**ASSIGNMENT NO 1**

SUBMITTED TO, SUBMITTED BY,

Prof. Rahulnath H A Ananya.B

TKMCE MCA108

TKMCE

**TITLE: CYCLE 1**

**SUBJECT: 20MCA133 - Web Programming Lab**

**1.Install any linux distro of your choice**

* A Linux distribution often abbreviated as distro an operationg system made from a software collection that is based upon the Linux kernel and often a package management system.
* The versions/types or kinds of linux os are called Linux distributions or distro.
* There are hundreds of Linux OS or distributers these days.
* Eg: Arch, Cent Os, Debian, fedora,Ubuntu etc..
* I choose UBUNTU to install in my PC.

**Steps in installing Ubuntu**

* For this i decided to install linux using Virtual Machine
* Because virtual machine offers the freedom of running linux on an existing os already installed in our computer.
* Therefore virtual machine software like oracle virtual machine can install ubuntu in easy way

**Steps followed by me are as follows:**

I

1. I downloaded Virtual Box 6.1.16 oracle VM Virtual Box

1.1 Then i selected appropriate package depending upon my processor

1.2 Then click on the windows host for selecting packages

1.3 After the download is completed , Open the setup file

1.4 Click on next on Setup dialog box

1.5 Select the directory to install virtual box and click next

1.6 Next click yes on the dialog box appeared showing a warning message and then click on install

1.7 Now installation will start, once it completed click on Finish button to start virtual box

1.8 Now virtual box dashboard will appear

II DOWNLOADING UBUNTU

1. Then downloaded ubuntu
2. Once the download is completed,next i created a machine in virtual box
   1. Open virtual box and click on the new button
   2. Then another window appears,there give the name of our os which i installed in virtual box,then selected os like linux and its version then click next
   3. Next allocated the Ram size to my virtual os. I give 1024 mb ram to run ubuntu,then click on next
   4. Then to run os in virtual box, i have to create virtual hard disk so clicked on create a virtual hard drive now and click on create button
   5. Then selected VHD(virtual hard disk) option and click on next
   6. Then on another dialog box appeared, click on dynamically allocated and click on next.
   7. Next selected file location and size ie;allocate memory to our virtual hard drive then click on create button
   8. Now i can see the machine name in left panel

III INSTALLING UBUNTU

1. Install ubuntu on the Machine
2. Then selected the machine and click on the settings
3. Then selected the folder option and selected the ubuntu iso file, then clicked on start button
4. Then check the settings to make sure about the general, display and storage specifications are correct
5. At that time I found an issue like invalid settings detected

5.1Then i tried to study about the issue and after sometime i found that the issue is based up on vt or virtualization technology

5.2 VT is in disabled state for my system, ie the issue

5.3 Then i fix it by go to bios settings and enabled the VT

6.After that i click on the new button in VM , then start button

7. Then a “Welcome” screen appears and click on the “install ubuntu” button

8.Then click on “continue” button on another window

9.Then selected ‘Erase disk and install ubuntu’

10.Then pressed on Install now

11 Next given correct options to the questions ask by ubuntu

10.Now i make my own account in it using preferred name, username and password

10.Then clicked on ‘continue’ button

11.Then installation is finished after sometime

12.After that ,clicked on the Restart Now button

13.Then a screen appeared saying that Please remove installation media and then press Enter,then i do it

14.Then enter the password and press Enter

15.Now i can see the ubuntu desktop,but it is so small

15.1 So i learnt and tried to resolve it,using VirtualBox Guest Additions

15.2 For that i clicked on Devices and then Insert Guest Additions CD images

15.3 Then clicked Run to start installing it

15.4 Then i give the password for root privilege

15.5 After installation is finished, i pressed Return key for closing the terminal

16. Then restarted the system to enabling the changes,then click on Shutdown on the menu and then click the Restart button

17.After all these steps i successfully completed the installation of ubuntu using Virtual box.

**2. Linux Commands**

* Terminals which just send keystrokes to the server and displaying data they received on the screen.
* The commands were very useful to reduce the number of keystrokes needed, speeding up people’s use of the terminal even more.

**TERMINAL COMMANDS**

**1.pwd** : An abbreviation of ‘’**p**rint **w**orking **d**irectory’’. All it does is print out the shell’s current working directory.  The shell has a notion of a default location in which any file operations will take place. This is its working directory.

**2.cd command** : An abbreviation for ‘’**c**hange **d**irectory’’. We can change the working directory using the “cd” command.

