SHAHJALAL UNIVERSITY OF SCIENCE AND TECHNOLOGY

COURSE NAME: OPERATING SYSTEM

COURSE CODE: CSE 336

Date: 31-03-2019

SUBMITTED BY:

NAME: Tapu Das

REG NO: 2016331106

NAME: MD. Mesbah Uddin Waheed

REG NO: 2016331042

NAME: Towhid Ahmed Foysal

REG NO: 2016331044

SUBMITTED TO:

Ms. Ayesha Tasnim

Assistant Professor

Task-1:

Echo command:

just type man echo in Terminal.

Echo: displays a line of text.

Example: echo “Hello World”
Output will printed in new line saying Hello World.

Description:

-n : do not output the trailing new line.

-e : enable interpretation of backlash escapes.

-E : disable interpretation of backlash escapes.

If –e is in effect, following sequences are:

\\ backslash

\a alert

\b backspace

\e escape

\n new line

\t horizontal tab

\v vertical tab

Example:

Input: echo hello world

Output: hello world

Input: echo -n hello world

Output: hello world and you have to give your another command from this same line.

Input: echo -e ‘hello \n world’

Output: hello
world

Input: echo –e ‘hello \t world’

Output: hello world

Input: echo -e ‘hello \v world’

Output: hello
world

Assigning a value to variable:

First assign a value suppose x=10 then press ENTER.

Again
input: echo “the value of x is $x”

Output: the value of x is 10

Task-2:

MAN: an interface to the on line reference manuals.

Inuput: echo $SHELL

Output: Bash is the name of the login shell that is currently use.

Task-3:

Who- who is logged on. Print information about users who are currently logged in.

Who –H: Prints the username, time of login.
who –q: How many use currently logged in and their names.

Who-r: current level user is running. It's usually 3 or 5.

CAT: concatenate files and prints the standard outputs.

Suppose there is two text file animal and people.

Animal	people
Dog	Susan
Cat	Mary

Input: cat Animal people

Output: dog
cat

Susan

Mary

If we want to merge them in a new file and don't want to be overridden then
Input: cat animal people > newfile(new text file)
output: dog
cat

Susan
Mary

If we animal in the same file and want to be overridden then
Input: cat animal > newfile(new text file)
output: dog
cat

Cd: no manual entry for cd but cd is used to enter into folder or directories.

Cp: Files can be moved and copied by this command.

Ps: displays processes currently running in shell.

Ls: **ls** is a Linux shell command that lists directory contents of files and directories.

Display All Information about Files/Directories Using ls -l

Mv: mv is used to move one or more files or directories from one place to another in file system like UNIX. It has two distinct functions:
(i) It rename a file or folder.
(ii) It moves group of files to different directory.

Example: mv text.txt dir/

Rm: rm stands for remove here. rm command is used to remove objects such as files, directories, symbolic links and so on.

Mkdir: it is used to create a directory or folder in Ubuntu.

Rmdir: it is used to remove the folder or directories but if there is anything in the directory the remove operation will be failed

Echo: echo is used to print anything in output.

More: more command is used to view the text files in the command prompt, displaying one screen at a time in case the file is large (For example log files).

Less: Less command is linux utility which can be used to read contents of text file one page(one screen) per time. It has faster access because if file is large, it don't access complete file, but access it page by page.

Date: date command is used to display the system date and time. you can also set the time.

Time: time command in Linux is used to execute a command and prints a summary of real-time, user CPU time and system CPU time spent by executing a command when it terminates.

Kill: *kill* command in Linux (located in /bin/kill), is a built-in command which is used to terminate processes manually.

History: *history* command is used to view the previously executed command.

Chomd: the chmod command is used to change the access mode of a file.

Pwd: pwd stands for Print Working Directory. It prints the path of the working directory, starting from the root.

Cal: If a user wants a quick view of calendar in Linux terminal, cal is the command for you. By default, cal command shows current month calendar as output.

Logout: it logs you out of the terminal.

Task-4:

sed: SED command in UNIX is stands for stream editor and it can perform lot's of function on file like, searching, find and replace, insertion or deletion.

Using sed command to delete the first character and last in each line of a file:
newfile::

My name is susan

I am 13 years old

I read in class nine

sed command: sed 's/^.\(.*\).$/\1/' newfile.txt

Output:

y name is susan

am 13 years ol

read in class nin

grep: The grep filter searches a file for a particular pattern of characters, and displays all lines that contain that pattern.

Newfile:

\

My name is susan

I am 13 years old

I read in class nine

How many times I occurs in the file every line?

Command: grep –c I Newfile.txt

Output: 2

Awk:

(a) Scans a file line by line
(b) Splits each input line into fields
(c) Compares input line/fields to pattern
(d) Performs action(s) on matched lines
Example:
newFile.txt:

User	Marks	Reg No
A,	89,	674674
B,	65,	746584
C,	100,	568347

Command: awk –F "," '{print $2, $3;}' newFile.txt

Output:

Marks	Reg No
89	674674
65	746584
100	568347

Where "," is the separator sign. $2 and $3 means 2nd and 3rd column.

