Introduction to Operating Systems

(What is Operating System, What OS does, Structure of OS, Evolution of OS, Operating System Functions, Main Functions of OS, Types of OS)

What is OS?

- Operating System is a software, which makes a computer to actually work.
- It is the software the enables all the programs we use.
- The OS organizes and controls the hardware.
- OS acts as an interface between the application programs and the machine hardware.

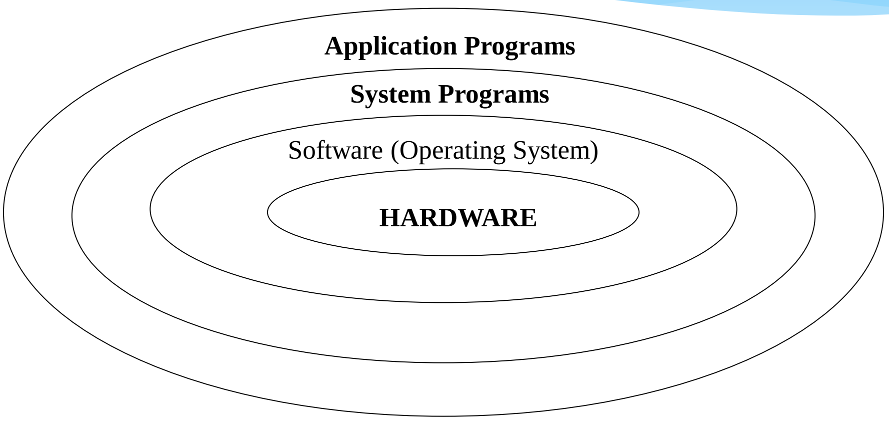
- Examples: Windows, Linux, Unix and Mac OS, etc.,

What OS does?

An operating system performs basic tasks such as,

- controlling and allocating memory,
- prioritizing system requests,
- controlling input and output devices,
- facilitating networking and
- managing file systems.

Structure of Operating System



Structure of Operating System

- The structure of OS consists of 4 layers:

1. **Hardware**

   Hardware consists of CPU, Main memory, I/O Devices, etc.,

2. **Software (Operating System)**

   Software includes process management routines, memory management routines, I/O control routines, file management routines.

3. **System programs**

   This layer consists of compilers, Assemblers, linker etc.

4. **Application programs**

   This is dependent on users need. Ex. Railway reservation system, Bank database management etc.,

Evolution of OS

- The evolution of operating systems went through seven *major phases.*

- Six of them significantly changed the ways in which users accessed computers through the open shop, batch processing, multiprogramming, timesharing, personal computing, and distributed systems.

- In the seventh phase the foundations of concurrent programming were developed and demonstrated in model operating systems.

Evolution of OS

Major Phases	Technical Innovations	Operating Systems
Open Shop	The idea of OS	IBM 701 open shop (1954)
Batch Processing	Tape batching, First-in, first-out scheduling.	BKS system (1961)
Multi-programming	Processor multiplexing, Indivisible operations, Demand paging, Input/output spooling, Priority scheduling, Remote job entry	Atlas supervisor (1961), Exec II system (1966)
Timesharing	Simultaneous user interaction, On-line file systems	Multics file system (1965) Unix (1974)
Concurrent Programming	Hierarchical systems, Extensible kernels, Parallel programming concepts, Secure parallel languages	RC 4000 system (1969), 13 Venus system (1972), 14 Boss 2 system (1975).
Personal Computing	Graphic user interfaces	OS 6 (1972) Pilot system (1980)
Distributed Systems	Remote servers	WFS file server (1979) Unix United RPC (1982) 24 Amoeba system (1990)

Batch Processing:

- In Batch processing same type of jobs batch *(BATCH- a set of jobs with similar needs)* together and execute at a time.
- The OS was simple, its major task was to transfer control from one job to the next.
- The job was submitted to the computer operator in form of punch cards. At some later time the output appeared.
- The OS was always resident in memory. (Ref. Fig. next slide)
- Common Input devices were card readers and tape drives.
- Common output devices were line printers, tape drives, and card punches.
- Users did not interact directly with the computer systems, but he prepared a job (comprising of the program, the data, & some control information).

