Thadomal Shahani Engineering College

Bandra (W.), Mumbai - 400 050.

CERTIFICATE

Certify that Mr./~~Miss~~ KARAN MANOJ CHOPRA

of IT Department, Semester 4 with

Roll No. 36 has completed a course of the necessary

experiments in the subject O.S. LAB under my

supervision in the **Thadomal Shahani Engineering College**

Laboratory in the year 2021 - 2022

Teacher In- Charge Head of the Department

Date _____ Principal

THADOMAL SHAHANI
TSEC
ENGINEERING COLLEGE

CONTENTS

SR. NO.	EXPERIMENTS	~~PAGE NO.~~ LO.	DATE	TEACHERS SIGN.
~~*~~	~~Basic concepts of operating System~~			
1.	Study of Unix general purpose utility commands	LO1	30/01/22	
2.	Study of Vi Editor	~~LO6~~ LO6	30/01/22	
3.	Study of unix file and directory permissions	LO2	06/02/22	
4.	Study of User management commands/ Study of Unix file and directory permissions	LO5	22/02/22	29/02
5.	Programming using shell script (Basic)	LO4	29/02/22	
6.	Programming using Shell Script (Adv)	LO4	29/02/22	
7.	Study of sed command.	LO2	15/03/22	
8.	Study of grep command.	LO3	15/03/22	
9.	Study & implement perl scripting	LO1	22/03/22	
10.	Study & implement hashing in perl Scripting	LO1	29/03/22	
	Written Assignment - 1	LO1		
	Written Assignment - 2	LO1		

Topic :

Page No. :
Date. : / /

UNIX LAB

Karan Chopra
512-36

LAB ASSIGNMENT 1

AIM: To study basic UNIX commands.

THEORY: Some basic commands are-

(a) cd :

→ The cd command is used to change the current directory
Syntax ⇒ cd <directory name>

(b) pwd :

→ The pwd command is used to display the location of current working directory
Syntax ⇒ pwd

(c) ls :

→ The ls command is used to display a list of content of a directory
Syntax ⇒ ls

(d) cat:

→ The cat command can be used to create a file, display content of the file, copy the content of one file to another file, and more.
Syntax ⇒ cat [OPTION]... [FILE]...

(e) vi:

→ The vi command helps in using the vi editor.
Syntax ⇒ vi <fileName>

Topic :

Page No. :
Date. : / /

(f) Uptime:

→ It is used to find out how long the system is active (running)

Syntax ⇒ uptime [-options]

(g) cal:

→ It shows a quick view of the calender in the Linux terminal.

Syntax ⇒ cal

(h) clear:

→ It clears the contents of the terminal

Syntax ⇒ clear

(i) whoami:

→ It shows the name of the currently logged-in user

Syntax ⇒ whoami

TERMINAL OUTPUT: (In the document attached)

CONCLUSION: We have successfully used and implemented basic commands in UNIX.

Topic : Page No. : Date. : / /

Topic : Page No. : Date : / /

LAB ASSIGNMENT 2

Karan Chopra
SI2-36

AIM : To study commands of Vi editor

THEORY:

- The vi editor is elaborated as visual editor
- It is advisable to work on vi editor as it is
  - feature rich
  - Provides endless possibilities to work on files
- There are two modes
  - command mode
  - Insert mode.

Commands:

(a) vi :
→ Used to open a new file.
Syntax : vi <filename>

(b) I :
→ insert text at the beginning of line.

(c) i :
→ Start typing before the current character.

(d) O :
→ Start typing on a new line after the current line

Topic :

Page No. :
Date. : / /

(e) O
→ Start typing on a new line before the current line

(f) yy
→ copies the entire line
- 4yy will copy 4 lines and so on.

(g) dd
→ cut/delete the entire line
- 3dd will cut/delete 3 lines and so on

(h) p
→ paste the cut or copied part

(i) r
→ replace one character at a time

(j) R
→ keep replacing characters until <esc> is pressed

(k) w
→ save file but not quit vi

(l) ZZ
→ quits VI and saves edits.

TERMINAL OUTPUT: (In the document attached)

CONCLUSION: We have successfully used and implemented basic commands in vi editor

Topic :

Page No. :
Date. : / /

LAB ASSIGNMENT 8

Karan Chopra
SI2-36

AIM: To study Unix file system, file directory, permissions, etc.

