

Lab #3 – Shell Commands

Type the following into a file called *headandtail*:

Read the help pages for commands *grep*, *cut*, *head*, and *tail*.

When using *grep* with *-n*, each line of output is prefixed with a line number within its input file.

Check the output of the *grep* with line number.

When using *cut*, you can change the delimiter using *-d*.

Command *head* outputs the first part of files.

Check for the options for output the first *n* lines.

Command *tail* outputs the last part of files.

Part I – *head* and *tail*

Practice the following and understand the results.

```
head -n 5                # type in at least 5 lines
head headandtail
head -n 2 headandtail
head -c 10               # type "Joy to the world"
head -c 10 headandtail
tail -n 1 headandtail
head -n 5 headandtail | tail -n 2
```

Now use *pipe* to connect *head* and *tail* to print out the second and third lines of file *headandtail*.

Part II – *grep*

Type this command:

```
grep -n line < headandtail
```

Check the print out and observe the line numbers separated by colons. Pipe this result with command *cut* to print out only the column of line numbers.

Then pipe the above column of line numbers with command *head* to print out the first line number of the line containing word *line*.

Part III

Use commands *grep*, *cut*, *head* and *tail* to write a bash script called *selector*. It accepts two words as arguments, and prints out the portion of the given file between the line of the first appearance of the first word (this line included) and the line of the first appearance of the second word (this line included).

You can assume that the first occurrence of the first word always appears earlier than the first occurrence of the second word.

Sample run:

```
>>>>> selector line delimiter
```

When using *grep* with *-n*, each line of output is prefixed with the 1-based line number within its input file.

Check the output of the *grep* with line number.

When using *cut*, you can change the delimiter using *-d*.

Part IV (optional)

Use bash *if*-statement and *test expression* to check if the first word appears after the second word. If so, inform the user about this.