**Assignment on**



**What is linux ?**

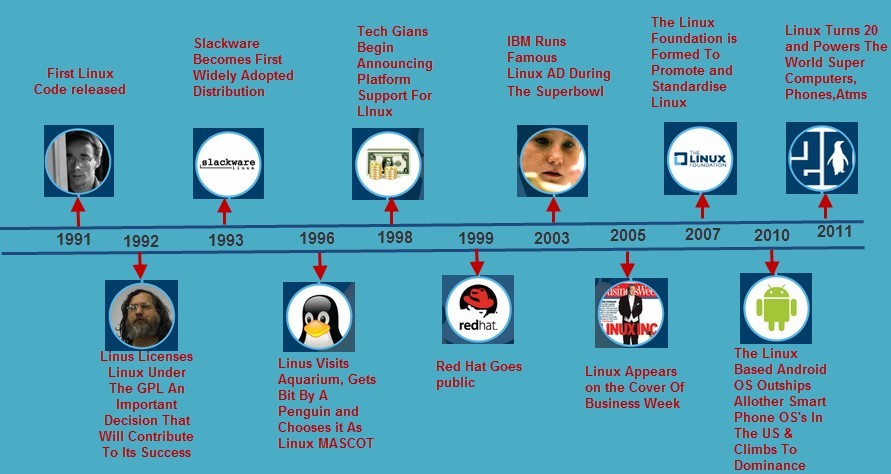
* It is a Unix like Operating System.
* It can run on 32-bit and 64 bit Software.
* Linux Os ,is freely distributable, and a cross- platform.
* Supports Multiple Processors.
* True Multitasking and Multi-User Os.

**History of Linux**

* In 1991, **Linus Torvalds** a student at the university of **Helsinki, Finland**, thought to have a freely available academic version of Unix started writing its own code.
* Later this project became the Linux kernel.
* He wrote this program specially for his own PC as he wanted to use Unix 386 Intel computer but couldn't afford it. He did it on MINIX using GNU C compiler. GNU C compiler is still the main

choice to compile Linux code but other compilers are also used like Intel C compiler.

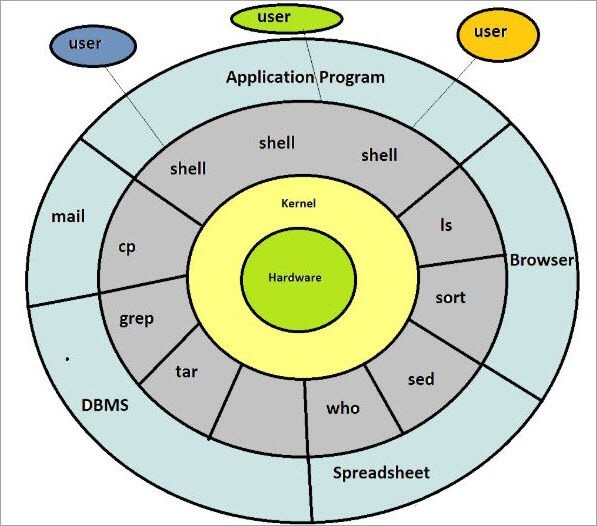
* He started it just for fun but ended up with such a large project. Firstly he wanted to name it as 'Freax' but later it became 'Linux'.
* Linux uses most of its tools from GNU software and are under GNU copyright. In 1992, he released the kernel under GNU General Public License.



**Language used:**

* + The core components of Linux like Drivers, Kernels are written in **C language**.
  + The GNU parts of the Linux OS are also written in **ANSI C**, including the compiler itself (gcc) and glibc.

**Architecture:**



Linux Architecture:-

**Kernel:** Kernel is the core of the Linux based operating system. It virtualizes the common hardware resources of the computer to provide each process with its virtual resources. This makes the process seem as it is the sole process running on the machine. The kernel is also responsible for preventing and mitigating conflicts between different processes.

