COLLEGE OF ENGINEERING TRIVANDRUM

LAB RECORD
Free And Open Source Software Lab

Submitted By
Belbin G Kunjumon
Roll No : 05
S2 CSE

CONTENTS

1.GETTING STARTED WITH LINUX BASIC COMMANDS FOR DIRECTORY OPERATIONS

1.1 AIM

Getting started with Linux basic commands for directory operations, displaying direc-tory structure in tree format etc.

1.2 OVERVIEW

A directory in Linux is similar to a folder in windows OS. Files are organized in to directories and sub- directories. In Linux, path begins at the root directory which is the top-level of the file system and is represented as a forward slash (/) . Forward slash is used to separate directory and file names.

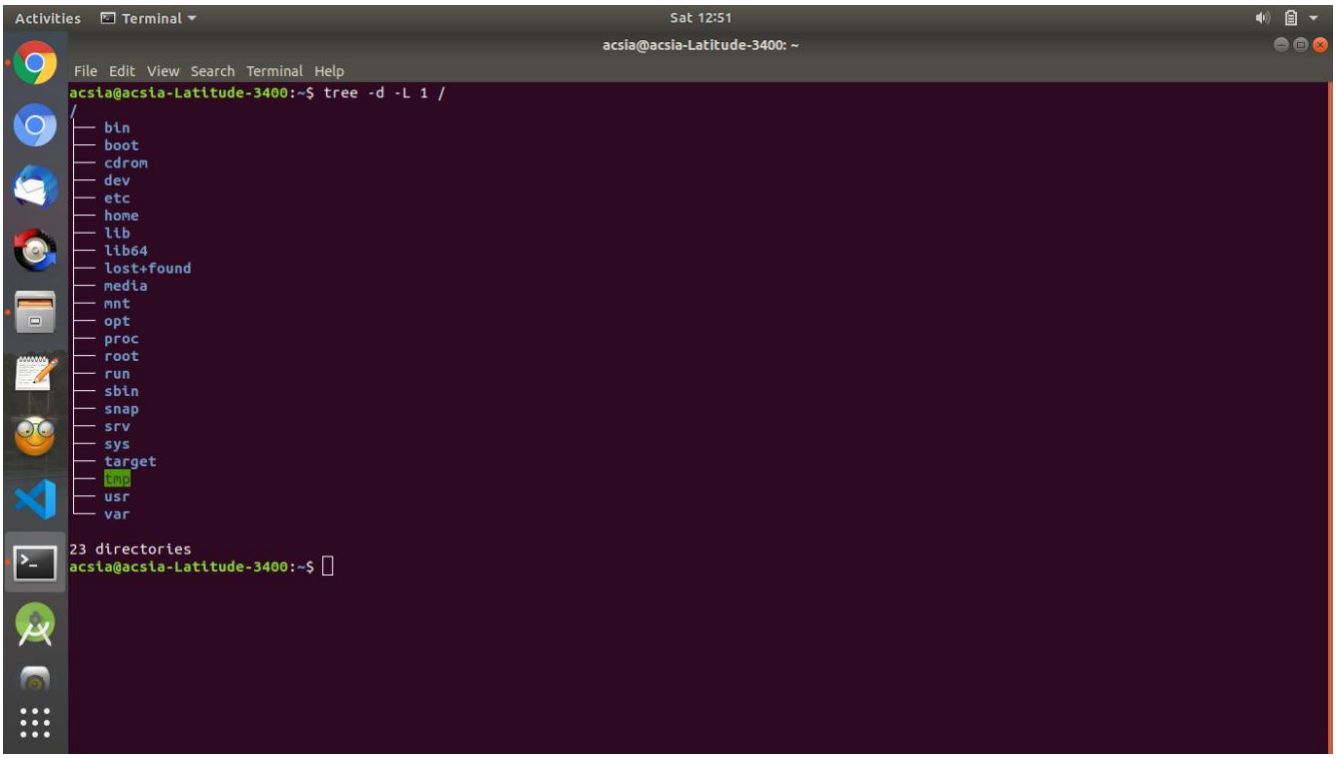
Basic Commands

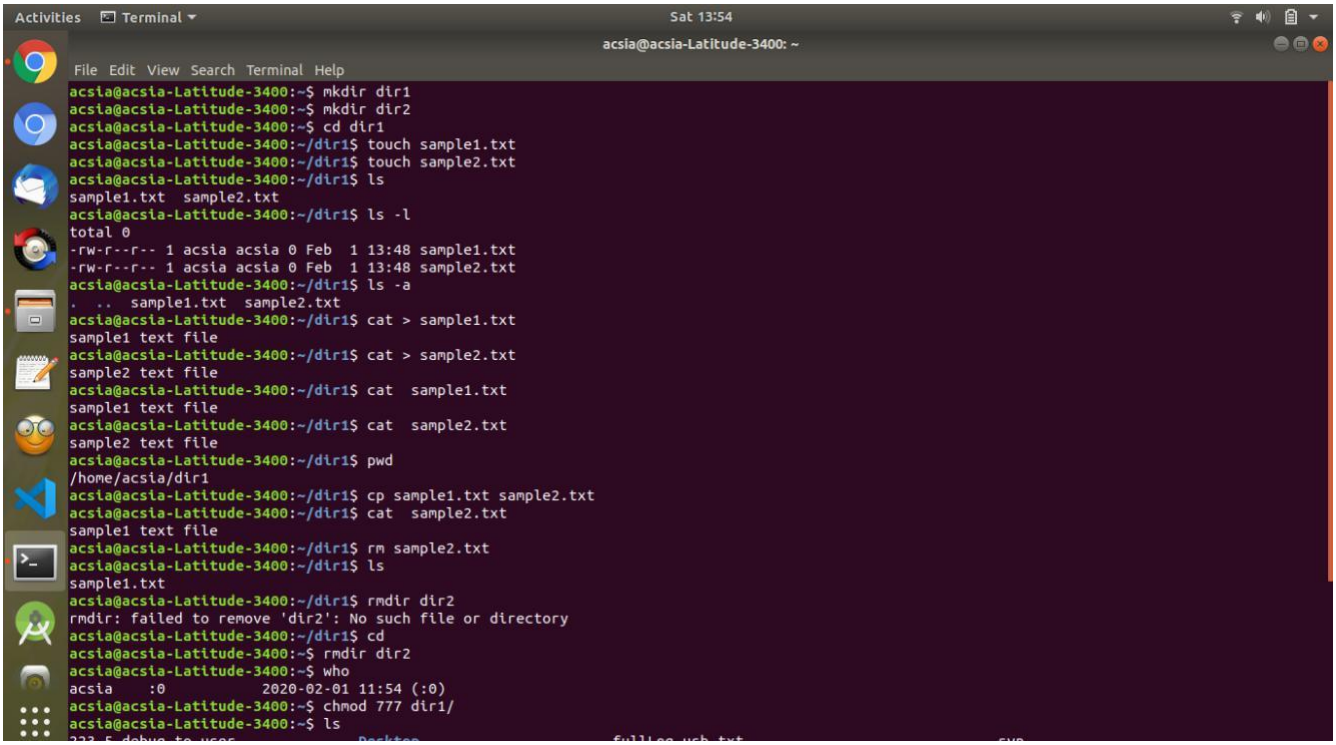
Command	Operation
touch	filename create a new file
mkdir	dirname create a new directory
pwd	prints present working directory
cd	dirname change directory
cat	filename view contents of a file
more	filename view contents of a file one scornful at a time
less	filename similar but faster than more
ls	list files in a directory
ls -l	provide long listing of all the files
ls -l -h	provides sizes in human readable form
ls -F	mark all executables with * and directories with /
ls -a	show all files in the present directory with special dot files
cp file1 file2	copying files
cp -r dirname1 dirname2	copy directories
rm filename	remove a file
rmdir -r dirname	remove a non empty directory
clear	clear the contents of the terminal
locate	search for a specified filename
man	commandname view help of the specified command name
chmod [options] mode filename	change file permissions
chown [options] filename	Change file ownership
kill [options] pid	Kill a process
who [options]	Display who is logged in
top	Display the resources being used in your system
ln [options] source [desitination]	Create a shortcut
tree	Print the directory structure in tree format

Directory structure

File name	Content
/bin	Essential user command binaries
/boot	Static files and boot loader
/dev	Device files
/etc	Host specific system configuration
/home	User home directories
/lib	Essential shared libraries and kernel modules
/mnt	Mount point for devices
/opt	Add on application software packages
/sbin	System binaries
/tmp	Temporary files
/usr	User utilities and applications
/var	variable files
/root	home directory for the root user

1.3 OUTPUT





1.4 RESULT

Several basic linux commands and the direcotry structure of linux were studied. The directory structure was printed in tree format using tree command.

