Intersect Australia Ltd
PO Box H58, Australia Square
Sydney NSW 1215

T +61 2 8079 2500
F +61 2 9262 3040

ABN 67 131 752 657

Introduction to Unix for HPC

Answer Sheet

Mairéad Stephens | Jeff Christiansen | 24 May 2013 | 2.0

Shell Primer

Unit 1: Run Spot, Run

Exercise 1(a) – Running commands

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

Exercise 1(b) – Flags and Parameters

Standard UNIX commands

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

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

Exercise 1(c) – The Calendar

`man cal`	displays the manual page for the calendar function
`cal -1 04 1980`	displays the single month view of April in 1980 (04/1980)
`cal 04 1980`	displays the single month view of April in 1980 (04/1980) -1 is the default view, so this command generates the same output as the previous command
`cal 02 04 1980`	displays the single month view of April 1980, with the date 02 highlighted (02/04/1980)
`cal -m 02 04 1980`	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)	ls -la	show the directory listing for the current directory
(2)	**`ls training -or-`** **`ls -l training`**	show the directory listing for the training directory ("long" or detailed view)
(3)	**`cd training`**	change directory (cd) to a subdirectory called "training"
(4)	**`ls /home/hpc22/ -or-`** **`ls ~ -or-`** **`ls ~/hpc22`**	Show the directory listing for current user using either ○ Absolute path ○ Home directory of current user ~ ○ Home directory of specified user ~hpc22

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

(1)	○ **`../regurgitator.sh`** ○ **`~/regurgitator.sh`** ○ **`~hpc22/regurgitator.sh`** ○ **`/home/hpc22/regurgitator.sh`** ○ **`../regurgitator.sh -d`**	○ run the regurgitator.sh script from the **parent directory of my current location** ○ run the regurgitator.sh script from **my home directory** ○ run the regurgitator.sh script from **hpc85's home directory** ○ run the regurgitator.sh script from the **absolute path to**

		hpc85's home directory ○ 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
(2)	`pwd`	Path Working directory command will list the current working directory
(3)	`ls .`	Will give a listing of all files in the current working directory (current dir)
(4)	`ls ../../hpc23`	Will give a listing of all files in hpc23's home directory

(5)	`ls --recursive ~ -or-`
	`ls --recursive ~hpc22 -or-`
	`ls --recursive .. -or-`
	`ls --recursive /home/hpc22`
		Will give a recursive listing of all files in the parent directory and sub-directories

Exercise 2(c): Making Directories

(1)	`cd ~`	Change directory to my home directory

(2)	`mkdir experiments`
	`mkdir "experiments/experiment one"`
	`mkdir "experiments/experiment one/step one"`
	`mkdir "experiments/experiment`
		Make a directory called "experiments" in ~
		Make a directory called "experiment one" in ~/"experiment one"
		Make a directory called

	`one/step two"`
	`cd "experiment two"`
	`mkdir "step one"`
	`mkdir "step two"`
		"experiments"
		Make a directory called "experiments"

(2)	`ls --recursive ~/experiments -or-` `tree ~/experiments`	Show the directory structure for the newly created "experiments" folder

Exercise 2(d): Moving and Copying Files

(1)	`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"`

(2)	`cp ~/regurgitator.sh ~/copy_of_regurgitator.sh`
(3)	`mv ~/copy_of_regurgitator.sh ~/experiments`

Exercise 3(a)

(1)	**`cd ~/training`**	The **`cd`** command will change directory to the training directory
(2)	**`cat poem.txt`**	The **`cat`** command will print out the contents of the poem.txt file
(3)	**`wc –w poem.txt`**	The **`wc`** command will count the newlines, words and byte counts for a file. Using the –w flag will only show the number of words in a file
(4)	**`cat poem.txt poem.txt`**	This will print out the contents of the poem.txt file twice
(5)	**`less poem.txt`**	The **`less`** command will page through the contents of the poem.txt file one page at a time
(6)	**`tail poem.txt`**	The **`tail`** command will display the last 10 lines of the poem.txt file
(7)	**`cat poem.rtf`**	The **`cat`** command print out the contents of the poem.rtf file
(8)	**`du`**	The **`du`** command will display the disk space being used by all of the files in the current directory
(9)	**`du -all`**	The **`du`** 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)	**`cd ~/training`**	Change directory to the training directory
(2)	**`fortune`** **`fortune`**	The **`fortune`** command will give you a once off fortune (like a Chinese fortune cookie)
(3)	**`fortune > my_fortune.txt`** **`cat my_fortune.txt`**	The `>` will redirect the output of the **`fortune`** command into a file called **`my_fortune.txt`** The **`cat`** command print out the contents of the **`my_fortune.txt`** file
(4)	**`fortune > my_fortune.txt`** **`cat my_fortune.txt`**	As above
(5)	**`fortune >> my_fortune.txt`** **`cat my_fortune.txt`**	The `>>` will append the output of the **`fortune`** command into a file called **`my_fortune.txt`** The **`cat`** command print out the contents of the **`my_fortune.txt`** file
(5)	**`sort long_poem.txt | uniq > copy_long_poem.txt`** **`cat copy_long_poem.txt`**	The **`sort`** command will sort the contents of the **`long_poem.txt`** The **`uniq`** command will remove any duplicate (redundant) lines from the poem The `>` will redirect the output of the **`fortune`** command into a file called **`copy_long_poem.txt`**

Exercise 5(a): Checking the environment variables

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

Exercise 5(b): Manipulating the Environment

(1)	`export MYNAME=Joe`	The **`export`** command will set the value of the MYNAME environment variable to "**`Joe`**"
(2)	`export MYNAME=Thurbon`	The **`export`** command will set the value of the MYNAME environment variable to "**`Thurbon`**"
(3)	`export MYNAME=Joe :$MYNAME`	The **`export`** command will set the value of the MYNAME environment variable to "**`Joe:Thurbon`**"

(4)	**cd ~** **mv regurgitator.sh ./bin** **regurgitator.sh**	The **cd** command will change your current working directory to your home directory The **mv** command will move the **regurgitator.sh file to the bin** directory The final command attempts to run the **regurgitator.sh** file, but as it doesn't exist in the ~ directory, this command should fail
(5)	**export PATH=$PATH:/home/hpc22/bin**	The **export** command will set the value of the PATH environment variable to the existing PATH variable plus the addition of your bin directory **:/home/hpc22/bin**
(6)	**export PS1="\u@my_new_prompt:\w>"**	You can set the PS1 to whatever you want. The sample here will result in the users prompt being changed to the following: ○ **hpc85@my_new_prompt:~>**

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.github.com/IntersectAustralia/TrainingMaterials/master/IntroToU
nixHPC/70_text_strings.txt
clustalw2 -INFILE=70_text_strings.txt -PWMATRIX=BLOSUM -PWGAPOPEN=1000
```