* Monolithic Kernel
* Hybrid kernels
* Exo kernels
* Micro kernels

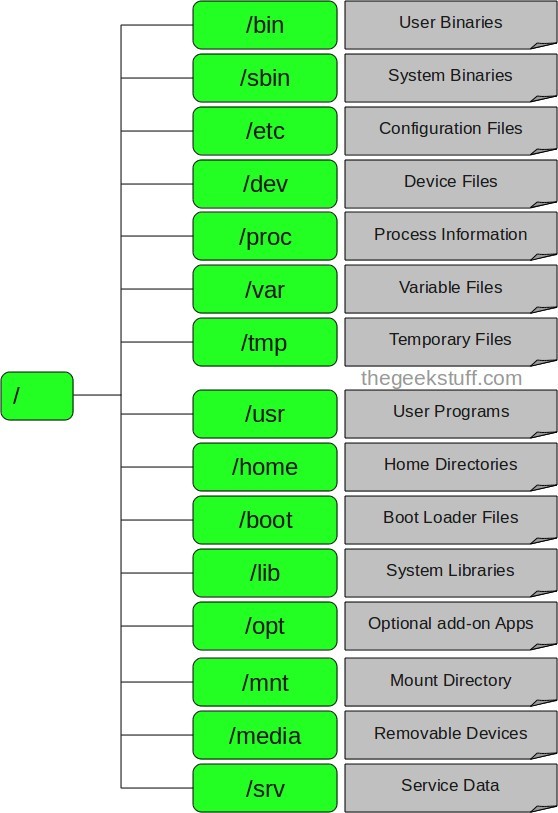
**System Library:** It is the special types of functions that are used to implement the functionality of the operating system.

**Shell:** It is an interface to the kernel which hides the complexity of the kernel’s functions from the users. It takes commands from the user and executes the kernel’s functions.

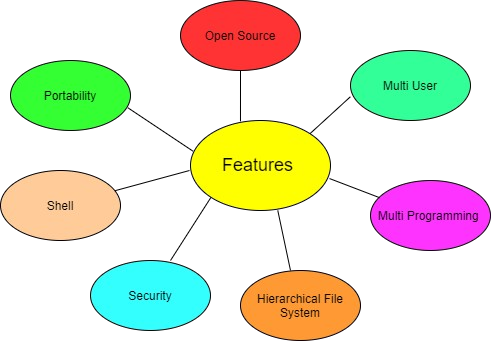
**Hardware Layer:** This layer consists all peripheral devices like RAM/ HDD/ CPU etc.

**System Utility:** It provides the functionalities of an operating system to the user.

Linux File Structure



Root



Features of Linux OS:-

Why linux is called secure ?

Linux Security provides core security capabilities for Linux environments: multi-engine anti- malware with vital Integrity Checking for endpoints and servers.

* + Provides protection against unauthorized access within the corporate network
  + Can protect your mixed environment against both Windows and Linux malware
  + Scans malware on mail servers, web servers and file servers, as well as endpoints
  + Protects system files against unauthorized modifications
  + Extensive monitoring and alerting functions to notify administrators about infected content
  + Easy deployment and management with Policy Manager, even for multiple servers
  + Superior detection rates with multiple scanning engines

**Basic commands -**

**$** show the terminal.

In linux commands commands follow a general syntax.

## $<command name> [option]

1. **pwd command**

Use the pwd command to find out the path of the current working directory (folder) you’re in. The command will return an absolute (full) path, which is basically a path of all the directories that starts with a forward slash (/). An example of an absolute path is /home/username.

## syntax - $pwd [options]

**Options**

* + **-L (logical)** - Use PWD from environment, even if it contains symbolic links
  + **-P (physical)** - Avoid all symbolic links
  + **-help** - Display this help and exit
  + **-version** - Output version information and exit

1. **cd command**

To navigate through the Linux files and directories, use the **cd** command. It requires either the full path or the name of the directory, depending on the current working directory that you’re in.

