Shell Script

Outline

Things you should do are written in bold.

Suggested dialog is in normal text.

[Notes: are in italics and square brackets]

```
Command-line excerpts and code fragments are in shaded fixed-width
font.
```

Slide 4 - Intro

Q: Do intro question (slide 4) to judge if this module is needed and if so where to start.

Slide 5 – Grab files

Everyone should get a copy of the files and install in an appropriate directory

Mac – will download to Downloads and decompress. Copy folder to desktop. Open up a terminal window.

```
cd Desktop/swcNCL
```

Linux –

```
wget http://tinyurl.com/8aqwhea
unzip swcNCL.zip
cd swcNCL
```

Cigwin –
Install wget and unzip

```
wget http://tinyurl.com/8aqwhea
unzip swcNCL.zip
cd swcNCL
```

Slide 5 – What is the shell?

Just talk through text on slide

Slide 4 – File and directory commands

Run through these commands in a terminal students can follow along.

Tell me who I am

```
> who
nasm3    console  Oct 12 00:48
nasm3    ttys000  Oct 21 11:59
nasm3    ttys001  Oct 21 15:28
nasm3    ttys002  Oct 21 15:30
> whoami
nasm3    ttys002  Oct 21 15:30
```

[Note: under Cygwin who reports nothing whoami reports 'user']

What directory am I in?

```
> pwd
/Users/nasm3/Desktop/swcNCL
```

List files in this directory

```
> ls
Dissertation.txt      SC - Shell.ppt        myText.txt
ImportantWork         SC-ShellScript.docx   oldResults.dat
Private               Tests                 results.dat
Public                Work
Rubbish               clean
```

List only files that end .txt

```
> ls *.txt
Dissertation.txt      myText.txt
```

Indicate '/' at the end of directories

```
> ls *.txt
Dissertation.txt      SC - Shell.ppt        myText.txt
ImportantWork/        SC-ShellScript.docx   oldResults.dat
Private/              Tests/                results.dat
Public/               Work/
Rubbish/              clean
```

List hidden files

```
> ls -a
.                     Public                clean
..                    Rubbish               myText.txt
```

```
.DS_Store             SC - Shell.ppt        oldResults.dat
Dissertation.txt      SC-ShellScript.docx   results.dat
ImportantWork         Tests
Private               Work
```

Show how many blocks a file uses

```
> ls -a
total 2088
  16 Dissertation.txt         0 Tests
   0 ImportantWork            0 Work
   0 Private                 16 clean
   0 Public                  16 myText.txt
   0 Rubbish               1032 oldResults.dat
 416 SC - Shell.ppt         520 results.dat
  72 SC-ShellScript.docx
```

[Note: Block size is system dependant – often ~512 bytes]

Change to directory Work

```
> cd Work
```

[Note: Bash is highly case-sensitive]

Now shows contents of Work

```
> ls
DataDump              SoftwareCarpentry     thisWeeksWork.txt
DocumentsToRead       SourceCode
Papers                results.txt
```

Make a new directory called Results

```
> mkdir Results
```

List directory – now shows directory Results

```
> ls
DataDump              Results               results.txt
DocumentsToRead       SoftwareCarpentry     thisWeeksWork.txt
Papers                SourceCode
```

Change to directory Results

```
> cd Results
```

Edit the file info.txt (create as not there already) and enter the following text:
This directory holds the results of my first experiment in sleeping
I started this work on 22/10/2012

```
> nano info.txt
```

Copy info.txt to info-old.txt

```
> cp info.txt info-old.txt
```

Didn’t mean to copy, wanted to rename…

```
> rm info-old.txt
> mv info.txt info-old.txt
```

Output contents of file

```
> cat info-old.txt
```

Go back up

```
> cd ..
```

Copy entire directory

```
> cp -r Results Backup
```

Edit the file results.txt – add the last two lines

```
> nano results.txt
```

```
Monday 5 hours
Tuesday 4 hours
Wednesday 8 hours
Thursday 2 hours
Friday 10 hours
Saturday 3 hours
Sunday 11 hours
Monday 4 hours
Tuesday 5 hours
Wednesday 7 hours
Thursday 4 hours
Friday 6 hours
```

