College of Engineering Chengannur

cs232

FREE AND OPEN SOURCE SOFTWARE LAB REPORT

Goutham G

S4D

22

Certificate

Name : Class :

Roll No : Exam No :

This is certified to be the bonafide record of practical work done in Free and Open Source Software as per Syllabus of class .... in the Lab during the academic year 20 /20

....

Teacher In-charge *Head of Dept.*

....

Examiner's Signature *Principal*

Date : .... *Institution Rubber Stamp*

Index

1. Linux Commands

• ls

ls with no option list files and directories in bare format where we won’t be able to view details like file types, size, modified date and time, permission and links etc.

ls	List Files in current directory
ls-l	Long list files in current directory
ls-ltr	List latest modification file or directory date as last

• alias

alias command instructs the shell to replace one string with another string while executing the commands.
The **unalias** utility is used to remove an existing alias

alias CD=”ls-ltr”	Defined ‘CD’ as ‘ls-ltr’
unalias CD	Remove the definition of CD

• cat

cat (concatenate) command is very frequently used in linux.It reads data from file and give their content as output.It helps us to create,view,concatenate files.

cat filename	Show the content of filename
cat > filename	Create a file named filename

• sort

SORT command is used to sort a file, arranging the records in a particular order. By default, the sort command sorts file assuming the contents are ASCII. Using options in sort command, it can also be used to sort numerically

Sort <filename>	Print the sorted contents of the filename

• cut

The **cut** command in UNIX is a command for cutting out the sections from each line of files and writing the result to standard output

Cut –b 1,2 filename	print first and second byte from filename

- **paste**

It is used to join files horizontally (parallel merging) by outputting lines consisting of lines from each file specified, separated by **tab** as delimiter, to the standard output.

```
paste <filename1> <filename2>
```

- **join**

The j**oin** command lets you combine lines of two files on a common field. join field is the first, delimited by blanks. When FILE1 or FILE2 (not both) is -, read standard input

```
Join <filename1> <filename2>
```

- **mkdir**

It is used to create a new directory

```
mkdir <filename>        create a directory with filename
```

- **cd**

cd is used to change directory

```
cd  <directory_name>
```

- **man**

To show manual of inputed commands.

- **grep**

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

```
grep [OPTION] PATTERN [Files]
```

- **touch**

It is used to create a file without any content.

```
touch  <filename>
```

2. Scripting Task

This task is done to calculate CGPA of all students in S4D class from s1 and s2 results. In this experiment ,we first downloaded the s1 result 2018 from Ktu sites and changed the pdf file to text file by using the command :

```
pdftotext -nopgbrk filename.pdf
```

Select the Computer science students by using :

```
grep --no-group-separator -A3 'CHN17CS' filename.txt | tr '\n' ' ' | sed
'S/\ CHN/\nCHN/g' > newfile.txt
```

Change the grade value to points by using :

```
sed -i 's/(grade)/ poit /g' newfilename.txt
```

Calculate the SGPA of all students by using awk command :

```
awk '{s=$3*4+$6*4+$9*4+$12*3+$15*3+$18*3+$21*1+$24*1+
$27*1}{r=s/24}

{print r}' filename.txt >new_filename.txt
```

Paste the Name , Register No: , SGPA in a file

Complete the s2 result 2018 SGPA calculation as per the above commands Paste the Register No: , SGPA of s2 into a file

Add two SGPA results and get CGPA of each students by using awk command :

```
awk '{s=$2+$3}{r=s/2}{print r}' s1_s2.txt >cgpa.txt
```

An “echo” command is used to display line of text/string that are passed as an argument. Echo is used here to display the output as Name , Reg.No: and CGPA of each student.

Then displays our CGPA calculation by using : `cat filename.txt`

From this experiment we can familiarize more commands and its uses in Linux and to know that how to calculate CGPA of each student by using Linuxcommands.

3. Networking Task

This task is done to familiarize with the commands that can be used to set up a computer network and share data. Two or more computer connected through network media called computer network. There are number of network devices or media are involved to form computer network.

- ## ifconfig

  ifconfig(interface configuration) command is used to configure the kernel-resident network interfaces. It’s used at the boot time to set up the interfaces as necessary. After that, it is usually used when needed during debugging or when you need system tuning. Also, this command is used to assign the IP address and netmask to an interface or to enable or disable a given interface

  ifconfig [..OPTIONS] [INTERFACE]

  Ifconfig [interface] display details about the given [interface]

  ifconfig [interface] [ip] assigns an ip for the [interface]

  ifconfig [interface] up/down enables or disables [interface]

- ## Ping

  PING (Packet INternet Groper) command is the best way to test connectivity between two nodes

- ## route.