Another scenario is if you want to switch to a completely new directory, for example,**/home/username/Movies**. In this case, you have to type **cd** followed by the directory’s absolute path: **cd /home/username/Movies**.

**syntax - $ cd [option]**

## Options

* **cd ..** (with two dots) to move one directory up
* **cd or cd ~** to go straight to the home folder
* **cd-** (with a hyphen) to move to your previous directory

1. **ls command**

The **ls** command is used to view the contents of a directory. By default, this command will display the contents of your current working directory.

If you want to see the content of other directories, type **ls** and then the directory’s path. For example, enter

**ls /home/username/Documents** to view the content of **Documents**.

# Syntax - $ls [option]

Option-

* **ls -R** will list all the files in the sub-directories as well
* **ls -a** will show the hidden files
* **ls -al** will list the files and directories with detailed information like the permissions, size, owner, etc.

## cat command

**cat** (short for concatenate) is one of the most frequently used commands in Linux. It is used to list the contents of a file on the standard output (sdout). To run this command, type **cat** followed by the file’s name and its extension. For instance: **cat file.txt**.

Syntax - $cat [option]<filename> Here are other ways to use the **cat** command:

* **$cat > filename -** creates a new file
* **$cat filename1 filename2>filename3 -** joins two files (1 and 2) and stores the output of them in a new file (3)
* **$cat filename | tr a-z A-Z >output.txt -** to convert a file to upper or lower case use.
* **$cat filename** - display single file
* **$cat file1 file 2** - display multiple files
* **$cat -n filename** - line number in file output
* **$cat -s filename -** suppress repeated lines
* **$cat file1 >> file2 -** append the contents of one file to the end of another file
  + **$tac filename** - display content in reverse order using tac command.
  + **$cat -E "filename"** -can highlight the end of line.
  + **$cat -- "-dashfile"** - open dashed files.
  + **$cat "filename" | more** - show that much content, which could fit in terminal and will ask to show more.
  + **$cat \*.txt** - display the content of all text files in the folder.

1. **cp command :** It is used to copy files from the current directory to a different directory.**cp** stands for copy. It creates an exact image of a file on a disk with different file name. **cp command** require at least two filenames in its arguments.

## Syntax - $ cp [options] source dest Options

* $**cp -a** -archive files
* **$cp -f** -force copy by removing the destination file if needed
* **$cp -i** -interactive - ask before overwrite
* **$cp -l -**link files instead of copy
* **$cp -L** -follow symbolic links
* **$cp -n** -no file overwrite
* **$cp -R -**recursive copy (including hidden files)
* **$cp -u** -update - copy when source is newer than dest
* **$cp -v** -verbose - print informative messages

1. **mv command :** It is used to move files, although it can also be used to rename files.It moves group of files to different directory.

No additional space is consumed on a disk during renaming. This command normally **works silently** means no prompt for confirmation..

## $ mv [options] source dest Options

* $mv -f - force move by overwriting destination file without prompt
* $mv -i - interactive prompt before overwrite
* $mv -u - update - move when source is newer than destination
* $mv -v - verbose - print source and destination files
* $man mv - help manual

1. **mkdir command :**It is used to make a new directory.With mkdir, you can also set permissions, create multiple directories (folders) at once, and much more.

## syntax- mkdir [option] dir\_name Options

* + **mkdir directory\_name -**Creates a directory in the current location
  + **mkdir {dir1,dir2,dir3,dir4}** -Creates multiple directories in the current location. Do not use spaces inside

{}

* + **mkdir –p directory/path/newdir -**Creates a directory structure with the missing parent directories (if any)
  + **mkdir –m777 directory\_name -**Creates a directory and sets full read, write, execute permissions for all users
  + **mkdir –v directory\_name(s) -**Creates a directory in the current location

1. **head command -** The head command, as the name implies, print the top N number of data of the given input. By default, it prints the first 10 lines of the specified files.

