

Space Applications For Environment

SAR 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 1.1

Mar. 11, 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-Knopix CD**
- 2. Copy data to FAT formatted storage device**
- 3. How to boot a PC from GIS-Knoppix CD**
- 4. Preparation for starting GRASS**

Contents

Section 2. SAR Image Processing in Latitude-Longitude Coordinate System

- 1. Starting and Terminating GRASS**
- 2. Defining coordinate system and region**
- 3. Import images**
- 4. Display images**
- 5. Raster calculations**
- 6. Create RGB image**
- 7. Terrain Correction**
- 8. Extract deforested areas using RGB image**

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?

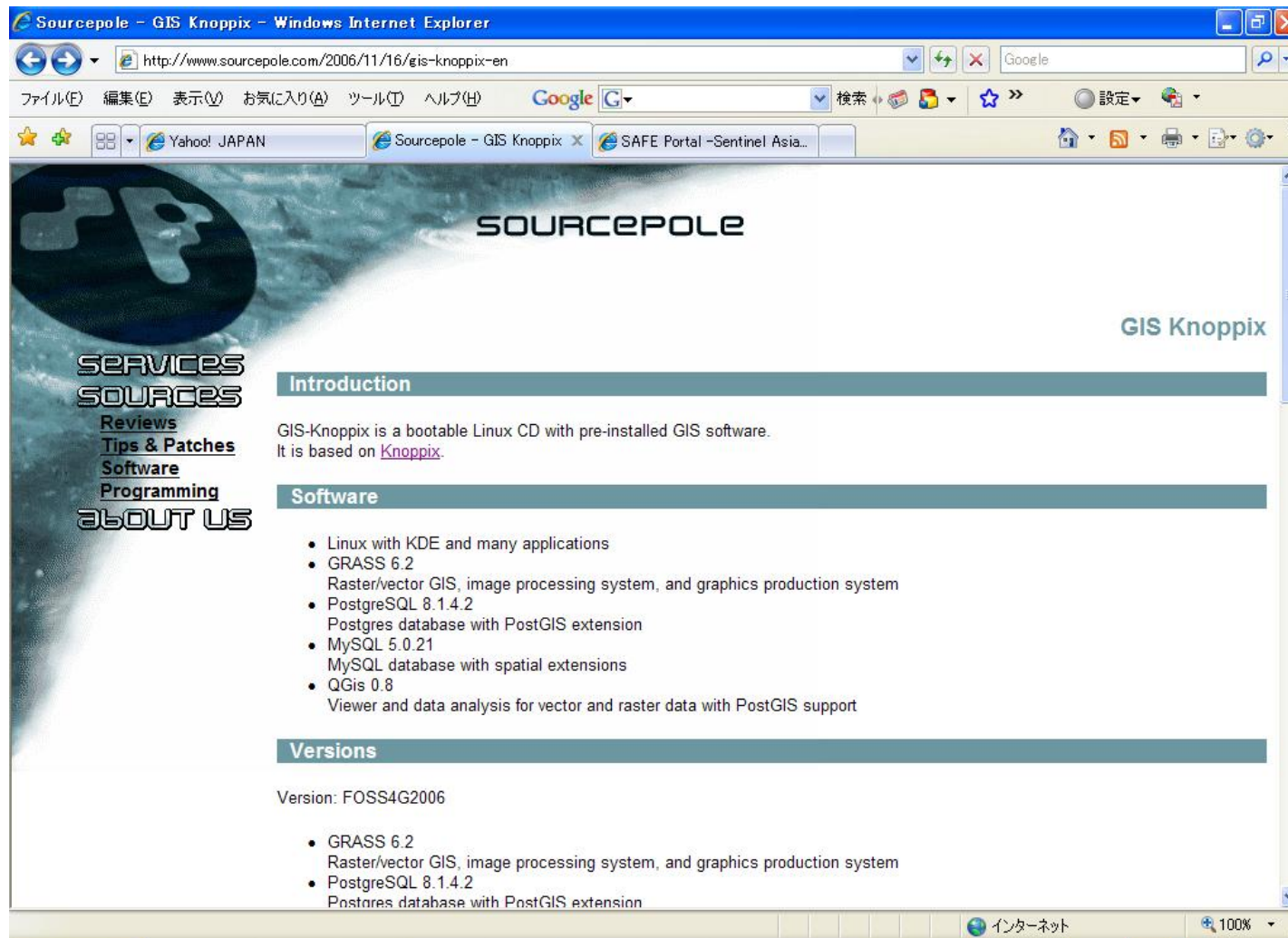
KNOPPIX is a bootable Live system on CD or DVD, consisting of a representative collection of GNU/Linux software, automatic hardware detection, and support for many graphics cards, sound cards, SCSI and USB devices and other peripherals. KNOPPIX can be used as a productive Linux system for the desktop, 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>)

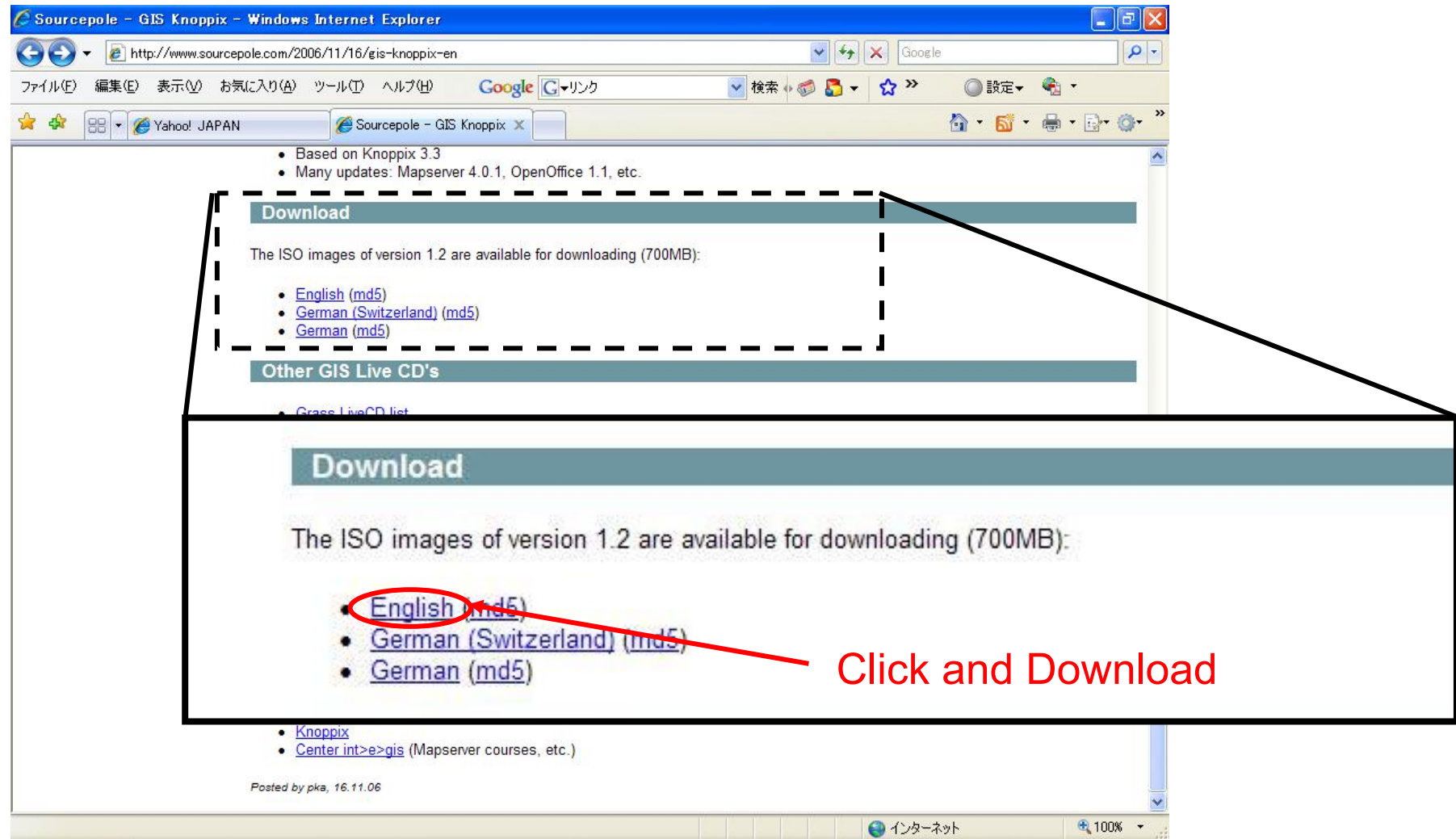
1.2 How to get GIS-Knoppix CD

Step 1: Go to

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



Step 2: Download the ISO images of version 1.2



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

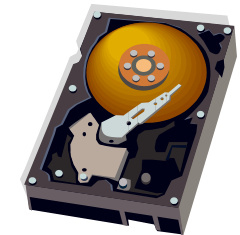
**Step 1: Make “gis_knoppix” directory
in your FAT formatted storage
device**

**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

- P2007_Vietnam_CH_100m_HH
- P2007_Vietnam_CH_100m_HV
- JSAR1997JF_Vietnam_CH_100m_ortho
- JSAR1998Aug_Vietnam_CH_100m_ortho
- SRTM3CGIAR_Vietnam_CH

from DVD

“ Vietnam/data/SAR ”

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.

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**

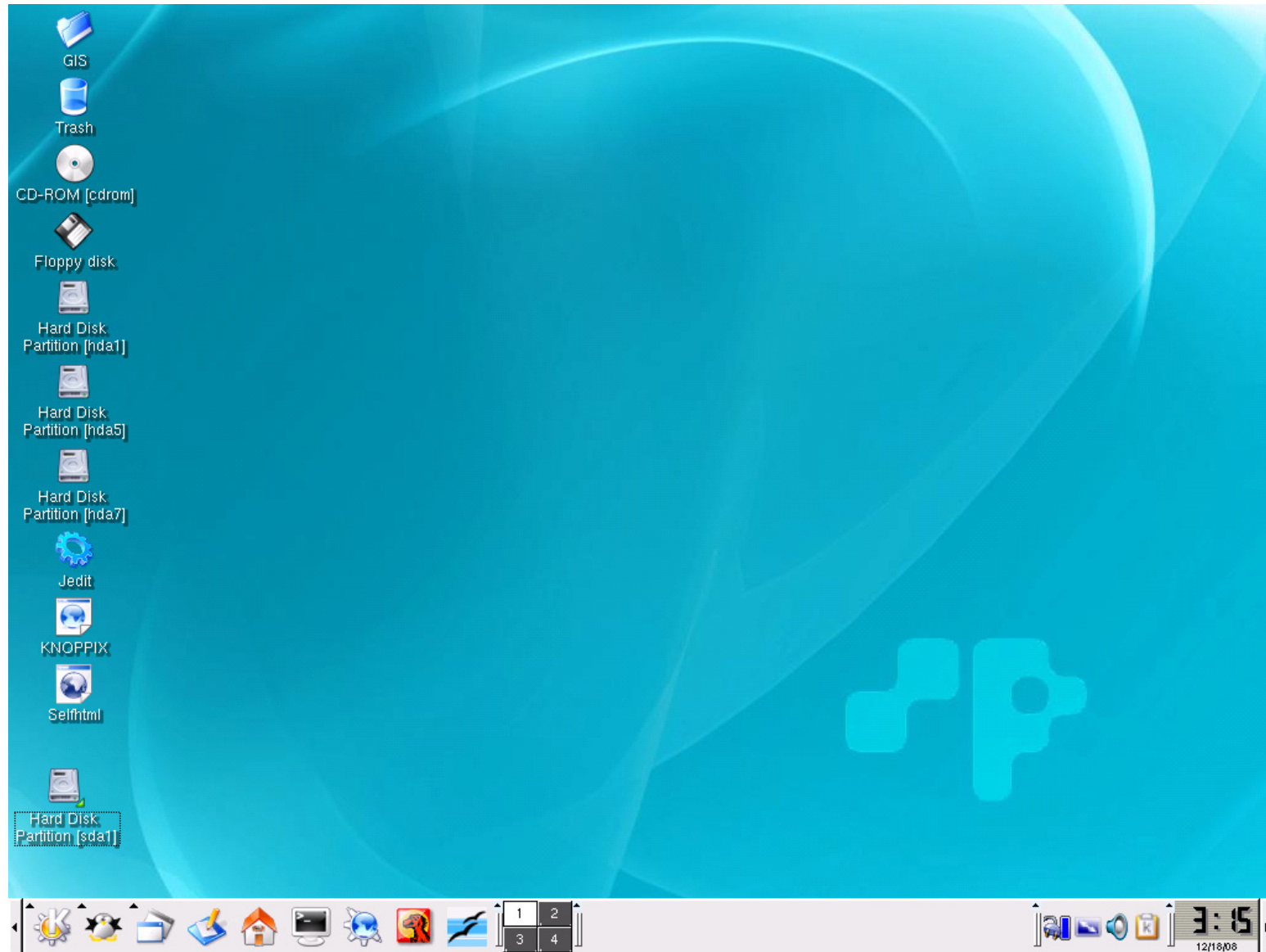
- 4.1 Boot PC from the GIS-knoppix CD

- 4.2 Change Keyboard layout

- 4.3 Make FAT formatted storage device writable

- 4.4b Move to “work” directory

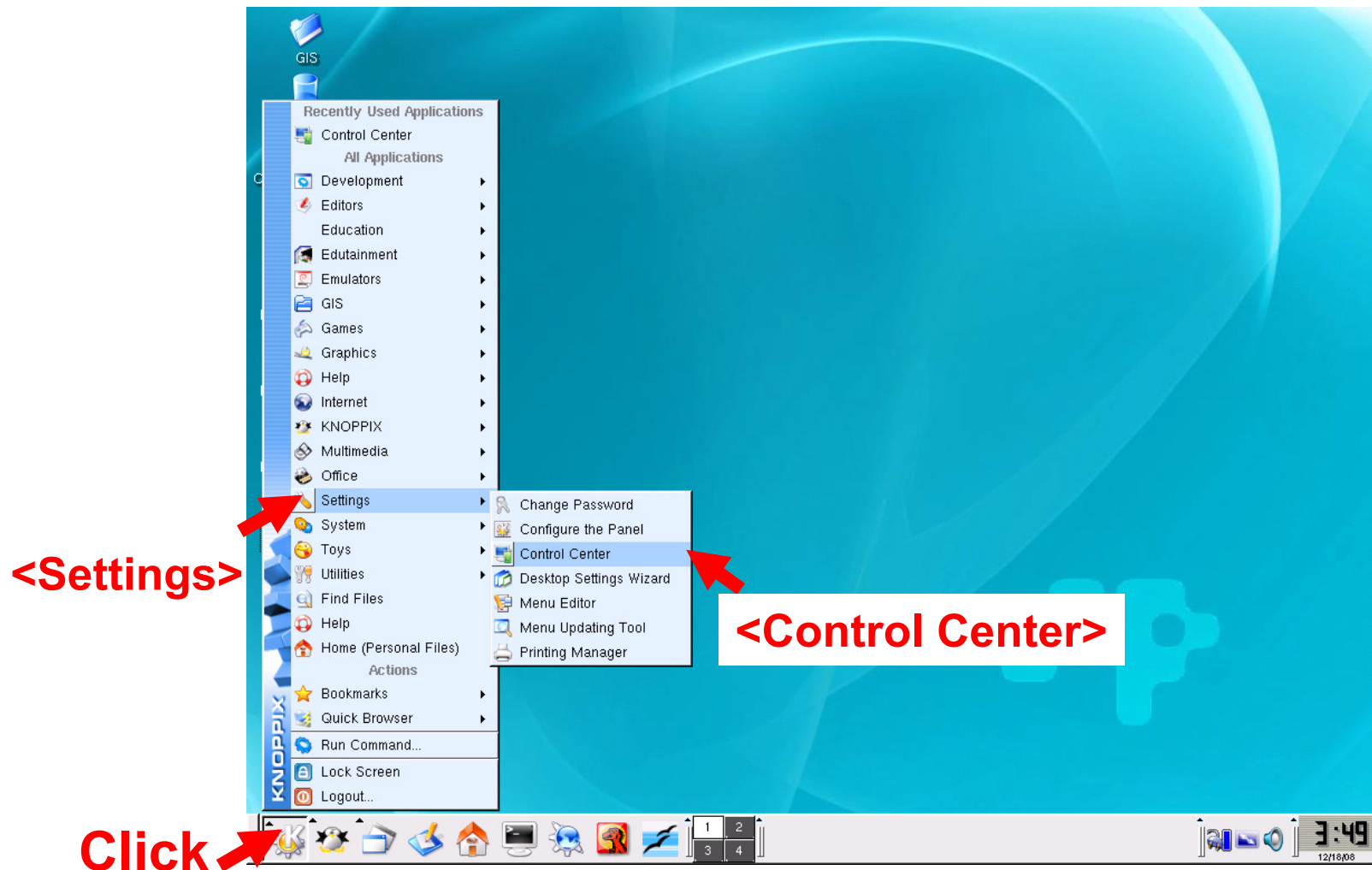
4.1 Boot PC from the GIS-knoppix CD



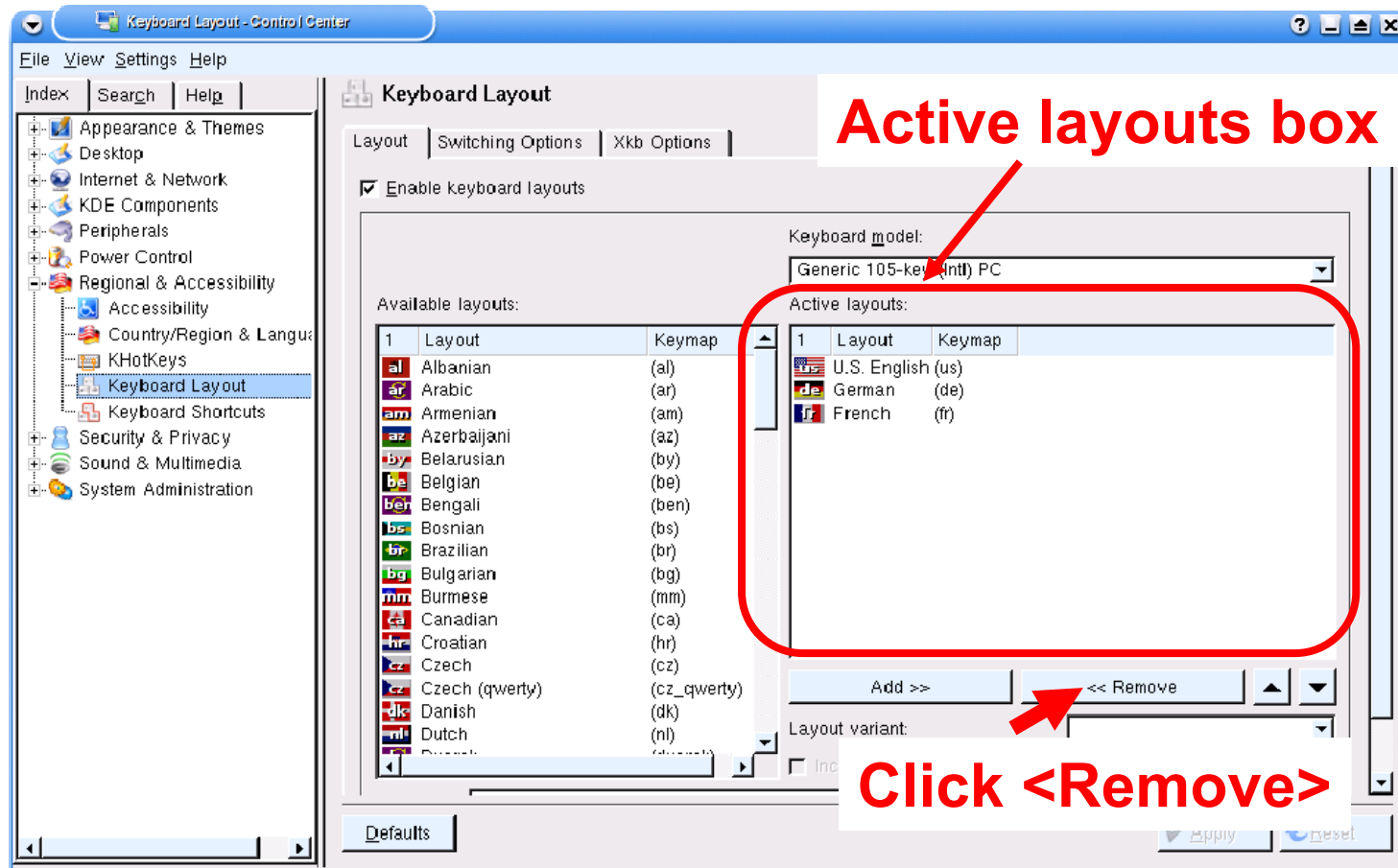
4.2 Change Keyboard layout

Step1 : Click icon at bottom-left corner

→ <Settings> → <Control Center>

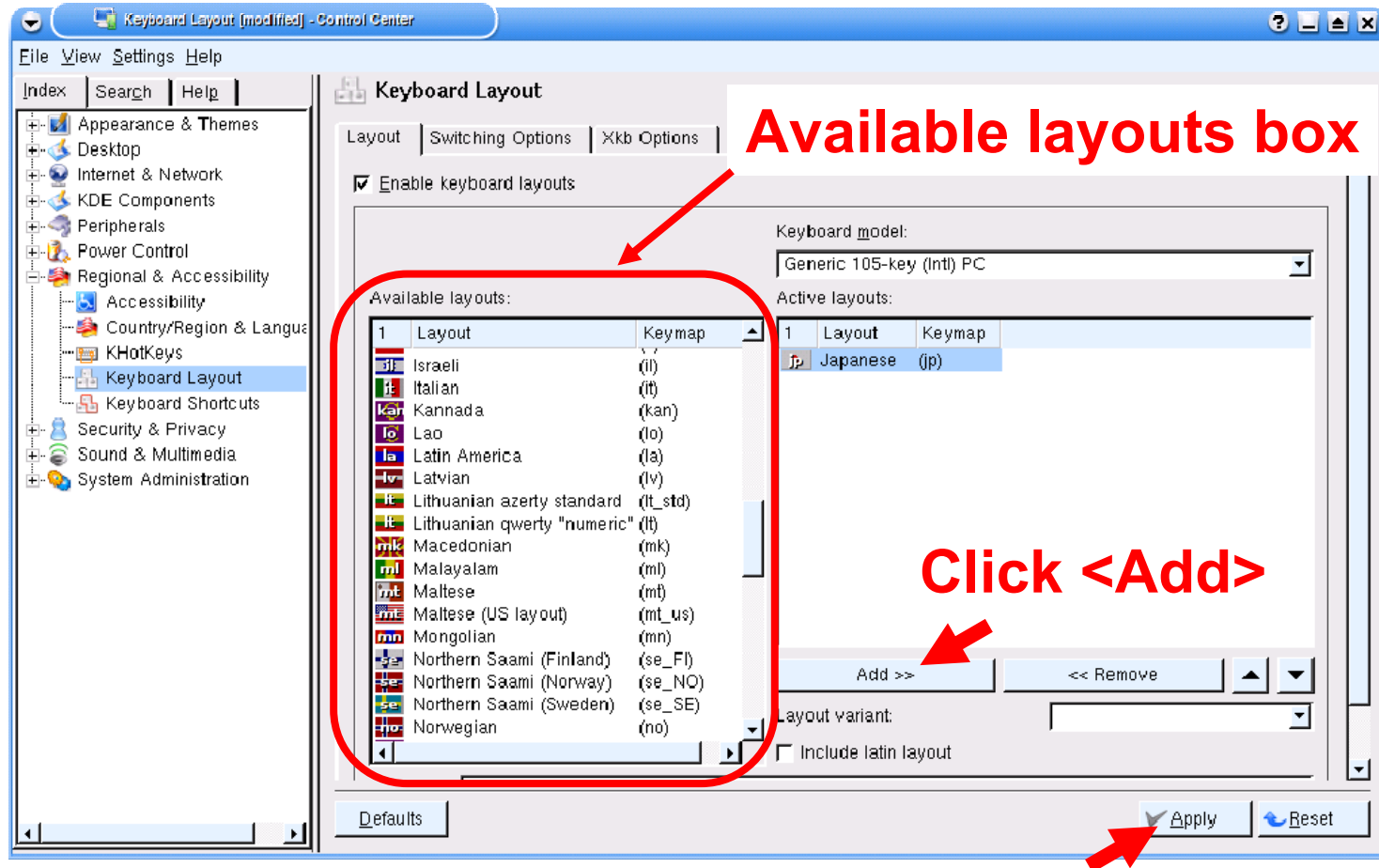


Step2 : Click <Regional & Accessibility >
→ Click <Keyboard Layout >



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

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

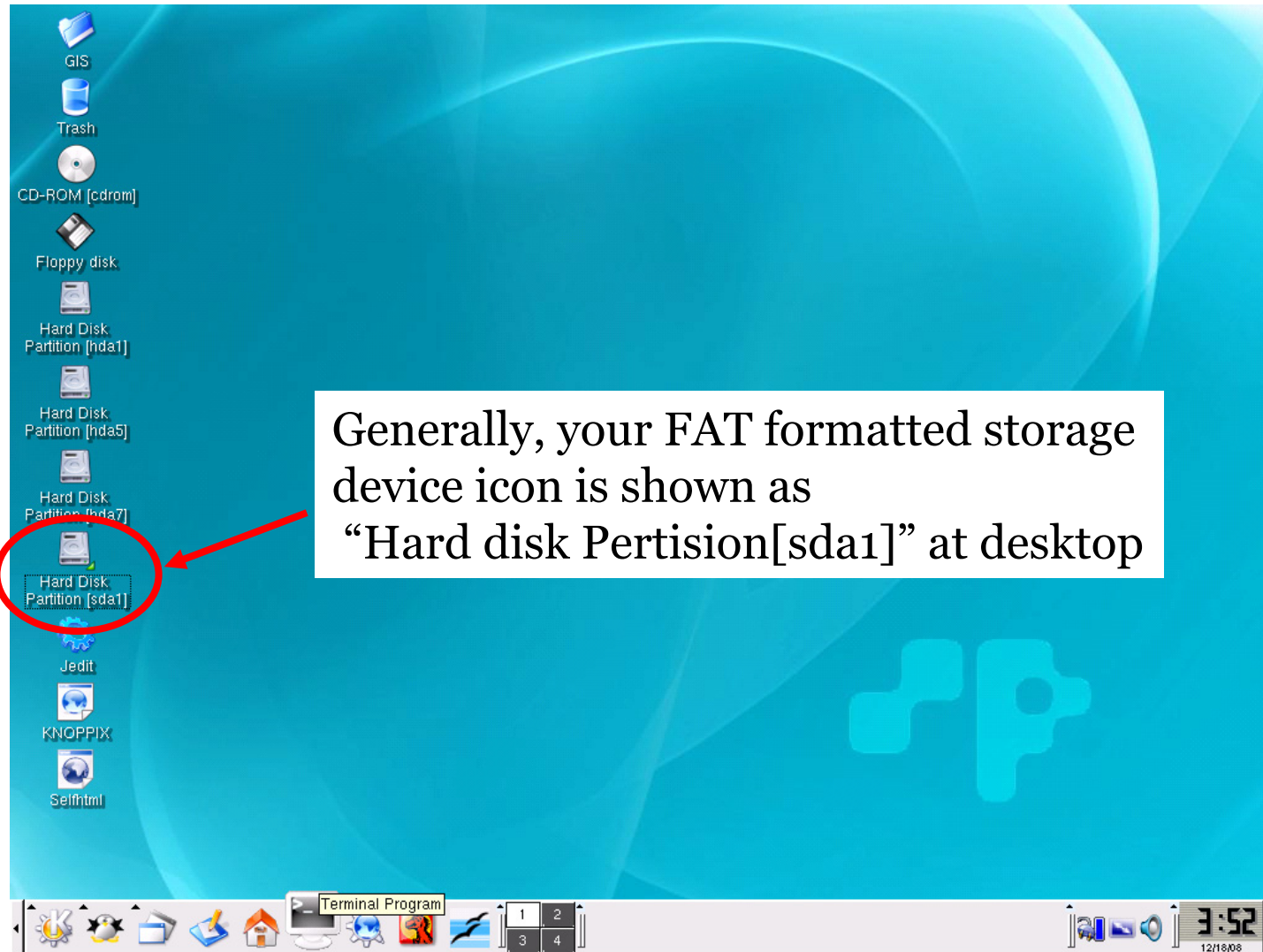


Step5 : Click <Apply >

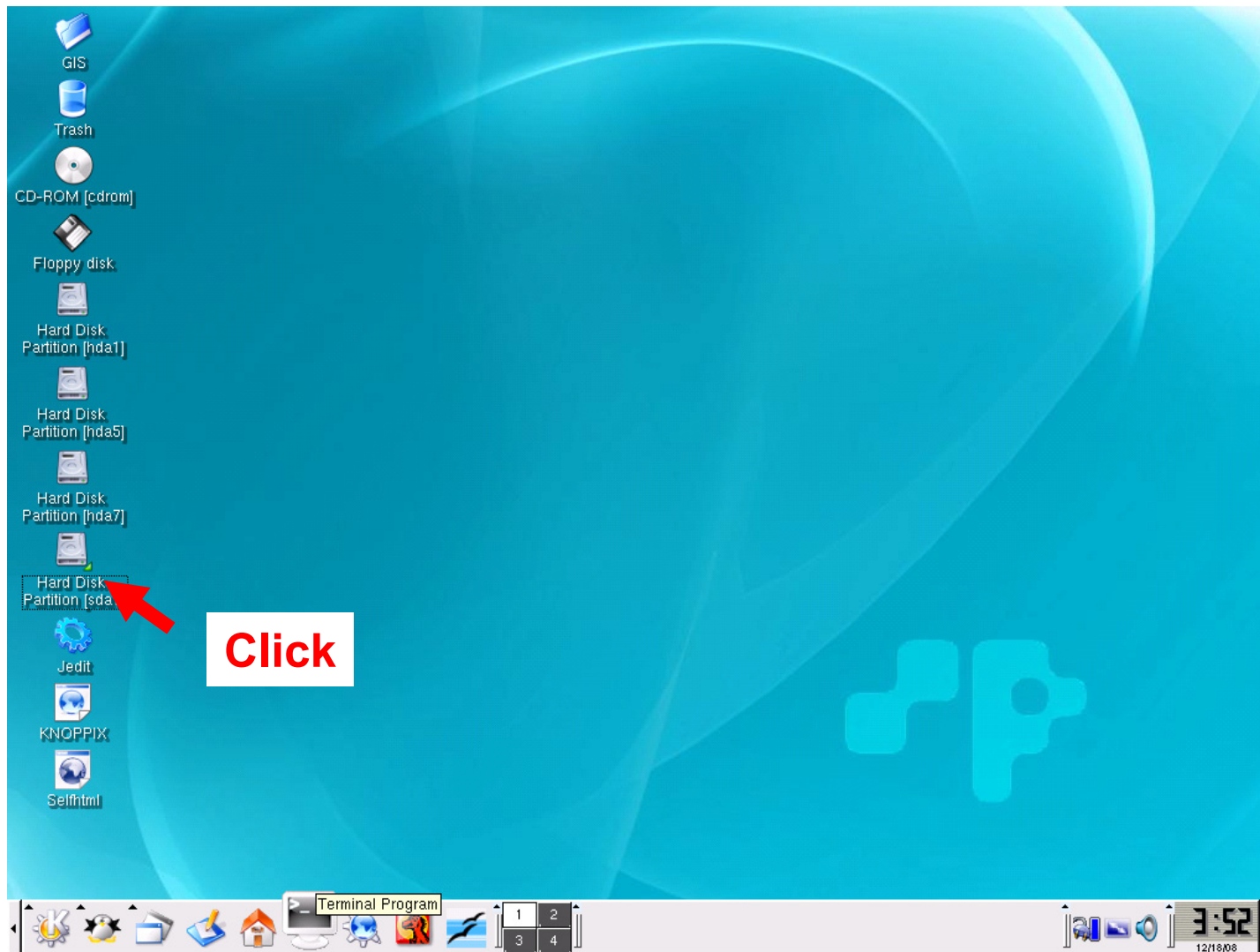
Click <Apply>

4.3 Make FAT formatted storage device writable

Step1 : Connect FAT formatted storage device to PC

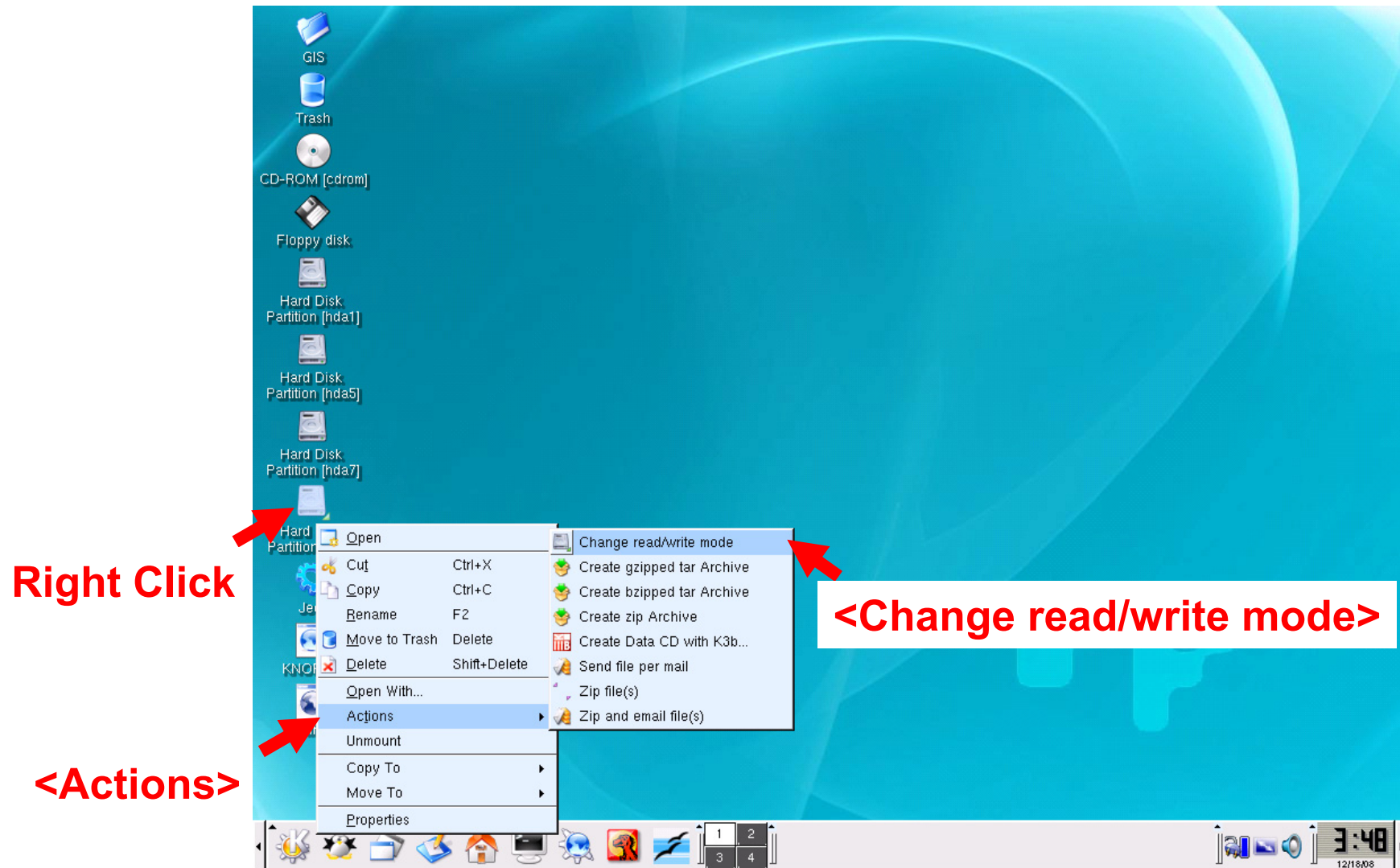


Step2 :Mount FAT formatted storage device [sda1] by clicking the [sda1] icon

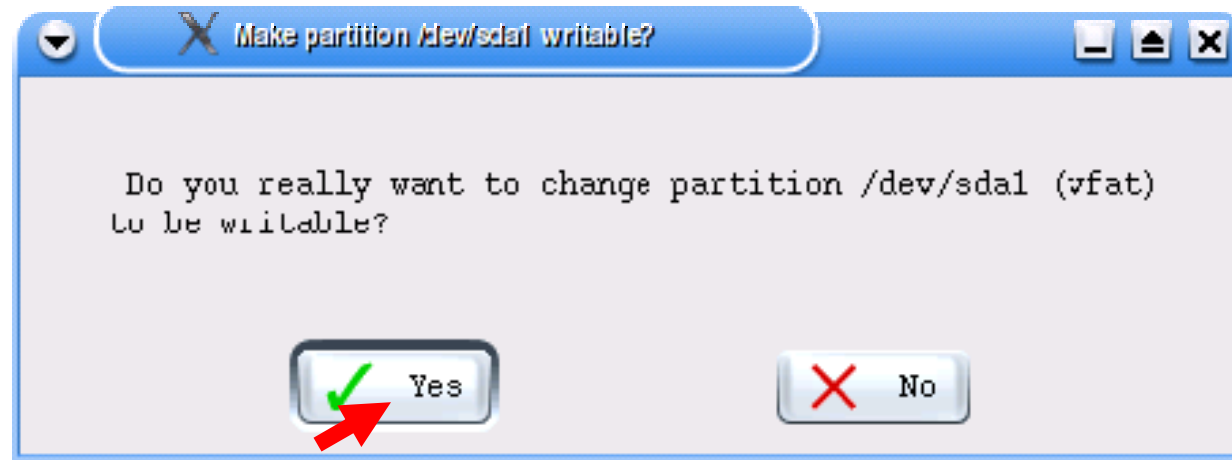


Step2 : Right click [sda1] icon on the desktop

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



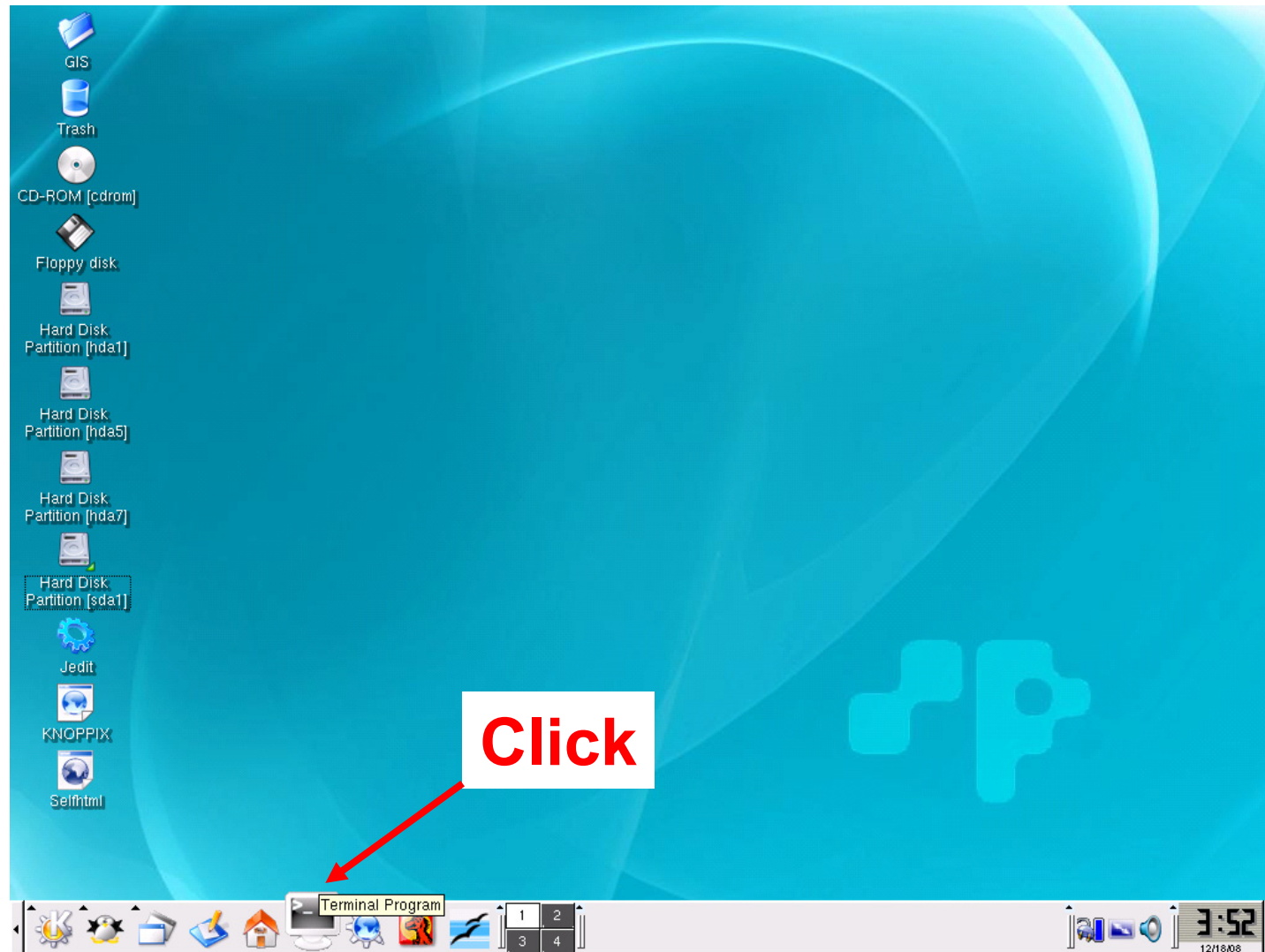
Step3 : Click <Yes >



Click

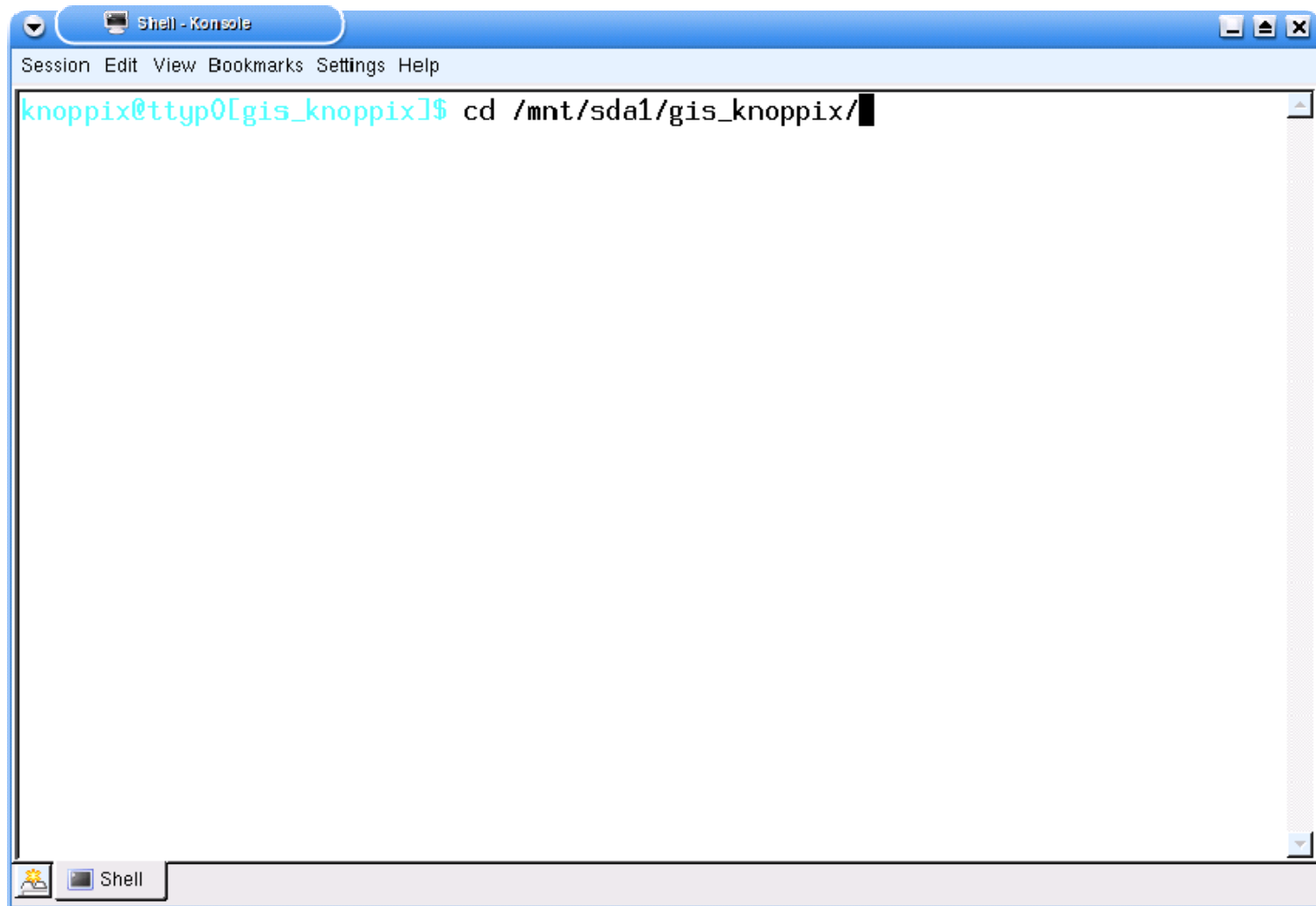
4.4a Make working directory

Step1 : Open “ Terminal Program ”