  Route manipulates the kernel's IP routing tables. Its primary use is to set up static routes to specific hosts or networks via an interface after it has been configured with the ifconfig program

  route displays route table

  route add default gw <ip> adds a default gateway

  route delete default gw <ip> removes default gateway from route table

4. SSH ,SCP,RSYNC

- ## SSH

ssh (SSH client) is a program for logging into a remote machine and for executing commands on a remote machine. It is intended to provide secure encrypted communications between two untrusted hosts over an insecure network. X11 connection, arbitrary TCP ports and UNIX-domain sockets can also be forwarded over the secure channel. ssh connects and logs into the specified destination, which may be specified as either [user@]hostname or a URI of the form ssh://[user@]hostname[:port] The user must prove his/her identity to the remote machine using one of the sevaral methods.

Connect to a remote server**:**

ssh user@remoteip

Commands like ls, cat, mkdir, cd, touch, rm, cp, mv, grep, du, nano work in the SSH shell

- ## SCP

The Secure Copy command is used to transfer files between two hosts. It uses the same authentication and security as the Secure Shell (SSH) protocol. It is a command line utility. Since it uses SSH the transferred data will be encrypted and transferred through a secure channel.

-r copy entire directories recursively

Sending a file (test.txt) to the server:

$ scp test.txt @:/remote/directory

Receiving a file (test.txt) from the server:

$ scp @:/remote/directory/test.txt /home/goutham/Desktop

Sending a directory (sample) to the server:

$ scp -r sample @:/remote/directory

- ## RSYNC

rsync is a fast, versatile file-copying tool. It can copy locally, to/from another host over any remote shell, or to/from a remote rsync daemon. It offers a large number of options that control every aspect of its behaviour and permit very flexible specification of the set of files to be copied. It is famous for its delta transfer algorithm, which reduces the amount of data sent over the network by sending only differences between the source files and the existing files in the destination. Rsync is widely use dfor backups and mirroring and as an improved copy command for everyday use.

-a archive combining into a single file
-v verbose gives a brief summary about the transferring data
-z compress data files during transfer
-h human-readable format
-a preserves the date and time and permission of the files

Basic syntax

rsync [options] <source> <destination>

Copy/Sync a file on a local computer

rsync -azvh <sourcefile> <destination>

Copy/Sync a directory on a local computer

rsync -azvh <sourcedirectory> <destination>

Copy/Sync a directory to a remote computer

rsync -azvh <sourcedirectory> user@serverip:<destination>

5. FTP Usage and Commands

FTP (File Transfer Protocol) is used to transfer files between two remote systems. It is network protocol similar to HTTP, but for file transfer. ftp clients are used to connect to ftp servers. After connection tools provided by FTP can be used for file transfer.

Usage

Connect to server

ftp <server>

Download file from server

get <file>

Download multiple files from server

mget <directory>

Upload file to server

put <file>

Upload multiple files to server

mput <server>

Change remote working directory

cd <directory>

Change local working directory

lcd <directory>

Change file permissions of remote file

chmod [options] [files]

Delete remote file

delete <filename>

Delete multiple remote files

mdelete <files>

Remove directory on remote server

rmdir <directory>

Exit ftp session

exit

6. Linux Installation

Installing a linux distribution on system requires the creation of an ext4 (sometimes ext3) formatted partition. Different directories like /usr /home /boot can be installed on separate partitions formatted as ext4. These should be mounted during init via fstab entries (which will be created by default for most linux distributions).

If the system uses UEFI then a separate partition formatted as FAT32 is required for ESP which contains boot managers and boot information for the UEFI.

The installation of required proprietry drivers may be provided with the installation image. If not it should be manually installed.

After the copying of required OS files localization settings like time, timezone, language, etc.. are configured. The installation interface might provide options for the creation of additional user accounts for the installed operation system. The password for the root user might be configured during installation, or set to a default password.

7. Setting Up Server

• FTP

An FTP server is a computer which has a file transfer protocol address and is dedicated to receiving an FTP connection. It requires TCP/IP network for functioning and is dependent on usage of dedicated servers with one or more FTP clients. Here VSFTPD (Very Secure FTP Daemon) have been used to create FTP server.

