# 2. Operating System and User Interface

Operating System (OS) is an interface between user and the computer applications that handles the input output (I/O) operations in the computer. An operating system is a program that acts as an intermediary between the application programs and the computer hardware. You cannot directly use computer applications (or programs) with computer hardware without a translation system between the hardware and the applications. This translation system is called the OS. The Windows or Mac OS (IOS) works behind-the-scenes to run your computer (i.e. the software and the hardware). It tells the computer what to do when it starts up (the process is called booting) and keeps track of your documents, files, and other software. It also provides the standard user interface component (like menus and the desktop) that you see when you look at your computer screen. Both the Windows and the Mac OS operating systems use a graphical interface (pictures or icons instead of text) that allow you to immerse yourself in multitasking (accessing multiple applications and files simultaneously). While working with windows OS you play with a series of boxes and icons, which can be opened and closed as needed.

**Operating System Components**

Process management

Memory management

File management

Secondary storage management

Networking

System Protection

Command-interpreter system

***The Operating System Layer***

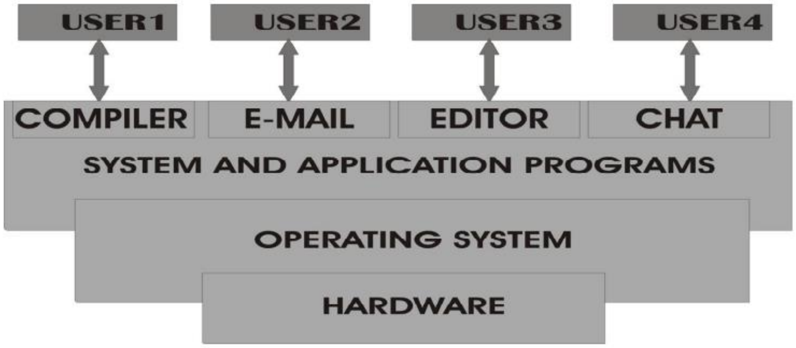


**Figure 3: Operating System Layers**

### Functions of the OS

1. Resource sharing: manages resources by ensuring proper and effective use.  Provision of virtual machine: Hidings the hardware details from the user.

2. Memory management: It manages the effective usage of internal memory, RAM while running multi- programs.  Protector and error handling.
3. File management, saving, opening the saved file, updating, copying and deleting.
4. Facilitates booting (loading the OS from disk to RAM).
5. Manages multitasking: determines which application should run, in what order, and how much time should be allowed for each application before giving another application access to run.

**Classification of OS**

This could be done based on the number of program they can handle at the same time or the number of user(s) that can be accommodated on the system simultaneously.  Hence we have single tasking and multitasking and, single user and multi-user OS.

Single user OS is multitasking but can only allow only one user at a time e.g. PC-DOS, MS-DOS, CP/M, OS/2. Multi-user OS is multitasking and at the same time multi-user (See the computer Network section for the architecture). This is done when a number of computers (workstation) connect to a central computer so that all the other computers can use the work on the central computer e.g. UNIX, XENIX, PC-MOS, windows NT, Linux and Novell. These OSs are called Network OS.

**Distributed operating System**

Distributed systems use multiple central processors to serve multiple real time application and

multiple users. Data processing jobs are distributed among the processors accordingly to which one can perform each job most efficiently. The processors communicate with one another through various communication lines (such as high-speed buses or telephone lines). These are referred as loosely coupled systems or distributed systems. Processors in a distributed system may vary in size and function. These processors are referred as sites, nodes, and computers and so on.

The advantages of distributed systems are following:

1. With resource sharing facility user at one site may be able to use the resources available at another node.
2. Speedup the exchange of data with one another via electronic mail.
3. If one site fails in a distributed system, the remaining sites can potentially continue operating.
4. Better service to the customers.
5. Reduction of the load on the host computer.
6. Reduction of delays in data processing

**Network operating System**

Network Operating System runs on a server and provides server the capability to manage data, users, groups, security, applications, and other networking functions. The primary purpose of the network operating system is to allow shared file and printer access among multiple computers in a network, typically a local area network (LAN), a private network or to other networks. Examples of network  operating systems are Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, and BSD.

The advantages of network operating systems are following:

1. Centralized servers are highly stable.
2. Security is server managed.
3. Upgrades to new technologies and hardware can be easily integrated into the system.
4. Remote access to servers is possible from different locations and types of systems.

The disadvantages of network operating systems are following.

1. High cost of buying and running a server.
2. Dependency on a central location for most operations.
3. Regular maintenance and updates are required.

**Computer Networks**

A computer network is a set of computers sharing resources located on or provided by network nodes. Computers use common communication protocols over digital interconnections to communicate with each other. These interconnections are made up of telecommunication network technologies based on physically wired, optical, and wireless radio-frequency methods that may be arranged in a variety of network topologies. Computer networks support many applications and services, such as access to the World Wide Web, digital video and audio, shared use of application and storage servers, printers and fax machines, and use of email and instant messaging applications.

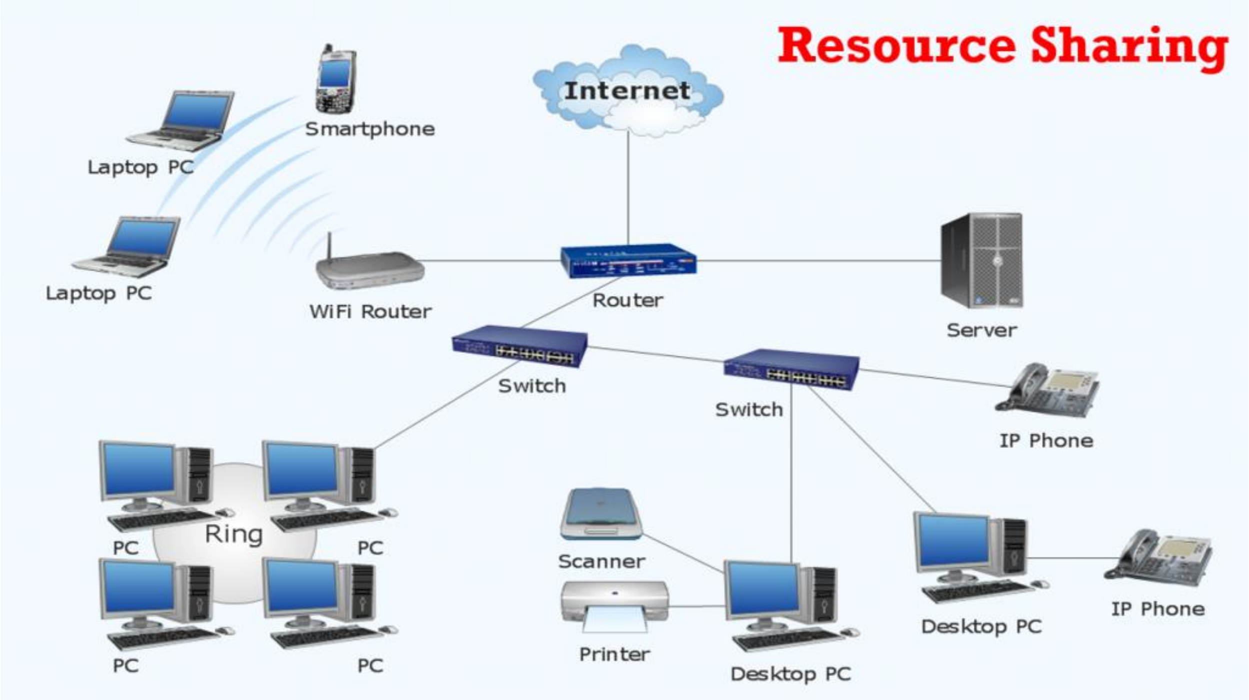
The nodes of a computer network can include personal computers, servers, networking hardware, or other specialized or general-purpose hosts. They are identified by network addresses and may have hostnames. Hostnames serve as memorable labels for the nodes and are rarely changed after initial assignment. Network addresses serve for locating and identifying the nodes by communication protocols such as the Internet Protocol.

