

Unix Shell Programming Lab
*CSE 2202 Operating Systems Concepts and
System Programming*

dipok.iit.nstu@gmail.com

September 25, 2019

Time	Maximum 3 hrs
Mark Points	100

1 Custom command in Unix

Suppose, we want to create a command ‘*me*’ and it prints the details about a person. Outputs’ template of command ‘*me*’:

Listing 1: ”Me”

```
<First Name><Middle Name><Last Name>  
<Email>  
<Date>
```

Steps to create a custom command, ‘*me*’

- Switch to ‘root’
- Go to ‘/bin’
- Create a file named ‘*me*’
- Edit ‘*me*’ using ‘gedit’ command and add your shell scripts
- Make ‘*me*’ executable
- Type ‘*me*’ and it will show the details about you

Type command ‘*me*’ every 10 minutes.

2 Command for Saving Terminal Logs

The following Unix Shell script named *lab* takes Unix command as arguments. It writes command and its output in a file named *logfile*.

```
#!/bin/sh
logfile='/tmp/182500.txt'
echo 'pwd'\$*$ | tee -a $logfile
$* | tee -a $logfile
```

logfile will be named after you i.e. '/tmp/182501.txt'. Now save command '*lab*'.

'*lab*' works like below:

```
$lab echo Hello World
```

'*lab*' is our custom command to save the command in our log file. '*echo Hello World*' is UNIX command and it prints 'Hello World'. From now, every command in terminal typed as '*lab* {command}'. Suppose we want to print 'Hello Word'.

```
$echo Hello Word # Unix Command
```

But we will type:

```
$lab echo Hello Word #Executing plus Saving Unix Command
```

Type command '*me*' with '*lab*' every 10 minutes.

```
$lab me
```

3 Creating our Custom Commands

In this section, we will create commands replicating popular UNIX commands. After creating those custom commands, we will use our own commands instead of UNIX commands when needed.

4 Working with '*lab*' and Replicated Commands

Suppose now we need to print 'Hello World'. Instead of using '*echo*', we use '*p*'. The commands will be like:

```
$lab p Hello World
```

4.1 Number of user

We want to know the number of users logged on now. Write a shell program named '*nu*' to print the number of users.

UNIX Command	Custom Command
echo	p
cp	c
mv	x
man	m
rm	r
find	f
grep	g
cat	o
ls	l
test	t
wc	w
date	d

Table 1: Commands Mapping

4.2 Phone Book

Create a command ‘pb’ to look up a person, add a person and remove a person from a ‘phonebook’ file. Suppose the ‘phonebook’ file is:

```
Alice 1773425
Bob 94534
John 75493
Doe 5356
```

Command schema of ‘pb’ is like

```
pb <method> <person> [phone] <phonebook>
```

< *method* > is one of three (‘lu’, ‘add’ and ‘rem’) commands whose represent look up a person, add a person and remove a person respectively. < *person* > is name of a person. [*phone*] is mandatory for ‘add’ and ‘remove’.

Now write program ‘pb’ to provide above mentioned facilities.

Hint: To search for a person it will be like below:

```
$lab pb lu 'Alice' phonebook
Alice 1773425
```

‘lab’ command for saving the terminal’s command and output.

4.3 Greetings

Write a program ‘greetings’ to print greeting according to day hour. If day hour is between 0 and 11 then “Good Morning”, between 11 and 17 then “Good Afternoon”, between 18 and 20 then “Good Evening” and between 21 and 23 then “Good Night”.

```
$lab greetings
Good Afternoon
```