Task-5:

Command: who > myfile (The result of who command is stored in myfile1)

Command: more myfile(By using more command we print the myfile1 contents)

The syntax for the more command is:

more [options] [files]

OPTIONS

Option Description

-c → Page through the file by clearing the window. (not scrolling).

-d → Displays "Press space to continue, 'q' to quit"

-f → Count logical lines rather than screen lines (wrapping text)

-l → Ignores form feed (^L) characters.

-r → Display all control characters.

-s → Displays multiple blank lines as one blank line.

-u → Does not display underline characters and backspace (^H).

-w → Waits for a user to press a key before exiting.

-n → Displays n lines per window.

+num → Displays the file starting at line number num.

+/pattern → Displays the file starting at two lines before the pattern.

T**ask – 6:**

$ date;who > myfile2

□□□□□ □□□□□ 29 20:26:35 +06 2019

$ more myfile2

asif :0 2019-03-29 19:20 (:0)

Task-7:

sed command: ‘s/\([^]*\)*\([^]*\)/\2\1/g'mytable

Task-8:

```
time {
for (( i=1; i<10000; i++ )); do
echo 1 >/dev/null
done
}
```

output:

```
real 0m0.002s
user 0m0.001s
sys 0m0.001s
```

```
#include<stdio.h>
int main()
{
printf("Hello World!");
return 0; }
```

output

real 0m0.285s

user 0m0.123s

sys 0m0.030s

Task-9:

#!/bin/bashcheck_file(){if [-z "${1}"] ;then echo "Please input something" return;fif="${1}"result="$(file $f)"if [[$result == *"cannot open"*]] ;then echo "NO FILE FOUND ($result) ";elif [[$result == *"directory"*]] ;then echo "DIRECTORY FOUND ($result) ";else echo "FILE FOUND ($result) ";fi}check_file ${1}

Output :

$ ./f.bash loginDIRECTORY FOUND (login: directory) $ ./f.bash ldasdasNO FILE FOUND (ldasdas: cannot open `ldasdas' (No such file or directory)) $ ./f.bash evil.php FILE FOUND (evil.php: PHP script, ASCII text)

Task-10:

#!/bin/bash

read -p "What is your first name? " firstname

firstname=${firstname^^}

echo "Hello, ${firstname}."

Output:

asif@asif-HP-EliteBook-Folio-9470m:~/Desktop$ ./hello.sh

What is your first name? steve

Hello, STEVE.

Task-11:

Write a shell script that determine the period for which a specified user is working on system.

Ans :

```
  echo -e "enter the user name :\c"
read usr
tuser=`who | tr -s " " | head -1 | cut -d " " -f1`
if [ "$tuser" = "$usr" ]
then
tm=`who | tr -s " " | head -1 | cut -d " " -f4`
uhr=`echo $tm | cut -d ":" -f1`
umin=`echo $tm | cut -d ":" -f2`
shr=`date "+%H"`
smin=`date "+%M"`
if [ $smin -lt $umin ]
then
shr=`expr $shr - 1`
smin=`expr $smin + 60`
fi
h=`expr $shr - $uhr`
m=`expr $smin - $umin`
echo "user name : $usr"
echo "login period : $h : $m"
else
echo "Inv
```

alid User"

fi

Task 12:

11. Write a Shell script that accepts a filename, starting and ending line numbers as arguments and displays all the lines between the given line numbers.

Ans:

```
echo "enter the filename"

read fname

echo "enter the starting line number"

read s

echo "enter the ending line number"

read n
sed -n $s,$n\p $fname | cat > newline
cat newline
```

Task 13:

write a shell script that deletes all lines containing a specific word in one or more files supplied as arguments to it

Ans:

```
if [ $# -eq 0 ]
then
echo "Please enter one or more filenames as argument"
exit
fi
echo "Enter the word to be searched in files"
read word
for file in $*
do
sed "/$word/d" $file | tee tmp
mv tmp $file
done
```

Task 14- (a)

write a shell script to extract a substring from a given string

```
 for file in *_P1*
do
  lb=${file%%.*}

  pu=${file%%.lane_*}
  pu=${pu#*.}

  num=${file%%_P*}
  num=${num##*_}
```

```
  id="$lb-$pu-$num"

  echo "id=$id pu=$pu lb=$lb"
done
```

Task 14-(b)

write a shell script to know the length of the given string

```
#!/bin/bash
string="$1"

if [ $# -eq 0 ]
then
        echo "Syntax: $(basename $0) string"
        exit 1
fi
length=$(echo ${#string})

echo "String length = $length"
```

command: bash file_name.sh “password”

ouput: String length = 8