Sequence Analysis Spring 2000 Dr. Richard Friedman
(212)305-6901 (76901) friedman@cuccfa.ccc.columbia.edu 130BB

Lesson 3 Advanced Unix file handling and job control commands

Assignment:

Read and work through *Unix Primer Plus* p. 72-76, 151-156, 162-167, 169-181, 255-263

Lab:

Work through the morefiles module in learn program.
Note: morefiles Sections 3.1b and 4.1c do not work properly.

Summary of commands:

Note: In this document different fonts have different meanings:

Times is used to explain commands.

`Courier is used to indicate commands and command options.`

`Courier italics are used to indicate command parameters, for example, filenames.`

`Courier bold is used to indicate commands that are not displayed.`

`Courier bold italics are used to indicate computer- generated output.`

Helvetica is used to indicate menu items.

`grep` *`GAATTC sample.tfa`*	Displays all lines containing 'GAATTC' that occur in the file 'sample.tfa'. grep stands for '**g**lobal **r**egular **e**xpression **p**arser'.
`grep` *`'like me' fly2`*	Displays all lines containing 'like me' that occur in the file 'fly2'.
`>`	Redirection operator.
`ls` *`a00.seq a01.seq`* `>` *`allnucfiles`*	Writes the names of the files a00.seq and a01.seq into the file allnucfiles.
`ls` *`a02.seq a03.seq`* `>` *`allnucfiles`*	Overwrites 'allnucfiles' with the names of the files a02.seq and a03.seq.
`>>`	Redirects and appends.
`ls` *`a02.seq a03.seq`* `>>` *`allnucfiles`*	Adds the names of the files a02.seq and a03.seq to allnucfiles.

`ls a*.seq > allnucfiles`
Writes the names of all files beginning with ‘a’ and ending with ‘.seq’ into the file ‘allnucfiles’.

`ls a*q > allnucfiles`
Writes the names of all files beginning with ‘a’ and ending with ‘q’ into the file ‘allnucfiles’.

`ls ?2?.pep`
Displays the names of all files whose names have ‘2’ as their second character and any character as their first and third character.

`ls fly*`
Displays:
1 fly 1 fly1 1 fly2

`ls fly?`
Displays:
1 fly1 1 fly2

‘*’ stands for all or no characters.
‘?’ stands for single characters.

`ls a2[3,5].seq`
Displays:
a23.seq a25.seq

`ls a2[3-5].seq`
Displays:
a23.seq a24.seq a25.seq

`ls a3[5-9].seq a4[0-6].seq`
Displays:
1 a35.seq 1 a38.seq 1 a41.seq 1 a44.seq
1 a36.seq 1 a39.seq 1 a42.seq 1 a45.seq
1 a37.seq 1 a40.seq 1 a43.seq 1 a46.seq

`.`
Stands for the directory you are in.

`..`
Stands for the directory immediately above the one you are in.

`cd ..`
Goes one directory up.

`cp -r /parnassu/users/playwrights/aeschylus/oresteia /parnassu/users/playwrights/euripides/`
Copies the directory:
‘/parnassu/users/playwrights/aeschylus/oresteia’
to the directory:
‘/parnassu/users/playwrights/aeschylus/’

`cp -r /parnassu/users/playwrights/aeschylus/oresteia .`
Does the same thing as the above command.

`cp -r ../aeschylus/oresteia .`
Does the same thing as the above two commands.

`cat hba_hobbit.pep > hobbitproteins`
Places the content of ‘hba_hobbit.pep’ in ‘hobbitproteins’.

`cat hbaz_hobbit.pep > hobbitproteins`
Overwrites ‘hobbitproteins’ with the content of ‘hbaz_hobbit.pep’.

`set noclobber`	Sets the machine so that files cannot be overwritten. When noclobber is set and
`cat hbaz_hobbit.pep > hobbitproteins`	is typed, the machine responds with:
hobbitproteins: File exists.	
`cat hbaz_hobbit.pep >! hobbitproteins`	Overrides "noclobber" so that 'hobbitproteins' is overwritten with the content of 'hbaz_hobbit.pep'.
`unset noclobber`	Allows files to be overwritten again.
`cat hbaz_hobbit.pep >> hobbitproteins`	Appends 'hbaz_hobbit.pep' to hobbitproteins.
`program > outputfile`	Redirects output of program to outputfile rather than to screen.
`program < inputfile`	Inputs contents of inputfile into program, rather than typing in the input.
`program < inputfile > outputfile`	Inputs contents of inputfile into program, rather than typing in the input, and simultaneously redirects output of program to outputfile rather than to screen.
`program  > outputfile < inputfile`	This will also work.
`inputfile > program  > outputfile`	This will not work.
`outputfile < program < inputfile`	This will also not work.
	In order for redirection to work: 1. The program name must be the first word on the line. 2. The redirection operators must be pointing in opposite directions.

