MRC | Cognition and Brain Sciences Unit

Linux primer

Jeff Berry

MRC Cognition and Brain Sciences Unit, Computing group

Motivation – Why do you need this?

- **Shared CBSU environment**
  - Central storage: images, scripts, documents, etc.
    - Backup (automatic)
  - High-Performance Computing: analysis
    - Speed (only if you know the “tricks”)

LINUX

- **Challenges/Tasks**
  - “Where are my documents?”, “Where are the tools?”, “Where am I?”
  - “Can I have this script/image?”
  - “Why cannot I open this document?”, “File I/O error???”
  - “How could I analyse 20 subjects in one go?”

- **Fun? → Efficiency**
  - Programming[1]

Basics

Unix Primer – Concept

- **File system**
  - Names (file and commands) are case sensitive
  - No file extensions
  - No drive letters, root = "/"
  - Directories can be network shares
    - */home, /imaging, /group, /mridata*

- home, Home, HOME?
- Well…
- "Where is drive U:?"

- **Permissions**[1]
  - User (owner)
  - Group
  - All

- "File I/O error!"
- "Am I in this group?"[2]

- **Path (*PATH*)**
  - Current folder is not assumed (unlike Windows or MATLAB)

- "But my script is here!"

Unix Primer – Desktop, here you are!

- **Applications**
  - Accessories:
    - gedit
    - Take Screenshot
  - System Tools
    - File Browser – **nautilus**
    - System Monitor
    - Terminal

- **Places**

Unix Primer – Terminal

- **Why?**
  - Faster and more powerful then GUI
    - No window drawing (But: You can call GUIs[1])
    - More functions (N.B.: GUIs are just wrappers)

  - Scripting: higher speed and flexibility
    - Wrappers
    - Batches → Parallelisation

Unix Primer – Terminal

- **Shell**
  - Encapsulated ("in a shell") environment
    - Execute commands (maximum as long as its "parent" terminal is open[1])
    - Variables[2]
    - Command history!

- **Command editor**
  - Command prompt
  - Interactive
  - Scrollable, editable, tab-completion

  - Keyboard short-cuts (a bit different for CSH and BASH):
    - History: Ctrl+R
    - Break: Ctrl+C

Unix Primer – Terminal

- **Structure of commands[1]:**
  - Executable [Arguments(s)]
    - [-/--]Option(s)
    - Input(s)
    - Output(s)
  - E.g[2].:

    fslview -h/--help

    Usage:[3]

    fslview [-m 3d|ortho|lightbox]
    [<baseimage>] [-l lutname] [-b low,hi] [-t transparency]
    [<overlay> [-l lutname] [-b low,hi] [-t transparency]] ...

    fslview -m ortho,lightbox swmStructural.nii spmT_0001.nii -l "Hot" -b 3,5 -t 0.5

Unix Primer – Useful Commands

Help/Info	***man, -h, --help***	Help
	man -k	find similar or related commands
	pwd	present working directory
	echo	display text or variable
Listing	*ls*	directory
	cat	file
	top	processes
	id -G -n <username>	list group memberships
Searching	*locate*	file
	which	executable
Navigate	*ssh*	compute machine
	cd	directory

Unix Primer – Useful Commands

Manipulate	*mkdir*	create directory
	cp (-r)	copy (directory)
	mv	move/rename
	rm (-r)	remove/delete irreversibly (directory)
	chown (-R)	change owner (directory)
	chmod (-R)	change permission (directory)
	vncserver	create/kill VNC session
	dos2unix, unix2dos	convert text file (EOL)
	wget <URL>	download
	tar -czf	Compress into .tar.gz
	tar -xzf	Extract from .tar.gz

Unix Primer – Useful Commands

Practice

- **1. “I have it somewhere…”**
  - Does *train12linux* have a *thresholds.txt* in his/her **imaging space**?
  - Can you read it? Why?
  - Can you write it? Why?

- **2. “Lets collaborate!”**
  - Op. 1.: Get a copy → Edit → Copy back
    - Can you do it? Why?
  - Op. 2.: Make it accessible
    - Parallel editing?

Advanced

Unix Primer – Make your life easier!

