



COLLEGE OF ENGINEERING CHENGANNUR

CS232

**FREE AND OPEN
SOURCE SOFTWARE
LAB RECORD**

Submitted by CHN17CS088

Name N S ATHUL ANAND

Class S4-D

Roll No. 39

Certificate

Name :

Class :

Roll No :

Exam No :

*This is certified to be the bonafide record of practical work done in
Data Structures as per Syllabus of class..... in the Laborataroy
during the academic year 20 /20*

.....

Teacher In-charge

.....

Head of Dept.

.....

Examiner's Signature

.....

Principal

Date :

Institution Rubber Stamp

Contents

1- Linux Commands	
2 -CGPA calculation.	
3 -Networking	
4 -SSH.	
-Rsync.	
-SCP	
5-FTP	
6- Linux Installation	
7 -HTTP.	
-FTP	
8- Package Management	
-apt	
-Perl.	
- LAMP Stack.	
-Kernel Compilation	
9-Own webpage on server creation.	

TASK1

Linux commands

APT Provides a high-level CLI (Command Line Interface) for the package management system and is intended as an interface for the end user which enables some options better suited for interactive usage by default compared to more specialized APT tools like aptcache and apt-get

APT-GET

It is a command-line tool which helps in handling packages in Linux

AWK

It is a scripting language used for manipulating data and generating reports **CAT**

Reads data from file and gives their content as output. It helps us to create, view, concatenate files

CD

Known as change directory command. It is used to change current working directory

CURL

A tool to transfer data to or from a server, using any of the supported protocols **ECHO**

Used to display line of text/string that are passed as an argument

GREP

Searches a file for a particular pattern of characters, and displays all lines that contain that pattern

IFCONFIG

Used to configure the kernel-resident network interfaces.

MAN

Used to display the user manual of any command that we can run on the terminal **MKDIR**

Allows the user to create directories. This command can create multiple directories at once

PWD

It prints the path of the working directory, starting from the root

RM

Used to remove objects such as files, directories, symbolic links and so on from the file system like UNIX

SED

Used for finding, filtering, text substitution, replacement and text manipulations like insertion, deletion search etc.

SUDO

Used as a prefix of some command that only superuser are allowed to run

TASK2

CGPA CALCULATION

Computing CGPA needs to download the PDFs of the results and the file containing register number and names of students in our class.

Then sort the PDFs to create a file containing register number and marks of all courses of each students per line.

Then the file is run through a C program to get a file containing register number and CGPA of students, one per line.

This file is joined with the file containing the register number and name of students to get the final file containing register number, name and CGPA of each student

<https://github.com/ceccs17d39/CS232/tree/master/task2>

Task3

Networking

Ifconfig

change the configuration of the network interfaces on your system. up and down are used for connection and disconnection

eth0 is the first **Ethernet** interface. (Additional Ethernet interfaces would be named eth1, eth2, etc.)

This type of interface is usually a **NIC** connected to the network by a **category 5** cable.

lo is the **loopback** interface. This is a special network interface that the system uses to communicate with itself.

wlan0 is the name of the first **wireless network** interface on the system. Additional wireless interfaces would be named wlan1, wlan route ip **route add** is our command to **add new route**.

/etc/resolv.conf resolv.conf file is the resolver configuration file. It is use to configure client side access to the Internet Domain Name System (DNS). This file defines which name servers to use.

TASK4 SSH

SSH, or *Secure Shell*, is a protocol used to securely log onto remote systems. It is the most common way to access remote Linux and Unix-like servers. SSH has remained popular because it is secure, light-weight, and useful in diverse situations. SSH provides a secure channel over an unsecured network in a client–server architecture, connecting an SSH client application with an SSH server.^[2] The protocol specification distinguishes between two major versions, referred to as SSH-1 and SSH-2. The standard TCP port for SSH is 22. SSH is generally used to access Unix-like operating systems

SSH works by connecting a client program to an ssh server. *ssh* is the client program. The *ssh server* is already running on the *remote_host*. When you change the configuration of SSH, you are changing the settings of the *sshd* server. SSH has remained popular because it is secure, lightweight, and useful in diverse situations.

Rsync

Rsync is a fast and extraordinarily 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 behavior 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 the differences between the source files and the existing files in the destination. Rsync is widely used for backups and mirroring and as an improved copy command for everyday use. Rsync is a command line utility. Rsync is installed in Ubuntu by default

The simplest method for backing up over a network is to use rsync via SSH (using the *e ssh* option). Alternatively, you can use the rsync daemon . backup only requires rsync and read/write access to the folders being synchronized.

