FREE AND OPEN SOURCE SOFTWARE

LAB REPORT

Name : ASHISH MATHEW JOHN

Roll No. : 13

Class : S4 D

KTU Reg No. : chn17cs030

INDEX

EXPERIMENT NO. 1

LINUX COMMANDS

DATE:04/02/2019

There are many linux commands , this section gives insight in to the most important commands of Linux system. The first task is done to familiarize with the most important Linux commands. The important commands and its uses are given below. To learn more about the various commands we used ‘man’ program followed by the name of the command

- ls : The program will list the contents of the current directory in short form
- ls -l : The ls -l command functions in the linux terminal to show all major directories filed with all details including file name, file size and date under a given file system.
- ls -lt : The ls -lt command functions in the linux terminal to show all major directories with its time band filed under a given file system.
- ls -ltr : The ls -ltr command functions in the linux terminal to show all major directories in reverse order.
- man : To show manual of inputed commands.
- mv : Move command allows user to move a file to other directories.
- clear : The clear command clears the screen, wipes out the board clean.
- mkdir : Make a directory command allows the user to make a new directory.
- cat : use the cat command to display contents of file. It is usually used to easily view programs.
- sudo : It stands for super user do. So, if you want any command tobe done with administrative or root privilages.
- cd : Change Directory command -This command allows the user change between file directories.
- grep : The grep command searches the substring inside the file
- chmod : Used to change the access mode of a file.
- touch : Admins can create a blank file within Linux with the touch command.

These are some of the commands that familiarized in the first day of the linux introduction class. I tested each command and desired output obtained. The commands and it’s outputs were uploaded according to the instruction given.

EXPERIMENT NO. 2

SCRIPTING TASK

DATE:13/02/2019

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.

EXPERIMENT NO. 3

NETWORKING TASK

DATE:04/04/2019

This task is done to familiarize with the commands that can be used to set up a computer network and share data. Computers are connected in a network to exchange information or resources each other. 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. Computer loaded with Linux Operating System can also be a part of network whether it is small or large network by its multitasking and multiuser natures. In this task we used hub and interconnected three computer systems using the following commands
1. ifconfig

ifconfig (interface configurator) command is use to initialize an interface, assign IP Address to interface and enable or disable interface on demand. With this command you can view IP Address and Hardware / MAC address assign to interface and also MTU (Maximum transmission unit) size.

Assigning IP Address and Gateway

Assigning an IP Address and Gateway to interface on the fly. The setting will be removed in case of system reboot.

```
# ifconfig eth0 192.168.50.5 netmask 255.255.255.0
```

Enable or Disable Specific Interface

To enable or disable specific Interface, we use example command as follows.
Enable eth0

```
# ifup eth0
Disable eth0
# ifdown eth0
```

2. PING Command

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

These are some commands familiarised using this task. I worked with each commands. The result uploaded according to the instruction given. Output of the commands verified.

EXPERIMENT NO. 4

SSH

DATE:04/04/2019

This task is done to familiarize with the important ssh commands. ssh stands for "**Secure Shell"**. It is a protocol used to securely connect to a remote server/system. ssh is secure in the sense that it transfers the data in encrypted form between the host and the client. It transfers inputs from the client to the host and relays back the output.
The basic commands in ssh are :

ssh remote_host

The remote_host in this example is the IP address or domain name that you are trying to connect to.
This command assumes that your username on the remote system is the same as your username on your local system.
If your username is different on the remote system, you can specify it by using the syntax :

ssh remote_username@remote_host

Once you have connected to the server, you will probably be asked to verify your identity by providing a password.
To exit back into your local session, you can use the command :

exit

There is a command that can be used to create a new directory. It simply creates a new directory with your chosen name. For this we can use the command :

`mkdir`

These are some commands familiarised using this task. I worked with each commands. The result uploaded according to the instruction given. Output of the commands verified.

