**Unix User Badge**

The set of three assessments for earning this badge are as follows:

1. A written set of questions that request you to perform a set of tasks (this document).
2. A real-time one-on-one session with the Professor in charge of this badge.
3. You must recruit and mentor one other individual to begin earning this badge.

There is no time limit on this part of the assessment and it may be completed at the learner’s pace.

This initial assessment is a written set of questions that asks you to perform a set of small tasks. The majority of work will require command-line proficiency. The set of UNIX/Linux commands to be tested will be in line with the set of Learning Goals. Your commands and answers (as obtained at the command-line) must be copied and pasted onto this test after each question and a hardcopy submitted to the Professor in charge of this badge. The initial question is done for you as an example of the formatting required, that is, after each question, you must show both the Linux command(s) and the command(s)’ output.

1. What is the command to see who is presently logged onto the server?

% **who**

mleblanc pts/1 2012-08-22 11:43 (wheaton\_10\_2853.wheatonma.edu)

foobar pts/2 2012-08-22 00:01 (wheaton\_10\_0000.wheatonma.edu)

1. Show the username you are currently using.
2. Show the current date.
3. Print the operating system name of the box you are using.
4. Make a new directory called food. Within this directory make a file called apple using the touch command. Create a hard link to apple called fruit. Show a “long” directory listing that shows file sizes and other information on each file.
5. Print your current working directory.
6. Return to your home directory without using a path.
7. From your home directory (without changing your current directory), list the files in the parent directory.
8. In your home directory, make another directory called stuff. Without changing your directory, copy all the files from fruit/ into stuff/. Your files should retain the same permissions. Show a listing of the files in stuff/.
9. Remove the directory stuff/ and all its contents with one command.
10. Write the command to list all the headers in /usr/include which begin with a vowel and end with  .h.
11. Explain what the following octal permissions represent:
12. octal permission value 755
13. octal permission value 644
14. octal permission value 220
15. Suppose you had a file called “runMe.sh”. Show the command that would change the privilege of this file so that only the user can read and execute this file. Set the privilege settings so even the user may not write to this file.
16. Using an editor of your choice (e.g., vi or emacs), create a file called foobar.dat. (Just type in a few bogus lines in the file and save it). Using the chmod(1) command, write the command used to ensure group and universe have no read and write permissions on the object foobar.dat. Show this done in two different commands: (i) with octal values and (ii) without using octal values.
17. Show the entire contents of your file one page at a time.
18. Show only the first 3 lines of your file.
19. Show only the last 2 lines of your file.
20. Show the number of lines, words, and characters in your file.
21. Show all the processes currently running on your box and their associated usage statistics.
22. What is the meaning of the columns of output from the ps command?
23. What is the significance of the TTY column in the process table? Why do some processes not have an entry in this column?
24. Suppose you have a runaway process with the ID = 1234567 that isn’t responding. Show the command that would force this process to be killed.

1. How many man pages are associated with the sleep command/function call? What chapters in the manual do they belong too?
2. How would you use the du command to report the total size of all directories that begin with the character 3 in a directory called /share/cs-pub?
3. What command would you use to list all the files beginning with any character but ending with r.h in /usr/include? Assume your current working directory is your home directory.
4. Explain what each of the following does?
5. ls -Rlat > /tmp/report
6. sort < numbers > results
7. CC \*.cpp \*.o 2>/dev/null
8. cat /etc/services | head -n 5
9. Show all the past commands you’ve typed in this session.
10. Re-execute the previous “ls” command.
11. Re-execute the very last command.
12. Store the output of the man page for the ifconfig command in a file called “ifconfig\_help.txt”.
13. Show all lines where the characters “static” appear in your file “ifconfig\_help.txt”.
14. Repeat the previous question but only show lines where the word “static” appears (that is, do not show lines that include words such as “statically”).
15. Send a network request to see if the server cs.wheatoncollege.edu is responding.
16. Show your hard disk statistics in “human-readable” form.
17. Show the disk usage for all files one page at a time.
18. Show the command to securely login to a remote server. Assume you have an account called “foo” on the cs.wheatoncollege.edu server.
19. Show the command to securely copy a file from your current directory to the “foo” account on the cs.wheatoncollege.edu server. The file should be placed into a top-level directory of the “foo” account call “stuff”.
20. Assume you have a directory called /fruit that is filled with hundreds of …. well, “fruitful data” files. Show the commands that will (a) make an archive (often called a “tar ball”) of this directory into one file called fruitTarBall and (b) compress (often referred to as “zip”) that tar ball.
21. Assume someone sends you a “zipped tar ball” called: lottaStuff.tar.gz. Show the command(s) that will decompress and extract the file into a directory.