University of Reading

UNIX PRIMER

A short introduction to the Unix operating system for novices

University of Reading

OBJECTIVE

- The objective of the course is to leave the student with enough knowledge of Unix commands to be able to
  - navigate the file structure
  - create directories and files
  - read files and modify them

University of Reading

COURSE OVERVIEW

- Using the Unix shell terminal
- Opening the terminal and logging in
- Where am I: the ***pwd*** command
- What files have I got: using the ***ls*** command and filtering
- Tips and tricks
- Creating a directory using the ***mkdir*** command
- Navigating the directory tree using the ***cd*** command
- The ***echo*** command
- Creating files using **>** and **>>** operators
- Copying a file or directories. The cp command
- Removing a file using ***rm***
- Reading a file using ***more*** and ***cat*** commands
- How to get out if it all goes wrong: ***ctrl&c***
- How to get help using ***man*** and switches

LOGGING IN

- Choose ‘MobaXterm’ from AppsAnywhere and click on ‘visit website’
- Click on the blue ‘Home Edition’ button
- Select ‘Open with Windows Explorer’, then double click on the .exe file and select ‘Run’ and MobaXterm should start
- Click ‘New Session’ and then in the ‘Remote host’ window enter cluster.act.rdg.ac.uk and specify your username
- Enter your password and you’ll see a message that you’re connected to the ’Reading Academic Computing Cluster’
- You’ll see your bash shell prompt, which looks something like this:

```
[username@racc-login-0-1 ~]$
```

WHERE AM I? - PWD

- 'pwd' is short for 'print working directory' (*not password!!!)*
- Type 'pwd' and press the 'return' key
- You will now see the directory where you are in the system. The output shows a path which is always your **home** directory when you first log in.

- Example:

```
[sxs98pmh@racc-login-0-1 ~]$ pwd
/home/users/sxs98pmh
```

University of Reading

WHAT FILES DO I HAVE? - LS

- 'ls' is short for 'list contents'
- Type 'ls' and press the 'return' key
- You will now see the contents of your home directory
- Example:

```
[sxs98pmh@racc-login-0-1 ~]$ ls
Desktop        Music      Videos
Documents      Pictures   test.txt
Downloads      Public
```

University of Reading

WHAT FILES DO I HAVE? - LS

- If you want more information about your files, such as size and permissions, use the '-l' switch
- Example:

```
[sxs98pmh@racc-login-0-1 ~]$ ls -l
total 7
drwxr-xr-x 2 sxs98pmh sxs     10 Feb  4 09:20 Desktop
drwxr-xr-x 3 sxs98pmh sxs      6 Nov  7 14:48 Documents
drwxr-xr-x 2 sxs98pmh sxs      2 Jun 26  2018 Downloads
drwxr-xr-x 2 sxs98pmh sxs      2 Jun 26  2018 Pictures
drwxr-xr-x 2 sxs98pmh sxs      2 Jun 26  2018 Public
drwxr-xr-x 2 sxs98pmh sxs      2 Jun 26  2018 Videos
drwxr-xr-x 2 sxs98pmh sxs      2 Jun 26  2018 test.txt
```

University of Reading

TIPS AND TRICKS

- Example:

  `[sxs98pmh@racc-login-0-1 ~]$ ls my_poems/poem1.txt`

  Press ‘Return’ to execute the above command

  Then the ‘up arrow’ to recall the command again

  Then the ‘home’ key

  Now delete the ‘ls’ and replace with ‘cp’ (copy) so it looks like this:

  `$ cp my_poems/poem1.txt my_poems/poem12.txt`

- You’ve now copied poem1.txt to poem12.txt

University of Reading

MORE TIPS AND TRICKS

- Another aid to speedy bashing is to copy and paste to the command line.
- To save using the mouse you can press ‘shift + insert’ keys together

- Bash features an auto-complete: start typing the name or path for a file then press the ‘Tab’ key. If only one possibility exists then bash will complete it in the command line. If many possibilities exists bash will show you the various options.
- Try this: ‘ls /v(tab) w(tab) h(tab)’
- You should now have a line that looks like this: ‘ls /var/www/html/’

CREATING A DIRECTORY - MKDIR

- To create a directory in the current folder simply type ‘mkdir directory-name’

- Example:

```
[sxs98pmh@racc-login-0-1 ~]$ mkdir my_poems
[sxs98pmh@racc-login-0-1 ~]$ ls
Desktop        Music       Videos
Documents      my_poems    Public
Downloads      Pictures    test.txt
```

University of Reading

NAVIGATING DIRECTORIES - CD

- 'cd' is short for 'change directory'
- To navigate into the 'my_poems' directory type 'cd my_poems'
- Remember Unix is case sensitive!
- The prompt shows you that you are in the 'my_poems' folder:

```
[sxs98pmh@racc-login-0-1 my_poems]$
```

- Of course there are no files in this folder yet!
- To return to your home directory type 'cd ~'. This returns you to your home directory from any location on the system.
- To go up one directory type 'cd ..'
- Return to your home directory again.

University of Reading

THE ECHO COMMAND

- The ‘echo’ command is a very simple but useful tool: all it does is output its argument to the screen.
- For the following sets of exercises, make sure you’re in your home directory.

- Example

```
$ echo “Mary had a little lamb”
Mary had a little lamb
```

University of Reading

CREATING FILES WITH '>'

- The > and >> operators are a shorthand way of creating and appending files respectively.
- To use them simply add them to the end of a command with a filename.
- We previously used the 'echo' command which outputs its arguments to the screen. We can *redirect* the output of echo into a file using '>'.