Scp

Scp (Secure Copy) is a command line tool to copy or transfer files across hosts. It uses the same kind of security mechanism like the *ssh* program. Infact it uses an *ssh* connection in the background to perform the file transfer. *scp* refers both to the "protocol" that defines how secure copy should work and the "program" (command) which is installed as a part of OpenSSH suite of tools. Scp is generally installed by default on most linux distros as a part of openssh packages. On ubuntu/debian for example, the openssh-client package provides the *scp* program

TASK5

FTP

The File Transfer Protocol (FTP) is a standard network protocol used for the transfer of computer files between a client and server on a computer network. FTP is the simplest file transfer protocol to exchange files to and from a remote computer or network. Similar to Windows, Linux and UNIX operating systems also have built-in command-line prompts that can be used as FTP clients to make an FTP connection

. FTP is built on a client-server model architecture using separate control and data connections between the client and the server. FTP users may authenticate themselves with a clear-text sign-in protocol, normally in the form of a username and password, but can connect anonymously if the server is configured to allow it. For secure transmission that protects the username and password, and encrypts the content, FTP is often secured with SSL/TLS (FTPS) or replaced with SSH File Transfer Protocol (SFTP).

1. `!` Escape to the Shell
2. `$` Execute macro
3. `?` print local help information
4. `account` Send account command to remote server
5. `append` Append to a file
6. `ascii` set ascii transfer type
7. `beep` beep when command completed
8. `binary` Set Binary transfer type
9. `bye` Terminate ftp session and exit
10. `case` toggle mget upper/lower case id mapping
11. `cd` Change remote working directory
12. `cdup` change remote working directory to parent directory
13. `chmod` Change file permissions of remote file
14. `close` Terminate FTP session
15. `cr` toggle carriage return stripping on ascii gets
16. `debug` toggle/set debugging mode
17. `delete` delete remote file
18. `dir` list contents of remote directory
19. `disconnect` terminate ftp session
20. `exit` terminate ftp sessions and exit
21. `form` set file transfer format
22. `get` receive file
23. `glob` toggle meta character expansion of local file names

- 24. hash toggle printing '#' for each buffer transferred
- 25. help display local help information
- 26. idle get (set) idle timer on remote side
- 27. image set binary transfer type
- 28. ipany allow use of any address family
- 29. ipv4 restrict address usage to ipv4
- 30. ipv6 restrict address usage to ipv6
- 31. lcd Change local working directory
- 32. ls list contents of remote directory
- 33. macdef define a macro
- 34. mdelete delete multiple files
- 35. mdir list contents of multiple remote directories
- 36. mget get multiple files
- 37. mkdir make directory on remote machine
- 38. mls list contents of multiple remote directories
- 39. mode set file transfer mode
- 40. modtime show last modification time of remote file
- 41. mput send multiple files
- 42. newer get file if remote file is newer than local file
- 43. nlist nlist contents of remote directory
- 44. nmap set templates for default file name mapping
- 45. ntrans set translation table for default file name mapping
- 46. open connect to remote ftp
- 47. passive enter passive transfer mode
- 48. prompt force interactive prompting on multiple commands
- 49. proxy issue command on an alternate connection 50. put send one file
- 51. pwd print working directory on remote machine
- 52. qc print ? in place of control characters on stdout
- 53. quit terminate ftp session and exit
- 54. quote send arbitrary ftp command
- 55. recv receive file
- 56. reget get file restarting at end of local file
- 57. rename rename file
- 58. reset clear queued command replies
- 59. restart restart file transfer at bytecount
- 60. rhelp get help from remote server
- 61. rmdir remove directory on remote machine
- 62. rstatus show status of remote machine
- 63. runique toggle store unique for local files

- 64. send send one file
- 65. sendport toggle use of PORT cmd for each data connection
- 66. site send site specific command to remote server
- 67. size show size of remote file
- 68. status show current status
- 69. struct set file transfer structure
- 70. sunique toggle store unique on remote machine
- 71. system show remote system type
- 72. tenex set tenex file transfer type
- 73. tick toggle printing byte counter during transfers
- 74. trace toggle packet tracing
- 75. type set file transfer type
- 76. umask get (set) umask on remote site
- 77. user send new user information
- 78. verbose toggle verbose mode

Task6

Linux installation

Linux is open-source, free to use kernel. It is used by programmers, organizations, profit and nonprofit companies around the world to create Operating systems to suit their individual requirements. To prevent hacking attempts, many organizations keep their Linux operating systems private. Many others make their variations of Linux available publicly so the whole world can benefit at large. These versions/ types /kinds of Linux operating system are called Distributions.