2. LINUX COMMANDS FOR OPERATIONS SUCH AS REDIRECTION, PIPES, FILTERS, JOB CONTROL, CHANGING OWNERSHIP/PERMISSIONS OF FILES/LINKS/DIRECTORY.

2.1 AIM

Linux commands for operations such as redirection, pipes, filters, job control, changing ownership/permissions of files/links/directory.

2.2 OVERVIEW

Redirection of standard Input/Output

• By default most command line programs send their output to the standard output which by default displays it on
commandName > fileName -Overwrites the file with the output of the command
commandName >> fileName - appends the file with the output of the command.
• Most of the command line programs accept its input from the standard input and by default gets its contents from the keyboard. Similar to standard output it can also be redirected.
sort < filename - sort command processes the contents of the file with the name filename.
sort < file1 > file2 processes the contents of file 1 and redirects its output to file 2

Pipes

Pipes are used to redirect the standard output of one command to the standard input of another command.
command1 | command2 the standard output of command 1 is redirected to the standard input of command 2.

Filters

Filters take the standard input and perform an operation upon it and sends the results to the standard output. This can be used to process information in powerful ways.

• *sort* - sorts the standard input and sends the output to standard output.
'sort filename' rearranges each line of file in alphabetical order and outputs it to the standard output.
• *uniq* - Given a sorted stream of data from standard input it removes the duplicate lines of data and returns the result to the standard output.
• *grep* - examines each line of data it receives from standard input and outputs all lines that contains a specific pattern of characters.
'grep "string" new.txt' outputs lines of text in new.txt which contain the word string.
• *fmt* - reads text from standard input and outputs formatted text to standard output.
'fmt filename' formats contents of filename and outputs it in standard output.
• *pr* - Takes data from the standard input and splits the data into pages with page breaks, footers and headers in preparation for printing.
'pr filename' displays the contents of the file one page after the other and returns the output to the standard output.

• *head* - Outputs the first few lines of the file and returns it to the standard output.
• *tail* - Outputs the last few lines of the file and returns it to the standard output.
• *tr* - Translates Characters. Can be used to perform tasks such as uppercase to lowercase conversions or changing the line termination characters from one type to another.
'tr [:lower:] [:upper:]' takes input from the keyboard and outputs each character of the input to uppercase characters and outputs it to the standard output.

Job Control

There are several commands used to control processes in Linux.
• *ps* - The ps commands lists the processes running in the system. 'ps lists' all processes running in the system
'ps aux' lists all processes running in the system.
• *kill* - sends a signal to the specified processes usually to stop the execution of the processes.
'kill -l' lists the signal names that can be sent to processes in Linux.
'kill pid' to kill the processes specified by the process id (pid) which can be obtained by the ps command.
'kill -s SIGKILL pid' is used to send SIGKILL signal to process with process id 'pid'. This command is used to forcefully kill a process without memory cleanup.
A signal is an asynchronous notification sent to a process or to a specific thread within the same process in order to notify it of an event that occurred.
• *jobs* - An alternate way of listing the processes. jobs is a shell builtin command which gives you information internal to the shell such as the job numbers.
'jobs' lists the jobs that the current shell is managing.
• *bg* - used to put a process in background.
• *fg* - used to put a process in foreground.

Permissions

chmod	modify file access rights
su	temporarily become super user
chown	change file ownership
chgrp	change file group ownership

Link

A link provides a connection between files. This provides the ability to have a single file or directory referred to through different names.
The nearest comparison with the Windows world is of a shortcut, but that is an unfair comparison as a link in Linux is far more powerful. A Windows shortcut is just a way of launching a file from a different place, whereas a link can make a file appear in multiple locations which is invisible to the applications.
There are two types of links that can be created. The first is a hard link and the other is a soft link (sometimes called symbolic link or symlink). The command to create these is the same - ln.
• *Hard Link* A hard link creates a second file that refers to the same file on the physical disk. This is achieved by having two filenames that point directly at the same file. This is normally used where file

entries (links) are on the same filesystem. When a hard link is created then all the names that link to that file are given the same status. Deleting one of the files will break the link, but the file can still exist under the other linked filenames. This works by maintaining a counter of the number of filenames that the file has. When the number of filenames reaches zero then the file is considered to be deleted and is removed. The number of filenames for a file can be seen using the 'ls -l' command. The number following the file permissions indicates the number of linked filenames.