THEORY:

(1) File Permission : These are divided into three types

(a) Owner permission: Determines what actions owner of the file can perform on the file.

(b) Group permission: Determines what actions a user who is member of a group that file belongs to, can perform on the file.

(c) Other permission - Indicates what actions all other users can perform on the file.

(2) Access Modes

(a) Read 'r' : Access to view file / directory's contents.
(b) Write 'w' : Access to change file / directory's content.
(c) Execute 'x' : Access to execute the file (binary or shell script)
Access to enter the directory.

r - - Only read permission for all users
rw - Read, write permission for members of group
r w x Read, write, execute for owner of group

(3) Various commands

chmod : Change file modes
chown : Change file owner
chgrp : Change file group owner

Syntax : chgrp [groupname] [filename]
chown [username] : [groupname] [filename]

(4) Head command

Prints the top N number of data of the given input.
Syntax : $ head state.txt

Tail command

Print the last N number of data of the given input
Syntax : $ tail state.txt

(5) Octal chmod
Enable to change the permission on a file

Page No. :
Date. : / /

Topic :

Octal value	File Permission Set	Permission Description
0	---	No permission
1	--x	Execute permission only
2	-w-	Write permission only
3	-wx	Write and Execute
4	r--	Read only permission
5	r-x	Read and execute
6	rw-	Read, write permission
7	rwx	Read, write and execute permission.

CONCLUSION: Successfully executed and Implement file permission commands.

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

LAB ASSIGNMENT 4

Karan Chopra
SI2-36

AIM: To Study and implement execution of User Management commands

THEORY:

The various user management commands are-

(a) SU and sudo-
Both su and sudo elevate privileges assigned to the current user
The main difference is that su requires the password of the target account while sudo requires the password of the current user

(b) login - it is used to establish a new session with the system. When called from a shell, login should be executed as exec login which will cause the user to exit from the current shell.

(c) exit - command in linux is used to exit the shell where it is currently running. It takes one or more parameters as [N] and exists the shell with a return of status N.

(d) useradd - is a low level utility for adding users.
It creates a new user account using the details specified on the command line plus some default values from the system.
By default, a group will also be created for the new user.

(e) Adduser - it is a more advanced version of useradd, which asks for more details while adding a user.

(f) groupadd - it creates a new group account using the values specified on the command line plus some default values from the system.
- The new group will be entered into the system files as needed.

(g) usermod - The usermod command modifies the system account files to reflect the changes that are specified on the command line

(h) chown - This command changes the user and/or ownership of each given file.
- If only an owner is given, that user is made owner of each given file, and the files' group is not changed.

(i) chage - This command changes the number of days between password changes and the date of last password change.
- This information is used by the system to determine when a user must change their password.

(j) chfn - This command changes the user fullname, office room number, office phone number and home phone number information for a user's account.
- it updates the finger information field in your /etc/passwd entry.

(k) Userdel - it is a low level utility for removing users.
- it modifies the system account files, deleting all entries that refer to the username LOGIN.

(l) gpasswd - It is used to administer /etc/group and /etc/gtshow
- Every group can have administrators, members and a password.

(m) whoami - It prints the user name of the currently logged in user.

CONCLUSION : We have successfully studied and executed user management commands.

LAB ASSIGNMENT 5

AIM: Basics of shell scripting

THEORY:

Shell- It is special use program which provides an interface to user to use operating system services.

Served shells : BASH (Bourne shell)
CSH (C shell)
KSH (Korn shell)

Bash shell

Begins will #!/bin/bash, otherwise the user's current shell will be used. Extension is .sh

expr

The expr command in Unix evaluates a given expression and display its coorresponding output

x = 4

y = 5

expr = $x + $y

For multiplication, expr $x* $y

→ comments

Cat /etc/shells → To check which shells your OS supports.

Execution of Bash file by two ways:

./abc.sh or bash abc.sh

CODE :

```
#!/bin/bash
echo "Hello"
echo "Enter two numbers"
read x
read y
sum = expr $x + $y
sub = expr $x - $y
mul = expr $x \* $y
div =expr $x / $y
echo "$sum"
echo "$sub"
echo "$mul"
echo "$div"
```

Output :

```
Hello
Enter two numbers
5
2
7
3
10
2
```

Arguements:

$0 → name of script
$1 → first argument
$4 → count number of arguments
$7 → command line

Reading input
read variable_name
read l
ech o "$1"

CONCLUSION: Bash shell is used to automate frequently performed operations, easy to use and portable
: we have successfully studied basics of shell scripting.