For setting up FTP server, vsftpd has been installed using apt (Advanced Package Tool) and opening ports using UFw (Uncomplicated Firewall) for incoming connections through (20/tcp, 21/tcp, 990/tcp, 40000:50000/tcp).

```
$ sudo ufw allow <port>
```

Then a new user account has been created for FTP access. VSFTPD accomplishes this with chroot jails. When chroot is enabled for local users, they are restricted to their home directory by default.

Clients are able to access using

```
$ ftp <username>@<ip>
```

- **HTTP**

HTTP or web server is computer which uses Hyper Text Tranfer Protocol to serve files to the clients in response to their requests. Apache is one of the most popular server software to host an HTTP server. It is designed to create web servers that have the ability to host one or more HTTP-based websites. It includes the ability to support multiple programming languages, server-side scripting, an authentication mechanism and database support

Installation using apt

```
$ apt install apache2
```

Adjusting firewall (ufw)

```
$ sudo ufw allow 'Apache Full'
```

Start Apache server

```
$ sudo systemctl start apache2
```

Stop Apache server

```
$ sudo systemctl stop apache2
```

Restart Apache server

```
$ sudo systemctl restart apache2
```

Reload Apache server

```
$ sudo systemctl reload apache2
```

Enable Apache server

```
$ sudo systemctl enable apache2
```

Disable Apache server

```
$ sudo systemctl disable apache2
```

8. Further Tasks

• Package Management

A package-management system is a collection of software tools that automates the process of installing, upgrading, configuring, and removing computer programs for a computer's operating system in a consistent manner. Operating system like Debian, Ubuntu uses apt, apt-get which comes pre-installed to the OS. Further package management tools like pip, npm, anaconda can be installed through apt. Packages which are not available in there libraries can be manually installed from the source.

Update package database

$ apt update

Upgrade installed packages

$apt upgade

Install new packages

$ sudo apt install <package_name>

Remove installed packages

$ sudo apt remove <package_name>

Search for packages

$ apt search <package_name>

Display content of a package

$ apt show <package_name>

List upgradable packages

$ apt list –upgradeable

List installed packages

$ apt list --installed

- PERL

Perl is a family of two high-level, general-purpose, interpreted, dynamic programming languages. It is a general-purpose programming language originally developed for text manipulation and now used for a wide range of tasks including system administration, web development, network programming, GUI development, and more.

Perl is a stable, cross platform programming language. It takes the best features from other languages, such as C, awk, sed, sh, and BASIC, among others. Its database integration interface DBI supports third-party databases including Oracle, Sybase, Postgres, MySQL and others. Perl works with HTML, XML, and other mark-up languages.

Program to print two numbers:

```
#!/usr/bin/perl
use strict;
use warnings;
print "Enter two numbers \n";
$a = <>; $b = <>;
 my $sum = $a+$b;
print "Sum = $sum\n";
```

- ## LAMP Stack

The LAMP (Linux Apache MySQL PHP) stack is the foundation for Linux hosted websites. It consist of four software layers which supports each other. Websites and web applications run on top of this underlying stack. Integration between these four layers makes it the best platform to host websites.

Linux: The operating system makes up the first layer. It sets the foundation for the stack model. All other layers run on top of this layer.

Apache: The second layer consists of HTTP server software. This layer resides on top of the Linux layer. It waits for the HTTP request from the clients and server the appropriate files (webpages) as per the request. It is also the most popular HTTP server side software.

MySQL: MySQL stores details that can be queried by scripting to construct a website. It usually sits on top of the Linux layer alongside Apache. In high end configurations, MySQL can be off loaded to a separate host server.

PHP: Sitting on top of them all is our fourth and final layer. The scripting layer consists of PHP or other similar web programming languages. Websites and web applications run within this layer

PHP program to add two numbers

```
<?php
      $x  = 3;
      $y  = 5;
      $sum = $x + $y;
      echo ,Sum is , . $sum;
?>
```

- ## Kernal Compilation

A kernel is the lowest level of easily replaceable software that interfaces with the hardware in your computer. It is responsible for interfacing all of your applications that are running in user mode down to the physical hardware, and allowing processes, known as servers, to get information from each other using inter-process communication .
We may need to compile our own kernel to add/remove some features present in the system. The kernel distributed with general settings which should run on all the possible installations. Thus they need to support a wide range of hardware. Some of the features may be built in the kernel while some of them may be built as modules.

