Assignment 3 grep, glob, filesystems, braces

CS2080-002 Programming with Unix

1. Pipelines

Write a pipeline of commands to:

a. Find the names of all the files in the /etc directory and its subdirectories

with lines containing either of these words: grep sed.

grep -lrw -e "grep" -e "sed" /etc 2>/dev/null

b. Then sort those file names

| sort

c. Then write the last 15 sorted names to grepsedfiles.txt

| tail -n 15 grepsedfiles.txt

d. All together

grep -lrw -e "grep" -e "sed" /etc 2>/dev/null | sort | tail -n 15 grepsedfiles.txt

2. File globs

Write separate commands to:

d. List all .log files in the parent of the current directory whose names are exactly 4

characters long (excluding the .log at the end of the name). For example: you

should list abcd.log but not pqr.log or longname.log

ls ../????.log

e. List all the files in the parent of the parent of the current directory whose name does not end in .dat (in other words, everything but .dat files)

ls ../../\* 2>/dev/null | grep -v “.dat$”

3. Command lists and process lists

a. Execute a command to print the current process ID and store it in a variable called current\_process\_id.

current\_process\_id=$$ or current\_process\_id=$(echo $$)

b. Execute this sequence of commands in a COMMAND LIST and write the output

to the file called output.txt. Here’s the sequence:

echo “Running process $$”

echo “goodbye”

{ echo “Running process $$”; echo “goodbye”} > output.txt

c. Display the contents of output.txt once execution has finished.

cat output.txt

d. Is the process ID printed inside the list the same as current\_process\_id?

It is the same number

e. Why is that?

Because $$ is a variable that stores the current process ID , and this command list is still a part of the same process

f. Execute the same command sequence in a PROCESS LIST and write the output to

the file called output2.txt.

( echo “Running process $$”; echo “goodbye”) > output2.txt

g. Display the contents of output.txt once execution has finished.

Cat output2.txt

h. Is the process ID printed inside the list the same as current\_process\_id?

Yes

i. Why is that?

Because like the command list, it is still a part of the same process as before

4. Jobs

You run the jobs command and see the output below.

$ jobs -l

[1] 32841 Running while true; do sleep 300 ; done

[2]- 32897 Running while true; do sleep 300 ; done

[3]+ 35944 Running while true; do sleep 300 ; done

a. Which job was started last?

The third job

b. Write the command to stop but not terminate the job that started first.

Kill -STOP 32841

c. Write the command to resume the job that you just stopped.

Kill -CONT 32841

d. Write the command to bring job 1 into the foreground

fg 1

5. Brace Expansion

a. What would the output of the last command in this sequence be?

$ mkdir ~/test1

$ cd $HOME/test1

$ touch data{“91”,”91backup”,"92real”}.log

$ touch {“green”,“red”}.{"txt","doc"}

$ ls # <== what is the output?

data91backup.log data91.log data92real.log green.doc green.txt red.doc red.txt

Given this command sequence:

$ mkdir ~/test2

$ cd ~/test2

$ touch raw\_data{"1","2"}.dat

$ touch input{"1","\_data2","3"}.dat

$ touch test{"2","3"}.dat

b.1 Write a command to list all the files that have data or test in their names. Use one ls command and one file glob with brace expansion

ls \*{“data”,”test”}\*

Now do the same thing a different way:

b.2 Write a command to list all the files that have data or test in their names.

Use one ls command and two separate file globs (no brace expansion).

ls \*data\* \*test\*

6. Aliasing

Assume you have just run the commands in question 5b. If you now run this

sequence of commands, what would the output of the last command be?

$ cd ~/test2

$ alias ll='ls \*[12]\*'

$ alias ss='sort'

$ alias hh='head -4'

$ ll | ss | hh <== what is the output?

7. Variables

a. What variable contains the process ID of:

i. the current process?

$$

ii. the parent of the current process?

PPID

b. Write the command you would use to create a variable called current\_process\_id and set its value to the ID of the current process

current\_process\_id=$$

c. Write the command you would use to create a variable called hello\_message and set its value to the string “Hello from the parent shell”

hello\_message=$(echo “Hello from the parent shell”)

d. Create a child process by starting a new instance of the Bash shell from the

current shell. Inside that child process, print the value of hello\_message. Then exit

the child process.

$ bash

$ # Add code to print value of hello\_message

????? # The output will be here, what do you see?

$ exit # Exit the child process

What value of hello\_message was printed?

It was blank

e. Back in the parent Bash shell, what command must you use to make the

hello\_message variable visible in child processes?

Export hello\_message

f. Once you have executed that command, repeat the steps in part d. What value

of hello\_message is now printed in the child process?

Hello from the parent shell

8. The PATH variable

The PATH environment variable contains a list of directories that Bash uses as a search

path when looking for programs.

1. What does the PATH variable on blanca currently contain?

usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/snap/bin

1. What character is used to separate each directory in the path?

It uses a : to separate paths

c. Say you want to add the directory ~/bin to the end of the PATH variable. Show

two ways of doing that:

c.1 First way

export PATH=”$PATH:$HOME/bin”

c.2 Second way

export PATH=”$PATH:~/bin”

9. Command substitution

a. Assume you have run the commands in question 5a.

Write one command that:

- lists the names of the files in ~/test2 whose names include the letters re and puts the resulting list into a variable called test2\_re\_files

test2\_re\_files= $(ls ~/test/\*{r,e}\* )

1. Write the command you would use to show the value of test2\_re\_files

Echo $test2\_re\_files

1. What do you expect the output of that command to be?

Raw\_data1.dat raw\_data2.dat test2.dat test3.dat

10. Redirection

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

/$ cd

$ mkdir dir9

$ cd dir9

$ fileA=”file1.txt”

$ fileB=”file2.txt”

$ fileC=”file3.txt”

$ echo “This is file A” > $fileA

$ echo “This is file 2” > $fileB

$ echo “This is file C” > $fileC

$ cp $fileC $fileA

$ mv $fileB $fileC

$ cat $fileA $fileB $fileC <== what is the output?

This is file C

Error, no file, was renamed

This is file 2

11. Disk space

You are running a program on a Linux machine, and it fails with this error: “No space left

on device”. Write the command you would use to:

1. Find out which drive is running out of space.

df -h

1. Find out which files on that drive are consuming the space.

du -ah / 2>/dev/null | sort -hr | head -n 10

Note: make the commands output their results in human readable format

12. Command quoting

Given this command sequence, what will the last two commands output?

$ cd ~/dir9

$ ls

hello1.txt hello2.txt hello3.txt other.txt

$ VAR1=hello

$ VAR2=\*.txt

$ VAR3=$VAR1$VAR2

$ VAR4='$VAR1$VAR2'

$ echo $VAR3

#<== 12.a What is output here?

$ echo $VAR4

#<== 12.b What is output here?

1. Command output:

Hello1.txt hello2.txt hello3.txt

1. Command output:

$VAR1$VAR2