, D, ...
```

A = (name of group)

B = (name of subgroup)

C, D, ... = (name of input data)

[] : space



```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > i.group group=Mod subgroup=subMod  
input=CH1,CH2,CH3,CH4,CH5,CH6,CH7  
group Mod - does not yet exist...  
Creating new group Mod  
Adding files to sub-group  
Adding raster map CH1  
Adding raster map CH2  
Adding raster map CH3  
Adding raster map CH4  
Adding raster map CH5  
Adding raster map CH6  
Adding raster map CH7  
Done.
```

12.1.2 Generates statistics from raster training data

```
> i.gensig trainingmap = A group = B subgroup = C signaturefile = D
```

A = (name of training map)

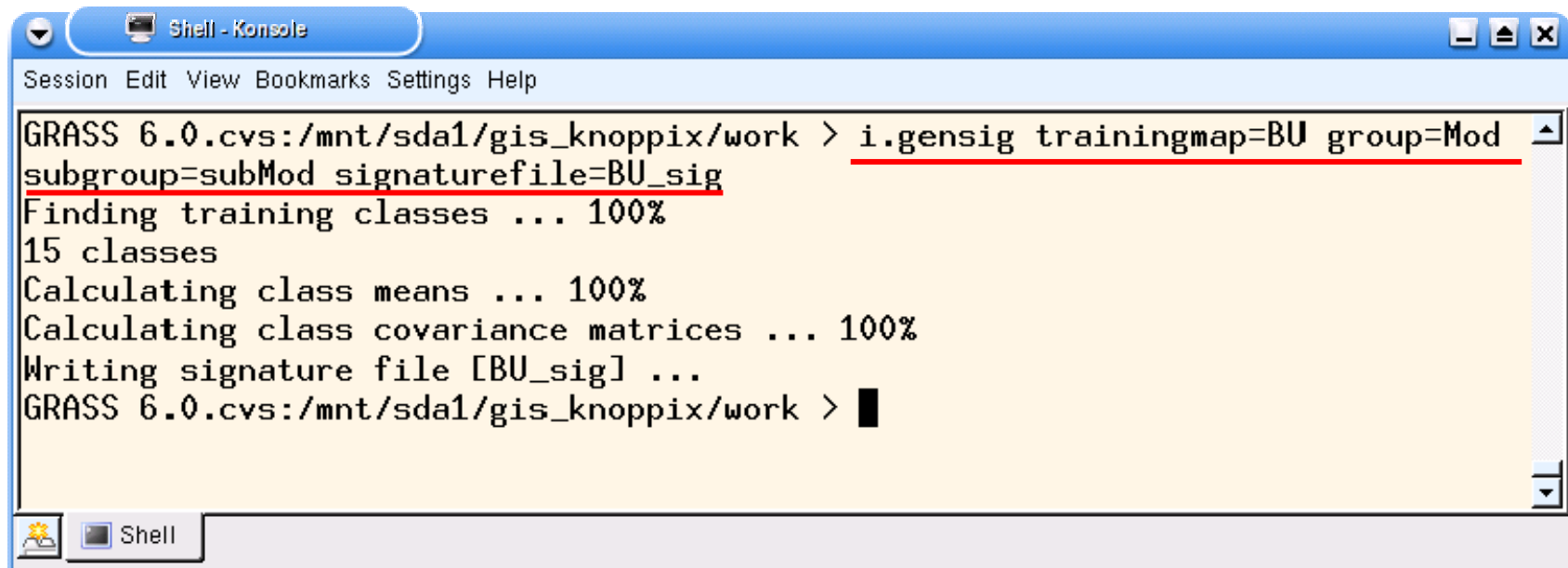
B = (name of image group)

C = (name of subgroup)

D = (output file name (signature file))

: space

i.gensig is a non-interactive method for generating input into *i.maxlik*.



```
Shell - Konsole
Session Edit View Bookmarks Settings Help
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > i.gensig trainingmap=BU group=Mod
subgroup=subMod signaturefile=BU_sig
Finding training classes ... 100%
15 classes
Calculating class means ... 100%
Calculating class covariance matrices ... 100%
Writing signature file [BU_sig] ...
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > █
```

12.1.3 Classify images by Maximum Likelihood method

```
> i.maxlik group = A subgroup = B sigfile = C class = D
```

A = (name of image group)

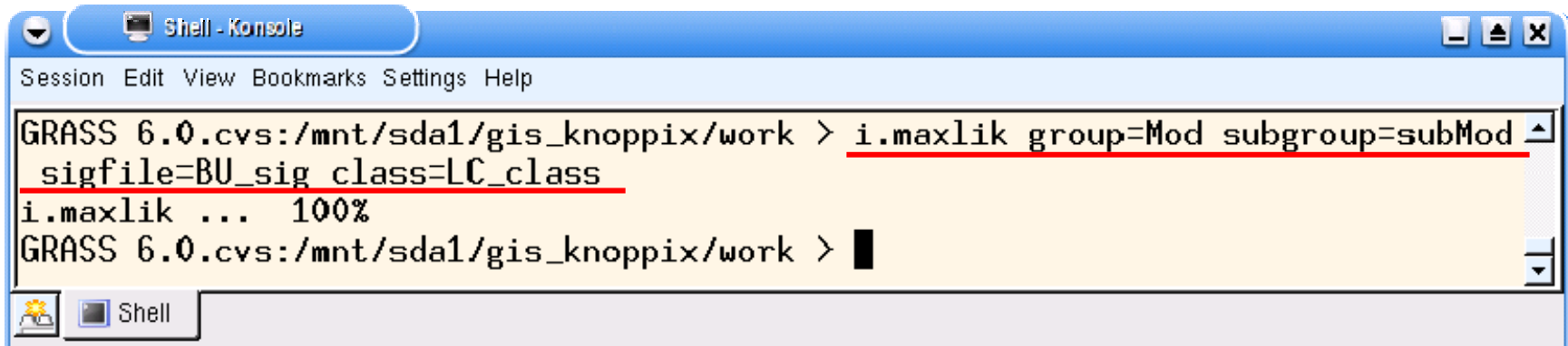
B = (name of subgroup)

C = (name of signaturefile)

D = (output file name)

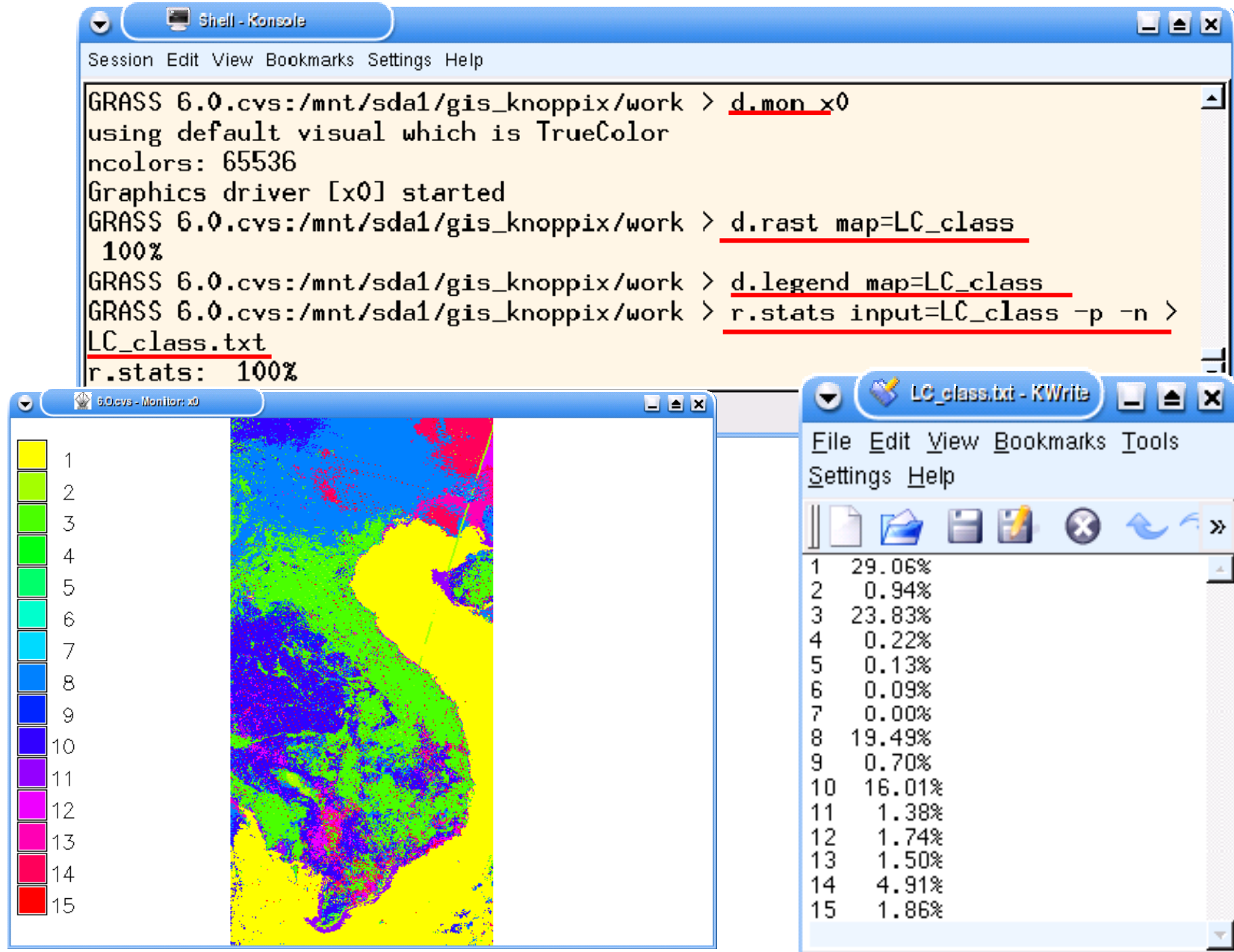
: space

i.maxlik is a maximum-likelihood discriminant analysis classifier.



```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > i.maxlik group=Mod subgroup=subMod  
sigfile=BU_sig class=LC_class  
i.maxlik ... 100%  
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > █
```

12.1.4 Display classification result



12.2 Supervised Classification using Mask

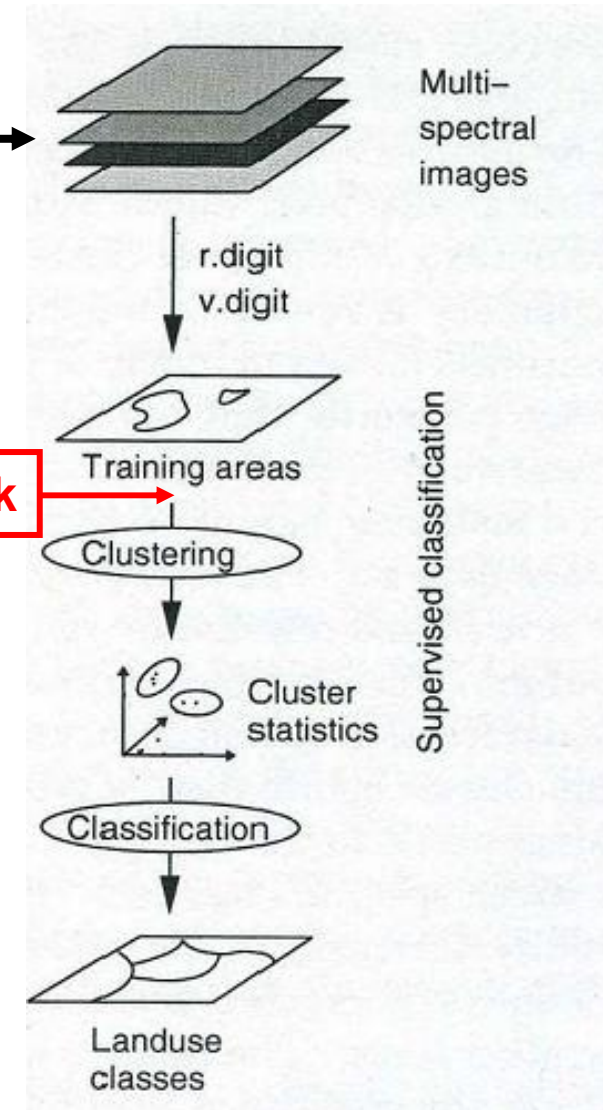
< Data List >

Images for classification

MODIS 10-days composite images
(CH1, CH2, CH3, CH4, CH5, CH6, CH7)

Images for Masking

- Land Sea Mask



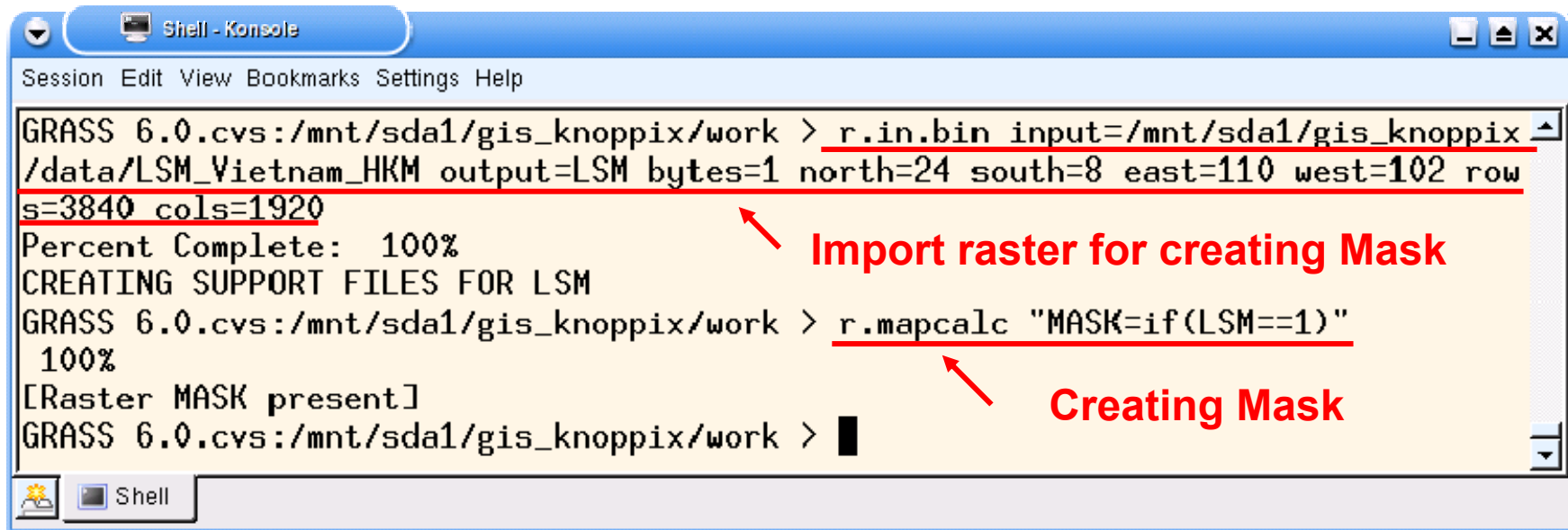
12.2.0 Import raster data (imagery files)

⇒ see “12.1.0”

12.2.1 Create group and subgroup of imagery files

⇒ see “12.1.1”

12.2.2 Import raster data and create mask



The screenshot shows a terminal window titled "Shell - Konsole" with a menu bar (Session, Edit, View, Bookmarks, Settings, Help). The terminal output shows the execution of two GRASS GIS commands. The first command, `r.in.bin`, imports a raster from a binary file. The second command, `r.mapcalc`, creates a mask based on the imported raster. Red arrows point from text labels to the respective commands in the terminal output.

```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > r.in.bin input=/mnt/sda1/gis_knoppix  
/data/LSM_Vietnam_HKM output=LSM bytes=1 north=24 south=8 east=110 west=102 row  
s=3840 cols=1920  
Percent Complete: 100%  
CREATING SUPPORT FILES FOR LSM  
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > r.mapcalc "MASK=if(LSM==1)"  
100%  
[Raster MASK present]  
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > █
```

Import raster for creating Mask

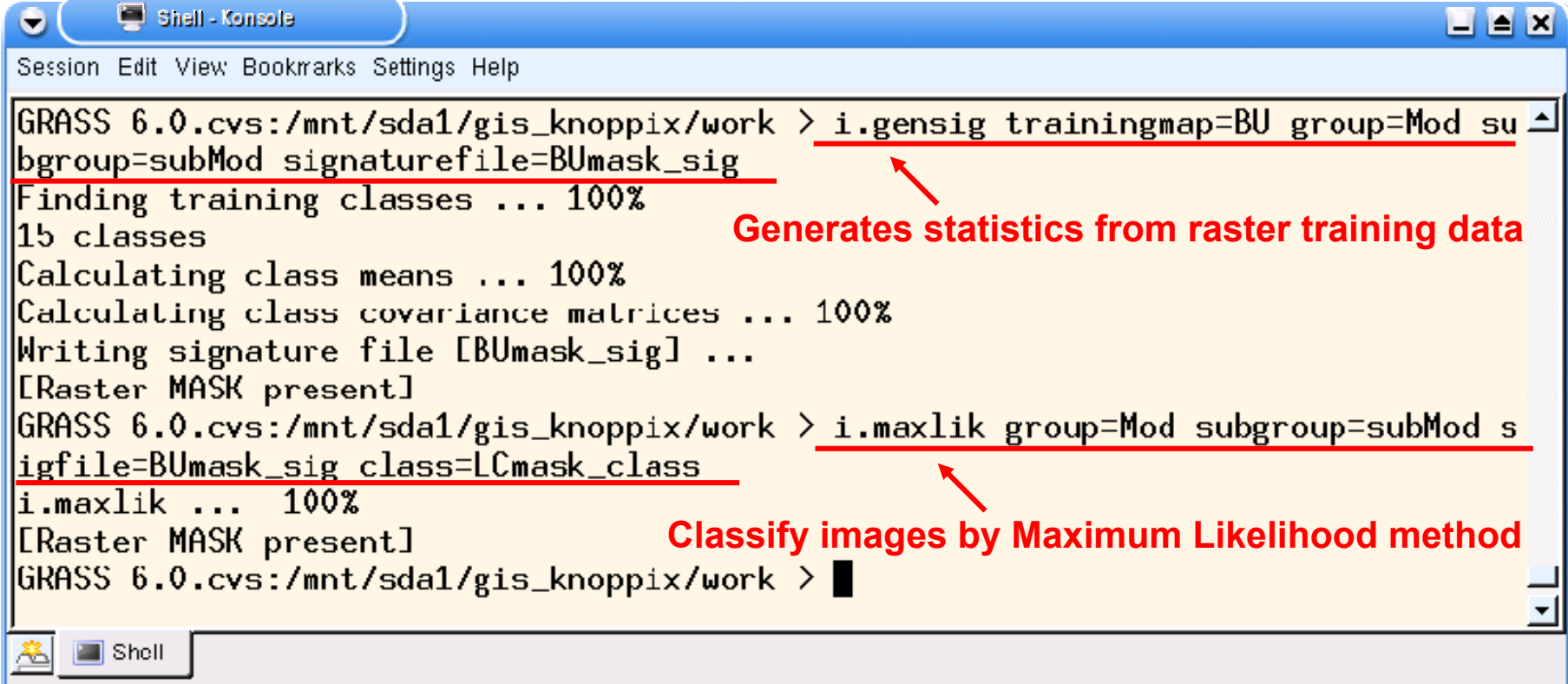
Creating Mask

12.2.3 Generates statistics from raster training data

⇒ see “12.1.2”

12.2.4 Classify images by Maximum Likelihood method

⇒ see “12.1.3”

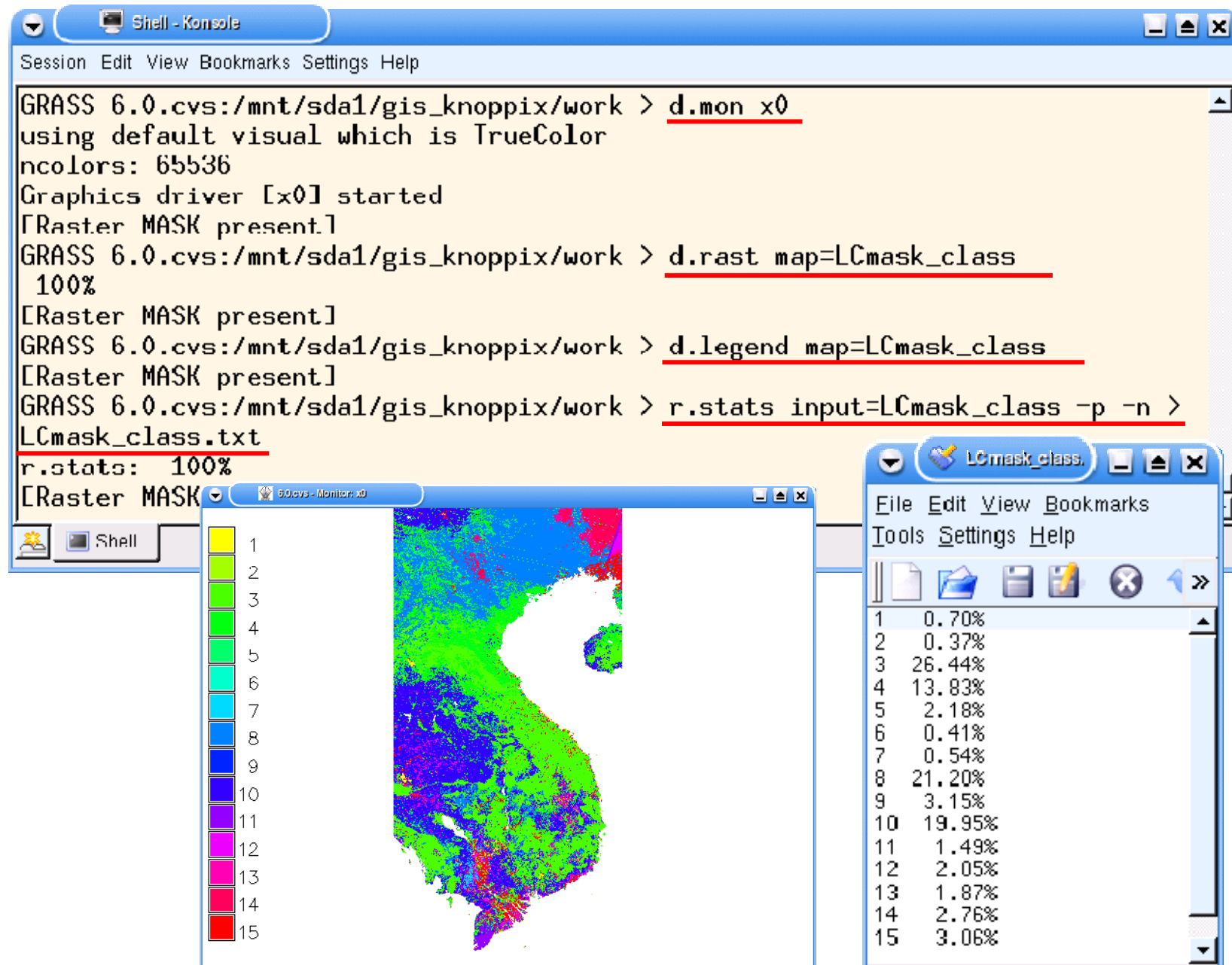