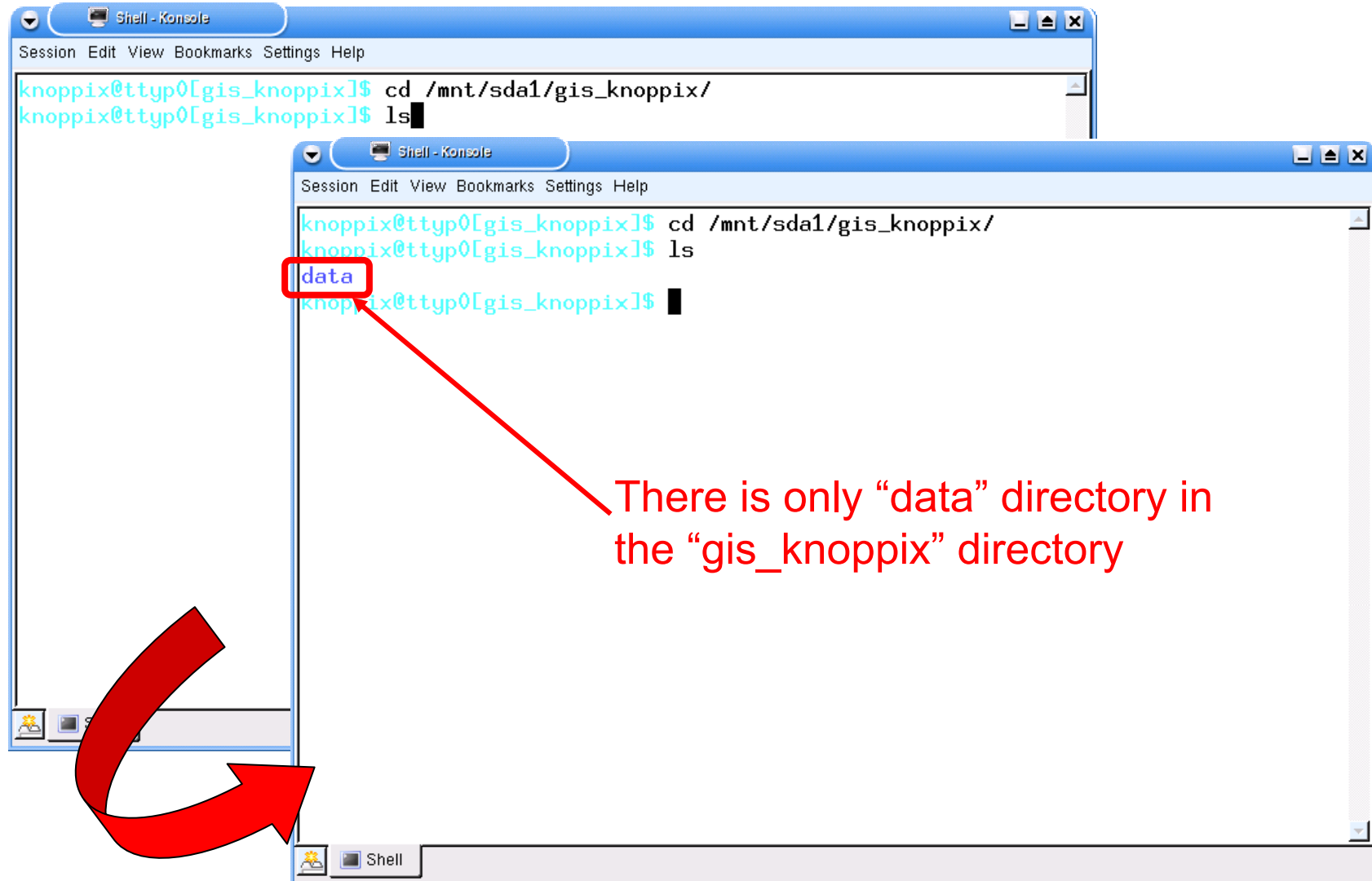


Step2 : On the Terminal window, write following command and hit <enter(return)> key

> 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
```

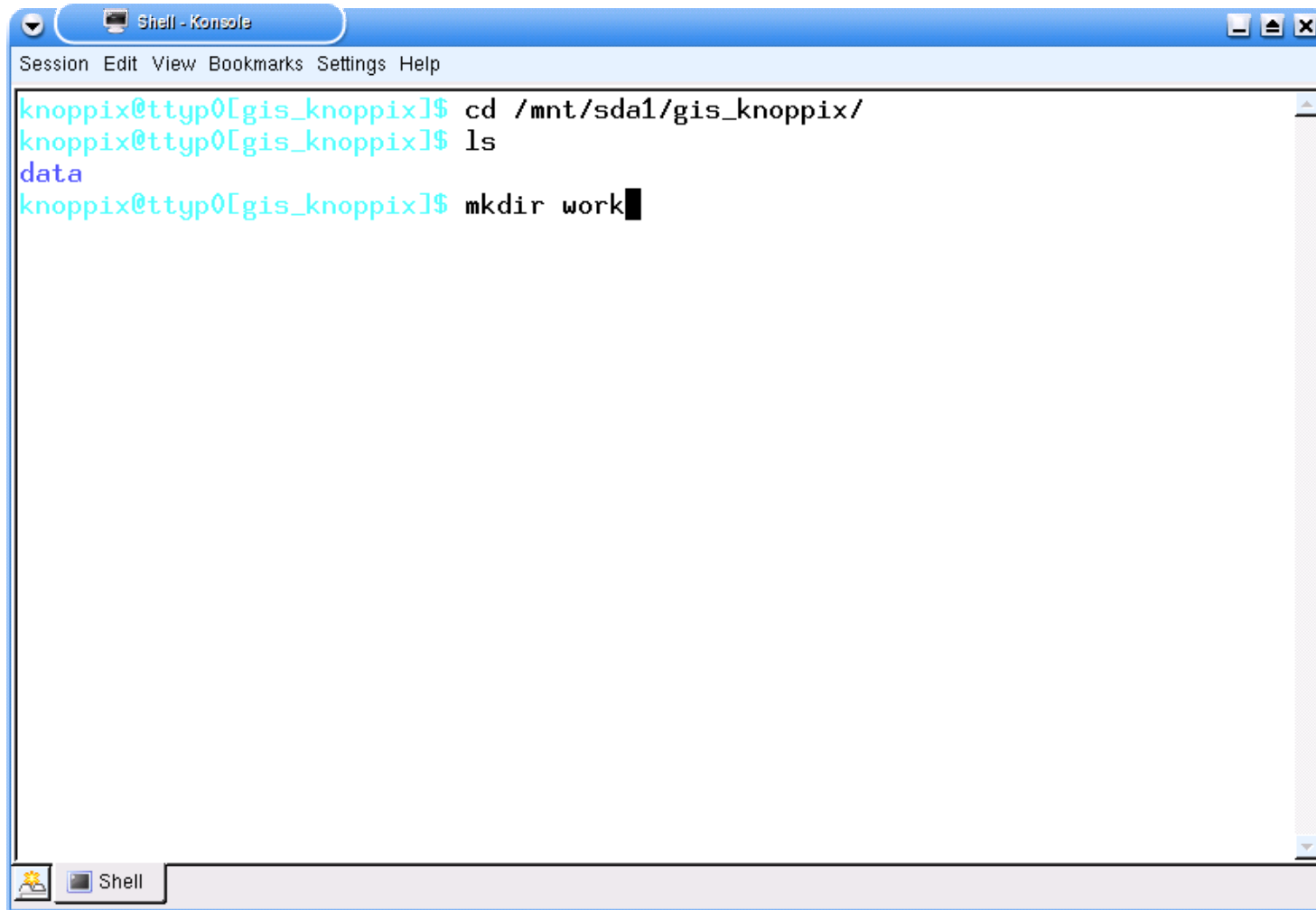
```
knoppix@tty0[gis_knoppix]$ cd /mnt/sda1/gis_knoppix/
knoppix@tty0[gis_knoppix]$ ls
data
knoppix@tty0[gis_knoppix]$
```

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

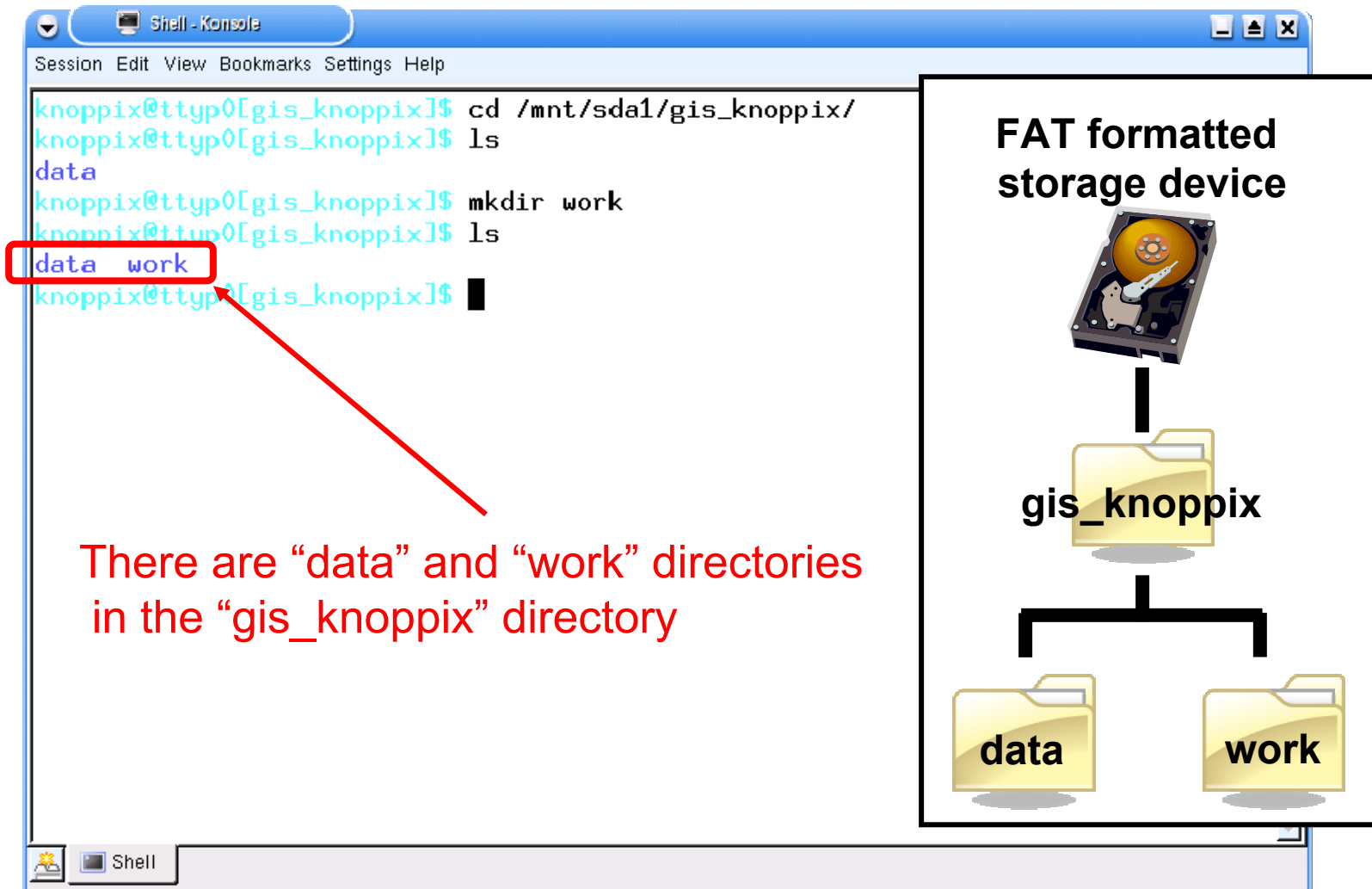
> mkdir work

: 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 box below the terminal.

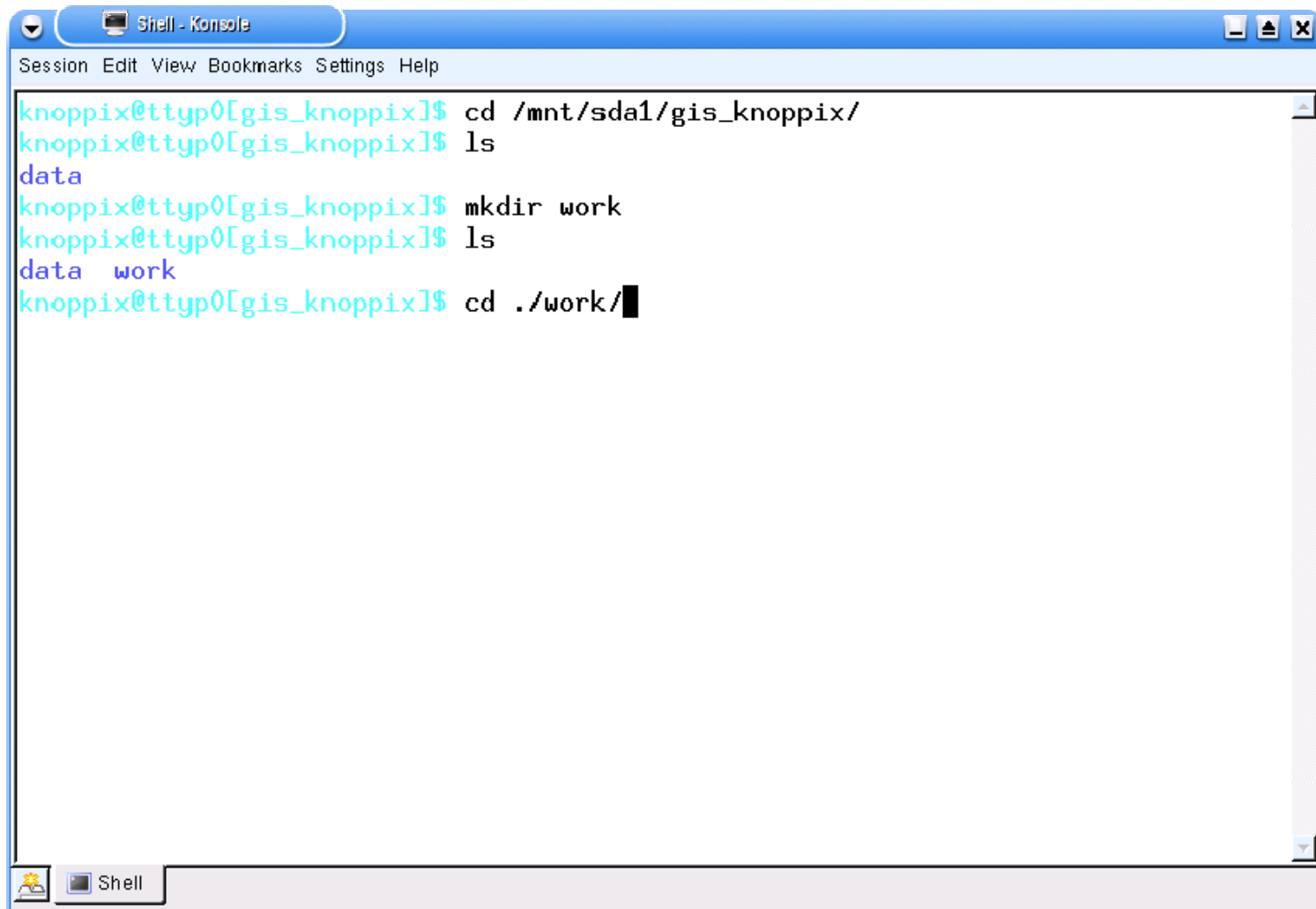
There are “data” and “work” directories in the “gis_knoppix” directory

To the right of the terminal, a diagram titled "FAT formatted storage device" illustrates the directory structure. 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".

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

> cd `.`./work

: space



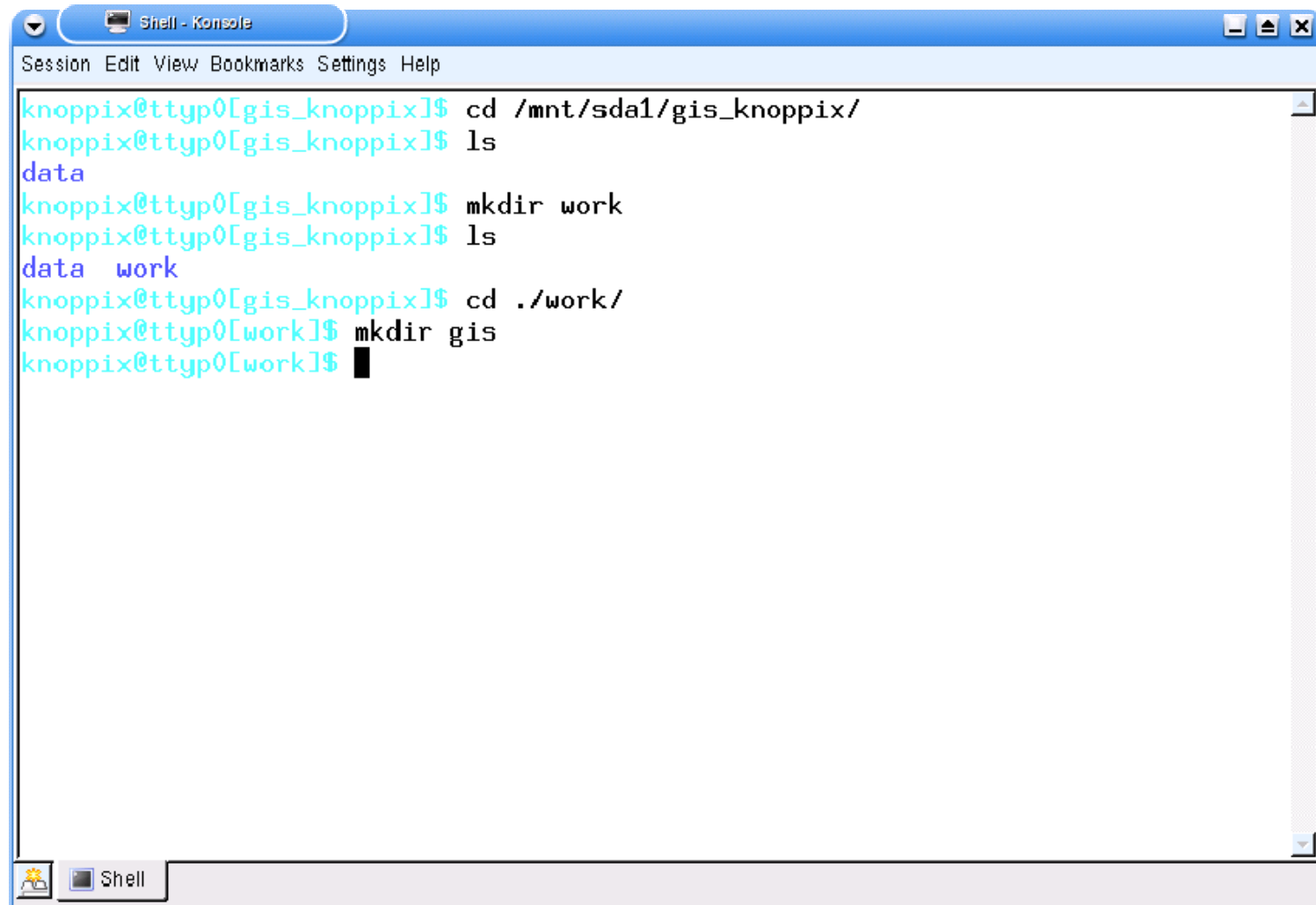
```
Shell - Konsole
Session Edit View Bookmarks Settings Help

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



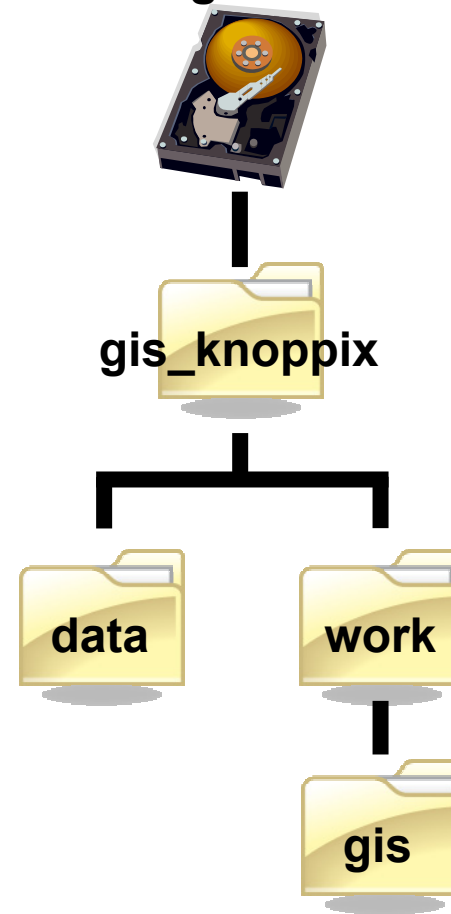
```
knoppix@tty0[gis_knoppix]$ cd /mnt/sda1/gis_knoppix/
knoppix@tty0[gis_knoppix]$ ls
data
knoppix@tty0[gis_knoppix]$ mkdir work
knoppix@tty0[gis_knoppix]$ ls
data  work
knoppix@tty0[gis_knoppix]$ cd ./work/
knoppix@tty0[work]$ mkdir gis
knoppix@tty0[work]$
```

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]$
```

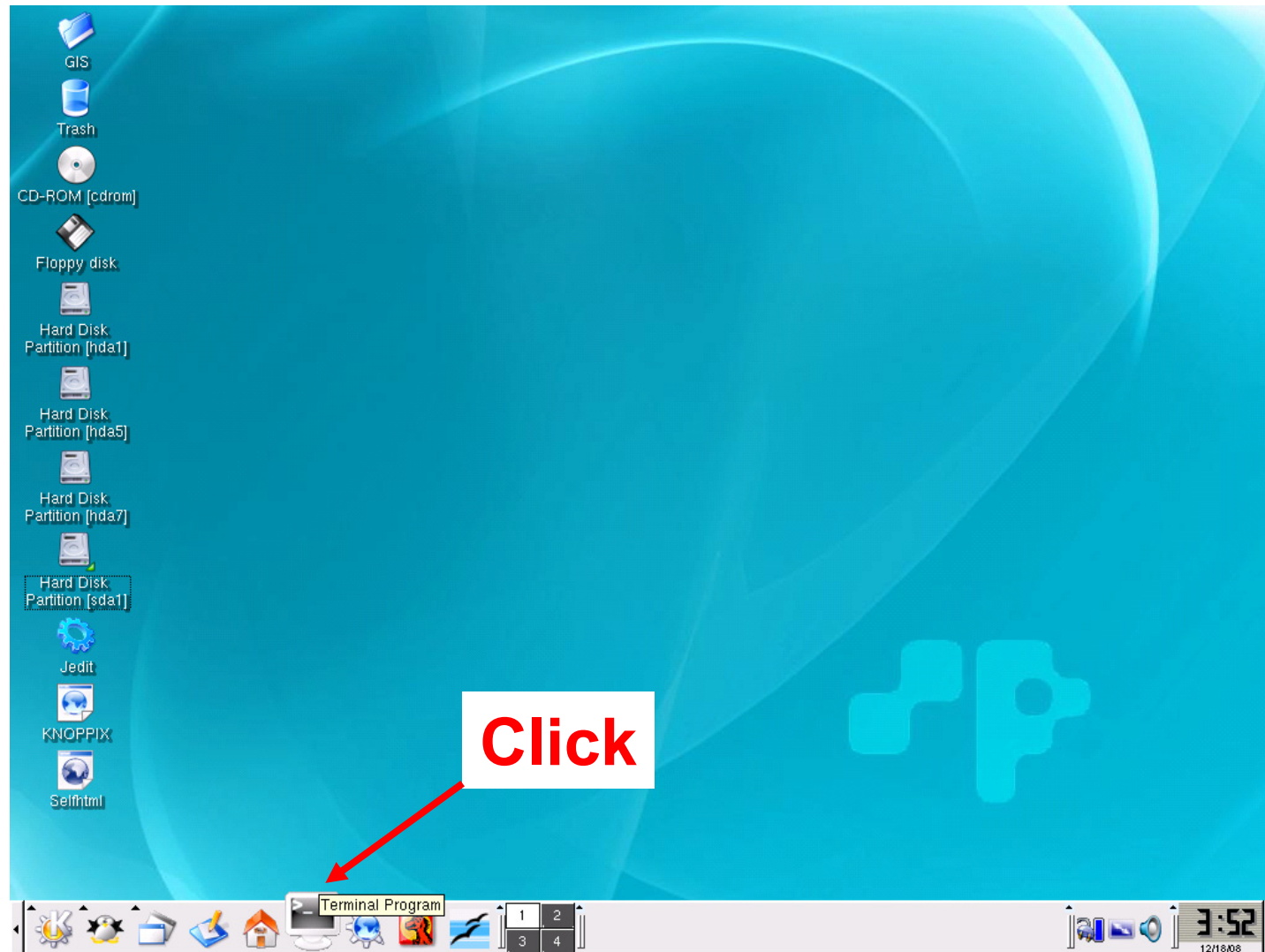
There is “gis” directory in the “work” directory

**FAT formatted
storage device**

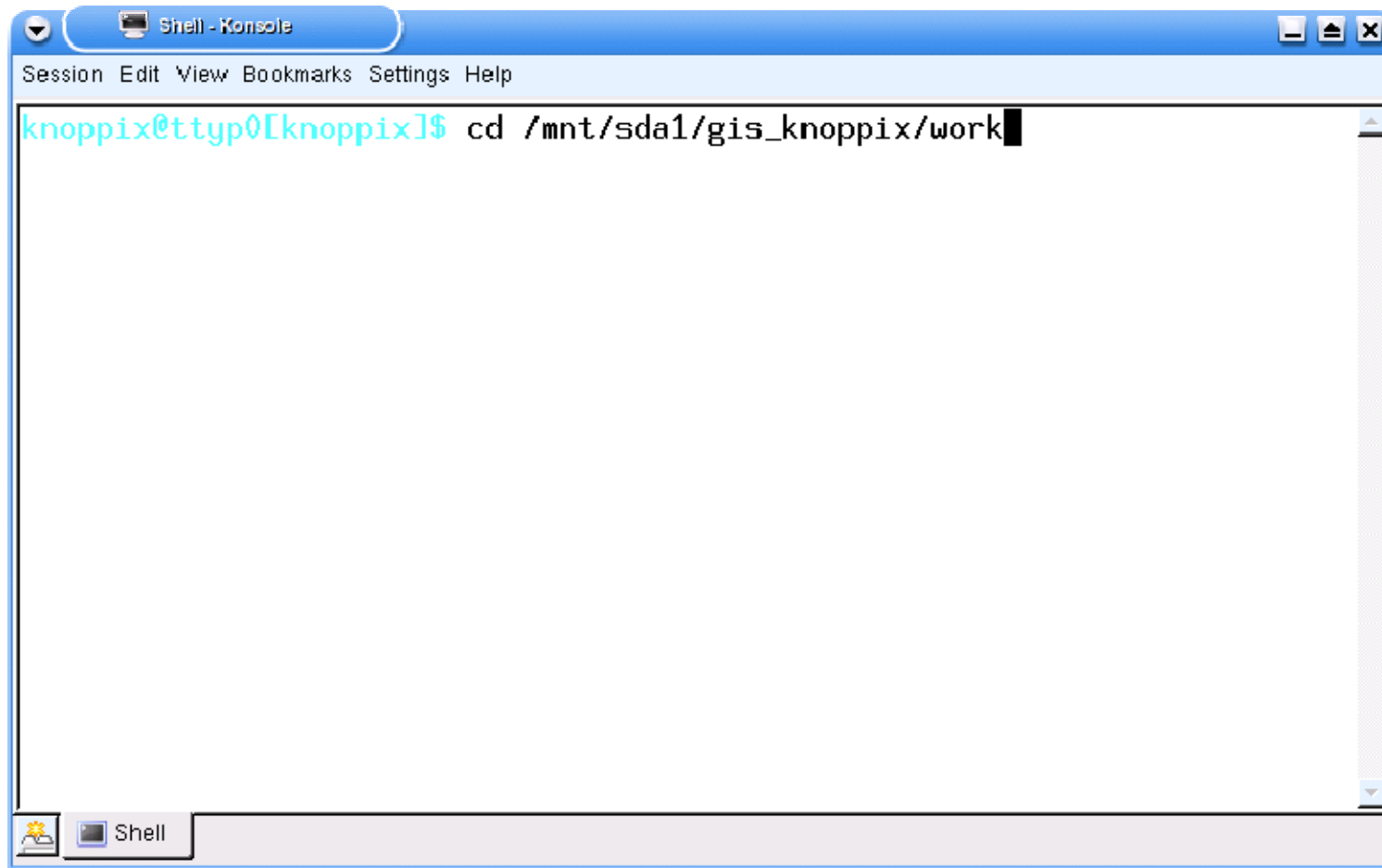


4.4b Move to “work” directory

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. SAR Image Processing in Latitude-Longitude Coordinate System

1. Starting and Terminating GRASS

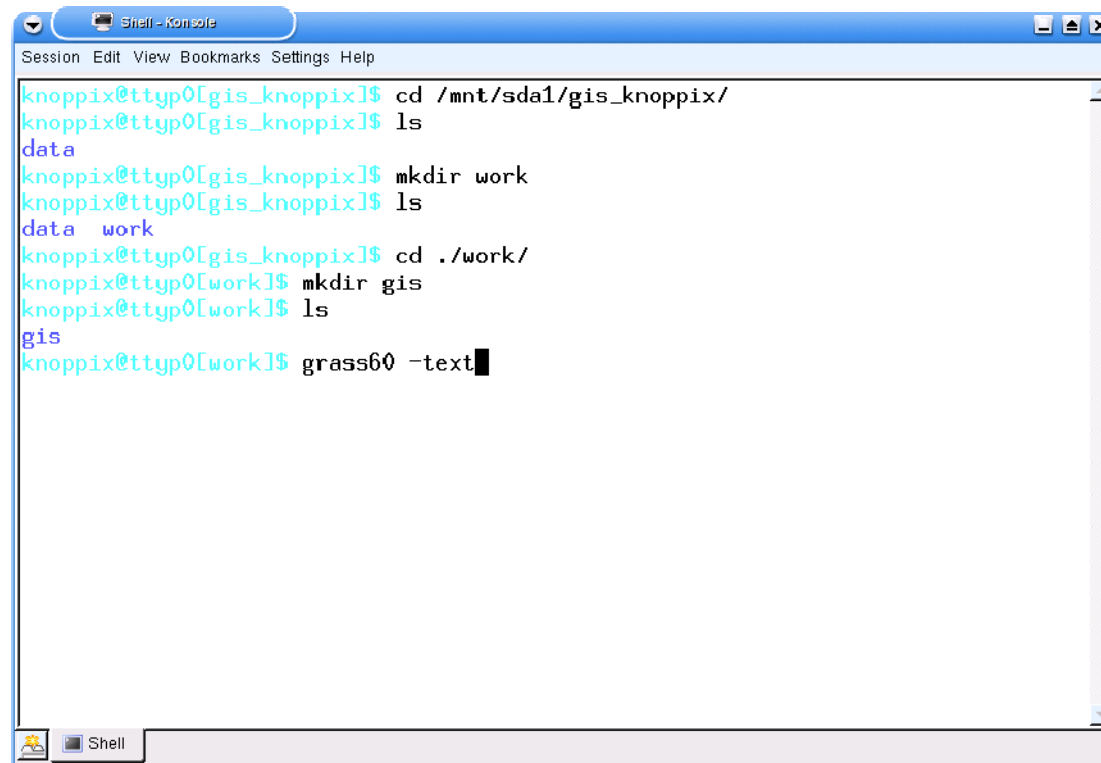
1.1 Starting GRASS

> 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
```

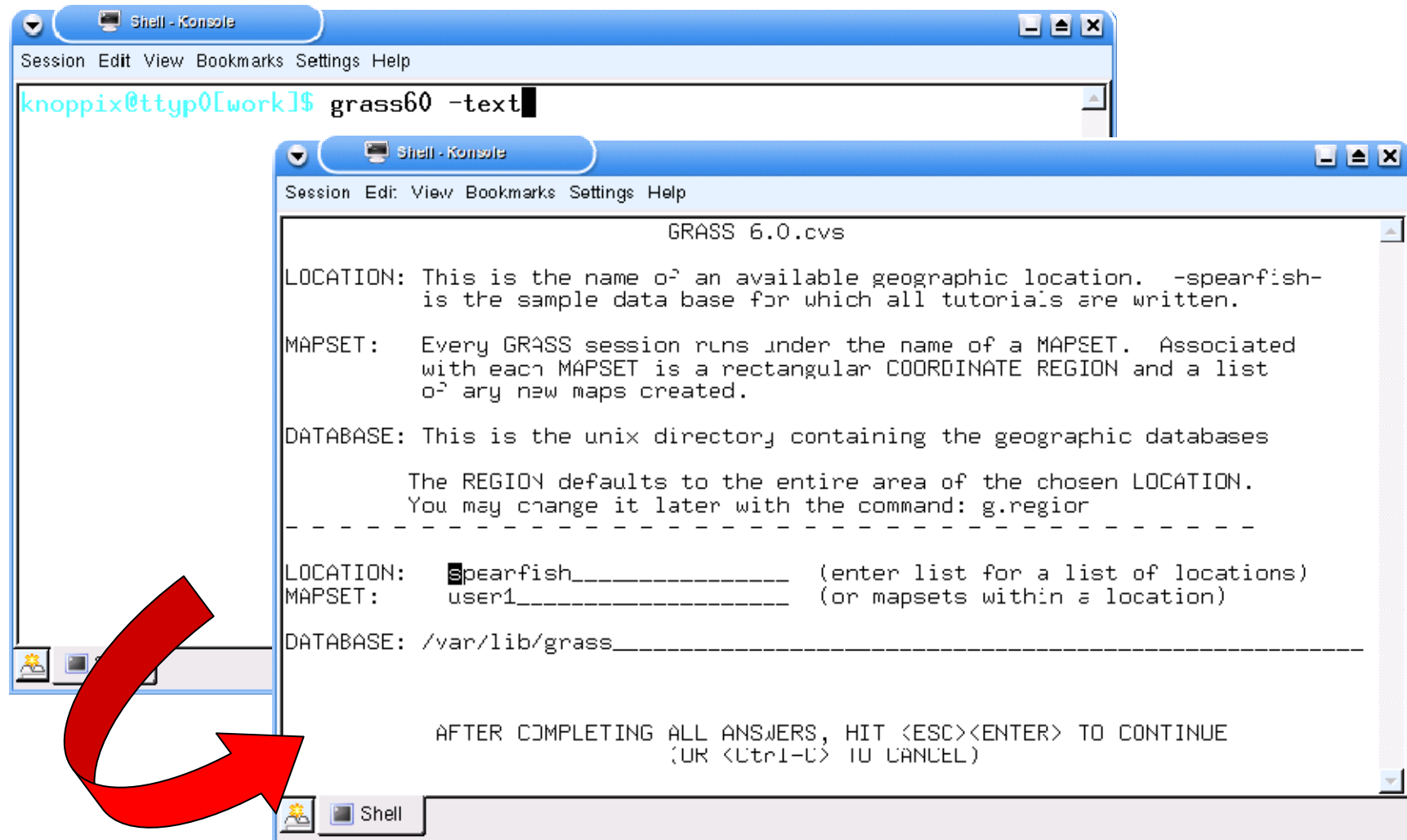

Section 2. SAR Image Processing in Latitude-Longitude Coordinate System

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_____
```

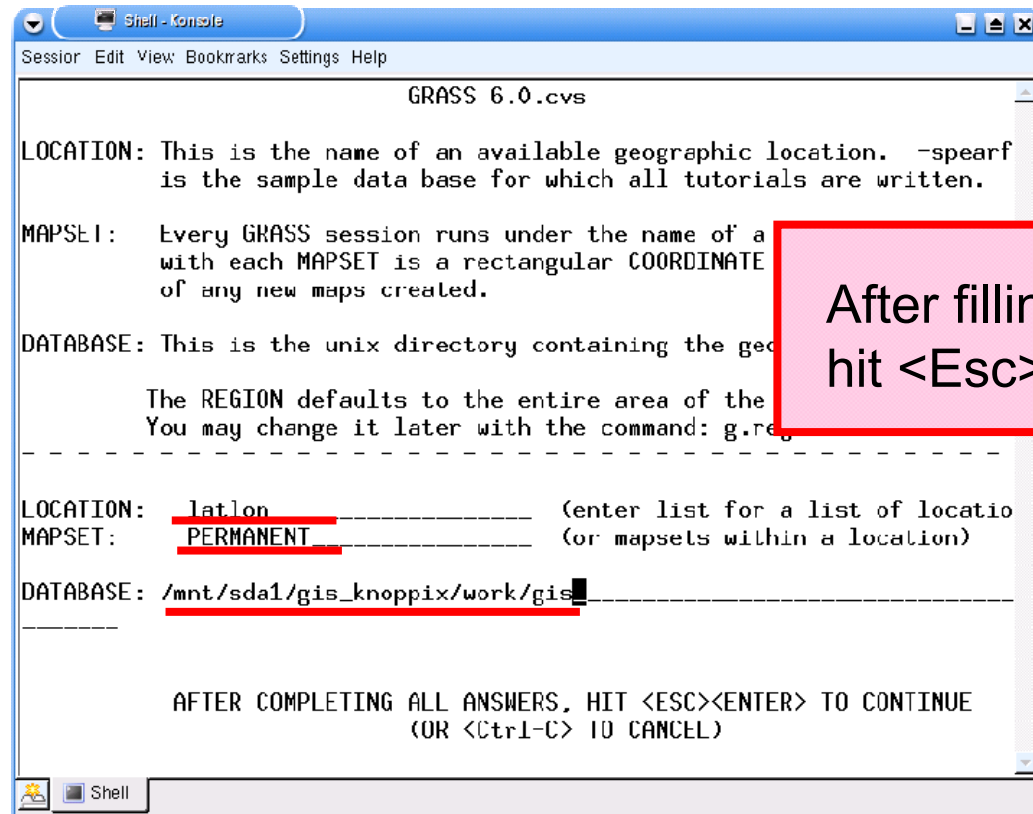
```
-----
AFTER COMPLETING ALL ANSWERS, HIT <ESC><ENTER> TO CONTINUE
(OR <Ctrl-C> TO CANCEL)
```

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 window displays the GRASS 6.0.cvs installation process. It explains the LOCATION, MAPSET, and DATABASE settings. The LOCATION is set to "latlon", the MAPSET to "PERMANENT", and the DATABASE to "/mnt/sda1/gis_knoppix/work/gis". The window also includes instructions on how to continue the installation after completing the answers.