EXPERIMENT NO. 5

SCP

DATE:04/04/2019

This task is done to familiarize with the important scp commands. SCP (secure copy) is a command line utility that allows you to securely copy files and directories between two locations.
With scp, you can copy a file or directory:

- From your local system to a remote system.
- From a remote system to your local system.
- Between two remote systems from your local system.

When transferring data with scp both the files and password are encrypted so that anyone snooping on the traffic doesn't get anything sensitive.

The basic syntax of SCP is :

scp source_file_path destination_file_path

Depending on the host, the file path should include the full host address, port number, username and password along with the directory path.if you are sending file from your local machine to a remote machine (uploading) the syntax will be:

scp ~/my_local_file.txt user@remote_host.com:/some/remote/directory

For copying file from remote host to local host (downloading), the syntax will be :

scp user@remote_host.com:/some/remote/directory ~/my_local_file.txt

Apart from it, there are a couple of extra options and functions that scp supports. By default scp will always overwrite files on the destination. scp provides a number of options that control every aspect of its behavior. The most widely used options are:

- `-P` Specifies the remote host ssh port.
- `-p` Preserves files modification and access times.
- `-q` Use this option if you want to suppress the progress meter and non-error messages.
- `-C`. This option will force scp to compresses the data as it is sent to the destination machine.
- `-r` This option will tell scp to recursively copy directories.

Thus by using this scp commands we transferred data to and from the server.

EXPERIMENT NO. 6

RSYNC

DATE:04/04/2019

This task is done to familiarize with the important RSYNC commands. Rsync (Remote Sync) is a most commonly used command for copying and synchronizing files and directories remotely as well as locally in Linux/Unix systems. With the help of rsync command you can copy and synchronize your data remotely and locally across directories, across disks and networks .
The basic syntax of rsync is :

rsync options source destination

Some common options used with rsync command are:
1. **-v** : verbose
2. **-r** : copies data recursively
3. **-a** : archive mode, archive mode allows copying files recursively and it also preserves symbolic links, file permissions, user & group ownerships and timestamps
4. **-z** : compress file data
5. **-h** : human-readable, output numbers in a human-readable format

We can install **rsync** package with the help of following command :

apt-get install rsync

Copy/Sync a File on a Local Computer:
This following command will sync a single file on a local machine from one location to another location. Here in this example, a file name backup.tar needs to be copied or synced to /tmp/backups/ folder.

rsync -zvh backup.tar /tmp/backups/

Copy/Sync a Directory on Local Computer
The following command will transfer or sync all the files of from one directory to a different directory in the same machine. Here in this example, /root/rpmpkgs contains some rpm package files and you want that directory to be copied inside /tmp/backups/ folder.

rsync -avzh /root/rpmpkgs /tmp/backups/

Copy a Directory from Local Server to a Remote Server
This command will sync a directory from a local machine to a remote machine. For example: There is a folder in your local computer “rpmpkgs” which contains some RPM packages and you want that local directory’s content send to a remote server, you can use following command.

rsync -avz rpmpkgs/ root@192.168.0.101:/home/

In this section we familiarized with various rsync commands that can be used for data transfer

EXPERIMENT NO. 7

FTP USAGE

DATE:11/03/2019

This task is done to familiarize with the important FTP commands.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. When transferring data over ftp the connection is not encrypted. For a secure data transfer, use SCP commands.
To open an ftp connection to a remote system use the `ftp` command followed by the remote server IP address or domain name:

```
ftp 192.168.42.77
```

If the connection is established, a confirmation message will be displayed and you will be prompted to enter your FTP username.Once you enter the username you will be prompted to type your password .If the password is correct, the remote server will display a confirmation message and the `ftp>` prompt. Through that we can communicate with the server and exchange information. We can also use sftp which conduct an interactive FTP session over a secure network connection.