Sort the contents of the file results.txt

```
> sort results.txt
```

List the first few line of the file

```
> head results.txt
```

Only list the first two lines

```
> head -2 results.txt
```

List the last three lines

```
> tail -3 results.txt
```

Create a directory called Backup

```
> mkdir Backup
```

Delete this directory – must be empty

```
> rmdir Backup
```

How many things have I got on this week?

```
> cd Work
> wc -l thisWeeksWork.txt
       6 thisWeeksWork.txt
```

[Note: Normal output is Lines, Words, Characters]

I’m only interested in the number of words

```
> wc -w thisWeeksWork.txt
     20 thisWeeksWork.txt
```

Get all the statistics

```
> wc -w thisWeeksWork.txt
     6     20    138 thisWeeksWork.txt
```

What if we want to know how many words in a large number of files?

```
> wc -w thisWeeksWork.txt
      12      36      196 results.txt
       6      20      138 thisWeeksWork.txt
      18      56      334 total
```

Sort the results file alphabetically

```
> wc -w thisWeeksWork.txt
Friday 10 hours
Friday 6 hours
Monday 4 hours
Monday 5 hours
Saturday 3 hours
Sunday 11 hours
Thursday 2 hours
Thursday 4 hours
Tuesday 4 hours
Tuesday 5 hours
Wednesday 7 hours
Wednesday 8 hours
```

Slide 5 – Pipes and filters

How many files in a directory

```
> ls | wc -l
     7
```

How many files contain the letter S?

```
> ls | grep S | wc -l
     2
```

[Note: grep – searches for lines that contain the specified string]

User exercise: How many files contain the letter t?

```
> ls | grep t | wc -l
     5
```

What if we want to add a title line:

```
> echo The number of words in each file is: ; wc -l *.txt
The number of words in each file is:
      12 results.txt
```

```
       6 thisWeeksWork.txt
      18 total
```

Redirecting input

```
> head -4 < thisWeeksWork.txt
Monday - Software Carpentry
Tuesday - Software Carpentry
Wednesday - Meet students
Thursday - Visit Cambridge
```

Redirecting output

```
> head -4 thisWeeksWork.txt > first4.txt
> cat first4.txt
Monday - Software Carpentry
Tuesday - Software Carpentry
Wednesday - Meet students
Thursday - Visit Cambridge
```

Slide 6 – File Access

Who can read/write/execute a file?

```
> cd SourceCode
> ls –la
total 8
drwxr-xr-x@ 4 nasm3  staff  136 13 May 22:52 .
drwxr-xr-x@ 9 nasm3  staff  306 21 Oct 19:19 ..
-rw-------@ 1 nasm3  staff  263 13 May 22:51 calcPi.py
-rw-rw-rw-@ 1 nasm3  staff    0 13 May 18:44 myBigIdea.txt
```

[Note: explain what bits mean]

More readable version

```
> ls –lah
total 8
drwxr-xr-x@ 4 nasm3  staff   136B 13 May 22:52 .
drwxr-xr-x@ 9 nasm3  staff   306B 21 Oct 19:19 ..
-rw-------@ 1 nasm3  staff   263B 13 May 22:51 calcPi.py
-rw-rw-rw-@ 1 nasm3  staff     0B 13 May 18:44 myBigIdea.txt
```

Make it so anyone can read calcPi.py

```
> chmod a+r calcPi.py
> ls –lha
total 8
drwxr-xr-x@ 4 nasm3  staff   136B 13 May 22:52 .
drwxr-xr-x@ 9 nasm3  staff   306B 21 Oct 19:19 ..
-rw-------@ 1 nasm3  staff   263B 13 May 22:51 calcPi.py
-rw-rw-rw-@ 1 nasm3  staff     0B 13 May 18:44 myBigIdea.txt
```

Make it so anyone can execute calcPi.py

```
> chmod a+x calcPi.py
> ls –lha
total 8
drwxr-xr-x@ 4 nasm3  staff   136B 13 May 22:52 .
drwxr-xr-x@ 9 nasm3  staff   306B 21 Oct 19:19 ..
-rw-------@ 1 nasm3  staff   263B 13 May 22:51 calcPi.py
-rw-rw-rw-@ 1 nasm3  staff     0B 13 May 18:44 myBigIdea.txt
```