```
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

-----

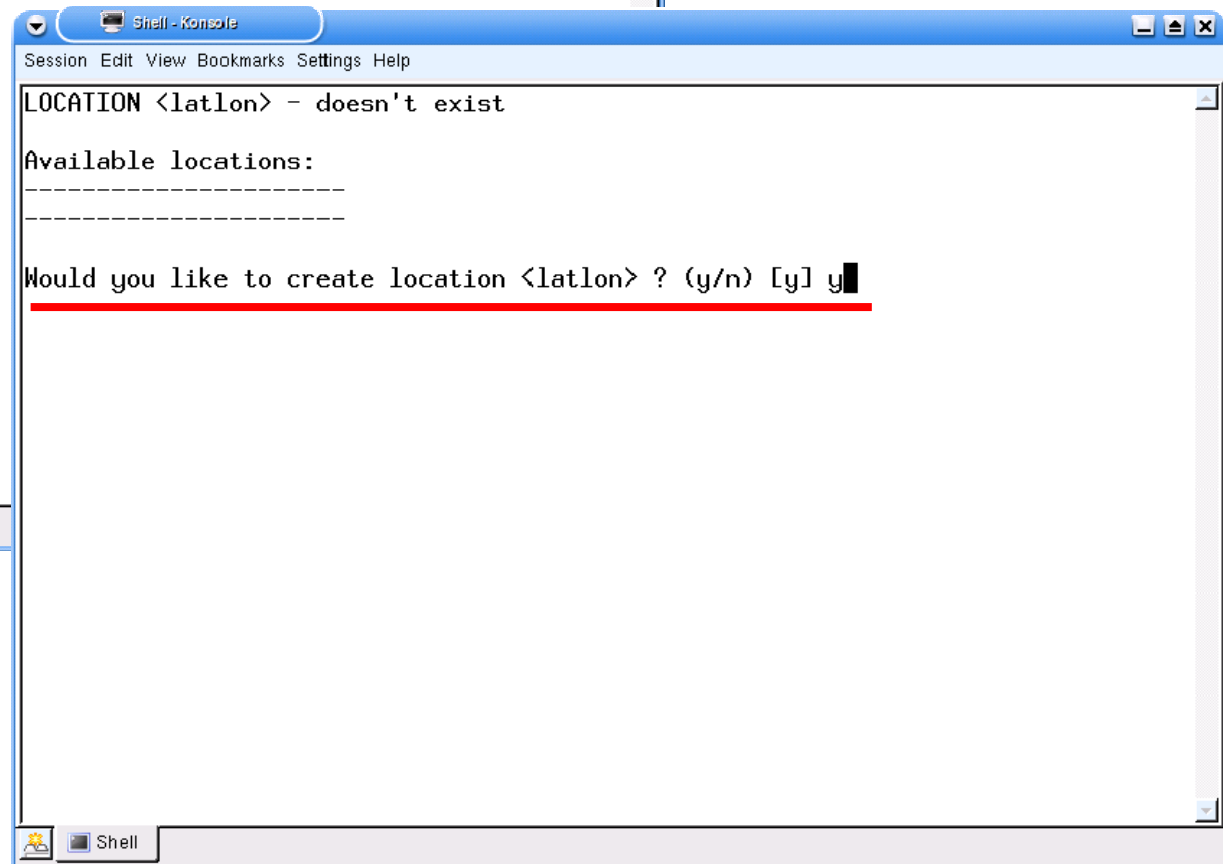
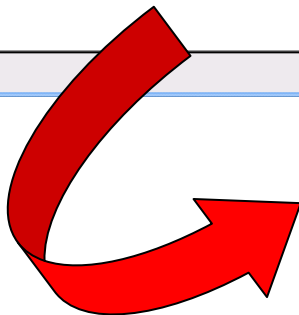
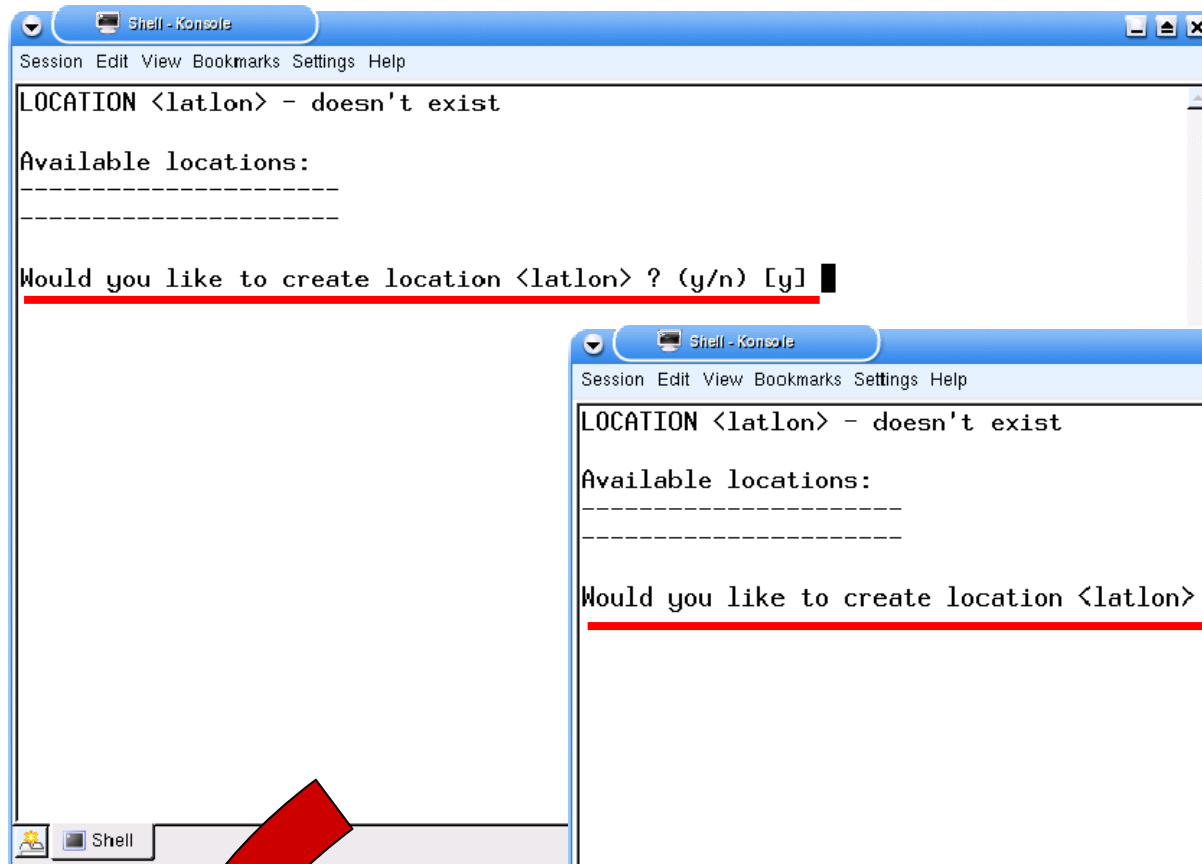
LOCATION:  latlon _____ (enter list for a list of locatio
MAPSET:   PERMANENT _____ (or mapsets within a location)
DATABASE: /mnt/sda1/gis_knoppix/work/gis _____

-----

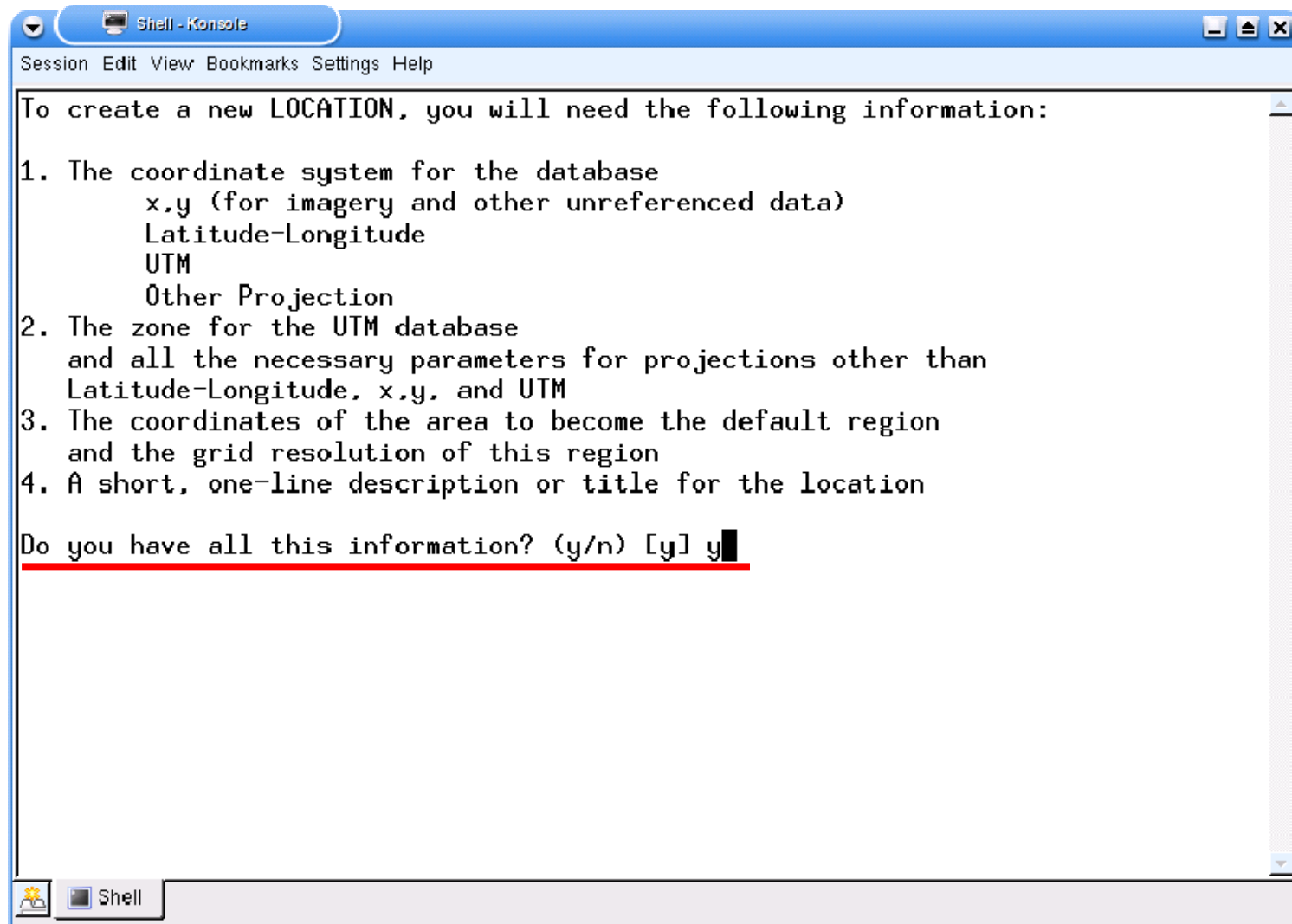
AFTER COMPLETING ALL ANSWERS, HIT <ESC><ENTER> TO CONTINUE
        (OR <Ctrl-C> TO CANCEL)
```

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

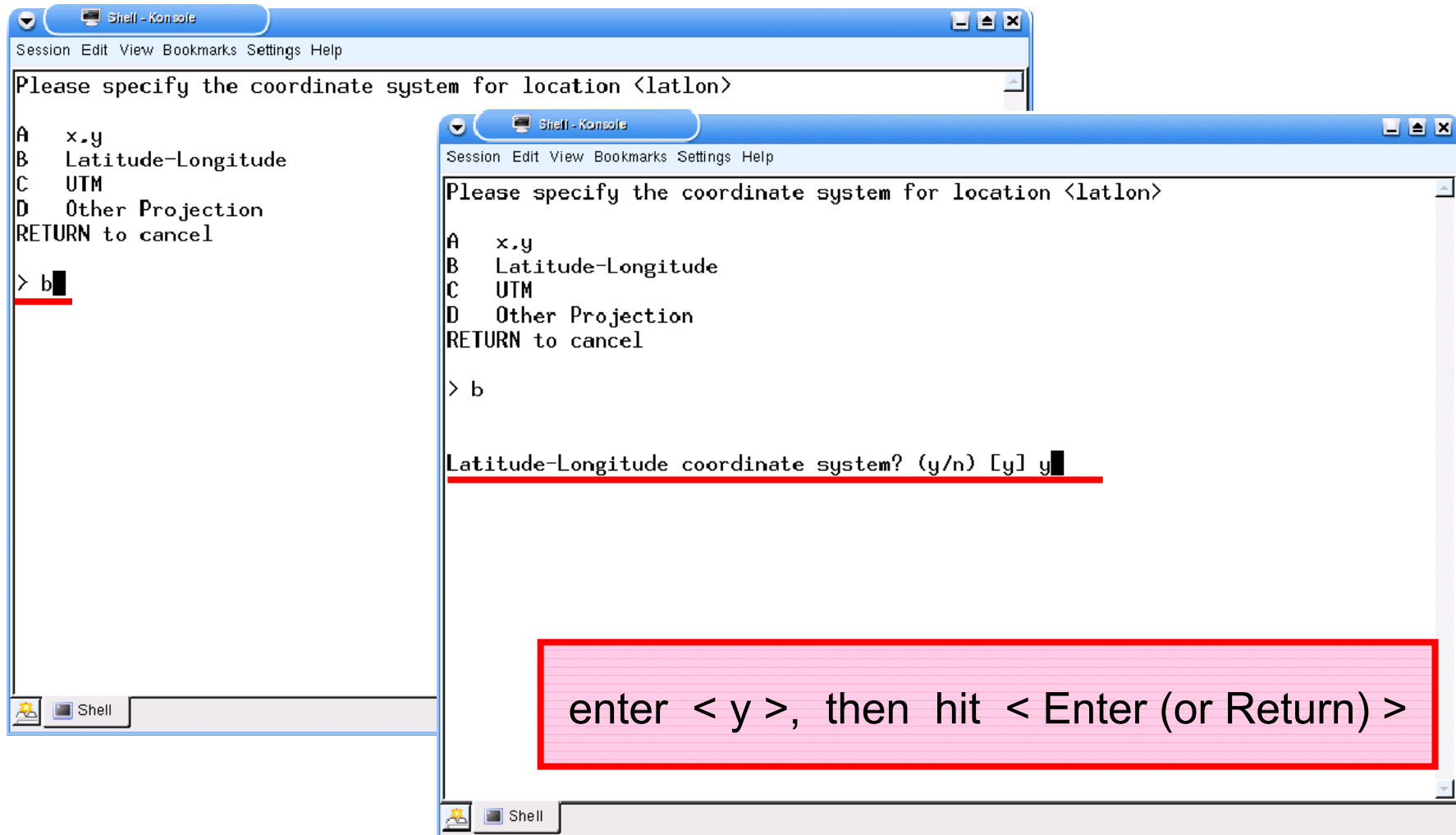
To create a new LOCATION, you will need the following information:

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

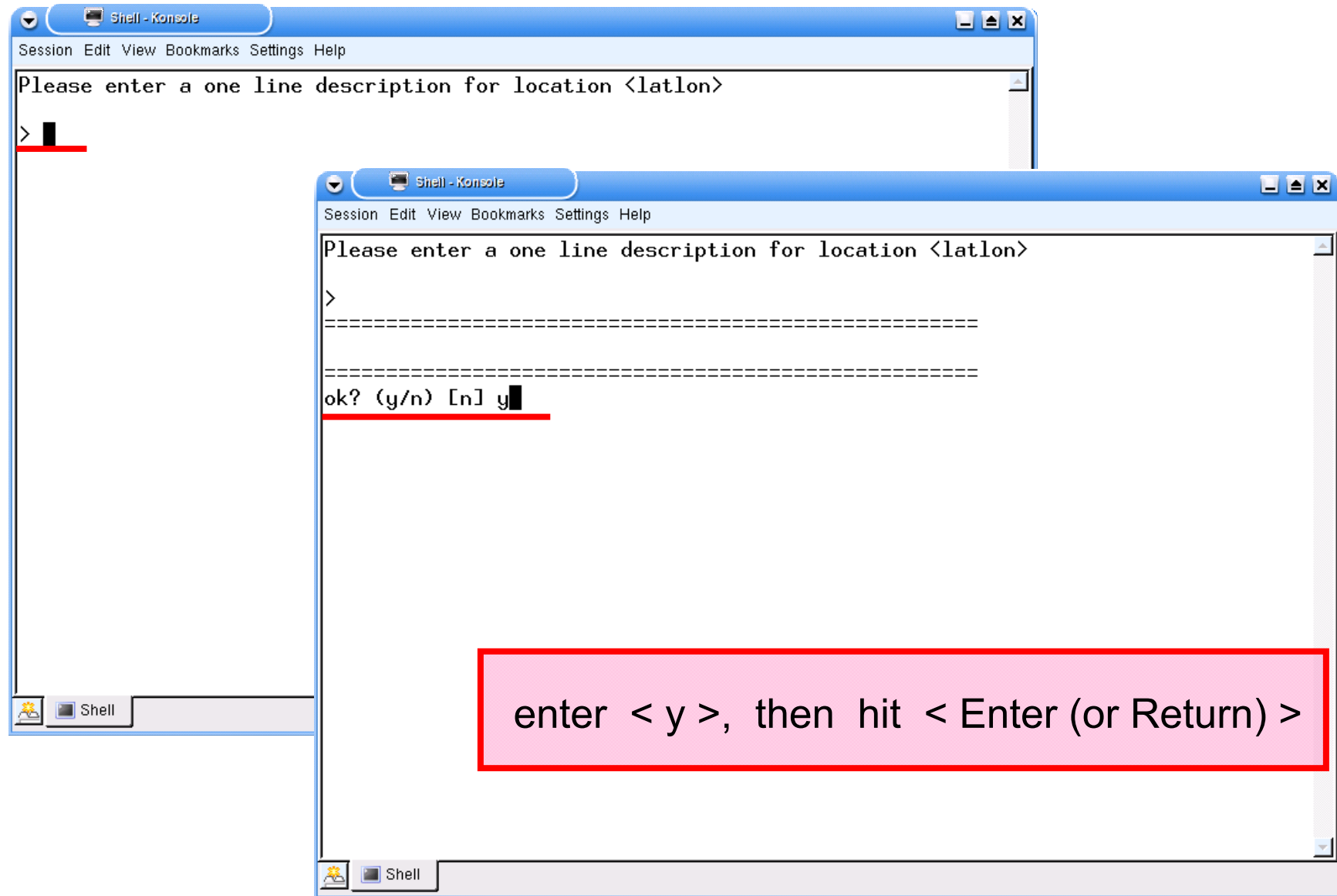
Do you have all this information? (y/n) [y] y
```

2.3 Selection of projection : Latitude-longitude coordinate

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

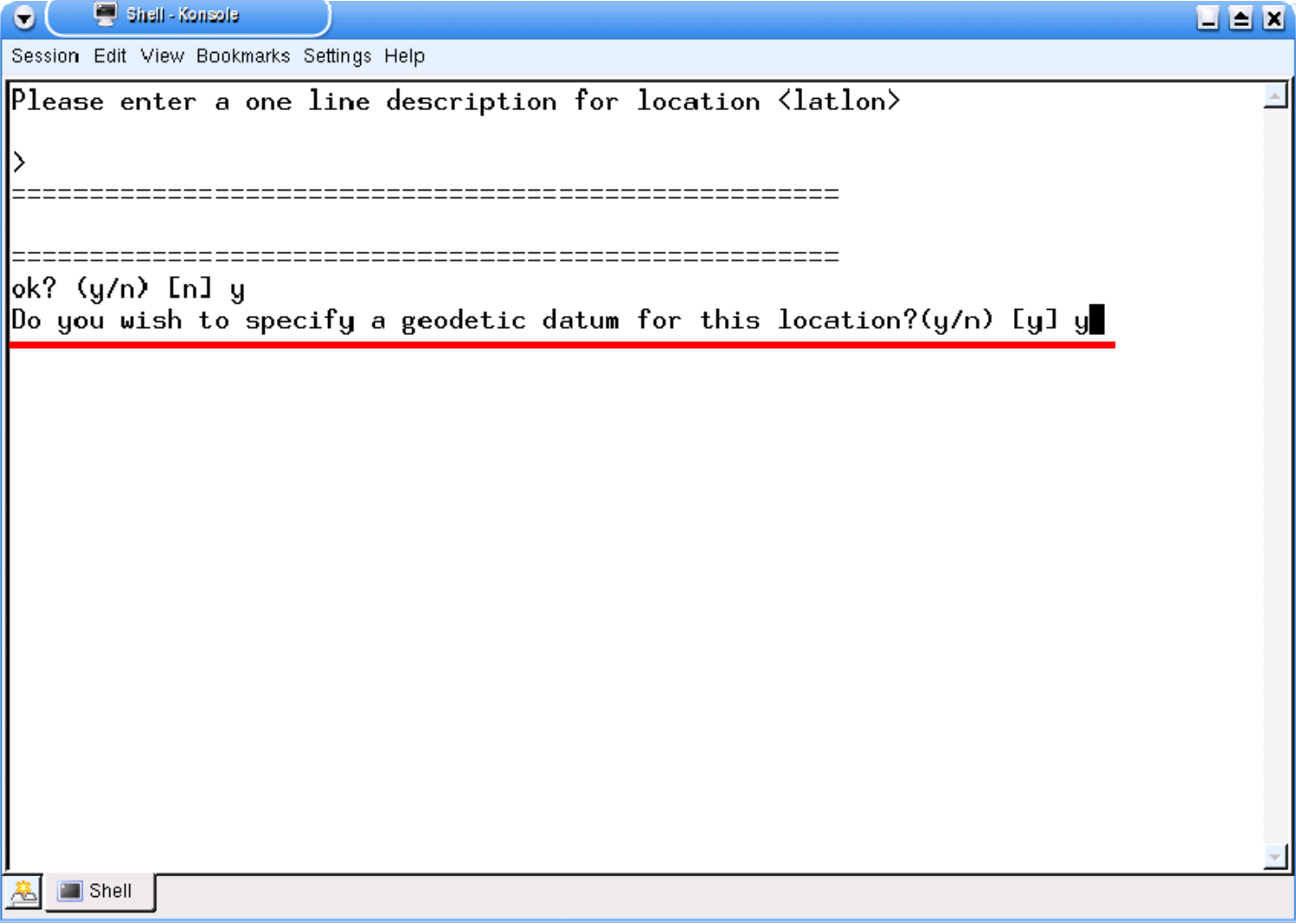


just hit < Enter (or Return) >



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

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

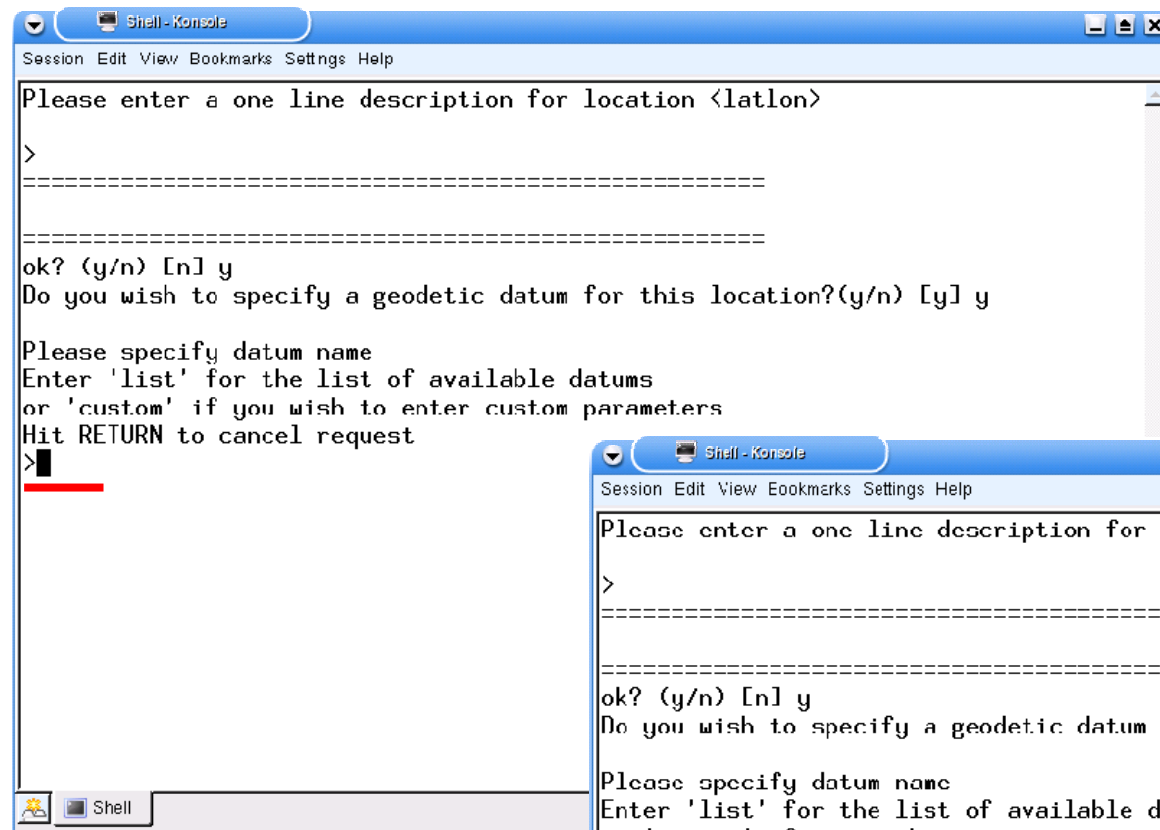


The screenshot shows a window titled "Shell - Konsole" with a menu bar containing "Session", "Edit", "View", "Bookmarks", "Settings", and "Help". The main text area contains the following prompts and input:

```
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
```

The last line of input is underlined in red. At the bottom left of the window, there is a taskbar with a "Shell" icon and label.

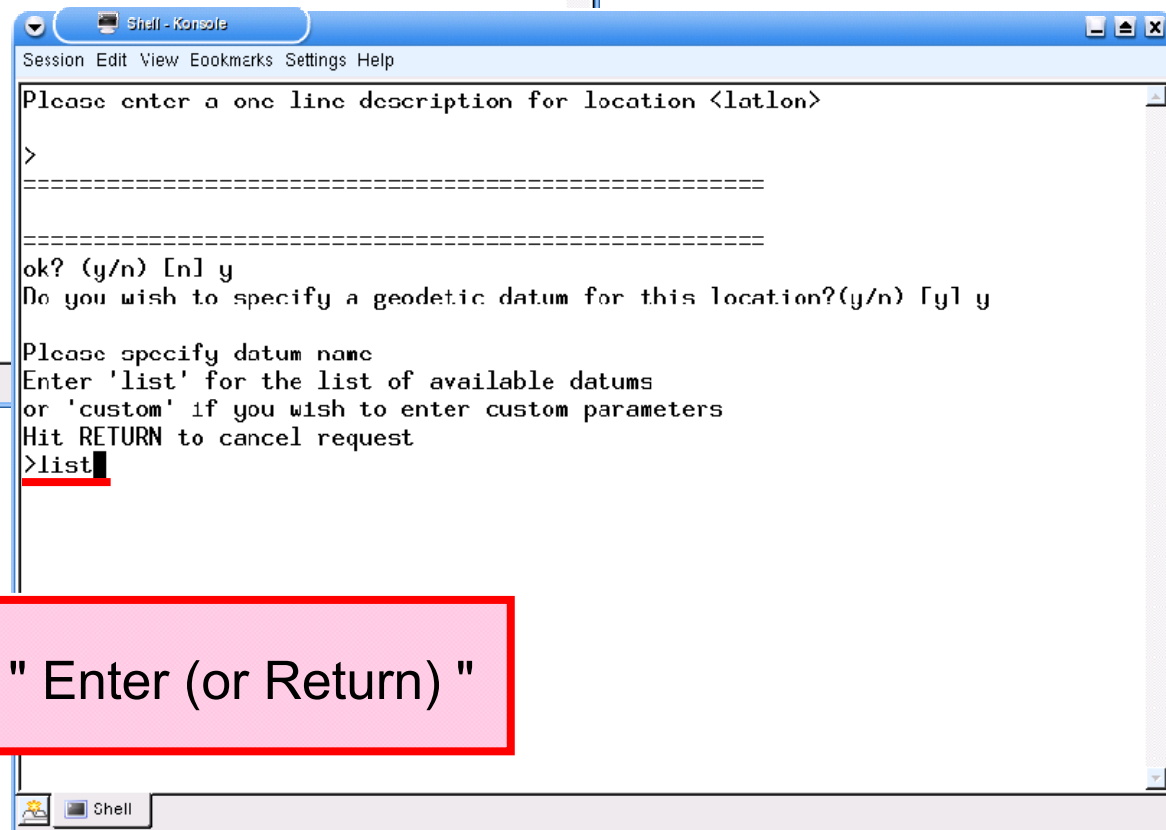
2.4 Selection of datum



```
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
>|
```



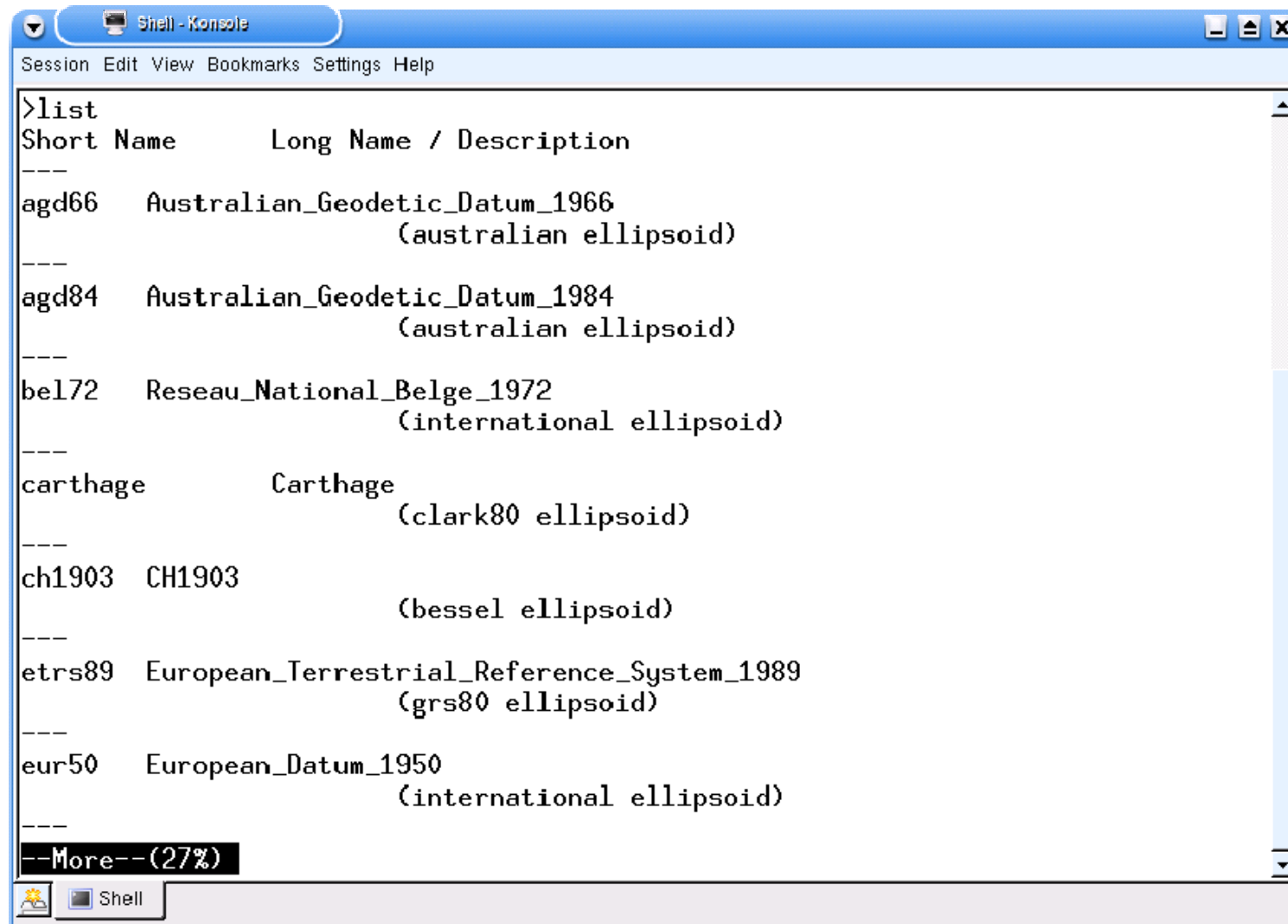
```
Shell - Konsole
Session Edit View Bookmarks Settings Help

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|
```

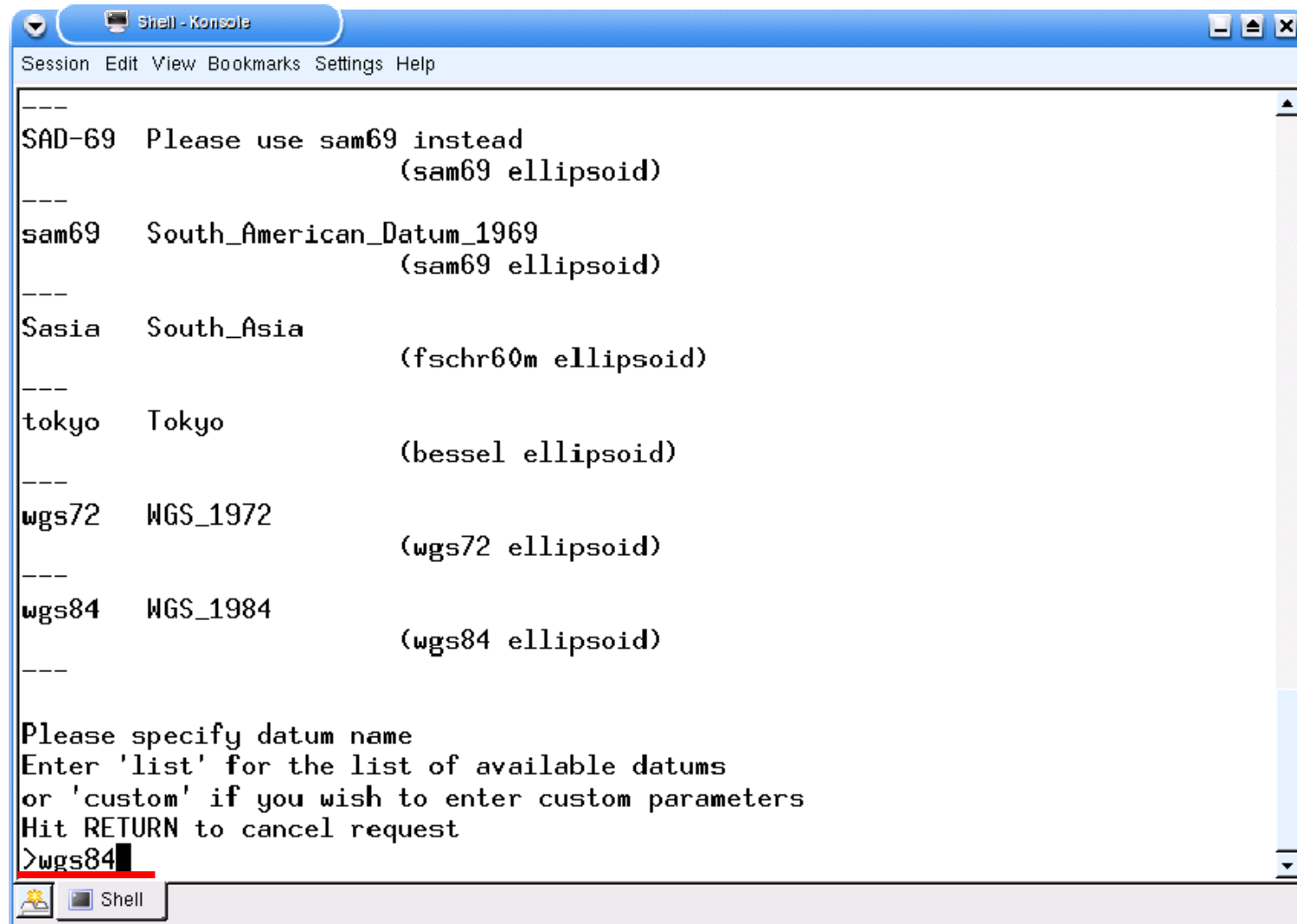
enter "list", then hit "Enter (or Return)"

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)
---
bel172  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%)
```

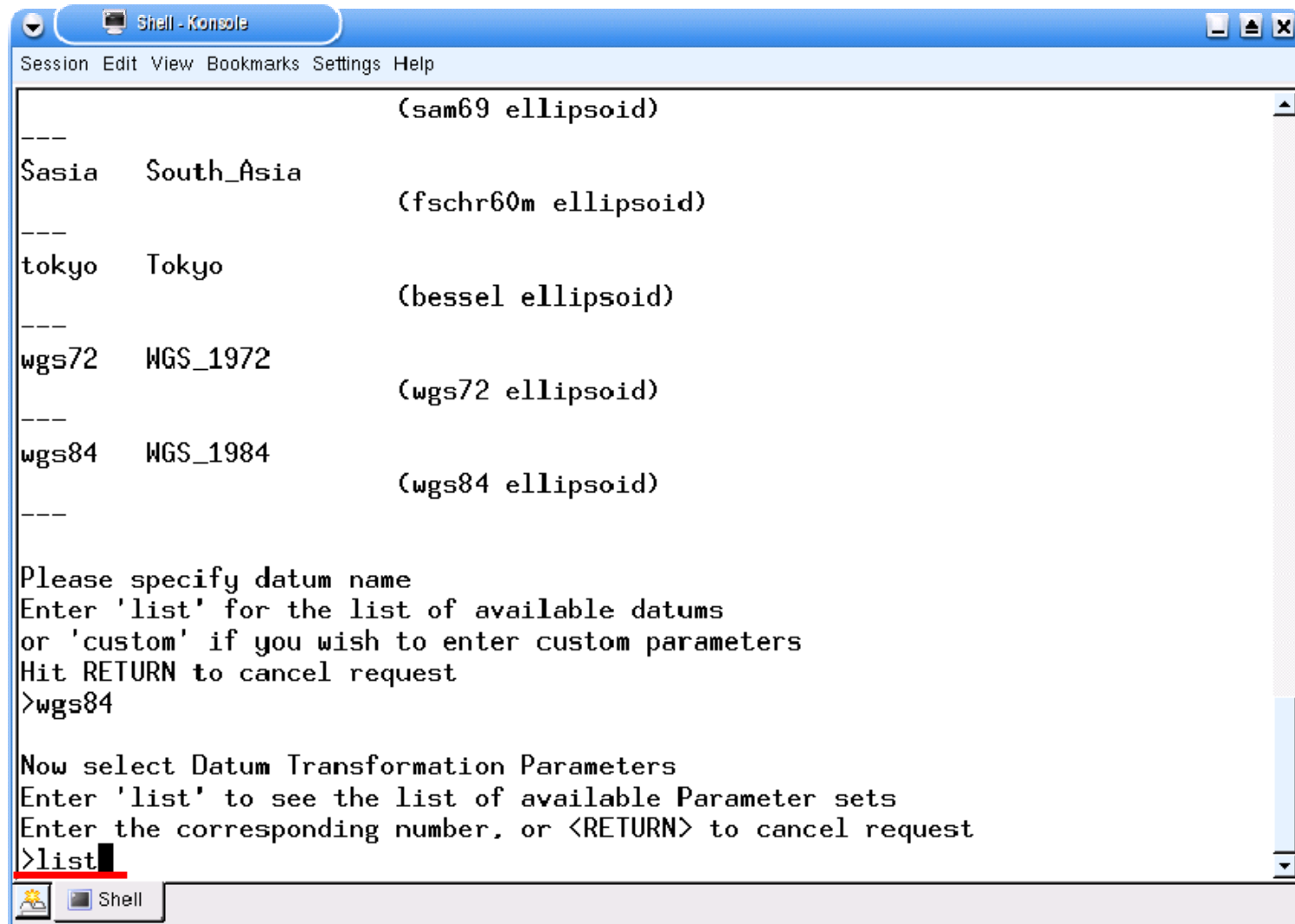
enter "wgs84", then hit "Enter (or Return)"



