2. What are two ways you can execute a shell script when you do not have execute permission for the file containing the script?

bash test.sh

source test.sh

. ./test.sh

 Can you execute a shell script if you do not have read permission for the file containing the script?

No, permission denied

3. What is the purpose of the PATH variable?

The **$PATH** variable is determines where the shell is looking for commands to execute. This variable contains a list of directories, separated by colon.

a. Set the PATH variable and place it in the environment so it causes the shell to search the following directories in order:

• /usr/local/bin

• /usr/bin

• /bin

• /usr/kerberos/bin

•The bin directory in your home directory

• The working directory

b. If there is an executable file named doit in /usr/bin and another file with the same name in your ~/bin directory, which one will be executed?

c. If your PATH variable is not set to search the working directory, how can you execute a program located there?

set the PATH .

d. Which command can you use to add the directory /usr/games to the end of the list of directories in PATH?

export PATH =$PATH: /usr/games

4. Assume you have made the following assignment:

$ person=zach

Give the output of each of the following commands.

a. echo $person

zarch

b. echo '$person'

$person

c. echo "$person"

zarch

10 mins

5. The following shell script adds entries to a file named journal-file in your home directory. This script helps you keep track of phone conversations and meetings.

Click here to view code image

$ cat journal

# journal: add journal entries to the file

# $HOME/journal-file

file=$HOME/journal-file

date >> $file

echo -n "Enter name of person or group: "

read name

echo "$name" >> $file

echo >> $file

cat >> $file

echo "----------------------------------------------------" >> $file

echo >> $file

a. What do you have to do to the script to be able to execute it?

#!/bin/bash

b. Why does the script use the read builtin the first time it accepts input from the terminal and the cat utility the second time?

6. Assume the /home/zach/grants/biblios and /home/zach/biblios directories exist. Specify Zach’s working directory after he executes each sequence of commands. Explain what happens in each case.

a. $ pwd

/home/zach/grants

$ CDPATH=$(pwd)

$ cd

$ cd biblios

b. $ pwd

/home/zach/grants

$ CDPATH=$(pwd)

$ cd $HOME/biblios

7. Name two ways you can identify the PID number of the login shell.

echo $$

ps

8. Enter the following command:

Click here to view code image

$ sleep 30 | cat /etc/services

Is there any output from sleep? Where does cat get its input from? What has to happen before the shell will display a prompt?

There is no output from sleep . The /etc/services file provides input for cat . The sleep command has to run to completion before you get command line prompt.

ADVANCED EXERCISES

9. Write a sequence of commands or a script that demonstrates variable expansion occurs before pathname expansion.

10. Write a shell script that outputs the name of the shell executing it.

echo $0

11. Explain the behavior of the following shell script:

This is related to IFS.

$ cat quote\_demo

twoliner="This is line 1.

This is line 2."

echo "$twoliner"

echo $twoliner

a. How many arguments does each echo command see in this script? Explain.

echo "$twoliner" - two lines

echo $twoliner - single line

b. Redefine the IFS shell variable so the output of the second echo is the same as the first.

12. Add the exit status of the previous command to your prompt so it behaves similarly to the following:

Click here to view code image

$ [0] ls xxx

ls: xxx: No such file or directory

$ [1]

13. The dirname utility treats its argument as a pathname and writes to standard output the path prefix—that is, everything up to but not including the last component:

$ dirname a/b/c/d

a/b/c

If you give dirname a simple filename (no / characters) as an argument, dir-name writes a . to standard output:

$ dirname simple

.

Implement dirname as a bash function. Make sure it behaves sensibly when given such arguments as /.

14. Implement the basename utility, which writes the last component of its pathname argument to standard output, as a bash function. For example, given the pathname a/b/c/d, basename writes d to standard output:

$ basename a/b/c/d

d

15. The Linux basename utility has an optional second argument. If you give the command basename path suffix, basename removes the suffix and the prefix from path:

Click here to view code image

$ basename src/shellfiles/prog.bash .bash

prog

$ basename src/shellfiles/prog.bash .c

prog.bash