Salgado Berardo, Joao Antonio 100280950 A10 Jul/06/2017 Mark:\_\_\_\_\_\_\_\_\_\_

Chapter 10: 1, 2, 6, 10, 11, 12, 13, 14, 16, 17, 30

**Chapter 10:**

**1.** What is the difference between a wild card and a regular expression?

**2.** What do these commands do?

(i) grep a b c searches for “a” in files “b” and “c”

(ii) grep <HTML> foo This command would not work, it should be between double quotes or use *grep \<HTML>*

(iii) grep “\*\*” foo The \* regular expression matches zero or more occurrences of the previous character, in this case, another \*.

(iv) grep \*. It is missing an argument.

**6.** Yes, they are both equivalent. The -v option inverts the match, selecting lines that don’t match.

The catch is the second caret (^) in the first command, that is missing in the grep command with -v option. If the second command didn’t have the -v option, one command would be the inverse of the other.

**10.** The shell executes the commands inside the back-tick first. So, it first executes grep, looking for the pattern fork in all C source files. The output of is piped to cut, that retrieves the first field delimited by “:”. The standard output of the cut command is piped to the sort command, that sorts the output which uses the -u (unique) option, to show the file name only one time.

The output of the commands inside back-ticks is used for ls to list the files ordered by modification date.

To use two commands instead of four, we can use the -l option from grep, that prints the name of the files, instead of normal output.

ls -t `grep -l fork \*.c`

**11.** How do you display the listing of all files in the current directory that have the

same permissions as ./foo?

**12.** Frame regular expressions to match these patterns:

(i) jefferies jeffery jeffreys grep jeff[er]\*[iy][es]\* file

(ii) hitchen hitchin hitching grep hitch[ei]ng\* file

(iii) Heard herd Hird grep [hH][ei][ar\*]d\* file

(iv) dix dick dicks dickson dixon grep di[cx][ko]\*[sn]\*[on]\* file

(v) Mcgee mcghee magee grep [Mm][ac]g[he]ee\*$

(vi) wood woodcock woodhouse.

**13.** How do these expressions differ?

(i) [ 0 - 9 ] \* and [ 0 - 9 ] [ 0 - 9 ] \*

(ii) ^[^^] and ^^^.

**14.** Frame a command sequence that looks at romeo’s mailbox to tell him either that

he has received a message from henry or that the Subject: line contains the

word urgent or immediate in lower- or uppercase.

**16.** Explain the significance of this command:

grep “^[^:]\*:[^:]\*:100:” /etc/passwd

**17.** How do you locate all lines containing printf, but not sprintf and fprintf,

anywhere but at the end of a line?

**30.** Specify the command sequence needed to remove the directory /usr/local/bin

from the PATH defined in $HOME/.profile.