```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > i.gensig trainingmap=BU group=Mod sub  
group=subMod signaturefile=BUmask_sig  
Finding training classes ... 100%  
15 classes  
Calculating class means ... 100%  
Calculating class covariance matrices ... 100%  
Writing signature file [BUmask_sig] ...  
[Raster MASK present]  
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > i.maxlik group=Mod subgroup=subMod s  
igfile=BUmask_sig class=LCmask_class  
i.maxlik ... 100%  
[Raster MASK present]  
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > █
```

Generates statistics from raster training data

Classify images by Maximum Likelihood method

12.2.5 Display classification result



12.2.6 Identification of land cover classes

Legend of Training Data
(LC map of Boston Univ.)
(see 11.1)

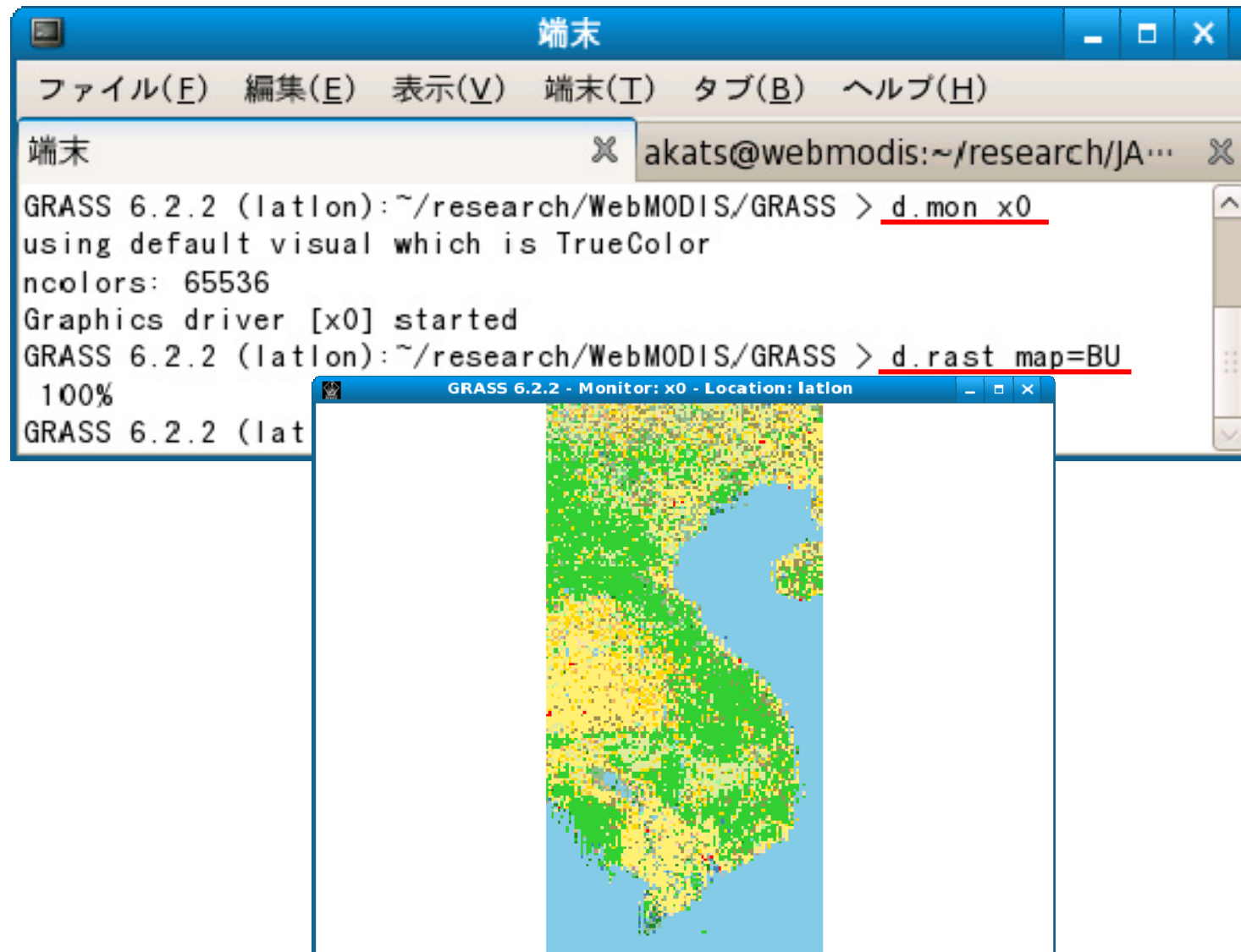
Legend of
classification result
(see 12.2.6)

Legend of LC map from Boston Univ.
0 : Water (and Goode's interrupted space)
1 : Evergreen Needleleaf Forest
2 : Evergreen Broadleaf Forest
3 : Deciduous Needleleaf Forest
4 : Deciduous Broadleaf Forest
5 : Mixed Forest
6 : Closed Shrublands
7 : Open Shrubland
8 : Woody Savannas
9 : Savannas
10 : Grassland
11 : Permanent Wetlands
12 : Croplands
13 : Urban and Built-up
14 : Cropland/Natural Vegetation Mosaic
15 : Snow and Ice
16 : Barren or Sparsely Vegetated

0	→	1 : Water
1	→	2 : ENF
2	→	3 : EBF
4	→	4 : DBF
5	→	5 : MF
6	→	6 : C-Shrub
7	→	7 : O-Shrub
8	→	8 : W-Savanna
9	→	9 : Savanna
10	→	10 : Grassland
11	→	11 : P-Wetland
12	→	12 : Cropland
13	→	13 : Urban
14	→	14 : C/N Mosaic
16	→	15 : Barren

12.3 Generating training areas from raster data

12.3.1 display raster data



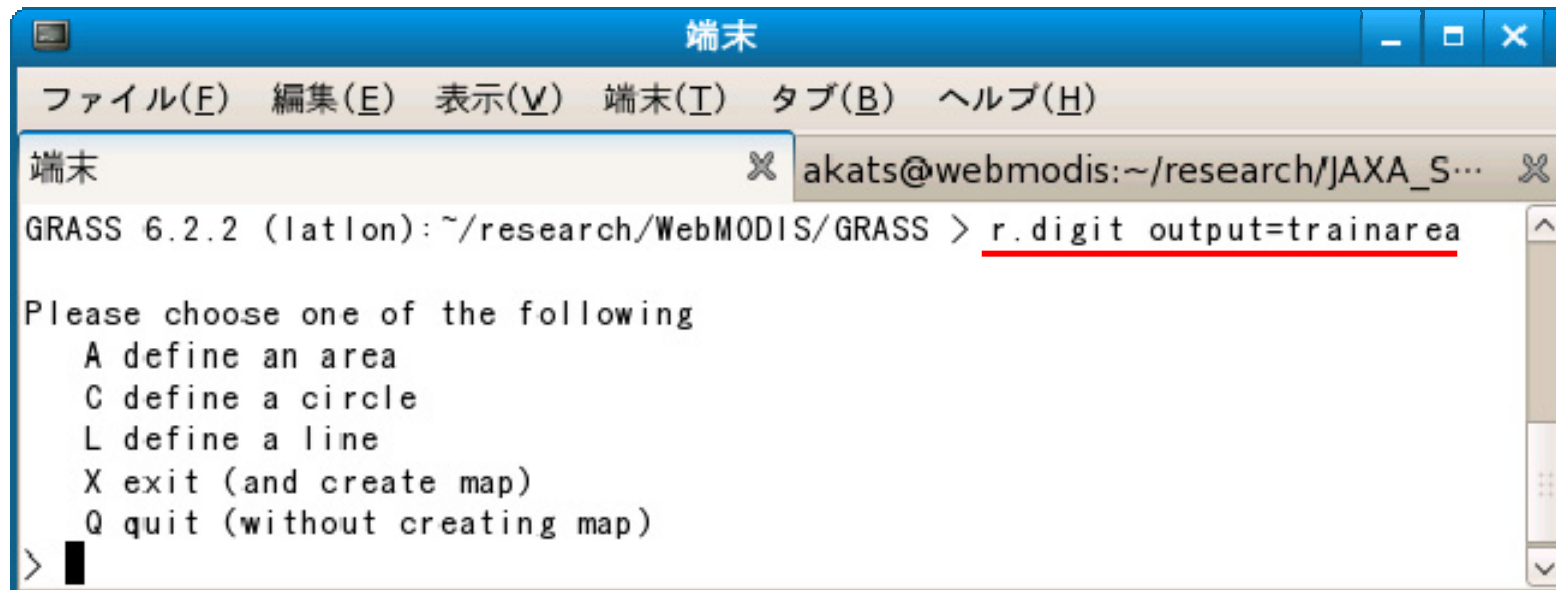
12.3.2 Select training area

□ : space

```
> r.digit □ output = A
```

A = (output file name)

The *r.digit* provides the user with a way to draw lines, areas, and circles on a monitor screen, and to save these features in a raster map. Lines, areas, and circles are to be drawn using a pointing device (mouse).



```
端末
ファイル(F) 編集(E) 表示(V) 端末(T) タブ(B) ヘルプ(H)
端末 akats@webmodis:~/research/JAXA_S...
GRASS 6.2.2 (latlon): ~/research/WebMODIS/GRASS > r.digit output=trainarea
Please choose one of the following
  A define an area
  C define a circle
  L define a line
  X exit (and create map)
  Q quit (without creating map)
> █
```

enter “ A “, then hit "Enter (or Return)"

12.3.3 Definition of area

The image shows two windows from the GRASS 6.2.2 GIS environment. The left window is a terminal titled '端末' (Terminal) showing the command prompt and the execution of `r.digit output=trainarea`. It displays the interactive menu for defining an area, with the 'a' option selected. A red dashed box highlights the 'Buttons:' section, and a red arrow points to it with the text 'How to select area'. The terminal also shows a list of coordinates and prompts for category and label. The right window is a monitor titled 'GRASS 6.2.2 - Monitor: x0 - Location: latlon' showing a map of a coastal area. A red dashed polygon is drawn on the map, with red stars at its vertices. Arrows point from the text 'click middle button' to the stars, and a large downward arrow points from the text 'click right button'.

端末

ファイル(F) 編集(E) 表示(V) 端末(T) タブ(B) ヘルプ(H)

端末 akats@webmodis:~/research/JAXA_SA...

GRASS 6.2.2 (latlon): ~/research/WebMODIS/GRASS > r.digit output=trainarea

Please choose one of the following

- A define an area
- C define a circle
- L define a line
- X exit (and create map)
- Q quit (without creating map)

> a

Buttons:

- Left: where am i
- Middle: mark point
- Right: done

How to select area

EAST: 106:54E NORTH: 19:26N
EAST: 107E NORTH: 18:16N
EAST: 108:02E NORTH: 18:28N
EAST: 108E NORTH: 19:30N
EAST: 106:52E NORTH: 19:28N

Enter the category number for this area: 1
Enter a label for category 1 [] cat1
1 [cat1]
Look ok? (y/n) [y] y

Please choose one of the following

- A define an area
- C define a circle
- L define a line
- X exit (and create map)
- Q quit (without creating map)

GRASS 6.2.2 - Monitor: x0 - Location: latlon

click middle button

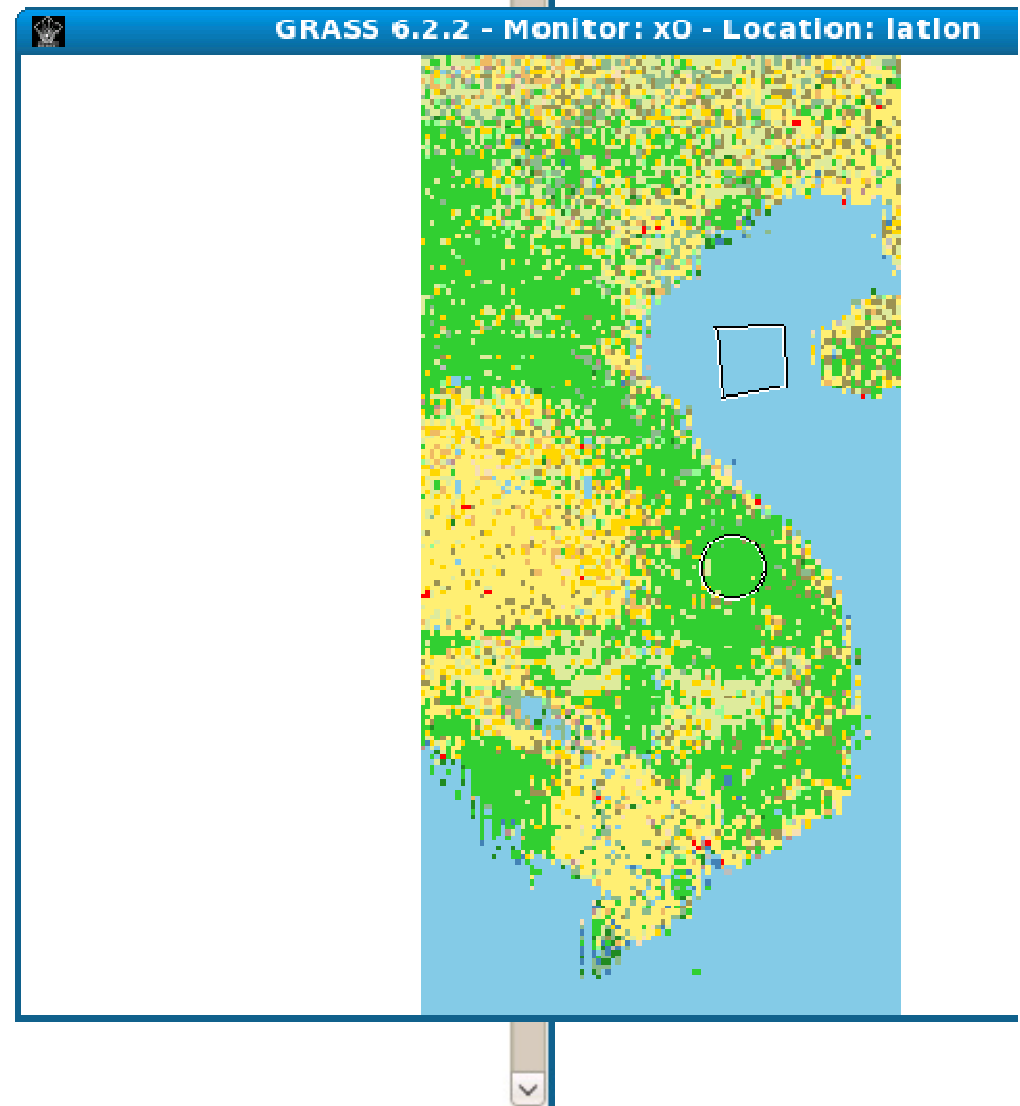
click right button

```
端末
ファイル(F) 編集(E) 表示(V) 端末(T) タブ(B) ヘルプ(H)
端末 akats@webmodis:~/research/JAXA_SA...
Please choose one of the following
  A define an area
  C define a circle
  L define a line
  X exit (and create map)
  Q quit (without creating map)
> c
Mark the center of the circle
Buttons:
  Left:  where am i
  Middle: mark point
  Right: done

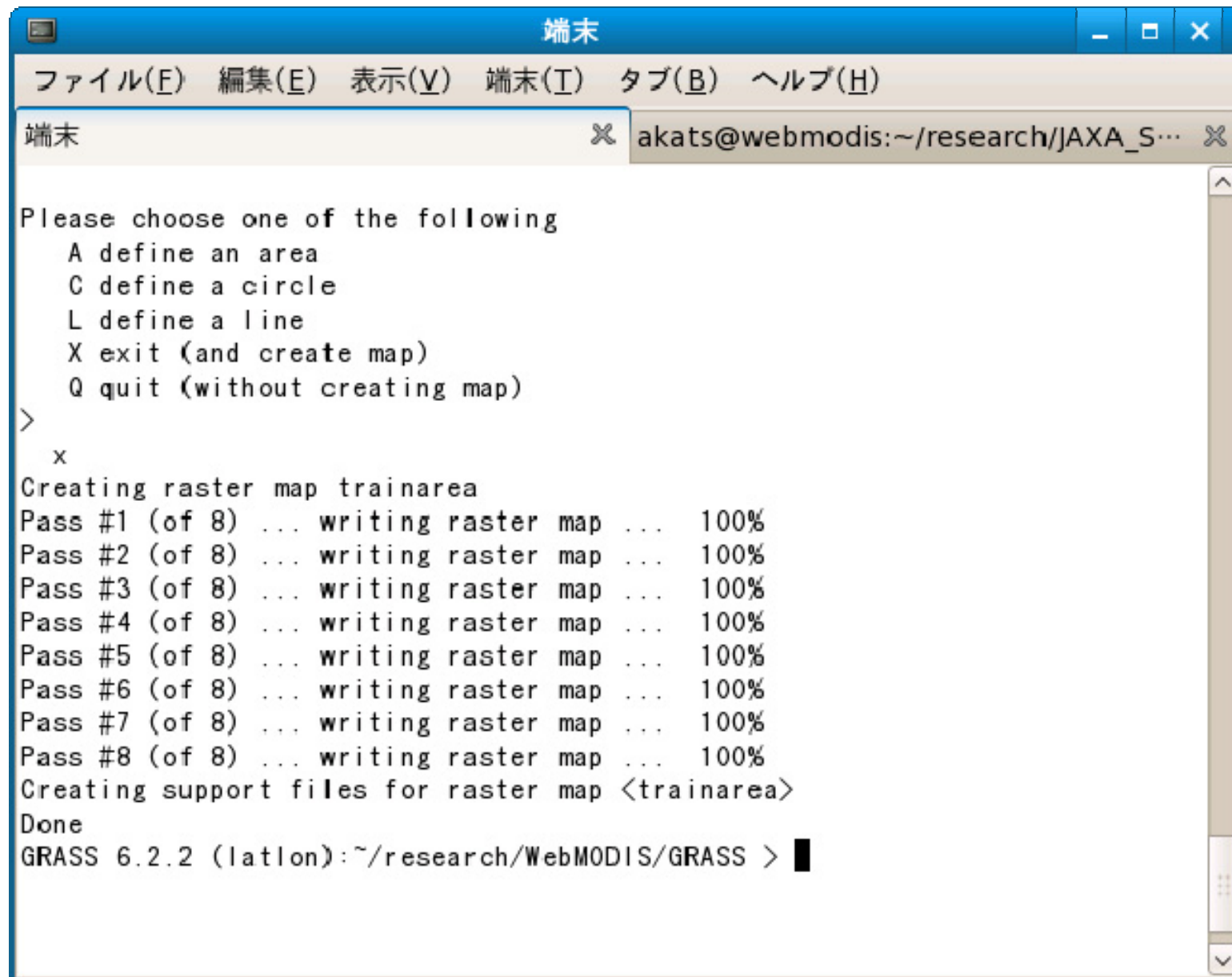
EAST: 107:10E          NORTH: 15:28N
Mark a point on the perimeter
Buttons:
  Left:  where am i
  Middle: mark point
  Right: done