```
---
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)"



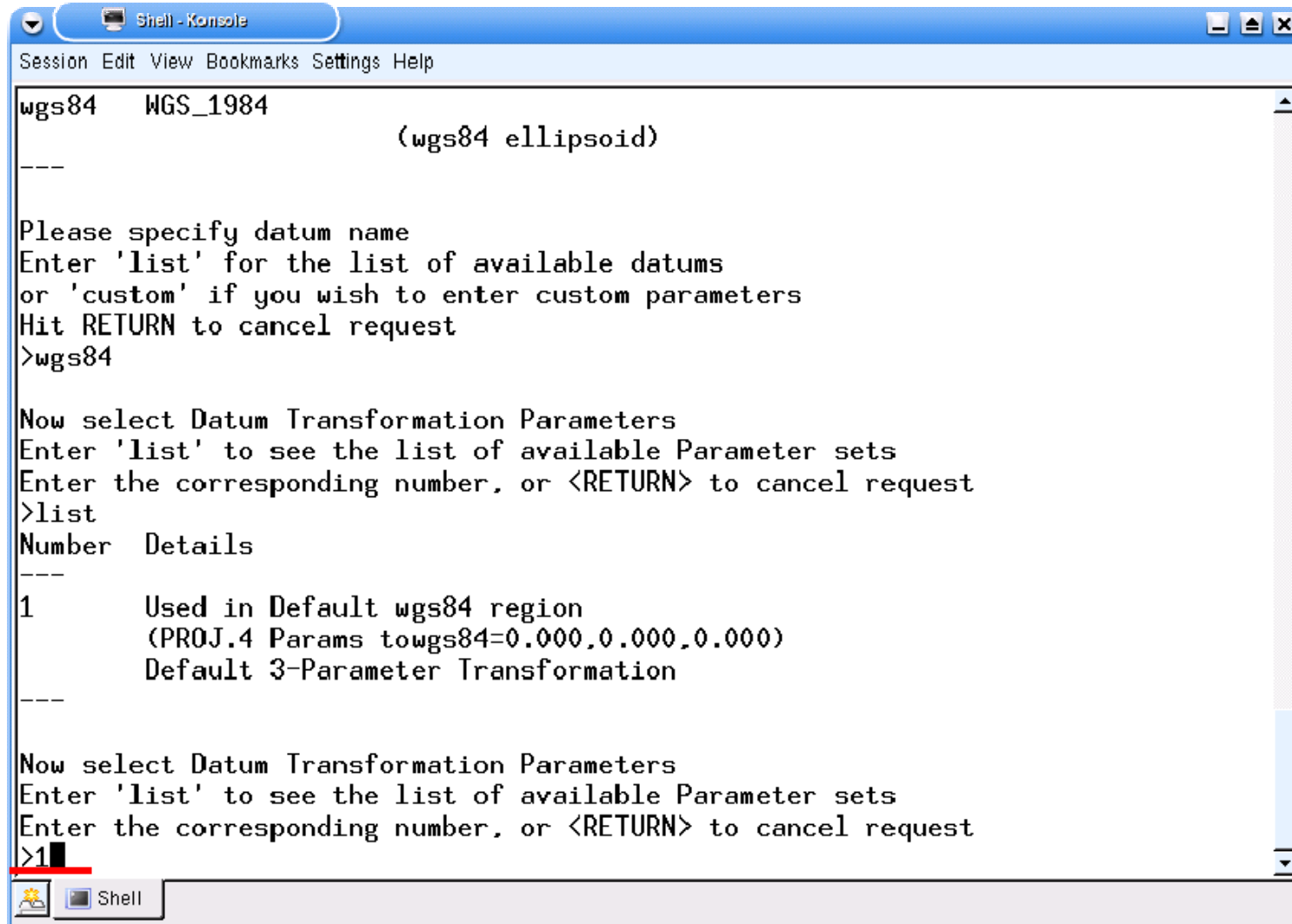
```
Shell - Konsole
Session Edit View Bookmarks Settings Help

----
(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

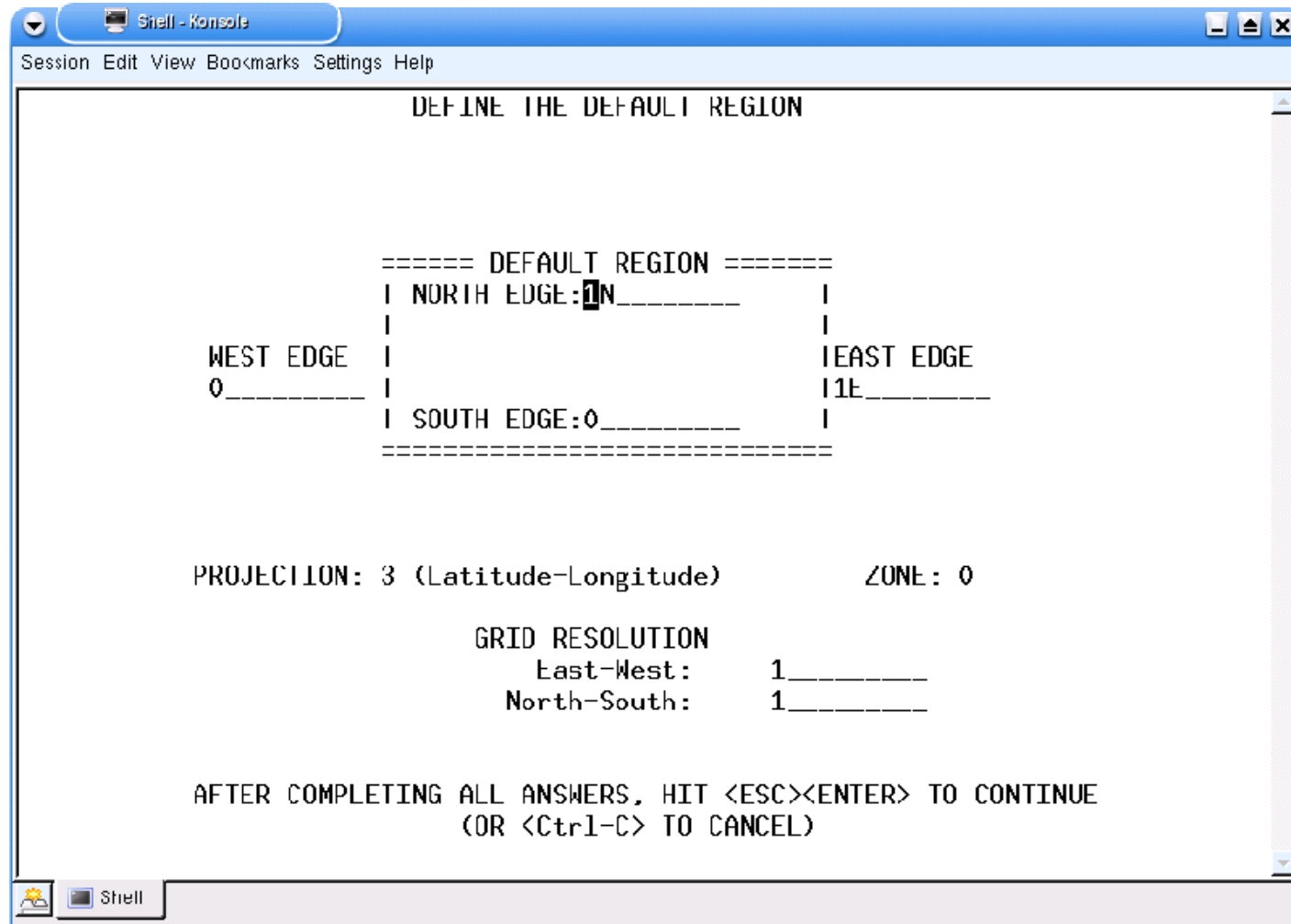
wgs84  WGS_1984
                                     (wgs84 ellipsoid)
---

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**

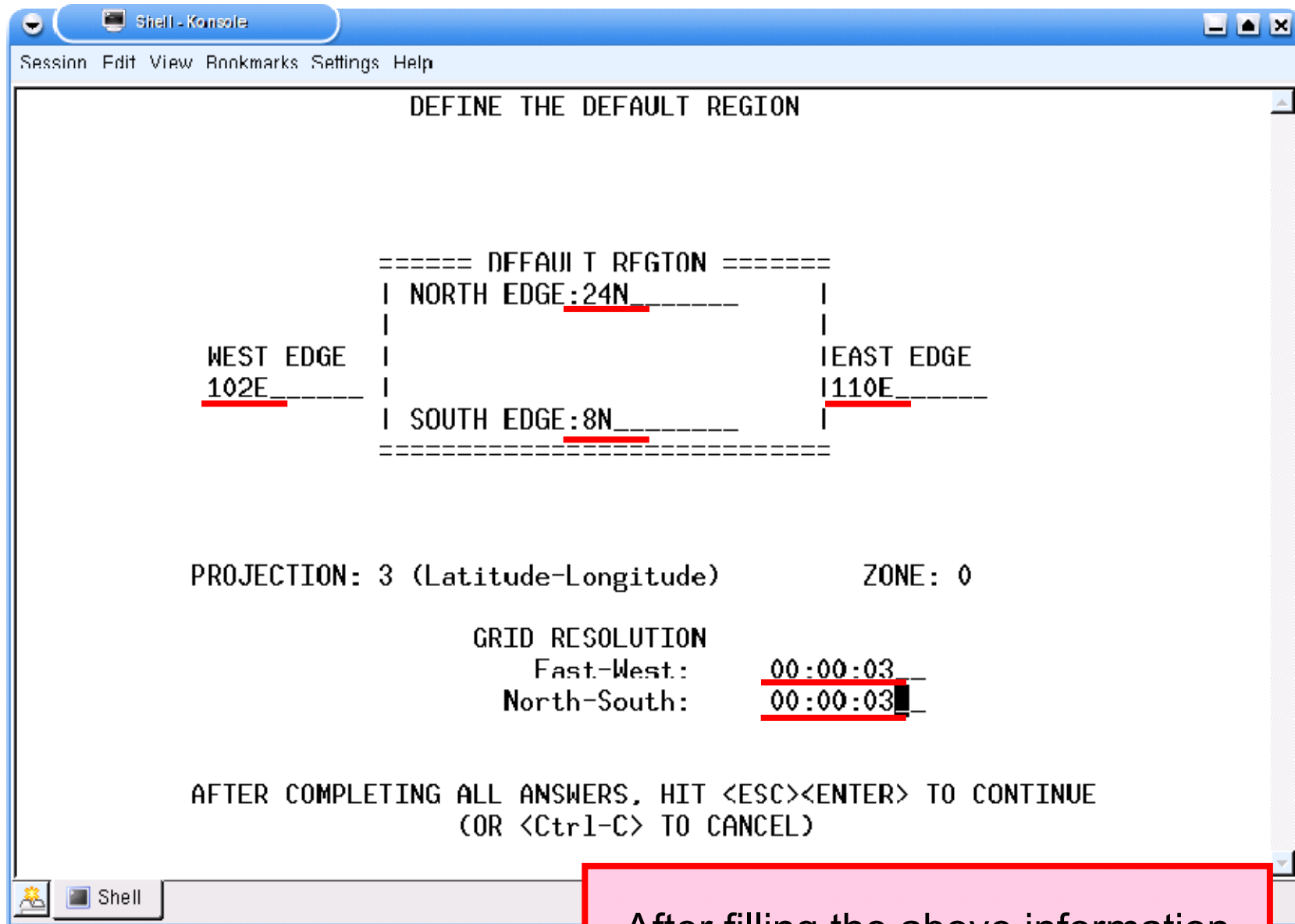
GRID RESOLUTION East-West : **00:00:03**

North-South : **00:00:03**

<= 1deg. : 1200pixels
(100m resolution)

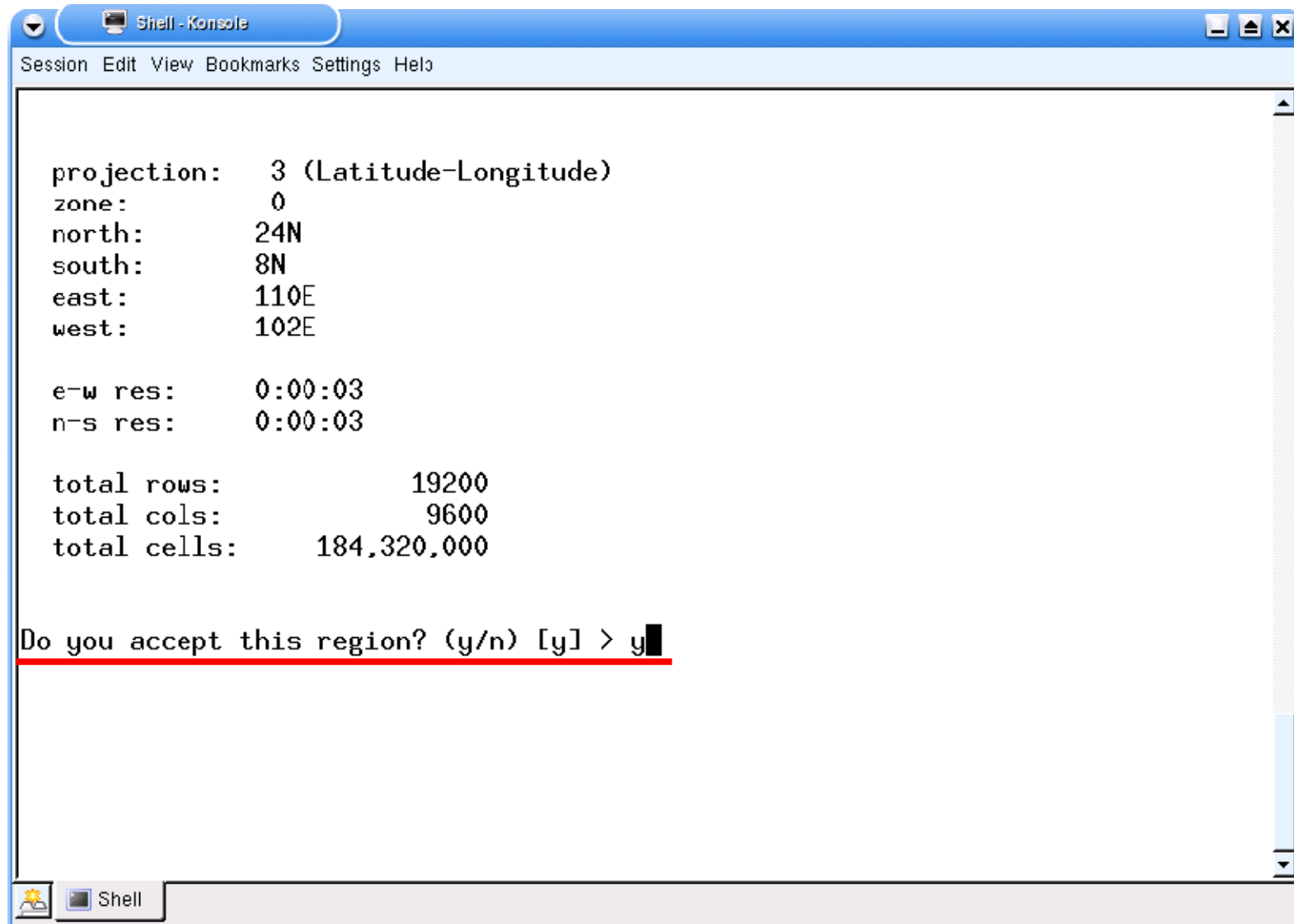
For latitude-longitude coordinate system, you have to define the GRID RESOLUTION in “degree:minutes:seconds (DMS)”.

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



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

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



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

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

e-w res: 0:00:03
n-s res: 0:00:03

total rows: 19200
total cols: 9600
total cells: 184,320,000
```

Below this information, a red line underlines the prompt: "Do you accept this region? (y/n) [y] > y". The cursor is positioned at the end of the input "y".

At the bottom of the window, there is a taskbar with a "Shell" icon and label.

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

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

e-w res:    0:00:03
n-s res:    0:00:03

total rows:          19200
total cols:          9600
total cells:    184,320,000

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

Hit RETURN -->
```

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

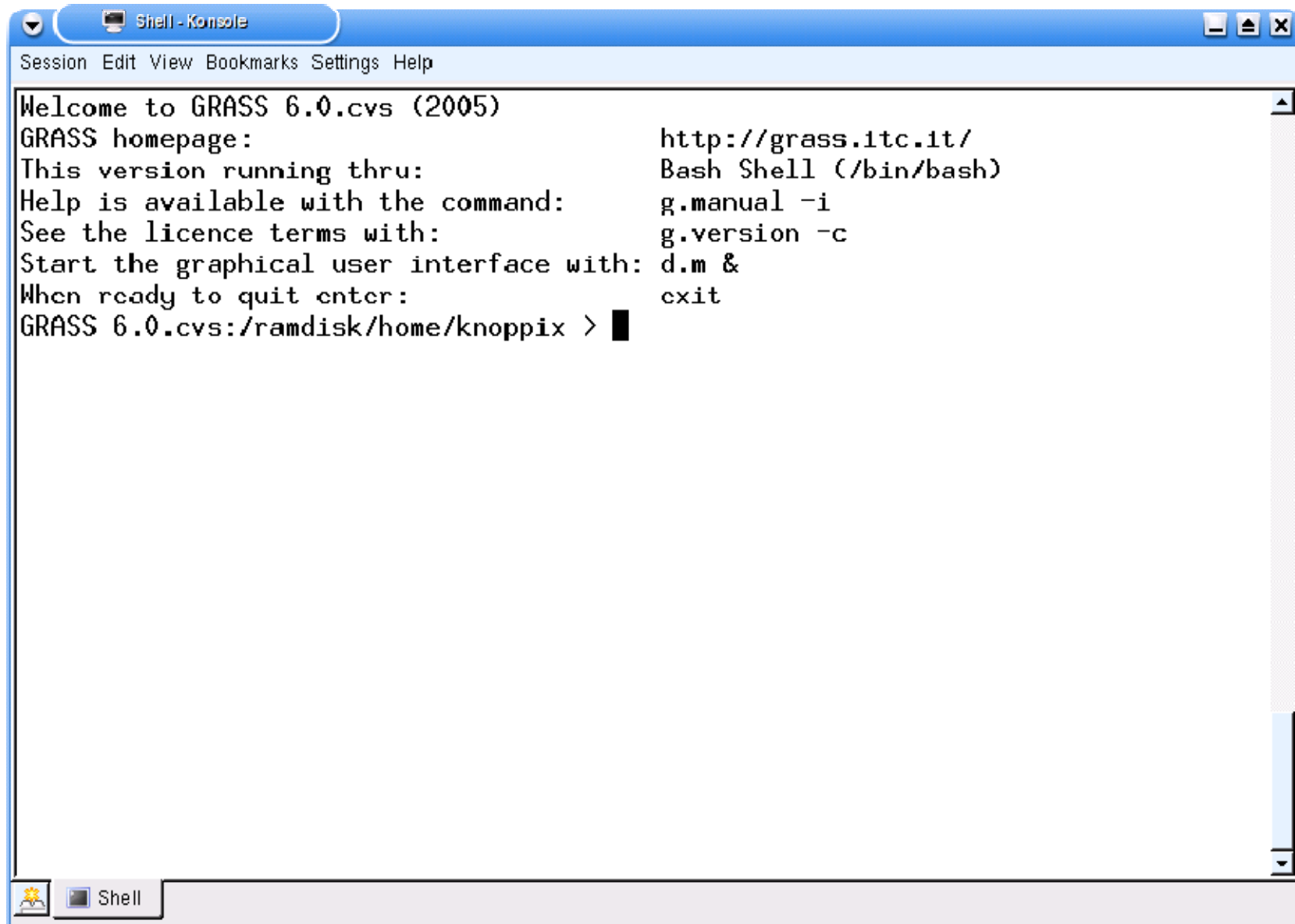
        The REGION defaults to the entire area of the chosen LOCATION.
        You may change it later with the command: g.region
-----

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. SAR Image Processing in Latitude-Longitude Coordinate System

3. Import images

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.

Data list

- PALSAR 100m mosaic data in Vietnam
- JERS-1 SAR 100m mosaic data in Vietnam
Orthorectified
- SRTM-DEM by CGIAR

3.1 Import SAR 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 = 4 (bytes)

D = 16 (north)

E = 11 (south)

F = 110 (east)

G = 107 (west)

H = 6000 (rows)

I = 3600 (cols)

: space

For Vietnam Central Highland Area

$\leq (24 - 8) * 1200 = 19200$ 1deg. : 1200pixels (100m resolution)

$\leq (110 - 102) * 1200 = 9600$ 1deg. : 1200pixels (100m resolution)

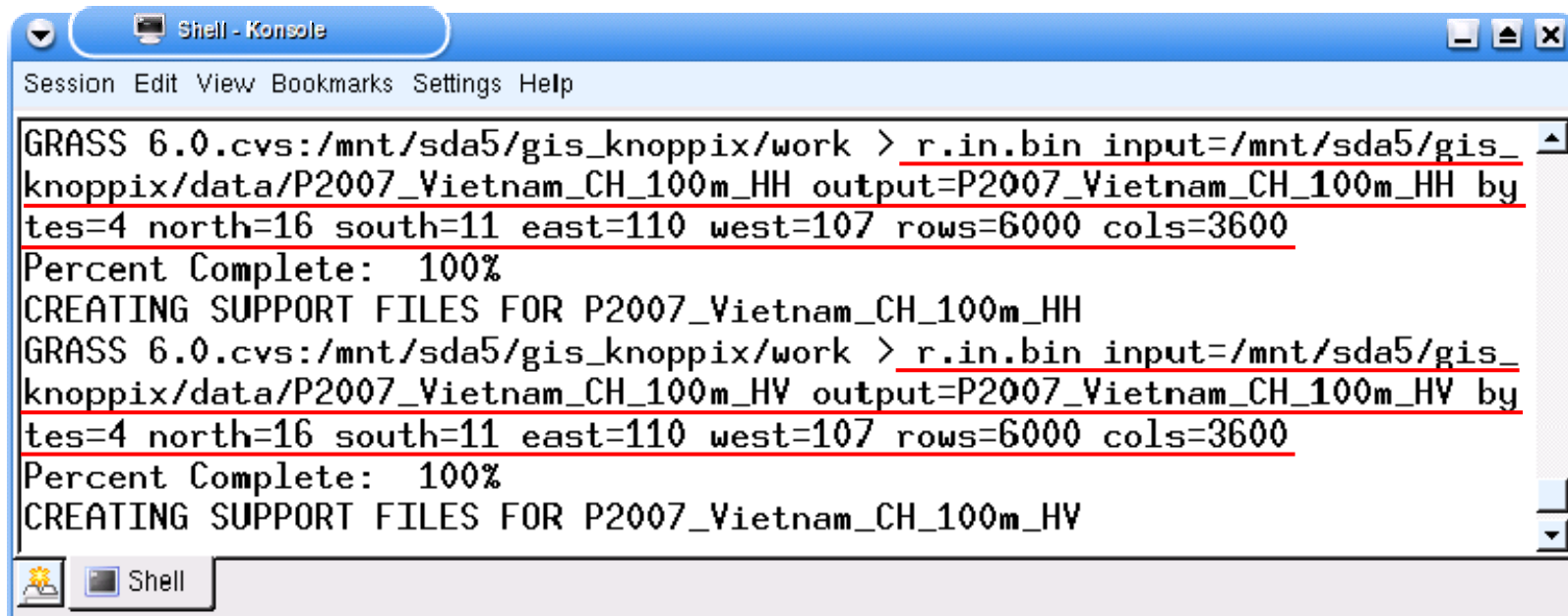
r.in.bin allows a user to create a (binary) GRASS raster map layer from a variety of binary raster data formats.

3.1.1 Import PALSAR 100m mosaic image

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

```
> r.in.bin input=/mnt/sda1/gis_knoppix/data/P2007_Vietnam_CH_100m_HH  
output=P2007_Vietnam_100m_CH_HH bytes=4 north=16 south=11  
east=110 west=107 rows=6000 cols=3600 : space
```

```
> r.in.bin input=/mnt/sda1/gis_knoppix/data/P2007_Vietnam_CH_100m_HV  
output=P2007_Vietnam_CH_100m_HV bytes=4 north=16 south=11  
east=110 west=107 rows=6000 cols=3600 : space
```



The screenshot shows a terminal window titled "Shell - Konsole" with a menu bar (Session, Edit, View, Bookmarks, Settings, Help). The terminal displays two commands and their outputs. The first command is `r.in.bin input=/mnt/sda5/gis_knoppix/data/P2007_Vietnam_CH_100m_HH output=P2007_Vietnam_CH_100m_HH bytes=4 north=16 south=11 east=110 west=107 rows=6000 cols=3600`, which completes at 100% and creates support files. The second command is `r.in.bin input=/mnt/sda5/gis_knoppix/data/P2007_Vietnam_CH_100m_HV output=P2007_Vietnam_CH_100m_HV bytes=4 north=16 south=11 east=110 west=107 rows=6000 cols=3600`, which also completes at 100% and creates support files. The terminal has a scrollbar on the right and a taskbar at the bottom with a "Shell" icon.

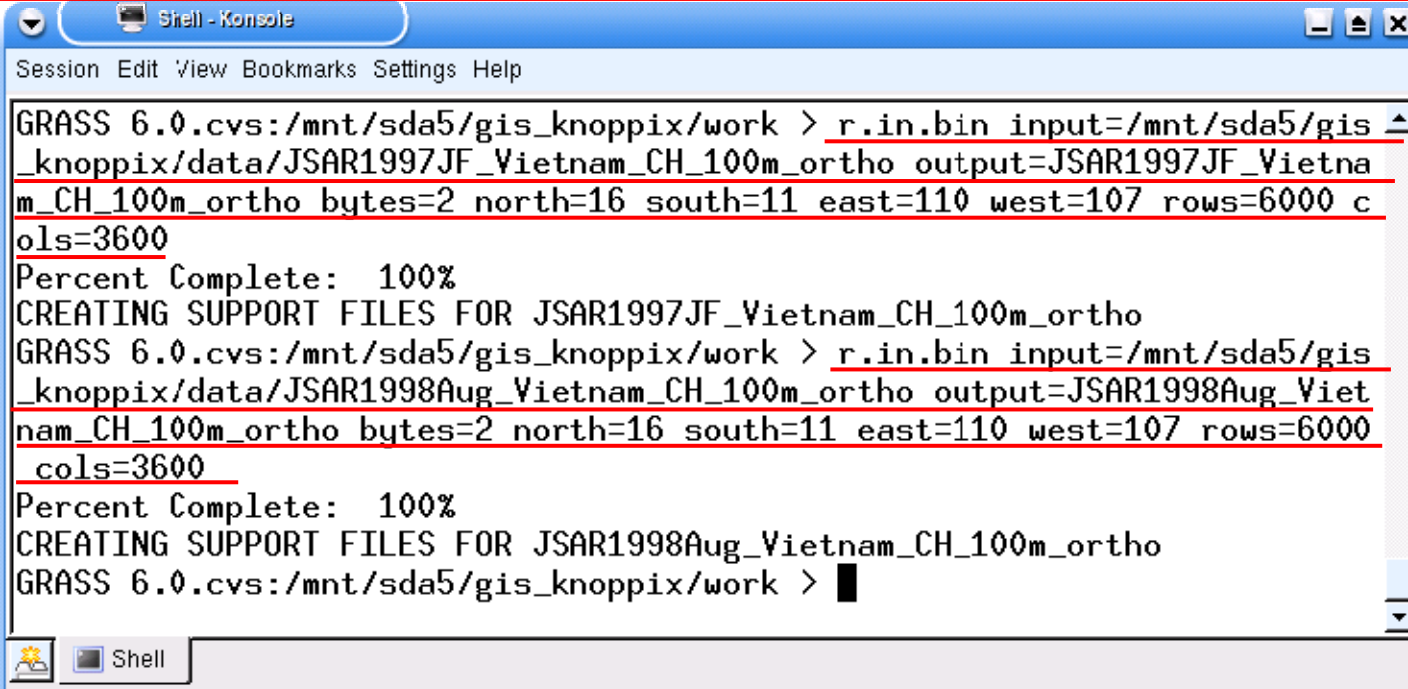
```
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.in.bin input=/mnt/sda5/gis_
knoppix/data/P2007_Vietnam_CH_100m_HH output=P2007_Vietnam_CH_100m_HH by
tes=4 north=16 south=11 east=110 west=107 rows=6000 cols=3600
Percent Complete: 100%
CREATING SUPPORT FILES FOR P2007_Vietnam_CH_100m_HH
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.in.bin input=/mnt/sda5/gis_
knoppix/data/P2007_Vietnam_CH_100m_HV output=P2007_Vietnam_CH_100m_HV by
tes=4 north=16 south=11 east=110 west=107 rows=6000 cols=3600
Percent Complete: 100%
CREATING SUPPORT FILES FOR P2007_Vietnam_CH_100m_HV
```

3.1.2 Import JERS-1 SAR 100m mosaic image

If you have stored JERS-1 SAR images in “data” directory:

```
> r.in.bin input=/mnt/sda1/gis_knoppix/data/J1997JF_Vietnam_CH_100m_ortho output=J1997JF_Vietnam_CH_100m_ortho bytes=2 north=16 south=11 east=110 west=107 rows=6000 cols=3600 : space
```

```
> r.in.bin input=/mnt/sda1/gis_knoppix/data/J1998Aug_Vietnam_CH_100m_ortho output=J1998Aug_Vietnam_CH_100m_ortho bytes=2 north=16 south=11 east=110 west=107 rows=6000 cols=3600 : space
```

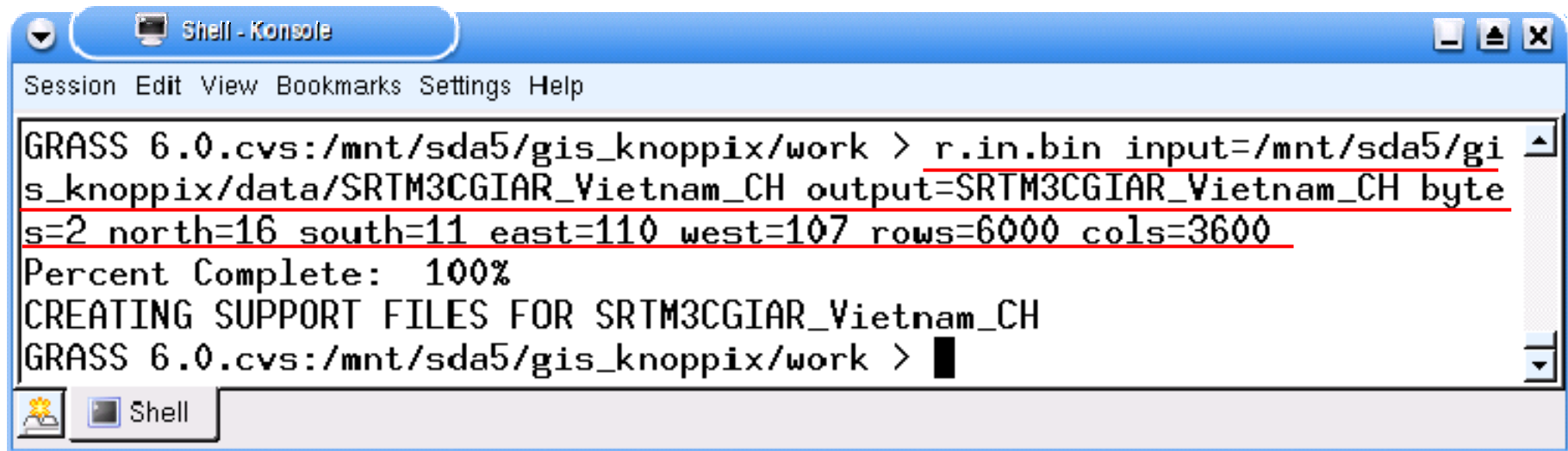


```
Shell - Konsole
Session Edit View Bookmarks Settings Help
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.in.bin input=/mnt/sda5/gis_knoppix/data/JSAR1997JF_Vietnam_CH_100m_ortho output=JSAR1997JF_Vietnam_CH_100m_ortho bytes=2 north=16 south=11 east=110 west=107 rows=6000 cols=3600
Percent Complete: 100%
CREATING SUPPORT FILES FOR JSAR1997JF_Vietnam_CH_100m_ortho
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.in.bin input=/mnt/sda5/gis_knoppix/data/JSAR1998Aug_Vietnam_CH_100m_ortho output=JSAR1998Aug_Vietnam_CH_100m_ortho bytes=2 north=16 south=11 east=110 west=107 rows=6000 cols=3600
Percent Complete: 100%
CREATING SUPPORT FILES FOR JSAR1998Aug_Vietnam_CH_100m_ortho
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > █
```

3.2 Import SRTM-DEM image

If you have stored SRTM-DEM images in “data” directory:

```
> r.in.bin input=/mnt/sda1/gis_knoppix/data/SRTM3CGIAR_Vietnam_CH  
output=SRTM3CGIAR_Vietnam bytes=2 north=16 south=11 east=110  
west=107 rows=6000 cols=3600
```



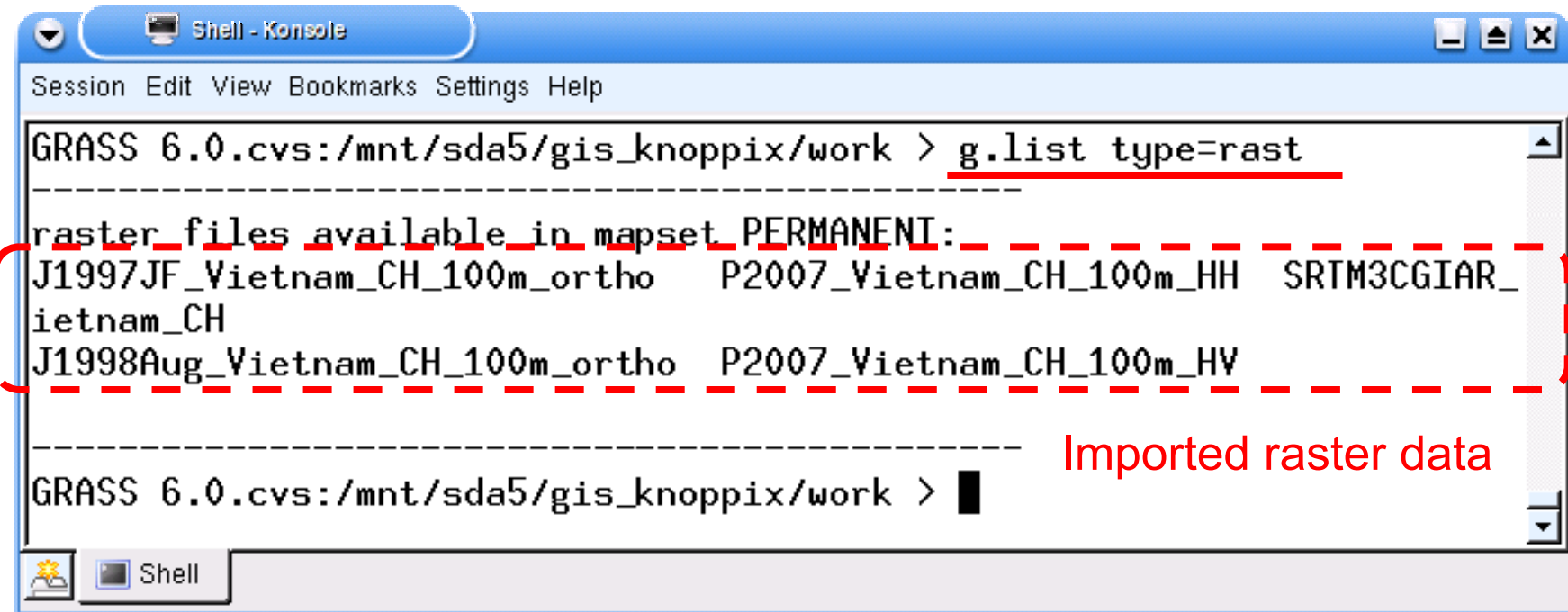
The screenshot shows a terminal window titled "Shell - Konsole". The terminal displays the command `r.in.bin input=/mnt/sda5/gis_knoppix/data/SRTM3CGIAR_Vietnam_CH output=SRTM3CGIAR_Vietnam_CH bytes=2 north=16 south=11 east=110 west=107 rows=6000 cols=3600` being executed in a GRASS 6.0.cvs environment. The output shows "Percent Complete: 100%" and "CREATING SUPPORT FILES FOR SRTM3CGIAR_Vietnam_CH". The terminal window has a menu bar with "Session", "Edit", "View", "Bookmarks", "Settings", and "Help". The status bar at the bottom shows "Shell" and a small icon.