The following screenshot shows that filename1 and filename2 are to linked files with one of the file denoted by the 2 (in this case the same file, but they could be to completely different files), the file not link is a single file denoted by the 1.
ls -l
total 2432
-rw-r–r– 2 stewart stewart 1241088 2009-01-23 15:26 filename1 -
rw-r–r– 2 stewart stewart 1241088 2009-01-23 15:26 filename2 -
rw-r–r– 1 stewart stewart 0 2009-01-23 15:26 not link
du -h
1.2M
ln filename1 filename2
The default for the ln command is a hard link. Assuming filename1 already exists filename2 is created using: ln filename1 filename2
• *Soft Link* A soft link is sometimes referred to as a symbolic link or symlink. A filename created as a soft link is a special file that has the pathname of the file to redirect to. Behind the scenes when you try and access a symlink it just goes to the filename referred to instead.

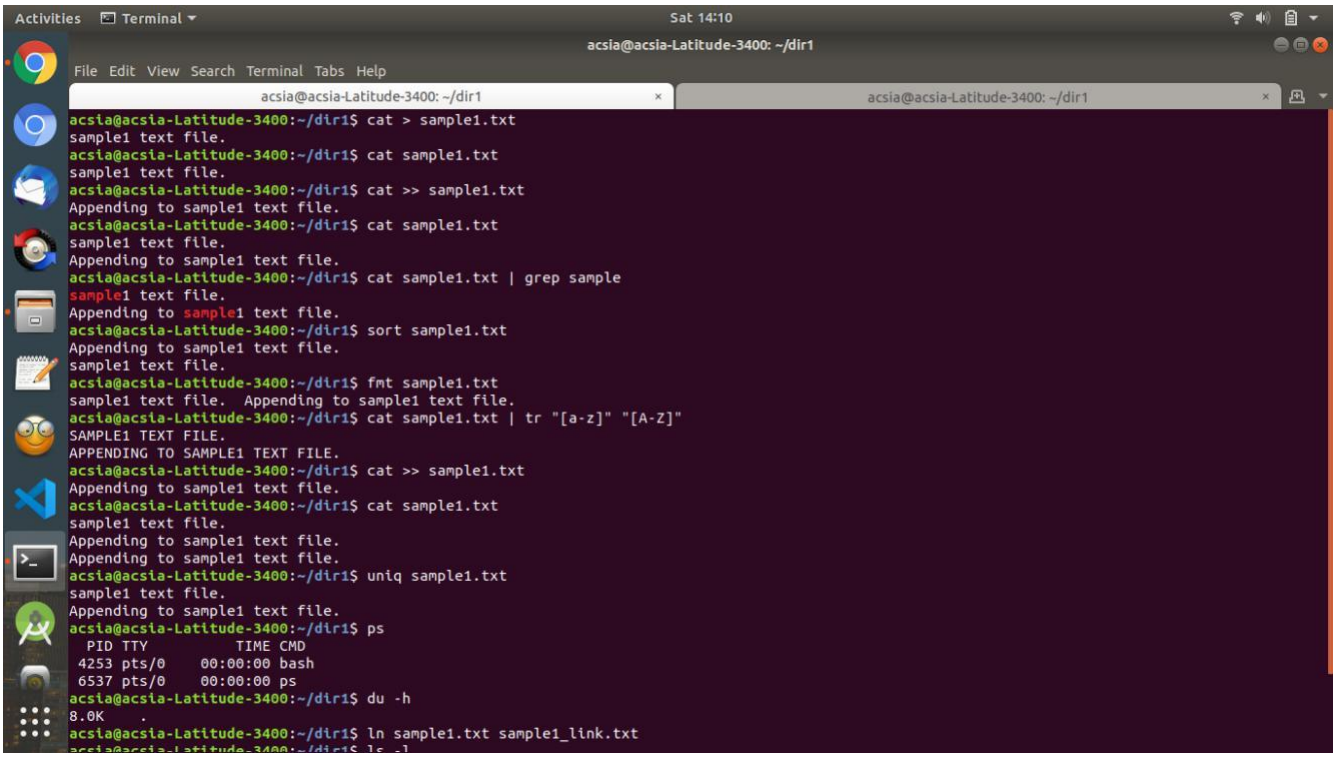
In the following example a file has been created called original file with a soft link to that same file called softlink tofile. As you can see the ls command makes it clear that this is a link through the l at the beginning of the file permissions and due to the reference notation after the filename. ls -l