Multiprogramming

- Multiprogramming is a technique to execute number of programs simultaneously by a single processor.
- In Multiprogramming, number of processes reside in main memory at a time.
- The OS picks and begins to executes one of the jobs in the main memory.
- If any I/O wait happened in a process, then CPU switches from that job to another job.
- Hence CPU in not idle at any time.

Multiprogramming

OS
Job 1
Job 2
Job 3
Job 4
Job 5

- Figure depicts the layout of multiprogramming system.
- The main memory consists of 5 jobs at a time, the CPU executes one by one.

Advantages:

- Efficient memory utilization
- Throughput increases
- CPU is never idle, so performance increases.

Time Sharing Systems

- Time sharing, or multitasking, is a logical extension of multiprogramming.
- Multiple jobs are executed by switching the CPU between them.
- In this, the CPU time is shared by different processes, so it is called as “Time sharing Systems”.
- Time slice is defined by the OS, for sharing CPU time between processes.
- Examples: Multics, Unix, etc.,

Operating Systems functions:

The main functions of operating systems are:

1. Program creation
2. Program execution
3. Input/Output operations
4. Error detection
5. Resource allocation
6. Accounting
7. protection

Types of OS:

Operating System can also be classified as,-

- **Single User Systems**

- **Multi User Systems**

Single User Systems:

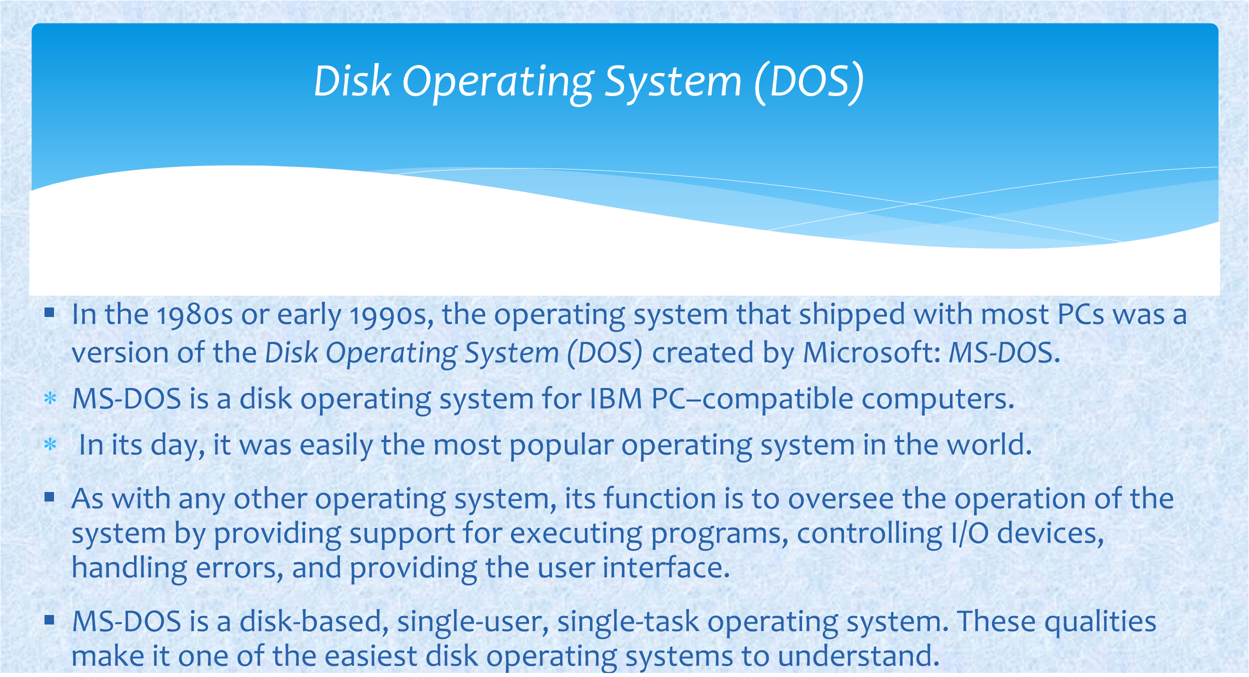
- Provides a platform for only one user at a time.