There are hundreds of Linux operating systems or Distributions available these days. Many of them are designed with a specific purpose in mind. For example, to run a web server or to run on network switches like routers, modems, etc

- Arch This Linux Distro is popular amongst Developers. It is an independently developed system. It is designed for users who go for a do-it-yourself approach.
- CentOS It is one of the most used Linux Distribution for enterprise and web servers. It is a free enterprise class Operating system and is based heavily on Red Hat enterprise Distro.
- Debian Debian is a stable and popular non-commercial Linux distribution. It is widely used as a desktop Linux Distro and is user-oriented. It strictly acts within the Linux protocols.
- Fedora Another Linux kernel based Distro, Fedora is supported by the Fedora project, an endeavor by Red Hat. It is popular among desktop users. Its versions are known for their short life cycle.
- Gentoo It is a source based Distribution which means that you need to configure the code on your system before you can install it. It is not for Linux beginners, but it is sure fun for experienced users.
- LinuxMint It is one of the most popular Desktop Distributions available out there. It launched in 2006 and is now considered to be the fourth most used Operating system in the computing world.
- OpenSUSE It is an easy to use and a good alternative to MS Windows. It can be easily set up and can also run on small computers with obsolete configurations.
- RedHat enterprise Another popular enterprise based Linux Distribution is Red Hat Enterprise. It has evolved from Red Hat Linux which was discontinued in 2004. It is a commercial Distro and very popular among its clientele.
- Slackware Slackware is one of the oldest Linux kernel based OS's. It is another easy desktop Distribution. It aims at being a 'Unix like' OS with minimal changes to its kernel.
- Ubuntu This is the third most popular desktop operating system after Microsoft Windows and Apple Mac OS. It is based on the Debian Linux Distribution, and it is known as its desktop environment

BASIC STEPS FOR INSTALLING LINUX DISTRIBUTIONS

1. Download the Linux distribution of your choice.. Linux distributions (known as "distros") are typically available for free to download in ISO format. You can find the ISO for the distribution of your choice at the distribution's website. This format needs to be burned to a CD or USB stick before you can use it to install Linux. This will create a Live CD or Live USB.

A Live CD or Live USB is a disk that you can boot into, and often contains a preview version of the operating system that can be run directly from the CD or USB stick. Install an image burning program, or use your system's built-in burning tool if you are using Windows 7, 8, or Mac OS X. Pen Drive Linux and UNetBootin are two popular tools for burning ISO files to USB sticks.

2. Boot into the Live CD or Live USB. Most computers are set to boot into the hard drive first, which means you will need to change some settings to boot from your newly-burned CD or USB. Start by rebooting the computer.

Once the computer reboots, press the key used to enter the boot menu. The key for your system will be displayed on the same screen as the manufacturer's logo. Typical keys include F12, F2, or Del.

For Windows 8 users, hold the Shift key and click restart. This will load the Advanced Startup Options, where you can boot from CD.

For Windows 10 users, go to advanced boot in settings and click "Restart Now."

If your computer doesn't give you direct access to the boot menu from the manufacturer's splash screen, it's most likely hidden in the BIOS menu. You can access the BIOS menu in the same way that you would get to the boot menu. At the manufacturer splash screen, the key should be listed in one of the bottom corners.

Once you're in the boot menu, select your live CD or USB. Once you've changed the settings, save and exit the BIOS setup or boot menu. Your computer will continue with the boot process.

3. Start the installation process. If you're trying out the distro, you can launch the installation from the application on the desktop. If you decided not to try out the distribution, you can start the installation from the boot menu.

You will be asked to configure some basic options, such as language, keyboard layout, and timezone.

4. Create a username and password. You will need to create login information to install Linux. A password will be required to log into your account and perform administrative tasks.

5. Set up the partition. Linux needs to be installed on a separate partition from any other operating systems on your computer if you intend dual booting Linux with another OS. A partition is a portion of the hard drive that is formatted specifically for that operating system. You can skip this step if you don't plan on dual booting.

Distros such as Ubuntu will set a recommended partition automatically. You can then adjust this manually yourself. Most Linux installations require at least 20 GB, so be sure to set aside enough room for both the Linux operating system and any other programs you may install and files you may create.

If the installation process does not give you automatic partitions, make sure that the partition you create is formatted as Ext4. If the copy of Linux you are installing is the only operating system on the computer, you will most likely have to manually set your partition size.