EAST: 107:22E          NORTH: 14:58N
Enter the category number for this circle: 2
Enter a label for category 2 [] cat2
2 [cat2]
Look ok? (y/n) [y] y

Please choose one of the following
  A define an area
  C define a circle
```

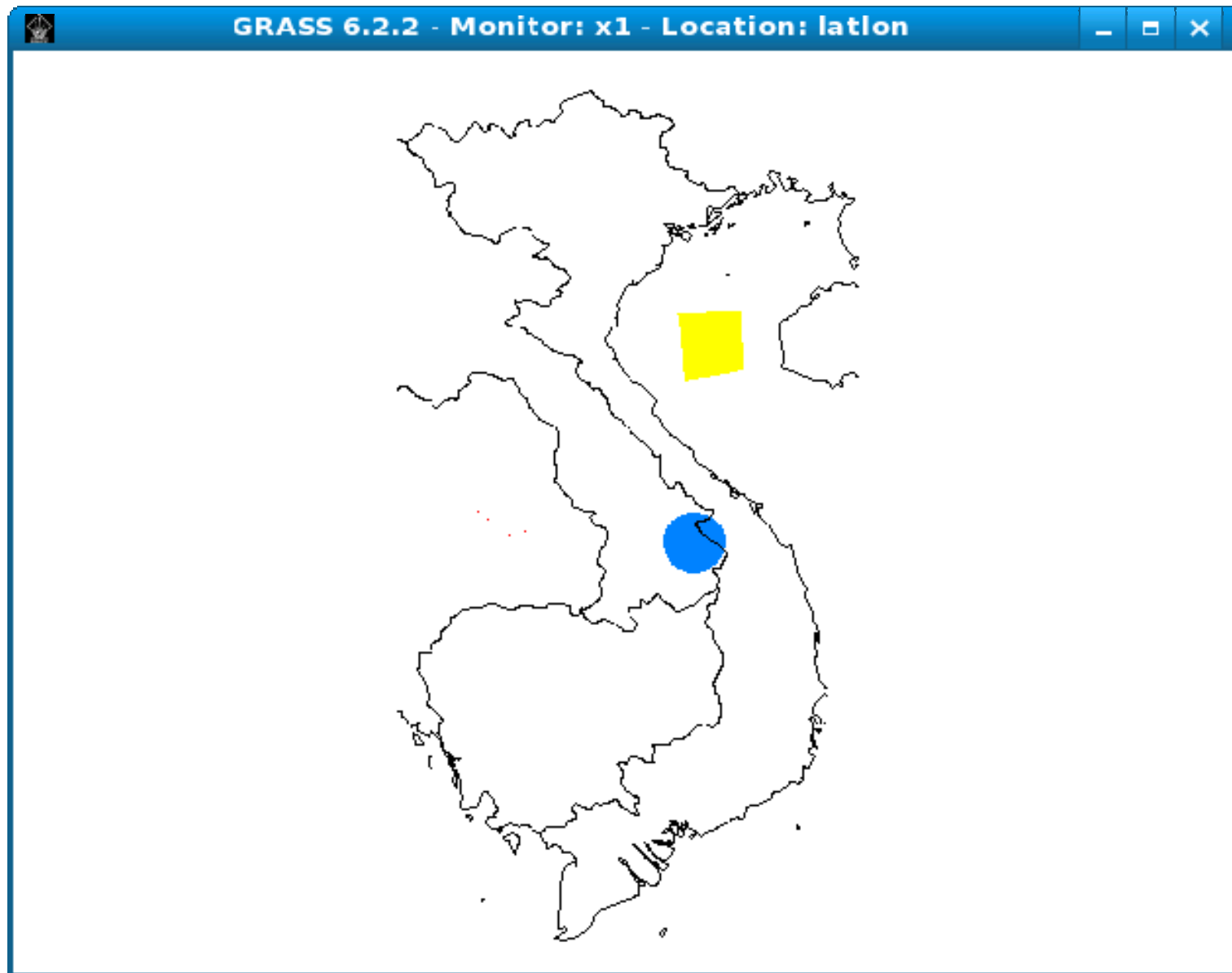


12.3.4 Exit and creating map



The screenshot shows a terminal window titled "端末" (Terminal) with a menu bar containing "ファイル(F)", "編集(E)", "表示(V)", "端末(T)", "タブ(B)", and "ヘルプ(H)". The terminal displays the following text:

```
端末
Please choose one of the following
  A define an area
  C define a circle
  L define a line
  X exit (and create map)
  Q quit (without creating map)
>
x
Creating raster map trainarea
Pass #1 (of 8) ... writing raster map ... 100%
Pass #2 (of 8) ... writing raster map ... 100%
Pass #3 (of 8) ... writing raster map ... 100%
Pass #4 (of 8) ... writing raster map ... 100%
Pass #5 (of 8) ... writing raster map ... 100%
Pass #6 (of 8) ... writing raster map ... 100%
Pass #7 (of 8) ... writing raster map ... 100%
Pass #8 (of 8) ... writing raster map ... 100%
Creating support files for raster map <trainarea>
Done
GRASS 6.2.2 (latlon): ~/research/WebMODIS/GRASS > █
```



12.4 Unsupervised Classification using Mask

< Data List >

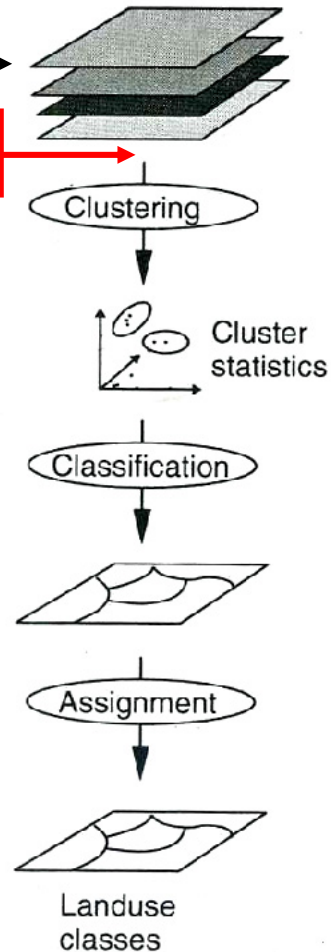
Images for classification

MODIS 10-days composite images
(CH1, CH2, CH3, CH4, CH5, CH6, CH7)

Images for Masking

- Land Sea Mask

Mask



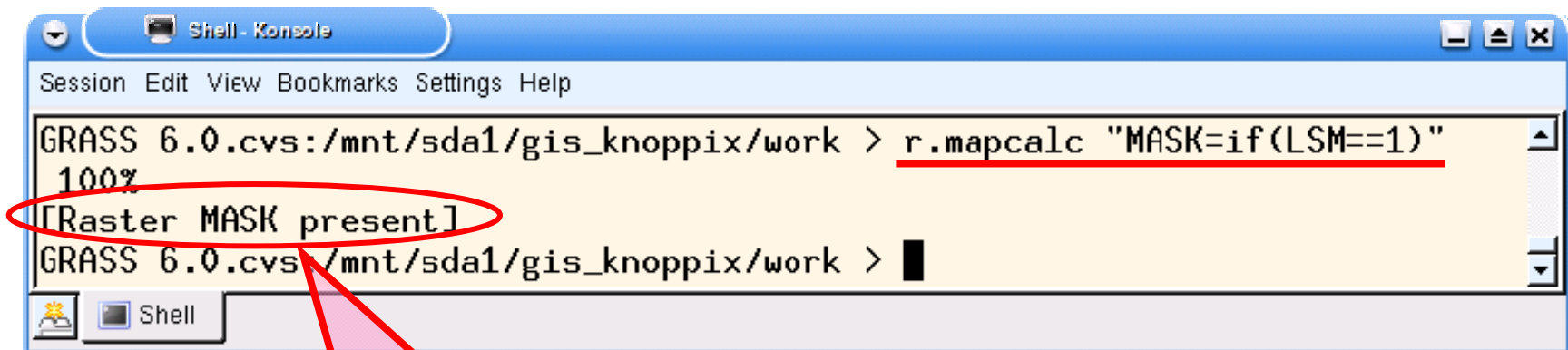
12.4.0 Import raster data (imagery files)

12.4.1 Create group and subgroup of imagery files

⇒ *see “12.1.1”*

12.4.2 Import raster data and create mask

⇒ *see “12.1.2”*



The screenshot shows a terminal window titled "Shell - Konsole". The command prompt is "GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work >". The command entered is r.mapcalc "MASK=if(LSM==1)". The output shows "100%" and "[Raster MASK present]". The text "[Raster MASK present]" is circled in red. Below the terminal window, there is a pink callout box with the text "The MASK is active".

```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > r.mapcalc "MASK=if(LSM==1)"  
100%  
[Raster MASK present]  
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > █
```

The MASK is active

12.4.3 Clustering of image data

```
> i.cluster group = A subgroup = B sigfile = C classes = D
```

A = (name of image group)

B = (name of subgroup)

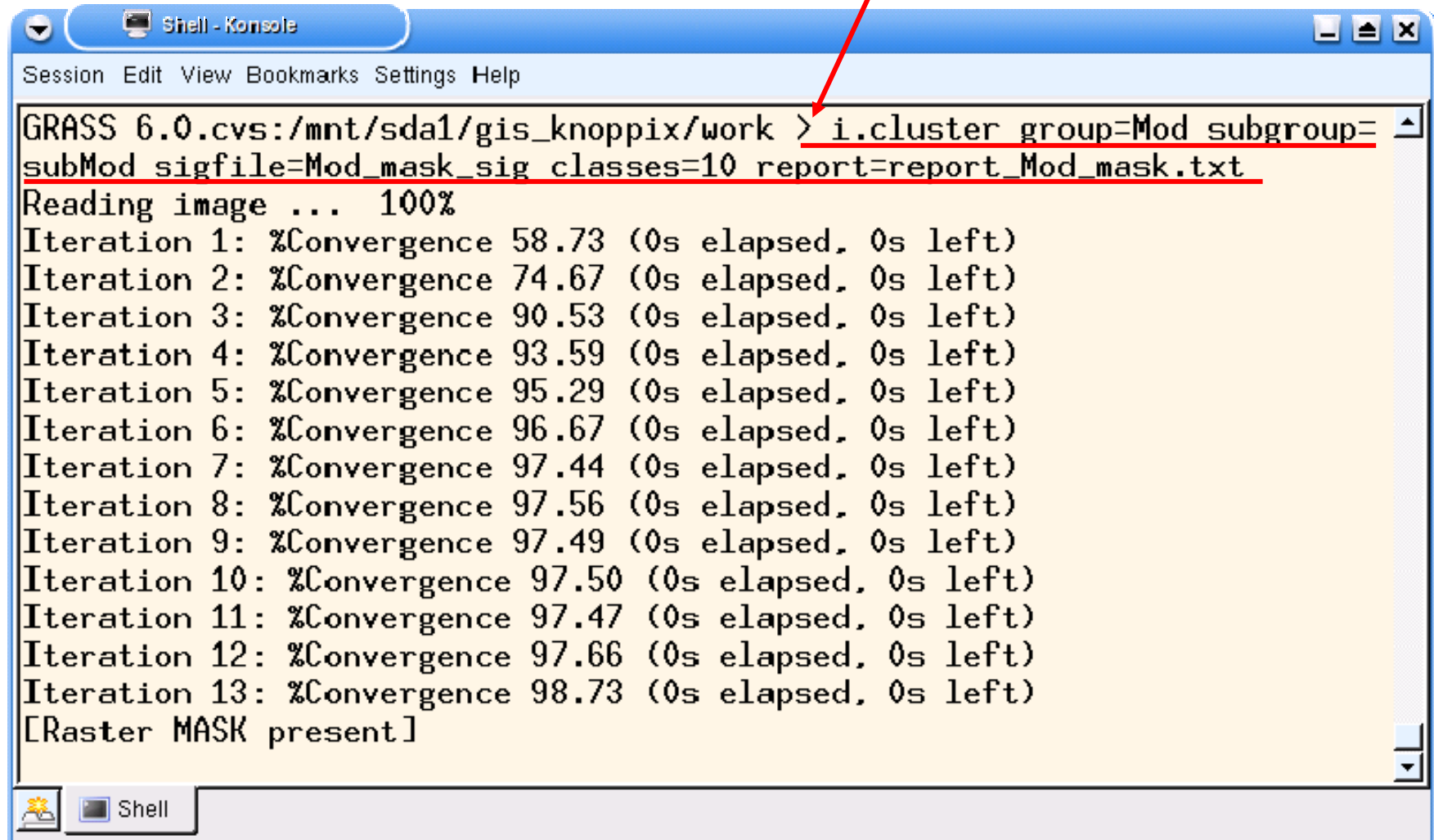
C = (File contains result signatures)

D = (Initial number of classes)

: space

i.cluster - An imagery function that generates spectral signatures for land cover types in an image using a clustering algorithm.

clustering of image data



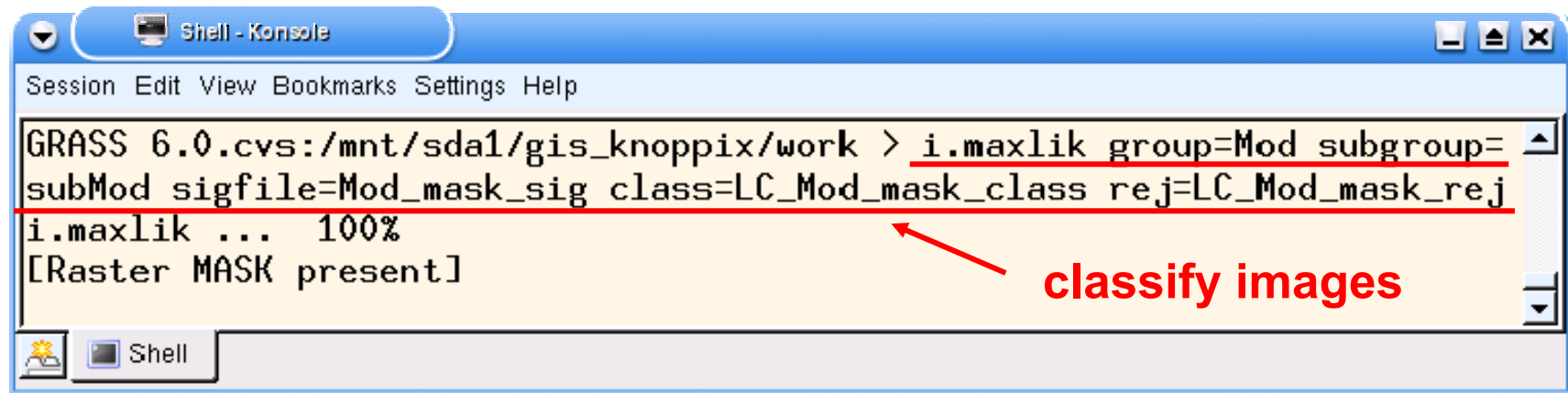
The screenshot shows a terminal window titled "Shell - Konsole" with a menu bar containing "Session", "Edit", "View", "Bookmarks", "Settings", and "Help". The terminal text is as follows:

```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > i.cluster group=Mod subgroup=  
subMod sigfile=Mod_mask_sig classes=10 report=report_Mod_mask.txt  
Reading image ... 100%  
Iteration 1: %Convergence 58.73 (0s elapsed, 0s left)  
Iteration 2: %Convergence 74.67 (0s elapsed, 0s left)  
Iteration 3: %Convergence 90.53 (0s elapsed, 0s left)  
Iteration 4: %Convergence 93.59 (0s elapsed, 0s left)  
Iteration 5: %Convergence 95.29 (0s elapsed, 0s left)  
Iteration 6: %Convergence 96.67 (0s elapsed, 0s left)  
Iteration 7: %Convergence 97.44 (0s elapsed, 0s left)  
Iteration 8: %Convergence 97.56 (0s elapsed, 0s left)  
Iteration 9: %Convergence 97.49 (0s elapsed, 0s left)  
Iteration 10: %Convergence 97.50 (0s elapsed, 0s left)  
Iteration 11: %Convergence 97.47 (0s elapsed, 0s left)  
Iteration 12: %Convergence 97.66 (0s elapsed, 0s left)  
Iteration 13: %Convergence 98.73 (0s elapsed, 0s left)  
[Raster MASK present]
```

A red arrow points from the text "clustering of image data" to the command `i.cluster` in the terminal. The command and its arguments are underlined in red in the original image. At the bottom of the window, there is a taskbar with a "Shell" icon and label.

12.4.4 Classify images by Maximum Likelihood method

→ see “12.1.3”

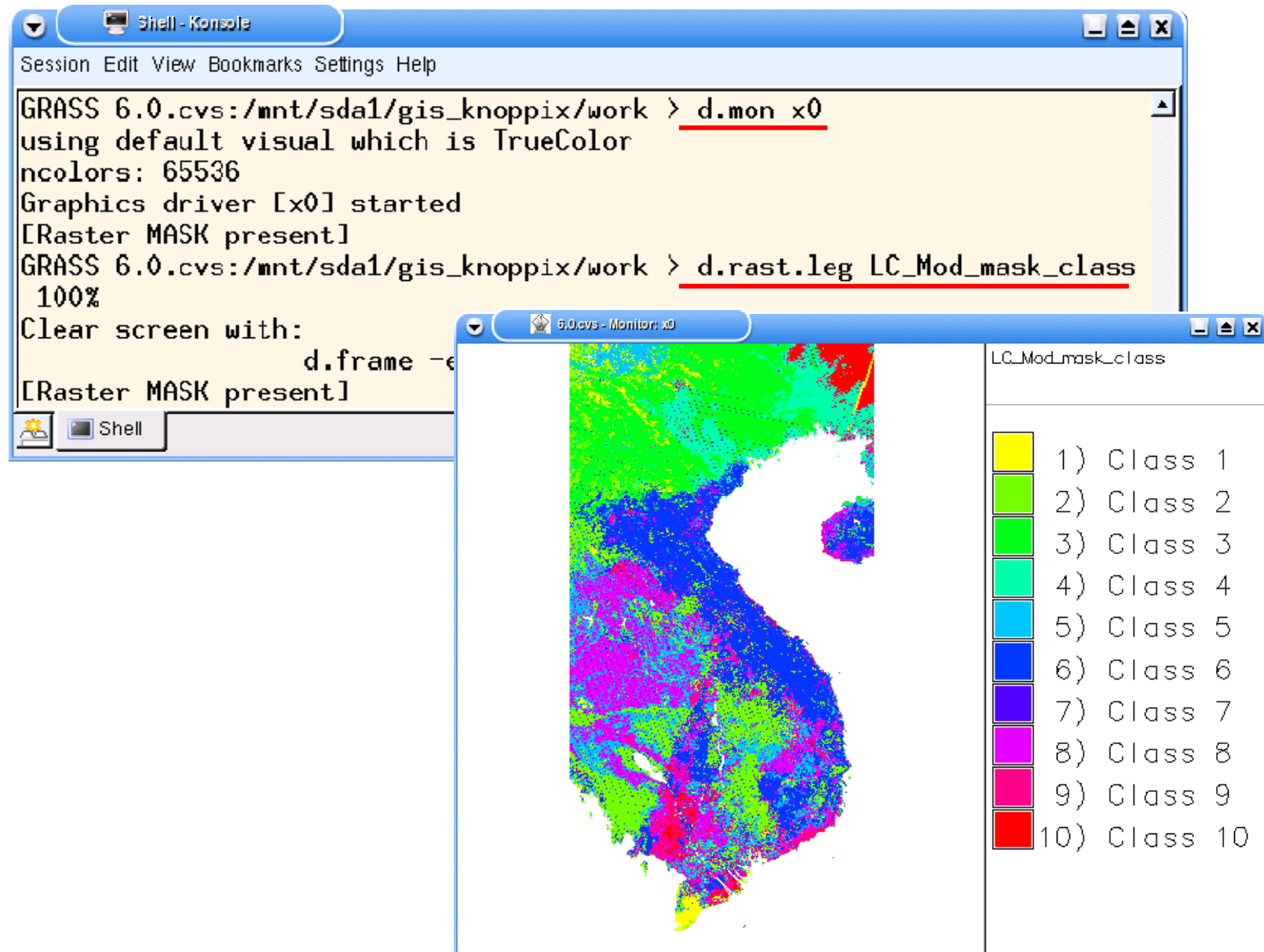


The screenshot shows a terminal window titled "Shell - Konsole". The terminal displays the following text:

```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > i.maxlik group=Mod subgroup=  
subMod sigfile=Mod_mask_sig class=LC_Mod_mask_class rej=LC_Mod_mask_rej  
i.maxlik ... 100%  
[Raster MASK present]
```

A red arrow points from the text "classify images" to the command line. The command line is underlined in red.

12.4.5 Display classification result



12.5 Reclassification of results


12.5.1 Check report statistics for original map

```
> r.report  $\square$  map = A
```

A = (original map)

\square : space

12.5.2 Reclassification

```
> r.reclass  $\square$  input = A  $\square$  output = B 
```

A = (original map)

```
> a b =  $\alpha$   $\square$  class1 
```

B = (reclassified map)

```
> c d e =  $\beta$   $\square$  class2 
```

a, b, ...
= (number of original map)

```
> f n o =  $\gamma$   $\square$  class3 
```

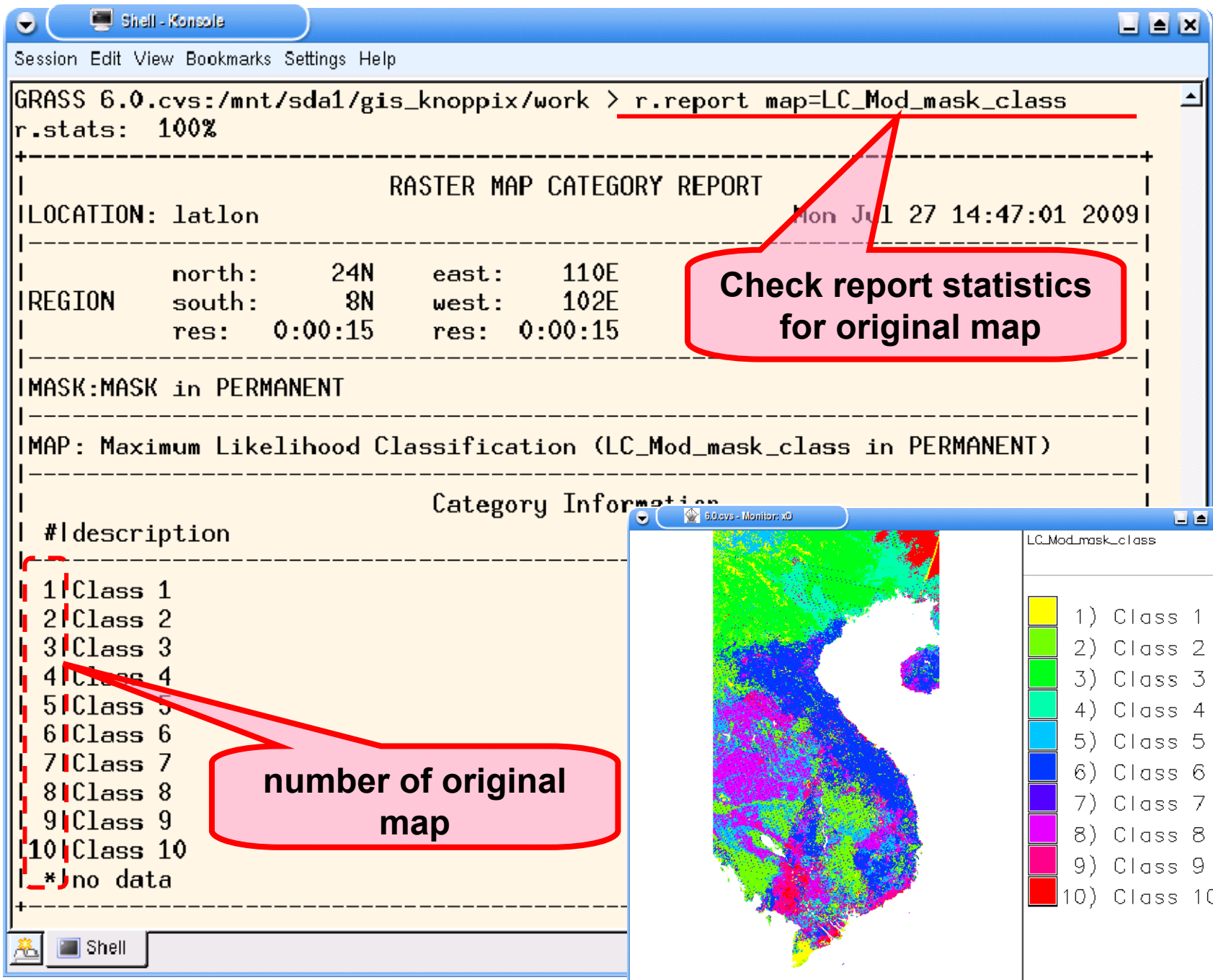
α, β, \dots
= (number of new map)

```
> ..... 
```

```
> end 
```

class1, class2, ...
= (category label of new map)

\square : space

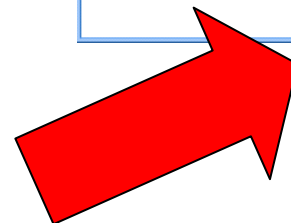
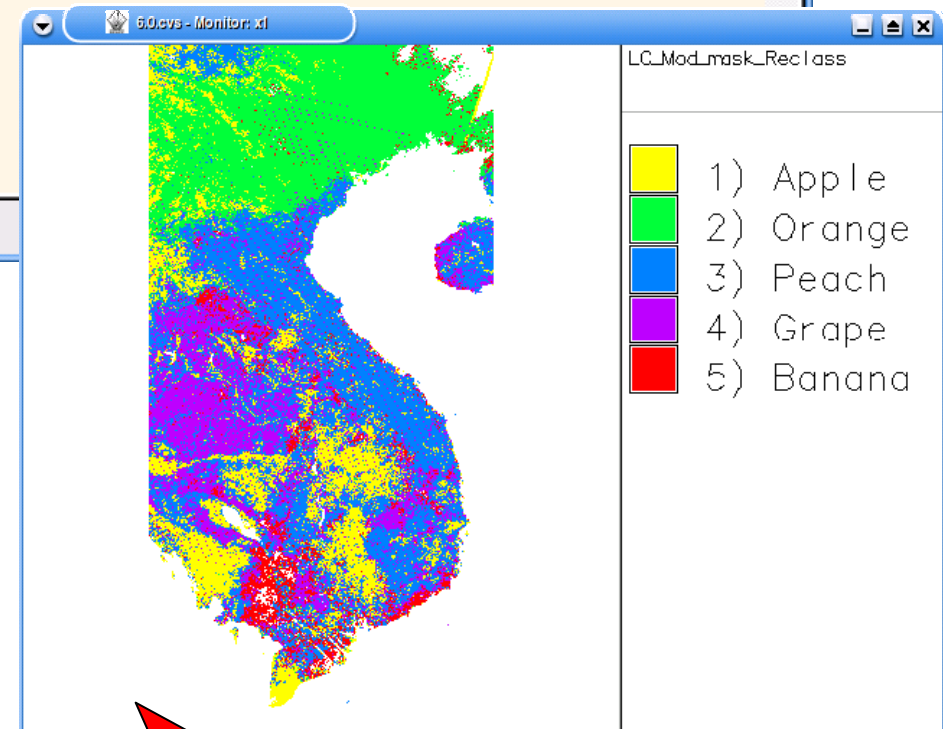
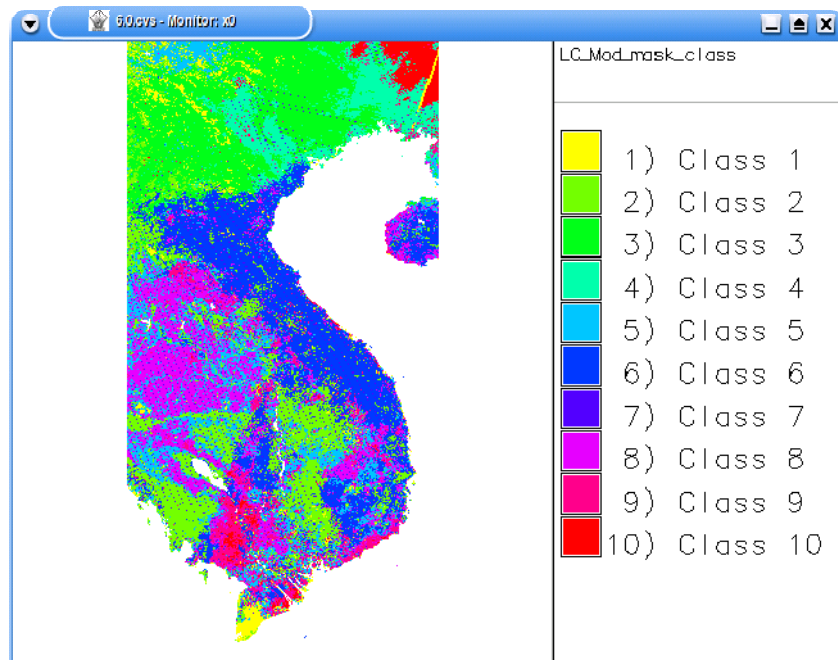


Shell - Konsole

Session Edit View Bookmarks Settings Help

```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > r.reclass input=LC_Mod_mask_class  
output=LC_Mod_mask_Reclass  
Enter rule(s), "end" when done, "help" if you need it  
Data range is 1 to 10  
> 1 2 = 1 Apple  
> 3 4 = 2 Orange  
> 5 6 = 3 Peach  
> 7 8 = 4 Grape  
> 9 20 = 5 Banana  
> end  
[Raster MASK present]
```

Reclassification



Section 2. Data Processing in Latitude-Longitude Coordinate System

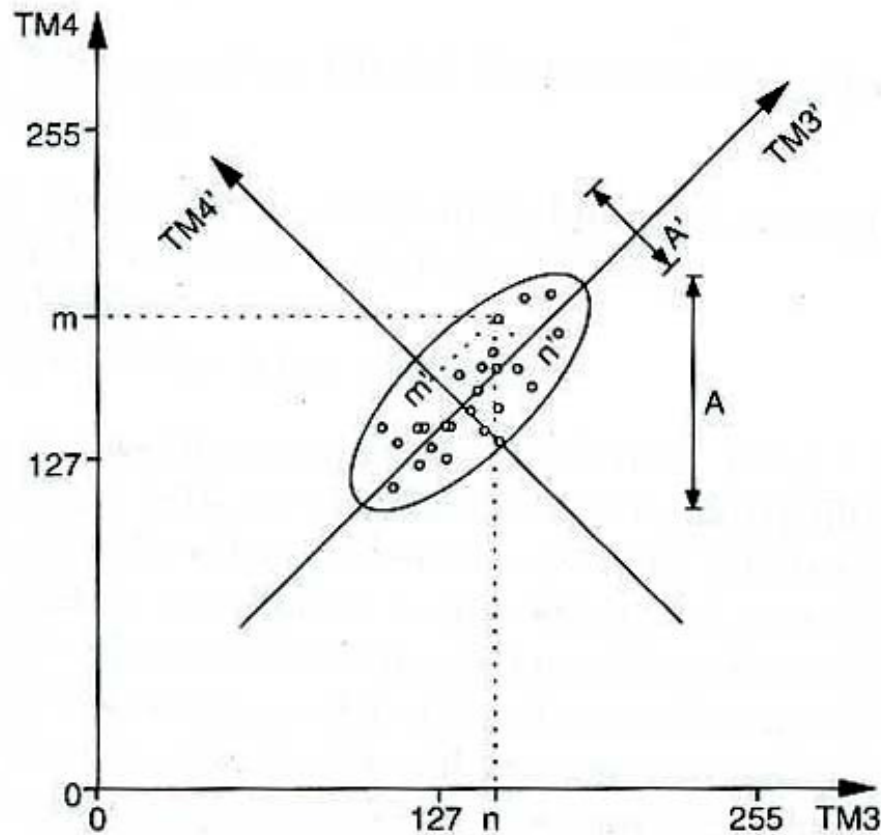
13. Principal Component Analysis

- Multispectral image channels often contain correlations due to similarities of the spectral response of the observed objects or slightly overlapping filter functions of the spectral sensor.
- This leads to redundancies within the data set.
- The “Principal Component Analysis (PCA)” has been developed to transform such a data set to a new data set without correlations between the channels.
- This will concentrate the image information in fewer image channels (reduction of image dimensionality), which is of particular interest for hyperspectral data.

- The PCA transforms the original multispectral data set to a new spectral coordinate system, the Principal Component axes, which are orthogonal to each other.
- In general, the first principal component (PC) image contains the maximum possible variance of the original images.
- The second principal component image contains the maximum possible variance not stored in the first PC image, as the second PC axis is orthogonal to the first PC axis.

- The number of PC images is identical to the number of input channels. The higher PC images explain remaining variances. Since the amount of variance decrease from the first to the last PC, uncorrelated noise (and sometimes some remaining igh frequencies) is found in the PC image.

- PCA is sometimes used for image compression, as it allows the image information to be concentrated in fewer channels.
- PCA is also sometimes used to generate additional channels to obtain more variables for classification process.



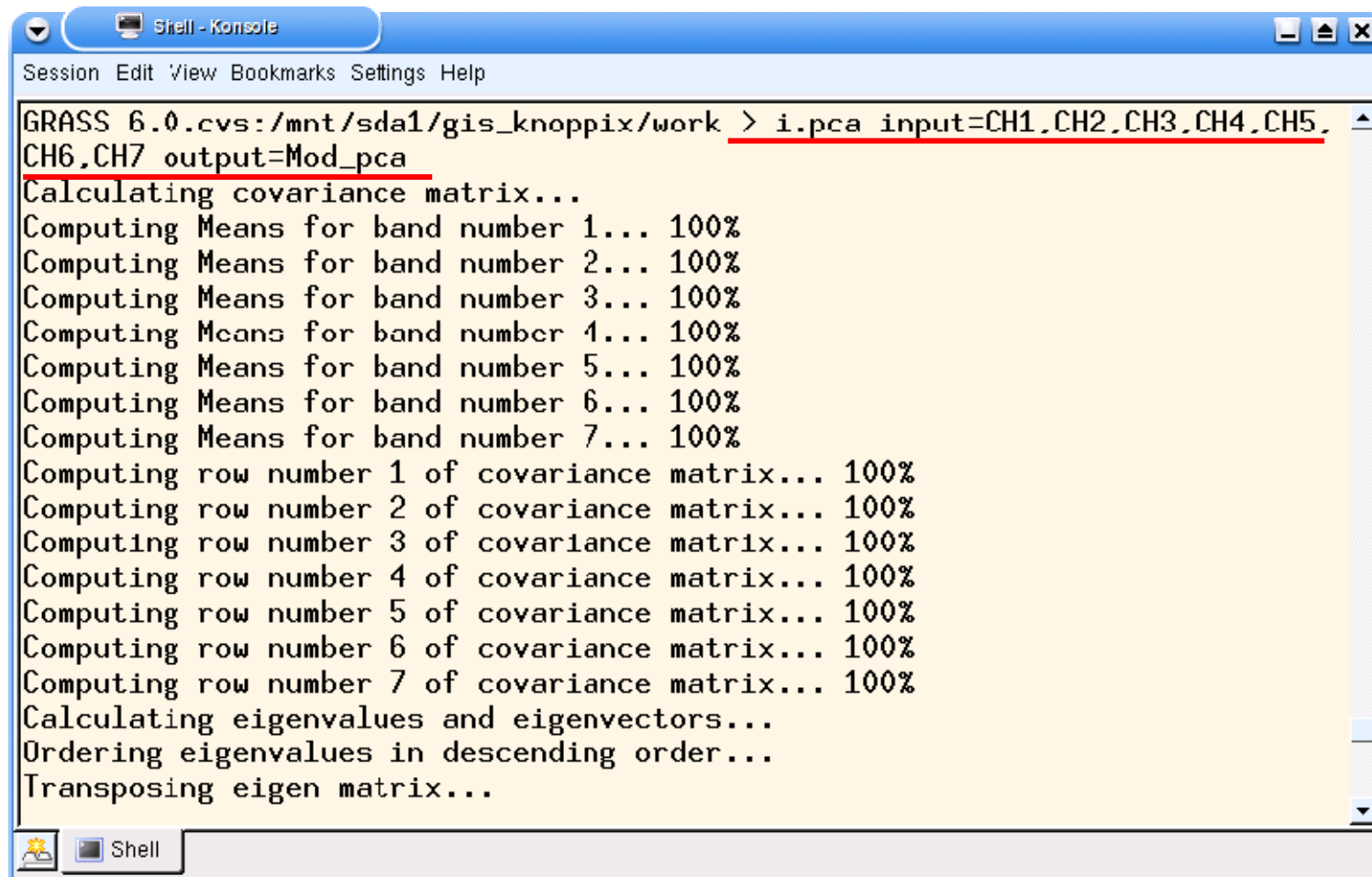
PCA applied to channel TM3 and TM4 of a LANDSAT-TM5 data set. Both the original spectral axes (channel TM3, TM4) and PC axes (PCA transformed channels TM3', TM4') are shown.

> i.pca input = A, B, C, ... output=a

A, B, C, ... = (input file name)

a = a prefix for the transformed PC image files,
which will be enumerated incrementally

: space

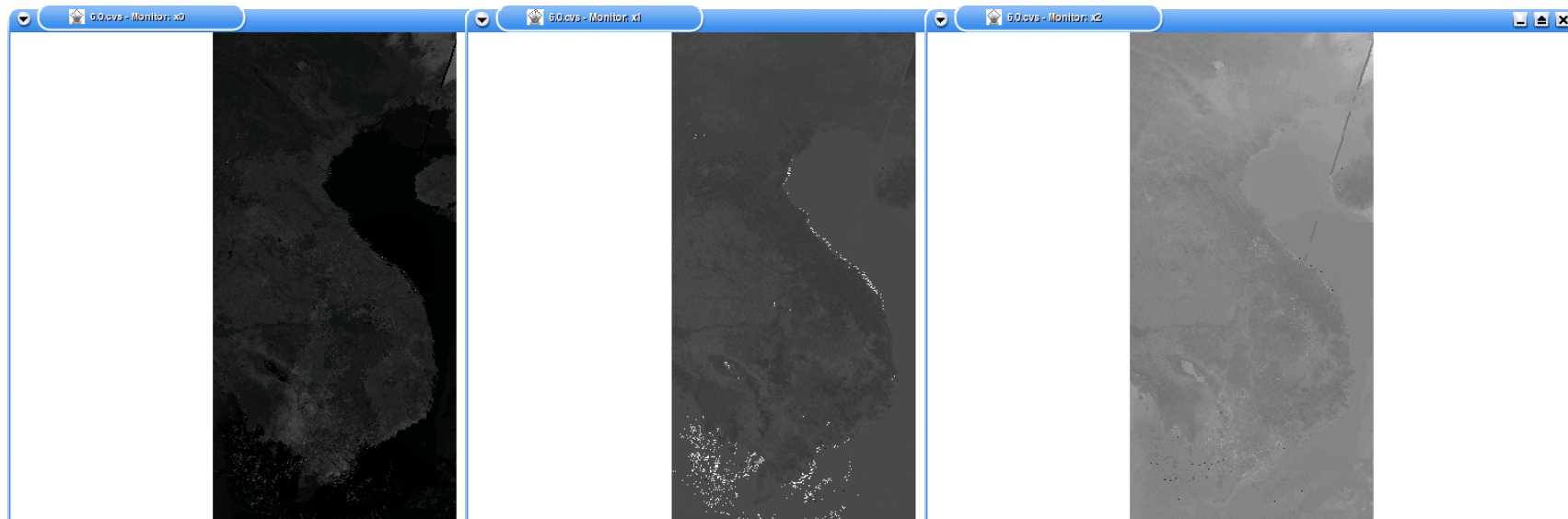