The various commands that are used in kernel compilation are :

- ***curl -fLO "https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.0.9.tar.xz"***

- ***sudo apt-get install git fakeroot build-essential ncurses-dev xz-utils libssl-dev bc flex libelf-dev bison***

- ***tar xvzf linux-5.0.9.tar.xz***

- ***cd linux-5.0.9***

- ***cp /boot/config-4.19.0-kali4-amd64 .config***

- ***make menuconfig***

Once the .config file has been set for the custom kernel, within the source directory run the following command to compile:

- make
- make modules_install
- sudo make install

These are the commands that are used for kernel compilation. Since Compilation time varies from as little as fifteen minutes to over an hour, depending on your kernel configuration and processor capability ,sometimes it’s a time consuming task.

9. Own Webpage

An nginx server (14.139.189.217) is configured in such a way that it directs the URL http://14.139.189.217/cs17d/cs17dxx to the corresponding home directory of user account (cs17dxx) in search of index.html as the source file.

An index.html file with experiment details and hyperlinks to the github repository is created locally. Then transferred to the home directory of student in the server using scp.

```
$ scp index.html <user>@<server>:/home/directory
```

10. Menu Driven Calculator

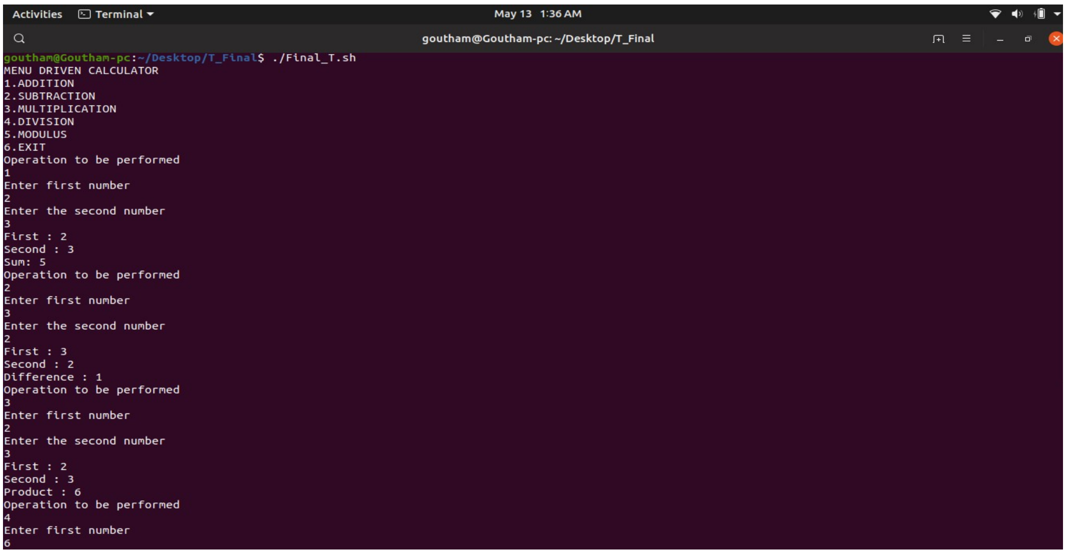
(Detailed Report)

Shell Script:

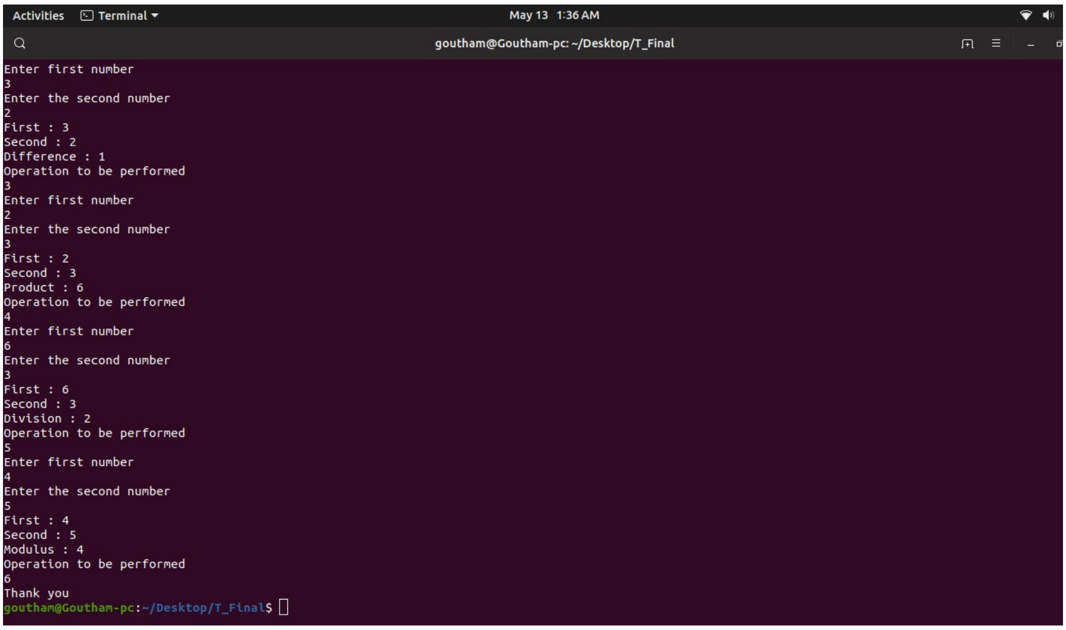
```
#!/bin/shh



            echo "MENU DRIVEN CALCULATOR"
            echo "1.ADDITION"
            echo "2.SUBTRACTION"
            echo "3.MULTIPLICATION"
            echo "4.DIVISION"
            echo "5.MODULUS"
            echo "6.EXIT"
            op=0
            until [ $op -eq "6" ]
            do
                    echo "Operation to be performed"
                    read op
                    if [ $op -eq "6" ]
                    then
                            echo "Thank you "
                            break
                    fi
                    echo "Enter first number"
                    read first
                    echo "Enter the second number"
                    read second

                    echo "First : $first"
                    echo "Second : $second"


                    if [ $op -eq "1" ]
                    then
                            sum=`expr $first + $second`
                            echo "Sum: $sum"



                    elif [ $op -eq "2" ]
                    then
                            diff=`expr $first - $second`
                            echo "Difference : $diff"



                    elif [ $op -eq "3" ]
                    then
                            mul=`expr $first \* $second`
                            echo "Product : $mul"
```

```
        elif [ $op -eq "4" ]
        then
                div=`expr $first /  $second`
                echo "Division : $div"


        elif [ $op -eq "5" ]
        then
                mod=`expr $first % $second`
                echo "Modulus : $mod"



        Fi
Done
```

OUTPUT



```
goutham@Goutham-pc:~/Desktop/T_Final$ ./Final_T.sh
MENU DRIVEN CALCULATOR
1.ADDITION
2.SUBTRACTION
3.MULTIPLICATION
4.DIVISION
5.MODULUS
6.EXIT
Operation to be performed
1
Enter first number
2
Enter the second number
3
First : 2
Second : 3
Sum: 5
Operation to be performed
2
Enter first number
3
Enter the second number
2
First : 3
Second : 2
Difference : 1
Operation to be performed
3
Enter first number
2
Enter the second number
3
First : 2
Second : 3
Product : 6
Operation to be performed
4
Enter first number
6
```



```
Enter first number
3
Enter the second number
2
First : 3
Second : 2
Difference : 1
Operation to be performed
3
Enter first number
2
Enter the second number
3
First : 2
Second : 3
Product : 6
Operation to be performed
4
Enter first number
6
Enter the second number
3
First : 6
Second : 3
Division : 2
Operation to be performed
5
Enter first number
4
Enter the second number
5
First : 4
Second : 5
Modulus : 4
Operation to be performed
6
Thank you
goutham@Goutham-pc:~/Desktop/T_Final$ 
```

Command Description

- #!/bin/sh

Before you add anything else to your script, you need to alert the system that a shell script is being started. This is done using the **shebang**construct

- echo

It is used to print .eg: echo “Hello World”

- read

It is an input statement . eg: read variable_name

- until

to execute a set of commands until a condition is true

syntax:

```
until command
do
  Statement(s) to be executed until command is true
done
```

- if….elif….fi

The **if...elif...fi** statement is the one level advance form of control statement that allows Shell to make correct decision out of several conditions.

Syntax :

```
if [ expression 1 ]
then
        Statement(s) to be executed if expression 1 is true
elif [ expression 2 ]
then
        Statement(s) to be executed if expression 2 is true
elif [ expression 3 ]
then
        Statement(s) to be executed if expression 3 is true
else
        Statement(s) to be executed if no expression is true
fi
```

- Declaring Variables:

variable_name=value

- Accessing Variables

echo $variable_name