SYNTAX: cd /

pwd

* Important thing is that the directory separator is a forward slash ("/"), not the backslash that you may be used to from Windows or DOS systems. Although the “/” directory is sometimes referred to as the **root** directory, the word “root” has another meaning. **Root** is also the name that has been used for the superuser since the early days of Unix.
* To go up to the parent directory, in this case back to “/”, use the special syntax of two dots (**. .**) when changing directory and note that the space between **cd** and **. .** , unlike in DOS you can’t just type **cd.** . as one command.
* “cd “on its own is a quick shortcut to get back to your home directory.
* We can also use “. .” more than once if you have to move up through multiple levels of parent directories.

**3.whoami command** : This command will remind you of your username if you are not sure about it.In this command you’ll need to replace USERNAME with your own username,then we can find the username.

**4.mkdir command** : “mkdir” is short for ‘make directory’.  we create the temporarydirectory inside */*temp. Without the forward slash at the start the mkdir command would try to find a temp directory inside the current working directory, then try to create a temporary directory inside that.

* We can create a few subdirectories by the following syntax:

mkdir dir1 dir2 dir3

* This time we’ve added three things after the mkdir command.
* Those things are referred to as parameters or arguments, and different commands can accept different numbers of arguments.
* The mkdir command expects at least one argument, whereas the cd command can work with zero or one, but no more.

**5.ls (list) command**: The lscommand is a command-line utility for listing the contents of a directory or directories given to it via standard input. The lscommand supports showing a variety of information about files, sorting on a range of options and recursive listing.

**6.-p :** It is called an option or a switch. Options are used to modify the way in which a command operates, allowing a single command to behave in a variety of different ways.

**7.cat command:** The cat short for “concatenate“ command is one of the most frequently used command in Linux/Unix like operating systems. “cat” command allows us to create single or multiple files, view contain of file, concatenate files and redirect output in terminal or files. All we need to do is to add the greater-than character (">") to the end of our command line, followed by the name of the file to write to. If you pass more than one filename to cat it will output each of them, one after the other, as a single block of text.

**8.echo command:** “echo” just prints its arguments back out again. **“**echo” command inlinux is used to display line of text/string that are passed as an argument . This is a built in command that is mostly used in shell scripts and batch files to output status text to the screen or a file.

**9. cat t\* command**: “cat t\* ” command means that concatenate all the files whose names start with a *t* and are followed by zero or more other characters.

**10.more and less command:** The more command helps you navigate outputs from commands in a user-friendly way. The more command also allows the user do scroll up and down through the page.

less is a command that displays file contents or command output one page at a time in your terminal. less is most useful for viewing the content of large files or the results of commands that produce many lines of output. The content displayed by less can be navigated by entering keyboard shortcuts.

**11.mv (move) command:** eg: mv combined.txt dir1, putting our combined*.*txt file into our dir1 directory, using the mv (move) command. Which is similar to the “cp” command, except that with “mv” the file is physically moved from one place to another, instead of being duplicated, as with cp.

**12.cp command:** cp is the command used in Unix and Linux to copy your files or directories. Copies any file with the extension “. txt” to the directory “newdir” if the files do not already exist, or are newer than the files currently in the directory. This command is used to copy files or group of files or directory.

**13. rm (remove) command:** rm command is used to remove objects such as files, directories, symbolic links and so on from the file. To be more precise, rm removes references to objects from the filesystem, where those objects might have had multiple references for example, a file with two different names.

**14. rmdir (remove directory):** rmdir will only delete empty folders. The rmdir command removes each and every directory specified in the command line only if these directories are empty. So if the specified directory has some directories or files in it then this cannot be removed by rmdir command.

**15. wc (word count):** As the name implies, it is mainly used for counting purpose. It is used to find out number of lines, word count, byte and characters count in the files specified in the file arguments. Its options are -l , --lines - Print the number of lines.-w , --words - Print the number of words.-m , --chars - Print the number of characters.-c , --bytes - Print the number of bytes.-L , --max-line-length - Print the length of the longest line.

**16.man command:** This command is used to display the user manual of any command that we can run on the terminal. It provides a detailed view of the command which includes NAME, SYNOPSIS, DESCRIPTION, OPTIONS, EXIT STATUS, RETURN VALUES, ERRORS, FILES, VERSIONS, EXAMPLES, AUTHORS and SEE ALSO.

**17.sort command:** sort command sorts the contents of a text file, line by line. sort is a standard command line program that prints the lines of its input or concatenation of all files listed in its argument list in sorted order. If we were to sort the contents of the file alphabetically, that would do the trick. Unix offers a sort command to do exactly that.

**18. su command.** Short for ‘**s**uper**u**ser’ or ‘**s**witch **u**ser’, and allows you to change to another user on the machine without having to log out and in again.

**19.logout command**: Use the logout command (or **Ctrl-D** shortcut) to return to their user-level account.

**20.reset command:** “reset command” is used to clear the terminal by running the reset command. Used to initialize the terminal. This is useful once a program dies leaving a terminal in an abnormal state.