```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > i.pca input=CH1,CH2,CH3,CH4,CH5,CH6,CH7 output=Mod_pca
Calculating covariance matrix...
Computing Means for band number 1... 100%
Computing Means for band number 2... 100%
Computing Means for band number 3... 100%
Computing Means for band number 4... 100%
Computing Means for band number 5... 100%
Computing Means for band number 6... 100%
Computing Means for band number 7... 100%
Computing row number 1 of covariance matrix... 100%
Computing row number 2 of covariance matrix... 100%
Computing row number 3 of covariance matrix... 100%
Computing row number 4 of covariance matrix... 100%
Computing row number 5 of covariance matrix... 100%
Computing row number 6 of covariance matrix... 100%
Computing row number 7 of covariance matrix... 100%
Calculating eigenvalues and eigenvectors...
Ordering eigenvalues in descending order...
Transposing eigen matrix...
```



```
Shell - Konsole
Session Edit View Bookmarks Settings Help

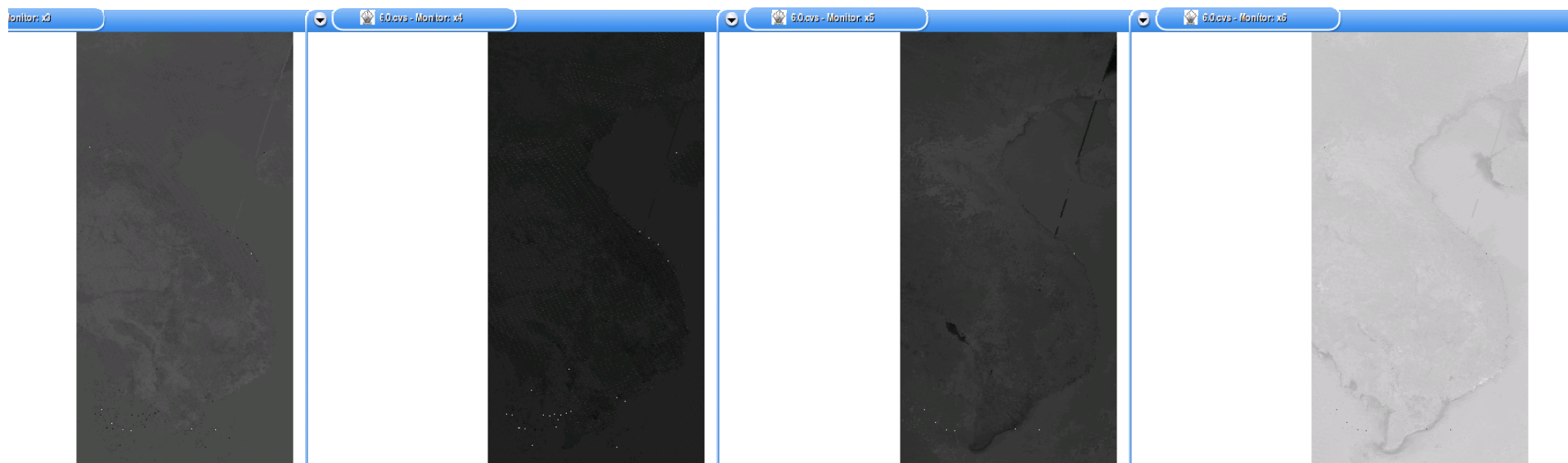
Transposing eigen matrix...
Transforming data...
Mod_pca.1: 100%
Mod_pca.1: Rescaling the data to 0.255 range ... 100%
Color table for [Mod_pca.1] set to grey
Mod_pca.2: 100%
Mod_pca.2: Rescaling the data to 0.255 range ... 100%
Color table for [Mod_pca.2] set to grey
Mod_pca.3: 100%
Mod_pca.3: Rescaling the data to 0.255 range ... 100%
Color table for [Mod_pca.3] set to grey
Mod_pca.4: 100%
Mod_pca.4: Rescaling the data to 0.255 range ... 100%
Color table for [Mod_pca.4] set to grey
Mod_pca.5: 100%
Mod_pca.5: Rescaling the data to 0.255 range ... 100%
Color table for [Mod_pca.5] set to grey
Mod_pca.6: 100%
Mod_pca.6: Rescaling the data to 0.255 range ... 100%
Color table for [Mod_pca.6] set to grey
Mod_pca.7: 100%
Mod_pca.7: Rescaling the data to 0.255 range ... 100%
Color table for [Mod_pca.7] set to grey
Eigen vectors:
( 0.15 0.46 0.06 0.12 0.60 0.48 0.40 )
( -0.03 -0.25 -0.02 -0.04 -0.28 -0.14 0.91 )
( 0.56 -0.07 0.54 0.52 -0.32 0.12 -0.04 )
( -0.01 -0.38 -0.22 -0.18 -0.28 0.83 -0.07 )
( 0.13 -0.76 0.12 0.05 0.62 -0.10 -0.03 )
( -0.68 -0.03 0.71 -0.04 -0.00 0.16 0.00 )
( 0.43 0.08 0.37 -0.82 -0.01 -0.04 -0.00 )
```



PC1

PC2

PC3



PC4

PC5

PC6

PC7

Section 2. Data Processing in Latitude-Longitude Coordinate System

14. 3-D display

14.1 Start of NVIZ

```
> nviz  elevation = A  color = B
```

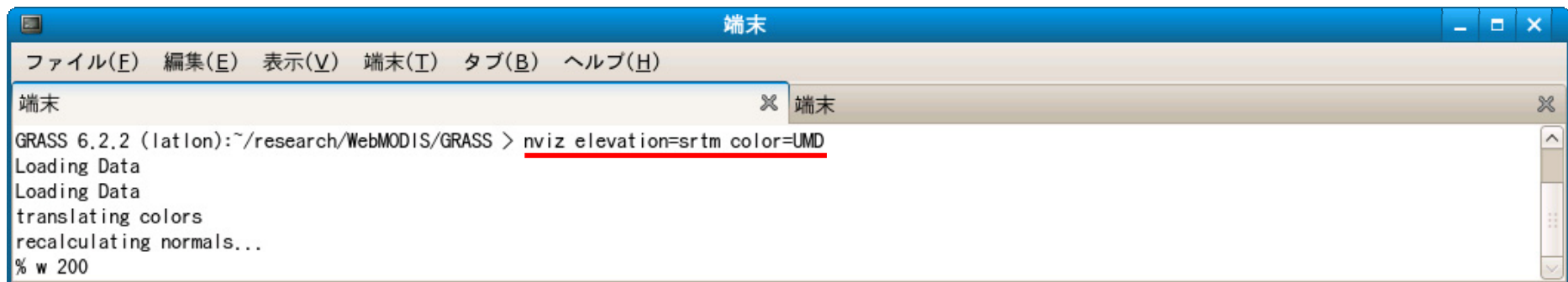
A = (input file name for elevation)

B = (input file name for texture)

: space

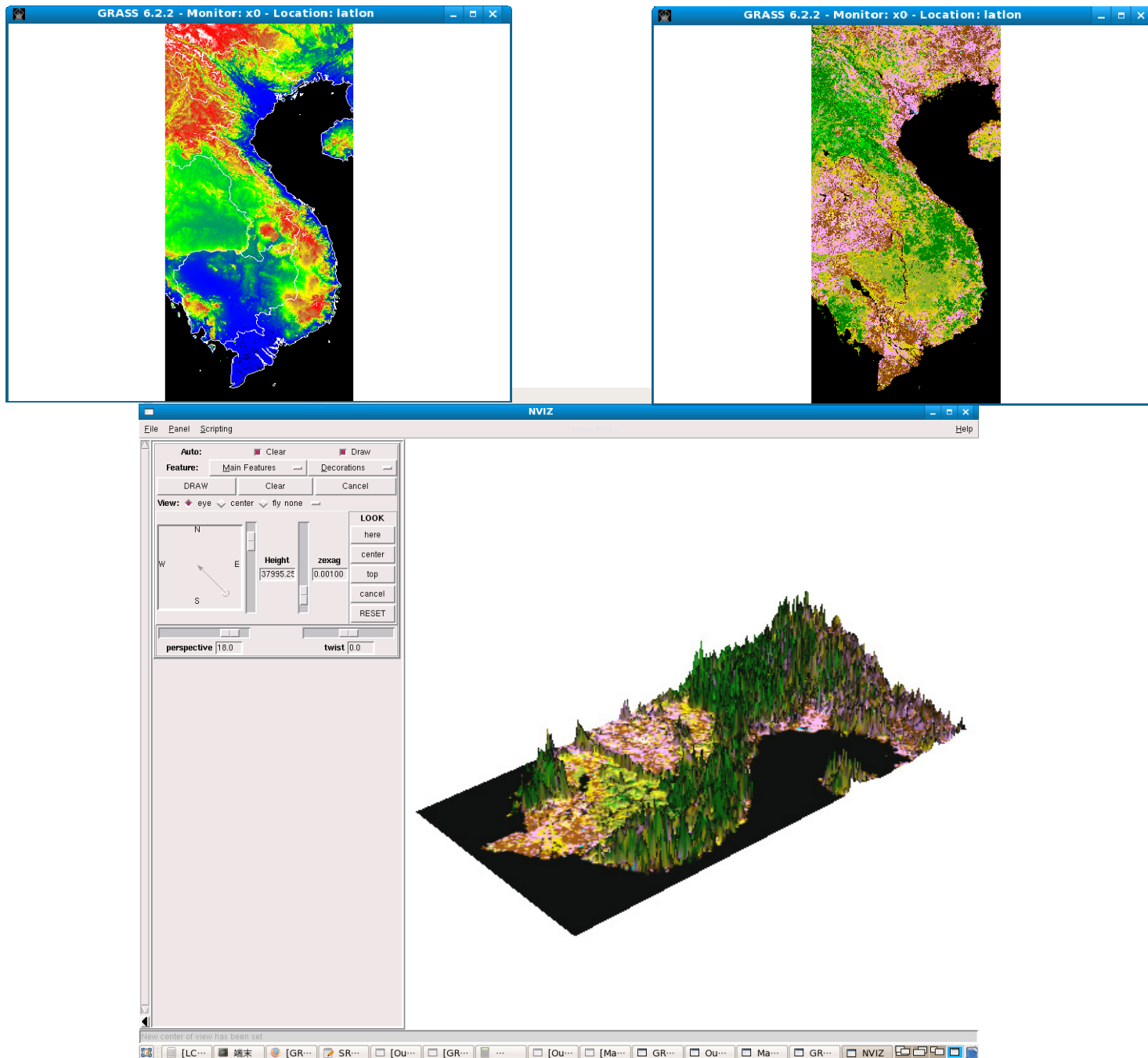
Settings such as viewing vector can be changed in GUI.

Finally, you'll have a following bird-eye view map. The height comes from “A”, and the surface color comes from “B” value.



The screenshot shows a terminal window titled "端末" (Terminal) with a menu bar containing "ファイル(F)", "編集(E)", "表示(V)", "端末(T)", "タブ(B)", and "ヘルプ(H)". The terminal content shows the command `nviz elevation=srtm color=UMD` being executed in a GRASS 6.2.2 environment. The output includes "Loading Data", "Loading Data", "translating colors", and "recalculating normals...". The prompt is `% w 200`. The command line is highlighted with a red underline.

```
端末
ファイル(F) 編集(E) 表示(V) 端末(T) タブ(B) ヘルプ(H)
端末
GRASS 6.2.2 (latlon):~/research/WebMODIS/GRASS > nviz elevation=srtm color=UMD
Loading Data
Loading Data
translating colors
recalculating normals...
% w 200
```



Section 2. Data Processing in Latitude-Longitude Coordinate System

15. Export raster

15.1 Export a raster to a binary file

15.2 Export a raster to a TIFF file

15.3 Export displayed image to PNG file

15.1 Export a raster to a binary file

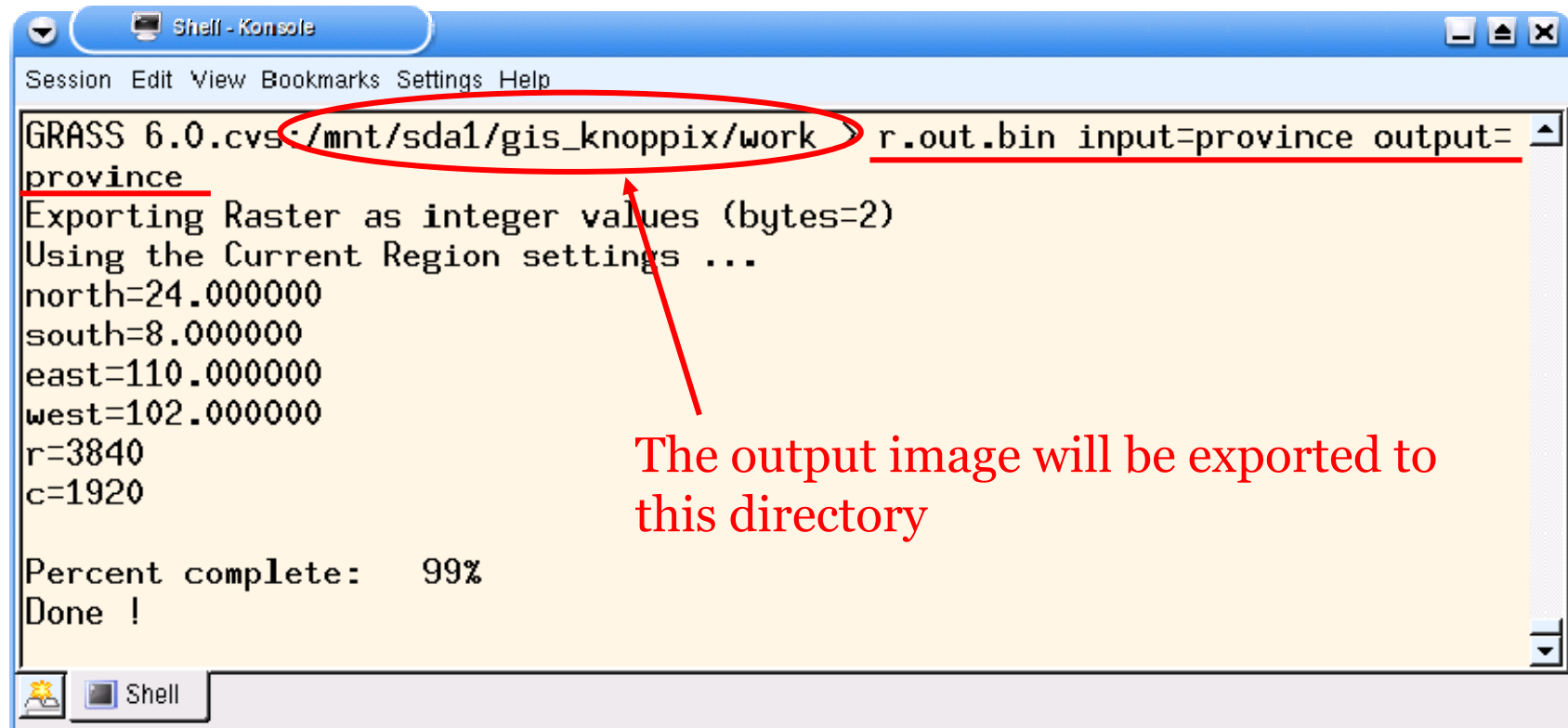
␣ : space

```
> r.out.bin ␣ input = " A " ␣ output = " B "
```

A = (input file name)

B = (output file name)

The *r.out.bin* program exports a GRASS raster map to a binary array file.



```
GRASS 6.0.cvs /mnt/sda1/gis_knoppix/work > r.out.bin input=province output=
province
Exporting Raster as integer values (bytes=2)
Using the Current Region settings ...
north=24.000000
south=8.000000
east=110.000000
west=102.000000
r=3840
c=1920

Percent complete: 99%
Done !
```

The output image will be exported to this directory

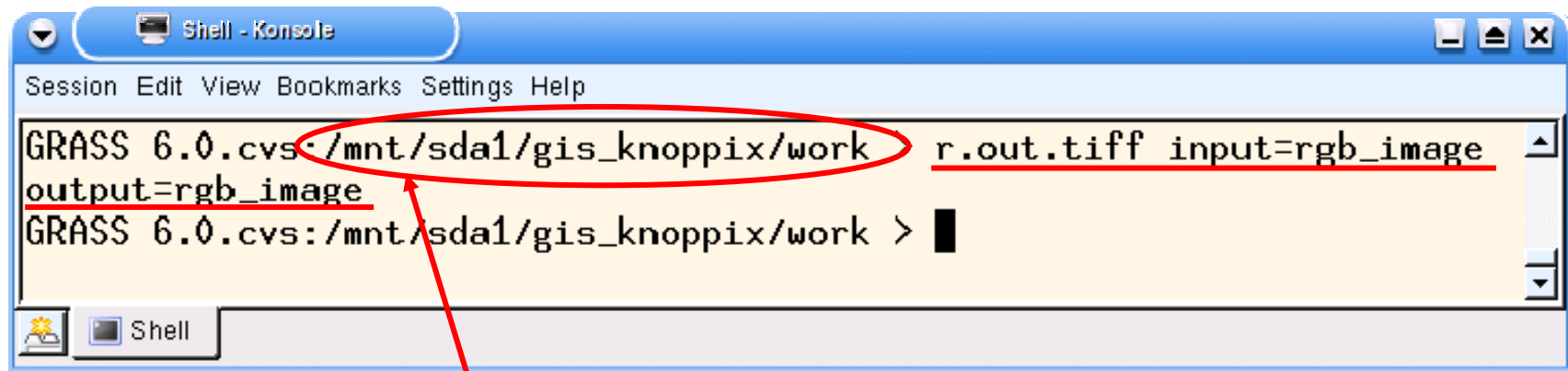
15.2 Export a raster to a TIFF file

□ : space

```
> r.out.tiff □ input = A □ output = B
```

A = (input file name)

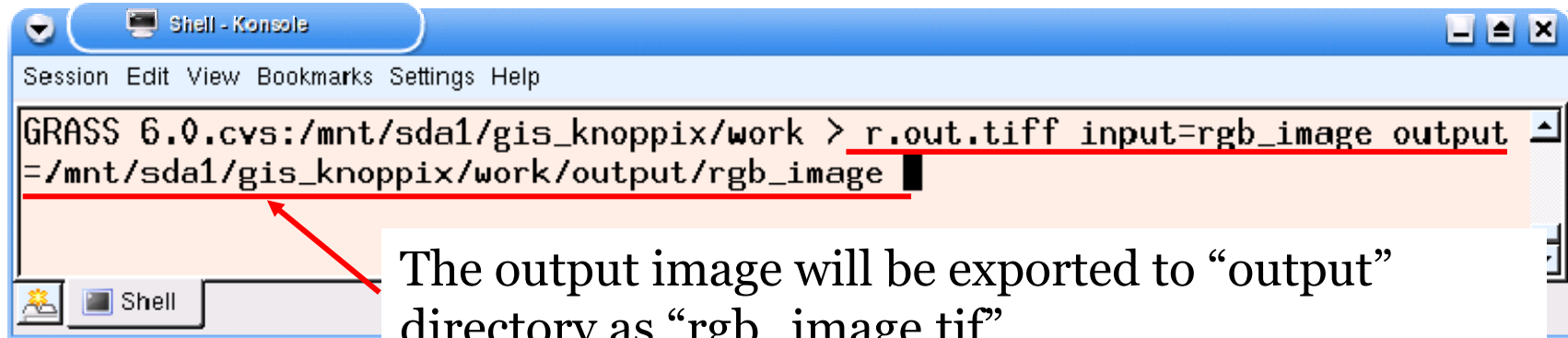
B = (output file name)