Below are some of the most common FTP commands :

- `help` or `?` - list all available FTP commands.
- `cd` - change directory on the remote machine.
- `lcd` - change directory on the local machine.
- `ls` - list the names of the files and directories in the current remote directory.
- `mkdir` - create a new directory within the current remote directory.
- `pwd` - print the current working directory on the remote machine.
- `delete` - remove a file in the current remote directory.
- `rmdir`- remove a directory in the current remote directory.
- `get` - copy one file from the remote to the local machine.
- `mget` - copy multiple files from the remote to the local machine.
- `put` - copy one file from the local to the remote machine.
- `mput` - copy one file from the local to the remote machine.

These are some commands familiarised using this task. I worked with each commands. The result uploaded according to the instruction given. Output of the commands verified.

EXPERIMENT NO. 8

OS INSTALLATION

DATE:11/03/2019

This task is done to study how to install an OS in to a system.Linux is open-source, free to use kernel. It is used by programmers, organizations,
profit and non-profit companies around the world to create Operating systems to suit their individual requirements. It is free to download and install on any computer. Because it is open source, there are a variety of different versions, or distributions, available developed by different groups.

In this task in the lab, we were given CD's to install linux in the preferred systems inside the lab.it just takes maximum 30 min to complete it just by inserting it in the system. The different steps that is performed are :

• Boot into the Live CD: 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.

• Try out the Linux distribution before installing. Most Live CDs can launch a live environment, giving you the ability to test it out before making the switch. You won't be able to create files, but you can navigate around the interface and decide if it's right for you.

• Start the installation process. If you're trying out the distribution, 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.

• 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.

• 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.

• Boot into Linux.

These are the various steps performed to install an OS in to a system. Through these steps I successfully completed OS installation task and uploaded the result according to the instruction given

EXPERIMENT NO. 9

HTTP AND FTP SERVER

DATE:12/04/2019

In this task we familiarized with various commands that are used to set up a http and ftp servers

FTP server:

An FTP Server (which stands for File Transfer Protocol Server) is a software application which enables the transfer of files from one computer to another. FTP is a way to transfer files to any computer in the world that is connected to the internet. 11/04/2019 one act as a client, the other act as a server. Here I used the vsftpd program which is a very popular FTP server that is used by many servers today to set up ftp server

The commands used to setup a ftp server are :

- *sudo apt install vsftpd* -installs vsftpd
- *sudo vsftpd* -configure vsftpd
- *ftp localhost* -view

HTTP server:

A web server processes incoming network requests over HTTP and several other related protocols. The primary function of a web server is to store, process and deliver web pages to clients. The communication between client and server takes place using the Hypertext Transfer Protocol (HTTP).

In this task I used nginx HTTP server.The nginx HTTP server is also a frequently used web server in the world. It provides many powerful features, including dynamically loadable modules, robust media support, and extensive integration with other popular software.

The different steps to setup a http server are:

1. Installing nginx commands:
   sudo apt update
   sudo apt install apache2
2. Adjusting firewall
   sudo ufw app list
   sudo ufw allow Apache
   sudo ufw status
3. checking web server
   sudo systemctl status apache2

These are the various steps performed to set up http and ftp servers. This task helped us to familiarize with various commands that can be used to setup a http and ftp server.The task is successfully completed and output is uploaded according to the instruction given

EXPERIMENT NO. 10

PACKAGE MANAGEMENT

DATE:29/04/2019

In this task we studied various commands to manage different packages in a computer system. It includes the commands used to download, update and upgrade the apps in the system through the terminal. The different commands that can be used for package management are :

- 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 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.
- apt-get upgrade - Upgrades all packages if thereare updates available.
- 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.

These are the various commands that are used in this task for package manegement. I worked with each commands and the desired output obtained . The results are uploaded according to the instruction given.

EXPERIMENT NO. 11

PERL

DATE:29/04/2019

In this task we studied the basics of perl programming. Perl 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.Perl is included by default with most GNU/Linux distributions. Usually, one invokes Perl by using a text editor to write a file and then passing it to the perl program. Perl scripts can be named anything but conventionally end with ".pl". All statements in perl are terminated with a semicolon .
Using perl I have created two programmes, one programe to print ‘Hello” and another programe to find the sum of two numbers.