**BASH** **SCRIPTS**

1. **/bin/sh :**Thistells Unix that the file is to be executed by /bin/sh. This is the standard location of a Bourne-compatible shell on just about every Unix/Linux system.

1. #! : This is a special directive which Unix treats specially. It means that even if we are using csh, ksh, or anything else as our interactive shell, that what follows should be interpreted by the Bourne shell.
2. **echo command** :  Used in scripting language and batch files to display a line of text/string on standard output or a file.echo will automatically put a single space between its parameters. This is a built in command that is mostly used in shell scripts.
3. **chmod command :** chmod is the command and system call which is used to change the access permissions of file system objects (files and directories). It is also used to change special mode flags.The name is an abbreviation of change mode.
4. **VAR=value :** Variables are the symbolic name for a chunk of memory to which we can assign values, read and manipulate its contents. VAR=value, the shell sees the "=" symbol and treats the command as a variable assignment.
5. **expr command:** This is external program command,which only expects numbers.
6. **read command**: read is a bash builtin command that reads the contents of a line into a variable. which reads a line from standard input into the variable supplied. It is primarily used for catching user input but can be used to implement functions taking input from standard input.
7. **export command**: export command has a fundamental effect on the scope of variables. In order to really know what's going on with our variables, we will need to understand something about how this is used.
8. **. (dot) command :** In order to receive environment changes back from the script, we must source the script - this effectively runs the script within our own interactive shell, instead of spawning another shell to run it.For this we can use the .(dot) command.
9. **For Loop:** A for loop is classified as an iteration statement i.e. it is the repetition of a process within a bash script. for simply loops through whatever input it is given, until it runs out of input. For example, you can run a command or task 5 times or read and process list of files using a for loop. for loops iterate through a set of values until the list is exhausted
10. **While Loops:** The bash while loop is a control flow statement that allows code or commands to be executed repeatedly based on a given condition. Here the echo and read statements will run indefinitely until the condition occurs.
11. **(:) colon :** (:) is a shell builtin that is basically equivalent to the true command. The colon (:) always evaluates to true; whilst using this can be necessary sometimes, it is often preferable to use a real exit condition.
12. **Escape characters:** Escape characters are used to remove the special meaning from a single character. A non-quoted backslash, \, is used as an escape character in Bash. for example, that the use of double quotes (") characters affect how spaces and TAB characters are treated.
13. **Test command**: test is a builtin command of the Bash shell that can test file attributes, and perform string and arithmetic comparisons. test is not often called directly. test is more frequently called as “[“ . “[“ is a symbolic link to test, just to make shell programs more readable.
14. **If .. Then .. Else:** If the TEST-COMMAND evaluates to True , the STATEMENTS1 will be executed. We can have only one else clause in the statement.
15. **Case statement:** Bash case statement is the simplest form of the bash if-then-else statement. Syntax of bash case statement. ... When a match is found all of the associated statements until the double semicolon (;;) are executed. After the first match, case terminates with the exit status of the last command that was executed**.**
16. **Internal Field Seperator (IFS**): The Internal Field Separator (IFS) that is used for word splitting after expansion and to split lines into words with the read builtin command. The default value is <space><tab><newline>. The IFS is a special shell variable. You can change the value of IFS as per your requirments.
17. **Backtick ( ` ) :** The backtick (`) is also often associated with external commands. The backtick is used to indicate that the enclosed text is to be executed as a command. There is another syntax for the same purpose: Instead of using backticks, we can use $(..). The advantage of this is that it can be nested, which the backticks cannot.
18. **Function:** It's a small chunk of code which you may call multiple times within your script. In Bash, defining a function is as easy as setting it either in the script file we're writing or in a separate file.
19. **Logical AND:** The (&&) would execute the second command only, if the execution of first command SUCCEEDS, i.e., the exit status of the first command is 0. This command is very useful in checking the execution status of last command**.**
20. **Logical OR:** (||) This is a binary operator, which returns true is either of the operand is true or both the operands are true and returns false if none of them is false.
21. **Logical NOT:** (!) is boolean operator, which is used to test whether expression is true or not. For example, if file not exists, then display an error on screen.
22. **GIT AND GITHUB**
23. **Install GIT**

Git provides hosting for software development and version control. It offers the distributed version control and source code management functionality of Git, plus its own features**.**

**STEPS I FOLLOWED WHILE INSTALLING GIT**

1. I already have an account in GIT website, first of all i signed in and then learned how to install git

2. I downloaded GIT from git.scm.com

3. From there i clicked on Download for Windows 64 bit

4.When prompted, clicked on Run button

5.Then i selected destination location and click next

6.Then i selected the components like On the desktop and click next

7.Then adjusted the path and configure the line ending conversion then clicked next

8.After all configurations clicked on install button

9.After installation completed, Git launched.

