Vanier College

Faculty of Careers and Technical Programs

Department of Computer Science Technology

Advanced UNIX

Lab #7

Title: Working with the BASH Shell

Student Names: Marissa Gonçalves
Amy Yip
Yulia Bakaleinik

Submitted to Florin Pilat

November 6, 2020

Review Questions *(p.336-339)*:

1. False
2. C. exported
3. False
4. A. /etc/profile
5. C. `set`
6. D. `fi`
7. A. `cd /home/user1 && echo "welcome home"`
8. B. `echo ~`
   C. `echo $HOME`
9. C. 1
10. D. !
11. C. When you use the `cat` command at the command prompt with the intention of viewing a text file, the date appears instead.
12. D. Begin the line with #.
13. C. You did not append the Standard Error to the Error file, and as a result it was overwritten when the command was run a second time.
14. True
15. A. Nothing is wrong with the command.
16. A. Standard Input
    B. Standard Deviation
17. A. `read`
18. True
19. D. It appends Standard Output to a file.
20. B. The answer is not red nor blue.

Project 7.1 *(p.339-340)*:

1.

```
Fedora 26 (Workstation Edition)
Kernel 4.16.11-100.fc26.x86_64 on an x86_64 (tty2)

server13 login: root
Password:
Last login: Thu Oct 22 17:17:31 on pts/0
[root@server13 ~]#
```

2.

```
[root@server13 ~]# touch sample1 sample2
[root@server13 ~]# ls -F
aclfile  anaconda-ks.cfg  directory/  file*  sample1  sample2
```

3. There is a Standard Output shown for the first two newly created files since "sample1" and "sample2" exist in the home directory (~). In addition, there is also one Standard Error displayed, since the "sample3" file doesn't exist in the home directory (~).

```
[root@server13 ~]# ls -l sample1 sample2 sample3
ls: cannot access 'sample3': No such file or directory
-rw-r--r--. 1 root root 0 Nov  5 11:25 sample1
-rw-r--r--. 1 root root 0 Nov  5 11:25 sample2
```

4. There is no Standard Output displayed since the output is redirected to "file" due to the > operand, however, there is one Standard Error since "sample3" still doesn't exist in the home directory (~).

```
[root@server13 ~]# ls -l sample1 sample2 sample3 > file
ls: cannot access 'sample3': No such file or directory
```

5. The contents of "file" include the output for"sample1" and "sample2" since the output has been redirected to "file" based on the previous `ls` command with the > operand.

```
[root@server13 ~]# cat file
-rw-r--r--. 1 root root 0 Nov  5 11:25 sample1
-rw-r--r--. 1 root root 0 Nov  5 11:25 sample2
```

6. There is a Standard Output still showing the first two files, however, there is no Standard Error since the error message for “sample3” has been redirected to “file” due to the 2> operand.

```
[root@server13 ~]# ls -l sample1 sample2 sample3 2>file
-rw-r--r--. 1 root root 0 Nov  5 11:25 sample1
-rw-r--r--. 1 root root 0 Nov  5 11:25 sample2
[root@server13 ~]# _
```

7. The contents of ‘file” display the “sample3” file error message since they have been redirected to “file” based on the preceding command with the 2> operand. In addition, the previous contents of “file” have not been retained, since a new file was created to redirect the error message instead.

```
[root@server13 ~]# cat file
ls: cannot access 'sample3': No such file or directory
```

8. There is no Standard Output nor Standard Error displayed on the screen, since the output is redirected to “file” and the error message is redirected to “file2” due to the > and 2> operands.

```
[root@server13 ~]# ls -l sample1 sample2 sample3 > file 2>file2
[root@server13 ~]# _
```

9. The contents of “file” include the output since the previous `ls` command with the > and 2> operands redirected the output to “file”.

```
[root@server13 ~]# cat file
-rw-r--r--. 1 root root 0 Nov  5 11:25 sample1
-rw-r--r--. 1 root root 0 Nov  5 11:25 sample2
```

10. The contents of “file2” include the error message since the previous `ls` command with the > and 2> operands redirected the error message to “file2”.

```
[root@server13 ~]# cat file2
ls: cannot access 'sample3': No such file or directory
```

11. There is no Standard Output nor Standard Error displayed on the screen, since the output and error message is redirected to “file” due to the 2>&1 operand.

```
[root@server13 ~]# ls -l sample1 sample2 sample3 > file 2>&1
[root@server13 ~]# _
```

12. The contents of “file” include the error message and output since the preceding `ls` command with the > and 2>&1 operands redirected the error message and output to “file”.

```
[root@server13 ~]# cat file
ls: cannot access 'sample3': No such file or directory
-rw-r--r--. 1 root root 0 Nov  5 11:25 sample1
-rw-r--r--. 1 root root 0 Nov  5 11:25 sample2
```

13. The Standard Output for the “sample1” and “sample2” files is displayed on the screen and not the Standard Error since the error message for “sample3” is redirected to “file2” due to the >&2 and 2> operands.

```
[root@server13 ~]# ls -l sample1 sample2 sample3 >&2 2>file2
-rw-r--r--. 1 root root 0 Nov  5 11:25 sample1
-rw-r--r--. 1 root root 0 Nov  5 11:25 sample2
```

