Intro to Computing

Why N How
Nov 7, 2013

Jon Kaiser

Outline/Topics

- Shell (c-shell)
- Unix Commands
- Text Viewers
- Text Editors
- Redirection
- File Permissions
- Finding MRI (DICOM) data
- Organizing MRI (DICOM) data
- Scripting
- HELP!?!?

Background

: Shell :

What is it?
- command-line interpreter
- interaction with Unix OS
- extremely powerful and robust

How do I use it?
- Mac: Applications > Utilities > Terminal
- Windows: Download PuTTY (ssh client)

http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

- Linux (CentOS 6):
  - Applications > System Tools > Terminal
  - Right-click on Desktop > 'Open Terminal'

man

: Unix Commands :

Display help (manual) and list of options for any Unix commands.

man [options] <command>

man man
man ls

ls

: Unix Commands :

List directory contents

ls [options] [directory]

ls -l	long list
ls -a	all files, including hidden files
ls --color=auto	colorize the output
ls -r	in reverse order
ls -t	sort by modification time

New concepts (wildcards):

ls * and ls ?

mkdir

: Unix Commands :

Make a directory

mkdir [options] <directory>

mkdir -p make parent directories

cd and pwd

: Unix Commands :

change directory

cd [directory]

Examples:

```
cd Desktop
cd /cluster/scratch/
cd ..
cd ~kaiser/
cd
```

print working directory (pwd)

cp

: Unix Commands :

copy files and directories

cp [options] <source> <destination>

cp -r	recursive
cp -i	interactive
cp -u	update

mv

: Unix Commands :

move (rename) files and directories

mv [options] <source> <destination>

mv -i	interactive
mv -u	update

more

: Text Viewers :

Text file viewer

more [options] <file>

Useful keystrokes:

spacebar	scroll down one page
return	scroll down one line
/<string>	search for <string>
q	quit

less

: Text Viewers :

Text file viewer. More advanced than more

less [options] <file>

Useful keystrokes:

spacebar	scroll down one page
b	scroll up one page
return	scroll down one line
y	scroll up one line
/<string>	search for <string>
q	quit

cat

: Text Viewers :

Concatenate files to standard output (terminal)

cat [options] [file]

cat -n	number output lines
cat -s	skip repeated blank lines
cat -v	show hidden characters

Text Editors

: Text Editors :

gedit [file]	Simple GUI
emacs [file]	GUI + keyboard shortcuts
pico [file]	Simple text-based
vi [file]	Complex text-based

LibreOffice is open-source MS Office clone
Applications > Office

Writer	~=	Word
Calc	~=	Excel
Impress	~=	Powerpoint
Base	~=	Access

Redirection

: Redirection :

Most commands produce some type of output to the terminal:

echo “hello world”

The output is either classified as standard output (stdout) or standard error (stderr).

Standard Output

: Redirection :

Standard output can be captured (>). This is useful for creating logfiles.

Will overwrite anything in the file:

```
echo “hello world” > logfile.txt
```

Will append to the file:

```
echo “hello world” >> logfile.txt
```

Standard Error and Output

: Redirection :

Standard error (stderr) can also be merged with stdout (>&). Useful for creating logfiles.

```
ls ghost.txt > logfile.txt
ls ghost.txt >& logfile.txt
```

```
ls ghost.txt >> logfile.txt
ls ghost.txt >>& logfile.txt
```

Standard Error

: Redirection :

Standard error can also be captured on its own with a little creativity (and difficulty). This is useful for creating logfiles.

(ls ghost.txt > stdout.txt) >& stderr.txt

If you don't want/care about the standard output you can redirect it to /dev/null

(ls ghost.txt > /dev/null) >& stderr.txt

Pipe

: Redirection :

Allows for the redirection from one command to another. Stdout of the first command becomes the standard input for the next.

ls | wc -l

Hint: wc returns number counts (lines, words, chars). The '-l' option returns just the number of lines.

Can be as long and as complex as you want

Groups

: File Permissions :

Unix groups are composed of users who share something in common (lab members)

Your group membership:

groups

Other users:

groups <username>

List of users in a specific group:

ypmatch <groupname> group

Read, Write and eXecute

: File Permissions :

There are three types of permissions and can be combined in any combination.

r, w, x, rw, rx, wx, rwx

Access	File	Directory
Read	Read	Read the names of files
Write	Modify	Create, destroy
eXecute	Run a script/program	Access file contents

Users (owner) Groups and Other

: File Permissions :

Each file or directory is assigned to an owner and a group

Owner – the creator of the file or directory

Group – Each member of the group

Other – Everyone else

Permissions can be assigned separately for each class

ls -l

: File Permissions :

-rwxrw-r-- 1 kaiser sysadm 2521 Nov 6 20:07 sample.txt

Long list format displays metadata

There are seven fields:

1. 10 permission flags
2. Link count (ignore)
3. Owner (kaiser)
4. Group (sysadm)
5. Filesize
6. Modification Date
7. Filename

A closer look: Permission flags

: File Permissions :

-rwxrw-r--

Permission Flags:

1	File Type ID (files: -, directories: d)
2,3,4	Read, Write and eXecute for owner
5,6,7	Read and Write for group members
8,9,10	Read for others

chmod

: File Permissions :

Change file permissions. Only the owner can change them.

chmod [options] <mode> <file>

Options:

-R change permissions recursively

Recursively allow group members write access on all everything in my labs storage.

chmod -R g+w /cluster/mylab/

mode (symbolic notation)

: File Permissions :

chmod <mode> <file>
chmod u+x sample.txt

How does the mode work?
[class] [+,-,=] [perm]

class	U (owner), G (group), O (others), A (all)
+,-,=	allow, restrict, set
perm	R (read), W (write), X (execute)

Take away write permission for others:
chmod o-w sample.txt
Set read access for everyone and owner execute:
chmod a=r,u+x sample.txt

mode (octal notation)

: File Permissions :

chmod <mode> <file>
chmod 754 sample.txt

The mode consists of three digits, each representing a class of users: owner, group and others.