total 440
-rw-r–r– 1 stewart stewart 446464 2009-01-23 15:21 original file
lrwxrwxrwx 1 stewart stewart 13 2009-01-23 15:20 softlink tofile -> original file
$ rm original file
$ ls -l
total 0
lrwxrwxrwx 1 stewart stewart 13 2009-01-23 15:20 softlink tofile -> original
file $ cat softlink tofile
cat: softlink tofile: No such file or directory
The -s option is used on the ln command to create a softlink.
ln -s original file softlink tofile

• Pitfalls while using links
- There are some things that you need to be aware of, particularly when using softlinks.
- Some programs (e.g. Apache) can be configured to not follow soft links (from 18a security
- When creating backup files you need to be aware of how the particular backup program / tool
- Using one method the program may not backup the file referenced resulting in an incomplete backup

- Using another method the program may end up backing up both as individualvfiles using double the space for that
- Using the second method of the above could result in the file being restored asva real file rather than as a link, breaking their connection
- The original file could be deleted without realising that there are other symlinked files referring to it Despite these pitfalls there are many advantages to using both hard and soft links to provide multiple references to files.

2.3 OUTPUT



2.4 RESULT

The linux commads for redirection of standard I/O, pipes, filters, job control and links in linux were studied and output was verified.

3. ADVANCED LINUX COMMANDS

3.1 AIM

Advanced linux commands curl, wget, ftp, ssh and grep.

3.2 OVERVIEW

curl

curl is a tool to transfer data from or to a server, using one of the supported protocols (DICT, FILE, FTP, FTPS, GOPHER, HTTP, HTTPS, IMAP, IMAPS, LDAP,LDAPS, POP3, POP3S, RTMP, RTSP, SCP, SFTP, SMB, SMBS, SMTP, SMTPS,TELNET and TFTP). The command is designed to work without user interaction.
'curl link' - gives information about the website and outputs the html code.
'curl -O link/file.html' - copies the html code of the website to file.html

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 replace rlogin and rsh, and provide secure encrypted communications between two untrusted hosts over an insecure network. X11 connections and arbitrary TCP/IP ports can also be forwarded over the secure channel. ssh connects and logs into the specified hostname (with optional user name). The user must prove his/her identity to the remote machine using one of several methods depending on the protocol version used. If command is specified, command is executed on the remote host instead of a login shell.

Basic Command

$ ssh remote host
The remote host in this example is the IP address or domain name that you are trying to connect to. If your username is different on the remote system, you can specify it by using this syntax: $ ssh remote username@remote host

TO COPY FILES
$ scp source destination

COPY A FILE FROM HOST TO LOCAL
$ scp localhostfile.txt
student@remotehost.example.com:/home/student/localhostfile.txt

COPY A FILE FROM LOCAL TO HOST
$ scp localhostfile.txt
student@remotehost.example.com:/home/student/localhostfile.txt

SHUTDOWN CONNECTED COMPUTER

$ ssh user@remote computer
$ sudo poweroff
$ sudo rebootwline

wget

GNU wget is a free utility for non-interactive download of files from the Web.It supports HTTP , HTTPS , and FTP protocols, as well as retrieval through HTTP proxies. Wget is non-interactive, meaning that it can work in the background, while the user is not logged on. This allows you to start a retrieval and disconnect from the system, letting Wget finish the work. By contrast, most of the Web browsers require constant user's presence, which can be a great hindrance when transferring a lot of data. Wget can follow links in HTML , XHTML , and CSS pages, to create local versions of remote web sites, fully recreating the directory structure of the original site. This is sometimes referred to as "recursive downloading." While doing that, Wget respects the Robot Exclusion Standard (/robots.txt). Wget can be instructed to convert the links in downloaded files to point at the local files, for offline viewing. Wget has been designed for robustness over slow or unstable network connections; if a download fails due to a network problem, it will keep retrying until the whole file has been retrieved. If the server supports regetting, it will instruct the server to continue the download from where it left off. wget http://www.openss7.org/repos/tarballs/strx25-0.9.2.1.tar.bz2 this command will download a single file and store it in the current repository. ewget -O wget.zip http://ftp.gnu.org/gnu/wget/wget-1.5.3.tar.gz Using -O (uppercase) option, downloads file with different file name. Here we have given wget.zip file name as show below.
wget -c http://www.openss7.org/repos/tarballs/strx25-0.9.2.1.tar.bz2 Restart a download which got stopped in the middle using wget -c option.
wget -mirror [WebsiteName] If you wish to retain a copy of any website that you may like to refer to/read locally, or maybe save a copy of your blog to the hard disk as back up, you may execute the wget command with mirror option. wget -i download-file-list.txt allows you to download multiple files stored in download file-list.txt simultaneously.