6. Boot into Linux. Once the installation is finished, your computer will reboot. You will see a new screen when your computer boots up called "GNU GRUB". This is a boot loader that handles Linux installations. Pick your new Linux distro from the list. This screen may not show up if you only have one operating system on your computer. If this screen isn't being presented to you automatically, then you can get it back by hitting shift right after the manufacturer splash screen. If you install multiple distros on your computer, they will all be listed here.
7. Check your hardware. Most hardware should work out of the box with your Linux distro, though you may need to download some additional drivers to get everything working.

Some hardware requires proprietary drivers to work correctly in Linux. This is most common with graphics cards. There is typically an open source driver that will work, but to get the most out of your graphics cards you will need to download the proprietary drivers from the manufacturer.

In Ubuntu, you can download proprietary drivers through the System Settings menu. Select the Additional Drivers option, and then select the graphics driver from the list. Other distros have specific methods for obtaining extra drivers.

You can find other drivers from this list as well, such as Wi-Fi drivers.

8. Start using Linux. Once your installation is complete and you've verified that your hardware is working, you're ready to start using Linux. Most distros come with several popular programs installed, and you can download many more from their respective file repositories.

TASK7

HTTP AND FTP SERVERS

HTTP is the abbreviation of "HyperText Transfer Protocol". The protocol specifies how the information must be requested and how the responses are formed, so we have two important actors here: the HTTP Client (well known as Browser) and the HTTP Server. The server is not restricted to server static content, it also serves dynamic content generated on fly from a database or similar. NGINX is a free, open-source, high-performance HTTP server and reverse proxy, as well as an IMAP/POP3 proxy server. NGINX is known for its high performance, stability, rich feature set, simple configuration, and low resource consumption

apt install nginx-To install nginx server

nginx -s start-To start nginx server

nginx -s stop-To stop nginx server

sudo systemctl status nginx-to view status

The way nginx and its modules work is determined in the configuration file. By default, the configuration file is named nginx.conf and placed in the directory /usr/local/nginx/conf, /etc/nginx, or /usr/local/etc/nginx. The most common server hosting directory (server root) is /var/www/html/. It contains configuration settings like ports, hostnames, etc.. By creating an index.html at the server root directory and restart nginx will make nginx load the index.html file when the IP address of the server machine is accessed through the same machine or a machine connected to the same network

FTP stands for File Transfer Protocol. It was written by Abhay Bhushan and published in 1971. FTP is supported by all the operating systems and browsers. It is a client-server based protocol. How FTP works

Step a: Client connects to server on port 21.

Step b: Server responds and ask for authentication.

Step c: Client decides weather to connect passively or actively and authenticate with credentials(user name password).

Step d: If it is an active connection, server opens port 20 for data transfer and gives ftp prompt after successful authentication.

there are several FTP servers available for you to use, commercial and open source. Vsftpd has some security features which makes it on the top like: Can run as a normal user with privilege separation. Supports SSL/TLS FTP connections. Can jail users into their home directories.. Remote file transfer can be done through any FTP clients in conjunction with an FTP server. Linux has a builtin FTP client, ftp. \$ sudo apt-get install vsftpd >>\$ systemctl start vsftpd >>\$ ftp localhost Then type the username enter. You will see the FTP prompt. And now type any FTP command to interact with the FTP server.

TASK8

Package management

Contemporary distributions of Linux-based operating systems install software in pre-compiled packages, which are archives that contain binaries of software, configuration files, and information about dependencies. Furthermore, package management tools keep track of updates and upgrades so that the user doesn't have to hunt down information about bug and security fixes.

Without package management, users must ensure that all of the required dependencies for a piece of software are installed and up-to-date, compile the software from the source code (which takes time and introduces compiler-based variations from system to system), and manage configuration for each piece of software. Without package management, application files are located in the standard locations for the system to which the developers are accustomed, regardless of which system they're using.