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

LAB ASSIGNMENT 6

Karan Chopra

AIM: To study and implement advanced concepts of shell scripting.
: To study about various loops.

THEORY:

If statement in shell programming -

Syntax -

```
if [logical expression]
then
:
:
else
fi
```

for loop -

Syntax -

```
for variable in 'list of variables'
do
    command 1
    command 2
done
```

Eg:

```
for x in 1 2 3 4 5
do
    echo "The value of x is $x"
done
```

Output -

```
1
2
3
4
5
```

While loop
Syntax -

```
while [logical expression]
do
:
:
done
```

Eg -

```
#!/bin/bash
num = 0
while [$num -lt 10]
do
    echo $num
    num = $[$num+1]
done
```

Arithmetic operations using double parenthesis (()):
It is used to perform integer arithmetic operations

Until loop - The until loop is used for repeating the set of instructions for the time the specified logical expression is false. The moment the logical expression becomes true, the control will come out of the loop.

Syntax :

```
until logical_expression do
:
:
done
```

Eg. Program to print sum of ~~even~~ even numbers upto 50

```
#!/bin/bash
s=0
n=2
until [$n -gt 50]
do
s=$((($n+$s))
((n+=2))
echo $n
done
echo $s
```

output

```
4
6
8
10
:
:
52
650
```

switch case:

file name: switch.sh

```
case "$1" in
1) echo "I am in case 1"
   ;;
2) echo "I am in case 2"
   ;;
3) echo "I am in case 3"
   ;;
4) echo "I am in case n"
   ;;
esac
```

→ Arrays

Eg: #!/bin/bash
array (dog, cat bat mouse);
echo " ${array[1]}"

→ Reading elements of an array

Program:

```
echo -n "Enter the total numbers:"
read n
echo "Enter the numbers"
i=0
while [ $i -lt $n ]
do
    read a[$i]
    i='expr $i+1'
done
echo ${a[@]}
```

Output

```
Enter the total numbers:
Enter the numbers:
9
8
1
2
3
9
8
1
2
3
```

CONCLUSION:

Thus, we have studied some of the advanced concepts of shell script like loops and Arrays. Some programs depicting these concepts are also executed.

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

LAB ASSIGNMENT 7 & 8

Karan Chopra

21/03/22

AIM: To understand and execute commands related to sed and grep in Linux

THEORY: SED command in UNIX stands for stream editor and it can perform many functions on a file like searching, find and replace, insertion or deletion

Eg-

We take a bank.txt file and execute various 'sed' commands are the same

1) Print first 3 lines:

```
sed '3q' bank.txt
```

O/P

```
101   ADITYA   0       14/11/2000   CURRENT
102   Anil     10000   20/05/2011   Saving
103   Naman    0       20/08/2009   current
```

2) Repeat line no. 5

```
sed '5p' bank.txt
```

3) Repeat two lines 3rd and 5th

```
sed '3,5p' bank.txt
```

4) Print nth line from file:

```
sed -n '5p' bank.txt
```

5) Print 2nd to 5th line

```
sed -n '2,5p' bank.txt
```

6) Print the last record on file

```
sed -n '$p' bank.txt
```

7) From 8th to last line

```
sed -n '8,$p' bank.txt
```

8) Lines except m to n:

```
sed -n '5,$!p' bank.txt
```

9) Print all lines except a few:

```
sed -n '2,9!p' bank.txt
```

10) print all records

```
sed '$q' bank.txt
```

11) Print all records except last line

```
sed -n '$!p' bank.txt
```

12) combine multiple commands (-e)

```
sed -n -e '1,2p' -e '7,9p' -e '$p' bank.txt
```

→ sed script:

```
cat abc.txt
1,2p
7,9p
$p
```

Terminal

```
sed -n -f abc.txt bank.txt
```

13) Find occurance of a word

```
sed -n '/current/p' bank.txt
```

14) Replace :

```
sed 's/current/curr/g' bank.txt
```

↑ Substitute ↑ global

15) Writing to a file

```
sed -n '/[cC]urrent/w clist.txt' bank.txt
```

16) Grep : Grep is used to search for a string of characters in a specified file

Options:

1. -c : This prints only a count of the lines that match a pattern.
2. -h : Display the matched lines, but do not display the file name
3. -i : Ignores, case for matching.