Could do this as one command:

```
> chmod a+rx calcPi.py
> ls –lha
total 8
drwxr-xr-x@ 4 nasm3  staff   136B 13 May 22:52 .
drwxr-xr-x@ 9 nasm3  staff   306B 21 Oct 19:19 ..
-rw-------@ 1 nasm3  staff   263B 13 May 22:51 calcPi.py
-rw-rw-rw-@ 1 nasm3  staff     0B 13 May 18:44 myBigIdea.txt
```

Slide 7 – Finding things

How do we find every occurrence of a string in a file?

```
> cd ../SoftwareCarpentry
> grep boot Blurb.txt
testing, and task automation. In this two-day boot camp, short tutorials will
sessions for 6 to 8 weeks extending the material from the boot camp.
boot camp.)
Content: The syllabus for this boot camp will include:
```

How many places does the word ‘the’ appear in the text?

```
> grep the Blurb.txt
to be informed of similar events in the future, please e-mail us on
teaching them basic computing skills like program design, version control,
both to help one another, and to apply what they have learned to their own
sessions for 6 to 8 weeks extending the material from the boot camp.
Who: The course is aimed at postgraduate students and other scientists who are
to help them work more productively.
packages installed. (The list will be sent to participants a week before the
        •      using the shell to do more in less time
in organizing further training at their own institution. By registering you
are stating that you will attend both days of the workshop and participate in
the online exercises and follow-up sessions in the following weeks. We will
notify all applicants as to whether they have a spot no later than April 13.
Contact: For further information please e-mail us at swc2012@ncl.ac.uk.
```

Though this picks up all other words such as then, them, their. To remove these use the –w option

```
> grep –w the Blurb.txt
to be informed of similar events in the future, please e-mail us on
sessions for 6 to 8 weeks extending the material from the boot camp.
packages installed. (The list will be sent to participants a week before the
       •      using the shell to do more in less timeare stating that you will
attend both days of the workshop and participate in
the online exercises and follow-up sessions in the following weeks. We will
```

How to match on multiple words. Find all occurances of ‘boot camp’

```
> grep ‘boot camp’ Blurb.txt
testing, and task automation. In this two-day boot camp, short tutorials will
sessions for 6 to 8 weeks extending the material from the boot camp.
boot camp.)
Content: The syllabus for this boot camp will include:
```

Which lines does the words ‘boot camp’ appear on?

```
> grep –n ‘boot camp’ Blurb.txt
13:testing, and task automation. In this two-day boot camp, short tutorials will
17:sessions for 6 to 8 weeks extending the material from the boot camp.
26:boot camp.)
28:Content: The syllabus for this boot camp will include:
```

Which lines don’t contain ‘boot camp’

```
> grep –v ‘boot camp’ Blurb.txt
Newcastle University / May 2012
April 7th, 2012
Leave a comment
Go to comments

This course is now fully booked and has a long waiting list. If you would like
to be informed of similar events in the future, please e-mail us on
swc2012@ncl.ac.uk.
When: May 14 – May 15, 2012. 9am to 5pm.
Where: Newcastle University, Newcastle upon Tyne
What: Our goal is to help scientists and engineers become more productive by

…

Requirements: Participants must bring a laptop with a few specific software
packages installed. (The list will be sent to participants a week before the

       •      using the shell to do more in less time
       •      using version control to manage and share information
       •      basic Python programming
       •      how (and how much) to test programs
       •      working with relational databases
Registration: Please use this form to register your interest by April 6. We
will allocate places to individuals and groups we believe will benefit, while
aiming for a balance of subjects and geographical areas. We strongly encourage
learners to attend with colleagues, so please create or join a team when you
sign up. Ideally we would like teams of 3 to 6 members, and we will give
preference to teams including one more knowledgeable member who is interested
in organizing further training at their own institution. By registering you
are stating that you will attend both days of the workshop and participate in
```

```
the online exercises and follow-up sessions in the following weeks. We will
notify all applicants as to whether they have a spot no later than April 13.

Contact: For further information please e-mail us at swc2012@ncl.ac.uk.
```

