



# Introduction to Unix and HPC

## Answer Sheet

**Mairéad Stephens | Jeff Christiansen | 24 May 2013 | 2.0**

# Shell Primer

## Unit 1: Run Spot, Run

### Exercise 1(a) – Running commands

<b>ls</b>	Directory listing
<b>w</b>	Shows who is logged on to the system and what they are doing
<b>w hpc22</b>	Shows login, idle time, and what user hpc22 is doing
<b>finger</b>	Shows login, username & login details of users on the system
<b>finger hpc22</b>	Shows user hpc22's login details including name, idle time, login time as well as some of their environment settings
<b>date</b>	Prints the system time and date
<b>uptime</b>	Tells you how long the system has been running

### Exercise 1(b) – Flags and Parameters

#### Standard UNIX commands

<b>ls</b>	Lists directory contents of current directory
<b>ls -l</b>	(long) Lists directory contents of current directory using long listing format
<b>ls -a</b>	(all) Lists directory contents of current directory and does not ignore entries starting with .
<b>ls --all</b>	(all) Lists directory contents of current directory and does not ignore entries starting with . ( <i>alternative form of previous command</i> )
<b>ls -la</b>	(all+long) Lists directory contents of current directory using long listing format and does not ignore entries

	starting with .
<code>ls --all -l</code>	(all+long) Lists directory contents of current directory using long listing format and does not ignore entries starting with .
<code>ls --format=horizontal</code>	Lists directory contents of current directory horizontally
<code>ls --format=single-column</code>	Lists directory contents of current directory in a single column

### Exercise 1(c) – The Calendar

<code>man cal</code>	displays the manual page for the calendar function
<code>cal -l 04 1980</code>	displays the single month view of April in 1980 (04/1980)
<code>cal 04 1980</code>	displays the single month view of April in 1980 (04/1980) -l is the default view, so this command generates the same output as the previous command
<code>cal 02 04 1980</code>	displays the single month view of April 1980, with the date 02 highlighted (02/04/1980)
<code>cal -m 02 04 1980</code>	displays the single month view of April 1980, with the date 02 highlighted (02/04/1980) with Monday as the first day of the week (instead of Sunday which is the default)

## Unit 2: Where am I?

### Exercise 2(a): Finding your way around

(1)	<code>ls -la</code>	show the directory listing for the current directory
(2)	<code>ls training -or- ls -l training</code>	show the directory listing for the training directory ("long" or detailed view)
(3)	<code>cd training</code>	change directory (cd) to a subdirectory called "training"
(4)	<code>ls /home/hpc22/ -or- ls ~ -or- ls ~/hpc22</code>	Show the directory listing for current user using either <ul style="list-style-type: none"> <li>○ Absolute path</li> <li>○ Home directory of current user ~</li> <li>○ Home directory of specified user ~hpc22</li> </ul>

### Exercise 2(b): Finding your way around (part 2)

(1)	<ul style="list-style-type: none"> <li>○ <code>../regurgitator.sh</code></li> <li>○ <code>~/regurgitator.sh</code></li> <li>○ <code>~hpc22/regurgitator.sh</code></li> <li>○ <code>/home/hpc22/regurgitator.sh</code></li> <li>○ <code>../regurgitator.sh -d</code></li> </ul>	<ul style="list-style-type: none"> <li>○ run the regurgitator.sh script from the <b><u>parent directory of my current location</u></b></li> <li>○ run the regurgitator.sh script from <b><u>my home directory</u></b></li> <li>○ run the regurgitator.sh script from <b><u>hpc85's home directory</u></b></li> <li>○ run the regurgitator.sh script from the <b><u>absolute path to</u></b></li> </ul>
-----	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