Each permission type is assigned a value:
Read = 4 Write = 2 Execute = 1
Sum the permission values to get the mode number

Give owner rwx, group rx, and others r
chmod 754 sample.txt
Give owner and group rw, and nothing for others
chmod 660 sample.txt

chgrp

: File Permissions :

Change group. Can only change the files of which you're the owner

chgrp [options] <group> <file>

Change the group designation for a file:
chgrp kaiserlab sample.txt

Editor Note: Use this command for files. To change the group of a directory and all it's sub-directories, it is best to use the script on the following slide.

setgrp

: File Permissions :

Recursively:

1. Changes group in given directory
2. Any newly created files will also have the assigned group
3. Allows group read and write permissions
4. Removes others write permission

/usr/pubsw/bin/setgrp [options] <group> <dir>

Options:

-o <user>	set owner
-p	remove others permission
-s	just change group, not permissions

findsession

: Finding MRI data :

Find images for a scan session at Martinos

findsession [options] <subject name>

-o yyyy-mm-dd	search on a given date
-e	exact match on subject
-l	search by SubjectID
-p	search by project
-x	search by experimenter
-h	help

nmrgateway

: Finding MRI (DICOM) data :

Pull images for scan at MGH (main campus)

https://gpfs07.nmr.mgh.harvard.edu/nmrgateway/search_by_mrn.html

- Must have approval from IRB to access and transfer images from MGH PACS.
- Must have /cluster/ space.
- Email Martinos Help Desk to request access and attach active IRB.

unpacksdcmdir

: Organizing MRI (DICOM) data :

Unpack DICOM data from a directory

Display list of runs from a scan session:

unpacksdcmdir -src /path/to/source/ -targ /path/to/target/ -fsfast -scanonly /path/to/target/output.txt

output.txt:

```
1          localizer  ok  512 512   3   1 6142643
2     SagMPRAGE8Min  ok  256 192 160   1 6142657
3     SagMPRAGE4Min  ok  256 256  60   1 6142077
4     T1ep2dHighRes  ok  256 256  21   1 6139499
5  ep2d_mosaic_mgh  ok   64  64  21 196 6133604
6  ep2d_mosaic_mgh  ok   64  64  21 196 6130701
7  T2TSEAxialHiRes  ok  512 512  21   1 6128022
```

Editor Note: dcmunpack has replaced this command. Many of the options remain the same. Run 'dcmunpack' to get help.

unpacksdcmdir

: Organizing MRI (DICOM) data :

Editor Note: dcmunpack has replaced this command. Many of the options remain the same. Run 'dcmunpack' to get help.

Unpack DICOM data from a directory

If you know which run you want and you don't need a summary file, convert a single run:

unpacksdcmdir -src /path/to/source/ -targ /path/to/target/ -generic -run 2 orig mgz 002.mgz

Finds all dicoms in the 2nd run, and converts them into an mgz file at this location: /path/to/target/orig/002.mgz

Good for FS recons

Scripting

: Scripting :

Scripting can be very powerful. Among other things:

- Run a series of commands sequentially
- Loop through a series of subjects

Every stand-alone script must begin with this line and be executable:

```
#! /bin/csh -f
chmod u+x script.csh
```

And typically they have the .csh suffix

C-Shell Script

: Scripting :

```
#! /bin/csh -f

set subj = $1
set source = $2

echo “Unpacking runs 2 and 3”

unpacksdcmdir -src $source -targ $SUBJECTS_DIR/
$subj/mri/ -generic -run  2 orig mgz 002.mgz -run 3
orig mgz 003.mgz

echo “Running recon all”

recon-all -s $subj -all
```

C-Shell Script

: Scripting :

The script will unpack runs 2 and 3 from your dicom directory and then recon all for the subject.

To run:

./script.csh <subj> </path/to/dicoms/>

When you include data on the command line, like <subj>, the value is available within the script as the variable $1.

```
set subj = $1
set source = $2
```

foreach loops

: Scripting :

foreach loops allow you to iterate over a set, or through a group (i.e. - subjects).

```
foreach <var> ( {set} )
   echo <var>
end
```

The first iteration will assign <var> to the first value in {set}.
The second iteration will assign <var> to the second value in {set}.
Until there are no more values in {set}.

foreach loops

: Scripting :

```
foreach i ( subject1 subject2 subject3 )
  echo “recon-all for $i”
  recon-all -s $i -all
end
```

Can be run in a stand-alone script, or directly in the terminal!

Other control structures

: Scripting :

```
if ( expression ) then
  # do something
else if ( expression ) then
  # do something else
else
  # last resort
endif
```

```
while ( expression )
  # do something
end
```

Resources

: HELP!?!? :

1. man page (for Unix commands)
2. The Google
3. http://help.nmr.mgh.harvard.edu
4. http://faq.nmr.mgh.harvard.edu
4. Ask your friends
5. Ask your enemies
6. email help@nmr

help@nmr

: HELP!?!? :

Answer these questions in your email

Who?
What?
Where?
When?
How?

The more information we have, the quicker

Thanks

: Fin :

If you have any questions, please ask them now. Don't email me later. Better yet, email help@nmr.