- They are popularly associated with Desk Top operating system which run on standalone systems where no user accounts are required.
- Example: DOS

Multi-User Systems:

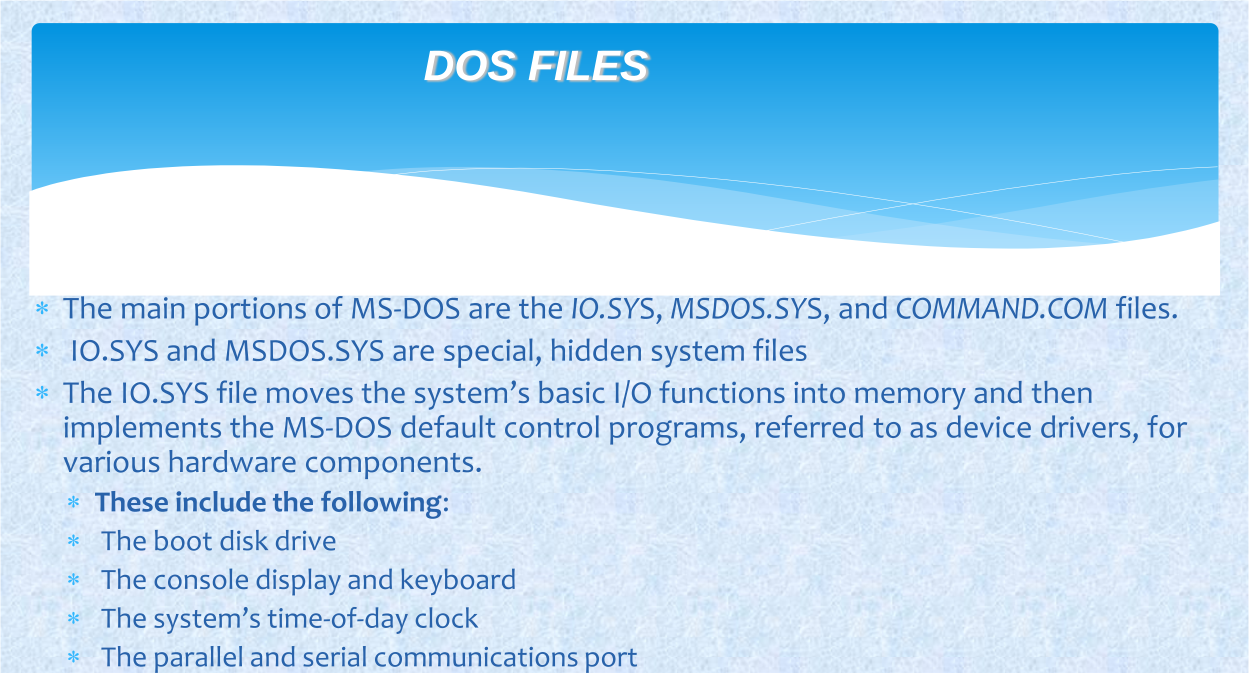
- Provides regulated access for a number of users by maintaining a database of known users.
- Refers to computer systems that support two or more simultaneous users.
- Another term for *multi-user* is *time sharing*.
- Ex: All mainframes are multi-user systems.
- Example: Unix

Disk Operating System (DOS)



Disk Operating System (DOS)

- In the 1980s or early 1990s, the operating system that shipped with most PCs was a version of the *Disk Operating System (DOS)* created by Microsoft: *MS-DOS*.
- MS-DOS is a disk operating system for IBM PC–compatible computers.
- In its day, it was easily the most popular operating system in the world.
- As with any other operating system, its function is to oversee the operation of the system by providing support for executing programs, controlling I/O devices, handling errors, and providing the user interface.
- MS-DOS is a disk-based, single-user, single-task operating system. These qualities make it one of the easiest disk operating systems to understand.



DOS FILES

- The main portions of MS-DOS are the *IO.SYS*, *MSDOS.SYS*, and *COMMAND.COM* files.
- IO.SYS and MSDOS.SYS are special, hidden system files
- The IO.SYS file moves the system's basic I/O functions into memory and then implements the MS-DOS default control programs, referred to as device drivers, for various hardware components.
  - **These include the following**:
  - The boot disk drive
  - The console display and keyboard
  - The system's time-of-day clock
  - The parallel and serial communications port

Popular DOS Versions

MS-DOS 3

- With DOS 3.0, released in summer 1984, Microsoft continued to include additional DOS features and to support more powerful hardware. DOS 3.0 supported hard drives larger than 10MB, as well as enhanced graphics formats.
- Three revisions—3.1, 3.2, and 3.3—provided additional innovations

MS-DOS 4

- By 1988 it was apparent that the wave of the future was the graphical interface, and DOS 4 provided users with the DOS Shell, a utility much like the Windows File Manager. Actually, DOS Shell was simply a scaled-down version of Windows that allowed users to manage files, run programs, and do routine maintenance, all from a single screen. The DOS Shell even supported a mouse.

