Module 1.4 Chapter 5 The Shell Questions

Summer 2018 Intro to Systems Administration

By: L Snedden

* 1. points

1. What does the shell ordinarily do while a command is executing? What should you do if you do not want to wait for a command to finish before running another command?

**While a command is executing the shell usually sleeps. If you don’t want to wait for a command to finish executing, you can execute that command in the background. To do this add “&” to the end of the command.**

1. Using sort as a filter, rewrite the following sequence of commands using pipes and a tee to take the output from the list file, sort it, then send it to both the lpr and a file called text.

$ **sort list > temp**

$ **lpr temp**

$ **rm temp**

**$ cat list | sort | lpr**

1. What is a PID number? Why are these numbers useful when you run processes in the background? Which utility displays the PID numbers of the commands you are running?

**Process Identification Number or PID are assigned to all process currently running on the system. The “ps” utility can show PID numbers.**

1. Assume the following files are in the working directory shown below from the ls command:

$ **ls**

intro notesb ref2 section1 section3 section4b notesa ref1 ref3 section2 section4a sentrev

Give commands for each of the following, using wildcards to express filenames with as few characters as possible.

1. List all files that begin with **section**.

**$ ls section**\*

* 1. List the **section1**, **section2**, and **section3** files only.

**$ ls section[1-3]**

* 1. List the **intro** file only.

**$ ls i**\*

* 1. List the **section1**, **section3**, **ref1**, and **ref3** files.

**$ ls [13]\***

1. Refer to the info or man pages to determine which command will
   1. Display the number of lines in its standard input that contain the*word* **a** or **A**.

**$ command | grep -wci a**

1. Display only the names of the files in the working directory that contain the pattern **$(.**

**$ ls $\(**

1. List the files in the working directory in reverse alphabetical order.

**$ ls -r**

1. Send a list of files in the working directory to the printer, sorted by size.

**$ ls -S | lpr**

1. Give a command to
   1. Redirect standard output from a sort command to a file named **phone\_list**. Assume the inputfile is named **numbers**.

**$ sort numbers > phone\_list**

**c.** Create a file named **book** that contains the contents of two other files: **part1** and **part2**.

**$ cat part[12] > book**

**7.** The lpr and sort utilities accept input either from a file named on the command line or from standard input.

**a.** Name two other utilities that function in a similar manner. **Cat and grep.**

**b.** Name a utility that accepts its input only from standard input. **Xargs.**

* 1. Give an example of a command that uses grep. **grep "literal\_string" filename**

1. With both input and output redirected. **grep -i file2 < file1**
2. With only input redirected. **grep line < file1**
3. With only output redirected. **grep -n " literal\_string " \* > output-file**
4. Within a pipeline. **grep line < file1 | sort -r**
5. Which filenames would a subsequent ls command (B) display given the contents of the directory shown in (A)?

(A)$ **ls**

abc abd abe abf abg abh

(B)$ **ls ?b\***

**Output: abc abd abe abf abg abh**

**All of the files in the directory would be output because the command is looking for any file that has b as the second letter and anything after b.**