Ftp

The FTP (File Transfer Protocol) utility program is commonly used for copying files to and from other computers. These computers may be at the same site or at different sites thousands of miles apart. FTP is a general protocol that works on UNIX systems as well as a variety of other (non-UNIX) systems.
To connect your local machine to the remote machine, type ftp machinename where machinename is the full machine name of the remote machine, e.g., some-url.com. If the name of the machine is unknown, you may type ftp IP-Address where machinennumber is the net address of the remote machine, e.g., 129.82.45.181. In either case, this command is similar to logging onto the remote machine. If the remote machine has been reached successfully, FTP responds by asking for a loginname and password. When you enter your own loginname and password for the remote machine, it returns the prompt
ftp >
and permits you access to your own home directory on the remote machine. You should be able to move around in your own directory and to copy files to and from your local machine using the FTP interface commands.

FTP Interface Commands

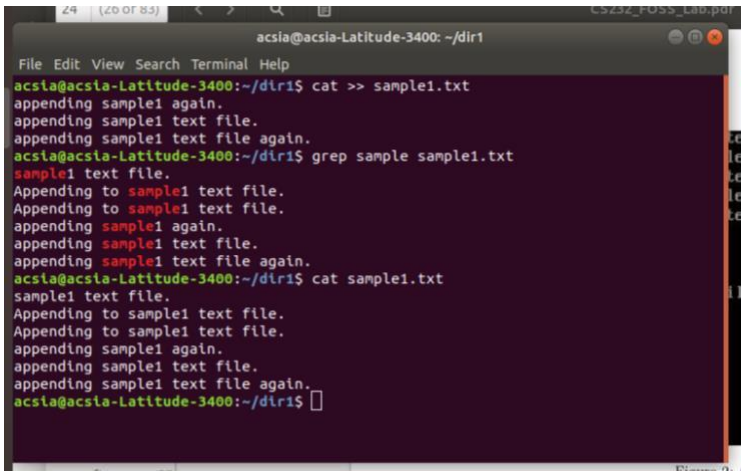
?	request help or information about FTP Commands
cd	change directory in remote machine
close	terminate a connection with remote computer
delete	delete a file in remote computer
get	get a copy of a file in the remote machine to local machine
ls	list the names of files in the current directory in the remote machine
mkdir	make a new directory in the remote machine
pwd	print the working directory in remote machine
quit	exit ftp environment

grep

GREP : Global Regular Expression Print, Searches for text in a file, Can search for simple words. Can look for regular expression - more complex character strings(words followed by any no of spaces, followed by a digit or lowercase letter). Searching the given string

‘$ grep <literal string> ’ filename To search a specific string in a specified file Case insensitive search
$ grep -i ?string? filename
$ grep -i ?string? FILE PATTERN
-This searches for given string/pattern case insensitively.
Simple regular expressions
?[0-9]? look for any digit
?[a-zA-Z]? look for one upper or lower case letter
?.? look for one charector
?.*? any number of charectors
?n.? a literal decimal point
?n.161:? dot, then 161, then colon
?n.161[:]? dot, then 161, then colon or space
Advanced regular expressions
Look for lines that hold either string1 or string2
$ grep -E ?(string1—string2)? filename
Lines that have string1 followed by string2 on the same line, but possibly with other charectors in between.
$ grep ?string1.*string2? filename
String1 has to be at the beginning of the line.
$ grep ?string1? filename Look for it at the end of the line. $ grep ?string1$? filename

3.3 OUTPUT



3.4 RESULT

The advanced linux commands ftp, ssh, wget, grep and curl were studied.