CONCLUSION: Thus, we have executed various 'sed' and 'grep' commands successfully.

LAB ASSIGNMENT 9

Karan Chopra
5/12/36

AIM: To study and implement Perl scripting

THEORY:

1. PERL is Practical Extraction Reporting Language
2. It is used for text processing and automation
3. PERL is an interpreted language.

→ Variables in Perl:

- Scalar (starts with $)
- Array (starts with @)
- Hashes (starts with %)

Scalar → Integers, Float
Scalar → String

→ Arrays in PERL: Array variables are produced by "@"

Eg: @ages = (25, 20, 21);

→ Hash Data: Set of key/value pair is called a hash. Each key in a hash structure are unique and type of strings. The values associated with these keys are scalar. Each key is associated with a single value.

Eg: %data = ('Danish', 28, 'Raju', 40, 'Ritesh', 25);

Key Value

→ PERL

1. Perl file extension is .pl or .pm
2. Execute :

perl scriptname.pl

Program - WAP in perl Script to show the operations of variables and arrays.

code -

```
#!/bin/perl
$emp-name = "Varun";
$emp-age = 20;
$emp-salary = 1.1,
$emp;
print "Age = $emp-age\n";
print "Name = $emp-name\n";
print "salary = $emp-salary\n";
print " x = $emp\n")
$emp-age = 21;
print "new age = $emp-age\n";
@games = ("Cricket", "Football", "Hockey")
@num = (1,2,3,4,5)
print "\$num[0] = $num[0]\n";
print "\$num[1] = $num[1]\n"
print "\$games[0] = $games[0]\n";
print "\$games[1] = $games[1]\n";
```

Output -

```
Age = 20
Name = Varun
Salary = 1.1
X =
```

new age =21
$num[0] = 1
$num[1] = 2
$games[0] = Cricket
$games[1] = Football

Program - WAP in Perl Script to take input of 5 students' names and their CGPA. Print the sum and average of their CGPAs.

Code -
```
#!/bin/perl
print "Enter the names of 5 students \n";
for ($i=0; $i<5; $i++)
{
    $student[$i] = <STDIN>;
    chomp @student;
}
print "Enter the CGPAs of 5 students\n";
for ($i=0; $i<5; $i++)
{
  $cgpa[$i] = <STDIN>
  chomp @cgpa;
}
$sum =0
for ($i=0; $i<5; $i++)
{
    $sum = $sum + $cgpa[$i];
}
print "-$sum\n"
print "$sum/5";
```

```
print "$avg";
```

Output:

```
enter the names of 5 students
Varun
Ram
Joe
John
Rose
enter the CGPAS of 5 students
9
8
9
8
9
43
8.6
```

CONCLUSION: Thus, we have studied in brief, what is PERL script; implemented some programs to understand the concepts and executed them successfully.

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

LAB ASSIGNMENT 10

AIM: Study and implement Hashing in Perl scripting

THEORY:

Hash data: set of key/value pairs is called a hash. Each key in a hash structure are unique and type of strings. The values associated with these keys are scalar. Each key is associated with a single value.

Eg: %data = ('Danish', 28, 'Raju', 40, 'Ritesh', 25);

key value.

PROGRAM:

Write a program to find the maximum of scores among 3 departments.

```
#!/usr/bin/perl
%data = ('IT', 10, 'computer', 9, 'EXTC', 9.5);

if ($data{'IT'} > $data{'computer'})
{
        if ($data{'EXTC'} > $data{'IT'})
        {
            print "The team with maximum points is
                    EXTC with points $data{'EXTC'}\n";
        }
        if ($data{'EXTC'} < $data{'IT'})
        {
```

```
                print "The team with maximum points is
                                IT with points $data
            }                                   {'IT'} \n";
    }
if ($data{'IT'}  < $data {'computer'})
{
    if ($data {'EXTC'} > $data {'computer'})
    {
        print "The team with max. points is EXTC
                with points $data{'EXTC'} \n";
    }
    if ($data {'EXTC'} < $data {'computer'})
    {
        print "The team with max. points is computer with
                points $data {'computer'} \n";
    }
}
```

Output -

The team with maximum points is IT with points 10

CONCLUSION - We have successfully executed hash commands in Perl scripting.