Computer networks may be classified by many criteria, including the transmission medium used to carry signals, bandwidth, communications protocols to organize network traffic, the network size, the topology, traffic control mechanisms, and organizational intent.

LAN: Local Area Network is small network setup within the office premises to share the resources from Server to clients (individual PCs).

WAN: Wide Area Network is setup within the organization in a given location to share the resources within organization.

WWW: World Wide Wed is the largest Network of computers using Internet Protocol (IP) to share the resources worldwide. See the following figure to get an idea of how a modern day organization utilizes the LAN, WAN and WWW network to accomplish the computing needs.

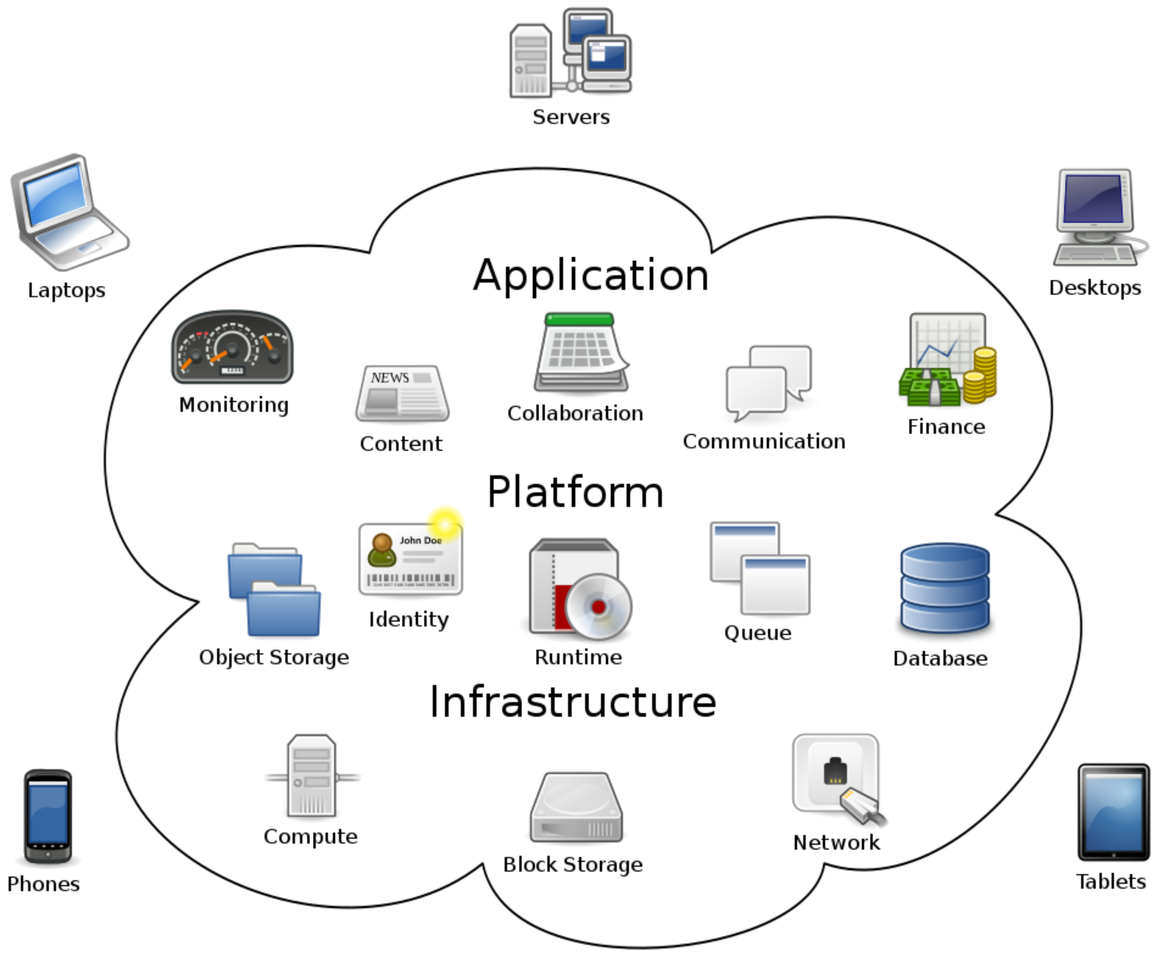


A typical network setup in the modern-day computing environment

## Cloud Computing

Cloud computing is the on-demand availability of computer system resources, especially data storage (cloud storage) and computing power, without direct active management by the user. Large clouds often have functions distributed over multiple locations, each of which is a data center. Cloud computing relies on sharing of resources to achieve coherence and typically uses a pay-as-you-go model, which can help in reducing capital expenses but may also lead to unexpected operating expenses for users. The advantages of using cloud services are lower implementation costs, more secure platform to store and retrieve the resources and availability of latest technology to use. However, the downside is the exposure of sensitive to data to third party service provider and the possibility of infringement of property rights.

The big player in cloud services are: Azure from Microsoft, AWS from Amazon, Google Cloud, and Oracle.



Cloud computing metaphor: the group of networked elements providing services does not need to be addressed or managed individually by users; instead, the entire provider-managed suite of hardware and software can be thought of as an amorphous cloud.

**Virtual Machines**

A Virtual Machine (VM) is a compute resource that uses software instead of a physical computer to run programs and deploy apps. One or more virtual "guest" machines run on a physical "host" machine.  Each virtual machine runs its own operating system and functions separately from the other VMs, even when they are all running on the same host. This means that, for example, a virtual MacOS virtual machine can run on a physical PC.

Virtual machine technology is used for many use cases across on-premises and cloud environments. More recently, public cloud services are using virtual machines to provide virtual application resources to multiple users at once, for even more cost efficient and flexible compute.

**What are virtual machines used for?**

VMs allow a business to run an operating system that behaves like a completely separate computer in an app window on a desktop. VMs may be deployed to accommodate different levels of processing power needs, to run software that requires a different operating system, or to test applications in a safe, sandboxed environment.