## Syntax: head [OPTION]... [FILE]...

Options -

**$head -n num <filename > :** Prints the first ‘num’ lines instead of first 10 lines. num means number of lines.

**$head -c num <filename> :** Prints the first ‘num’ bytes from the file specified.

1. **tail command -** The tail command, as the name implies, print the last N number of data of the given input. By default it prints the last 10 lines of the specified files.

**Syntax: $t**ail [OPTION]... [FILE]...

Options

* + **$tail <filename> -** display last 10 lines of file
  + **$tail +n <file\_name>**,-data will start printing from line number ‘n’ till the end of the file specified.
  + **$tail -n num <file name> -** display last ‘num’ lines instead of last 10 lines. **num** is mandatory to be specified in command otherwise it displays an error.
  + **$tail -c num <file\_name> -** Prints the last ‘num’ bytes from the file specified.

1. **touch command -** It is used to create a file without any content. The file created using touch command is empty. This command can be used when the user doesn’t have data to store at the time of file creation.

**syntax - $touch <filename> -** we can create multiple files

## Options

* **$touch filename -a** - change the access time only
* **$touch filename -c -** if the file does not exist, do not create it
* **$touch filename -d -** update the access and modification times
* **$touch filename -m** -change the modification time only
* **$touch filename -r** - use the access and modification times of file
* **$touch filename -t** -creates a file using a specified time

1. **rm command - rm** stands for remove here. **rm command** is used to remove objects such as files, directories, symbolic links and so on from the file system like UNIX.

**syntax - rm** [*OPTION*]... *FILE*...

## Options

* + **-f, --force** -ignore nonexistent files, never prompt
  + **-i -** prompt before every removal
  + **-I** - prompt once before removing more than three files, or when removing recursively.
  + **--interactive[=WHEN]** -prompt according to WHEN: never, once (-I), or always (-i).
  + **--one-file-system** - removing a hierarchy recursively, skip any directory that is on a file system different from that of the corresponding command line argument.
  + **--no-preserve-root** -do not treat `/' specially
  + **--preserve-root** - do not remove `/' (default)
  + **-r, -R, --recursive** - remove directories and their contents recursively
  + **-v, --verbose -** explain what is being done

1. **file command -** used to determine the type of a file. ***.file*** type may be of human-readable(e.g. ‘ASCII text’) or MIME type(e.g. ‘text/plain; charset=us-ascii’). This command tests each argument in an attempt to categorize it.

## syntax - file [option] [filename] Options -

* + **$file -b filename :** This is used to display just file type in brief mode.
  + **$file \* <option>** :Command displays the all files’s file type.
  + **$file directoryname/\* option :** This is used to display all files filetypes in particular directory.
  + **$file [range]\* option:** To display the file type of files in specific range.
  + **$file -c option:** Cause a checking printout of the parsed form of the magic file.
  + **$file -f option:** Read the names of the files to be examined from namefile (one per line) before the argument list. Either namefile or atleast one filename argument must be present; to test the standard input, use ‘-’ as a filename argument.
  + **$file -F option :** File and file type are separated by *:*. But we can change separator using -F option.
  + **$file -i option:** To view mime type of file.
  + **$file -s option:** For special files.

**13 . rename command - rename** command in Linux is used to rename the named files according to the regular expression *perlexpr*. It can change the name of the multiple files.

syntax - rename [options] expression replacement file...

Options

* $**rename -s:** This option renames the files ignoring the symbolic links.
* $**rename -v:** This option is used to show which files is being renamed, if there is any.
* **$rename -n :** This option comes into play when the user wants to see only the final change.
* **$rename -o :** This option will not going to overwrite the existing files.
* **$rename -V:** This option will show the version information and exit.

**14. grep command -** The 'grep' command stands for **"global regular expression print"**. grep command filters the content of a file which makes our search easy.

The 'grep' command is generally used with pipe **(|)**.

