Assignment 4 More Bash Commands

CS2080 Programming with Unix

1. Permissions

1.a What is the octal equivalent of this file’s permission block?

lrwxrw-r--. 1 jill domain-users 10240 Sep 15 10:35 script.sh -> script\_1.sh

rwx (7) rw- (6) r—(4) = 764

1.b What type of file is script.sh?

Symbolic link or soft link

1.c What command would you use to allow group members to execute the file script.sh but leave all other permissions as before? Use character codes (symbolic mode).

chmod g+x script.sh

1.d What command would you use to set these permissions for the file script.sh (over-riding all previous permissions)? Use octal values (numeric mode).

- user: read, write, execute

- group: read, execute

- others: no permissions

chmod 750

1.e Write the command you would use to change the owner of script.sh to kris. Assume that you are logged in as a normal user.

sudo chown kris script.sh

1.f Write the command you would use to change the group owner of script.sh to admin. Assume that you are logged in as a normal user.

sudo chgrp admin script.sh

1.g The permissions block for this file has an octal equivalent of 754, what should the characters in the permissions block be?

-??????????. 1 Edda domain-users 2400 Sep 17 12:22 script9.sh

What goes here?

-rwxr-xr--

Command Sequence A

We’ll use this sequence of commands as a starting point for several questions.

$ pwd

/home/joe/data

$ ls ../logs

data1.txt data1.log dataA.txt data.log dirA dirQ dirR

$ ls ../logs/dir\*

logs/dirA:

logs/dirQ:

dataX.log dataXY.log dataY.log

logs/dirR:

A.txt B.txt C.txt A.log B.log C.log

$

2. Command quoting and aliases

Given Command Sequence A above, and assuming that the commands are executed without changing the working directory shown at the top, what do you expect the output of the three last lines to be?

1: $ alias print\_result='cat result'

2: $ rm result

3: $ start=../logs

4: $ ls $start/dirQ/\*[!Xx]\* > result; print\_result # 2.a

5: $ ls '$start/dirR[!Xx]\*' > result; print\_result # 2.b

6: $ ls $start/dir\*/\*[g]\* > result; print\_result # 2.c

2.a Output from line 4

dataX.log dataXY.log dataY.log

2.b Output from line 5

Error, no such file or directory

2.a Output from line 6

dataX.log dataXY.log dataY.log A.log B.log C.log

3. Compression and archiving

3.a Write the command to make an archive file containing every .log file in dirR. Call the archive file dirR.tar and put it in logs/dirA.

tar -cf ../logs/dirA/dirR.tar ../logs/dirR/\*.log

3.b Write the command to list the contents of dirR.tar. (use tar options)

Tar -tf ../logs/dirA/dirR.tar

3.c Write the commands to extract the contents of dirR.tar to a new directory:

logs/backup. (you need mkdir first)

mkdir ../logs/backup

tar -xf ../logs/dirA/dirR.tar -C ../logs/backup

4. Variables

4.a This command has three errors:

$ $answer = 42

- Re-write the command without the errors

answer=42

4.b This command has two errors

$ message\_string= Here is the message

- Re-write the command without the errors

Message\_string=”Here is the message”

4.c Given that the variables above have been defined correctly, what will be the output of these commands?

$ echo answer message\_string

answer message\_string

$ echo $answer $message\_string

42 Here is the message

4.d When executing this sequence of commands:

1: $ test\_string=Test\_test\_test

2: $ echo $test\_string

3: Test\_test\_test

4: $ bash

5: $ echo $test\_string

6: # What is the output here?

7: $

4.d.1 What do you expect the output on line 6 to be?

Assuming line 6 is supposed to have $ in front of test\_string:

Nothing, its blank

4.d.2 Re-write line 1 so that line 6 works

export test\_string=Test\_test\_test

4.f Given these commands:

$ string\_3=Hello

$ string\_4="how"

$ string\_5="are

> you

>?"