Consider the following example :

```
$a=5;
$b=7;
$c = $a + $b;
print “The sum is $c \n
```

Here a and b are two integer values and c returns the sum of a and b . If we save this file with .pl extension ,then we can run it from the command line as follows

perl name.pl

If we don't want to type "perl" in order to run the script, we can put this line:

#!/usr/bin/perl

at the start (be sure to use the correct path on your system), and do chmod +x name.pl to make it executable. Then we type ./name.pl to run it.

Thus in this task I have familiarized with the basics of perl programming , worked with each commands and desired output obtained. The results are uploaded according to the instructions given

EXPERIMENT NO. 12

KERNEL COMPILATION

DATE:29/04/2019

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. The task is successfully completed and the output is uploaded according to the instruction given

EXPERIMENT NO. 13

LAMP STACK(Detailed report)

DATE:25/04/2019

In this task we familiarized with various commands to set up a website usong 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.

The Four Layers of a LAMP Stack

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. This layer resides on top of the Linux layer. 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.

To setup a webpage using lampstack first of all I installed the different layers of lamp stack ,which are Apache , MySQL(Mariadb) and PHP using the command:

apt-get install

After you have Ubuntu up and running, you'll want to make sure that everything on your system is current. To do that, open the terminal and type in the commands shown in the next page

Code (Shell script)

- #!/bin/sh
- sudo apt update
- sudo apt install apache2
- sudo systemctl enable apache2
- sudo apt-get install mariadb-server mariadb-client
- sudo mysql_secure_installation
- sudo mysql -u root -p
- sudo apt-get install software-properties-common
- sudo add-apt-repository ppa:ondrej/php
- sudo apt install php7.2 libapache2-mod-php7.2 php7.2-common php7.2-mbstring php7.2-xmlrpc php7.2-soap php7.2-gd php7.2-xml php7.2-intl php7.2-mysql php7.2- cli php7.2-zip php7.2-curl
- sudo nano /var/www/html/index.php
- sudo systemctl restart apache2
- firefox http://localhost

Type script:

```
cec@cec:~$ sudo ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
  CUPS
  Nginx Full
  Nginx HTTP
```

```
  Nginx HTTPS
  OpenSSH
cec@cec:~$ sudo ufw app info "Apache Full"
Profile: Apache Full
Title: Web Server (HTTP,HTTPS)
Description: Apache v2 is the next generation of the omnipresent Apache web
server.

Ports:
  80,443/tcp
cec@cec:~$ sudo ufw allow in "Apache Full"
Rule added
Rule added (v6)
cec@cec:~$
cec@cec:~$
cec@cec:~$     sudo apt install php libapache2-mod-php php-mysql
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  gyp libc-ares2 libhttp-parser2.8 libjs-async libjs-inherits
  libjs-is-typedarray libjs-node-uuid libssl1.0-dev libuv1 libuv1-dev
  nodejs-doc
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libapache2-mod-php7.2 php-common php7.2 php7.2-cli php7.2-common php7.2-json
  php7.2-mysql php7.2-opcache php7.2-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  libapache2-mod-php libapache2-mod-php7.2 php php-common php-mysql php7.2
  php7.2-cli php7.2-common php7.2-json php7.2-mysql php7.2-opcache
  php7.2-readline
0 upgraded, 12 newly installed, 0 to remove and 53 not upgraded.
Need to get 4,067 kB of archives.
After this operation, 18.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://in.archive.ubuntu.com/ubuntu cosmic/main amd64 php-common all 1:62 [11.9 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu cosmic-updates/main amd64 php7.2-common amd64
7.2.17-0ubuntu0.18.10.1 [888 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu cosmic-updates/main amd64 php7.2-json amd64 7.2.17-
0ubuntu0.18.10.1 [18.9 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu cosmic-updates/main amd64 php7.2-opcache amd64
7.2.17-0ubuntu0.18.10.1 [166 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu cosmic-updates/main amd64 php7.2-readline amd64
7.2.17-0ubuntu0.18.10.1 [12.3 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu cosmic-updates/main amd64 php7.2-cli amd64 7.2.17-
0ubuntu0.18.10.1 [1,444 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu cosmic-updates/main amd64 libapache2-mod-php7.2
amd64 7.2.17-0ubuntu0.18.10.1 [1,388 kB]
```