```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work r.out.tiff input=rgb_image  
output=rgb_image  
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work >
```

The output image will be exported to this directory

If you want to export images to another directory,



```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > r.out.tiff input=rgb_image output  
=/mnt/sda1/gis_knoppix/work/output/rgb_image
```

The output image will be exported to “output” directory as “rgb_image.tif”.

15.3 Export displayed image to PNG file

```
> d.mon ␣ start = PNG
```

```
> d.rast ␣ map = A
```

A = (input file name1)

```
> d.vect ␣ map = B
```

B = (input file name2)

```
> d.mon ␣ stop = PNG
```

␣ : space

This will write a file named *map.png* to be created in your current directory

The screenshot shows a terminal window titled "Shell - Konsole" with a menu bar (Session, Edit, View, Bookmarks, Settings, Help). The terminal displays the following commands and output:

```
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.mon start=PNG
PNG: GRASS_TRUECOLOR status: FALSE
PNG: collecting to file: map.png,
      GRASS WIDTH=640. GRASS HEIGHT=480
Graphics driver [PNG] started
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.rast map=rgb image
100%
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.vect map=border type=boundary
color=yellow
/usr/local/grass6.0.cvs-i686-pc-linux-gnu-22_01_2005/drivers
hile loading shared libraries: libtk.so.0: cannot open shared object file:
No such file or directory
dbmi: Protocol error (invalid table/column name or unsupported column type)
WARNING: Cannot open driver 'ogr'
WARNING: Cannot open OGR DBMI driver.
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.vect map=city icon=basic/point.
size=20 color=red fcolor=red
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.vect map=city display=attr att.
rcol=label lsize=20 lcolor=red
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > d.mon stop=PNG
Monitor 'PNG' terminated
GRASS 6.0.cvs:/mnt/sda1/gis_knoppix/work > █
```

Four red arrows point to specific commands, each with a circled number and a label:

- ① display RGB image (points to d.mon start=PNG)
- ② display border (points to d.vect map=border type=boundary color=yellow)
- ③ display symbol (points to d.vect map=city icon=basic/point.size=20 color=red fcolor=red)
- ④ display label (points to d.vect map=city display=attr attrcol=label lsize=20 lcolor=red)

The terminal window has a taskbar at the bottom with a "Shell" icon and a "Shell" label.

① display RGB image



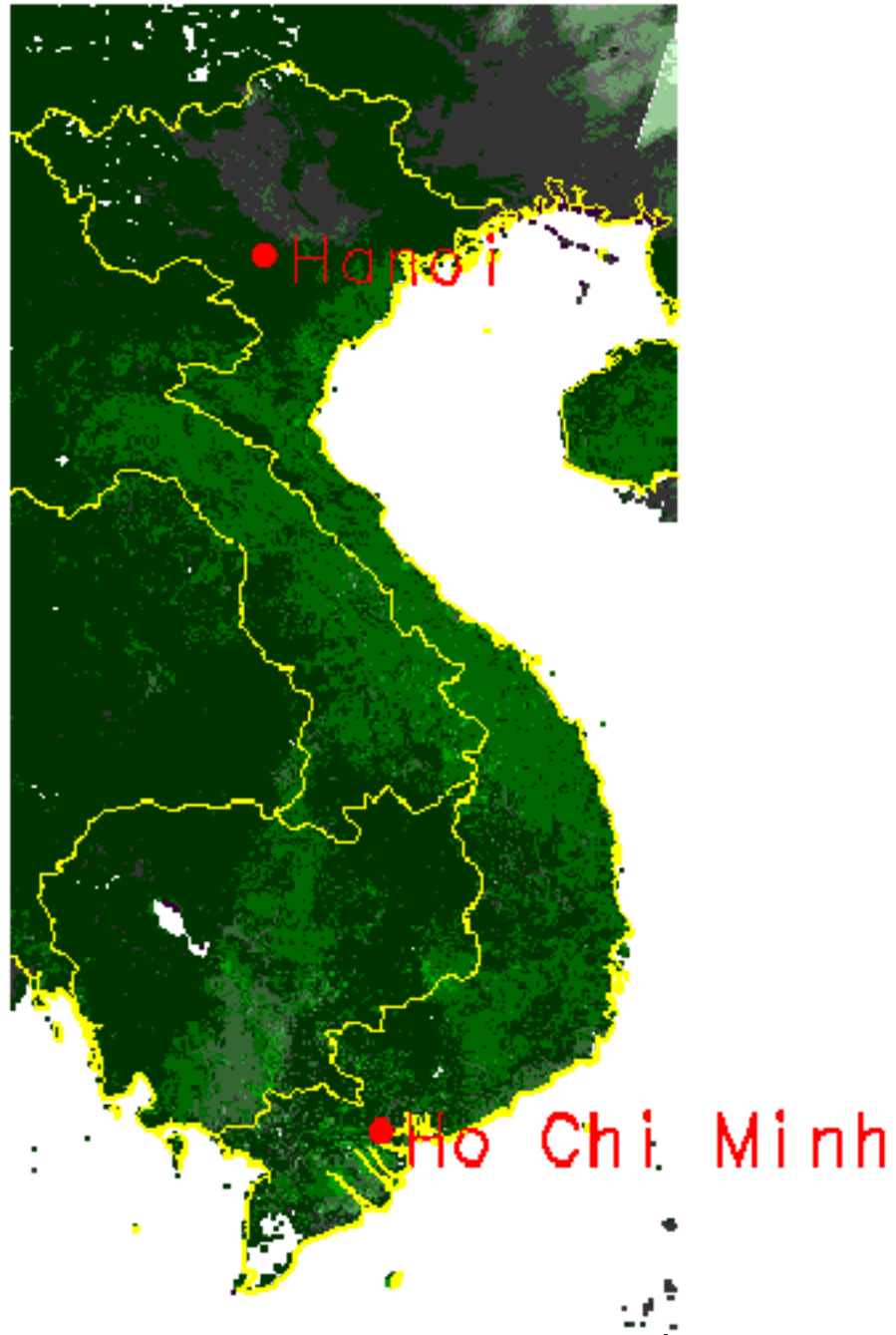
② display border



③ display symbol



④ display label



Section 2. Data Processing in Latitude-Longitude Coordinate System

Reference

1. GRASS GIS 6.2.4cvs Reference Manual
http://grass.itc.it/grass62/manuals/html62_user/index.html
2. Introduction to GIS, Dr. Kenlo NASAHARA (NISHIDA, maiden name),
Tsukuba University, Japan
<http://ryuiki.agbi.tsukuba.ac.jp/~nishida/lecture/o7-GIS/>
3. Remote Sensing Image Processing using GRASS , Dr. Junichi SUSAKI,
Kyoto University, Japan
<http://www.envinfo.uee.kyoto-u.ac.jp/user/susaki/>
4. Markus Neteler and Helena Mitasova, Open source GIS
– A GRASS GIS approach – Third edition, Springer 2008
5. GDF Hannover: GRASS 6 Course material
<http://www.gdf-hannover.de/media.php?id=o&lg=en>

Appendix A

GRASS command list

r.in.bin : Import a binary raster file into a GRASS raster map layer.

SYNOPSIS

r.in.bin [-sfdbh] **input**=*string* **output**=*name* [**title**=*"phrase"*] [**bytes**=*integer*]
[**north**=*float*] [**south**=*float*] [**east**=*float*] [**west**=*float*] [**rows**=*float*] [**cols**=*float*]
[**anull**=*float*] [--**overwrite**]

Flags:

- s** Signed data (high bit means negative value)
- f** Import as Floating Point Data (default: Integer)
- d** Import as Double Precision Data (default: Integer)
- b** Byte Swap the Data During Import
- h** Get region info from GMT style header
- overwrite** Force overwrite of output files

Parameters:

input = <i>string</i>	Bin raster file to be imported
output = <i>name</i>	Name for output raster map
title = <i>"phrase"</i>	Title for resultant raster map
bytes = <i>integer</i>	Number of bytes per cell (1, 2, 4) Default: 1
north = <i>float</i>	Northern limit of geographic region (outer edge)
south = <i>float</i>	Southern limit of geographic region (outer edge)
east = <i>float</i>	Eastern limit of geographic region (outer edge)
west = <i>float</i>	Western limit of geographic region (outer edge)
rows = <i>float</i>	Number of rows
cols = <i>float</i>	Number of columns
anull = <i>float</i>	Set Value to NULL

r.in.gdal : Import GDAL supported raster file into a binary raster map layer.

SYNOPSIS

r.in.gdal [-oefk] **input=string** **output=name** [**band=integer**] [**target=string**]
[**title="phrase"**] [**location=string**] [--**overwrite**]

Flags:

- o** Override projection (use location's projection)
- e** Extend location extents based on new dataset
- f** List supported formats then exit
- k** Keep band numbers instead of using band color names
- overwrite** Force overwrite of output files

Parameters:

input=string	Raster file to be imported
output=name	Name for output raster map
band=integer	Band to select (default is all bands)
target=string	Name of location to read projection from for GCPs transformation
title="phrase"	Title for resultant raster map
location=string	Name for new location to create

g.list : Lists available GRASS data base files of the user-specified data type to standard output.

SYNOPSIS

g.list [-f] **type**=*datatype* [**mapset**=*string*]

Flags: **-f** verbose listing (also list map titles)

Parameters: **type**=*datatype* data type
 Options: *rast,rast3d,vect,oldvect,asciivect,icon*
 labels,sites,region,region3d,group,3dvie
 mapset=*string* mapset to list (default: current search path)

d.mon : To establish and control use of a graphics display monitor.

SYNOPSIS

d.mon [-**ILprs**] [**start=string**] [**stop=string**] [**select=string**] [**unlock=string**]

Flags:

- l** List all monitors
- L** List all monitors (with current status)
- p** Print name of currently selected monitor
- r** Release currently selected monitor
- s** Do not automatically select when starting

Parameters:

start=string	Name of graphics monitor to start
stop=string	Name of graphics monitor to stop
select=string	Name of graphics monitor to select
unlock=string	Name of graphics monitor to unlock

d.rast : Displays and overlays raster map layers in the active display frame on the graphics monitor

SYNOPSIS

d.rast [-oix] **map=string** [**catlist=cat**[-cat][,cat[-cat],...]] [**vallist=val**[-val][,val[-val],...]] [**bg=color**]

Flags:

- o** Overlay (non-null values only)
- i** Invert catlist
- x** Don't add to list of rasters and commands in monitor

Parameters:

- map=string** Raster map to be displayed
- catlist=cat**[-cat][,cat[-cat],...] List of categories to be displayed (INT maps)
- vallist=val**[-val][,val[-val],...] List of values to be displayed (FP maps)
- bg=color** Background color (for null)
Options: *white,black,red,green,blue,yellow,magenta,cyan,aqua,grey,gray,orange,brown,purple,violet,indigo*

r.colors : Creates/Modifies the color table associated with a raster map layer .

SYNOPSIS

r.colors [-wql] map=*name* [color=*type*] [rast=*string*] [rules=*string*]

Flags:	-w	Keep existing color table
	-q	Quietly
	-l	List rules

Parameters:	map= <i>name</i>	Name of input raster map
	color= <i>type</i>	Type of color table
		Options: <i>aspect, grey, grey.eq, grey.log, byg, byr, gyr, rainbow, ramp, random, ryg, wave, rules</i>
		aspect : aspect oriented grey colors
		grey : linear grey scale
		grey.eq : histogram equalized grey scale
		grey.log : histogram logarithmic transformed grey scale
		byg : blue through yellow to green colors
		byr : blue through yellow to red colors
		gyr : green through yellow to red colors
		rainbow : rainbow color table
		ramp : color ramp
		random : random color table
		ryg : red through yellow to green colors
		wave : color wave
		rules : create new color table by rules
	rast= <i>string</i>	Raster map name from which to copy color table
	rules= <i>string</i>	Name of predefined rules file
		Options: <i>aspect, terrain, gyr, slope, elevation, grey, ramp, bcyr, evi, srtm, ryg, rainbow, wave, population, byr, ndvi, etopo2, byg</i>

r.report : Reports statistics for raster map layers.

SYNOPSIS

r.report [-hfqenNCi] **map**=string[,string,...] [**units**=string[,string,...]]
[**null**=string] [**pl**=integer] [**pw**=integer] [**output**=string] [**nsteps**=integer]

Flags:

- h** Suppress page headers
- f** Use formfeeds between pages
- q** Quiet
- e** Scientific format
- n** Filter out all no data cells
- N** Filter out cells where all maps have no data
- C** Report for cats fp ranges (fp maps only)
- i** Read fp map as integer (use map's quant rules)

Parameters:

map =string[,string,...]	Raster map(s) to report on
units =string[,string,...]	mi(les),me(ters),k(ilometers),a(cres),h(ectares), c(ell_counts),p(ercent_cover)
null =string	Character representing no data cell value Default: *
pl =integer	Page length (default: 0 lines)
pw =integer	Page width (default: 79 characters)
output =string	Name of an output file to hold the report
nsteps =integer	Number of fp subranges to collect stats from Default: 255

d.zoom : Allows the user to change the current geographic region settings interactively, with a mouse..

SYNOPSIS

d.zoom [-fphjr] **rast**=*string*[,*string*,...] [**vector**=*string*[,*string*,...]] [**zoom**=*float*]

Flags:

- f Full menu (zoom + pan) & Quit menu
- p Pan mode
- h Handheld mode
- j Just redraw given maps using default colors
- r Return to previous zoom

Parameters:

rast=*string*[,*string*,...] Name of raster map
vector=*string*[,*string*,...] Name of vector map
zoom=*float* magnification: >1.0 zooms in, <1.0 zooms out
Options: 0.001-1000.0
Default: 0.75

g.region : Program to manage the boundary definitions for the geographic region.

SYNOPSIS

g.region [-dplecmau3gb] [region=name] [rast=name[,name,...]] [rast3d=name]
[vect=name] [3dview=name] [n=value] [s=value] [e=value] [w=value] [t=value]
[b=value] [res=value] [res3=value] [nsres=value] [ewres=value] [tbres=value]
[zoom=name] [align=name] [save=name] [--overwrite]

Flags:

- d** Set from default region
- p** Print the current region
- l** Print the current region in lat/long on current ellipsoid/datum
- e** Print the current region extent
- c** Print the current region map center coordinates
- m** Print region resolution in meters (geodesic)
- a** Align region to resolution (default = align to bounds, works only for 2D resolution)
- u** Do not update the current region
- 3** Print also 3D settings
- g** Print the current region (shell script style)
- b** Print the maximum bounding box in lat/long on WGS84 (-g mode only)
- overwrite** Force overwrite of output files

Parameters:

region = <i>name</i>	Set current region from named region
rast = <i>name[,name,...]</i>	Set region to match this raster map
rast3d = <i>name</i>	Set region to match this 3D raster map (both 2D and 3D values)
vect = <i>name</i>	Set region to match this vector map
3dview = <i>name</i>	Set region to match this 3dview file
n = <i>value</i>	Value for the northern edge (format dd:mm:ss{N S})
s = <i>value</i>	Value for the southern edge (format dd:mm:ss{N S})
e = <i>value</i>	Value for the eastern edge (format ddd:mm:ss{E W})
w = <i>value</i>	Value for the western edge (format ddd:mm:ss{E W})
t = <i>value</i>	Value for the top edge
b = <i>value</i>	Value for the bottom edge
res = <i>value</i>	Grid resolution 2D (both north-south and east-west)
res3 = <i>value</i>	3D grid resolution (north-south, east-west and top-bottom)
nsres = <i>value</i>	North-south grid resolution 2D (format dd:mm:ss)
ewres = <i>value</i>	East-west grid resolution 2D (format dd:mm:ss)
tbres = <i>value</i>	Top-bottom grid resolution 3D
zoom = <i>name</i>	Raster map to zoom into
align = <i>name</i>	Raster map to align to
save = <i>name</i>	Save the current region to region definition file

d.legend : Displays a legend for a raster map in the active frame of the graphics monitor.

SYNOPSIS

d.legend [-vcnsmf] **map**=*string* [**color**=*string*] [**lines**=*integer*] [**thin**=*integer*]
[**labelnum**=*integer*] [**at**=*bottom,top,left,right*] [**use**=*catnum[,catnum,...]*] [**range**=*min,max*]

Flags:

- v** Do not show category labels
- c** Do not show category numbers
- n** Skip categories with no label
- s** Draw smooth gradient
- m** Use mouse to size & place legend
- f** Flip legend

Parameters:

map = <i>string</i>	Name of raster map
color = <i>string</i>	Sets the legend's text color Options: <i>red,orange,yellow,green,blue,indigo,violet,white,black,gray,brown,magenta,aqua,grey,cyan,purple</i> Default: <i>black</i>
lines = <i>integer</i>	Number of text lines (useful for truncating long legends) Options: <i>0-1000</i> Default: <i>0</i>
thin = <i>integer</i>	Thinning factor (thin=10 gives cats 0,10,20...) Options: <i>1-1000</i> Default: <i>1</i>

labelnum=*integer* Number of text labels for smooth gradient legend
Options: 2-100 Default: 5

at=*bottom,top,left,right* Screen coordinates to place the legend
(as percentage) Options: 0-100

use=*catnum[,catnum,...]* List of discrete category numbers/values
for legend

range=*min,max* Use a subset of the map range for the legend (min,max)

d.rast.leg : Displays a raster map and its legend on a graphics window

SYNOPSIS

d.rast.leg [-n] **map**=*string* [**num_of_lines**=*integer*]

Flags:

-n omit entries with missing label

Parameters: **map**=*string* raster input map

num_of_lines=*integer* Number of lines to appear in the legend

d.histogram : Displays a histogram in the form of a pie or bar chart for a user-specified raster file.

SYNOPSIS

d.histogram [-nqC] **map**=*string* [**color**=*string*] [**style**=*string*] [**nsteps**=*integer*]

Flags:

- n** Display information for null cells
- q** Gather the histogram quietly
- C** Report for ranges defined in cats file (fp maps only)

Parameters:

map = <i>string</i>	Raster map for which histogram will be displayed
color = <i>string</i>	Color for legend and title Options: <i>red,orange,yellow,green,blue,indigo,violet,white,black,gray,brown,magenta,aqua,grey,cyan,purple</i> Default: <i>black</i>
style = <i>string</i>	Indicate if a pie or bar chart is desired Default: <i>bar</i>
nsteps = <i>integer</i>	Number of steps to divide the data range into (fp maps only) Default: 255

d.grid : Overlays a user-specified grid in the active display frame on the graphics monitor

SYNOPSIS

d.grid [-gwnbt] **size**=*value* [**color**=*string*] [**origin**=*easting,northing*]
[**bordercolor**=*string*]

Flags:

- g** Draw geographic grid (referenced to current ellipsoid)
- w** Draw geographic grid (referenced to WGS84 ellipsoid)
- n** Disable grid drawing
- b** Disable border drawing
- t** Disable text drawing

Parameters:

size = <i>value</i>	Size of grid to be drawn
color = <i>string</i>	Sets the grid color, either a standard GRASS color or R:G:B triplet (separated by colons) Default: <i>gray</i>
origin = <i>easting,northing</i>	Lines of the grid pass through this coordinate Default: <i>0,0</i>
bordercolor = <i>string</i>	Sets the border color, either a standard GRASS color or R:G:B triplet Default: <i>brown</i>

d.erase : Erase the contents of the active display frame
with user defined color .

SYNOPSIS

d.erase [-fx] [**color**=*string*]

Flags: **-f** Remove all frames and erase the screen
 -x Don't add to list of commands in monitor

Parameters: **color**=*string* Color to erase with, either a standard GRASS color
 or R:G:B triplet (separated by colons)
 Default: *white*

v.external : Create a new vector as a read-only link to OGR layer.

Available drivers: GRASS,ESRI Shapefile,MapInfo File,UK .NTF,SDTS,
TIGER,S57,DGN,VRT,AVCBin,REC,Memory,CSV,GML,
KML,GMT,SQLite,ODBC,PGeo,PostgreSQL,MySQL

SYNOPSIS

v.external *dsn=string* [**output=***name*] [**layer=***string*] [--**overwrite**]

Flags: **--overwrite** Force overwrite of output files

Parameters: **dsn=string** OGR datasource name.

Examples:

ESRI Shapefile: directory containing shapefiles

MapInfo File: directory containing mapinfo files

output=*name* Output vector, if not given, available layers are
printed only

layer=*string* OGR layer name. If not given, available layers are
printed only.

Examples:

ESRI Shapefile: shapefile name

MapInfo File: mapinfo file name

v.in.ascii : Convert GRASS ascii file or points file to binary vector .

SYNOPSIS

v.in.ascii [-ztenb] [input=*string*] output=*name* [format=*string*] [fs=*string*]
[skip=*integer*] [columns=*string*] [x=*integer*] [y=*integer*] [z=*integer*]
[cat=*integer*] [--overwrite]

Flags:

- z** Create 3D file
- t** Do not create table in points mode
- e** Create a new empty map and exit. Nothing is read from input.
- n** Don't expect a header when reading in standard format
- b** Do not build topology in points mode
- overwrite** Force overwrite of output files

Parameters:

input=<i>string</i>	ASCII file to be converted to binary vector file, if not given reads from standard input
output=<i>name</i>	Name for output vector map
format=<i>string</i>	Input file format Options: <i>point, standard</i> Default: <i>point</i>
fs=<i>string</i>	Field separator Default:
skip=<i>integer</i>	Number of header lines to skip at top of input file (written to map history) Default: 0

columns=string	Columns definition for points mode in SQL style, for example: 'x double precision, y double precision, cat int, name varchar(10)'
x=integer	Number of column used as x coordinate (first column is 1) for points mode Default: 1
y=integer	Number of column used as y coordinate (first column is 1) for points mode Default: 2
z=integer	Number of column used as z coordinate (first column is 1) for points mode. If 0, z coordinate is not used. Default: 0
cat=integer	Number of column used as category (first column is 1) for points mode. If 0, unique category is assigned to each row and written to new column 'cat'. Default: 0

Parameters:	map = <i>name</i>	Name of input raster map
	color = <i>type</i>	Type of color table Options: <i>aspect, grey, grey.eq, grey.log, byg, byr, gyr, rainbow, ramp, random, ryg, wave, rules</i>
		aspect : aspect oriented grey colors
		grey : linear grey scale
		grey.eq : histogram equalized grey scale
		grey.log : histogram logarithmic transformed grey scale
		byg : blue through yellow to green colors
		byr : blue through yellow to red colors
		gyr : green through yellow to red colors
		rainbow : rainbow color table
		ramp : color ramp
		random : random color table
		ryg : red through yellow to green colors
		wave : color wave
		rules : create new color table by rules
	rast = <i>string</i>	Raster map name from which to copy color table
	rules = <i>string</i>	Name of predefined rules file Options: <i>aspect, terrain, gyr, slope, elevation, grey, ramp, bcyr, evi, srtm, ryg, rainbow, wave, population, byr, ndvi, etopo2, byg</i>

d.vect : Displays GRASS vector data in the active frame on the graphics monitor.

SYNOPSIS