$

Write the command to append the contents of string\_4 and string\_5 to the

string\_3. Note: You must use the string\_4 and string\_5 variables in your answer.

string\_3=”$string\_3, $string\_4 $string\_5”

4.g Given Command Sequence A, write commands to create two local variables

containing the absolute paths to:

- the existing dirQ in the /home/joe directory tree, call it DIR\_Q

- /var/log in the root directory, call it VAR

DIR\_Q=”/home/joe”

VAR=”/var/log”

4.h Using the variables DIR\_Q and VAR, write commands to:

mkdir $DIR\_Q/log\_data

cp $(grep -lrw -e "messages" -e "mail" $VAR) "$DIR\_Q/log\_data"

5. Viewing Processes

5.a Using BSD-style parameters (no -), write the command to display all the processes for all users, in user oriented format, including processes.

ps aux

5.b Write the same command using Unix-style parameters

Ps -ef

6. Controlling Jobs

1: $ jobs -l

2: [1] 341 Running { sleep 500; } &

3: [2]+ 365 Stopped sleep 500

4: [3]- 416 Running sleep 500 &

5: $

6.a Which of the jobs was not started as a background job?

The second job because it missing the & symbol

6.b Write the command to immediately kill job 3, use the job number in the command

Kill -9 416

6.c Write the command to re-start job 2, use the job number in the command

fg %2

6.d Write the command bring job 1 into the foreground, use the job number in the command

fg %1

7. Controlling Processes: use kill -l to see all signals and the numbers associated with them

Refer to this link for more info:

$ kill 35249 -18

7.a The command above has an error, write the command correctly

kill -18 35249

7.b What will the corrected command do?

Sends the SIGCONT signal to all processes in the same 35249 process group, and resumes any that were stopped

7.c Rewrite the command using a signal name rather than a number.

kill -SIGCONT 35249

7.e What kill command must have been sent before the kill command above was sent? There are two possibilities:

7.e.1

kill -19 35249 or

kill -SIGSTOP 35249

7.e.2

kill -20 35249

kill -SIGTSTP 35249

7.f When you press Ctrl-c,

7.f.1 What signal is sent?

Nicely kill/interrupt the process

Kill -SIGINT sigNumber

7.f.2 What process is it sent to?

The process in the foreground

7.f.3 What does that process do?

It interrupts and terminates the process

7.g When you press Ctrl-z,

7.g.1 What signal is sent?

Paused process and sends it to background in a stopped state

Kill -SIGTSTP sigNumber

7.g.2 What process is it sent to?

Current foreground process

7.g.3 What command can you use to reverse the effect of Ctrl-z (other than a

kill command)?

fg

8. The ls Command

$ ls \*[data,LATER,potata]\*

This command is supposed to find any file that has one of the strings “data” or “LATER” or "potata", anywhere in its name. It is not working. Re-write the command so it works.

ls \*{data,LATER,potata}\*

9. The grep Command

$ grep "hello", "HELLO" valFile.txt

This command is intended to find any of the strings HELLO or hello, in

valFile.txt. It returns an error. Re-write the command so that it works.

grep -E “hello|HELLO” valFile.txt

10. More grep

Write a sequence of commands that:

Finds all the files in the current directory with “Data” or “DATA” in their

names(ignores the case of the letters)

From that list of files, find the files that contain the string “time" or "TIME" and return just the names of the files.

grep -wli “time” $(ls | grep -i “data”)

11. More grep

Write a command pipeline that:

Passes the string “wqv qvd 9q3 93 qv wqv 9393 93qv 93 qv”

- into a grep command that matches any whole words that match one of

these words: qv wqv 93

- then sorts the result

- then writes the sorted list into a file called output.txt

Note: The grep command must output only the matching words

echo “wqv qvd 9q3 93 qv wqv 9393 93qv 93 qv “ | grep -Ewo “qv|wqv|93” | sort > output.txt

12 More grep

1: $ pwd

2: /home/student/dir12

3: $ ls

4: goodbyeFile.txt helloFile.txt data.txt numbers.txt

5: $ ls \*.txt | grep "hello"

6: # <== What is output here?

12.a What will be output on line 6?

helloFile.txt

12.b Explain why grep produces that output

First the command line lists at files in dir12 that end with .txt then that output in not the input for grep which it looking at the input for text with the string hello, which only helloFile.txt has

13 Disk space

Write the command to print the sizes (in human-readable form) and names of the ten

largest files in the current user’s home directory. (hint: needs du, sort and head)

Requirement

-Do not use the ls command.

Hints: sort -h gives a human-numeric-sort e.g., it compares human-readable numbers such as 2K, 1G

-sort outputs results in ascending order, find the option to reverse that.

du -h ~ | sort -hr | head -n 10