Get:8 http://in.archive.ubuntu.com/ubuntu cosmic/main amd64 libapache2-mod-php all 1:7.2+62 [2,956 B]
Get:9 http://in.archive.ubuntu.com/ubuntu cosmic-updates/main amd64 php7.2 all 7.2.17-0ubuntu0.18.10.1 [9,244 B]
Get:10 http://in.archive.ubuntu.com/ubuntu cosmic/main amd64 php all 1:7.2+62 [2,828 B]
Get:11 http://in.archive.ubuntu.com/ubuntu cosmic-updates/main amd64 php7.2-mysql amd64 7.2.17-0ubuntu0.18.10.1 [119 kB]
Get:12 http://in.archive.ubuntu.com/ubuntu cosmic/main amd64 php-mysql all 1:7.2+62 [2,000 B]
Fetched 4,067 kB in 30s (136 kB/s)
Selecting previously unselected package php-common.
(Reading database ... 158668 files and directories currently installed.)
Preparing to unpack .../00-php-common_1%3a62_all.deb ...
Unpacking php-common (1:62) ...
Selecting previously unselected package php7.2-common.
Preparing to unpack .../01-php7.2-common_7.2.17-0ubuntu0.18.10.1_amd64.deb ...
Unpacking php7.2-common (7.2.17-0ubuntu0.18.10.1) ...
Selecting previously unselected package php7.2-json.
Preparing to unpack .../02-php7.2-json_7.2.17-0ubuntu0.18.10.1_amd64.deb ...
Unpacking php7.2-json (7.2.17-0ubuntu0.18.10.1) ...
Selecting previously unselected package php7.2-opcache.
Preparing to unpack .../03-php7.2-opcache_7.2.17-0ubuntu0.18.10.1_amd64.deb ...
Unpacking php7.2-opcache (7.2.17-0ubuntu0.18.10.1) ...
Selecting previously unselected package php7.2-readline.
Preparing to unpack .../04-php7.2-readline_7.2.17-0ubuntu0.18.10.1_amd64.deb ...
Unpacking php7.2-readline (7.2.17-0ubuntu0.18.10.1) ...
Selecting previously unselected package php7.2-cli.
Preparing to unpack .../05-php7.2-cli_7.2.17-0ubuntu0.18.10.1_amd64.deb ...
Unpacking php7.2-cli (7.2.17-0ubuntu0.18.10.1) ...
Selecting previously unselected package libapache2-mod-php7.2.
Preparing to unpack .../06-libapache2-mod-php7.2_7.2.17-0ubuntu0.18.10.1_amd64.deb ...
Unpacking libapache2-mod-php7.2 (7.2.17-0ubuntu0.18.10.1) ...
Selecting previously unselected package libapache2-mod-php.
Preparing to unpack .../07-libapache2-mod-php_1%3a7.2+62_all.deb ...
Unpacking libapache2-mod-php (1:7.2+62) ...
Selecting previously unselected package php7.2.
Preparing to unpack .../08-php7.2_7.2.17-0ubuntu0.18.10.1_all.deb ...
Unpacking php7.2 (7.2.17-0ubuntu0.18.10.1) ...
Selecting previously unselected package php.
Preparing to unpack .../09-php_1%3a7.2+62_all.deb ...
Unpacking php (1:7.2+62) ...
Selecting previously unselected package php7.2-mysql.
Preparing to unpack .../10-php7.2-mysql_7.2.17-0ubuntu0.18.10.1_amd64.deb ...
Unpacking php7.2-mysql (7.2.17-0ubuntu0.18.10.1) ...
Selecting previously unselected package php-mysql.
Preparing to unpack .../11-php-mysql_1%3a7.2+62_all.deb ...
Unpacking php-mysql (1:7.2+62) ...
Setting up php-common (1:62) ...
Created symlink /etc/systemd/system/timers.target.wants/phpsessionclean.timer → /lib/systemd/system/phpsessionclean.timer.
Processing triggers for man-db (2.8.4-2) ...