		<b><u>hpc85's home directory</u></b> <ul style="list-style-type: none"> <li>Will run the "regurgitator.sh" script which will print the program name and arguments input. The -d flag will cause the current working directory to be printed</li> </ul>
(2)	<code>pwd</code>	Path Working directory command will list the current working directory
(3)	<code>ls .</code>	Will give a listing of all files in the current working directory (current dir)
(4)	<code>ls ../../hpc23</code>	Will give a listing of all files in hpc23's home directory
(5)	<code>ls --recursive ~ -or-</code> <code>ls --recursive ~hpc22 -or-</code> <code>ls --recursive .. -or-</code> <code>ls --recursive /home/hpc22</code>	Will give a recursive listing of all files in the parent directory and sub-directories

## Exercise 2(c): Making Directories

(1)	<code>cd ~</code>	Change directory to my home directory
(2)	<code>mkdir experiments</code> <code>mkdir "experiments/experiment one"</code> <code>mkdir "experiments/experiment one/step one"</code> <code>mkdir "experiments/experiment</code>	Make a directory called "experiments" in ~  Make a directory called "experiment one" in ~/"experiment one"  Make a directory called

	<pre>one/step two" cd "experiment two" mkdir "step one" mkdir "step two"</pre>	<p>"experiments"</p> <p>Make a directory called "experiments"</p>
(2)	<pre>ls --recursive ~/experiments - or- tree ~/experiments</pre>	<p>Show the directory structure for the newly created "experiments" folder</p>

## Exercise 2(d): Moving and Copying Files

(1)	<pre>cp ~/regurgitator.sh ~/experiments cp ~/regurgitator.sh /home/hpc22/experiments/"experiment one" cp ~/regurgitator.sh ~/experiments/"experiment two" cd ~ cp ./regurgitator.sh ~/experiments/"experiment one"/"step one" cd experiments cp ../regurgitator.sh "experiment one/step two" cd "experiment two" cp ~/regurgitator.sh ./"step one" cp /home/hpc22/regurgitator.sh "step two"</pre>
(2)	<pre>cp ~/regurgitator.sh ~/copy_of_regurgitator.sh</pre>
(3)	<pre>mv ~/copy_of_regurgitator.sh ~/experiments</pre>

### Exercise 3(a)

(1)	<code>cd ~/training</code>	The <code>cd</code> command will change directory to the training directory
(2)	<code>cat poem.txt</code>	The <code>cat</code> command will print out the contents of the poem.txt file
(3)	<code>wc -w poem.txt</code>	The <code>wc</code> command will count the newlines, words and byte counts for a file. Using the <code>-w</code> flag will only show the number of words in a file
(4)	<code>cat poem.txt poem.txt</code>	This will print out the contents of the poem.txt file twice
(5)	<code>less poem.txt</code>	The <code>less</code> command will page through the contents of the poem.txt file one page at a time
(6)	<code>tail poem.txt</code>	The <code>tail</code> command will display the last 10 lines of the poem.txt file
(7)	<code>cat poem.rtf</code>	The <code>cat</code> command print out the contents of the poem.rtf file
(8)	<code>du</code>	The <code>du</code> command will display the disk space being used by all of the files in the current directory
(9)	<code>du -all</code>	The <code>du</code> command will display the disk space being used by each of the files in the current directory, and a total for all files

## Exercise 4(a)

(1)	<code>cd ~/training</code>	Change directory to the training directory
(2)	<code>fortune</code> <code>fortune</code>	The <code>fortune</code> command will give you a once off fortune (like a Chinese fortune cookie)
(3)	<code>fortune &gt;</code> <code>my_fortune.txt</code> <code>cat my_fortune.txt</code>	The <code>&gt;</code> will redirect the output of the <code>fortune</code> command into a file called <code>my_fortune.txt</code> The <code>cat</code> command print out the contents of the <code>my_fortune.txt</code> file
(4)	<code>fortune &gt;</code> <code>my_fortune.txt</code> <code>cat my_fortune.txt</code>	As above
(5)	<code>fortune &gt;&gt;</code> <code>my_fortune.txt</code> <code>cat my_fortune.txt</code>	The <code>&gt;&gt;</code> will append the output of the <code>fortune</code> command into a file called <code>my_fortune.txt</code> The <code>cat</code> command print out the contents of the <code>my_fortune.txt</code> file
(5)	<code>sort long_poem.txt</code> <code>  uniq &gt;</code> <code>copy_long_poem.txt</code> <code>cat</code> <code>copy_long_poem.txt</code>	The <code>sort</code> command will sort the contents of the <code>long_poem.txt</code> The <code>uniq</code> command will remove any duplicate (redundant) lines from the poem The <code>&gt;</code> will redirect the output of the <code>fortune</code> command into a file called <code>copy_long_poem.txt</code>