**B. Setup your GIT account and create 3 repositories, one for each lab course you currently have, namely Web Programming, Data Structures, Programming Lab(Python).**

**STEPS**

1. First of all, I go to Git website

2. Then signed in to my account

3. Then i clicked on the new repository button from the File in the top-left side of the screen

5.Then a dialog box appears and there i give the name ,description of the new repository ,and there is an option to initialize the repository with a README

6.Like this i created 3 repositories with names DATA\_ATRUCTURE, PROGRAMMING\_LAB\_PYTHON, WEB\_PROGRAMMING.

1. **Clone these repositories to the local machine using GIT Commands.**
2. For Cloning an Existing Repository, i use the command “git clone”
3. If i want to get a copy of an existing Git repository — for example, a project you’d like to contribute to — the command we need is git clone.
4. By using “ git clone <url> “ command ie, url of the repository we have to clone,I cloned that.
5. Then i cloned those 3 repositories for each lab

**4.Install PHP, MySQL, Python3 and NodeJS And Configure environment variables if required.**

**1.PHP**

* I downloaded PHP from the site [www.php.net](http://www.php.net)
* From this site i downloaded the version 7.4.12
* I downloaded the thread safe php
* After downloading i copied it and the extract it
* Then saved in a folder
* Then copy the path and then set the environment variable
* In system variable, go to path ,then click edit and take a blank space and pasted the path
* After setting it, for checking installation ,I go to command prompt and run it administrator
* Then typed php -v for seeing the version
* After that I can see the Version 7.4.12.

**2.MySQL**

* I downloaded MySQL from the site [www.mysql.com](http://www.mysql.com)
* From the downloads section,scrolled down and clicked on the MySQL Community GPL Downloads
* Then click on MySQL installer for windows
* From the next window i selected the first option ie,windows (x86,32 bit),MSI Installer for download
* Then clicked on No thanks,just start the download, then installer starts downloading
* After installer downloading was completed, i run it
* Then there is several options like selecting the setup type and i choose custom type
* Then selected the available products like MySQL server 8.0 and then MySQL server 8.0.21
* Then i go to Applications and selected MySQL workbench,then MySQL workbench 8.0 and then MySQl workbench 8.0.21
* Then selected the MySQL shell and then clicked on MySQL shell 8.0 and then selected the MySQL shell 8.0.21
* Then click on execute
* Then selected Standalone MySQL server
* Then set the server configuration by default
* Then created the Root Account Password
* Then Add a user and click on next
* Then go to the windows service and configured the MySQL server as a windows service,then apply the configuration
* Then clicked the Finish button
* The installation is completed and then I connect to MySQL server using the username and password
* Also checked the new connection by clicking MySQL connections and through the setup new connection dialog box.
* Then connect MySQL using command prompt
* Then copied the URL of mysql server from program files
* Then typed the command mysql -u root -p
* Then entered the password

**3.Python3**

1. I downloaded Python from the site [www.python.org](http://www.python.org)
2. From the download section I downloaded 3.9.0 version
3. After the file downloaded, i extracted that file
4. Then i opened the file and clicked the customize installation
5. Then i checked the Add Python 3.9 to path option
6. Then selected the optional features by default
7. Next i customize install location
8. Then click on Install button
9. Then clicked on yes and installation started
10. After that a window showing “Setup was successful message “ was appeard,
11. Then i clicked on close button

**4.NodeJS**

* I downloadedNodeJS from the website [www.nodejs.org](http://www.nodejs.org)
* From there i downloaded NodeJS 64 bit version
* Then I run the NodeJS installer and acceped the Liscense Agreement and clicked Next
* Then i kept all the default settings as it is
* Then clicked on install button
* Once the installation is installed,clicked the Finish button
* Then i opened the command prompt, and checked the installation was successful
* For that i typed the command node -v to see the version

**5. Install a programming IDE for your lab courses**. **Install Visual Studio code**

**Steps in installing Visual Studio code**

1. I downloaded Vistual Studio code from the site code.visualstudio.com
2. After downloading it, i clicked on the run button
3. Then I accepted the agreement and clicked Next button
4. Then set the destination location and clicked next button
5. Then created the start menu folder and click Next button
6. Then i selected Additional Tools options and clicked next button
7. Then checked Launch Visual Studio Code and clicked the Install button
8. Then click on Finish button

**6.URL to the Web Programming Lab Repository that you created.**

* <https://github.com/Ananya31-tkm/WEB_PROGRAMMING.git>

**REFERENCES**

* <https://ubuntu.com/tutorials/command-line-for-beginners#1-overview>
* <https://www.shellscript.sh/>
* <https://guides.github.com/>
* <https://git-scm.com/>