**Syntax:** command | grep **<searchWord>**

**Example:** cat marks.txt | grep 9

## grep options :

**grep -v** : The 'grep -v' command displays lines not matching to the specified word.

**grep -i** : The 'grep -i' command filters output in a case-insensitive way.

**grep -A** : The grep -A command is used to display the **line after the result**. **grep -B** : Thegrep -B command is used to display the **line before the result**.

**grep -C** : The grep -C command is used to display the **line after and line before** the result.

**15 .vi command -** The vi editor opens in this mode, and it only **understands commands.**

In this mode, you can, **move the cursor and cut, copy, paste the text.**

This mode also saves the changes you have made to the file

**syntax - $**vi <filename\_NEW> or <filename\_EXISTING>

## Supporting commands -

* i - Insert at cursor (goes into insert mode)
* a - Write after cursor (goes into insert mode)
* A - Write at the end of line (goes into insert mode)
* ESC - Terminate insert mode
* u - Undo last change
* U - Undo all changes to the entire line
* o - Open a new line (goes into insert mode)
* dd - Delete line
* 3dd - Delete 3 lines.
* D - Delete contents of line after the cursor
* C - Delete contents of a line after the cursor and insert new text. Press ESC key to end insertion.
* dw - Delete word
* 4dw - Delete 4 words
* cw - Change word
* x - Delete character at the cursor
* r - Replace character
* R - Overwrite characters from cursor onward
* s - Substitute one character under cursor continue to insert
* S - Substitute entire line and begin to insert at the beginning of the line
* ~ - Change case of individual character

1. **ping command -** The ping command stands for **(Packet INternet Groper)**. It checks the connectivity between two nodes that is whether a server is reachable or not. ping command keep executing and sends the packet until you interrupt.

To stop from execution press **ctrl + c.**

## Syntax : $ping [OPTIONS] DESTINATION

**Example : $**ping javatpoint.com

You can use IP address also with ping command.

**Example:** ping 2.2.2.2

## Option :

-c : To limit the ping packet without using **ctrl + c** use option c followed by the number of packet to be send.

**Syntax: $**ping -c **<number> <destination>**

1. **cal command -** cal command shows current month calendar as output.

**cal** command is a calendar command in Linux which is used to see the calendar of a specific month or a whole year.

**Syntax - $**cal <month> <year>

1. **Sudo Command -** The **sudo command** allows you to run programs with the security privileges of another user (by default, as the superuser). It prompts you for your personal password and confirms your request to execute a **command** by checking a file, called **sudoers** , which the system administrator configures.

## Syntax - $sudo [option]

Options

* + **$sudo -V:** The -V (version) option causes sudo to print the version number and exit. If the invoking user is already root, the -V option will print out a list of the defaults sudo was compiled with.
  + **$sudo -l:** The -l (list) option will print out the commands allowed (and forbidden) the user on the current host.
  + **$sudo -h or –help:** The -h (help) option causes sudo to print a usage message and exit.
  + **$sudo -v:** If, given the -v (validate) option, sudo will update the user’s timestamp, prompting for the user’s password if necessary.
  + **$sudo -k:** The -k (kill) option to sudo invalidates the user’s timestamp.So, the next time sudo is run a

password will be required.

* + **$sudo -b:** The -b (background) option tells sudo to run the given command in the background.

1. **Man command** - The "man" is a short term for manual page. In unix like operating systems such as linux, man is an interface to view the system's reference manual.

A user can request to display a man page by simply typing man followed by a space and then argument. Here its argument can be a command, utility or function. A manual page associated with each of these arguments is displayed.

**Syntax of man:** man [option(s)] keyword(s)

But generally [option(s)] are not used. Only keyword is written as an argument.

**Example** : man ls

## Options :

* + $man -aw -List all available sections of a command.
  + $man -a - To view all man pages of a command.
  + $sman -k (apropos) - Shows a list of results in man page containing a keyword match.
  + $man -f, whatis - It displays description from manual page if available.
  + $man Whereis - Used to determine location of a man page

***THANK YOU***