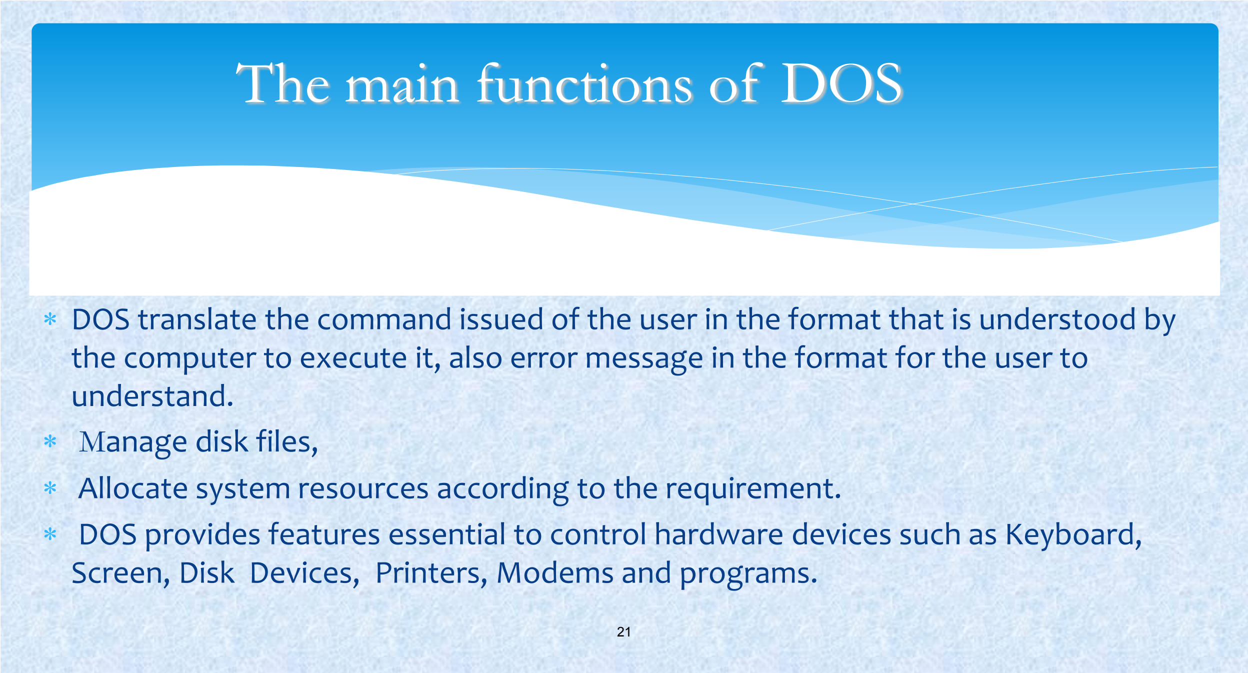
Popular DOS Versions

MS-DOS 5

- There were several important features introduced in the 1991 release of DOS 5.0. First of all, the ability to load drivers into reserved (upper) memory was a relief to those people who were constantly running out of conventional memory. This feature allowed more complex DOS programs. Several software utilities made their debut. The most commonly used utility introduced at this time was EDIT.COM. Also added were QBASIC.EXE, DOSKEY, UNFORMAT, and UNDELETE.