```
d.vect [-vacix] map=name [type=string[,string,...]] [display=string[,string,...]]  
[attrcol=string] [icon=string] [size=integer] [layer=integer] [cats=range]  
[where=sql_query] [width=integer] [wcolumn=string] [wscale=float] [color=string]  
[fcolor=string] [rgb_column=string] [llayer=integer] [lcolor=string] [bgcolor=string]  
[bcolor=string] [lsize=integer] [font=string] [xref=string] [yref=string] [minreg=float]  
[maxreg=float]
```

Flags:	-v	Run verbosely
	-a	Get colors from map table column (of form RRR:GGG:BBB)
	-c	Random colors according to category number (or layer number if 'layer=-1' is given)
	-i	Use values from 'cats' option as line ID
	-x	Don't add to list of vectors and commands in monitor (it won't be drawn if the monitor is refreshed)

Parameters:

- map=name** Name of input vector map
- type=string[,string,...]** Type Options: *point,line,boundary,centroid,area,face*
Default: *point,line,boundary,centroid,area,face*
- display=string[,string,...]** Display Options: *shape,cat,topo,dir,attr,zcoor*
Default: *shape*
- attrcol=string** Name of column to be displayed
- icon=string** Point and centroid symbol Options:
*basic/marker,basic/x,basic/point,basic/pushpin,basic/cross2,
basic/star,basic/diamond,basic/octagon,basic/arrow2,basic/box,
basic/arrow1,basic/circle,basic/triangle,basic/cross1,
demo/smrk,demo/muchomurka,extra/compass,
extra/fancy_compass,extra/airport,extra/4pt_star,extra/adcp,
extra/dive_flag,extra/alpha_flag,extra/half-circle,extra/target*
Default: *basic/x*
- size=integer** Icon size Default: 8
- layer=integer** Layer number. If -1, all layers are displayed. Default: 1
- cats=range** Category values Example: 1,3,7-9,13
- where=sql_query** WHERE conditions of SQL statement without 'where'
keyword. (example: income < 1000 and inhab >= 10000)
- width=integer** Line width Default: 0
- wcolumn=string** Name of column for line widths (these values will
be scaled by wscale)
- wscale=float** Scale factor for wcolumn Default: 1

Parameters:	color=string	Line color	Default: <i>black</i>
	fcolor=string	Area fill color	Default: <i>200:200:200</i>
	rgb_column=string	Name of color definition column (for use with -a flag) Default: <i>GRASSRGB</i>	
	llayer=integer	Layer for labels	Default: <i>1</i>
	lcolor=string	Label color	Default: <i>red</i>
	bgcolor=string	Label background color	Default: <i>none</i>
	bcolor=string	Label border color	Default: <i>none</i>
	lsize=integer	Label size (pixels)	Default: <i>8</i>
	font=string	Font name	Options: <i>cyrilc,gothgbt,gothgrt,gothitt,greekc,greekcs,greekp,greeks,italicc,italiccs,italict,romanc,romancs,romand,romans,romant,scriptc,scripts</i> Default: <i>romans</i>
	xref=string	Label horizontal justification	Options: <i>left,center,right</i> Default: <i>left</i>
	yref=string	Label vertical justification	Options: <i>top,center,bottom</i> Default: <i>center</i>
	minreg=float	Minimum region size (average from height and width) when map is displayed	
	maxreg=float	Maximum region size (average from height and width) when map is displayed	

v.to.rast : Converts a binary GRASS vector map layer into a GRASS raster map layer.

SYNOPSIS

v.to.rast *input=name output=name [use=string] [column=name] [layer=integer] [value=float] [rows=integer] [--overwrite]*

Flags: **--overwrite** Force overwrite of output files

Parameters:	input=name	Name of input vector map	
	output=name	Name for output raster map	
	use=string	Source of raster values:	
		attr - read values from attribute table	
		cat - use category values	
		val - use value specified by value option	
		z - use z coordinate (points or contours only)	
		dir - output as flow direction (lines only)	
		Options: <i>attr,cat,val,z,dir</i>	
		Default: <i>attr</i>	
	column=name	Column name (type must be numeric)	
	layer=integer	Layer number	Default: 1
	value=float	Raster value	Default: 1
	rows=integer	number of rows to hold in memory	Default: 4096

r.composite : Combines red, green and blue map layers into a single composite map layer .

SYNOPSIS

r.composite [-dc] **red=string green=string blue=string** [**levels=integer**]
[**lev_red=integer**] [**lev_green=integer**] [**lev_blue=integer**] **output=string**
[**--overwrite**]

Flags: **-d** Dither
 -c Use closest color
 --overwrite Force overwrite of output files

Parameters: **red=string** Name of raster map layer to be used for.
 green=string Name of raster map layer to be used for.
 blue=string Name of raster map layer to be used for.
 levels=integer Number of levels to be used for each component.
 Options: 1-256 Default: 32
 lev_red=integer Number of levels to be used for. Options: 1-256
 lev_green=integer Number of levels to be used for. Options: 1-256
 lev_blue=integer Number of levels to be used for. Options: 1-256
 output=string Name of raster map to contain results .

r.out.tiff : Exports a GRASS raster file to a 8/24bit TIFF image file at the pixel resolution of the currently defined region.

SYNOPSIS

r.out.tiff [-ptlv] **input**=*string* **output**=*string* [**compression**=*string*]

Flags:

- p** TIFF Palette output (8bit instead of 24bit).
- t** Output TIFF world file
- l** Output Tiled TIFF
- v** Verbose mode.

Parameters:

input = <i>string</i>	Existing raster file name
output = <i>string</i>	File name for new TIFF file.
compression = <i>string</i>	TIFF file compression
	Options: <i>none,packbit,deflate,lzw</i>
	Default: <i>none</i>

d.what.rast : Allows the user to interactively query the category contents of multiple raster map layers at user specified locations within the current geographic region.

SYNOPSIS

d.what.rast [-1tc] **map**=*name*[,*name*,...] [**fs**=*character*]

Flags:

- 1** Identify just one location
- t** Terse output. For parsing by programs
- c** Print out col/row for the entire map in grid resolution of the region

Parameters:

map = <i>name</i> [, <i>name</i> ,...]	Name of existing raster map(s)
fs = <i>character</i>	Field separator (terse mode only)
	Default: :

r.mapcalc : performs arithmetic on raster map layers. New raster map layers can be created which are arithmetic expressions involving existing raster map layers, integer or floating point constants, and functions

operators in *r.mapcalc*

Operator	Meaning	Type	Priority
^	Exponent	arithmetical	5
%	Rate (Modulo)	arithmetical	4
/	Division	arithmetical	4
*	Multiplication	arithmetical	4
+	Addition	arithmetical	3
-	Subtraction	arithmetical	3
==	equal	logical	2
!=	unequal	logical	2
>	greater than	logical	2
>=	greater than or equal	logical	2
<	less than	logical	2
<=	less than or equal	logical	2
&&	and	logical	1
	or	logical	1
#	pre-separation operator	arithmetical	-

features in *r.mapcalc*

Feature	Meaning	Type
abs(x)	return absolute value of x	* ^a
atan(x)	inverse tangent of x (result is in degrees)	F ^b
atan(x,y)	inverse tangent of y/x (result is in degrees)	F
cos(x)	cosine of x (x is in degrees)	F
double(x)	convert x to double-precision floating point	F
eval([x,y,...],z)	evaluate values of listed expr, pass results to z	
exp(x)	exponential function of x	F
exp(x,y)	x to the power y	F
float(x)	convert x to floating point	F
graph(x,x1,y1[x2,y2..])	convert the x to a y based on points in a graph	F
if	decision options:	*
if(x)	1 if x not zero, 0 otherwise	
if(x,a)	a if x not zero, 0 otherwise	
if(x,a,b)	a if x not zero, b otherwise	
if(x,a,b,c)	a if x > 0, b if x is zero, c if x < 0	
int(x)	convert x to integer [truncates]	
isnull(x)	check if x = NULL	I
log(x)	natural log of x	F
log(x,b)	log of x base b	F
max(x,y[,z...])	largest value of those listed	*
median(x,y[,z...])	median value of those listed	*
min(x,y[,z...])	smallest value of those listed	*
mode(x,y[,z...])	mode value of those listed	*
not(x)	1 if x is zero, 0 otherwise	
rand(a,b)	random value between a and b	
round(x)	round x to nearest integer	I ^c
sin(x)	sine of x (x is in degrees)	F
sqrt(x)	square root of x	F
tan(x)	tangent of x (x is in degrees)	F

internal variables in *r.mapcalc*

Variable	Meaning
row()	current row of moving window
col()	current col of moving window
x()	current x-coordinate of moving window
y()	current y-coordinate of moving window
ewres()	current east-west resolution
nsres()	current north-south resolution
null()	NULL value

g.remove : Removes data base element files from the user's current mapset.

SYNOPSIS

g.remove [**rast**=string[,string,...]] [**rast3d**=string[,string,...]] [**vect**=string[,string,...]]
[**oldvect**=string[,string,...]] [**asciivect**=string[,string,...]] [**icon**=string[,string,...]]
[**labels**=string[,string,...]] [**sites**=string[,string,...]] [**region**=string[,string,...]]
[**region3d**=string[,string,...]] [**group**=string[,string,...]] [**3dview**=string[,string,...]]

Parameters:	rast =string[,string,...]	rast file(s) to be removed
	rast3d =string[,string,...]	rast3d file(s) to be removed
	vect =string[,string,...]	vect file(s) to be removed
	oldvect =string[,string,...]	oldvect file(s) to be removed
	asciivect =string[,string,...]	asciivect file(s) to be removed
	icon =string[,string,...]	icon file(s) to be removed
	labels =string[,string,...]	labels file(s) to be removed
	sites =string[,string,...]	sites file(s) to be removed
	region =string[,string,...]	region file(s) to be removed
	region3d =string[,string,...]	region3d file(s) to be removed
	group =string[,string,...]	group file(s) to be removed
	3dview =string[,string,...]	3dview file(s) to be removed

r.describe : Prints terse list of category values found in a raster map layer.

SYNOPSIS

r.describe [-1rqdi] map=*string* [nv=*string*] [nsteps=*integer*]

Flags:

- 1** Print the output one value per line
- r** Only print the range of the data
- q** Quiet
- d** Use the current region
- i** Read fp map as integer
- overwrite** Force overwrite of output files

Parameters:

map = <i>string</i>	Name of raster map
nv = <i>string</i>	string representing no data cell value Default: *
nsteps = <i>integer</i>	number of quantization steps Default: 255

r.rescale.eq : Rescales histogram equalized the range of category values in a raster map layer.

SYNOPSIS

r.rescale.eq [-q] **input**=*string* [**from**=*min,max*] **output**=*string* **to**=*min,max* [**title**=*"phrase"*] [--overwrite] **Flags:**

Flags: **-q** Quiet
 --overwrite Force overwrite of output files

Parameters:	input = <i>string</i>	The name of the raster map to be rescaled
	from = <i>min,max</i>	The input data range to be rescaled (default: full range of input map)
	output = <i>string</i>	The resulting raster map name
	to = <i>min,max</i>	The output data range
	title = <i>"phrase"</i>	Title for new raster map

r.univar : Calculates univariate statistics from the non-null cells of a raster map.

SYNOPSIS

r.univar [-qg] map=*name*

Flags:

-q	Quiet mode
-g	Print the stats in shell script style

Parameters: **map=*name*** Name of input raster map

r.univar calculates univariate statistics of a raster map. This includes the number of cells counted, minimum and maximum cell values, range, arithmetic mean, population variance, standard deviation, and coefficient of variation.

r.null : Creates explicitly the NULL-value bitmap file.

SYNOPSIS

r.null [-fincr] **map=string** [**setnull=val[-val][,val[-val],...]**] [**null=float**]

Flags:

- f** Only do the work if the map is floating-point
- i** Only do the work if the map is integer
- n** Only do the work if the map doesn't have a NULL-value bitmap file
- c** create NULL-value bitmap file validating all data cells
- r** remove NULL-value bitmap file

Parameters:

map=string	Raster map for which to edit null file
setnull=val[-val][,val[-val],...]	List of cell values to be set to NULL
null=float	The value to replace the null value by

r.stats : Generates area statistics for raster map layers.

SYNOPSIS

r.stats [-1**AacplqnNgxCri**] **input**=string[,string,...] [**fs**=character|space|tab]
[**nv**=string] [**output**=string] [**nsteps**=integer]

Flags:

- 1 One cell (range) per line
- A Print averaged values instead of intervals
- a Print area totals
- c Print cell counts
- p Print APPROXIMATE percents (total percent may not be 100%)
- l Print category labels
- q Quiet
- n Suppress reporting of any NULLs
- N Suppress reporting of NULLs when all values are NULL
- g Print grid coordinates (east and north)
- x Print x and y (column and row)
- C Report for cats fp ranges (fp maps only)
- r Print raw indexes of fp ranges (fp maps only)
- i Read fp map as integer (use map's quant rules)

Parameters:	input = <i>string[,string,...]</i>	Raster input maps(s)	
	fs = <i>character space tab</i>	Output field separator	Default: <i>space</i>
	nv = <i>string</i>	String representing no data cell value	
		Default: *	
	output = <i>string</i>	Output file name	
	nsteps = <i>integer</i>	Number of fp subranges to collect stats from	
		Default: 255	

i.group : Creates and edits groups and subgroups of imagery files.

SYNOPSIS

i.group [-rl] **group**=*string* [**subgroup**=*string*] [**input**=*string*[,*string*,...]]

Flags: **-r** Remove selected files from specified group
 -l List files from specified (sub)group

Parameters:	group = <i>string</i>	Name of imagery group
	subgroup = <i>string</i>	Name of imagery sub-group
	input = <i>string</i> [, <i>string</i> ,...]	Name of raster(s) to include in group

i.gensig : Generates statistics for i.maxlik from raster map layer.

SYNOPSIS

i.gensig **trainingmap=string** **group=string** **subgroup=string** **signaturefile=string**

Parameters:	trainingmap=string	ground truth training map
	group=string	imagery group
	subgroup=string	subgroup containing image files
	signaturefile=string	resultant signature file

i.maxlik : An imagery function that classifies the cell spectral reflectances in imagery data based on the spectral signature information generated by either *i.cluster*, *i.class*, or *i.gensig*.

SYNOPSIS

i.maxlik [-q] **group**=*string* **subgroup**=*string* **sigfile**=*string* **class**=*string*
[**reject**=*string*]

Flags: **-q** Run quietly

Parameters:	group = <i>string</i>	Imagery group to be classified
	subgroup = <i>string</i>	Subgroup containing image files to be classified
	sigfile = <i>string</i>	Signatures to use for classification
	class = <i>string</i>	Raster map to hold classification results
	reject = <i>string</i>	Raster map to hold reject threshold results

r.digit : provides the user with a way to draw lines, areas, and circles on a monitor screen, and to save these features in a cell file

THE PROCESS:

1. Choose to define an area or line, quit, or finish. If you quit, the session exits with nothing created. If you choose to finish (*done*), you will be prompted for a new map name; the new map is then created.
2. If you choose to make an area or line you must identify the category number for that area or line.
3. Using the mouse trace the line or circumscribe the area; or, finish (go to Step 1).

i.cluster : An imagery function that generates spectral signatures for land cover types in an image using a clustering algorithm. The resulting signature file is used as input for i.maxlik, to generate an unsupervised image classification

SYNOPSIS

i.cluster [-q] **group**=*string* **subgroup**=*string* **sigfile**=*string* **classes**=*integer*
[**seed**=*string*] [**sample**=*row_interval,col_interval*] [**iterations**=*integer*]
[**convergence**=*float*] [**separation**=*float*] [**min_size**=*integer*] [**reportfile**=*string*]

Flags: **-q** Run quietly

Parameters:

group = <i>string</i>	Group of imagery files to be clustered	
subgroup = <i>string</i>	Subgroup name in the above group	
sigfile = <i>string</i>	File contains result signatures	
classes = <i>integer</i>	Initial number of classes	Options: 1-255
seed = <i>string</i>	File contains initial signatures	
sample = <i>row_interval,col_interval</i>	Sampling intervals (by row and col)	
		; default: ~10,000 pixels
iterations = <i>integer</i>	Maximum number of iterations	Default: 30
convergence = <i>float</i>	Percent convergence	Options: 0-100
		Default: 98.0

separation=*float*

Cluster separation

Default: *0.0*

min_size=*integer*

Minimum number of pixels in a class

Default: *17*

reportfile=*string*

Name of an output file to contain final report

i.pca : Principal components analysis (pca) program for image processing.

SYNOPSIS

i.pca **input=string[,string,...]** **output=string** [**rescale=min,max**] [--**overwrite**]

Flags:

--overwrite Force overwrite of output files

Parameters:

input=string[,string,...]

input layer name

output=string

output layer name

rescale=min,max

Rescaling range output (For no rescaling use 0,0)

Default: 0,255

nviz : Visualization and animation tool for GRASS data.

Appendix B

How to use shell script

1. How to create shell script

Step 1. Create script by using text editor

Step 2. save it in work directly as _____.sh

Step 3. make the script