- **File browser - nautilus**
  - Create Bookmarks
  - Once a browser, always a browser

- **Symbolic links**
  - References to folders/files
  - Create: *ln -s <original> <link>*
  - Delete: *rm <link>*, *unlink <link>*

1. Search *pathdef.m*
2. Copy it to the top of your imaging space
3. Create a link called *link_to_pathdef.m*
4. Delete (new) *pathdef.m*
5. Open *link_to_pathdef.m* (*cat*, *gedit*)

- **Aliases:**
  - Short-cuts to command/script/program
  - Define: *alias* (with no arguments, lists existing aliases)
  - Un-define: *unalias*
  - E.g.:
    - *alias vnc 'vncserver :99 -geometry 1280x920'*
    - *alias matlab_my 'cd ~/Documents/MATLAB;/hpc-software/matlab/r2013a/bin/matlab'*

Unix Primer – Command-line

- **(Re)direction**
  - "|": pipe output to another command; e.g.: *ls /bin | more*
  - ">" and ">>": redirect output to a file
    - e.g.> *ls /bin > binlist*
  - "2>": redirect error (useful in scripts)
    - 2>: redirect error to a file; e.g.> *ls /bin /junk 2> errors*
    - 2>&1: redirect error to the output; e.g.> *ls /bin /junk > binlist 2>&1*
  - They can be combined into "mini-program":[1]
    - Practice!

Unix Primer – Command-line

- **Beginner:** “What is the maximum T-value in *spmT_0001.nii*?”
  - Save result into *maxval.txt*
  - Use *fslstats* and *awk '{print $2}'* !

    fslstats spmT_0001 -R | awk '{print $2}' >> maxval.txt

- **Intermediate:** “What TR has been used for *dicom.dcm*?”
  - Use *mri_probedicom*, *grep* and *awk '{print $3}'* !

    mri_probedicom --i dicom.dcm | grep TR | awk '{print $3}'

- **Advanced:** What sequence has been used for *dicom.dcm*?
  - Use *mri_probedicom*, *grep*, *awk '{print $2}'* and *sed 's/+AF8//g'*!

    mri_probedicom --i dicom.dcm | grep tProtocolName | awk '{print $3}' | sed 's/+AF8//g'

Unix Primer – Useful Commands (toolkit)

grep <filename>	Print lines with pattern fit
grep -c pttrn <filename>	Print number of lines with pattern fit
awk '{print $2}' <filename>	Print second entry only (of each line)
sed 's/ptrn1/ptrn2/' <filename>	Print content with ptrn1 replaced with ptrn2 (first instance only)
sed 's/ptrn1/ptrn2/g' <filename>	Print content with ptrn1 replaced with ptrn2 (globally)
printf "format" inpt	Formatted print (~ MATLAB)
echo "statement" | bc	Calculator
seq n	Sequence of integer up to n (useful for loops)

Unix Primer –Scripts

- **Definition:**
  - Collection of commands (in a structured way)

- **Login scripts**
  - */home/<username>/system.cshrc* - *CBSU-specific general .cshrc*
  - */home/<username>/.login* - *on login*
  - */home/<username>/.cshrc* - *on opening a shell*
    - Add your script storage folder to the PATH
    - Store aliases
    - Configure FSL, FreeSurfer

Scripting

Unix Primer – Scripts

E.g.: (.bashrc)

```
export PS1='\u@\h:\w\> ‘                                    # setup environmental variable
alias matlab_2013a='/hpc-software/matlab/r2013a/bin/matlab‘  # define alias
alias imagej='cd /home/ta02/Programs/ImageJ;./run‘          # define alias

if [ -d ${HOME}/bin ]; then                                 # start condition
        export PATH=${HOME}/bin:${PATH}                        # setup environmental variable
        for i in ${HOME}/bin/*; do                             # start finite loop (based on a list)
                if [ -d $i ]; then                                # start condition
                        export PATH=${i}:${PATH}                     # setup environmental variable
                fi                                                # end condition
        done                                                   # end loop
        . ${HOME}/bin/defaults.sh >> /dev/null                 # source a script discarding its output
fi                                                          # end condition

fsl_setup_noecho                                            # launch a script (already in the PATH)
freesurfer_setup_noecho                                     # launch a script (already in the PATH)
if [ -d ${HOME}/matlab ]; then rm -r ${HOME}/matlab; fi     # one-line condition block
```

Unix Primer – Scripts

- **Scripting "Languages"**
    - Any interpreter:
        - shells – */bin/sh, /bin/bash, /bin/tcsh, etc*
        - interpreted languages - *perl, python, etc*
- **Syntax**
    - Depends on the scripting language or shell used

Unix Primer – Scripts

- **Execution**
  - command line
    - execute with interpreter and script file
    - e.g. *bash test.sh, perl test.pl, python test.py, etc*
    - uses “parent” shell’s environment
  - embedded
    - interpreter in the first line of the header, with #! (use the full path!)
    - e.g. *#!/bin/bash, #!/usr/bin/perl*
    - must have execute permission on the file
    - execute with script file alone
    - e.g. *test.sh, test.pl, test.py, etc.*

Further Information

- **Linux:**
  - Basics:
    - http://imaging.mrc-cbu.cam.ac.uk/methods/unixsurvivalguide
    - http://www.ee.surrey.ac.uk/Teaching/Unix
  - Scripting:
    - http://tldp.org/LDP (both Beginner and Advanced)