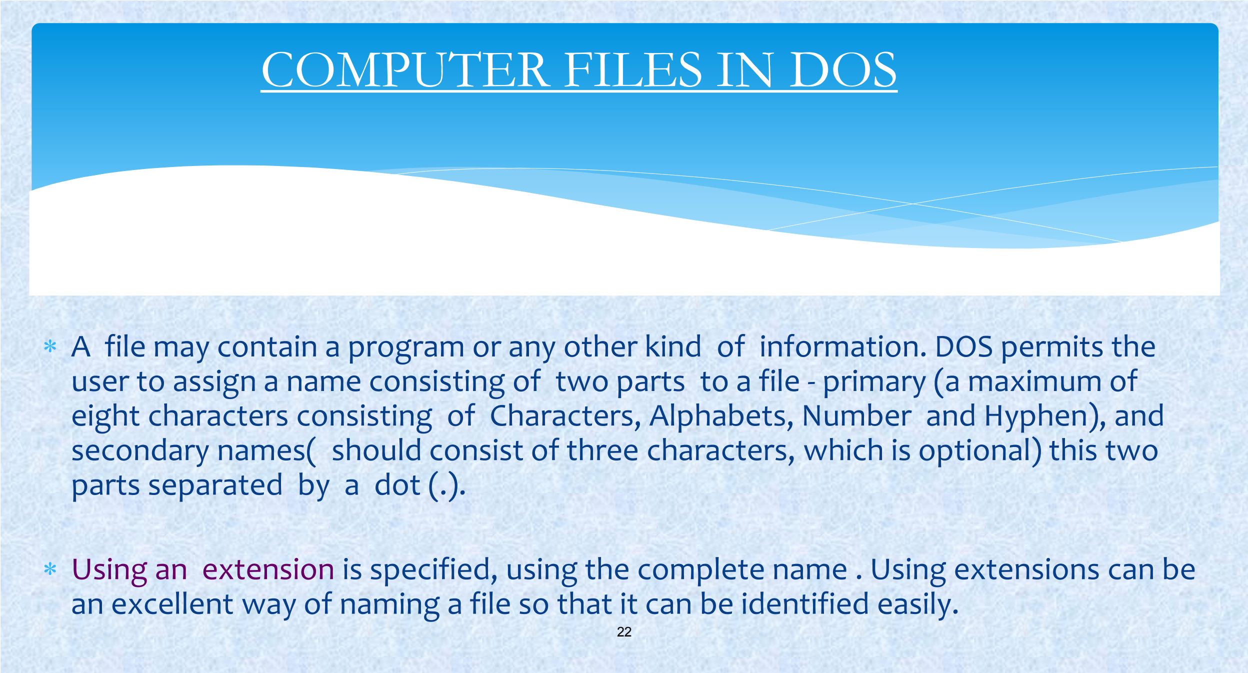
MS-DOS 6

- Released in 1993 to excellent sales, DOS 6.0 offered a number of new commands and configurable options. Another enhancement in DOS 6.0 was EMM386.EXE, which allowed the system to pool extended and expanded memory. DOS 6.0 has subsequently been revised a number of times. As of this date, DOS 6.22 is the most current MS-DOS version available as a stand-alone operating system.



The main functions of DOS

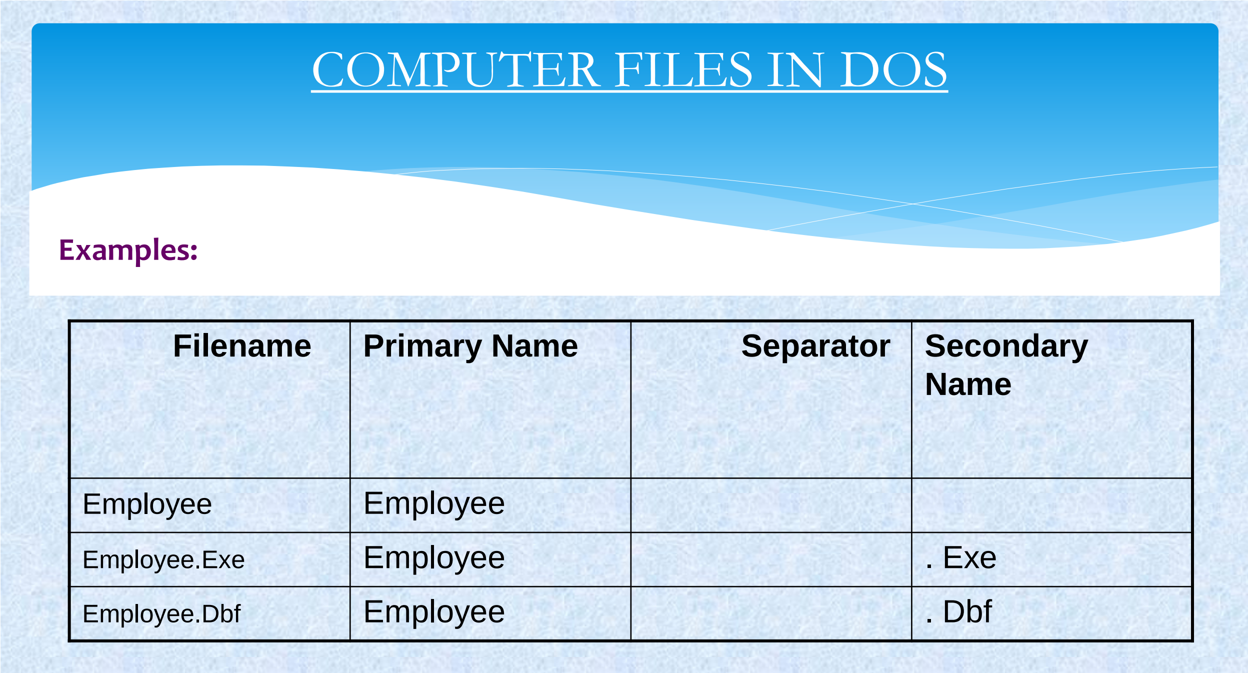
- DOS translate the command issued of the user in the format that is understood by the computer to execute it, also error message in the format for the user to understand.
- Manage disk files,
- Allocate system resources according to the requirement.
- DOS provides features essential to control hardware devices such as Keyboard, Screen, Disk Devices, Printers, Modems and programs.