### Exercise 5(a): Checking the environment variables

(1)	<code>echo \$PATH</code>	The <code>echo</code> command will display what the <code>PATH</code> environment variable is set to
	<code>echo \$HOME</code>	The <code>echo</code> command display what the <code>HOME</code> environment variable is set to
	<code>echo \$TEMP</code>	The <code>echo</code> command display what the <code>TEMP</code> environment variable is set to
	<code>echo \$PS1</code>	The <code>echo</code> command display what the <code>PS1</code> environment variable is set to
(2)	<code>env</code>	Will display all the environment variables and what they are set to
(3)	<code>env   grep PATH</code> <code>&gt; path.txt</code>	<p>The <code>env</code> command will display the environment variables and what they are set to</p> <p>The <code>grep</code> command will search for any instances of the word "PATH"</p> <p>The <code>&gt;</code> will redirect the output of the two commands into a file called <code>path.txt</code></p>

### Exercise 5(b): Manipulating the Environment

(1)	<code>export MYNAME=Joe</code>	The <code>export</code> command will set the value of the <code>MYNAME</code> environment variable to "Joe"
(2)	<code>export MYNAME=Thurbon</code>	The <code>export</code> command will set the value of the <code>MYNAME</code> environment variable to "Thurbon"
(3)	<code>export MYNAME=Joe</code> <code>: \$MYNAME</code>	The <code>export</code> command will set the value of the <code>MYNAME</code> environment variable to "Joe:Thurbon"

(4)	<pre>cd ~ mv regurgitator.sh ./bin regurgitator.sh</pre>	<p>The <code>cd</code> command will change your current working directory to your home directory</p> <p>The <code>mv</code> command will move the <code>regurgitator.sh</code> file to the <code>bin</code> directory</p> <p>The final command attempts to run the <code>regurgitator.sh</code> file, but as it doesn't exist in the <code>~</code> directory, this command should fail</p>
(5)	<pre>export PATH=\$PATH:/home/hpc22/bin</pre>	<p>The <code>export</code> command will set the value of the <code>PATH</code> environment variable to the existing <code>PATH</code> variable plus the addition of your <code>bin</code> directory <code>:/home/hpc22/bin</code></p>
(6)	<pre>export PS1="\u@my_new_prompt:\w&gt;"</pre>	<p>You can set the <code>PS1</code> to whatever you want. The sample here will result in the users prompt being changed to the following:</p> <ul style="list-style-type: none"> <li>○ <code>hpc85@my_new_prompt:~&gt;</code></li> </ul>

# HPC

## Exercise 3: Putting it all together

Sample script:

```
#!/bin/bash -login
#PBS -l walltime=00:10:00
#PBS -l nodes=1:ppn=1
#PBS -l mem=100mb
#PBS -P a40
#PBS -M fred@intersect.org.au
#PBS -m abe
cd $PBS_O_WORKDIR
source /usr/share/modules/init/bash
module load clustalw
mkdir /home/hpc##/training/clustal_example/
cd /home/hpc##/training/clustal_example/
wget
https://raw.githubusercontent.com/IntersectAustralia/TrainingMaterials/master/IntroToU
nixHPC/70_text_strings.txt
clustalw2 -INFILE=70_text_strings.txt -PWMATRIX=BLOSUM -PWGAPOPEN=1000
```