LEBANESE AMERICAN UNIVERSITY

School of Arts and Science

Department of Computer Science and Mathematics

CSC 326: Operating Systems

Spring 2016

Files and Directories

1. Create a new directory named **“lab1”**.
2. Change directory to **“lab1”**.
3. Create an empty file in **“lab1”** directory named **“lab1.txt”**.
4. Create a new subdirectory inside **“lab1”** named **“sublab1”**.
5. Write the sentence **“I should master the Linux commands learned in lab”** in **“lab1.txt”**.
6. Print the content of **“lab1”** to **stdout**.
7. View content of **“lab1.txt”**.
8. Print the current working directory.
9. Print the number of lines, words and characters in **“lab1.txt”**.
10. Print the content of **“lab1”** to a file named **“newlab1.txt”**.
11. Copy the content of **“lab1.txt”** to **“newlab1.txt”**.
12. Move **“lab1.txt”** to **“sublab1”**.
13. Rename **“newlab1.txt”** to **“lab1_new.txt”**.
14. Copy **“lab1_new.txt”** from **“lab1”** to **“sublab1”**.
15. Display the first 10 lines of **“lab1_new.txt”.**
16. Display the last line of **“lab1_new.txt”.**
17. Display all lines of **“lab1_new.txt”** using a command that doesn’t read the whole file before starting.

Searching

18. Create a new file named **“test.txt”** containing the sentence **“hello world”**.
19. Concatenate files **“test.txt”** and **“lab1_new.txt”** into a file **“concatenate.txt”.**
20. Search for **“lab1.txt”** and delete it. (Hint: read the man page of **“find”** command).
21. Search for all **“txt”** files, find the file that contains the word **“hello”**, and count only the number of characters in the file. (Hint: read the man page of **“grep”** command).
22. Search for the word “master” in “**lab1_new.txt”** using the command “less”**.**
23. Issue the following command “yes “master” | head -$(($RANDOM % 100 + 1)) >> lab1_new.txt”. Find the number of occurances of the word “master” in **“lab1_new.txt”**.

Process Information

24. List all currently running processes in full format listing.
25. List all currently running processes owned by the logged in user.

26. List the details of one process using its PID.
27. Count the number of all running processes.
28. Count the number of all running processes including Light Weight Processes (threads).
29. List the number of threads running per a specific process (use same PID of question c).
30. Print only the process details of “bash”.
31. Print the running Linux version.
32. Print the current Linux Kernel version.
33. Display what user you are logged in as.

Permissions

34. Create a file called “secret.txt” containing “I am a secret file”.
35. Change the permission of this file so that no user can write to it.
36. Use “ls” to show the permissions of this file.
37. Try to add some text to “secret.txt” using “cat” command.
38. Change the permission of the file so that no use can read from it.
39. Use “ls” command to show the permissions of this file.
40. Try to read the file.

Extras

41. Create a new file named “names.txt”, type in some names you know and save it. After doing so, using the “sort” command, output a new file named “names_sorted.txt” that contains the list of names sorted alphabetically.
42. Use “names.txt” and “names_sorted.txt” that you created previously to display all names in both files that contain the letter “a”. The output should be sorted alphabetically and printed to stdout without using a temporary file. (Hint: use pipes).
43. Create 3 empty files named “first_file.txt”, “first_file_2.txt”, “mouse_file.txt”, and “house_file.txt” using 1 command line. List all files ending with “txt”. List all files starting with “first”. List all files containing “file”. List all files containing the word “ouse”.
44. Run the command “sleep” for 1000 seconds in the background. List the processes running in the background only.
45. Make sure the sleep command is still running in the background, if not, just start it again and then kill it using its PID.
46. Run the sleep command in the background again for 1000 seconds and terminate it using its job number.
47. Show a list of all the files in a classified manner according to the data they contain.
48. Use the “echo” command to append “Show the differences between files” to “first_file.txt” and append “display the difference between files” to “first_file_2.txt”. Compare the contents of the 2 files and display the result.
49. Substitute the word “your” by “you’re” in the following sentence: “your my favorite cousin”. Use the “echo” command piped with “sed”.
50. Repeat the same process of the above question but using files this time. Save “your my favorite cousin” to a text file and do the substitution. Make sure the result is saved to the same file using and extra switch to the “sed” command.

51. Create a hard link using the “ln” command to one of the text files. List the contents of both files. Delete the original file and list the contents of the link. Now create a soft link using the “ln –s” command to one of the files and repeat the process.