COMPUTER FILES IN DOS

- A file may contain a program or any other kind of information. DOS permits the user to assign a name consisting of two parts to a file - primary (a maximum of eight characters consisting of Characters, Alphabets, Number and Hyphen), and secondary names(should consist of three characters, which is optional) this two parts separated by a dot (.).

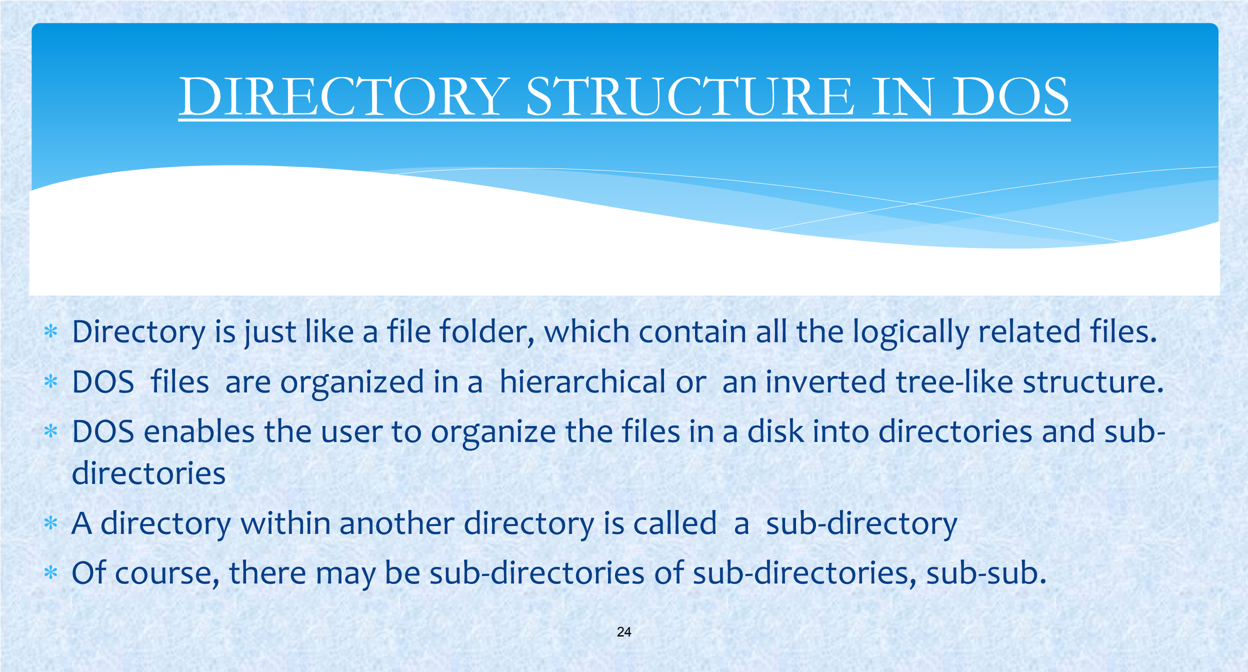
- Using an extension is specified, using the complete name . Using extensions can be an excellent way of naming a file so that it can be identified easily.