14. The contents of “file2” include the error message from “sample3” since the previous `ls` command with the >&2 and 2> operands redirected the error message to “file2”.

```
[root@server13 ~]# cat file2
ls: cannot access 'sample3': No such file or directory
```

15.

```
[root@server13 ~]# date >> file
[root@server13 ~]# _
```

16. The contents of “file” include the “sample3” error message, file list output, as well as the date and time since the output from the previous `date` command appended to “file” with the >> operand.

```
[root@server13 ~]# cat file
ls: cannot access 'sample3': No such file or directory
-rw-r--r--. 1 root root 0 Nov  5 11:25 sample1
-rw-r--r--. 1 root root 0 Nov  5 11:25 sample2
Thu Nov  5 13:26:11 EST 2020
```

17.

```
[root@server13 ~]# date >> file
[root@server13 ~]#
```

18. The contents of “file” include the “sample3” error message, file list output, as well as two dates and timestamps since an additional output from another preceding `date` command with the >> operand appended to “file” once more. According to the contents from “file”, the first `date` command was executed on November 5th, 11 seconds after 1:26 P.M., while the second `date` command was executed on November 5th, 31 seconds after 1:27 P.M.

```
[root@server13 ~]# cat file
ls: cannot access 'sample3': No such file or directory
-rw-r--r--. 1 root root 0 Nov  5 11:25 sample1
-rw-r--r--. 1 root root 0 Nov  5 11:25 sample2
Thu Nov  5 13:26:11 EST 2020
Thu Nov  5 13:27:31 EST 2020
```

19. We have received an error message since an extra operand is needed for the command to execute successfully and indicate that the replacement should take place in the “hosts’ file.

```
[root@server13 ~]# tr o 0 /etc/hosts
tr: extra operand ‘/etc/hosts’
Try 'tr --help' for more information.
```

20. Once this command has been executed, all lowercase “o”s are replaced with uppercase “O”s and redirected to the specified file, as shown in this output displaying the contents of the “hosts” file in the /etc directory.

```
[root@server13 ~]# tr o O </etc/hosts
127.0.0.1   lOcalhOst lOcalhOst.lOcaldOmain lOcalhOst4 lOcalhOst4.lOcaldOmain4
::1         lOcalhOst lOcalhOst.lOcaldOmain lOcalhOst6 lOcalhOst6.lOcaldOmain6
```

21.

```
[root@server13 ~]# exit
```

```
Fedora 26 (Workstation Edition)
Kernel 4.16.11-100.fc26.x86_64 on an x86_64 (tty2)

server13 login: _
```

Discovery Exercises *(p.347-348)*:

2.

```
[marissa@server13 ~]$ cp /etc/hosts ~
[marissa@server13 ~]$ cd
[marissa@server13 ~]$ tr a A <hosts | sort -r | pr -d >hosts
```

The first command copies the “hosts” file from the /etc directory to the home directory (~). The second command changes the directory to the home directory (~). Finally, the third command replaces all lowercase “a”s from the “hosts” file with capital “A”s, reverse-sorts the “hosts” file alphanumerically and formats the “hosts” file double-spaced. In addition, the pipes (|) in the third command are used to redirect the Standard Output of one command to another command as a Standard Input. So, the Standard Output of the `tr` command is redirected as a Standard Input to the `sort` command, as well as the Standard Output of the `sort` command being redirected as a Standard Input to the `pr` command.

6.

```
[marissa@server13 ~]$ vi tasks
```

```
#! /bin/bash
# Basically, this file displays a list of currently logged-in users,
# the system's host name, time and date, disk usage, the current
# working directory and the pathname to the BASH shell.
echo "6.a) Current Users on the System: "
who
echo "b) System's Host Name: "
hostname
echo "c) Time and Date: "
date
echo "d) Disk Usage: "
df
echo "e) Current Working Directory: "
pwd
echo "f) Pathname to the BASH Shell: "
echo $BASH
~
~
~
~
~
~
"tasks" 17L, 440C
```

```
[marissa@server13 ~]$ chmod +x tasks
```

```
[marissa@server13 ~]$ ./tasks
6.a) Current Users on the System:
marissa  tty2         2020-11-05 14:41 (/dev/tty2)
b) System's Host Name:
server13.domain.com
c) Time and Date:
Thu Nov  5 14:51:24 EST 2020
d) Disk Usage:
Filesystem             1K-blocks    Used Available Use% Mounted on
devtmpfs                 2524872       0   2524872   0% /dev
tmpfs                    2537156       0   2537156   0% /dev/shm
tmpfs                    2537156    1600   2535556   1% /run
tmpfs                    2537156       0   2537156   0% /sys/fs/cgroup
/dev/mapper/fedora-root  13759208 5522892   7517676  43% /
/dev/sda1                  999320  127952    802556  14% /boot
tmpfs                    2537156     148   2537008   1% /tmp
tmpfs                     507428      28    507400   1% /run/user/42
tmpfs                     507428    4664    502764   1% /run/user/1000
e) Current Working Directory:
/home/marissa
f) Pathname to the BASH Shell:
/bin/bash
```