What if you're not sure of the case that letters are in?

```
> grep –iw the Blurb.txt
to be informed of similar events in the future, please e-mail us on
sessions for 6 to 8 weeks extending the material from the boot camp.
Who: The course is aimed at postgraduate students and other scientists who are
packages installed. (The list will be sent to participants a week before the
Content: The syllabus for this boot camp will include:
       •      using the shell to do more in less time
are stating that you will attend both days of the workshop and participate in
the online exercises and follow-up sessions in the following weeks. We will
```

Finding files:

List all the stuff I've got?

```
> cd ../..
> find .
.
./clean
./Dissertation.txt
./ImportantWork
./ImportantWork/NewTheorem
./myText.txt
./oldResults.dat
./Private
…
```

List only the directories

```
> find . –type d
.
./ImportantWork
./ImportantWork/NewTheorem
./Private
./Public
./Rubbish
./Rubbish/RejectedPaper
./Tests
./Tests/Testing1
…
```

What things are empty?

```
> find . –empty
./ImportantWork/NewTheorem
./Private
./Public
./Tests/Testing1
./Tests/Testing2
./Tests/Testing3
./Work/DataDump
```

```
./Work/DocumentsToRead
./Work/Papers
./Work/SourceCode/myBigIdea.txt
```

I don’t want to go that deep!

```
> find . –empty –maxdepth 2
./ImportantWork/NewTheorem
./Private
./Public
./Tests/Testing1
./Tests/Testing2
./Tests/Testing3
./Work/DataDump
./Work/DocumentsToRead
./Work/Papers
```

Do I have any files writable by anyone?

```
> find . –perm -a=w
./Work/SourceCode/myBigIdea.txt
```

What are all my .txt files?

```
> find . -name \*.txt
./Dissertation.txt
./myText.txt
./Work/results.txt
./Work/SoftwareCarpentry/Blurb.txt
./Work/SourceCode/myBigIdea.txt
./Work/thisWeeksWork.txt
```

How many words are in each of my .txt files?

```
> find . -name \*.txt -exec wc -w {} \;
     333 ./Dissertation.txt
     439 ./myText.txt
      36 ./Work/results.txt
     394 ./Work/SoftwareCarpentry/Blurb.txt
       0 ./Work/SourceCode/myBigIdea.txt
      20 ./Work/thisWeeksWork.txt
```

Slide 8 – Job Control

```
> cd Work/SourceCode
```

Want to run my program…

```
> ./calcPi.py
```

But I need to quickly check something

```
> ^Z
```

I’ve now got control back

```
> wc –l calcPi.py
      14 calcPi.py
```

Now back to my code…

```
> fg
```

What if I’d prefer my program to work in the background?

```
> ^Z
> bg
```

Or start my job in the background to start with

```
> ./calcPi.py > result.txt &
[1] 20114
```

If I plan to log out…

```
> nohup ./calcPi.py > results.txt &
[1] 20118
```

List my running jobs

```
> jobs
[1]+  Running                 nohup ./calcPi.py > results.txt &
```

List all process I’m running

```
> ps
  PID TTY           TIME CMD
17737 ttys000    0:00.01 -bash
19365 ttys001    0:00.01 -bash
19380 ttys002    0:00.16 -bash
20118 ttys002    0:05.21 python ./calcPi.py
```

Killing off a running job

```
> kill 20118
[1]+  Terminated              nohup ./calcPi.py > results.txt
```