```
> chmod u+x _____.sh
```

Step 4. run the script

```
> sh _____.sh
```

2. How to use GRASS commands in shell script

```
grass60 /mnt/sda1/gis_knoppix/work/gis/latlon/PERMANENT <<EOF
```



write down GRASS commands

```
EOF
```


3. Shell script of Image import

```
grass60 /mnt/sda1/work/gis/latlon/PERMANENT <<EOF
```

```
r.in.bin input=/mnt/sda1/gis_knoppix/data/MOD_Vietnam2007040110_  
HKM_CH1.raw output=CH1 bytes=2 north=24 south=8 east=110 west=102  
rows=3840 cols=1920
```

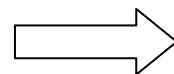
```
r.in.bin input=/mnt/sda1/gis_knoppix/data/MOD_Vietnam2007040110_  
HKM_CH2.raw output=CH2 bytes=2 north=24 south=8 east=110 west=102  
rows=3840 cols=1920
```

.....

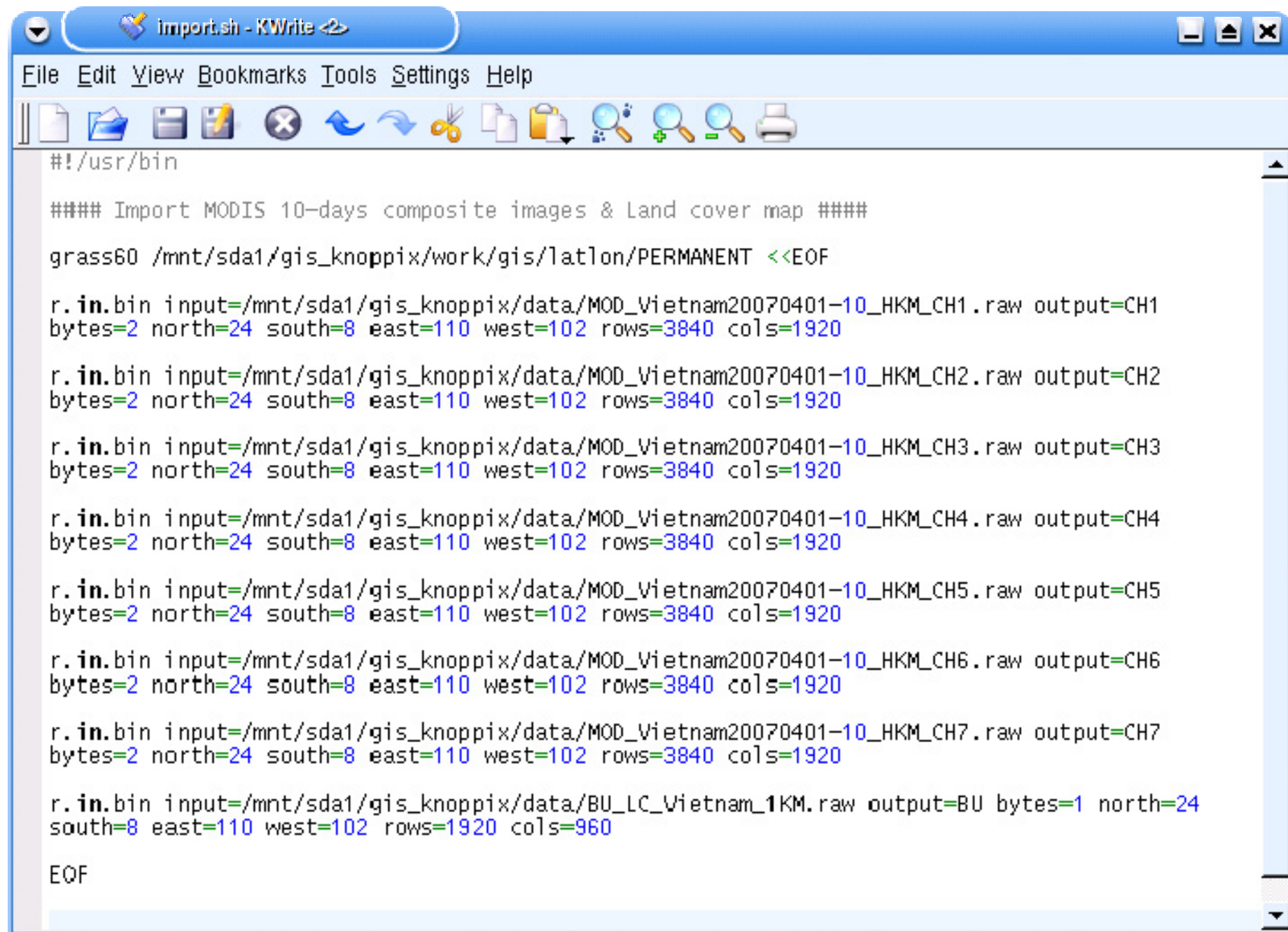
```
r.in.bin input=/mnt/sda1/gis_knoppix/data/MOD_Vietnam2007040110_  
HKM_CH7.raw output=CH7 bytes=2 north=24 south=8 east=110 west=102  
rows=3840 cols=1920
```

```
r.in.bin input=/mnt/sda1/gis_knoppix/data/BU_LC_Vietnam_1KM.raw output=BU  
bytes=1 north=24 south=8 east=110 west=102 rows=1920 cols=960
```

```
EOF
```



save as “ import.sh ”



```
#!/usr/bin

#### Import MODIS 10-days composite images & Land cover map ####

grass60 /mnt/sda1/gis_knoppix/work/gis/latlon/PERMANENT <<EOF

r.in.bin input=/mnt/sda1/gis_knoppix/data/MOD_Vietnam20070401-10_HKM_CH1.raw output=CH1
bytes=2 north=24 south=8 east=110 west=102 rows=3840 cols=1920

r.in.bin input=/mnt/sda1/gis_knoppix/data/MOD_Vietnam20070401-10_HKM_CH2.raw output=CH2
bytes=2 north=24 south=8 east=110 west=102 rows=3840 cols=1920

r.in.bin input=/mnt/sda1/gis_knoppix/data/MOD_Vietnam20070401-10_HKM_CH3.raw output=CH3
bytes=2 north=24 south=8 east=110 west=102 rows=3840 cols=1920

r.in.bin input=/mnt/sda1/gis_knoppix/data/MOD_Vietnam20070401-10_HKM_CH4.raw output=CH4
bytes=2 north=24 south=8 east=110 west=102 rows=3840 cols=1920

r.in.bin input=/mnt/sda1/gis_knoppix/data/MOD_Vietnam20070401-10_HKM_CH5.raw output=CH5
bytes=2 north=24 south=8 east=110 west=102 rows=3840 cols=1920

r.in.bin input=/mnt/sda1/gis_knoppix/data/MOD_Vietnam20070401-10_HKM_CH6.raw output=CH6
bytes=2 north=24 south=8 east=110 west=102 rows=3840 cols=1920

r.in.bin input=/mnt/sda1/gis_knoppix/data/MOD_Vietnam20070401-10_HKM_CH7.raw output=CH7
bytes=2 north=24 south=8 east=110 west=102 rows=3840 cols=1920

r.in.bin input=/mnt/sda1/gis_knoppix/data/BU_LC_Vietnam_1KM.raw output=BU bytes=1 north=24
south=8 east=110 west=102 rows=1920 cols=960

EOF
```

4. Shell script for RGB image

```
grass60 /mnt/sda1/work/gis/latlon/PERMANENT <<EOF
```

```
r.colors map=CH1 color=grey
```

```
r.colors map=CH2 color=grey
```

```
r.composite red=CH1 green=CH2 blue=CH1 output=RGB
```

```
EOF
```

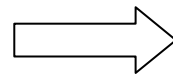
➡ save as “ RGB.sh ”

5. Shell script of NDVI calculation

```
grass60 /mnt/sda1/work/gis/latlon/PERMANENT <<EOF
```

```
r.mapcalc "NDVI=1.0*(CH2-CH1)/(CH2+CH1)"
```

```
EOF
```



save as “ NDVI.sh ”

6. Shell script of land cover classification

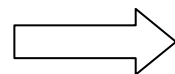
```
grass60 /mnt/sda1/work/gis/latlon/PERMANENT <<EOF
```

```
i.group group=MOD subgroup=subMOD input=CH1,CH2,CH3,CH4,CH5,CH6,  
CH7
```

```
i.gensig trainingmap=BU group=MOD subgroup=subMOD signaturefile=lcsig.sig
```

```
i.maxlik group=MOD subgroup=subMOD sigfile=lcsig.sig class=LCC.class
```

```
EOF
```



save as “ LCC.sh ”

7. Shell script of Image export in PNG format

```
grass60 /mnt/sda1/work/gis/latlon/PERMANENT <<EOF
```

```
d.mon start=PNG
```

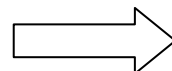
```
d.rast map=LCC.class
```

```
d.legend map=LCC.class
```

```
d.grid size=1
```

```
d.mon stop=PNG
```

```
EOF
```



save as “ export.sh ”

8. Main script

```
sh import.sh  
sh RGB.sh  
sh NDVI.sh  
sh LCC.sh  
sh export.sh
```

⇒ save as “ main.sh ”

If you want to skip a process, put # on the top of the line.

ex. If you want to skip creating RGB image,

⇒ **#** sh RGB.sh

Appendix C

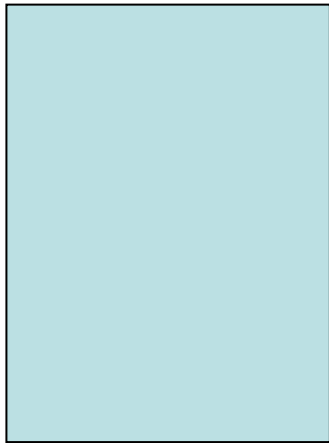
Data list

1. MODIS 10-days composite image (raster)

File name ; MOD_VietnamYYYYMMDD-DD_HKM_XXX.raw

DD-DD ; 01-10 / 11-20 / 21-30

XXX ; CH1~CH7, CH31, CH32, DOY, RAZ, SZA, VZA



spatial resolution = 500 m

byte number = 2 bytes

Max Lat = 24 (north)

Min Lat = 8 (south)

Max Lon = 110 (east)

Min Lon = 102 (west)

x size = 1920 (cols)

y size = 3840 (rows)

2. Land Cover Map (raster)

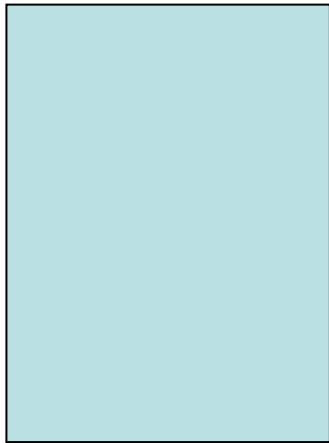
File name ; XXX_LC_Vietnam_1KM.raw

XXX ; UMD (University of Maryland)

; BU (Boston University)

; JRC (Joint Research Centre)

; USGS (U.S. Geological Survey)



spatial resolution = 1 km

byte number = 1 bytes

Max Lat = 24 (north)

Min Lat = 8 (south)

Max Lon = 110 (east)

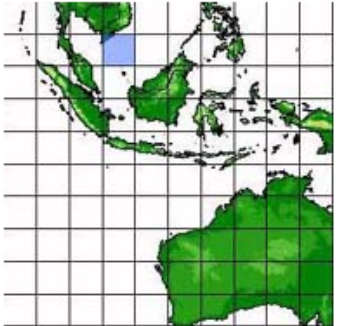

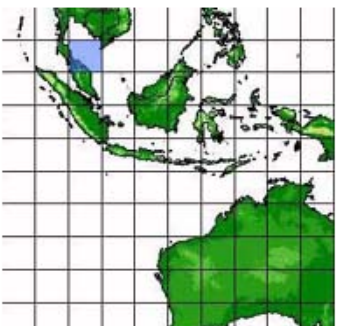
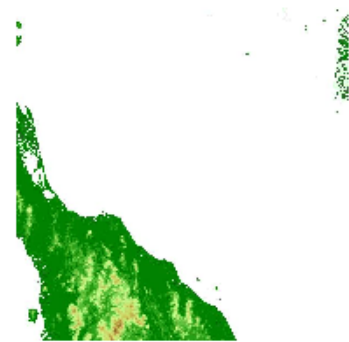
Min Lon = 102 (west)

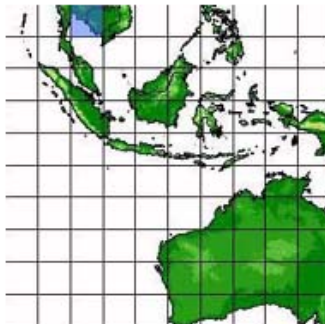
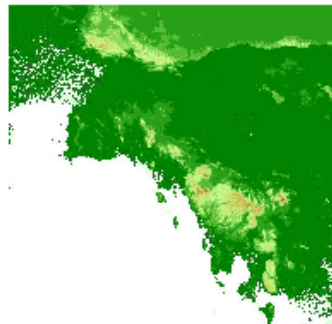
x size = 960 (cols)

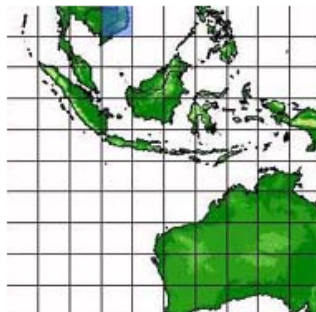
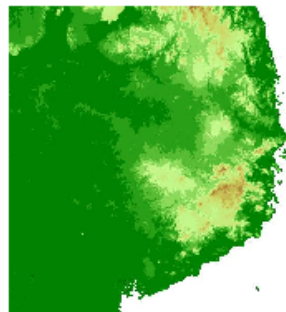
y size = 1920 (rows)

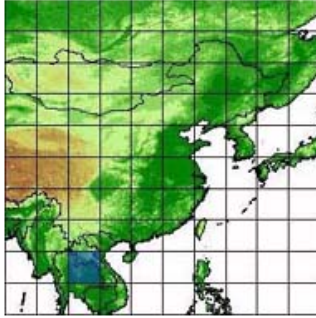
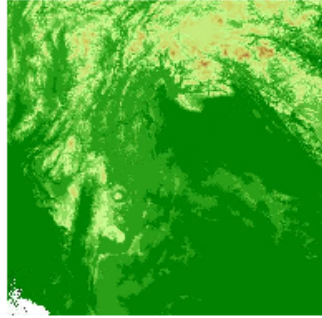
3. SRTM image (raster)

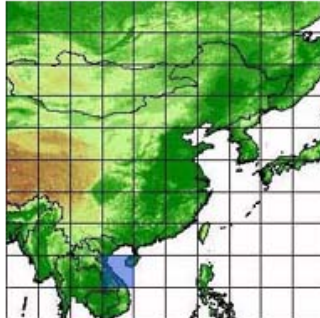
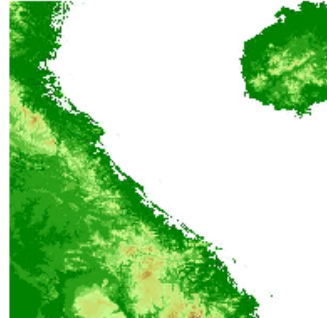
spatial resolution = 90 m

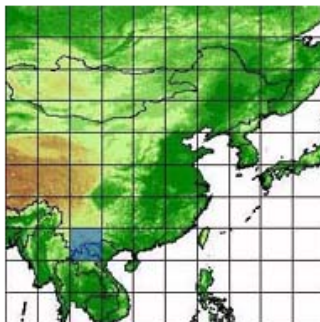
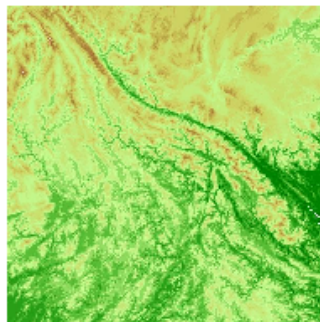
Description	Location	Image
<p>Product : SRTM 90m DEM</p> <p>Data File Name : srtm_58_11.zip</p> <p>Mask File Name: srtm_mk_58_11.zip</p> <p>Latitude min: 5 N max: 10 N</p> <p>Longitude min: 105 E max: 110 E</p> <p>Center point : Latitude 7.50 N Longitude 107.50 E</p>		
Description	Location	Image
<p>Product : SRTM 90m DEM</p> <p>Data File Name : srtm_57_11.zip</p> <p>Mask File Name: srtm_mk_57_11.zip</p> <p>Latitude min: 5 N max: 10 N</p> <p>Longitude min: 100 E max: 105 E</p> <p>Center point : Latitude 7.50 N Longitude 102.50 E</p>		

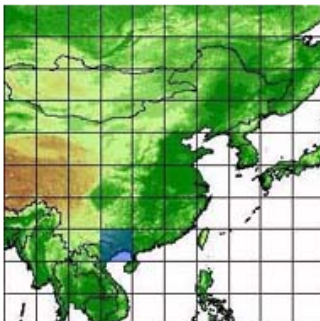
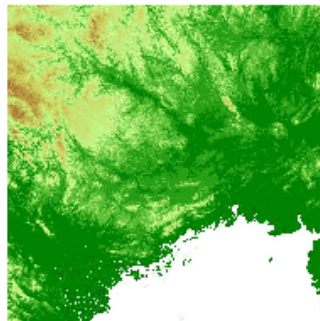
Description	Location	Image
<p>Product : SRTM 90m DEM</p> <p>Data File Name : srtm_57_10.zip</p> <p>Mask File Name: srtm_mk_57_10.zip</p> <p>Latitude min: 10 N max: 15 N</p> <p>Longitude min: 100 E max: 105 E</p> <p>Center point : Latitude 12.50 N Longitude 102.50 E</p>		

Description	Location	Image
<p>Product : SRTM 90m DEM</p> <p>Data File Name : srtm_58_10.zip</p> <p>Mask File Name: srtm_mk_58_10.zip</p> <p>Latitude min: 10 N max: 15 N</p> <p>Longitude min: 105 E max: 110 E</p> <p>Center point : Latitude 12.50 N Longitude 107.50 E</p>		

Description	Location	Image
<p>Product : SRTM 90m DEM</p> <p>Data File Name : srtm_57_09.zip</p> <p>Mask File Name: srtm_mk_57_09.zip</p> <p>Latitude min: 15 N max: 20 N</p> <p>Longitude min: 100 E max: 105 E</p> <p>Center point : Latitude 17.50 N Longitude 102.50 E</p>		

Description	Location	Image
<p>Product : SRTM 90m DEM</p> <p>Data File Name : srtm_58_09.zip</p> <p>Mask File Name: srtm_mk_58_09.zip</p> <p>Latitude min: 15 N max: 20 N</p> <p>Longitude min: 105 E max: 110 E</p> <p>Center point : Latitude 17.50 N Longitude 107.50 E</p>		

Description	Location	Image
<p>Product : SRTM 90m DEM</p> <p>Data File Name : srtm_57_08.zip</p> <p>Mask File Name: srtm_mk_57_08.zip</p> <p>Latitude min: 20 N max: 25 N</p> <p>Longitude min: 100 E max: 105 E</p> <p>Center point : Latitude 22.50 N Longitude 102.50 E</p>		

Description	Location	Image
<p>Product : SRTM 90m DEM</p> <p>Data File Name : srtm_58_08.zip</p> <p>Mask File Name: srtm_mk_58_08.zip</p> <p>Latitude min: 20 N max: 25 N</p> <p>Longitude min: 105 E max: 110 E</p> <p>Center point : Latitude 22.50 N Longitude 107.50 E</p>		

5. Asian countries (raster)

File name ; ASIA_HKM_ROI

spatial resolution = 500 m

byte number = 1 bytes

Max Lat = 60 (north)

Min Lat = -10 (south)

Max Lon = 150 (east)

Min Lon = 60 (west)

x size = 21600 (cols)

y size = 16800 (rows)

6. Land Sea Mask (raster)

File name ; LSM_Vietnam_HKM

spatial resolution = 500 m

byte number = 1 bytes

Max Lat = 24 (north)

Min Lat = 8 (south)

Max Lon = 110 (east)

Min Lon = 102 (west)

x size = 1920 (cols)

y size = 3840 (rows)

6. World border (vector)

File name ; world_borders.shp, world_borders.shx, world_borders.dbf

7. World administrative boundary (vector)

File name ; admin02.shp, admin02.shx, admin02.dbf

before 2003