Setting up php7.2-common (7.2.17-0ubuntu0.18.10.1) ...

Creating config file /etc/php/7.2/mods-available/calendar.ini with new version

Creating config file /etc/php/7.2/mods-available/ctype.ini with new version

Creating config file /etc/php/7.2/mods-available/exif.ini with new version

Creating config file /etc/php/7.2/mods-available/fileinfo.ini with new version

Creating config file /etc/php/7.2/mods-available/ftp.ini with new version

Creating config file /etc/php/7.2/mods-available/gettext.ini with new version

Creating config file /etc/php/7.2/mods-available/iconv.ini with new version

Creating config file /etc/php/7.2/mods-available/pdo.ini with new version

Creating config file /etc/php/7.2/mods-available/phar.ini with new version

Creating config file /etc/php/7.2/mods-available/posix.ini with new version

Creating config file /etc/php/7.2/mods-available/shmop.ini with new version

Creating config file /etc/php/7.2/mods-available/sockets.ini with new version

Creating config file /etc/php/7.2/mods-available/sysvmsg.ini with new version

Creating config file /etc/php/7.2/mods-available/sysvsem.ini with new version

Creating config file /etc/php/7.2/mods-available/sysvshm.ini with new version

Creating config file /etc/php/7.2/mods-available/tokenizer.ini with new version
Setting up php7.2-readline (7.2.17-0ubuntu0.18.10.1) ...

Creating config file /etc/php/7.2/mods-available/readline.ini with new version
Setting up php7.2-json (7.2.17-0ubuntu0.18.10.1) ...

Creating config file /etc/php/7.2/mods-available/json.ini with new version
Setting up php7.2-opcache (7.2.17-0ubuntu0.18.10.1) ...

Creating config file /etc/php/7.2/mods-available/opcache.ini with new version
Setting up php7.2-mysql (7.2.17-0ubuntu0.18.10.1) ...

Creating config file /etc/php/7.2/mods-available/mysqlnd.ini with new version

Creating config file /etc/php/7.2/mods-available/mysqli.ini with new version

Creating config file /etc/php/7.2/mods-available/pdo_mysql.ini with new version
Setting up php7.2-cli (7.2.17-0ubuntu0.18.10.1) ...

update-alternatives: using /usr/bin/php7.2 to provide /usr/bin/php (php) in auto mode
update-alternatives: using /usr/bin/phar7.2 to provide /usr/bin/phar (phar) in auto mode
update-alternatives: using /usr/bin/phar.phar7.2 to provide /usr/bin/phar.phar (phar.phar) in auto mode

Creating config file /etc/php/7.2/cli/php.ini with new version
Setting up libapache2-mod-php7.2 (7.2.17-0ubuntu0.18.10.1) ...

Creating config file /etc/php/7.2/apache2/php.ini with new version
Module mpm_event disabled.
Enabling module mpm_prefork.
apache2_switch_mpm Switch to prefork
apache2_invoke: Enable module php7.2
Setting up php-mysql (1:7.2+62) ...
Setting up libapache2-mod-php (1:7.2+62) ...
Setting up php7.2 (7.2.17-0ubuntu0.18.10.1) ...
Setting up php (1:7.2+62) ...

RESULT

Thus in this task I have familiarized with various commands that are used to setup a webpage using Lamp stack. The task is completed successfully and results are uploaded according to the instructions.