```
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.in.bin input=/mnt/sda5/gi  
s_knoppix/data/SRTM3CGIAR_Vietnam_CH output=SRTM3CGIAR_Vietnam_CH byte  
s=2 north=16 south=11 east=110 west=107 rows=6000 cols=3600  
Percent Complete: 100%  
CREATING SUPPORT FILES FOR SRTM3CGIAR_Vietnam_CH  
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > █
```

3.3 List imported raster data

```
> g.list type = rast
```

␣ : space



The screenshot shows a terminal window titled "Shell - Konsole" with a menu bar (Session, Edit, View, Bookmarks, Settings, Help). The command prompt is "GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work >". The command g.list type=rast has been entered and executed. The output is "raster_files available in mapset PERMANENT:" followed by a list of files: "J1997JF_Vietnam_CH_100m_ortho", "P2007_Vietnam_CH_100m_HH", "SRTM3CGIAR_ietnam_CH", "J1998Aug_Vietnam_CH_100m_ortho", and "P2007_Vietnam_CH_100m_HV". A red dashed box encloses the file names. A red arrow points to the first file name. The text "Imported raster data" is written in red to the right of the output. The prompt "GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work >" is followed by a black cursor.

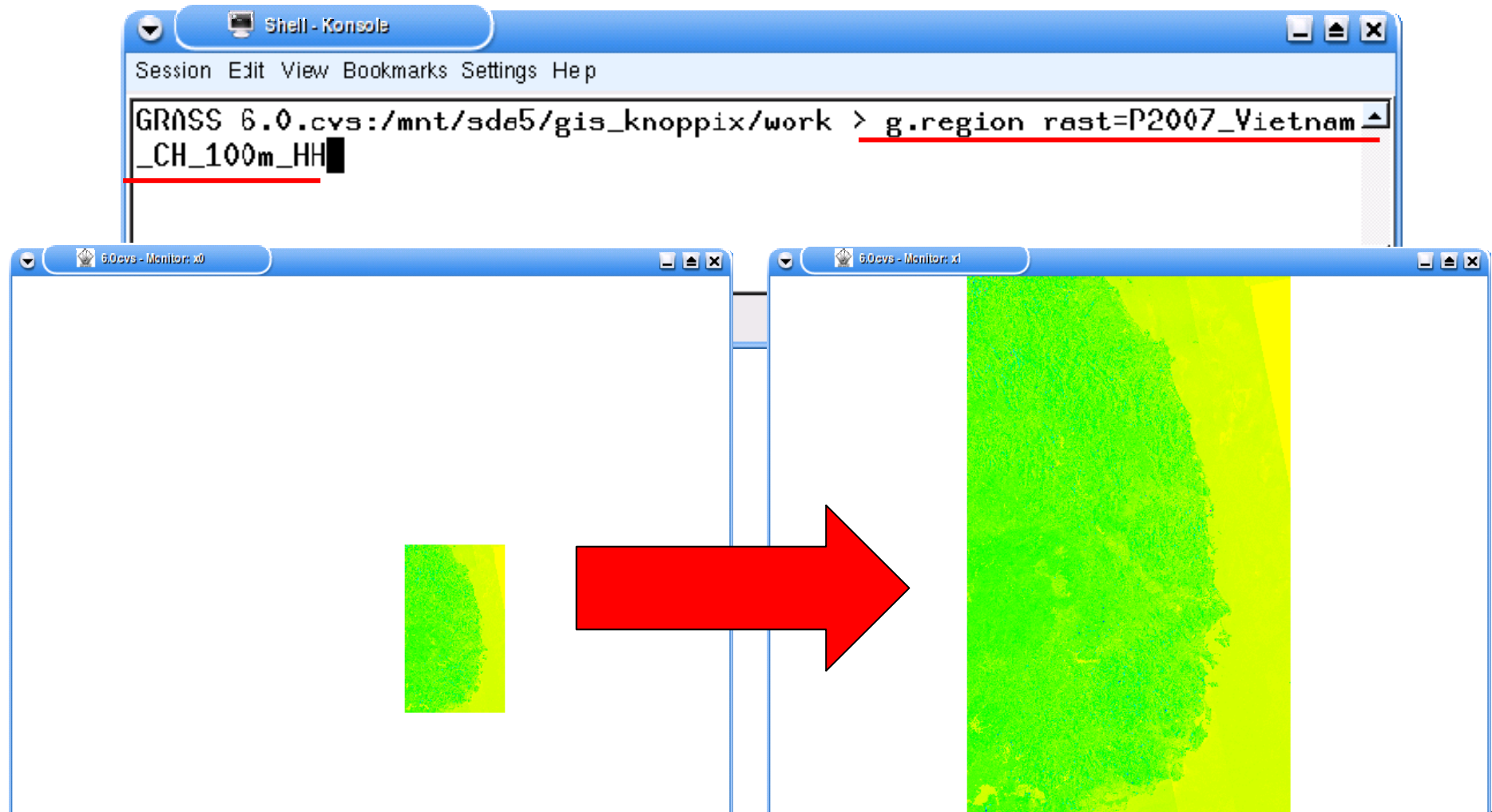
```
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > g.list type=rast
-----
raster_files available in mapset PERMANENT:
J1997JF_Vietnam_CH_100m_ortho  P2007_Vietnam_CH_100m_HH  SRTM3CGIAR_
ietnam_CH
J1998Aug_Vietnam_CH_100m_ortho  P2007_Vietnam_CH_100m_HV
-----
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > █
```

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

3.4 Set region to match imported raster map

> g.region rast = (name of imported raster map)

: space



Section 2. SAR Image Processing in Latitude-Longitude Coordinate System

4. Display images

Step 0. Import raster data

⇒ See “3. Import 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
```

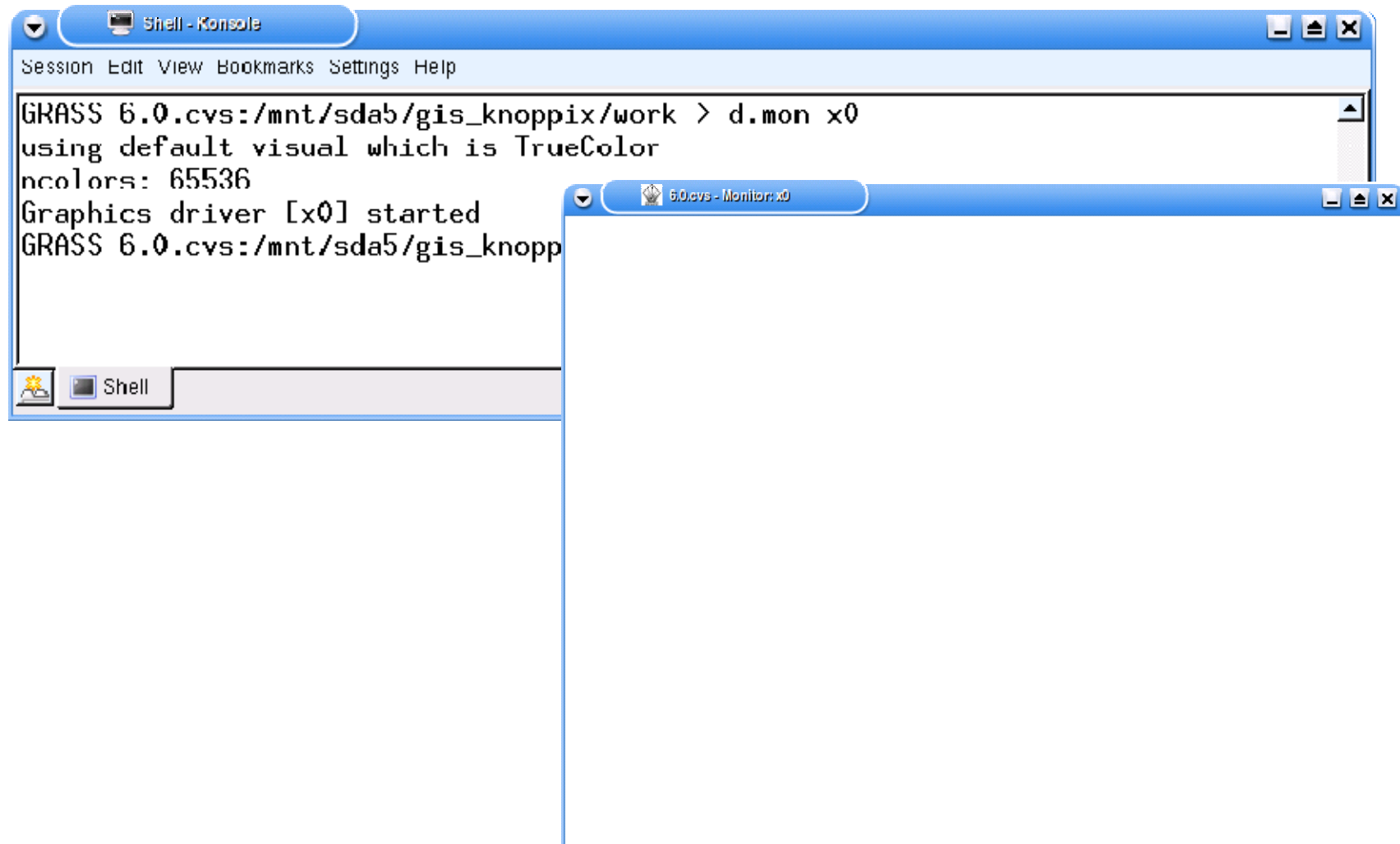
d.rast displays raster map layer(s) *name* in the active display frame on the graphics monitor.

4.1 Launch Monitor

> d.mon x#

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

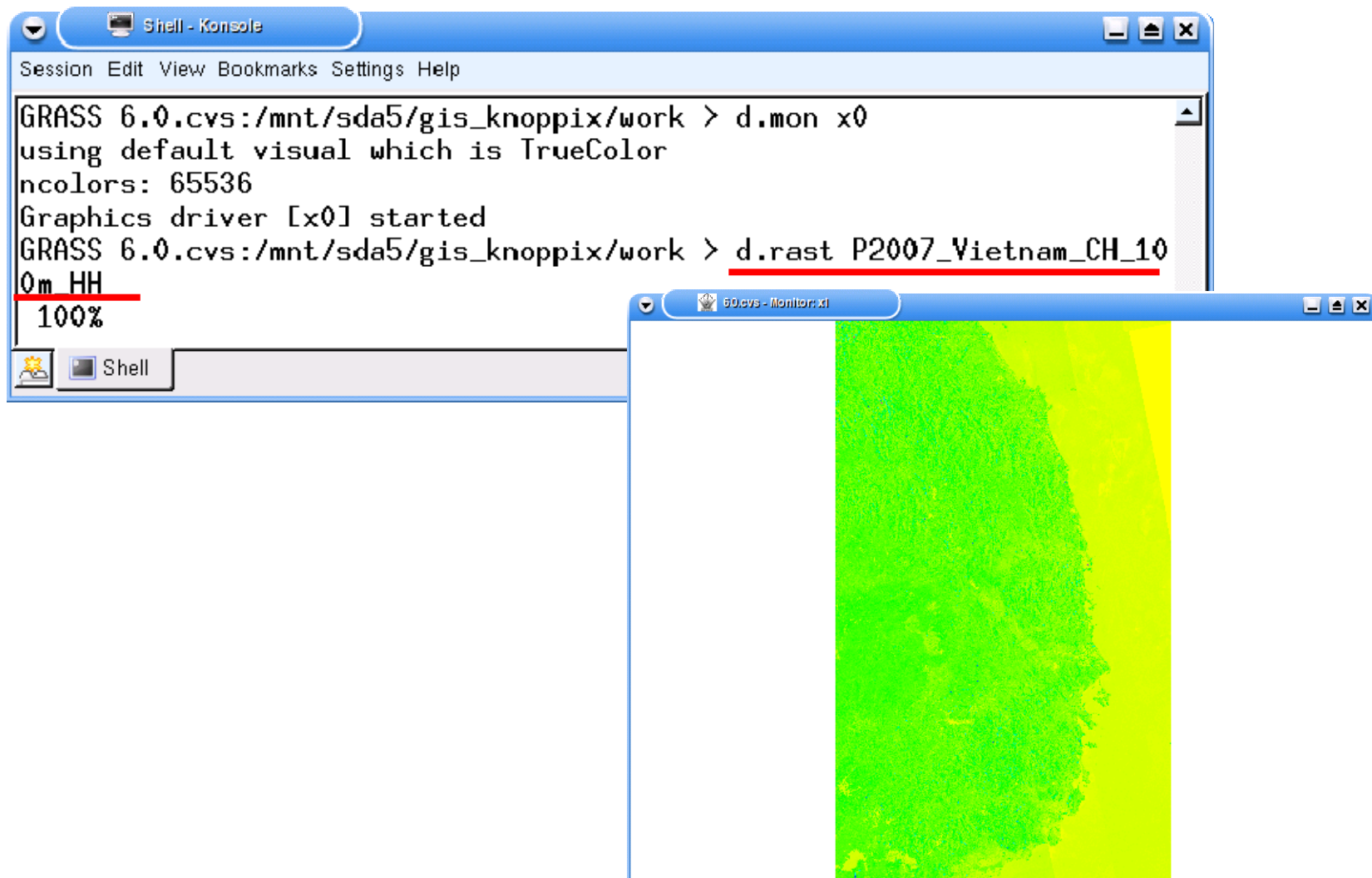
: space



4.2 Display raster image

> d.rast map = A

A = (input file name) : space



4.3 Change image color

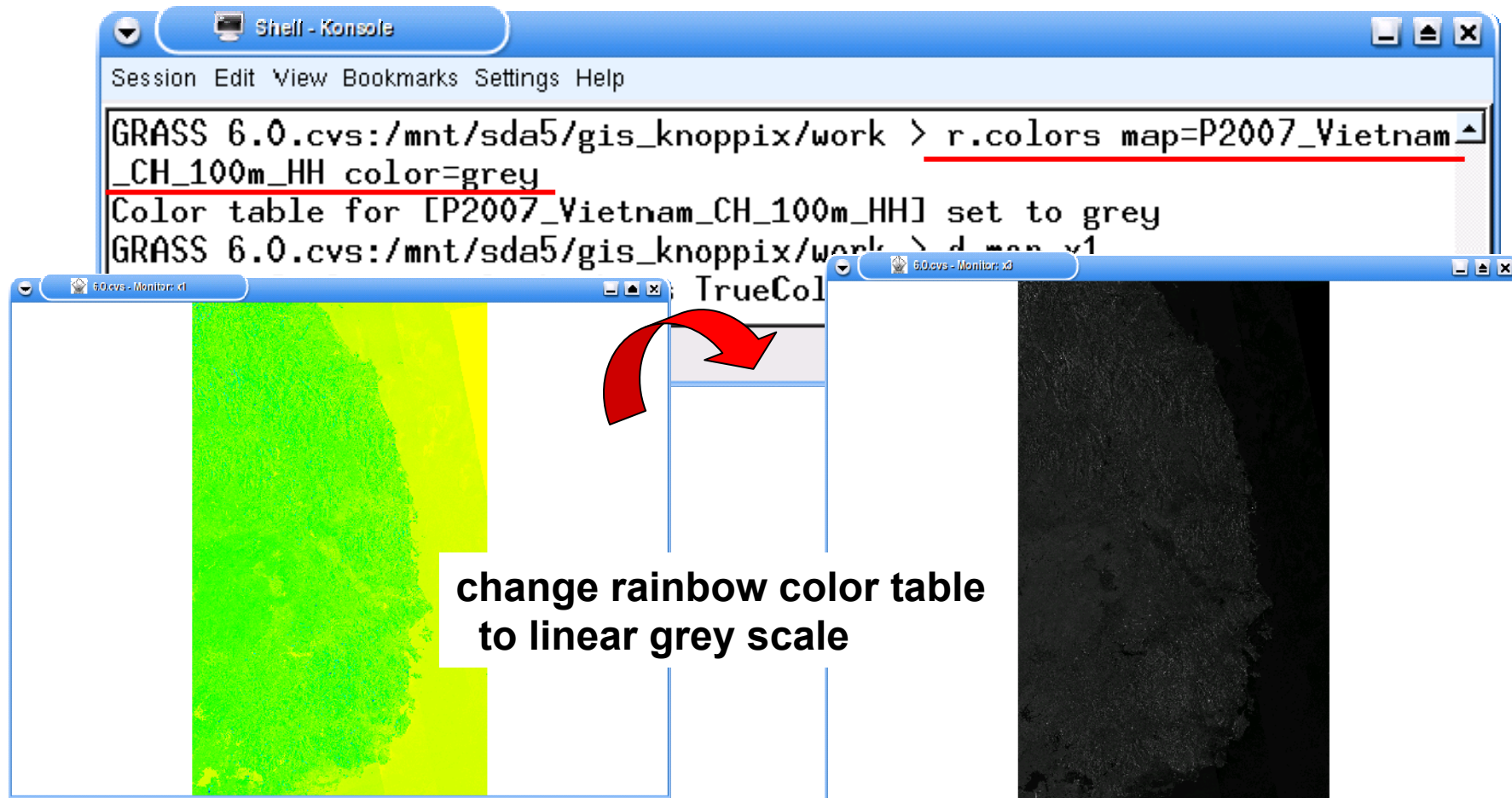
: space

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

A = (input file name)

B = (type of color table)

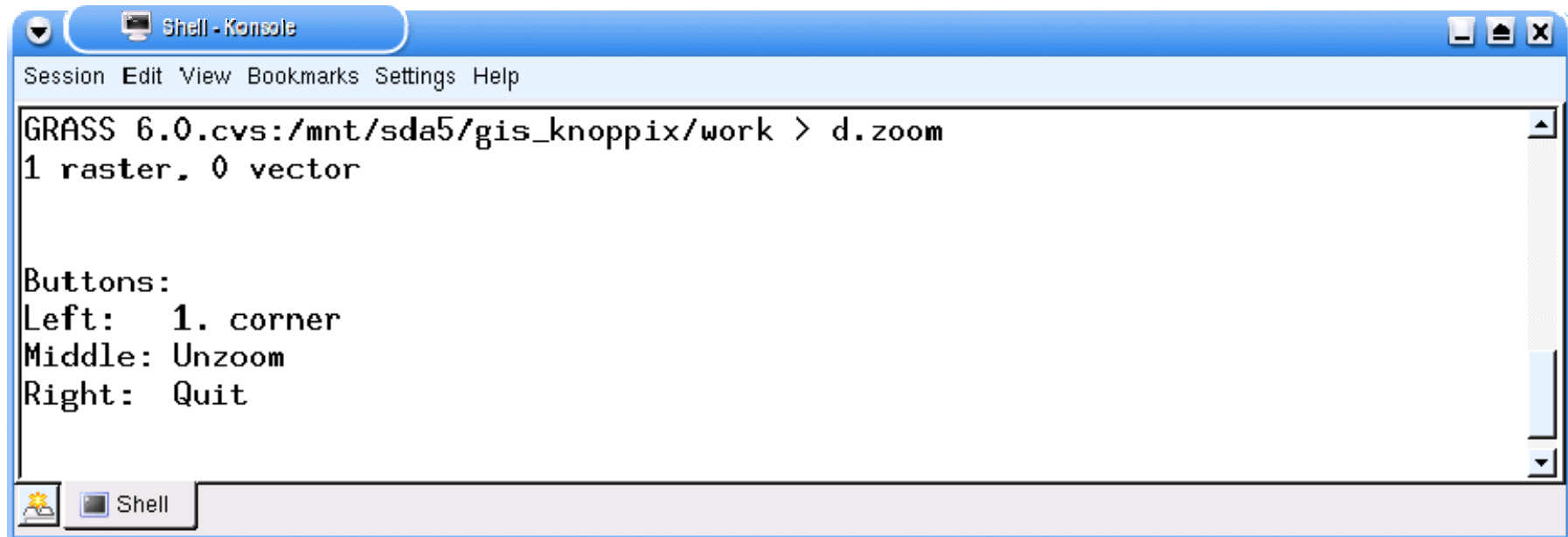
r.colors allows the user to create and/or modify the color table for a raster map layer.



4.4 Zoom in the image

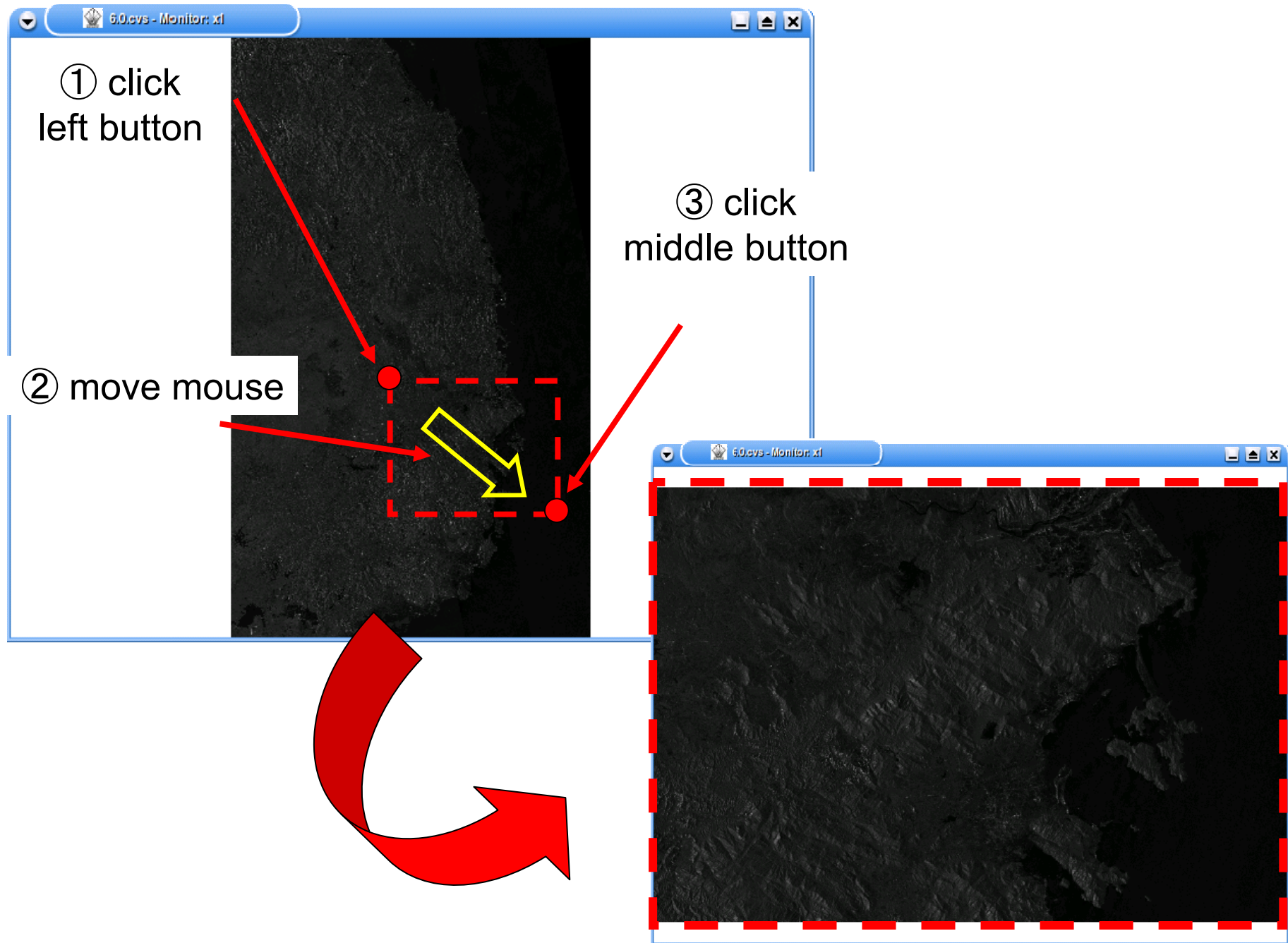
```
> d.zoom
```

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

A screenshot of a terminal window titled "Shell - Konsole". The window has a menu bar with "Session", "Edit", "View", "Bookmarks", "Settings", and "Help". The terminal text shows the command "d.zoom" being executed in a GRASS 6.0 environment. The output indicates that 1 raster and 0 vector layers are selected. It then lists the mouse button functions: Left for corner, Middle for Unzoom, and Right for Quit. The window has standard Linux window controls and a taskbar at the bottom with a "Shell" icon.

```
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > d.zoom
1 raster, 0 vector

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

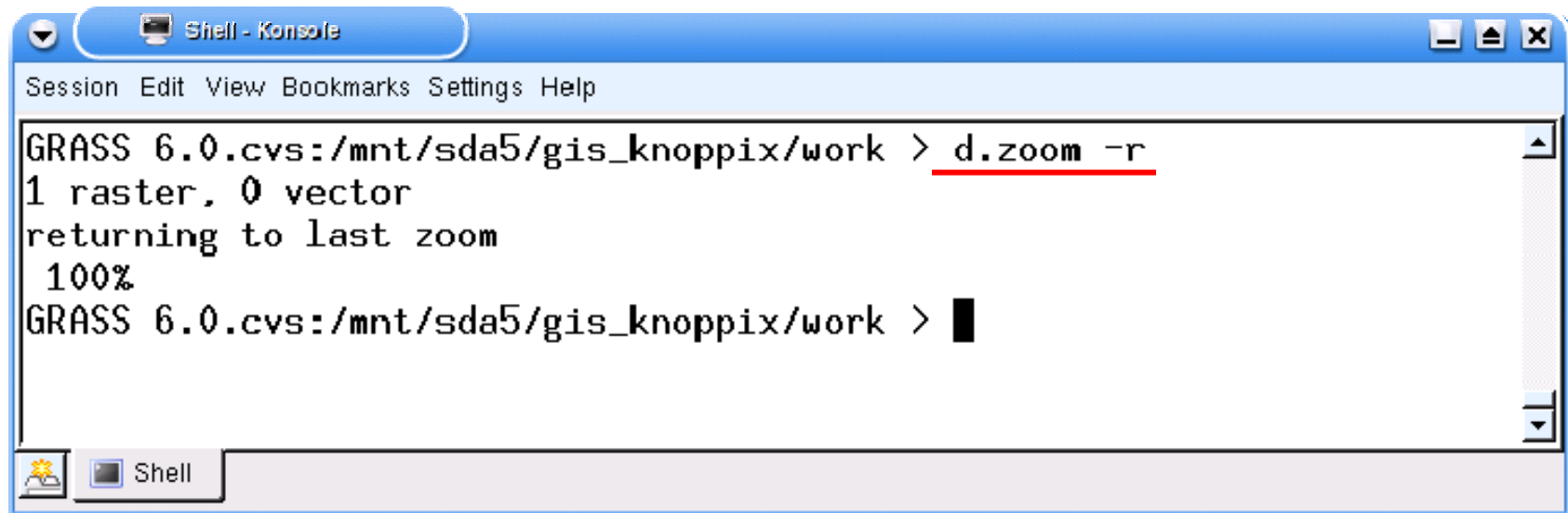


- Return to the previous zoom

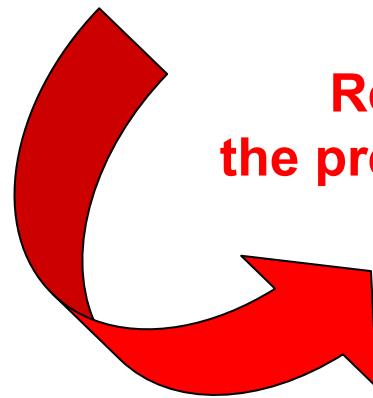
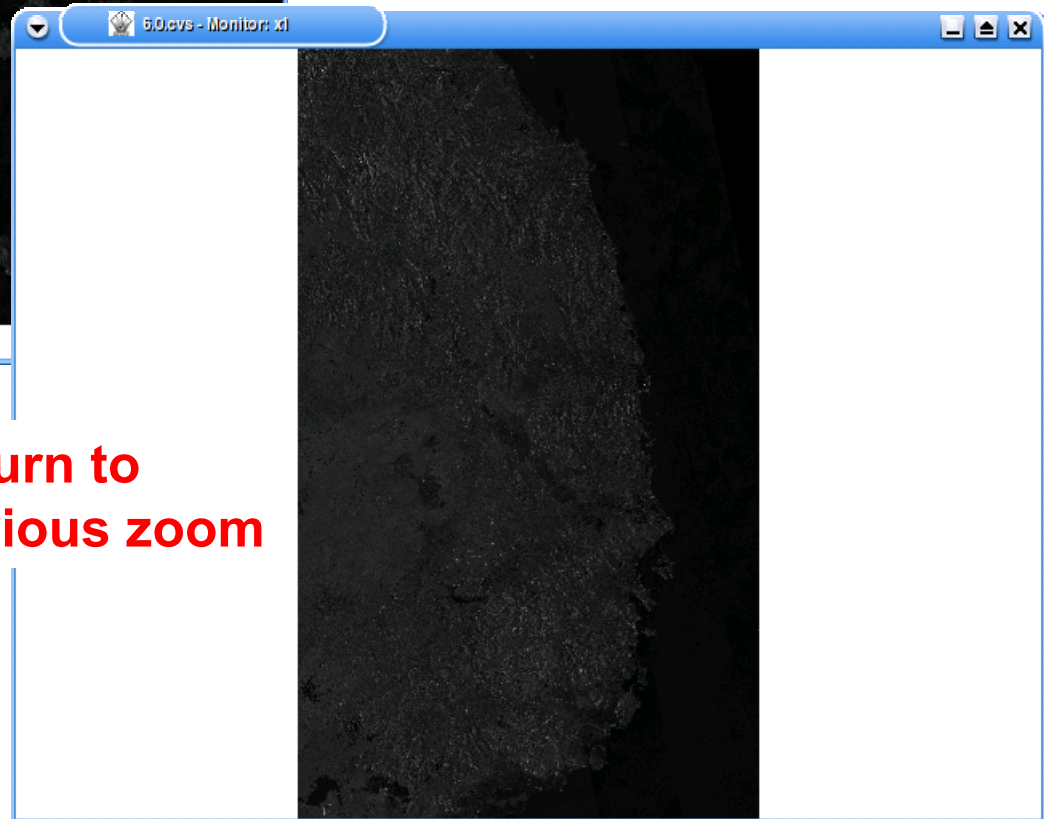
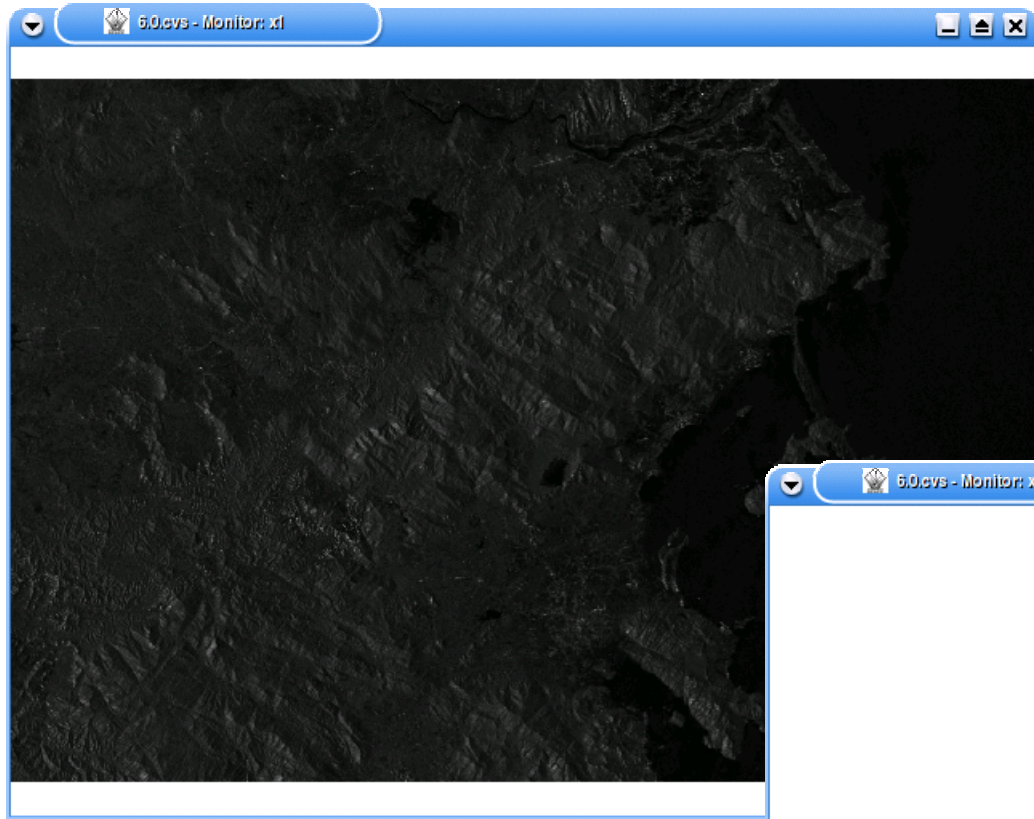
```
> d.zoom -r
```

: space

-r : Flag of “Return to previous zoom”



```
Shell - Konsole
Session Edit View Bookmarks Settings Help
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > d.zoom -r
1 raster, 0 vector
returning to last zoom
100%
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > █
```

**Return to
the previous zoom**

Section 2. Data Processing in Latitude-Longitude Coordinate System

5. Raster calculation

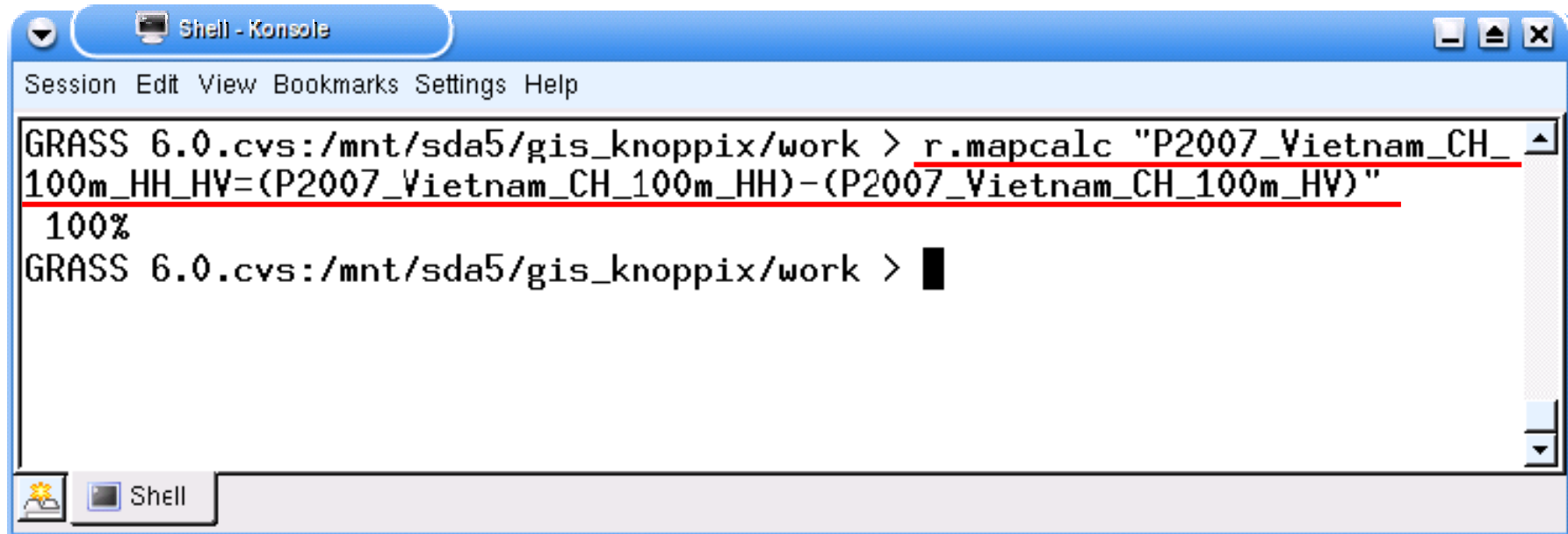
5.1 Calculation of “HH-HV”

> r.mapcalc "HH_HV = A - B"

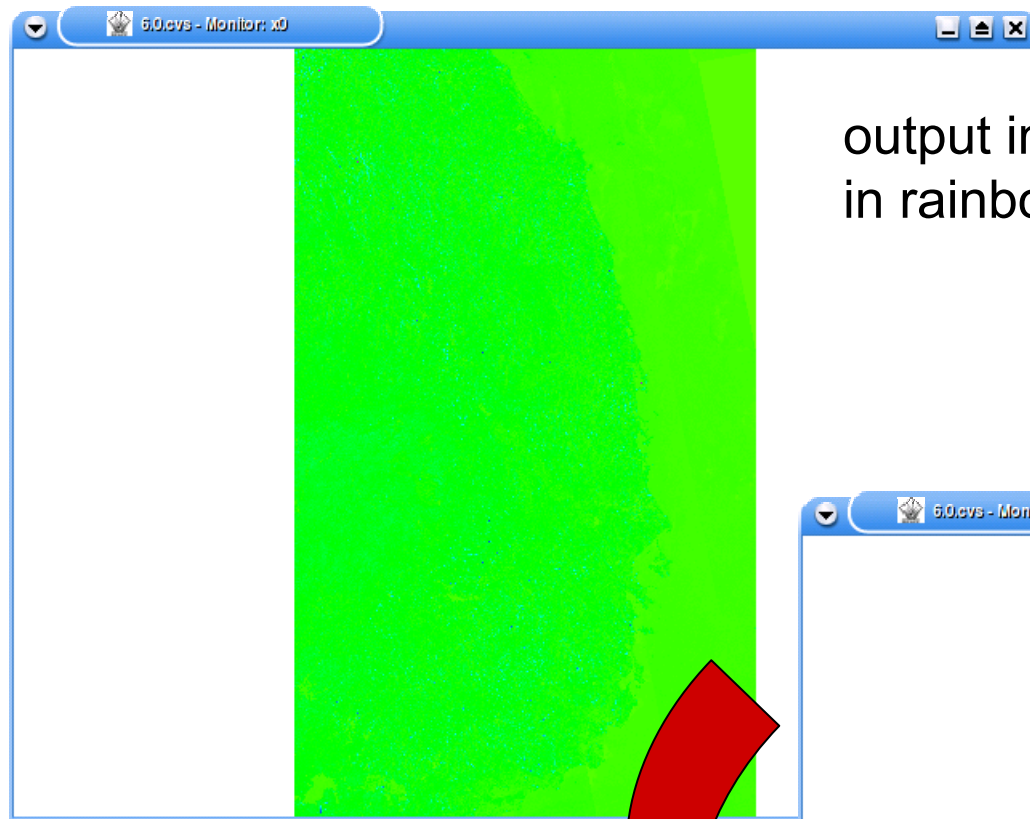
A = (name of HH image)

B = (name of HV image)

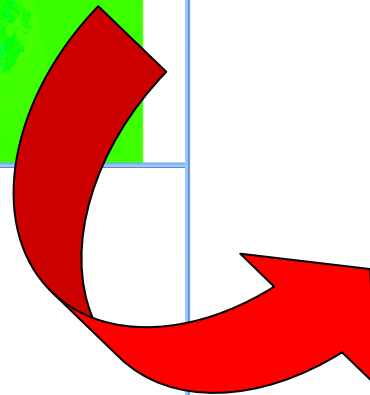
: space