Virtual machines have historically been used for server virtualization, which enables IT teams to consolidate their computing resources and improve efficiency. Additionally, virtual machines can perform specific tasks considered too risky to carry out in a host environment, such as accessing virus-infected data or testing operating systems. Since the virtual machine is separated from the rest of the system, the software inside the virtual machine cannot tamper with the host computer.

*Advantages of virtual machines*

Virtual machines are easy to manage and maintain, and they offer several advantages over physical machines:

VMs can run multiple operating system environments on a single physical computer, saving physical space, time and management costs.

Virtual machines support legacy applications, reducing the cost of migrating to a new operating system. For example, a Linux virtual machine running a distribution of Linux as the guest operating system can exist on a host server that is running a non-Linux operating system, such as Windows.

VMs can also provide integrated disaster recovery and application provisioning options.

*Disadvantages of virtual machines*

While virtual machines have several advantages over physical machines, there are also some potential disadvantages:

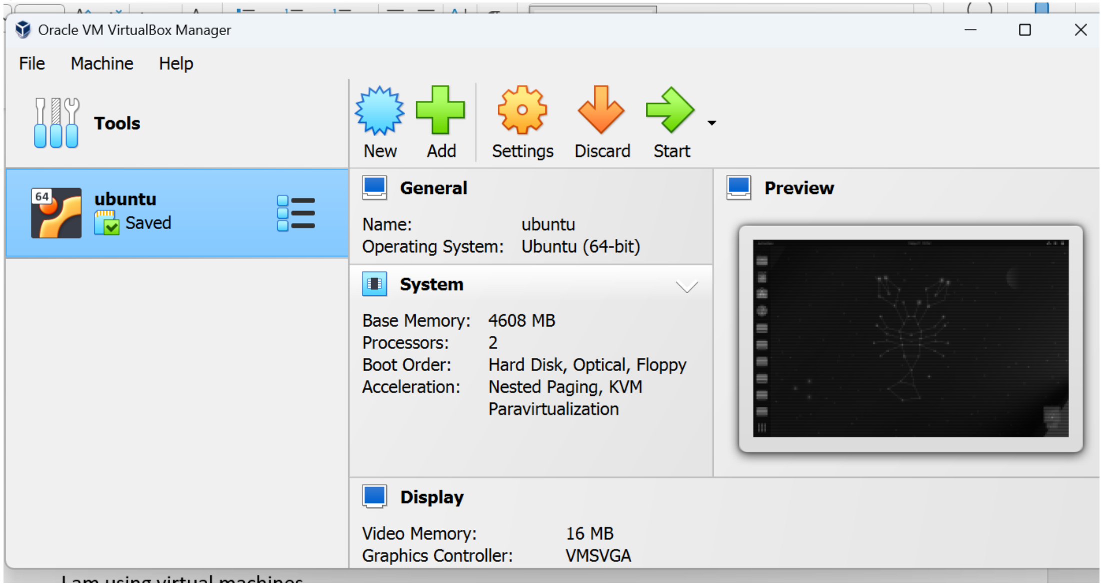
Running multiple virtual machines on one physical machine can result in unstable performance if infrastructure requirements are not met.

Virtual machines are less efficient and run slower than a full physical computer. Most enterprises use a combination of physical and virtual infrastructure to balance the corresponding advantages and disadvantages.

# Linux OS and File Systems

Unix was created at Bell Labs in 1970 written in the C programming language, which was developed at the same time. It supports large numbers of simultaneous users, runs with few alterations on many hardware platforms (provides some platform independence) and of course it was and is a simple, elegant, and easy to use (at least compared to its predecessors). Linux is a relatively new UNIX flavor derived from the work of Linus Torvalds, who was interested to develop a UNIX for academic use.  Linux is one of the many versions of the UNIX operating system.  Working on Linux means working on one of the flavors of UNIX. The main  advantage that Linux is absolutely free, you need not spend even the cost CD it  can be entirety free downloadable from the Internet. (No registration fees, no costs per user, free updates, and freely available Operating System and Networking Lab source code). It is portable (means can be configured on any processor like Intel, Solaris, etc.), dual-bootable, fast, reliable, secure and versatile. These properties make it popular among the System Administrators. While working on it you may realize many more important features and advantages of Linux. Most of the exercises are command line based but similar exercises you can try on your GUI based Linux.

I am using virtual machines called Oracle VM with ubuntu linux OS. By clicking on Start icon, we get the ubuntu on, the host is Windows OS.



General

Name: ubuntu
Operating System: Ubuntu (64-bit)

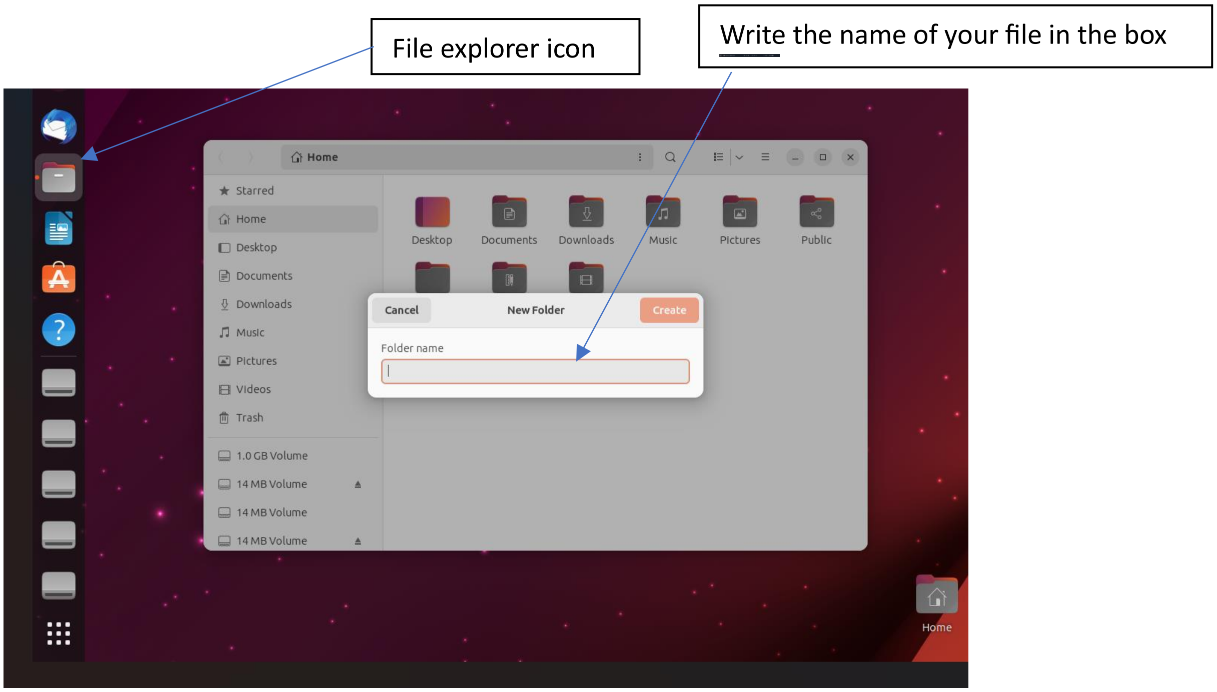
System

Base Memory: 4608 MB
Processors: 2
Boot Order: Hard Disk, Optical, Floppy
Acceleration: Nested Paging, KVM Paravirtualization