- Example:

```
$ echo "Mary had a little lamb" > my_poems/mary.txt
$ ls my_poems
mary.txt
```

University of Reading

READING FILES – MORE AND CAT

- The commands ‘more’ and ‘cat’ read a file and display it to the screen.
- ‘cat’ is short for ‘catalog’ and displays the entire file
- ‘more’ displays the file one page at a time

- Example:

```
[sxs98pmh@racc-login-0-1 ~]$ more my_poems/mary.txt
Mary had a little lamb
[sxs98pmh@racc-login-0-1 ~]$ cat my_poems/mary.txt
Mary had a little lamb
```

- The difference here is that for large files ‘more’ will prompt you to show another page, whereas ‘cat’ will read the entire file to the screen.

University of Reading

APPENDING FILES WITH '>>'

- The append file operator '>>' is used in a similar way to the '>' operator.

- Example:

```
$ echo "Its fleece was white as snow" >> my_poems/mary.txt
$ ls my_poems
mary.txt
$ more my_poems/mary.txt
Mary had a little lamb
Its fleece was white as snow
```

University of Reading

COPYING FILES

- The basic form is: ‘cp –[switch] filename newfilename’

- Example:

```
[sxs98pmh@racc-login-0-1 ~]$ cp my_poems/mary.dcsv
my_poems/mary2.dcsv
```

- The command can also copy directories and files. For this use the –r switch which will recursively copy the directories and files within it.

- `cp -r my_poems my_poems2`

University of Reading

REMOVING FILES - RM

- It goes without saying that this is one of the more dangerous commands, but you may need to use it.
- The basic form is: ‘rm –[switch] [filename]’

- Example:

```
[sxs98pmh@racc-login-0-1 ~]$ rm my_poems/mary.dcsv
```

- You can of course use wildcards like ? and * , **but this is where it gets dangerous**: for example ‘rm –R *.*’ will remove everything from the current directory and below it!!! (and there is no recovery/undo)

University of Reading

FILTERING

- To see filtering in action let's create some dummy files!
- Example:

```
$ echo "poem1" > my_poems/poem1.txt
$ echo "poem2" > my_poems/poem2.txt
$ echo "poem3" > my_poems/poem3.txt
$ echo "poem1" > my_poems/poem1.doc
$ echo "poem1" > my_poems/poem1.dcsv
$ echo "poem1" > my_poems/poem1.pdf
```

- Check if files are there with '`ls my_poems`'

University of Reading

FILTERING

- If you have many files and subdirectories you'll probably want to filter the results to only show the files/folders you are interested in.
- In Unix there are two 'wildcard' operators: '*' and '?'
- The asterisk '*' represents any set of characters, and the placeholder '?' is any character in a certain position.
- For example, to show only text files one could type: 'ls my_poems/*.txt'
- To show only files called 'poem', try: 'ls my_poems/poem.*'
- For several files called poem1.txt, poem2.txt, poem3.txt etc one could use '?' to filter for these: 'ls my_poems/poem?.txt'
- Try all 3 of the above commands out for yourselves.

University of Reading

MISSION ABORT – CTRL-C

- Sometimes you mistype a command or issue a command which takes a very long time or appears to hang.
- The way to get out of this is to press the ‘ctrl’ key and the ‘c’ key at the same time.
- You should then see the terminal prompt return.

HOW TO GET HELP - MAN

- All Unix commands come with on-board help which you can access via the 'man' command, which is short for 'manual'
- Let's try it for the 'ls' command:

```
[sxs98pmh@racc-login-0-1 ~]$ man ls
LS(1)
NAME
    ls - list directory contents
SYNOPSIS
    ls [OPTION]... [FILE]...
DESCRIPTION
    List information about the … … …
```

- Almost all inline help follows the above format.

HOW TO GET HELP - MAN

- The key points to realise are the syntax of the command and the options or ‘switches’

```
SYNOPSIS
     ls [OPTION]... [FILE]...
```

- Switches are sub commands which modify the output
- Earlier we used ‘ls –l’. Here we used the ‘-l’ switch to ask for the long format of list command
- For the file we’ve created, this translates to: ‘ls -l my_poems/mary.txt’
- Switches can be combined together e.g.: ‘ls –lr’, which means list in long format and in reverse order.

University of Reading

OTHER WAYS TO GET HELP

- As well as the ‘man’ command, nearly all Unix commands provide embedded help on their usage.
- Depending on the command, switches in the form ‘–h’, ‘--help’ and ‘-?’ can be appended to the command to show the commands’ usage.
- For example the help option for the ls command used earlier can be invoked by simply using ‘ls –help’
- The ‘more’ command help can be invoked by typing ‘more -?’
- The ‘man’ command usage can be invoked by typing ‘man –h’
- Invocation of help is not standardised, but one or other of the above will nearly always work.
- If all else fails, just use the ‘man’ command!

University of Reading

TIPS AND TRICKS

- Bash maintains a history of everything you typed, in a hidden file called .bash_history in your home directory.
- You can see this file if you type ‘ls -a .ba*’
- The dot in front of the filename represents a hidden file.
- Pressing the “up” arrow toggles through the previous command(s).
- This can be very useful if you want to list a file to see if it is there and then act upon it.

University of Reading

TIPS AND TRICKS: ALIASING

- If you use a command with a switch many times it can get a bit tedious.
- The 'ls –l' is often used many times.
- The bash shell allows you to 'alias' this command.
- To see how it works type: alias ll='ls –l'
- Now if you type 'll', this wil execute 'ls-l' and you will run the long form of the list command.

THE END!

- Happy Bashing! ☺

- Please give us your feedback by following this link below:

  https://forms.office.com/Pages/ResponsePage.aspx?id=xDv6T_zswEiQgPXkP_kOX7ArvOm3cbpHnixhCNWKRS9URTAwUlRRNFZCUE4yVlJXNVZHRENQR1JTMC4u