Assignment 2 Links, file commands, processes, pipelines

Command Sequence A

$ pwd

/home/student/assignments/homework1

$ ls ../../dir2/dir4

abcfile3.log myabcfile4.txt fileABC.log

$ ls -F ~/dir2

dir4/ filebX.log file2.txt fileABC.txt file3R.log baseFile.txt data.ile

$ ls $HOME/dir1

file21.txt file22.txt

1. Given command sequence A, without changing the current working directory, write the commands to:

a. Make a hard link in dir1 that links to baseFile.txt (in its current directory). Call it hardlink.txt

ln ~/dir2/baseFile.txt ~/dir1/hardlink.txt

b. Make a soft link in dir1 that links to baseFile.txt (in its current directory). Call it softlink.txt

ln -s ~/dir2/baseFile.txt ~/dir1/softlink.txt

c. List the contents of dir1 in the long format, include inode numbers in the output.

ls -il ~/dir1

2. If you then delete dir2/baseFile.txt what happens when you enter these commands:

a. cat ~/dir2/baseFile.txt

Errored, no file found

b. cat ~/dir1/softlink.txt

Errors again, no symbolic link found

c. cat $HOME/dir1/hardlink.txt

Did not error, would display contents of baseFile.txt

3. True or False?

a. A soft link can link to a directory

True

b. A soft link has the same inode number as the file it links to

False

c. A soft link can only link to a file on the same physical drive

False

d. A hard link has the same inode number as the file it links to

True

e. A hard link can only link to a file on the same physical drive

True

f. A hard link can link to a directory

False

g. A soft link can link to a hard link

True

4. Give the command that you would use to:

a. List the contents of the directory one level above the current working directory

ls ..

b. List the contents of the directory two levels above the current working directory

ls ../..

c. Given that the last command showed that the directory two levels above contains a directory called tmp, give the command to list the contents of tmp (without changing the current directory).

ls ../../tmp

5. Given command sequence A write the commands that you would use to:

a. Move the working directory up two levels

cd ../..

From that new working directory:

b. List all the .txt files in dir2

ls dir2

c. List all the files in dir2 that have a single uppercase character immediately before the dot. (fileX.log should be in the list, fileABC.txt should not)

ls dir2 | grep -E ^[^A-Z]\*[A-Z][^A-Z]\*$

d. List all the files in dir2 that contain the letter ‘b’ or 'B'.

ls dir2 | grep [bB]

e. Delete all the files in dir2 that contain the string “ile”

rm dir2/\*ile\*

f. Delete dir2 and all its files and sub-directories

rm -r dir2

g. Delete all the files in dir1 but keep the dir1 directory

rm dir1/\*

h. Copy all the log files in dir4 to dir1

cp dir2/dir4/\*.log dir1

\*\*\*Except, from the previous code, dir2 was deleted and so dir4 was deleted, so this code would error because the directories do not exist

6. Which parts of Linux control processes? Pick all that apply

a. The /proc directory files

Yes

b. The GNU system utilities

No

c. The kernel

Yes

d. The system call interface

No

e. The CPU

No

f. The Bash shell

No

7. Write the command to list all the processes on the system

a. Using standard (Unix style) options

ps -ef

b. Using BSD style options

ps aux

8. Controlling processes

a. Write the commands to temporarily stop process number 2343. There are two ways of doing this, show both.

a1. First way:

kill -STOP 2343

a2. Second way:

kill -s STOP 2343

b. Write the command to continue process number 2343

kill -CONT 2343

c. Write the command to immediately stop process number 2343

kill -KILL 2343

9. Command pipeline

Given that ~/dir9 contains several log files with names ending in ".log"

Write a command pipeline that does the following:

- Concatenates the contents of all the .log files in dir9 (without using a file)

- Sorts the resulting combined log

- Adds the sorted result to the end of a file called total.dat

cat ~/dir9/\*.log | sort >> total.dat

10. What is the output of the last command in this sequence?

$ echo “Baked Apple” >> fruit.txt

$ echo “Sliced Banana” >> fruit.txt

$ echo “Juiced Elderberry” > fruit.txt

$ echo “Candied Damson” >> fruit.txt

$ echo “Glace Cherry” > fruit.txt

$ echo “Grilled Grapefruit” >> fruit.txt

$ echo “Dried Fig” >> fruit.txt

$ < fruit.txt grep “i” | sort > sortedNames

$ cat sortedNames

Dried Fig

Grilled Grapefruit