Display

Video Memory: 16 MB
Graphics Controller: VMSVGA

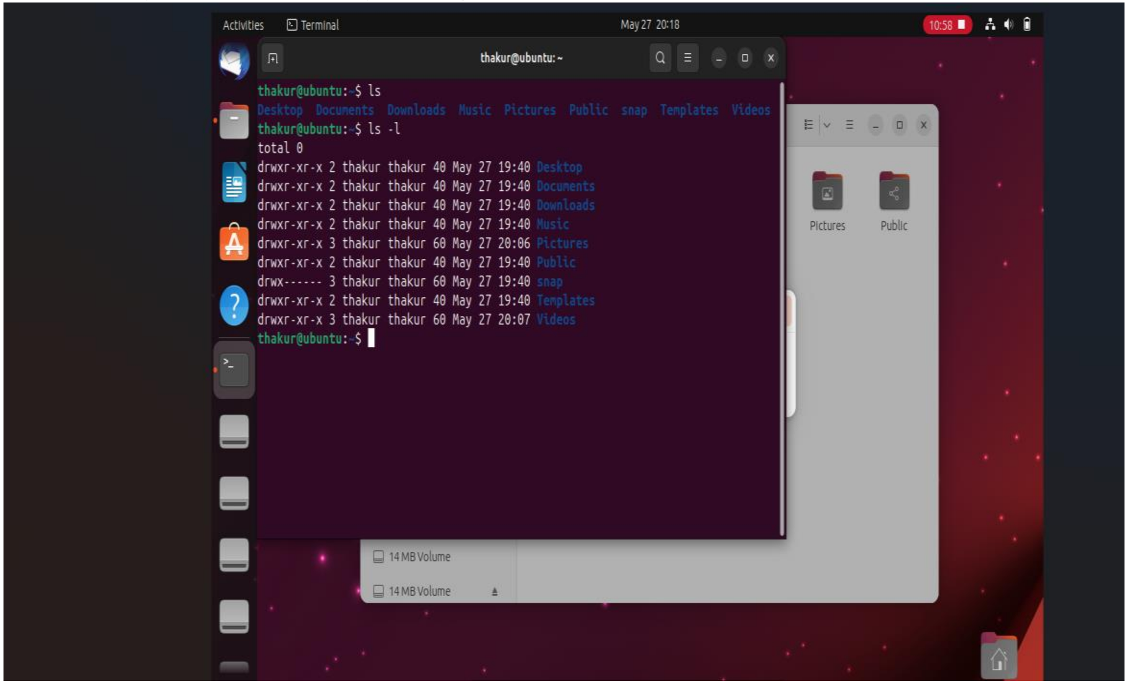
We can run OS commands either via command line interface (CLI) or graphical user interface (GUI). Ubuntu provides nice GUI to explore the file systems and you can create a folder/directory by right clicking on the folder icon as shown below:



File explorer icon

Write the name of your file in the box

Alternatively, we can use CLI by clicking on the Terminal icon and run the OS commands as shown below:



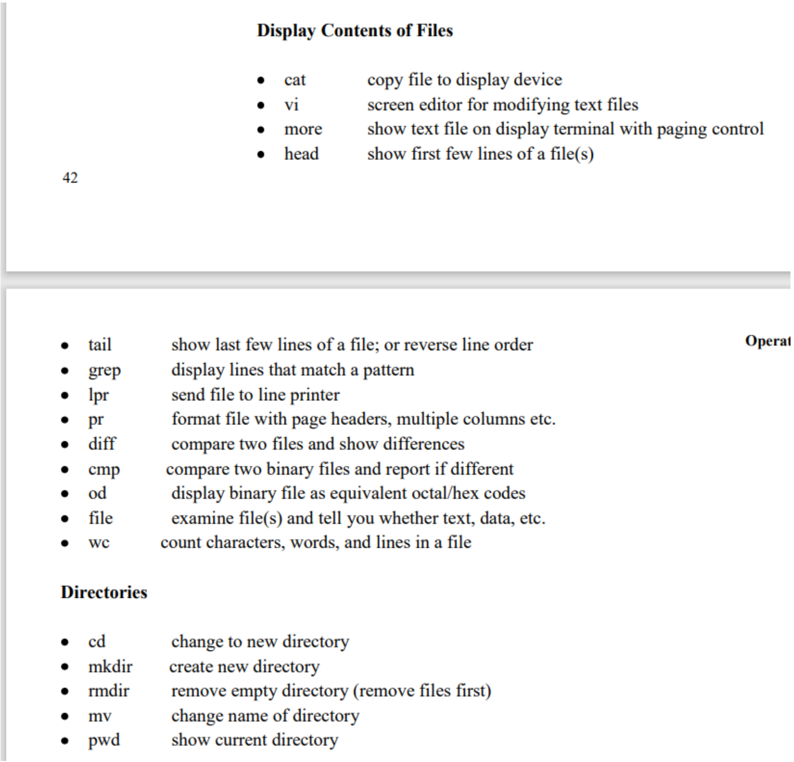
```
thakur@ubuntu:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  snap  Templates  Videos
thakur@ubuntu:~$ ls -l
total 0
drwxr-xr-x 2 thakur thakur 40 May 27 19:40 Desktop
drwxr-xr-x 2 thakur thakur 40 May 27 19:40 Documents
drwxr-xr-x 2 thakur thakur 40 May 27 19:40 Downloads
drwxr-xr-x 2 thakur thakur 40 May 27 19:40 Music
drwxr-xr-x 3 thakur thakur 60 May 27 20:06 Pictures
drwxr-xr-x 2 thakur thakur 40 May 27 19:40 Public
drwx------ 3 thakur thakur 60 May 27 19:40 snap
drwxr-xr-x 2 thakur thakur 40 May 27 19:40 Templates
drwxr-xr-x 3 thakur thakur 60 May 27 20:07 Videos
thakur@ubuntu:~$ 
```

Important Linux commands:

| | |
|---|---|
| date | show date and time |
| history | list of previously executed commands |
| pine | send or receive mail messages |
| msgs | display system messages |
| man | show on-line documentation by program name |
| info | on-line documentation for GNU programs |
| w, who | who is on the system and what are they doing |
| who am i | who is logged onto this terminal |
| top | show system status and top CPU-using processes |
| uptime | show one line summary of system status |
| finger | find out info about a user@system |

### File Management

| | |
|---|---|
| cat | combine files |
| cp | copy files |
| ls | list files in a directory and their attributes |
| mv | change file name or directory location |
| rm | remove files |
| ln | create another link (name) to a file |
| chmod | set file permissions |
| des | encrypt a data file with a private key |
| find | find files that match specified criteria |



