**Compilation System:**

\*preprocessing Phase: it includes header and expand macros, which means exporting the libraries of the header

\*Compilation Phase: it checks semantic and syntax errors

\*Assembly Phase: converts into binary form which is not runnable

\*Linking Phase: file is ready to be loaded into the main memory and excuted

->Process: tasks under execution

->Thread: it is a light weight process and it is a part of process. Each process may contain multiple threads

->multitasking: when we want to perform multiple task under one processor.

->multiprocess: when we want to execute more than one process

->multithreading: when we want to execute more than one function with in a single process

->comments go under text segment in memory

->functions are instack memory

->Between software and hardware there will be an Operating System

-> UNIX OS is made up of 3 parts :

\*Kernel : Kernel is the part of os. It act as interface between hardware and process of computer.

\*shell: shell is command interpreter of UNIX. . It act as interface between user and kernel.

\*user:

->Booting Process: It is the process which takes place when we off the and on the computer.

**Files:**

->file is a collection of data

->file content are treated as series of bytes

->devices are also treated as special file, file size can grow dynamically.

->internally each file is assigned a unique identification number called INODE(Information node 0r index node).

**File Naming Conventions:**

->maximum file length depends upon the kernel configuration(255)

->file names are case sensitive

->embedded spaces and tab names are not allowed

-> A file consists of only one INODE.

->INODE structure used to maintain information about the file

**Directory:**

->All the files are grouped together in the directory.

->Factors that comes under etc,bin,usr,dev,lib,tmp,home.

->bin under root is a executable file within os and bin under user is executable file install from third parties.

**Links-Hard Link:**

->hard links can be apply only on files.

->it act as physical file

-> if we delete the original file it will not effect the copy file

**Soft Link or Symbolic Link:**

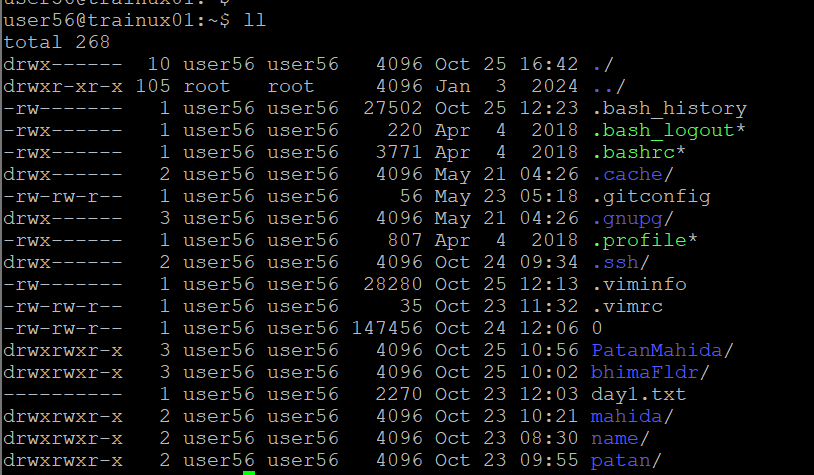
->soft link can be apply for both file and directory

->it act as short cut file

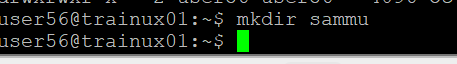
-> if we delete the original file it will effect the copy file

**UNIX COMMANDS:**

->~$ ll = it will give the all available files and to see authentication of a file or to see a file is in which mode.

****

->~$ mkdir sammu: Here sammu is the directory name which we want to create.



->$ cd sammu: change directory is used to change the directory to the current directory.

\*Here we can see the directory is changed to current directory.

A screenshot of a computer

Description automatically generated

->~/sammu$ touch file.txt: Here we are creating a txt file inside sammu directory.



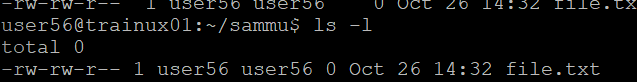
->~/sammu$ vi file.txt: The file will be open in vi editor and we can write.



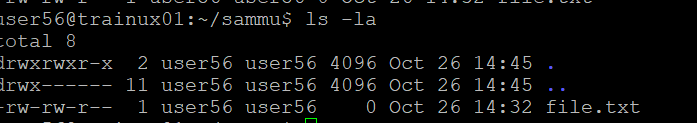
A screen shot of a computer

Description automatically generated

->~/sammu$ ls -l: It give non-hidden files .

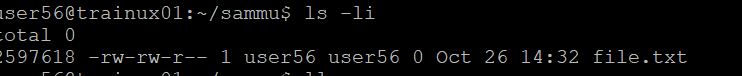


->~/sammu$ ls -la: it gives hidden files.



->~/sammu$ ls -li: To know the INODE number of a file.

\* Here 2597618 is the INODE number of a file file.txt.



->~/sammu$ man mkdir: It will give us the manual page information about any command, like how to use the commonds.



**File Permissions in UNIX: There are three file permissions in unix**

\*read=x

\*write=w

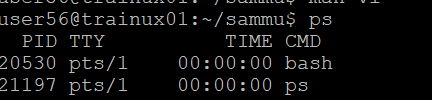
\*execute=x

->rwx-rw-r: Here rwx is for owner or user. For them the three permissions are given.

\*rw: Here rw is for group. For them only read and write permissions are given.

\*r: Here r is for others. For them only read permission is given.

->~/sammu$ ps: Gives no of processors running.



->~/sammu$ ps -ef: Gives no of processors running in the background.