```
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.mapcalc "P2007_Vietnam_CH_100m_HH_HV=(P2007_Vietnam_CH_100m_HH)-(P2007_Vietnam_CH_100m_HV)" 100%
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work >
```

output image of "HH-HV"
in rainbow color table



**change rainbow color table
to histogram-equalized grey scale**



Section 2. SAR Image Processing in Latitude-Longitude Coordinate System

6. Create RGB image

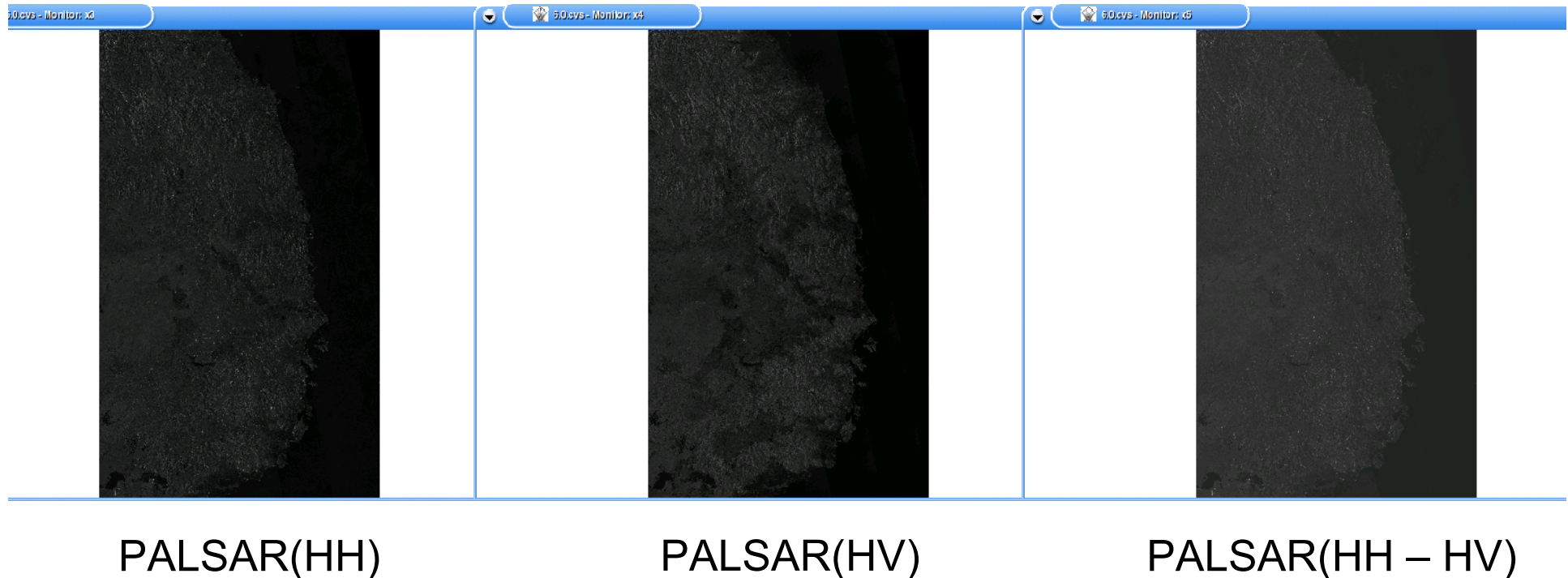
6.1 change color map to “grey scale map”

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

A = (input file name)

⇒ see “4.3 Change image color”

[] : space



6.2 Display three images in RGB composite

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

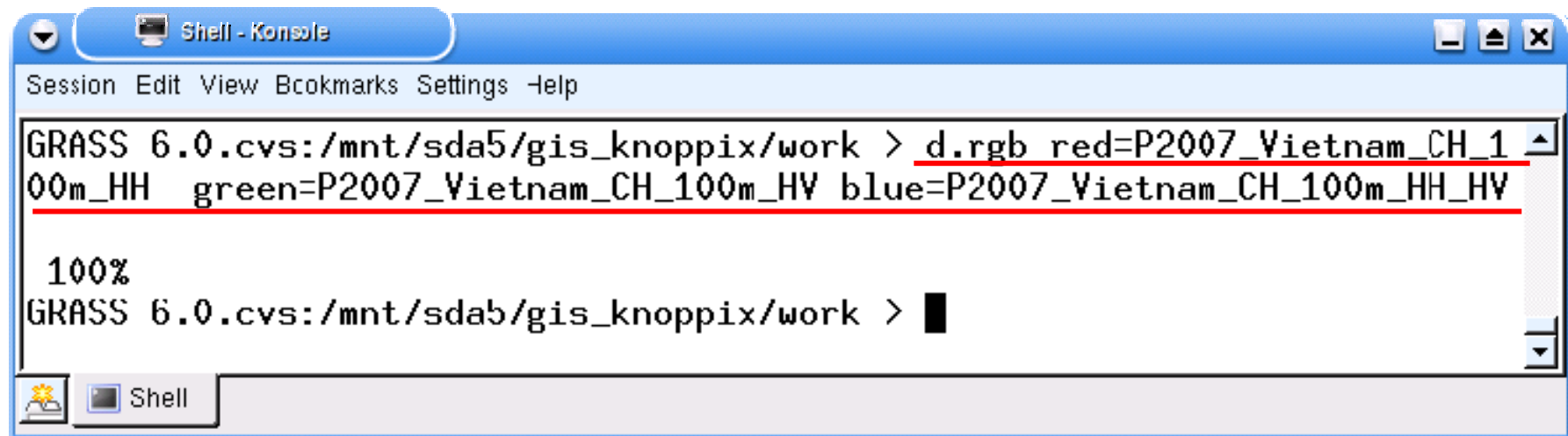
A = (file name1)

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.**



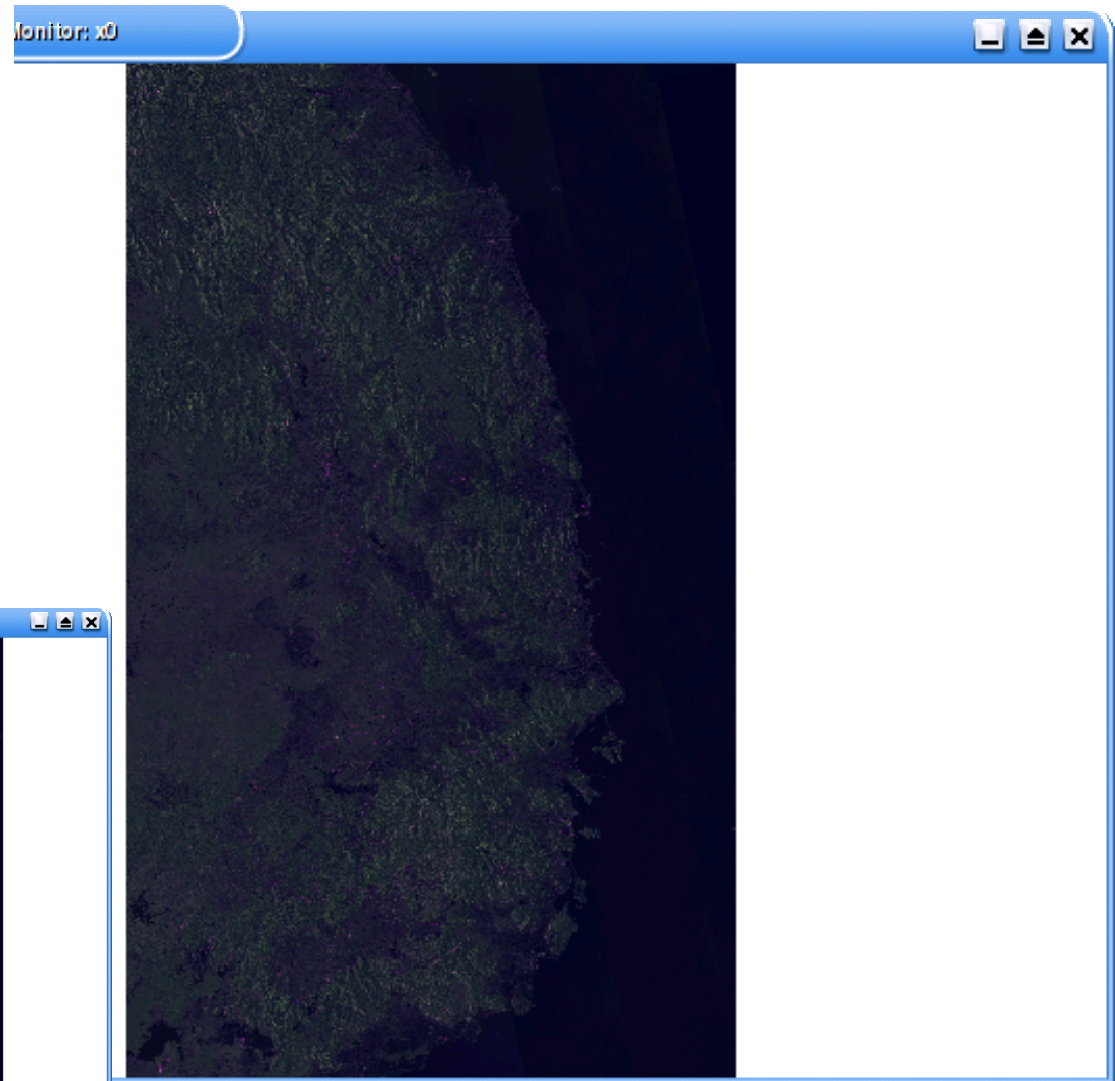
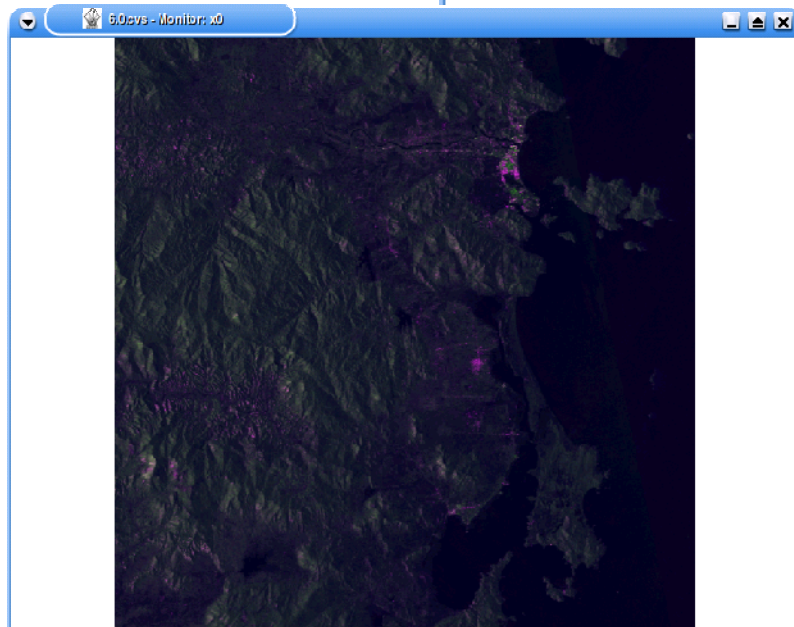
```
Shell - Konsole
Session Edit View Bcokmarks Settings -help
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > d.rgb red=P2007_Vietnam_CH_1
00m_HH green=P2007_Vietnam_CH_100m_HV blue=P2007_Vietnam_CH_100m_HH_HV
100%
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > █
```

In this case, **R=HH, G=HV, B=HH-HV**

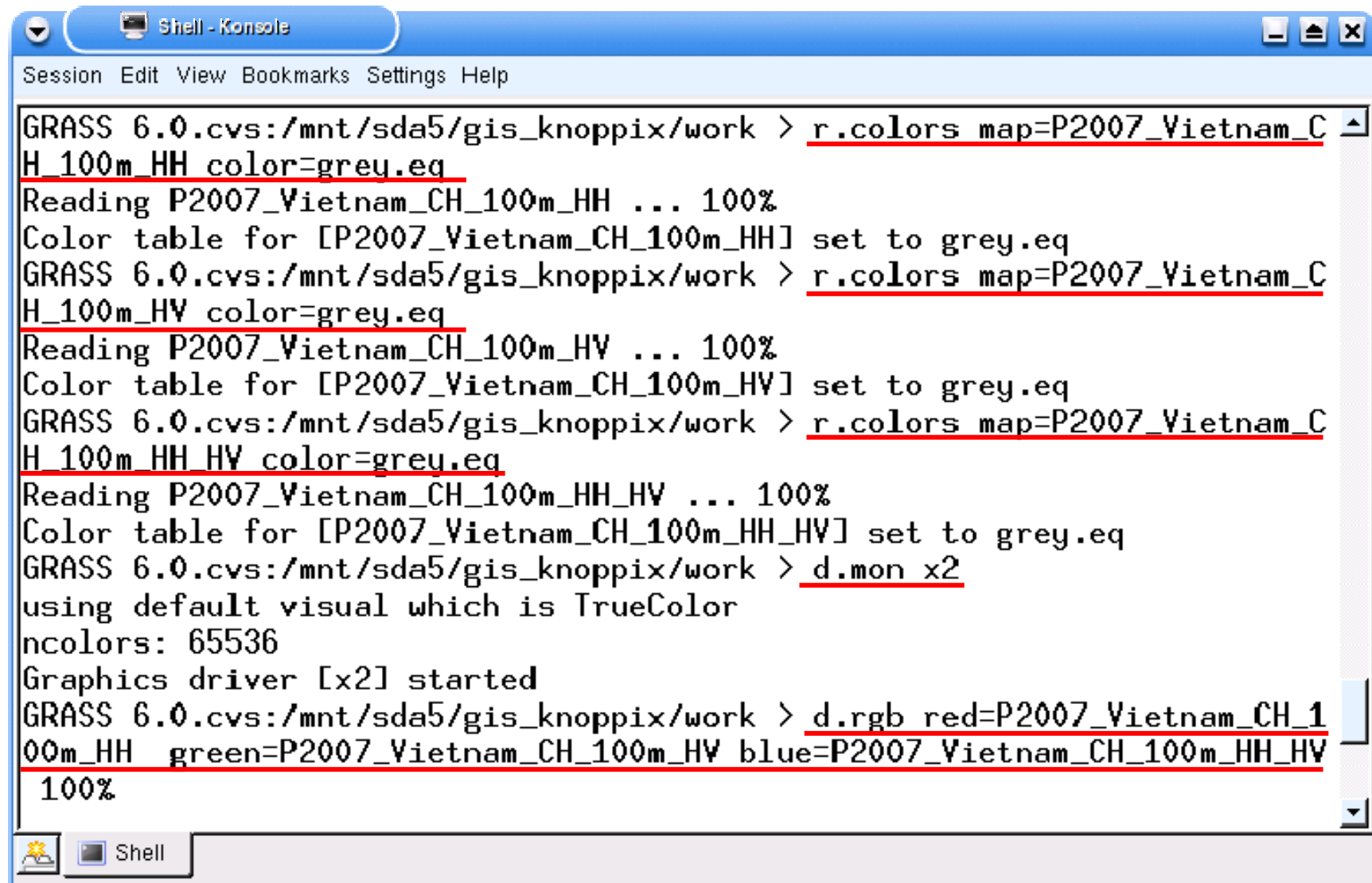
6.2.1 Display grey scale images in RGB

Forested area are represented in dark green color and cropland in dark purple.

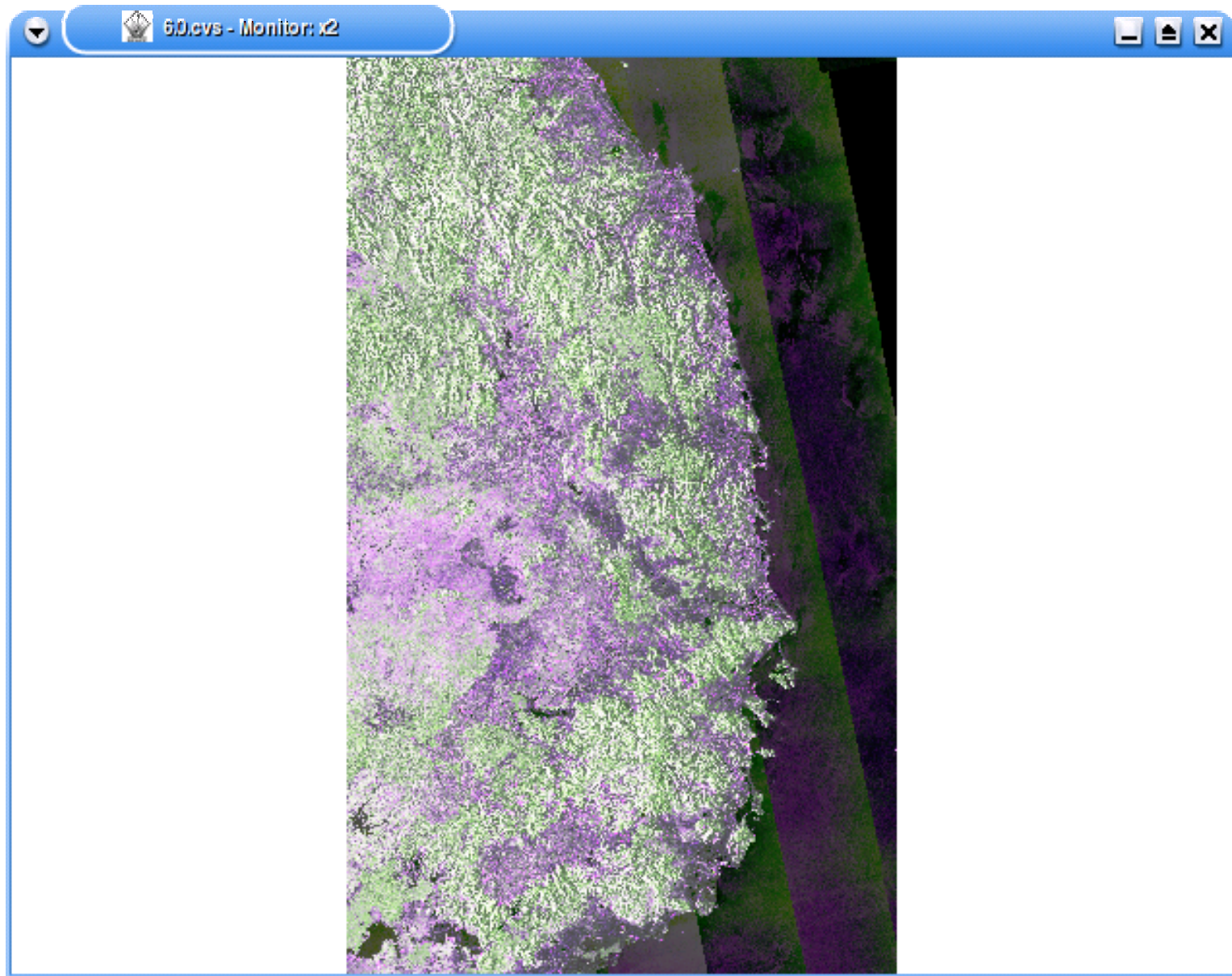
Cross-polarized response (HH) is dominated by the canopy contribution rather than the surface component.



6.2.2 Display histogram-equalized grey scale images in RGB



```
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.colors map=P2007_Vietnam_C  
H_100m_HH color=grey.eq  
Reading P2007_Vietnam_CH_100m_HH ... 100%  
Color table for [P2007_Vietnam_CH_100m_HH] set to grey.eq  
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.colors map=P2007_Vietnam_C  
H_100m_HV color=grey.eq  
Reading P2007_Vietnam_CH_100m_HV ... 100%  
Color table for [P2007_Vietnam_CH_100m_HV] set to grey.eq  
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.colors map=P2007_Vietnam_C  
H_100m_HH_HV color=grey.eq  
Reading P2007_Vietnam_CH_100m_HH_HV ... 100%  
Color table for [P2007_Vietnam_CH_100m_HH_HV] set to grey.eq  
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > d.mon x2  
using default visual which is TrueColor  
ncolors: 65536  
Graphics driver [x2] started  
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > d.rgb red=P2007_Vietnam_CH_1  
00m_HH green=P2007_Vietnam_CH_100m_HV blue=P2007_Vietnam_CH_100m_HH_HV  
100%
```



6.3 Create RGB image using three images

Step 0. Import images for R,G, B

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

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

A = (input file name)

[] : space

Step 2. composite images

```
> r.composite [red = A] [green = B] [blue = C] [output = D]
```

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

Step 3. save composite images as TIFF file

```
> r.out.tiff [input = D] [output = E]
```

D = (input file name) E = (output file name)

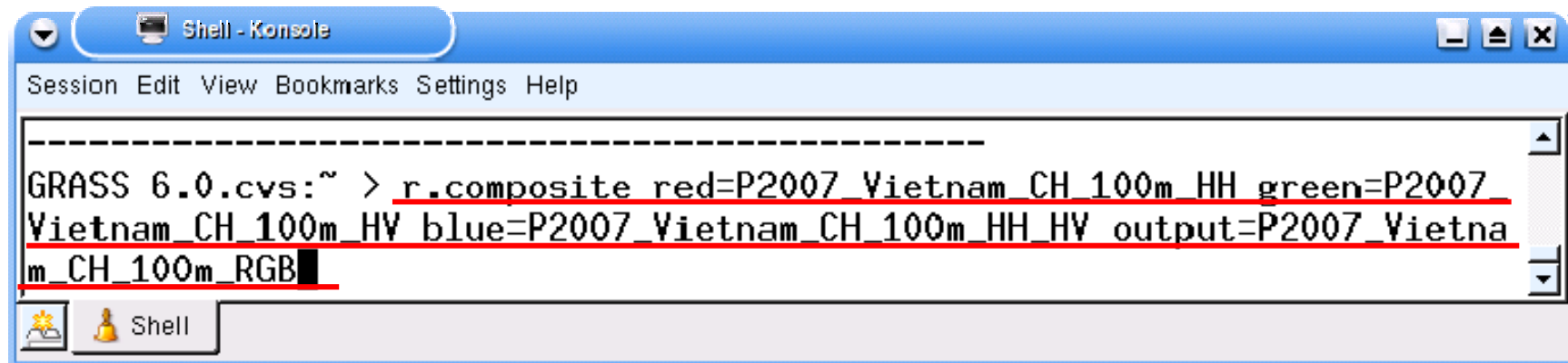
6.3.1 Create composite image

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

A, B, C = (input file name)

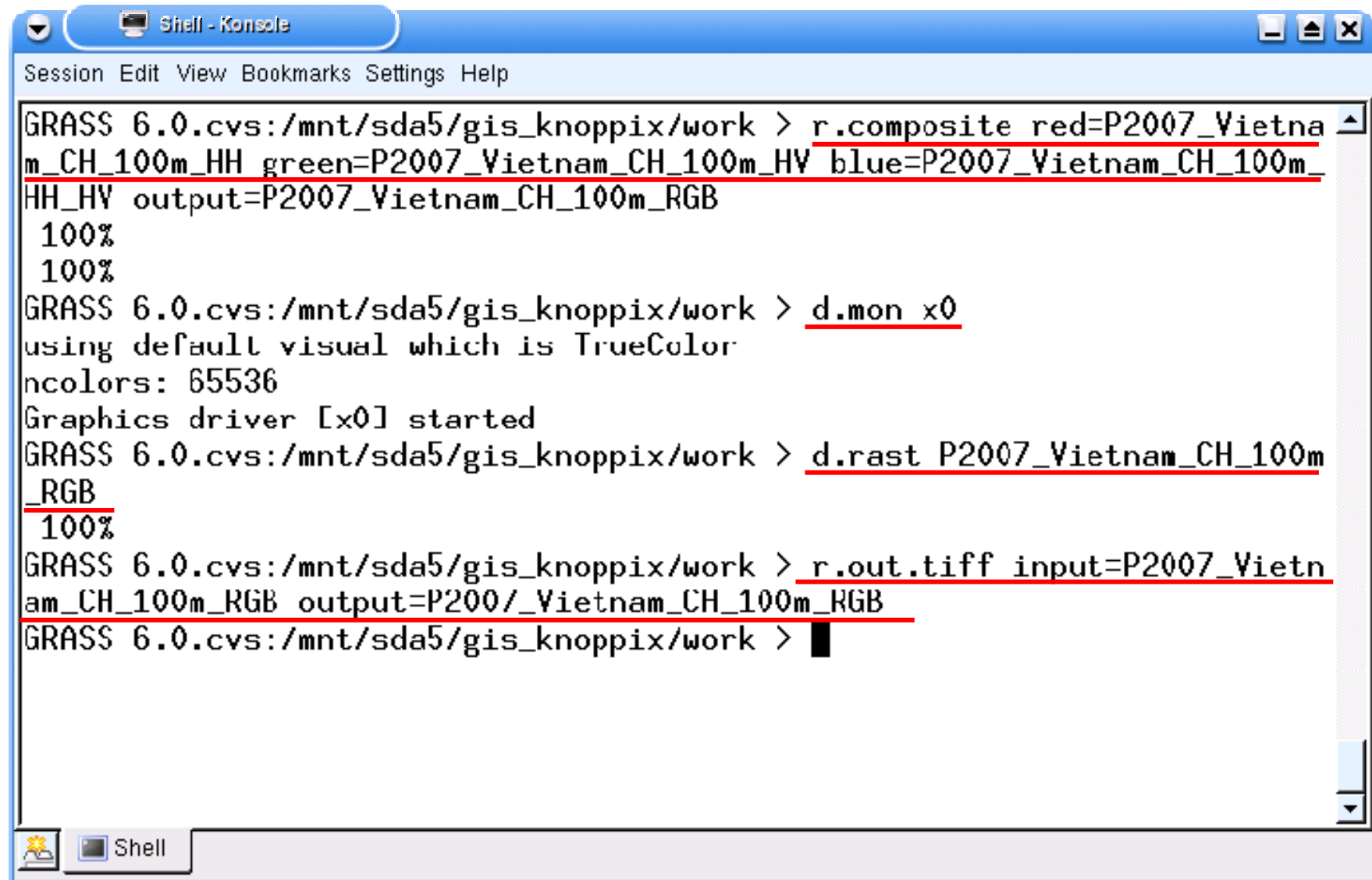
D = (output file name)

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

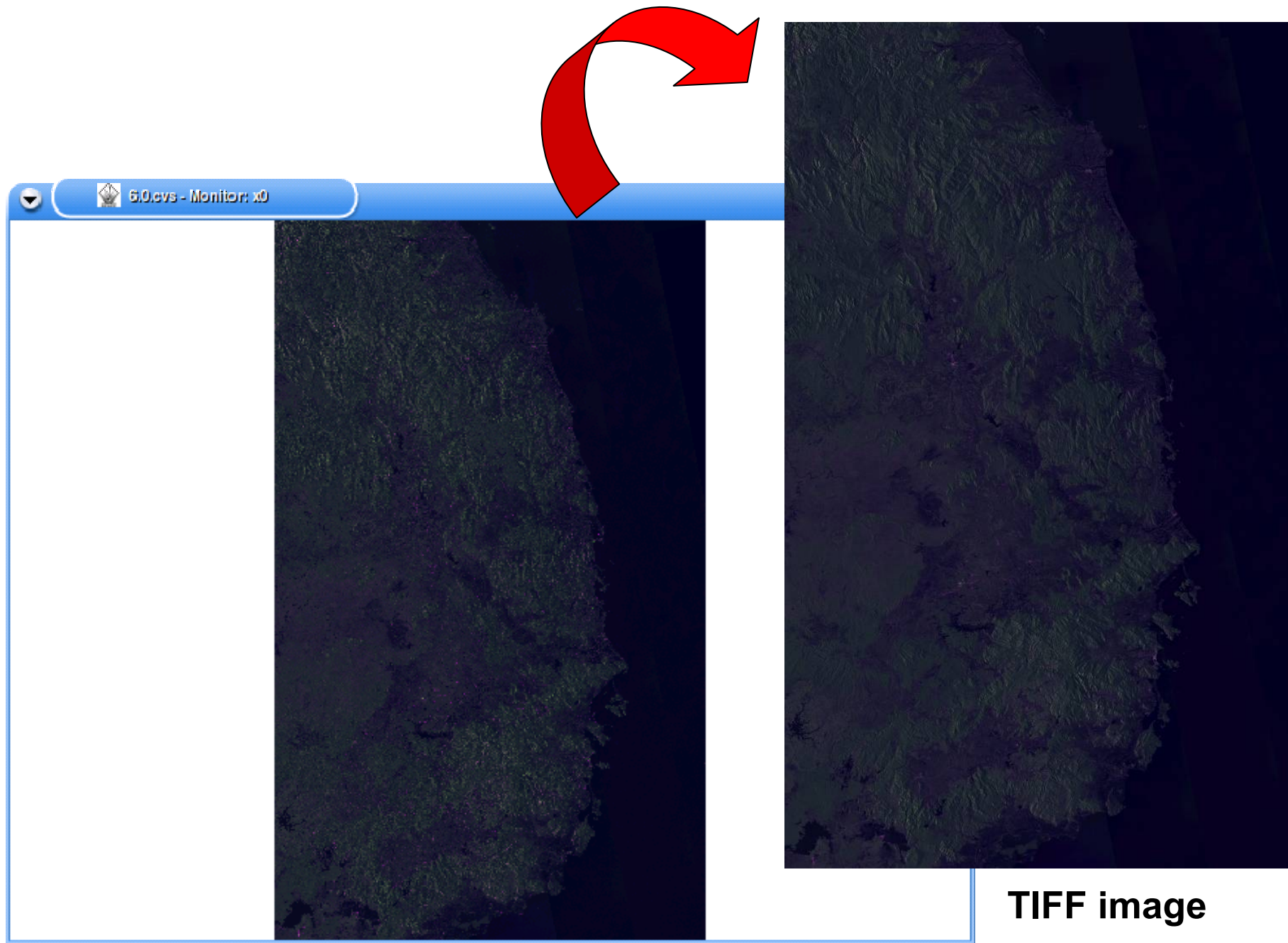


```
-----
GRASS 6.0.cvs:~ > r.composite red=P2007_Vietnam_CH_100m_HH green=P2007_Vietnam_CH_100m_HV blue=P2007_Vietnam_CH_100m_HH_HV output=P2007_Vietnam_CH_100m_RGB
```

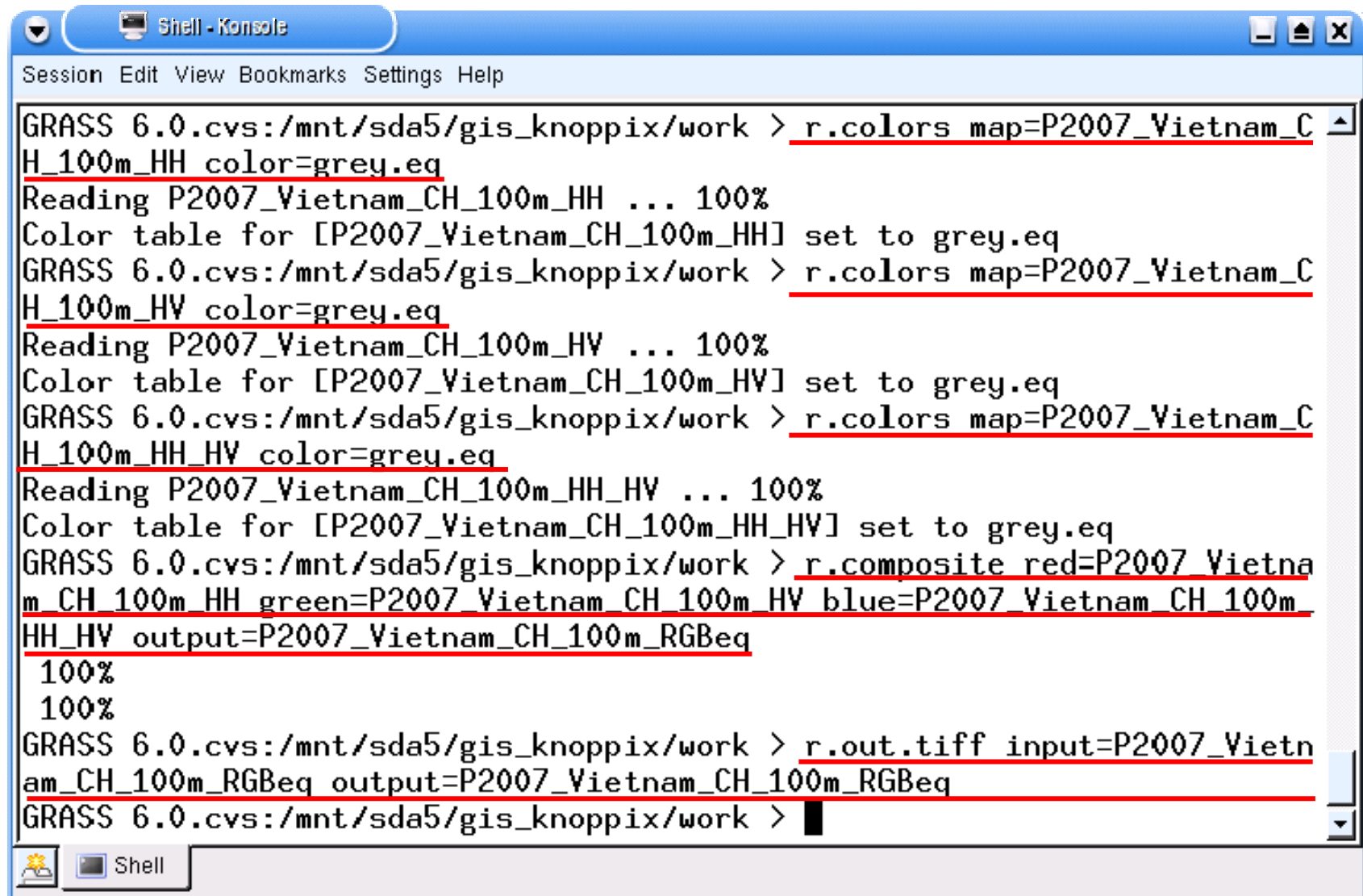
6.3.2 Create, display and export composite image



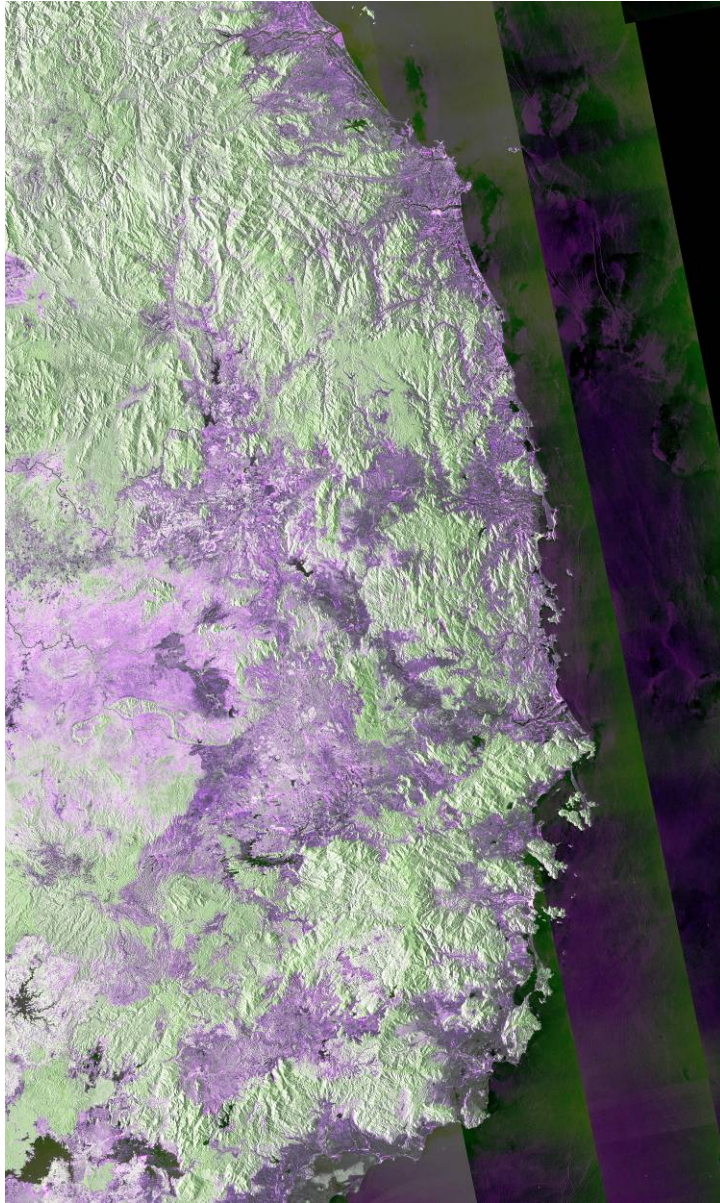
```
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.composite red=P2007_Vietna  
m_CH_100m_HH green=P2007_Vietnam_CH_100m_HV blue=P2007_Vietnam_CH_100m  
HH_HV output=P2007_Vietnam_CH_100m_RGB  
100%  
100%  
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > d.mon x0  
using default visual which is TrueColor  
ncolors: 65536  
Graphics driver [x0] started  
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > d.rast P2007_Vietnam_CH_100m  
_RGB  
100%  
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.out.tiff input=P2007_Vietn  
am_CH_100m_RGB output=P200/_Vietnam_CH_100m_RGB  
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > █
```



6.3.3 Create and export histogram-equalized grey scale images in RGB



```
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.colors map=P2007_Vietnam_CH_100m_HH color=grey.eq
Reading P2007_Vietnam_CH_100m_HH ... 100%
Color table for [P2007_Vietnam_CH_100m_HH] set to grey.eq
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.colors map=P2007_Vietnam_CH_100m_HV color=grey.eq
Reading P2007_Vietnam_CH_100m_HV ... 100%
Color table for [P2007_Vietnam_CH_100m_HV] set to grey.eq
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.colors map=P2007_Vietnam_CH_100m_HH_HV color=grey.eq
Reading P2007_Vietnam_CH_100m_HH_HV ... 100%
Color table for [P2007_Vietnam_CH_100m_HH_HV] set to grey.eq
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.composite red=P2007_Vietnam_CH_100m_HH green=P2007_Vietnam_CH_100m_HV blue=P2007_Vietnam_CH_100m_HH_HV output=P2007_Vietnam_CH_100m_RGBeq
100%
100%
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.out.tiff input=P2007_Vietnam_CH_100m_RGBeq output=P2007_Vietnam_CH_100m_RGBeq
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > █
```

Section 2. SAR Image Processing in Latitude-Longitude Coordinate System

7. Terrain Correction

7.1 Calculation of slope and aspect

7.2 Calculation of local incident angle

7.3 Terrain correction of PALSAR image

7.4 Terrain correction of JERS-1 SAR image

7.1 Calculation of slope and aspect

```
> r.slope.aspect [elevation=A [slope=B [aspect=C
```

A = (input DEM filename)

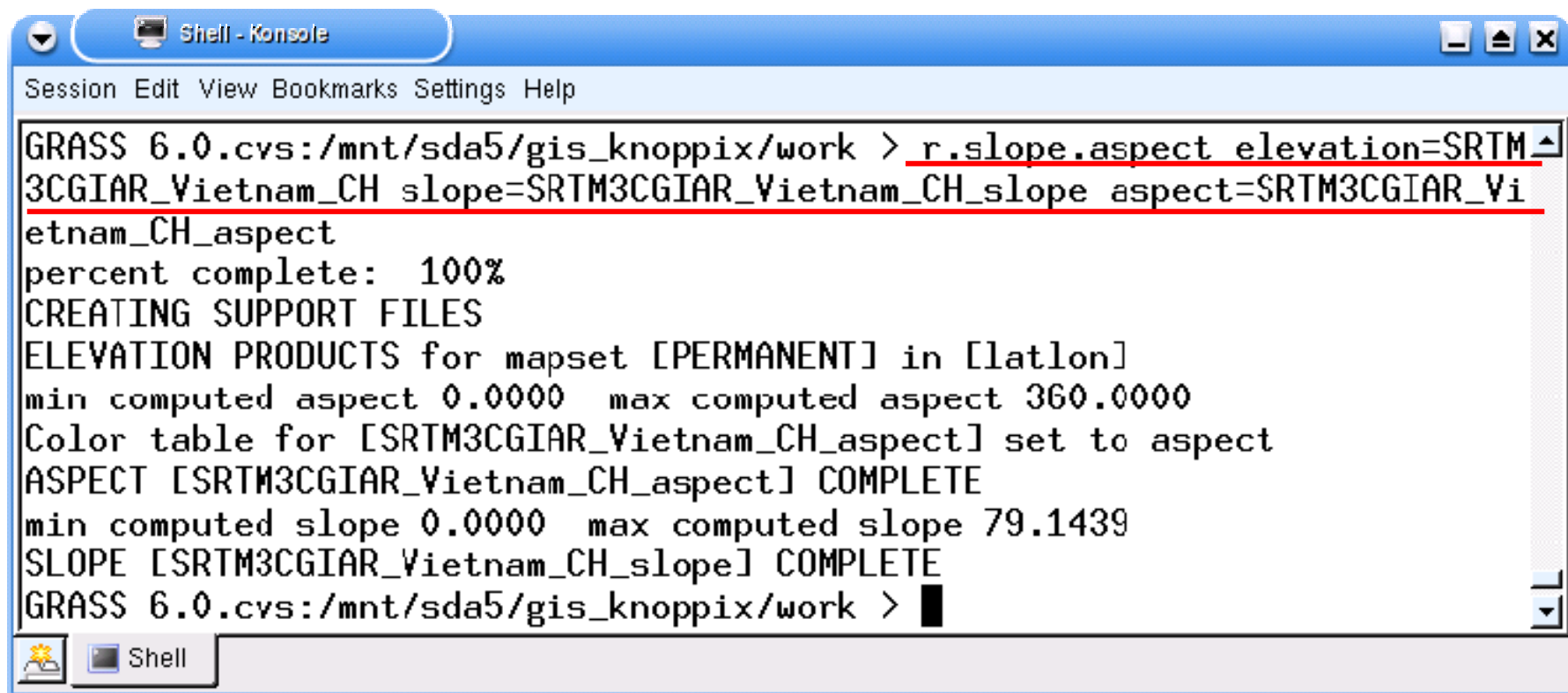
B = (output slope filename)

C = (output aspect filename)

[] : space

r.slope.aspect - Generates raster map layers of slope, aspect, curvatures and partial derivatives from a raster map layer of true elevation values. Aspect is calculated counterclockwise from east.

- The raster *slope* map layer will contain slope values, stated in degrees of inclination from the horizontal.
- The raster *aspect* map layer which is created indicates the direction that slopes are facing. The aspect categories represent the number degrees of east.. The aspect categories represent the number degrees of east and they increase counterclockwise: 90deg is North, 180 is West, 270 is South and 360 is East.