### Display Contents of Files

| | |
|---|---|
| cat | copy file to display device |
| vi | screen editor for modifying text files |
| more | show text file on display terminal with paging control |
| head | show first few lines of a file(s) |

| | |
|---|---|
| tail | show last few lines of a file; or reverse line order |
| grep | display lines that match a pattern |
| lpr | send file to line printer |
| pr | format file with page headers, multiple columns etc. |
| diff | compare two files and show differences |
| cmp | compare two binary files and report if different |
| od | display binary file as equivalent octal/hex codes |
| file | examine file(s) and tell you whether text, data, etc. |
| wc | count characters, words, and lines in a file |

### Directories

| | |
|---|---|
| cd | change to new directory |
| mkdir | create new directory |
| rmdir | remove empty directory (remove files first) |
| mv | change name of directory |
| pwd | show current directory |

## Description of Commonly Used UNIX Commands

The description for the most commonly used UNIX commands is given below in an alphabetic order.

a) cat

cat allows you to read multiple files and then print them out. You can combine files by using the > operator and append files by using >>.
Syntax: cat [argument] [specific file]
Example:

```
cat abc.txt
```

If you want to append three files (abc.txt, def.txt, xyz.txt), give the command as,

```
cat abc.txt def.txt xyz.txt > all
```

b) cd, chdir

cd (or chdir) stands for "change directory". This command is the key command to move around your file structure.
Syntax: cd [name of directory you want to move to]
When changing directories, start with / and then type the complete file path, like

```
cd /vvs/abc/def
```

in this eg. You are moving from vvs directory to abc directory then move to def directory .
You can also move relative to the current directory by typing

```
cd vvs/abc/def
```

To move relative to the parent directory of your current directory, use

```
Cd ../vvs/abc/def
```

c ) chmod

chmod (which stands for "change mode") changes who can access a particular file. A "mode" is created by combining the various options from who, opcode, and permission.
Syntax: chmod [option] mode file

If you look at a list of files using the long list command ls –l, you'll see the permissions, owner, file size, modification time, and filename. The first column of the list shows who can read, write, and execute the files or directories, in other words, the permissions. It basically shows who has permission to do what to a given file or directory. r stands for "read" and means that you're allowed to read the file or directory. w stands for "write" and gives permission to edit or change the file as well as create, move, rename, or remove a directory. x stands for "execute" which gives permission to run a file or search a directory. Every file or directory has four sets of rwx permissions. The first set represents the user (u), the second set represents the group (g), the third set represents other (o), and the fourth set represents all (a). The column will look like this: rwxrwxrwx

Each set of rwx represents user, group, and other respectively. Only the owner of a file or a privileged user may change the permissions on a file. There are two ways to change permissions on a file or directory, either numerically or by using lettered commands. Both ways use the command chmod. To add permissions to a file, you use +, to remove permissions you use-.

For example, take a file:

```
-rw-r--r-- 1 yash mony 476 Apr 14 17:13 vvs.txt
```

To allow a group (mony, in this case) "write" access, you would type:

```
chmod g+w vvs.txt
```

If you wanted to remove "read" ability from "other" you would type:

```
chmod o-r vvs.txt
```

It is also possible to specify permissions using a three-digit sequence. This is a more efficient way to change permissions (or at least it requires less typing), so use this method if it doesn't confuse you. Each type of permission is given an octal value. Read is given the value of 4, write is given the value of 2, and execute is given the value of 1. These values are added together for each user category. The permissions are changed by using a three-digit sequence with the first digit representing owner permission, the second digit representing group permission, and the third digit representing other permission. For example, if you wanted to make vvs.txt readable, writable, and executable for the user, readable and writable for the group, and readable for other, you would type: chmod 764 vvs.txt

The first digit means readable and writable for the user (4+2+1), the second digit means readable and writable for the group (4+2+0), and the third digit means readable for other (4+0+0). if you want to change the permissions on a directory tree use the -R option. chmod –R will recursively change the permissions of directories and their contents.

d) cp

The cp command copies files or directories from one place to another. You can copy a set of files to another file, or copy one or more files under the same name in a directory. If the destination of the file you want to copy is an existing file, then the existing file is overwritten. If the destination is an existing directory, then the file is copied into that directory.

Syntax: cp [options] file1 file2 If you want to copy the file favourites.html into the directory called laksh, you give

the command as:

cp favourites.html /vvs/laksh/

A handy option to use with cp is -r. This recursively copies a particular directory and all of its contents to the specified directory, so you won't have to copy one file at a time.

e) date

The date command can be used to display the date or to set a date. In unix the term date includes the time as well.

Syntax: date [option] [+format]

date [options] [string]

The first structure shows how date can be used to display the current date. A certain format can be specified in which the date should be displayed. Check the Unix manual for specific formats and options. The second structure allows you to set the date by supplying a numeric string. Only privileged users will be able to use this second command structure.

f) diff

diff displays the lines that differ between two given files.

Syntax: diff [options] [directory options] file1 file2

diff can be an extremely valuable tool for both checking errors and building new pages. If you run a diff between two files, you'll be shown what differences the files have line by line. The lines referring to file1 are marked with the < symbol. The lines referring to file2 are marked by the > symbol. If the file is a directory, diff will list .the file in the directory that has the same name as file2. If both of the files are directories, diff will list all the lines differing between all files that have the same name. If you have a file that is not working properly, it can be a great help to check it against a similar file that is working. It will often quickly alert you to a line of code that's missing. A handy option to use if you want to generally compare two files without noting the complex differences between them is the -h option (h stands for half-hearted). Using -i as an option will ignore differences in uppercase and lowercase characters between files, and -b will ignore repeating blanks and line breaks.

g) exit

The exit command allows you to terminate a process that is currently occurring. For example, if you wanted to leave a remote host that you were logged onto (see rlogin also), you should type exit. This would return you to your home host.

h) find

find searches through directory trees beginning with each pathname and finds the files that match the specified condition(s). You must specify at least one pathname and one condition.

Syntax: find pathname(s) condition(s)

There are several handy conditions you can use to find exactly what you want. The -name condition will find files whose names match a specified pattern. The structure for the name condition is: find pathname -name pattern

The condition -print will print the matching files to the pathname specified. -printcan also be used in conjunction with other conditions to print the output. If you wanted to find all the files named favorites.html in the directory Ram, then

you'd do this:

```
find /Ram -name favorites.html –print
```

This looks through the directory Ram and finds all the files in that directory that contain favorites.html, then prints them to the screen. Your output would look like this:

```
 /Ram/sixteen_candles/favorites.html

/Ram/favorites.html

/Ram/breakfast_club/favorites.html
```

All meta-characters (!, *, ., etc.) used with -name should be escaped (place a \ before the character) or quoted. Meta-characters come in handy when you are searching for a pattern and only know part of the pattern or need to find several similar patterns.

