**LAB REPORT NO 1**



Submitted by:  **Muhammad Ali**

Registration No:- **19PWCSE1801**

Class Section: A

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Submitted to:

**Engr.Mian Ibad Ali shah**

Data:(25,04,2021)

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

**Introduction to Unix: -**

UNIX is an operating system which was first developed in the 1960s, and has been under constant development ever since. By operating system, we mean the suite of programs which make the computer work. It is a stable, multi-user, multi-tasking system for servers, desktops and laptops.

UNIX systems also have a graphical user interface (GUI) similar to Microsoft Windows which provides an easy to use environment. However, knowledge of UNIX is required for operations which aren't covered by a graphical program, or for when there is no windows interface available, for example, in a telnet session.

**The kernel: -**

The kernel of UNIX is the hub of the operating system: it allocates time and memory to programs and handles the filestore and communications in response to system calls.

As an illustration of the way that the shell and the kernel work together, suppose a user types rm myfile (which has the effect of removing the file myfile). The shell searches the filestore for the file containing the program rm, and then requests the kernel, through system calls, to execute the program rm on myfile. When the process rm myfile has finished running, the shell then returns the UNIX prompt % to the user, indicating that it is waiting for further commands.

**The shell: -**

The shell acts as an interface between the user and the kernel. When a user logs in, the login program checks the username and password, and then starts another program called the shell. The shell is a command line interpreter (CLI). It interprets the commands the user types in and arranges for them to be carried out. The commands are themselves programs: when they terminate, the shell gives the user another prompt (% on our systems).

The adept user can customise his/her own shell, and users can use different shells on the same machine. Staff and students in the school have the tcsh shell by default.

The tcsh shell has certain features to help the user inputting commands.

Filename Completion - By typing part of the name of a command, filename or directory and pressing the key, the tcsh shell will complete the rest of the name automatically. If the shell finds more than one name beginning with those letters you have typed, it will beep, prompting you to type a few more letters before pressing the tab key again.

History - The shell keeps a list of the commands you have typed in. If you need to repeat a command, use the cursor keys to scroll up and down the list or type history for a list of previous commands.

**Basic command we understand in the lab: -**

**1). Is command: -**

Lists the names of files in a particular Unix directory. If you type the ls command with no parameters or qualifiers, the command displays the files listed in your current working directory. When you give the ls command, you can add one or more modifiers to get additional information.

**Example: ls**  
Result: Lists the names of files in your default directory, in alphabetical order.

**Example: ls -l**  
Result: Gives a "long listing" of the files in your directory. In addition to the file name, the long listing shows protection information, file owner, number of characters in file, and the date and time of the last change to the file.

**Example: ls -a**  
Result: Causes all your files to be listed, including those files that begin with a period (i.e., hidden files).

**2). Pwd command: -**

In **Unix**-like and some other operating systems, the **pwd command** (print working directory) writes the full pathname of the current working directory to the standard output.

**Example**: pwd

**Result:** home/desktop/os$

**3). whoami or uname -n command: -**

These command print out user name of the pc or user name of the os.

**Example:** whoami

Result: muhammad

**4). cd and cd.. :-**

cd command show current directory means current file in which user work. And cd.. command access user to previous directory.

**Example:** cd

Result: /home/muhammad$

**Example:** cd..

Result: /home/desktop$

**5).** **mkdir *name*** :-

Creates a directory in the spot specified by ***name***. If you wish to make a directory called ***my\_dir*** in your current directory then the command would be **mkdir *my\_dir***.

**rmdir *name*** :-

Removes a directory in the spot specified by name.

**Example:mkdir os**

**Example:rmdir os**

**2). Rm command: -**

Deletes specific files. You can enter more than one file specification on a command line by separating the file specifications with spaces.

Example: **rm newfile**  
Result: Deletes the file named "newfile."

Example: **rm newfile oldfile**  
Result: Deletes two files–"newfile" and "oldfile."

**3). cp command: -**

Makes copies of your files. You can use it to make copies of files in your default directory, to copy files from one directory to another directory, or to copy files from other devices.

Example: **cp fileone filetwo**  
Result: Copies the contents of fileone to a file named filetwo. Two separate files now exist.

**4). Chmod command:-**

Chmod command is the [command](https://en.wikipedia.org/wiki/Command_(computing)) and [system call](https://en.wikipedia.org/wiki/System_call) used to change the [access permissions](https://en.wikipedia.org/wiki/File-system_permissions) of [file system](https://en.wikipedia.org/wiki/File_system) objects ([files](https://en.wikipedia.org/wiki/Computer_file) and [directories](https://en.wikipedia.org/wiki/Directory_(computing))) sometimes known as modes. It is also used to change [special mode](https://en.wikipedia.org/wiki/Chmod#Special_modes) flags such as *[setuid](https://en.wikipedia.org/wiki/Setuid" \o "Setuid)*[and](https://en.wikipedia.org/wiki/Setuid" \o "Setuid)*[setgid](https://en.wikipedia.org/wiki/Setuid" \o "Setuid)* flags and a ['sticky' bit](https://en.wikipedia.org/wiki/Sticky_bit).. The request is filtered by the [umask](https://en.wikipedia.org/wiki/Umask" \o "Umask). The name is an abbreviation of *change mode*.

For example: drwxrwx---

The characters to the right of the "d" define permissions for each [*class*](https://en.wikipedia.org/wiki/File-system_permissions#Classes):

* the three leftmost characters, ,rwx define permissions for the *User* class (i.e. the file owner).
* the middle three characters,rwx, define permissions for the *Group* class (i.e. the [group](https://en.wikipedia.org/wiki/Group_identifier) owning the file)
* the last three characters,---, define permissions for the *Others* class. In this example, users who are not the owner of the file and who are not members of the *Group* (and, thus, are in the *Others* class) have no permission to access the file.

**5). history command: -**

The command is simply called history, but can also be accessed by looking at your . bash\_history in your home folder. By default, the history command will show you the last five hundred commands you have entered.

Example: history 5

Result :ls,pwd,mkdir.rmdir,chmod;

**6). Which date command: -**

Date command is used to display the system date and time. date command is also used to set date and time of the system. By default the date command displays the date in the time zone on which unix/linux operating system is configured. You must be the super-user (root) to change the date and time

**7). alias command: -**

alias command instructs the shell to replace one string with another string while executing the command(slike a nick name). When we often have to use a single big command multiple times, in those cases, we create something called as alias for that command.

**8). wc Command:-**

In sum, the wc command counts words and provides you with a summary of what is found. Unix will report to you how many words are in a particular file. The wc stands for word count. When you want to see how many words are typed in a file, you simply need to execute the command and then the filename with its absolute or relative path. You will be shown the word count as well as the number of bytes, words, and lines in files.

The wc command is an excellent way to provide current statistics on a file you may want to work with. For example, if you wanted to issue the tail command on a file you think may be very large, you can use the wc command on a file to quickly see how many lines are in it. You can then use the tail command to jump ...