

LAB 2: SHELL SCRIPTS

1. Write and Execute the following commands.

Create a file called as sample.txt and enter some text and execute the command

- a) `$wc < sample.txt`
- b) `$wc sample.txt > newfile`
`$cat newfile`
- c) `$(ls -x *.c ; echo; cat *.c) > c_progs_all.txt`

Do not create a file called sample1 and execute the following command.

- a) `$cat sample1 2>errorfile`
- b) `$cat sample1 2>>errorfile`

Pipes.

- a) `$who > user.txt`
`$cat user.txt`
- b) `$wc -l < user.txt`
- c) `$who | wc -l`
- d) `$ls | wc -l`
- e) `$ls | wc -l > fcount`

Command substitution

- a) `$echo The date today is `date``
- b) `$echo "There are `ls | wc -l` files in the current directory"`
- c) `$echo 'There are `ls | wc -l` files in the current directory'`

Shell variables

- a) `$count=5`
- b) `$echo $count`
- c) `$total=$count`
- d) `$echo $total`

Effects of Quoting and Escaping

- a) `message=You\ didn\'t\ enter\ the\ filename`
`$echo $message`
`message="You didn't enter the filename"`
`$echo $message`
- b) `$echo "The PATH is $PATH and the current directory is `pwd`"`

Where to use Shell variables

- a) `$progs=/home`
`$cd $progs ; pwd`
- b) `$mydir=`pwd` ; echo $mydir`
- c) `$size=`wc -c < foo.txt``
`$echo $size`
- d) `$base=foo ; ext=.c`
`$file=$base$ext`
`$cc -o $base $file`

2. Write and Execute the following using shell scripts.

Create a file script.sh and execute

```
#!/bin/bash
#script.sh : Simple shell script
echo "Today's date: `date`"
echo "This month's calendar:"
cal `date "+%m 20%y"`
echo "My shell: $SHELL"
```

Create the following interactive shell script.

```
#!/bin/sh
# file.sh: Interactive version - uses reads to take two inputs
#
echo "Enter the pattern to be searched: \c"
read pname
echo "Enter the file to be used: \c"
read fname
echo "Searching for $pname from file $fname"
grep "$pname" $fname
echo "Selected records shown above"
```

Command Line Arguments:

```
#!/bin/bash
#file1.sh: Uses command line arguments
#
echo "Program: $0 The number of arguments specified is $# The arguments are
$*"
grep "$1" $2
echo "\n Job Over"
```

exit and EXIT Status of Command :

exit 0	Used when everything went fine
exit 1	Used when something went wrong

`$grep director emp.lst > /dev/null; echo $? $? stores exit status of last command`

`$grep manager emp.lst > /dev/null; echo $? Failure in finding pattern`

`$grep manager emp3.lst > /dev/null; echo $? Failure in opening file`

Lab Exercises:

1. Write a shell code to accept a string from the terminal, and echo a suitable message if it doesn't have at least 10 characters using **case** and **expr** command.
2. Devise a script that accepts two directory names d1 and d2, and deletes those files in d2 which are identical to their names in d1.
3. Write a script that accepts filenames as arguments. For every filename, it should first check whether it exists in the current directory and then converts its name to uppercase, but only if a file with the new name doesn't exist.
4. Write a script that displays a special formatted listing showing the permissions, size, filename, last modification time, last access time of filenames supplied as arguments.
5. Write a shell script that displays the factorial of a given number.

Additional Exercises:

1. Write a shell script that takes two sorted numeric files as input and produces a single sorted numeric file without any duplicate contents.
2. Write a shell script that displays all the possible quadratic equation roots using case.