For example, if you are searching for a file that contains the word "favorite", then use the meta-character * to represent matching zero or more of the preceding characters.

This will show you all files which contain favorite.

```
find /Ram  -name '*favorite*' -print
```

This looks through the directory Ram and finds all the files in that directory that contain the word "favorite". The output would look like this:

```
/Ram/sixteen_candles/favorites.html

/Ram/favorites.html

/Ram/least_favorites.html

/Ram/breakfast_club/favorites.html

/Ram/favorite_line.html
```

The -user condition finds files belonging to a particular user ID or name.

i) grep

The grep command searches a file or files for lines that match a provided regular expression ("grep" comes from a command meaning to globally search for a regular expression and then print the found matches).

Syntax: grep [options] regular expression [files]

To exit this command, type 0 if lines have matched, 1 if no lines match, and 2 for errors. This is very useful if you need to match things in several files. If you wanted to find out which files in our vvs  directory contained the word "bca" you could use grep to search the directory and match those files with that word. All that you have to do is give the command as shown:

```
grep 'bca' /vvs/*
```

The * used in this example is called a meta-character, and it represents matching zero or more of the preceding characters. In this example, it is used to mean "all files and directories in this directory". So, grep will search all the files and directories in vvsand tell you which files contain "bca".

j) head

Displays the first ten lines of a file, unless otherwise stated.    Syntax: head [-n] [files]

For example, the following command will display the first 15 lines of favourites.html.

```
head -15 favourites.html
```

k) kill

kill ends the execution of one or more process ID's. In order to do this you must own the process or be designated a privileged user. To find the process ID of a certain job give the command ps.

Syntax: kill [options] PIDs

There are different levels of intensity to the kill command, and these can be represented either numerically or symbolically. kill -1 or HUP makes a request to the server to terminate the process, while kill -9 or kill KILL forces a process to terminate absolutely. Most politely, UNIX users will attempt to kill a process using-1 first before forcing a process to die.

l) ls

ls will list all the files in the current directory. If one or more files are given, ls will display the files contained within "name" or list all the files with the same name as "name". The files can be displayed in a variety of formats using various options.

Syntax: ls [options] [names]

ls is a command you'll end up using all the time. It simply stands for list. If you are in a directory and you want to know what files and directories are inside that directory, type ls. Sometimes the list of files is very long and it flies past your screen so quickly .You miss the file you want. To overcome this problem give the command as shown below:

```
ls | more
```

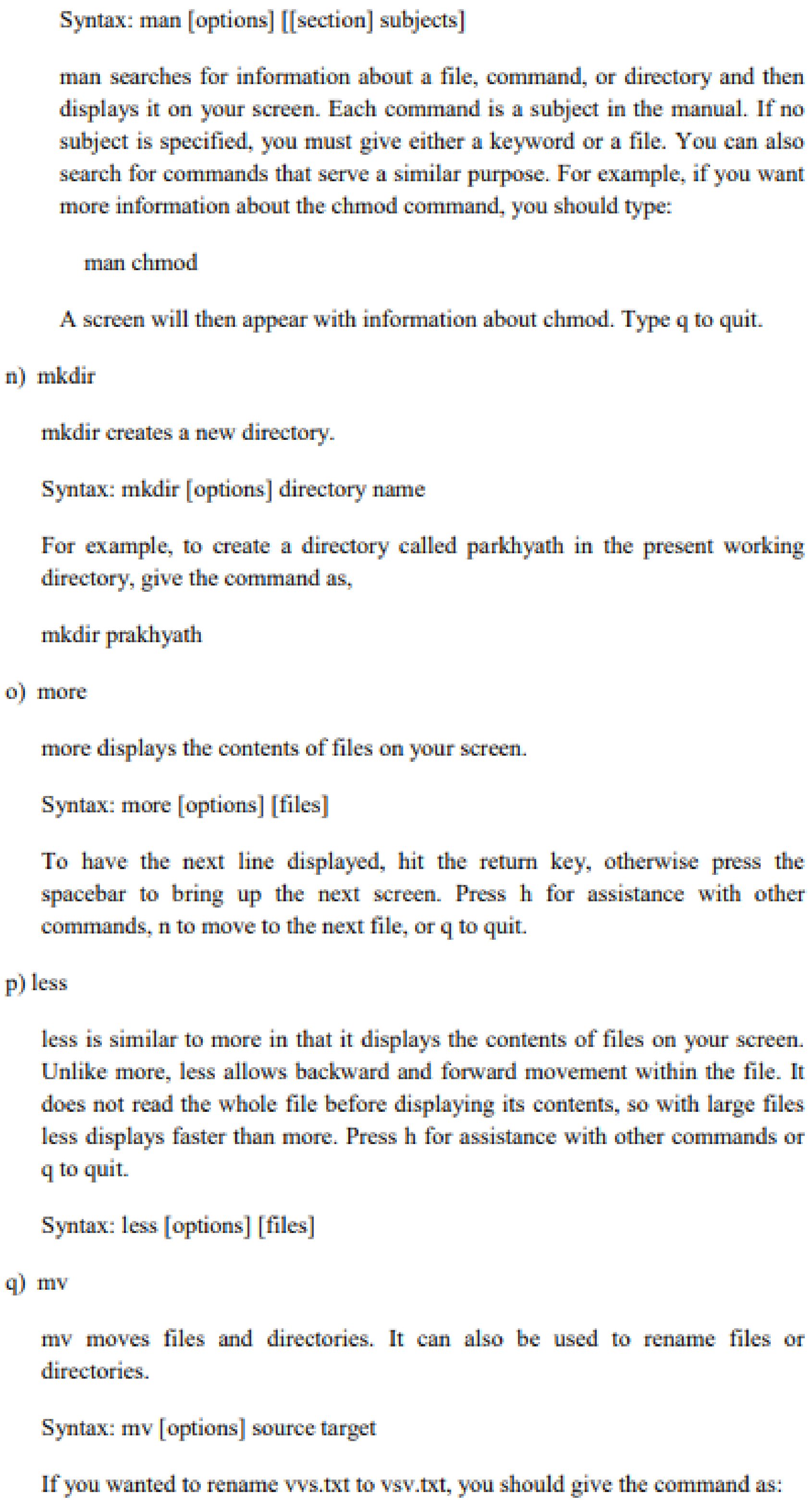
The character | (called pipe) is typed by using shift and the \ key. | more will show as many files as will fit on your screen, and then display a highlighted "more" at the bottom. If you want to see the next screen, hit enter (for moving one line at a time) or the spacebar (to move a screen at a time). | more can be used anytime you wish to view the output of a command in this way. A useful option to use with ls command is -l. This will list the files and directories in a long format. This means it will display the permissions (see chmod), owners, group, size, date and time the file was last modified, and the filename.

```
 drwxrwxr-x  vvs   staff  512  Apr 5 09:34 sridhar.txt

 -rwx-rw-r-- vvs   staff  4233 Apr 1 10:20 resume.txt

-rwx-r--r-- vvs   staff  4122 Apr 1 12:01 favourites.html
```

There are several other options that can be used to modify the ls command, and many of these options can be combined. -a will list all files in a directory, including those files normally hidden. -F will flag filenames by putting / on directories, @ on symbolic links, and * on executable files.

m) man

The man command can be used to view information in the online Unix manual.