`ls *seq > seqlist`	1. Lists the number of files ending in ‘seq’ in a file called ‘seqlist’.
`wc seqlist`	2. Counts the number of lines , word, and characters in the file called ‘seqlist’.
`rm -f seqlist`	3. Removes the file called ‘seqlist’. The net effect is to count the number of files ending in ‘seq’.
`ls *seq > seqlist; wc seqlist; rm -f seqlist`	Does the same thing as the previous command but is entered on one line.
`ls *seq | wc`	Lists the files ending in ‘seq’ and then counts the number of words, lines, and characters in that list. Does the same thing as the previous two commands but the file with the list of files ending in ‘seq’ is never mentioned explicitly. ‘|’ is called a pipe.
`command1 filename(s) | command2`	Takes the output of command1 and uses it as the input of command 2.
`gcg`	Initializes the GCG package.
`fasta`	Starts the Fasta program in foreground with terminal input.
`fasta &`	Starts the Fasta program in background. Doesn’t work because terminal input is impossible.
`fasta -nomon -in=hba_hobbit.pep -out=hba_hobbit.fasta -D &`	Starts the Fasta program in background so that it will run to completion.
`fg`	Moves program in background to foreground.
<ctrl> z	Stops program.
`bg`	Moves program in foreground to background. You can log off when background jobs are running, but you can’t put the jobs in foreground when you log on again. Therefore make sure that all of your background jobs have enough information to run before you logout.
`jobs`	Lists current jobs.
`stop %1`	Stops job #1.
`bg %1`	Puts job #1 in background.
`kill %2`	Kills job 2.
`ps`	Lists current processes of user (**p**rocess **s**tatus)
R	Job is running.
T	Job is stopped.
U,S,I	Job is idle.

`ps u`	Lists current process of user in more detail (**p**rocess **s**tatus **u**nabridged).
`ps a`	Lists current processes of everybody running (**p**rocess **s**tatus **a**ll).
`ps ua`	Lists current processes of everybody running in more detail (**p**rocess **s**tatus **u**nabridged **a**ll).
`kill 4032`	Kills job with process ID number 4032.
`kill -9 4032`	Kills otherwise unkillable job with process ID number 4032.
`chmod +x` *`commandfilename`*	Makes a file containing Unix commands executable. Executable files are displayed with an asterik, *e.g.* "comandfilename *". You do NOT include the star when you type the file's name.
`nohup` *`commandfilename`* `&`	Runs a comandfile in background and continues running it after you log off. Some programs require "nohup" to continue runnning in background after logoff. Others don't. I play it safe by including "nohup".
`cat` *`hbb_hobbit.pep`*	Mistyped filename. The machine responds:
`cat: cannot open hbb_hobbit.pep`	This can be corrected by typing:
`^bb^ba`	To which the machine responds:
`cat hba_hobbit.pep`	and displays the file.
`cat` *`hba_hobait.pep`*	Mistyped filename. The machine responds:
`cat: cannot open hba_hobait.pep`	
`^ba^bb`	If one types, as before,
`Modifier failed.`	The machine responds:
`^bai^bbi`	The correct way to correct this command is to type: and the machine displays the contents of your file.
`more` *`ba`*`<esc>`***`cchae`**	The machine completes the filename after the escape key is pressed.
`more` *`ip`*`<esc>`***`higenia_in_a`**`<esc>`***`ulis`**	If there are two files in the directory that fit the specification, upon typing escape the machine will fill in the file names up until the point at which the filenames disagree. Then type another letter to resolve the ambiguity followed by escape. The machine fills the desired filename in.
`find . -name` *`antigone`* `-print`	Searches for the file(s) named ‘antigone’ in the current directory and all of its sub-directories,

	recursively and displays the locations of all of the files.
`find . -name '*iph*' -print`	Searches for the file(s) containing "iph" in the current directory and all of its sub-directories, recursively and displays the locations of all of the files.
`history`	Lists the most recent commands with their numbers.
`his`	Cuccfa abbreviation for history.
`/parnassu/users/playwrights_7> u`	Command #7 is 'u'.
`!!`	Repeat most recent command.
`!7`	Repeat command #7.
`!c`	Repeat most recent command beginning in c.
`!ca`	Repeat most recent command beginning in ca.
`!$`	Last word of previous command.
`alias his 'history'`	Makes the computer understand 'his' to mean 'history'.
`source .login`	Activates .login file.

The following books are good for learning more about Unix:

Another good book, on the same level as *Unix Primer Plus*. but briefer, and less pedagogical, and complete, is *Learning the Unix Operating System*, by Grace Todino, John Strang, and Jerry Peek, 3rd. Ed., O'Reilly & Associates, Sebastapol CA, 1993. pp. 92. ISBN 1-56592-060-0.

Users familiar with VMS will find *Unix for VMS Users*, by Philip E. Bourne, Digital Press,1990, ISBN 1-55558-034-3, useful.

DEC OSF/1 (DEC Unix) Command and Shell Users Guide, Version 3.0 (or above), discussed above, is also written clearly enough to learn from, once you have read either *Unix Primer Plus* or *Learning the Unix Operating System*. We reemphasize that this book describes the specific version of Unix we use here.

Learning the Vi Editor, Linda Lamb, O'Reilly & Associates, Inc., 5th. Ed., O'Reilly & Associates, Sebastapol CA, 1994. pp. 174. ISBN 0-937175-67-6. An in-depth text for the vi editor.

Unix in a Nutshell: Berkeley Edition, by the Staff of O'Reilly & Associates, 1st Ed. O'Reilly & Associates, Sebastapol CA, 1986 (latest Update 1990, covering BSD 4.3). A good command-by-command description of Unix.

The Unix C Shell Field Guide, Gail Anderson and Paul Anderson. 1st Ed. Prentice-Hall, Englewood NJ, 1986. ISBN 0-13-937469-X. An excellent text on the features of the Unix C Shell, with emphasis on writing C-Shell programs.

Credits for texts used as examples:

Alfred Bester for Prologue to *The Stars my Destination.*
William Blake for *The Fly* and *The Tyger* from *Songs of Experience.*
Euripides for *The Bacchae, Iphigenia in Aulis, and Iphigenia in Tauris.*