```
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.slope.aspect elevation=SRTM3CGIAR_Vietnam_CH slope=SRTM3CGIAR_Vietnam_CH_slope aspect=SRTM3CGIAR_Vietnam_CH_aspect
percent complete: 100%
CREATING SUPPORT FILES
ELEVATION PRODUCTS for mapset [PERMANENT] in [latlon]
min computed aspect 0.0000 max computed aspect 360.0000
Color table for [SRTM3CGIAR_Vietnam_CH_aspect] set to aspect
ASPECT [SRTM3CGIAR_Vietnam_CH_aspect] COMPLETE
min computed slope 0.0000 max computed slope 79.1439
SLOPE [SRTM3CGIAR_Vietnam_CH_slope] COMPLETE
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > █
```

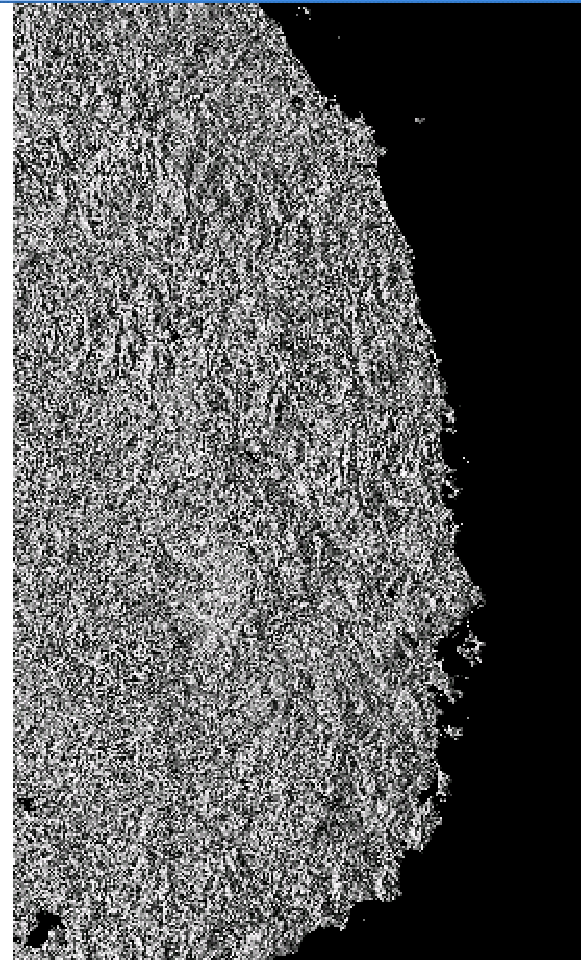
6.0cvs - Monitor: x2



Slope

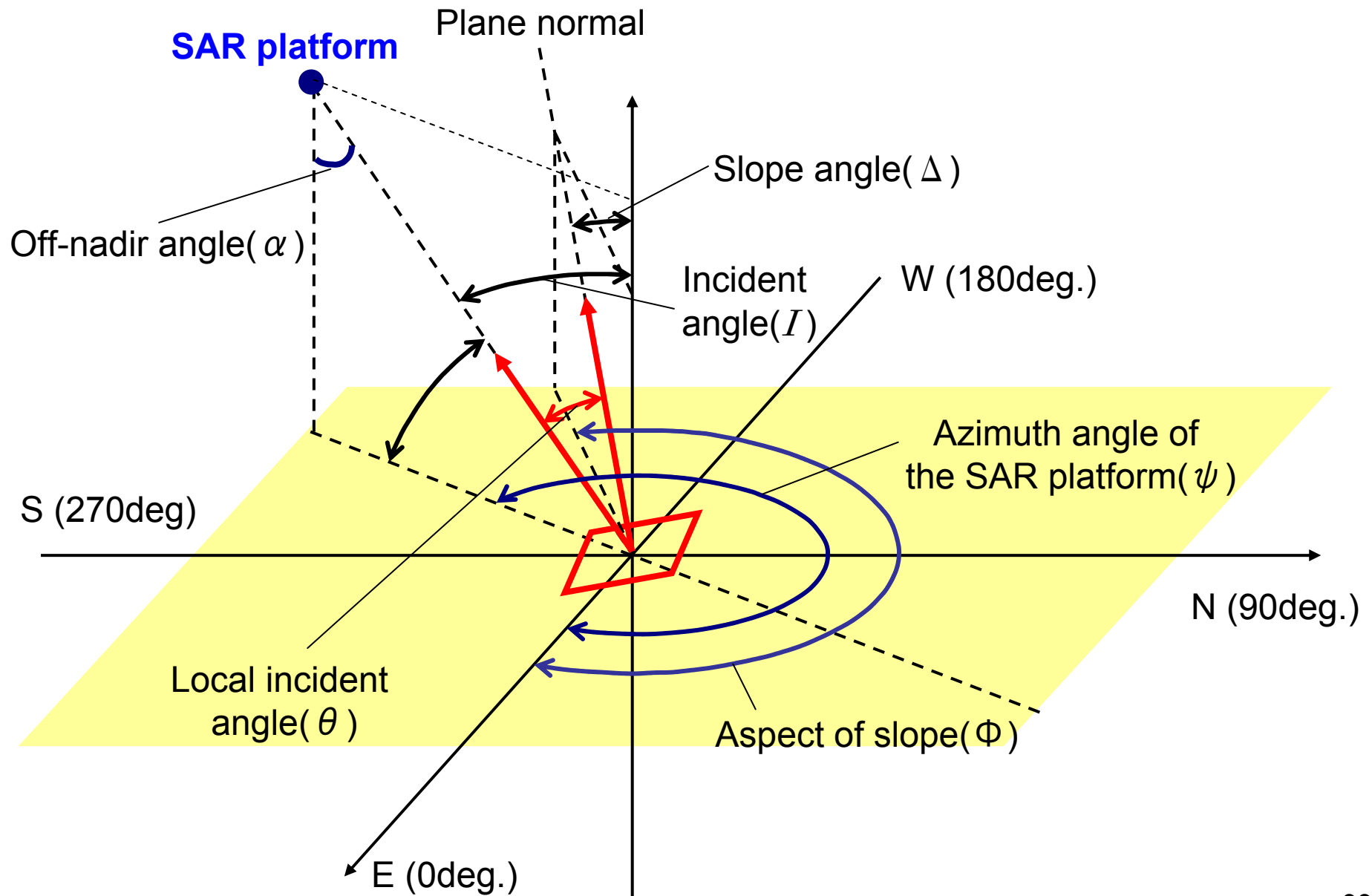


6.0cvs - Monitor: x3



Aspect

7.2 Calculation of local incident angle



```
> r.mapcalc "incident_angle=sin(A)*cos(B)*sin(I)*cos(ψ) +  
sin(A)*sin(B)*sin(I)*sin(ψ) + cos(A)*cos(I)"
```

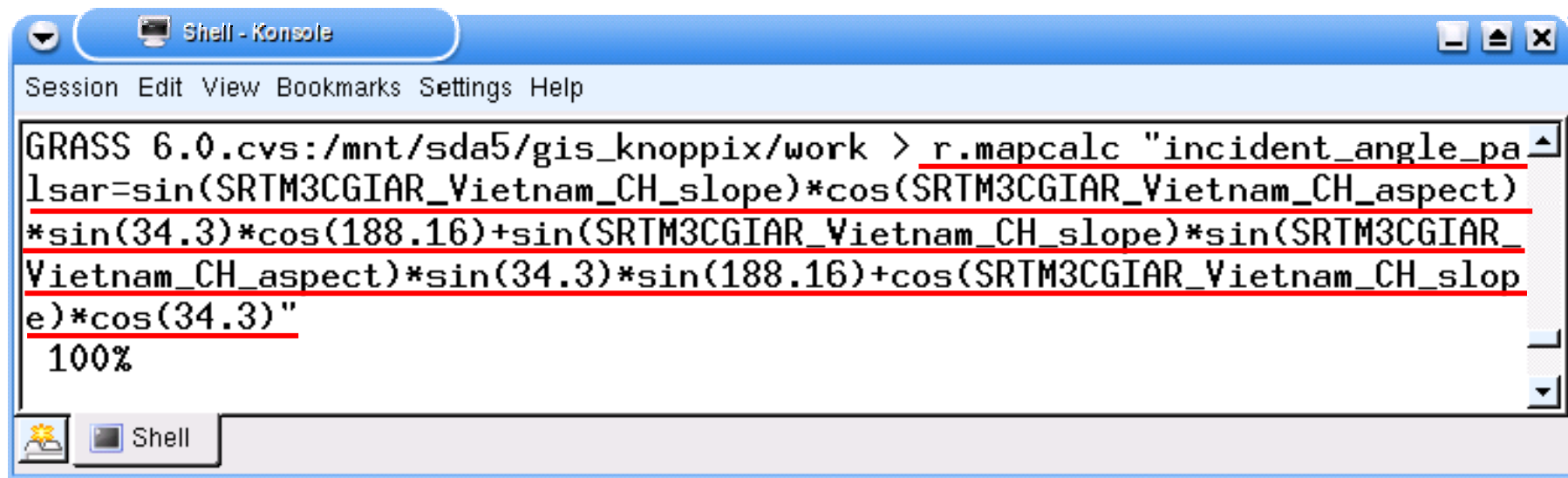
A = (Slope filename)

B = (Aspect filename)

I = Incident angle(= Off-nadir angle)

ψ = Azimuth angle of the SAR platform

␣ : space



```
Shell - Konsole  
Session Edit View Bookmarks Settings Help  
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.mapcalc "incident_angle_pa  
lsar=sin(SRTM3CGIAR_Vietnam_CH_slope)*cos(SRTM3CGIAR_Vietnam_CH_aspect)  
*sin(34.3)*cos(188.16)+sin(SRTM3CGIAR_Vietnam_CH_slope)*sin(SRTM3CGIAR  
Vietnam_CH_aspect)*sin(34.3)*sin(188.16)+cos(SRTM3CGIAR_Vietnam_CH_slop  
e)*cos(34.3)"  
100%
```

7.3 Terrain correction of PALSAR image

Step 0. Import DEM image

\Rightarrow *see* “ 3.2 ”

Step 1. Create slope and aspect images using DEM

\Rightarrow *see* “ 7.1 ”

Step 2. Create local incident angle image
using the slope and aspect images

\Rightarrow *see* “ 7.2 ”

$$l = 34.3^\circ$$

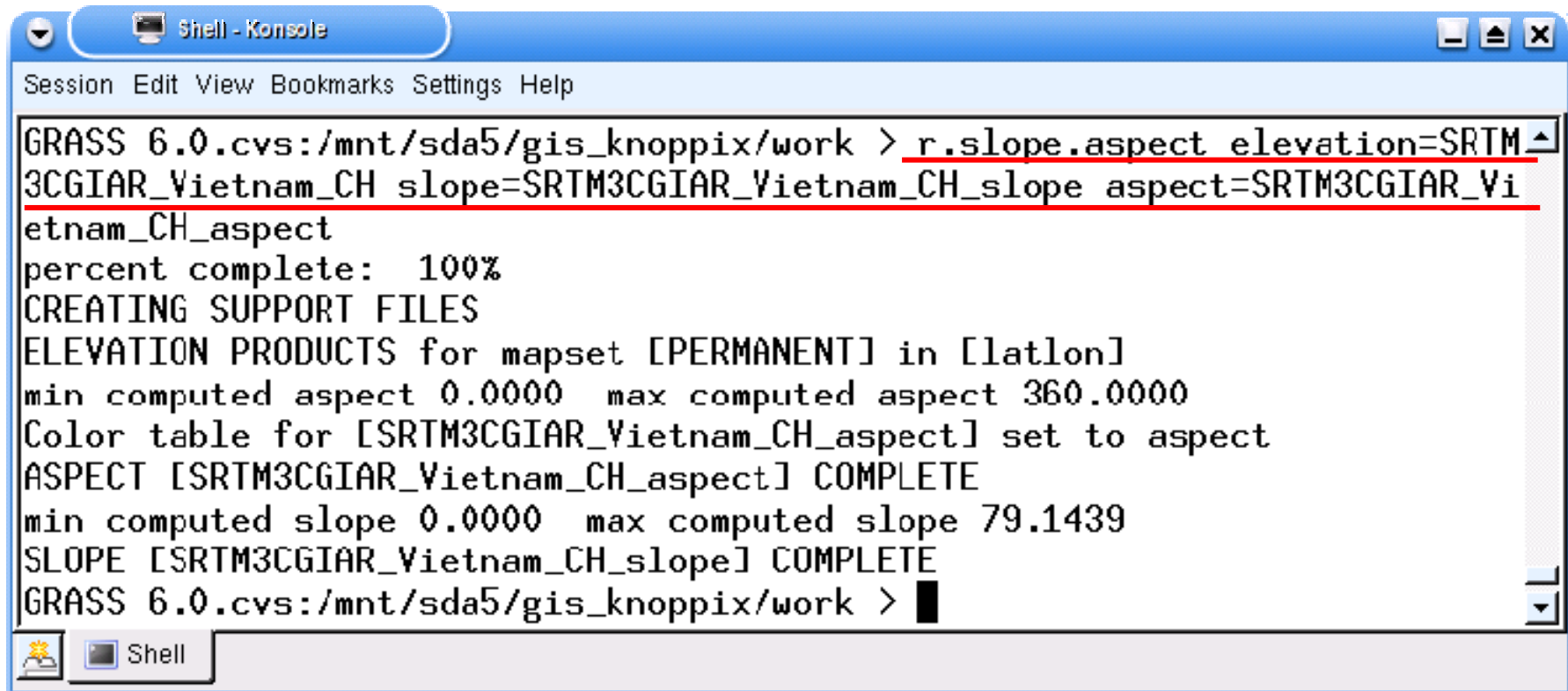
$$\psi = 188.16^\circ$$

Step 3. Correct terrain effects of PALSAR image
using local incident image

7.3.1 Create slope and aspect images using DEM

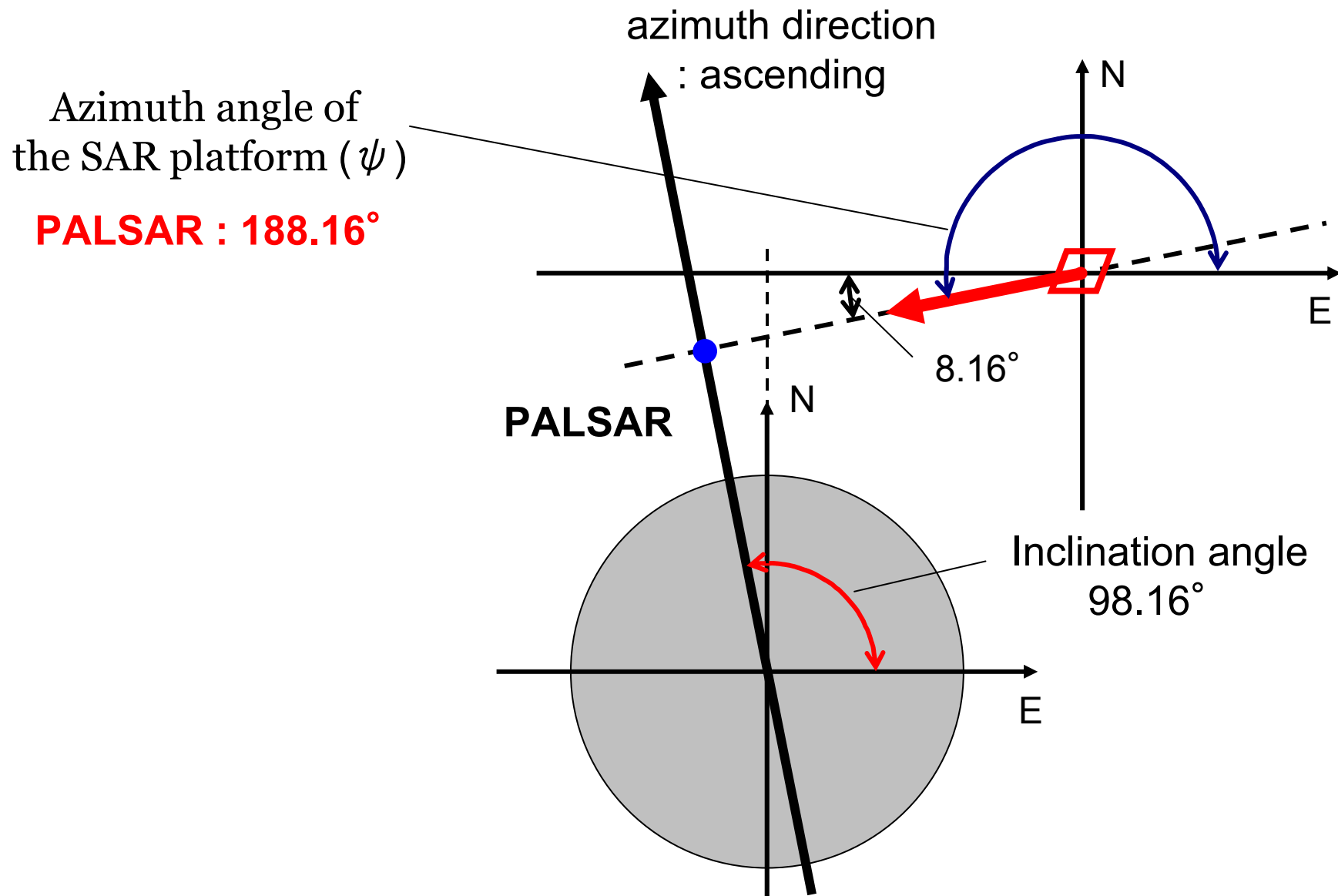
```
> r.slope.aspect elevation=SRTM3CGIAR_Vietnam_CH slope=SRTM3CGIAR_Vietnam_CH_slope aspect=SRTM3CGIAR_Vietnam_CH_aspect
```

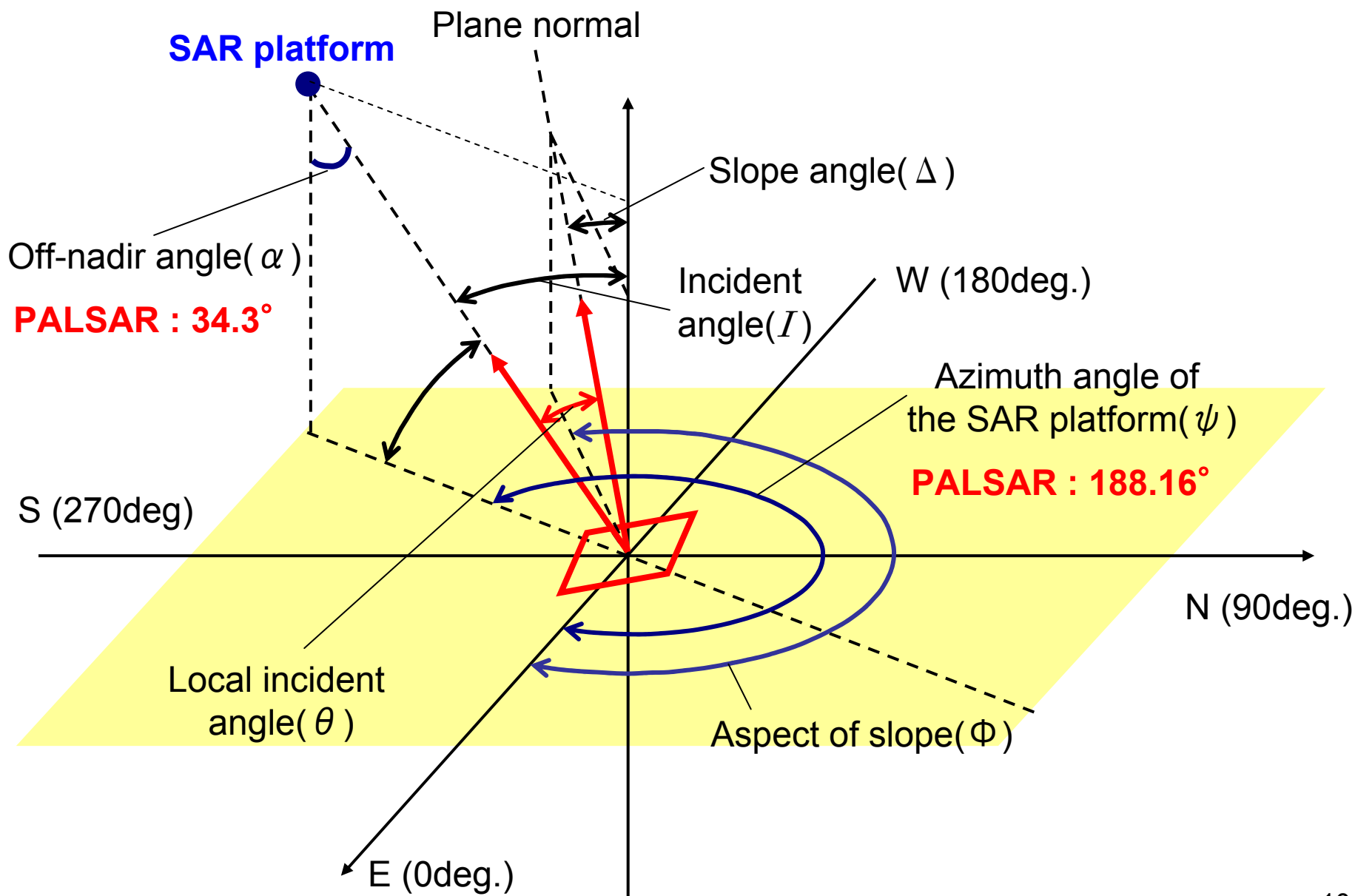
: space



```
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.slope.aspect elevation=SRTM3CGIAR_Vietnam_CH slope=SRTM3CGIAR_Vietnam_CH_slope aspect=SRTM3CGIAR_Vietnam_CH_aspect
percent complete: 100%
CREATING SUPPORT FILES
ELEVATION PRODUCTS for mapset [PERMANENT] in [latlon]
min computed aspect 0.0000 max computed aspect 360.0000
Color table for [SRTM3CGIAR_Vietnam_CH_aspect] set to aspect
ASPECT [SRTM3CGIAR_Vietnam_CH_aspect] COMPLETE
min computed slope 0.0000 max computed slope 79.1439
SLOPE [SRTM3CGIAR_Vietnam_CH_slope] COMPLETE
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work >
```

7.3.2 Create local incident angle image using the slope and aspect images






```
> r.mapcalc "incident_angle_palsar=sin(A)*cos(B)*sin(I)*cos( $\psi$ ) +  
sin(A)*sin(B)*sin(I)*sin( $\psi$ ) + cos(A)*cos(I)"
```

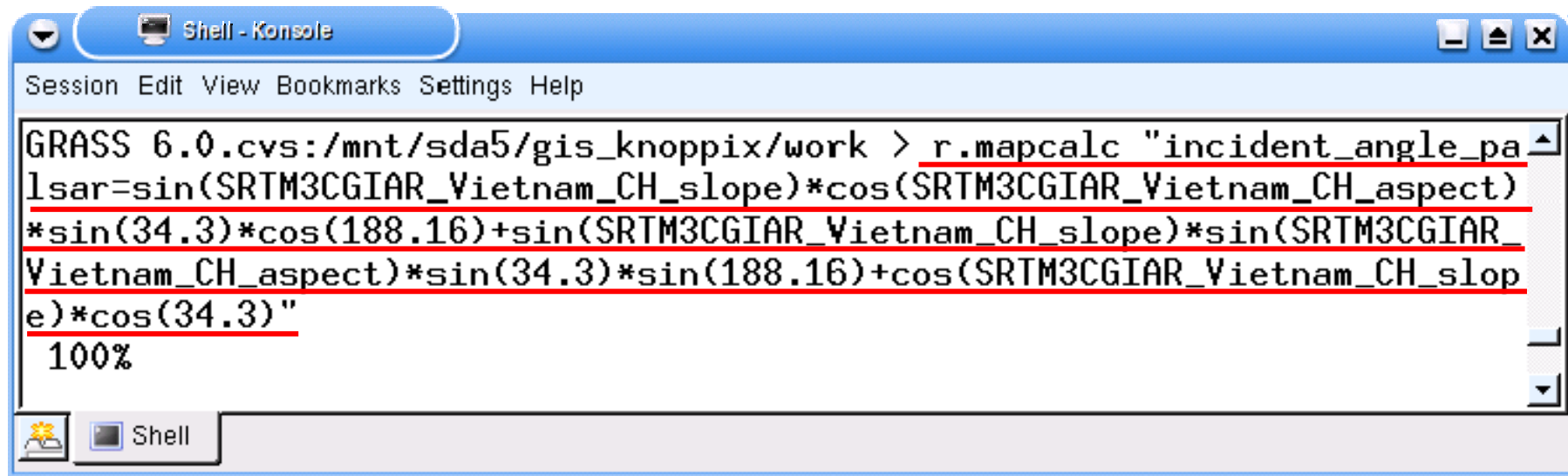
A = SRTM3CGIAR_Vietnam_CH_slope

B = SRTM3CGIAR_Vietnam_CH_aspect

I = 34.3

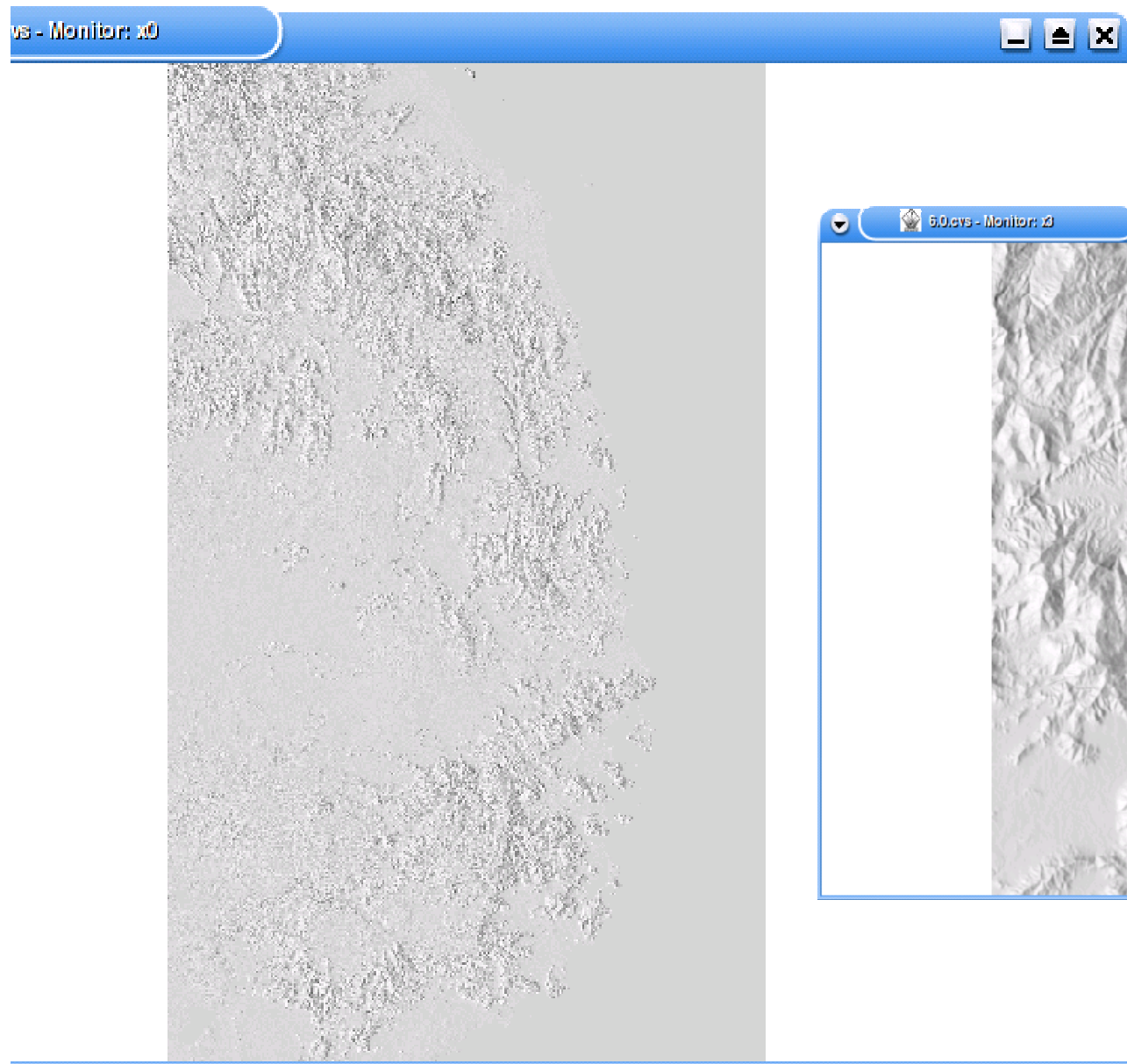
ψ = 188.16

␣ : space



The screenshot shows a terminal window titled "Shell - Konsole". The command prompt is "GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work >". The command being executed is `r.mapcalc "incident_angle_palsar=sin(SRTM3CGIAR_Vietnam_CH_slope)*cos(SRTM3CGIAR_Vietnam_CH_aspect)*sin(34.3)*cos(188.16)+sin(SRTM3CGIAR_Vietnam_CH_slope)*sin(SRTM3CGIAR_Vietnam_CH_aspect)*sin(34.3)*sin(188.16)+cos(SRTM3CGIAR_Vietnam_CH_slope)*cos(34.3)"`. The command is wrapped across several lines. Below the command, the output "100%" is displayed. The window has a menu bar with "Session", "Edit", "View", "Bookmarks", "Settings", and "Help". At the bottom, there is a taskbar with a "Shell" icon.

```
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.mapcalc "incident_angle_palsar=sin(SRTM3CGIAR_Vietnam_CH_slope)*cos(SRTM3CGIAR_Vietnam_CH_aspect)*sin(34.3)*cos(188.16)+sin(SRTM3CGIAR_Vietnam_CH_slope)*sin(SRTM3CGIAR_Vietnam_CH_aspect)*sin(34.3)*sin(188.16)+cos(SRTM3CGIAR_Vietnam_CH_slope)*cos(34.3)"  
100%
```



Zoom

7.3.3 Correct terrain effects of PALSAR image using local incident image

```
> r.mapcalc "P2007_Vietnam_CH_100m_HH_TC = 1.0*(P2007_Vietnam_CH_100m_HH) / ((incident_angle_palsar)^2+(1-cos(34.3)^2)"
```

□ : space

```
> r.mapcalc "P2007_Vietnam_CH_100m_HV_TC = 1.0*(P2007_Vietnam_CH_100m_HV) / ((incident_angle_palsar)^2+(1-cos(34.3)^2)"
```

□ : space

Terrain correction algorithm :

$$Rc = R / (\cos^2 \theta + (1 - \cos^2 34.3))$$

Rc : the corected value of R

R : SAR data value

θ : local incident angle

A. Terrain correction of PALSAR HH image

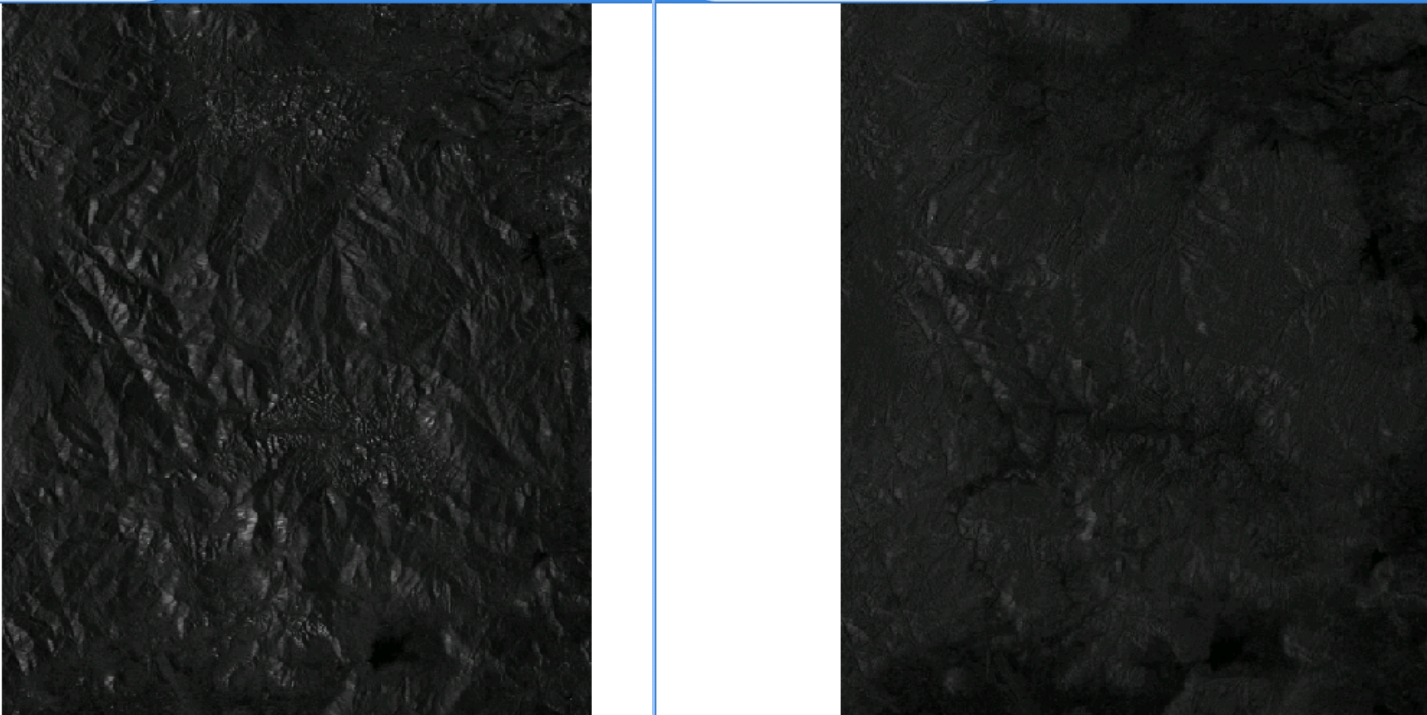
Shell - Konsole

Session Edit View Bookmarks Settings Help

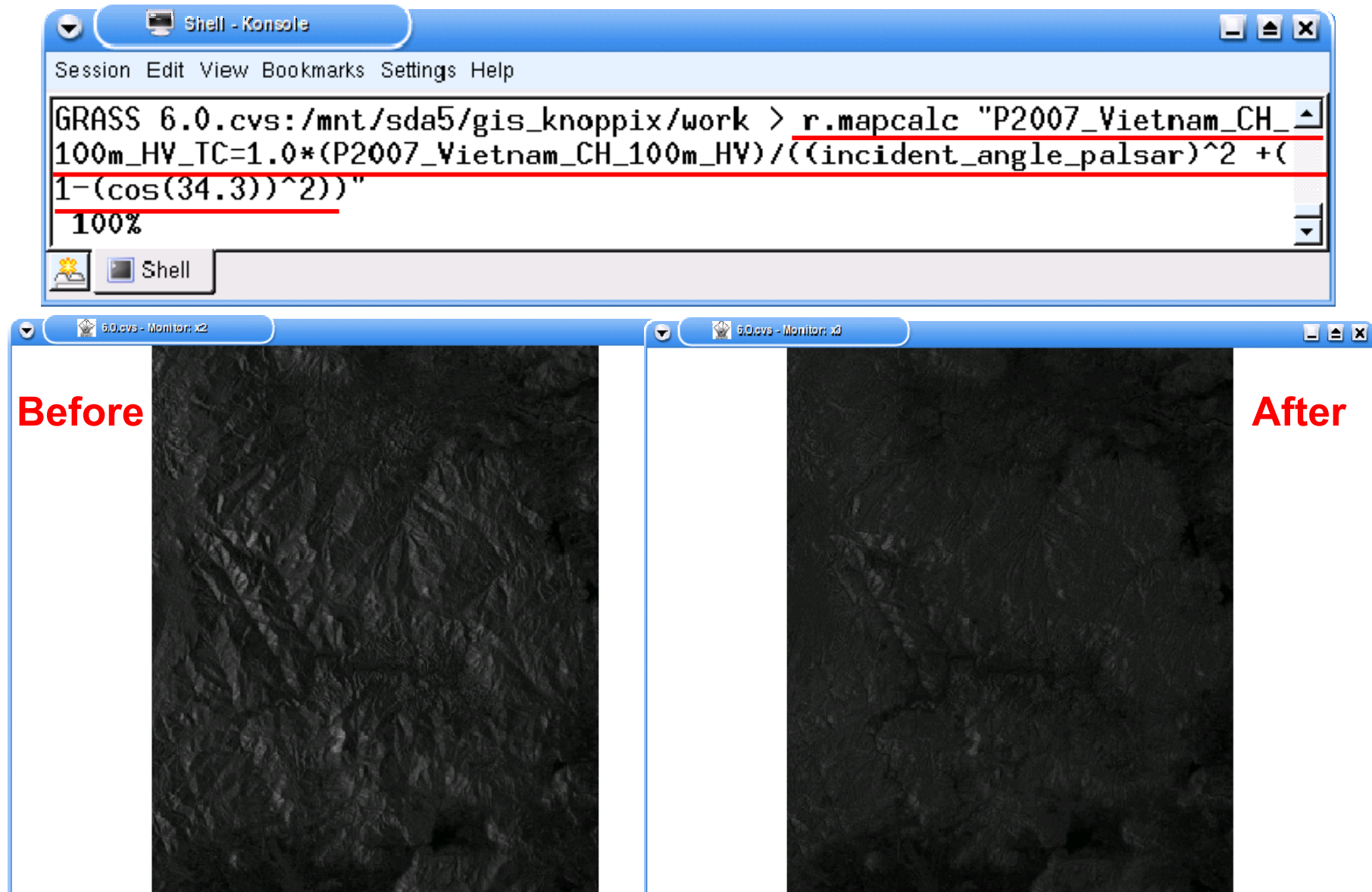
```
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.mapcalc "P2007_Vietnam_CH_100m_HH_TC=1.0*(P2007_Vietnam_CH_100m_HH)/((incident_angle_palsar)^2 +(1-(cos(34.3))^2))"  
100%
```

Before

After



B. Terrain correction of PALSAR HV image



7.4 Terrain correction of JERS-1 SAR image

Step 0. Import DEM image

\Rightarrow *see* “ 3.2 ”

Step 1. Create slope and aspect images using DEM

\Rightarrow *see* “ 7.3.1 ”

Step 2. Create local incident angle image
using the slope and aspect images

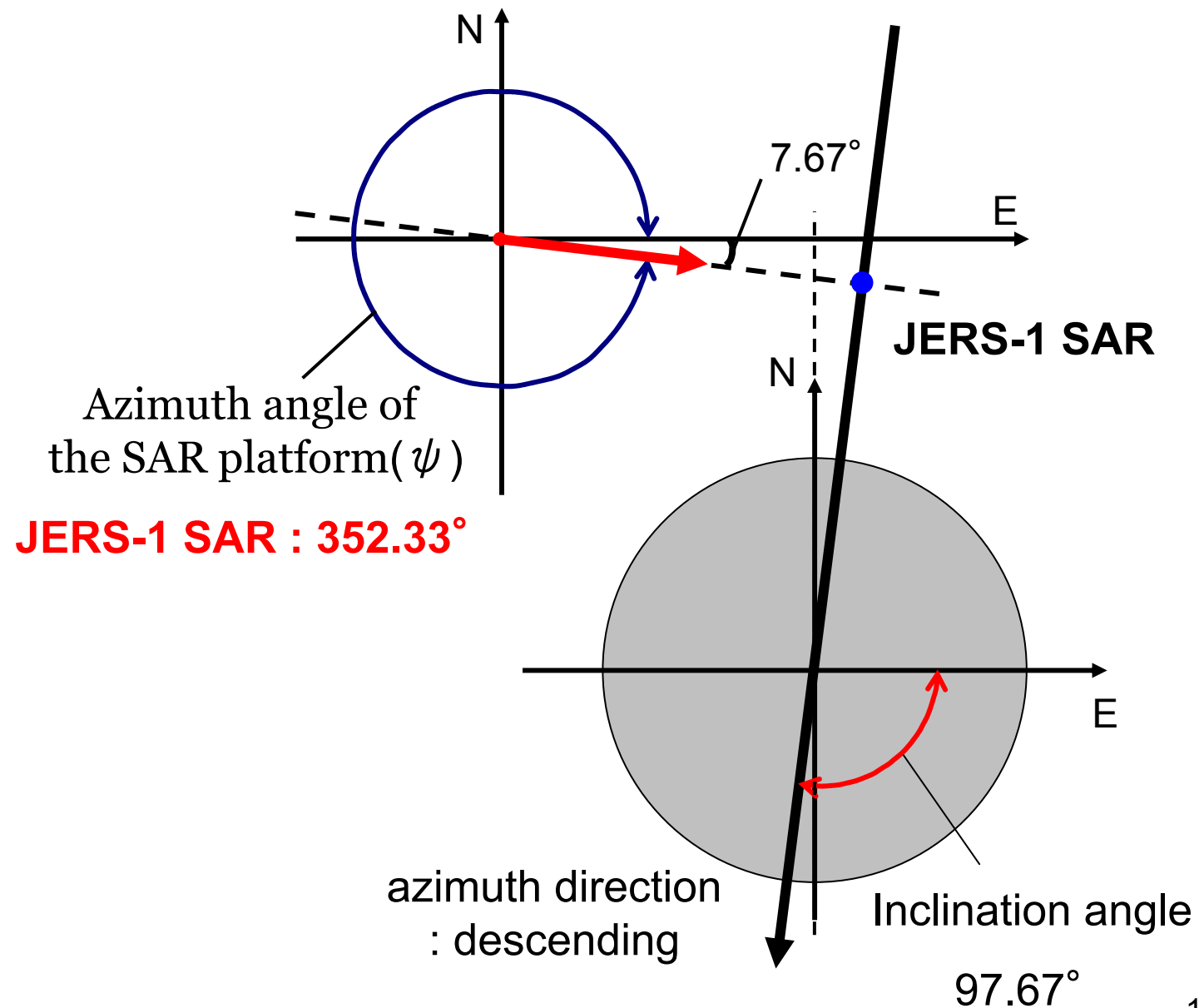
\Rightarrow *see* “ 7.2 ”

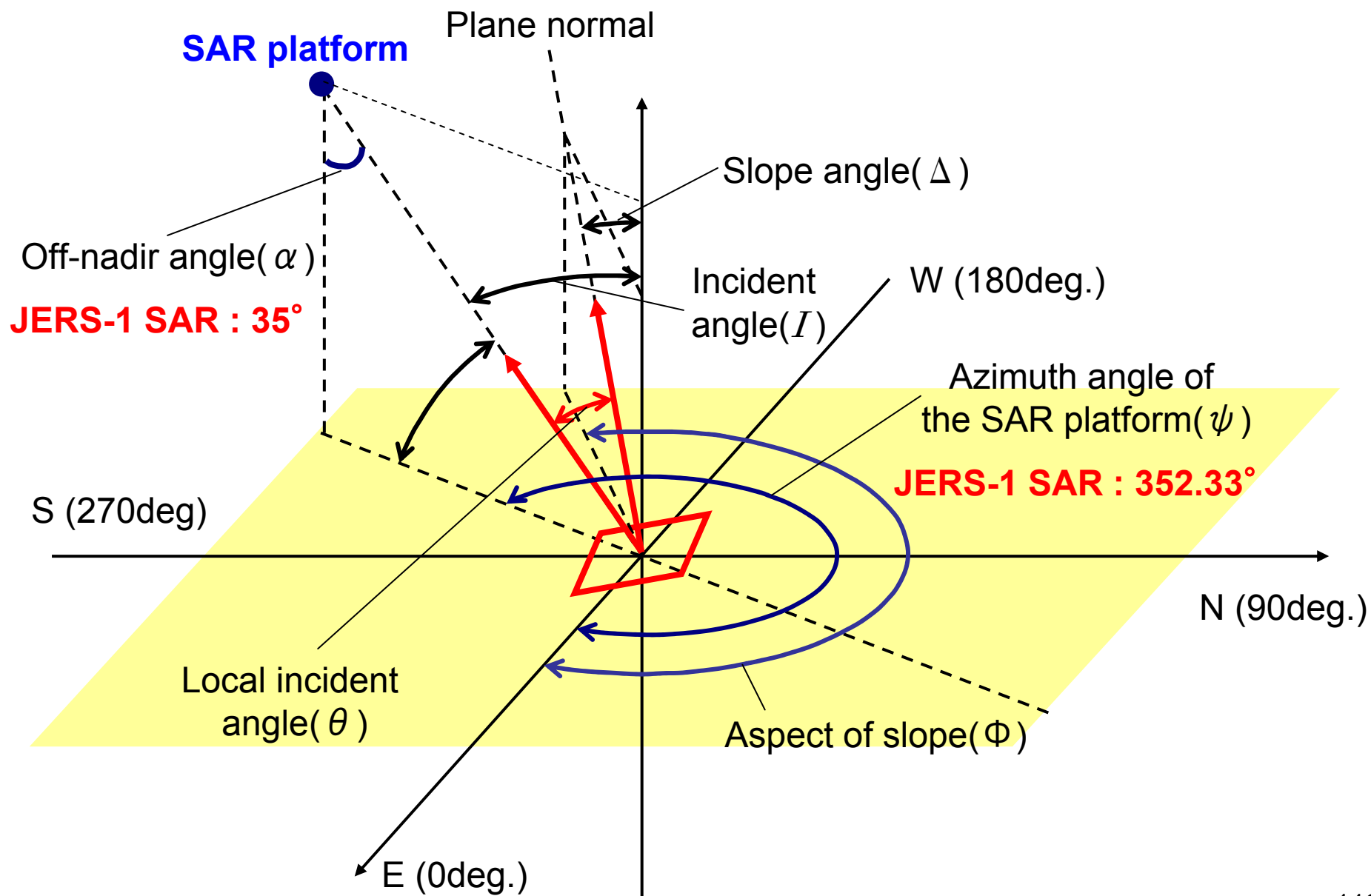
$$I = 35^\circ$$

$$\psi = 352.33^\circ$$

Step 3. Correct terrain effects of JERS-1 SAR image
using local incident image

7.4.1 Create local incident angle image using the slope and aspect images






```
> r.mapcalc "incident_angle_jsar=sin(A)*cos(B)*sin(I)*cos( $\psi$ ) +  
sin(A)*sin(B)*sin(I)*sin( $\psi$ ) + cos(A)*cos(I)"
```

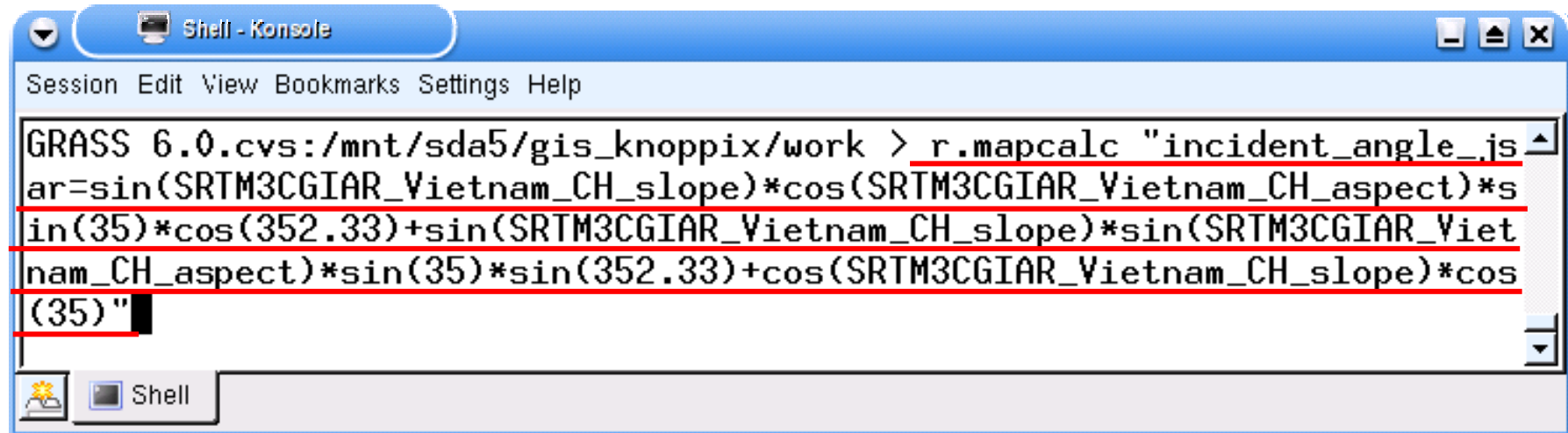
A = SRTM3CGIAR_Vietnam_slope

B = SRTM3CGIAR_Vietnam_aspect

I = 35

ψ = 352.33

␣ : space



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 text shows the command being executed in a GRASS 6.0 environment. The command is: `r.mapcalc "incident_angle_jsar=sin(SRTM3CGIAR_Vietnam_CH_slope)*cos(SRTM3CGIAR_Vietnam_CH_aspect)*sin(35)*cos(352.33)+sin(SRTM3CGIAR_Vietnam_CH_slope)*sin(SRTM3CGIAR_Vietnam_CH_aspect)*sin(35)*sin(352.33)+cos(SRTM3CGIAR_Vietnam_CH_slope)*cos(35)"`. The command is split across four lines in the image, with red underlines highlighting each line. The cursor is at the end of the fourth line.