Syntax: man [options] [[section] subjects]

man searches for information about a file, command, or directory and then displays it on your screen. Each command is a subject in the manual. If no subject is specified, you must give either a keyword or a file. You can also search for commands that serve a similar purpose. For example, if you want more information about the chmod command, you should type:

```
man chmod
```

A screen will then appear with information about chmod. Type q to quit.

n) mkdir

mkdir creates a new directory.

Syntax: mkdir [options] directory name

For example, to create a directory called parkhyath in the present working directory, give the command as,

```
mkdir prakhyath
```

o) more

more displays the contents of files on your screen.

Syntax: more [options] [files]

To have the next line displayed, hit the return key, otherwise press the spacebar to bring up the next screen. Press h for assistance with other commands, n to move to the next file, or q to quit.

p) less

less is similar to more in that it displays the contents of files on your screen. Unlike more, less allows backward and forward movement within the file. It does not read the whole file before displaying its contents, so with large files less displays faster than more. Press h for assistance with other commands or q to quit.

Syntax: less [options] [files]

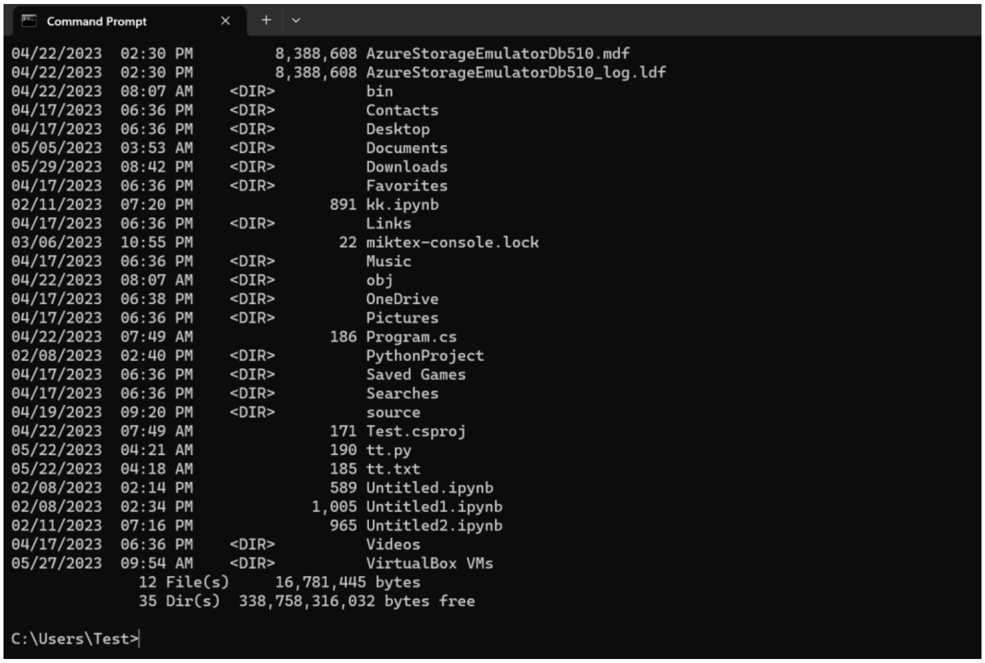
q) mv

mv moves files and directories. It can also be used to rename files or directories.

Syntax: mv [options] source target

If you wanted to rename vvs.txt to vsv.txt, you should give the command as:

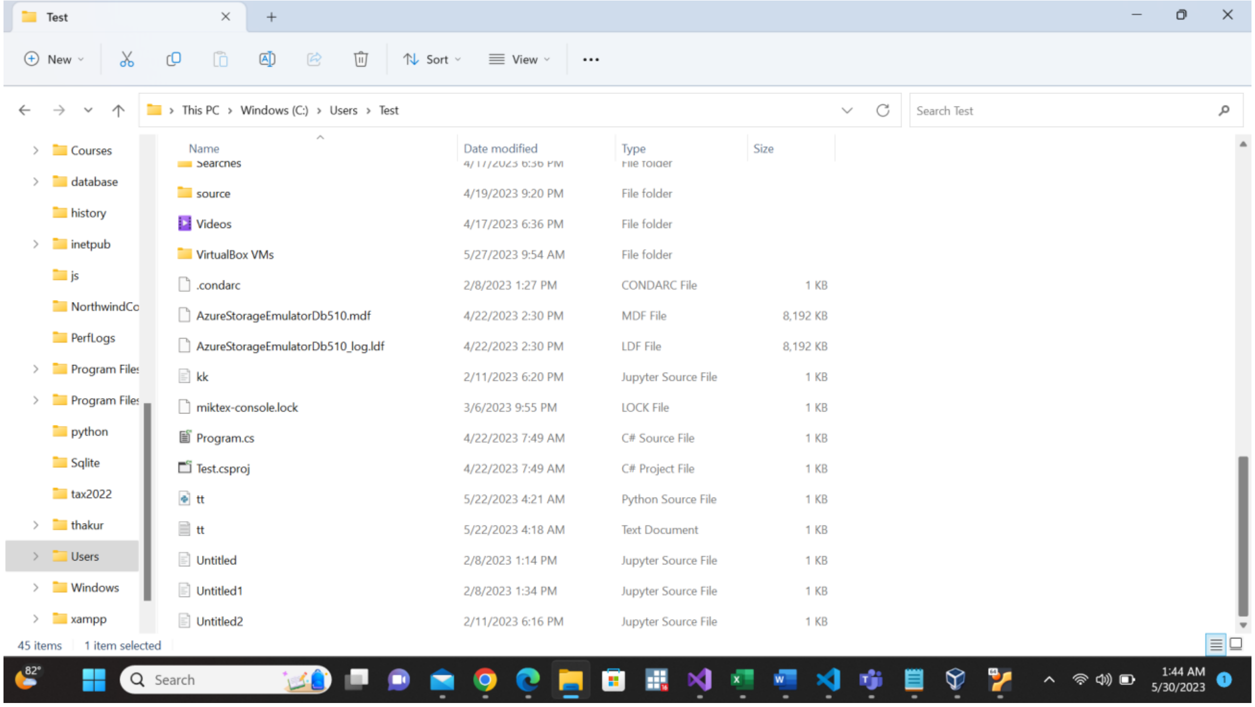
Similarly, we can run OS commands using CLI or GUI in Windows as seen in the classroom. Just type cmd in the search box and press Enter to goto command prompt as shown below:



```
04/22/2023  02:30 PM         8,388,608 AzureStorageEmulatorDb510.mdf
04/22/2023  02:30 PM         8,388,608 AzureStorageEmulatorDb510_log.ldf
04/22/2023  08:07 AM    <DIR>          bin
04/17/2023  06:36 PM    <DIR>          Contacts
04/17/2023  06:36 PM    <DIR>          Desktop
05/05/2023  03:53 AM    <DIR>          Documents
05/29/2023  08:42 PM    <DIR>          Downloads
04/17/2023  06:36 PM    <DIR>          Favorites
02/11/2023  07:20 PM               891 kk.ipynb
04/17/2023  06:36 PM    <DIR>          Links
03/06/2023  10:55 PM                22 miktex-console.lock
04/17/2023  06:36 PM    <DIR>          Music
04/22/2023  08:07 AM    <DIR>          obj
04/17/2023  06:38 PM    <DIR>          OneDrive
04/17/2023  06:36 PM    <DIR>          Pictures
04/22/2023  07:49 AM               186 Program.cs
02/08/2023  02:40 PM    <DIR>          PythonProject
04/17/2023  06:36 PM    <DIR>          Saved Games
04/17/2023  06:36 PM    <DIR>          Searches
04/19/2023  09:20 PM    <DIR>          source
04/22/2023  07:49 AM               171 Test.csproj
05/22/2023  04:21 AM               190 tt.py
05/22/2023  04:18 AM               185 tt.txt
02/08/2023  02:14 PM               589 Untitled.ipynb
02/08/2023  02:34 PM             1,005 Untitled1.ipynb
02/11/2023  07:16 PM               965 Untitled2.ipynb
04/17/2023  06:36 PM    <DIR>          Videos
05/27/2023  09:54 AM    <DIR>          VirtualBox VMs
              12 File(s)     16,781,445 bytes
              35 Dir(s)  338,758,316,032 bytes free

C:\Users\Test>
```