- `apt-get install package-name(s)` - Installs the package(s) specified, along with any dependencies.
- `apt-get remove package-name(s)` - Removes the package(s) specified, but does not remove dependencies.
- `apt-get autoremove` - Removes any orphaned dependencies, meaning those that remain installed but are no longer required.
- `apt-get clean` - Removes downloaded package files (.deb) for software that is already installed.
- `apt-get purge package-name(s)` - Combines the functions of remove and clean for a specific package, as well as configuration files.
- `apt-get update` - Reads the `/etc/apt/sources.list` file and updates the system's database of packages available for installation. Run this after changing `sources.list`.
- `apt-get upgrade` - Upgrades all packages if there are updates available. Run this after running `apt-get update`.
- While `apt-get` provides the most often-used functionality, APT provides additional information in the `apt-cache` command.
- `apt-cache search package-name(s)` - If you know the name of a piece of software but `apt-get install` fails or points to the wrong software, this looks for other possible names.
- `apt-cache show package-name(s)` - Shows dependency information, version numbers and a basic description of the package.
- `apt-cache depends package-name(s)` - Lists the packages that the specified packages depends upon in a tree. These are the packages that will be installed with the `apt-get install` command.
- `apt-cache rdepends package-name(s)` - Outputs a list of packages that depend upon the specified package. This list can often be rather long, it is best to pipe its output through a command, like `less`.

- `apt-cache pkgnames` - Generates a list of the currently installed packages on your system. This list is often rather long, so it is best to pipe its output through a program, like `less`, or direct the output to a text file.

Perl

Perl is a family of two high-level, general-purpose, interpreted, dynamic programming languages. "Perl" usually refers to Perl 5, but it may also refer to its redesigned "sister language", Perl 6. backronyms in use, "Practical Extraction and Reporting Language". Perl was originally developed by Larry Wall in 1987 as a general-purpose Unix scripting language to make report processing easier.

The Perl languages borrow features from other programming languages including C, shell script (sh), AWK, and sed Wall also alludes to BASIC and Lisp in the introduction to Learning Perl (Schwartz & Christiansen) and so on. They provide text processing facilities without the arbitrary data-length limits of many contemporary Unix commandline tools, facilitating manipulation of text files. Perl 5 gained widespread popularity in the late 1990s as a CGI scripting language, in part due to its then unsurpassed regular expression and string parsing abilities. Perl program files end with `".pl"`. an example of a perl program

```
use strict;
```

```
use warnings;
```

```
print "Enter two numbers \n";
```

```
$a = <>; $b = <>; my
```

```
$sum = $a+$b; print
```

```
"Sum = $sum\n";
```

LAMP stack

A LAMP Stack is a set of open-source software that can be used to create websites and web applications. LAMP is an acronym, and these stacks typically consist of the Linux operating system, the Apache HTTP Server, the MySQL relational database management system, and the PHP

programming language. Linux based web servers consist of four software components. These components, arranged in layers supporting one another, make up the software stack. Websites and Web Applications run on top of this underlying stack. The common software components that make up a traditional LAMP stack are:

- Linux: The operating system (OS) makes up our first layer. Linux sets the foundation for the stack model. All other layers run on top of this layer.
- Apache: The second layer consists of web server software, typically Apache Web Server. Web servers are responsible for translating from web browsers to their correct website.
- MySQL: Our third layer is where databases live. MySQL stores details that can be queried by scripting to construct a website. MySQL usually sits on top of the Linux layer alongside Apache/layer 2. 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 and/or other similar web programming languages. Websites and Web Applications run within this layer

Most commonly MySQL is replaced by MariaDB since the latter is an open source fork of the former. All the packages required for the lamp stack can be installed through

```
# apt install apache2 mariadb-server php \ libapache2-mod-php php-mysql
```

Deleting all the contents of `/var/www/html` and adding a new php file and then restarting apache2 server will result in the newly created php file

Kernel compilation

compiling a custom kernel has its advantages and disadvantages. However, new Linux user/admin find it difficult to compile Linux kernel. Compiling kernel needs to understand few things and then type a couple of commands. The linux kernels can be downloaded from <https://kernel.org/>.

The procedure to build (compile) and install the latest Linux kernel from source is as follows:

1. Grab the latest kernel from kernel.org
2. Verify kernel
3. Untar the kernel tarball
4. Copy existing Linux kernel config file
5. Compile and build Linux kernel 4.20.12
6. Install Linux kernel and modules (drivers)
7. Update Grub configuration
8. Reboot the system

Some commands for basic operations:

\$ make menuconfig-:Configuring the kernel

\$ make-:to compile kernel

\$ make modules_install-:compile kernel modules

make install-:to install the kernel

TASK9

OWN SERVER CREATION

An html file named index.html is created which links to the contents in list to the git.the index.html file is then placed in required server by accesing it with sftp
<http://14.139.189.217/cs17d/cs17d39>

The pages could be accessed as [http:// 14.139.189.217/cs17d/cs1739](http://14.139.189.217/cs17d/cs1739)