What’s going on?

```
> top
Processes: 97 total, 5 running, 92 sleeping, 578 threads               20:37:11
Load Avg: 1.77, 1.58, 1.51  CPU usage: 15.90% user, 4.9% sys, 80.0% idle
SharedLibs: 8384K resident, 2960K data, 0B linkedit.
MemRegions: 24082 total, 3174M resident, 21M private, 684M shared.
PhysMem: 2481M wired, 2747M active, 28M inactive, 5256M used, 2930M free.
VM: 228G vsize, 1039M framework vsize, 5546419(21) pageins, 2136971(0) pageouts.
Networks: packets: 14178641/4808M in, 17006914/13G out.
Disks: 2816085/103G read, 4621984/148G written.

PID    COMMAND      %CPU TIME     #TH  #WQ  #POR #MREG RPRVT  RSHRD  RSIZE
20124  top          6.2  00:00.34 1/1  0    24   33    1236K  244K   1816K
20122- mdworker32   0.0  00:00.24 3    1    47   144   2180K  14M    5836K
20110  mdworker     0.0  00:00.22 3    1    49   75    1360K  13M    3436K
19809  quicklookd   0.0  00:00.69 8    2    102  260   12M    13M    17M
19795  VirtualBoxVM 8.3  13:03.99 21/1 1    241  1113  1076M- 52M+   1120M
19791  VBoxSVC      0.1  00:48.96 13   1    131  143   2888K  16M    11M
19789  VBoxXPCOMIPC 0.1  00:23.76 1    0    21   34    288K   1740K  3256K
19785  VirtualBox   0.3  00:37.83 7    2    123+ 320+  6372K+ 47M    35M+
19628  hdiejectd    0.0  00:00.58 2    1    33   48    104K   492K   992K
19623  diskimages-h 0.0  00:18.29 3    1    75   87    3228K  8632K  5840K
19534- Microsoft Wo 0.6  05:51.46 6    1    137  755   22M    93M    74M
19511- Microsoft Po 0.3  01:29.52 9    2    174  1439  7912K+ 110M-  68M
19380  bash         0.0  00:00.16 1    0    17   25    248K   748K   864K
```

```
19379  login        0.0  00:00.01 1    0    22   54    104K   312K   912K
```

Slide 9 - Variables

Explain what shell variables are

What variables are there?

```
> set
Apple_PubSub_Socket_Render=/tmp/launch-Mpk2xQ/Render
BASH=/bin/bash
BASH_ARGC=()
BASH_ARGV=()
BASH_LINENO=()
BASH_SOURCE=()
BASH_VERSINFO=([0]="3" [1]="2" [2]="48" [3]="1" [4]="release" [5]="x86_64-apple-darwin10.0")
BASH_VERSION='3.2.48(1)-release'
COLUMNS=80
…
```

What is the value of $HOME?

```
> echo $HOME
/Users/nasm3
```

Creating my own variables

```
> MYVARIABLE=steve
> echo $MYVARIABLE
steve
```

Going to another shell…

```
> bash
> echo $MYVARIABLE
```

- gives nothing

```
> exit
```

If you want to take a variable with you…

```
> export MYVARIABLE
> bash
> echo $MYVARIABLE
steve
> exit
```

Slide 6 – Secure Shell

Logging into a remote computer

```
> ssh nasm3@unix.ncl.ac.uk
nasm3@unix.ncl.ac.uk's password:
Last login: Sat Oct 20 20:08:03 2012 from 94-193-69-
87.zone7.bethere.co.uk
Welcome to ALDRED.

Information on using Unix and the ISS Unix services is at
http://www.ncl.ac.uk/iss/unix/
```

```
[nasm3@aldred ~]$ exit
```

Copying a file over

```
> cd ../..
> scp Dissertation.txt nasm3@unix.ncl.ac.uk:
nasm3@unix.ncl.ac.uk's password:
Dissertation.txt                                 100% 2232     2.2KB/s   00:00
```

Running commands remotely

```
> ssh nasm3@unix.ncl.ac.uk ls
ssh nasm3@unix.ncl.ac.uk ls
nasm3@unix.ncl.ac.uk's password:
7za
Accounts.rtf
allRuns
CGC2011-J
condor
CondorHistory
condorTest
CUDA
Desktop
Dissertation.txt
ec2
Flood
GCG-graphs
Get_Condor_Config
Gnuplot
Mail
Matt
mergeSorted.txt.gz
multipleJob
old
removeComputer
resultsK0.csv
resultsK1.csv
resultsK2.csv
resultsK3.csv
resultsK4.csv
resultsK5.csv
resultsK6.csv
resultsK7.csv
ScanDel
simpleAdd
SimpleJavaVersion
singleJob
start.sh
status.tar.bz2
st.txt
test
```

```
Testing
tmp.txt
```