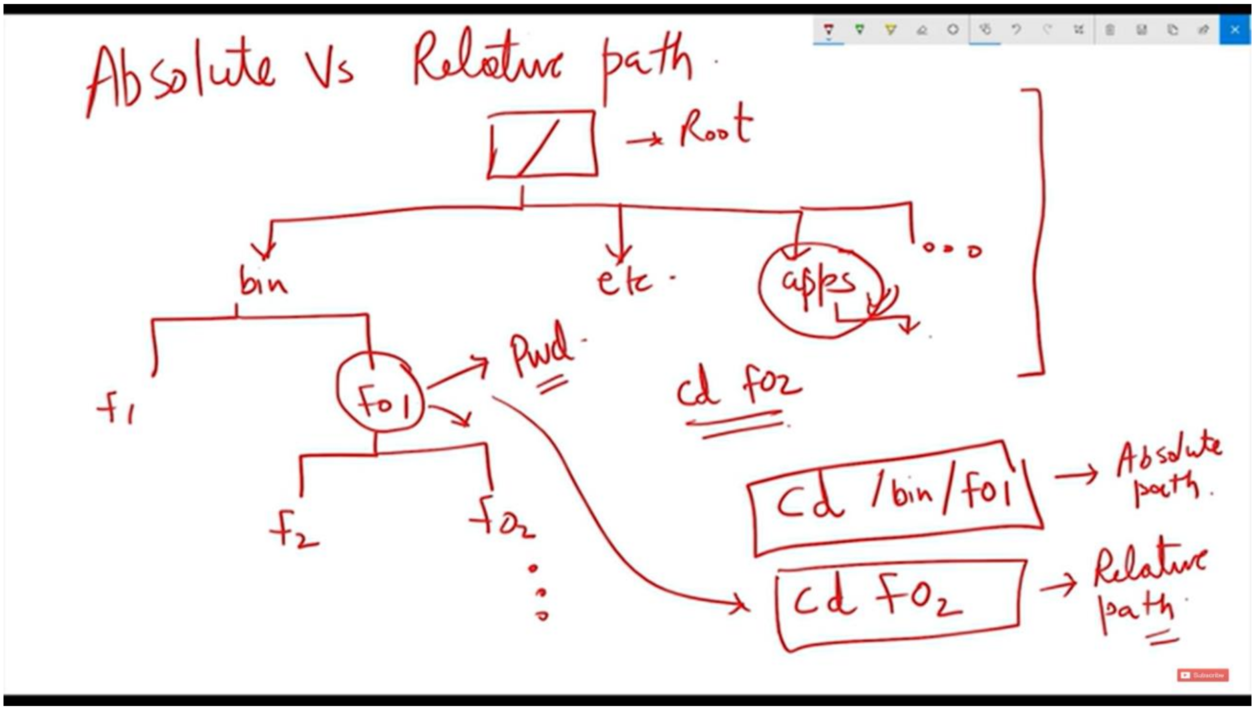
After invoking dir command we can see all the files and foler in the current directory named c:\Users\Test>

All the folders are indicated by <DIR> and the files are showin with file size and the filenames with extension. E.f., tt.py indicates this file is of python program, extnesion .py indicates so. File extension gives us some idea about the file type. The same information can be seen using GUI in Windows. Just clikc on the disk icon and see all the folders and files in the current directory as below:



| Name | Date modified | Type | Size |
|---|---|---|---|
| Searches | 4/17/2023 6:36 PM | File folder | |
| source | 4/19/2023 9:20 PM | File folder | |
| Videos | 4/17/2023 6:36 PM | File folder | |
| VirtualBox VMs | 5/27/2023 9:54 AM | File folder | |
| .condarc | 2/8/2023 1:27 PM | CONDARC File | 1 KB |
| AzureStorageEmulatorDb510.mdf | 4/22/2023 2:30 PM | MDF File | 8,192 KB |
| AzureStorageEmulatorDb510_log.ldf | 4/22/2023 2:30 PM | LDF File | 8,192 KB |
| kk | 2/11/2023 6:20 PM | Jupyter Source File | 1 KB |
| miktex-console.lock | 3/6/2023 9:55 PM | LOCK File | 1 KB |
| Program.cs | 4/22/2023 7:49 AM | C# Source File | 1 KB |
| Test.csproj | 4/22/2023 7:49 AM | C# Project File | 1 KB |
| tt | 5/22/2023 4:21 AM | Python Source File | 1 KB |
| tt | 5/22/2023 4:18 AM | Text Document | 1 KB |
| Untitled | 2/8/2023 1:14 PM | Jupyter Source File | 1 KB |
| Untitled1 | 2/8/2023 1:34 PM | Jupyter Source File | 1 KB |
| Untitled2 | 2/11/2023 6:16 PM | Jupyter Source File | 1 KB |

You can right click the mouse and create a new folder in the current directory easily. We must have a separate folder to manage the similar/related files. The file name should also be informative so that we can open the right file for the right information in the future when we need to work on that file.



```
cd fo2
cd /bin/fo1
cd fo2
```

Moreover, you need to keep track of of the file path to locate the particular file in the givne folder. E.g. If you are at bin folder and want to get to fo2 folder, then invoke cd fo1\fo2, and from fo2 to apps then you have to move to the rook first and start the journey from there. The command will be cd\apps.