```
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.mapcalc "incident_angle_jsar=sin(SRTM3CGIAR_Vietnam_CH_slope)*cos(SRTM3CGIAR_Vietnam_CH_aspect)*sin(35)*cos(352.33)+sin(SRTM3CGIAR_Vietnam_CH_slope)*sin(SRTM3CGIAR_Vietnam_CH_aspect)*sin(35)*sin(352.33)+cos(SRTM3CGIAR_Vietnam_CH_slope)*cos(35)"
```

7.4.2 Correct terrain effects of JERS-1 SAR image using local incident image

```
> r.mapcalc "JSAR1998Aug_Vietnam_CH_100m_ortho_TC = 1.0*(JSAR1997Aug_Vietnam_CH_100m_ortho) / ((incident_angle_jsar)^2+(1-cos(35)^2))"
```

□ : space

```
> r.mapcalc "JSAR1997JF_Vietnam_CH_100m_ortho_TC = 1.0*(JSAR1997JF_Vietnam_CH_100m) / ((incident_angle_jsar)^2+(1-cos(35)^2))"
```

□ : space

Terrain correction algorithm :

$$Rc = R / (\cos^2 \theta + (1 - \cos^2 35))$$

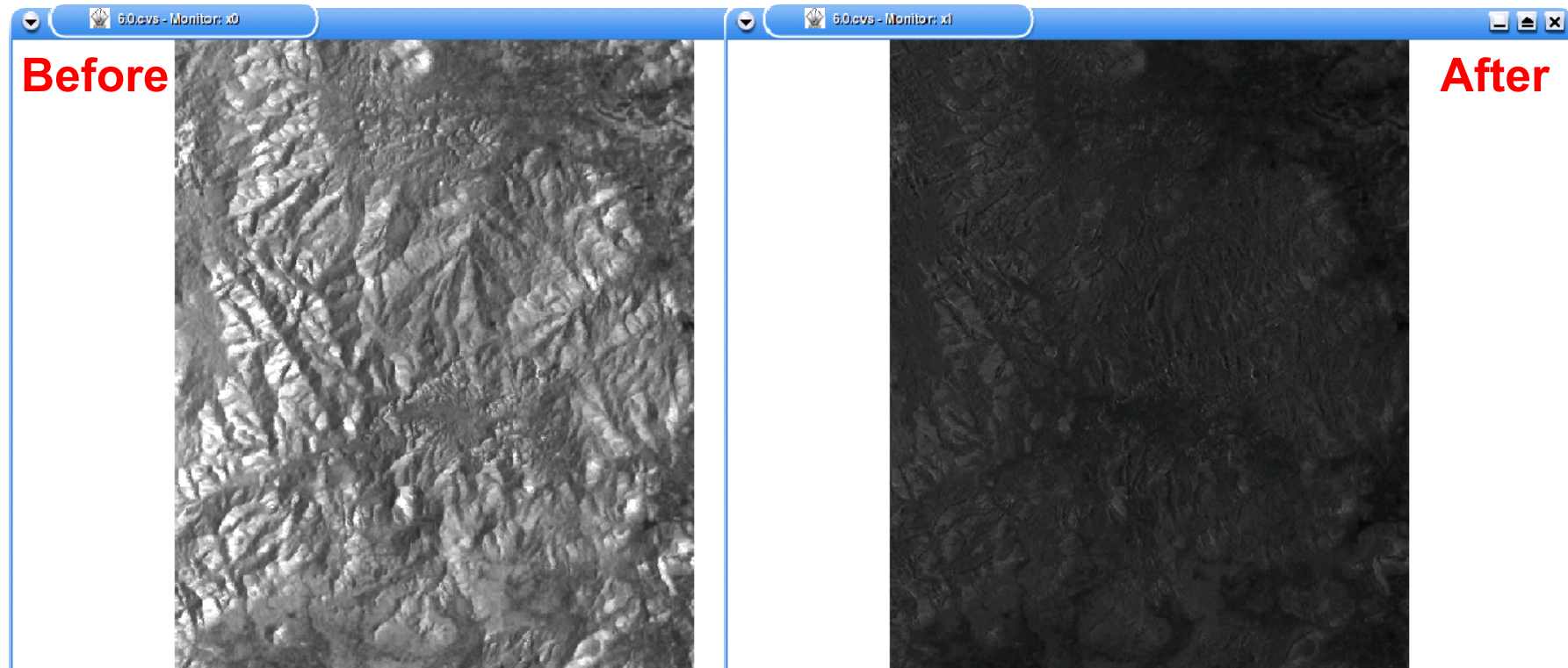
Rc : the corected value of R

R : SAR data value

θ : local incident angle

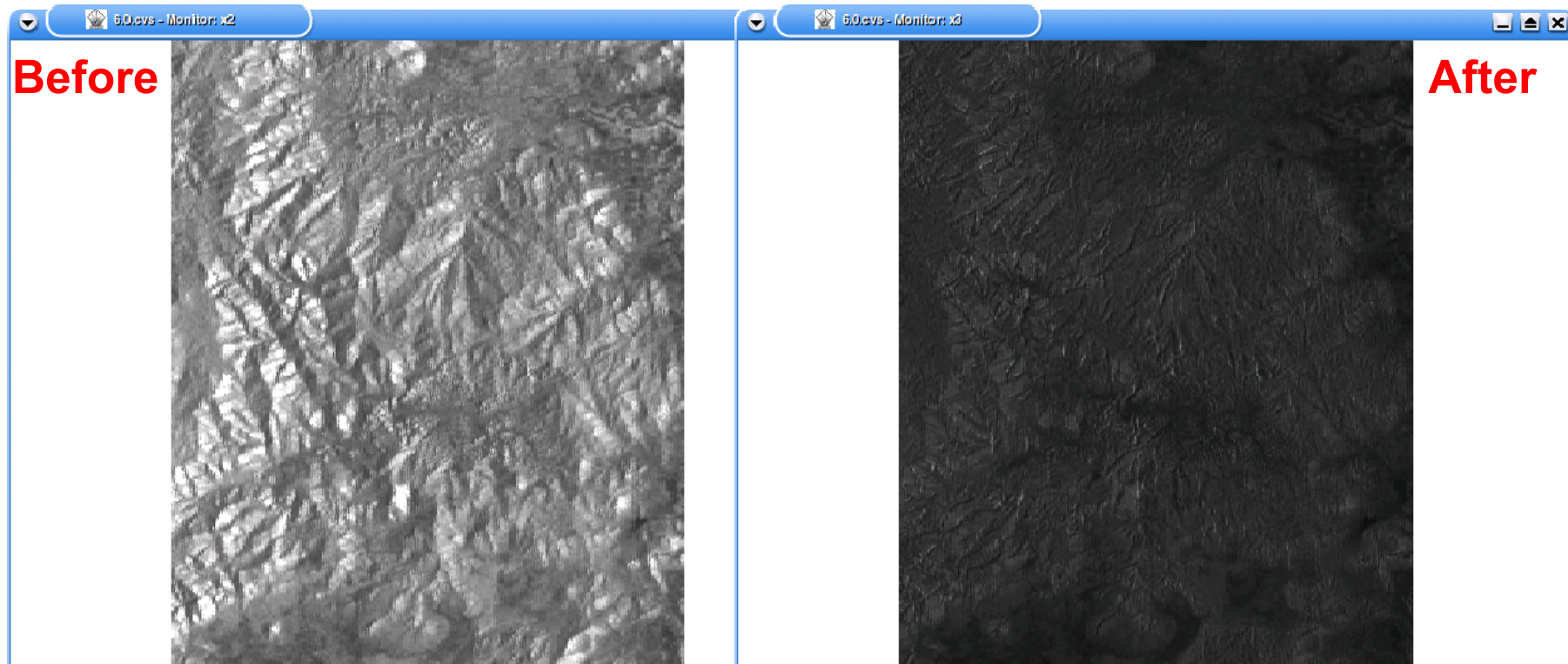
A. Terrain correction of JERS-1 SAR 1998 Aug. image

```
Shell - Konsole
Session Edit View Bookmarks Settings Help
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.mapcalc "JSAR1998Aug_Vietnam_CH_100m_ortho_TC=1.0*(JSAR1998Aug_Vietnam_CH_100m_ortho)/((incident_angle_jsar)^2+(1-(cos(35))^2))"
100%
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > █
```



A. Terrain correction of JERS-1 SAR 1997 Jan. & Feb. image

```
Shell - Konsole
Session Edit View Bookmarks Settings Help
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.mapcalc "JSAR1997JF_Vietna
m_CH_100m_ortho_TC=1.0*(JSAR1997JF_Vietnam_CH_100m_ortho)/((incident_an
gle_jsar)^2+(1-(cos(35))^2))"
100%
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > █
```



Section 2. SAR Image Processing in Latitude-Longitude Coordinate System

8. Extract deforested areas using RGB image

Step 0. Import PALSAR and JERS-1 SAR images

⇒ *see* “**3. Import data**”

Step 1. Correct terrain effects of PALSAR and JERS-1 SRA images

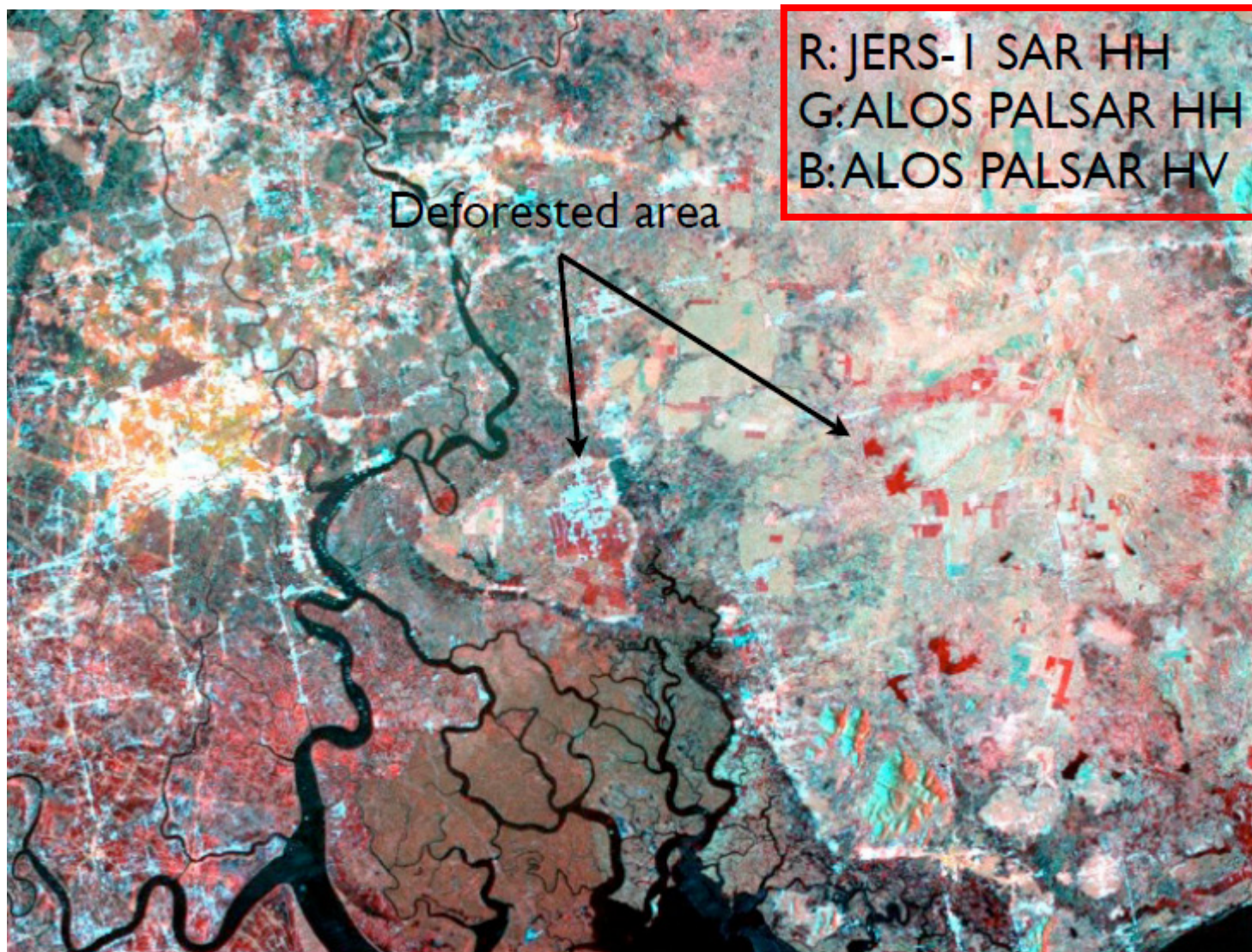
⇒ *see* “**7. Terrain Correction**”

Step 2. Change color table of terrain corrected images

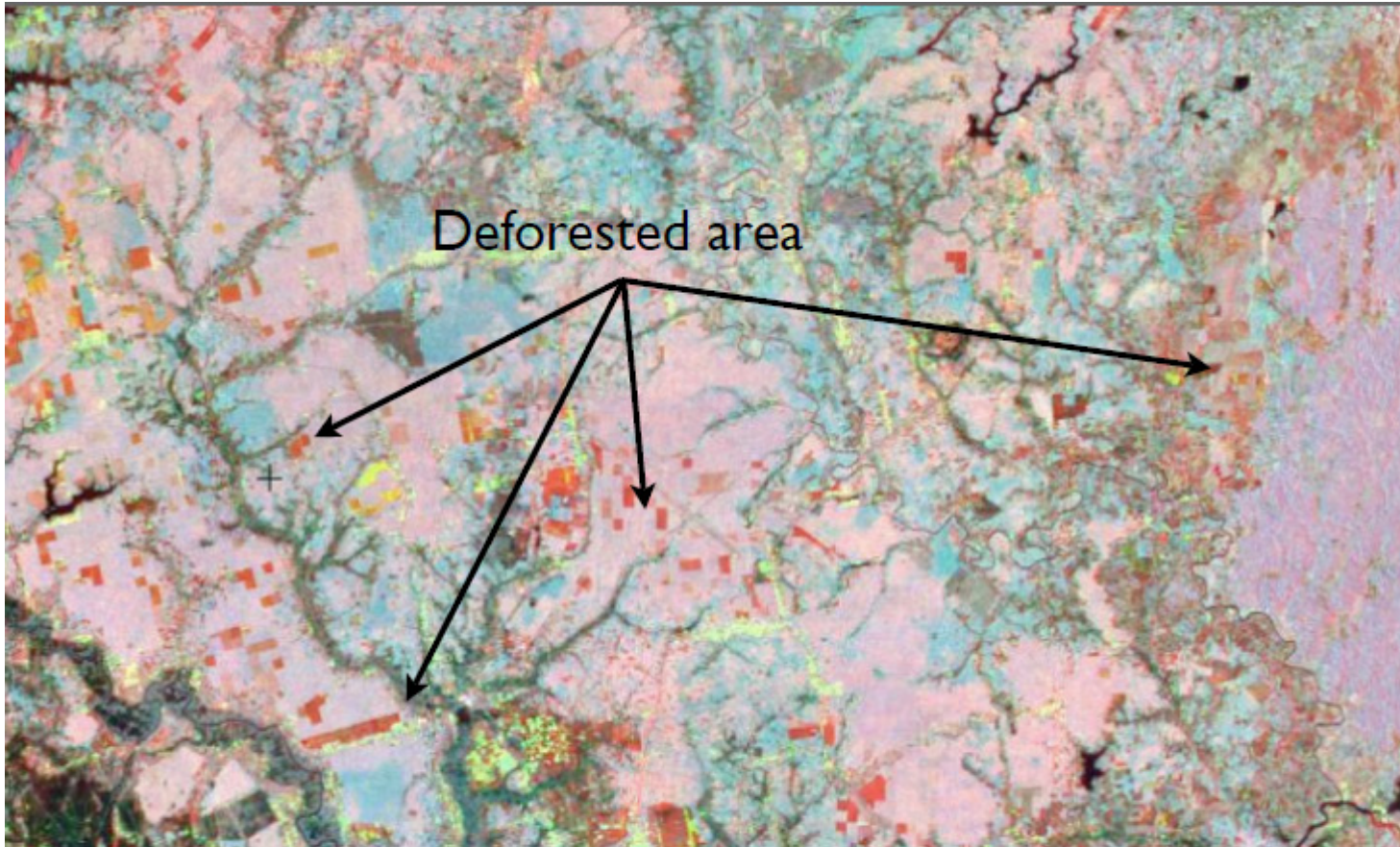
Step 3. Create RGB images using PALSAR and JERS-1 SRA images

Step 4. Export RGB image in TIFF format

8.1 Extract deforested areas using RGB image



[Takeuchi *et al.* 2008]



To extract deforested area, we use following combination :

R : JERS-1 SAR HH

G : PALSAR HH

B : PALSAR HV

8.2 Change color table of terrain corrected images

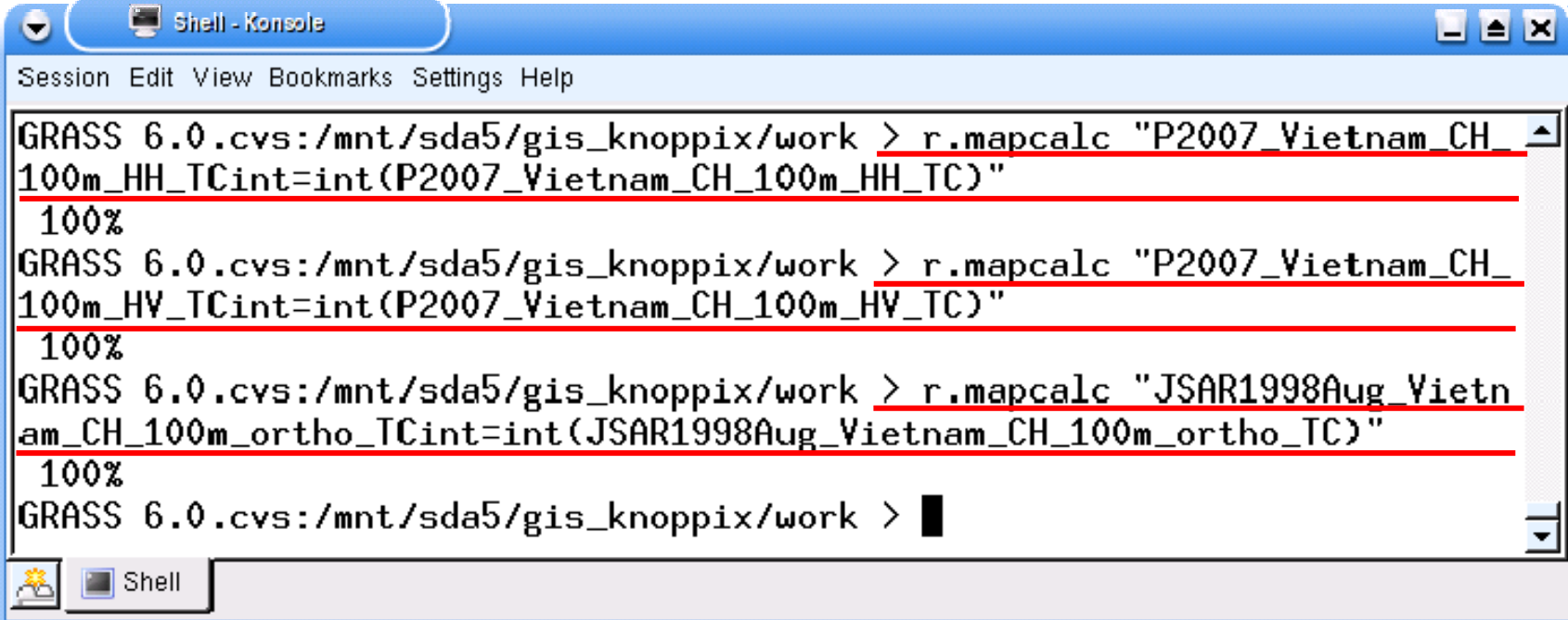
8.2.1 Change image values from floating point to integer

> r.mapcalc "A = int(B)"

A = output filename(integer)

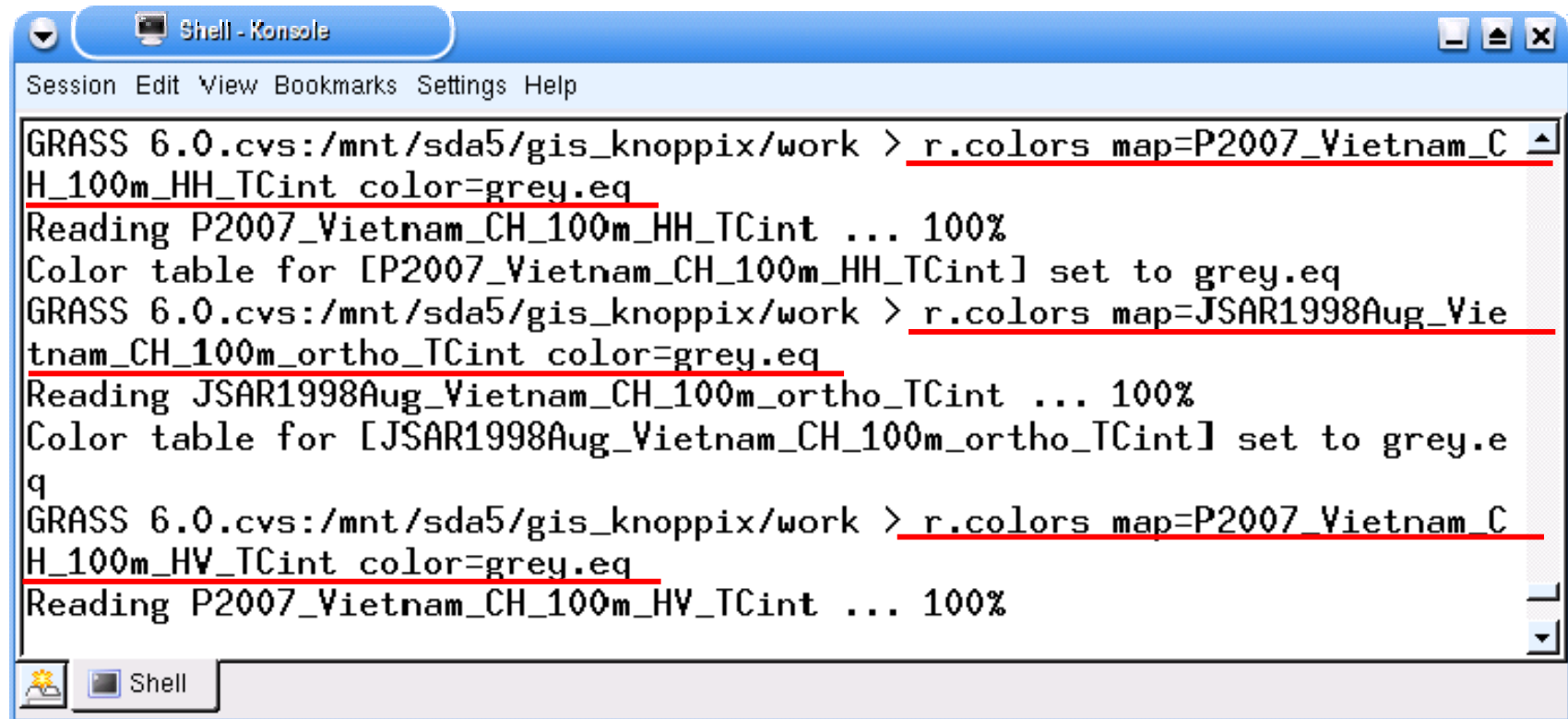
B = input filename(floating point)

: space



```
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.mapcalc "P2007_Vietnam_CH_100m_HH_TCint=int(P2007_Vietnam_CH_100m_HH_TC)"
100%
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.mapcalc "P2007_Vietnam_CH_100m_HV_TCint=int(P2007_Vietnam_CH_100m_HV_TC)"
100%
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.mapcalc "JSAR1998Aug_Vietnam_CH_100m_ortho_TCint=int(JSAR1998Aug_Vietnam_CH_100m_ortho_TC)"
100%
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > █
```

8.2.2 Change color table of terrain corrected image to histogram-equalized grey scale



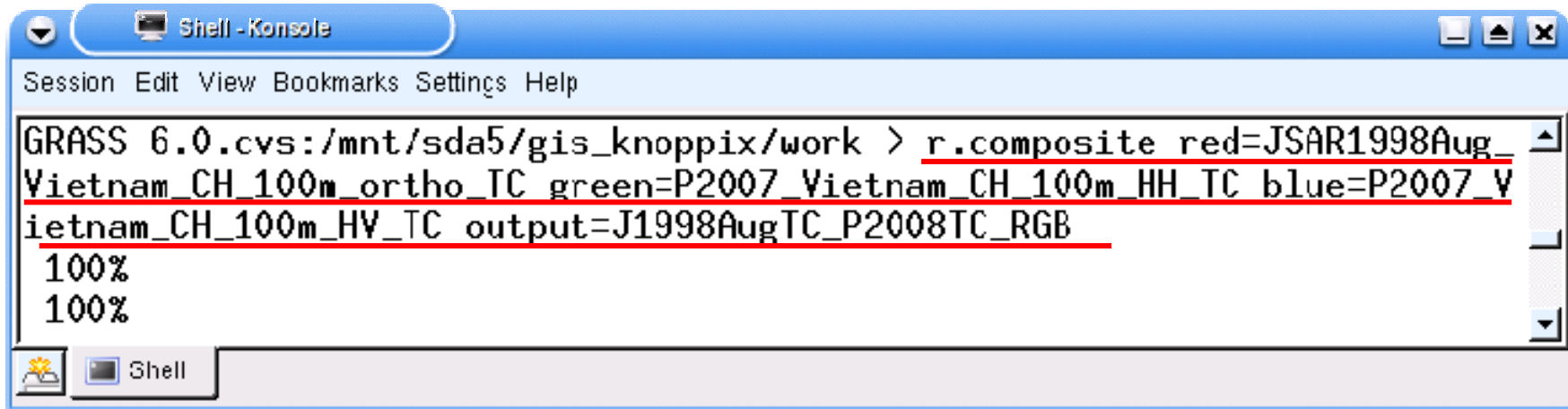
```
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.colors map=P2007_Vietnam_CH_100m_HH_TCint color=grey.eq  
Reading P2007_Vietnam_CH_100m_HH_TCint ... 100%  
Color table for [P2007_Vietnam_CH_100m_HH_TCint] set to grey.eq  
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.colors map=JSAR1998Aug_Vietnam_CH_100m_ortho_TCint color=grey.eq  
Reading JSAR1998Aug_Vietnam_CH_100m_ortho_TCint ... 100%  
Color table for [JSAR1998Aug_Vietnam_CH_100m_ortho_TCint] set to grey.eq  
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.colors map=P2007_Vietnam_CH_100m_HV_TCint color=grey.eq  
Reading P2007_Vietnam_CH_100m_HV_TCint ... 100%
```

8.3 RGB images using PALSAR and JERS-1 SRA images

R : JERS-1 SAR HH

G : PALSAR HH

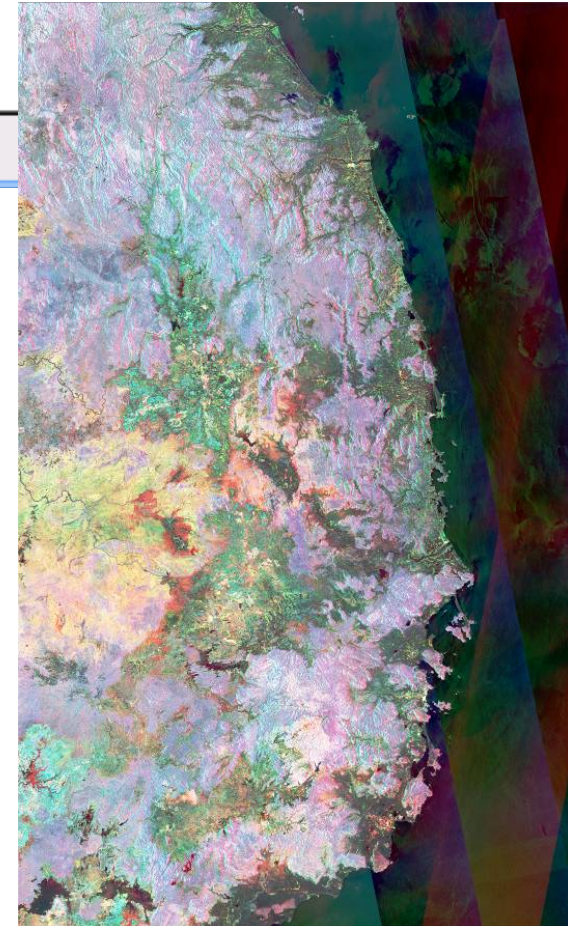
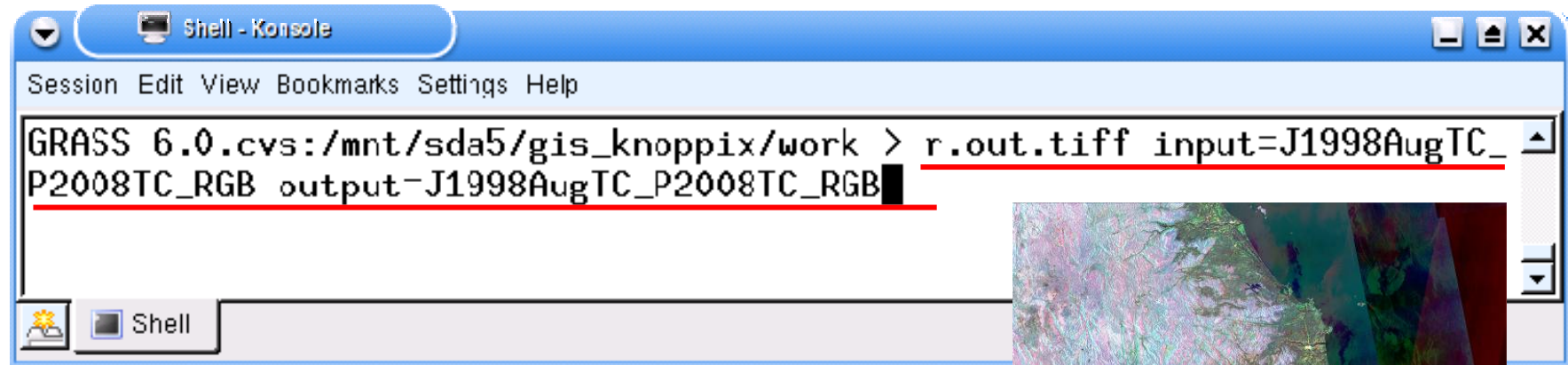
B : PALSAR HV

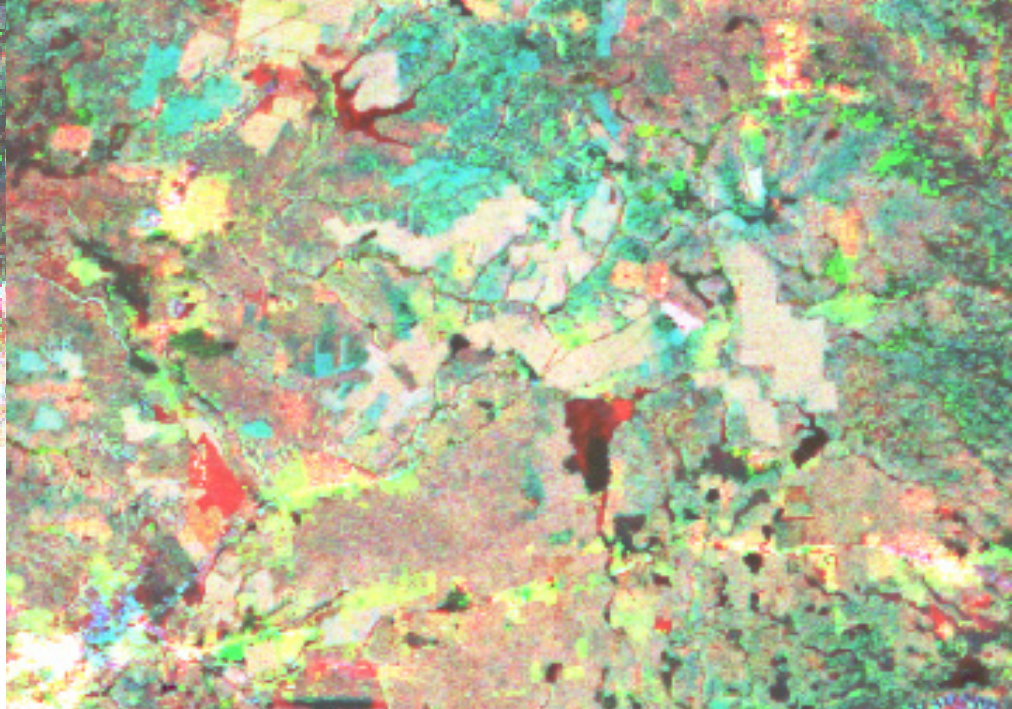
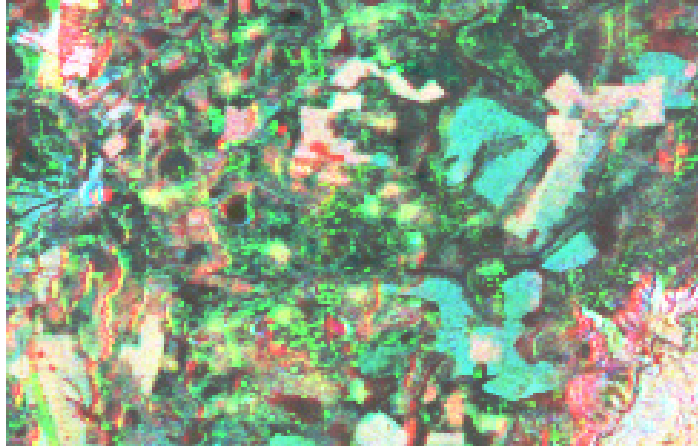
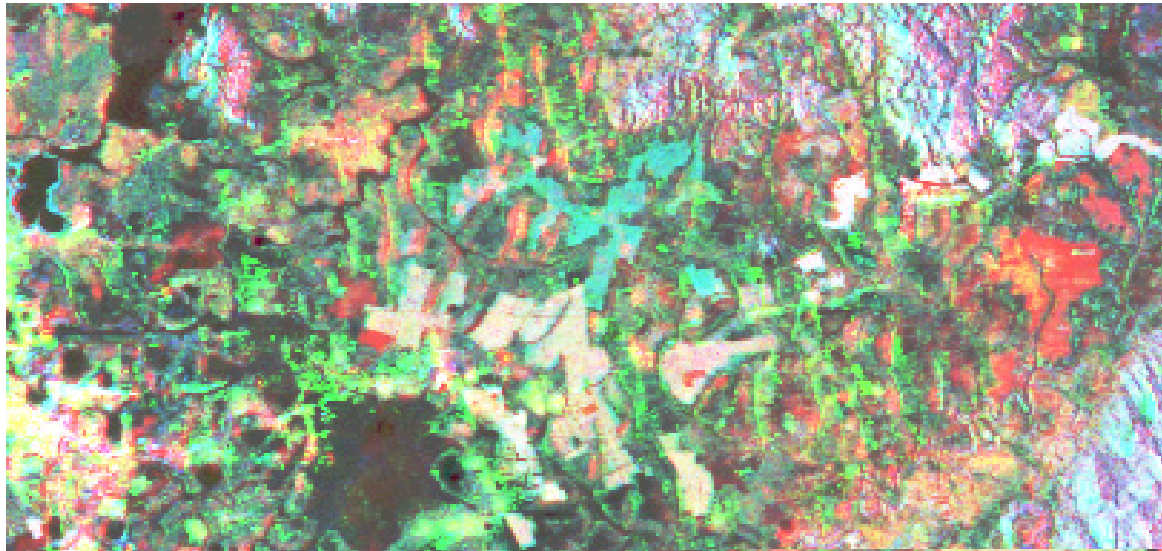


```
GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work > r.composite red=JSAR1998Aug_Vietnam_CH_100m_ortho_TC green=P2007_Vietnam_CH_100m_HH_TC blue=P2007_Vietnam_CH_100m_HV_TC output=J1998AugTC_P2008TC_RGB  
100%  
100%
```

The screenshot shows a terminal window titled "Shell - Konsole". The command prompt is "GRASS 6.0.cvs:/mnt/sda5/gis_knoppix/work >". The command entered is "r.composite red=JSAR1998Aug_Vietnam_CH_100m_ortho_TC green=P2007_Vietnam_CH_100m_HH_TC blue=P2007_Vietnam_CH_100m_HV_TC output=J1998AugTC_P2008TC_RGB". The command is underlined in red. The output shows "100%" on two lines.

8.4 Export RGB image in TIFF format





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/07-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=0&lg=en>
6. W. Takeuchi, P. Rakwatin, M. Shimada, and T. Nagano. Investigating the
performance of JERS-1 SAR and ALOS PALSAR mosaics for forest cover
mapping in Vietnam . The 15th Session of the Asia-Pacific Regional Space
Agency Forum in Hanoi, Vietnam. 2008.

Appendix A

GRASS command list

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

SYNOPSIS

r.in.bin [-sfdh] **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**=*name*
 color=*type*

Name of input raster map

Type of color table

Options: *aspect, grey, grey.eq, grey.log, byg, byr, gyr, rainbow, ramp, random, ryg, wave, rules*

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=*string*

Raster map name from which to copy color table

rules=*string*

Name of predefined rules file

Options: *aspect, terrain, gyr, slope, elevation, grey, ramp, bcy, evi, srtm, ryg, rainbow, wave, population, byr, ndvi, etopo2, byg*

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.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 [-1AacplqnNgxCri] **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.