COMPUTER FILES IN DOS

Examples:

Filename	Primary Name	Separator	Secondary Name
Employee	Employee		
Employee.Exe	Employee		. Exe
Employee.Dbf	Employee		. Dbf



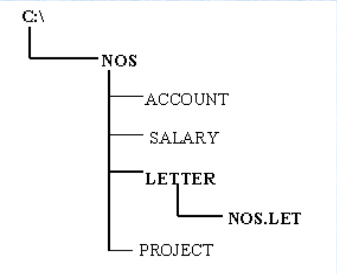
DIRECTORY STRUCTURE IN DOS

- Directory is just like a file folder, which contain all the logically related files.
- DOS files are organized in a hierarchical or an inverted tree-like structure.
- DOS enables the user to organize the files in a disk into directories and sub-directories
- A directory within another directory is called a sub-directory
- Of course, there may be sub-directories of sub-directories, sub-sub.

USING PATH TO SPECIFY THE LOCATION OF FILES

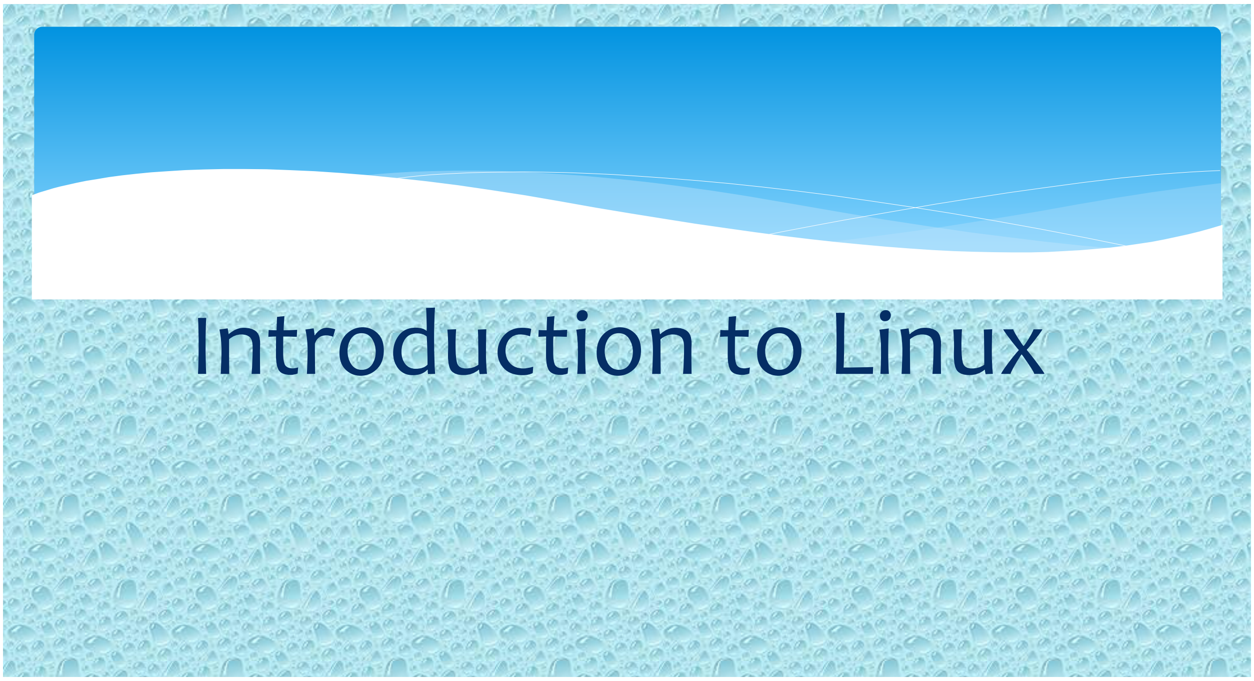
- A path is the route that leads from the root directory of a drive to the file you want to use.
- **For example** , to access **C:\NOS\LETTER\NOS.LET**
- **Where** (C:) represent the drive the file is on , 1st (\) is the root directory. 2 nd (\) separates the NOS directory from the LETTER sub-directory .

  3rd (\) separates the LETTER sub-directories from the file name, NOS.LET.

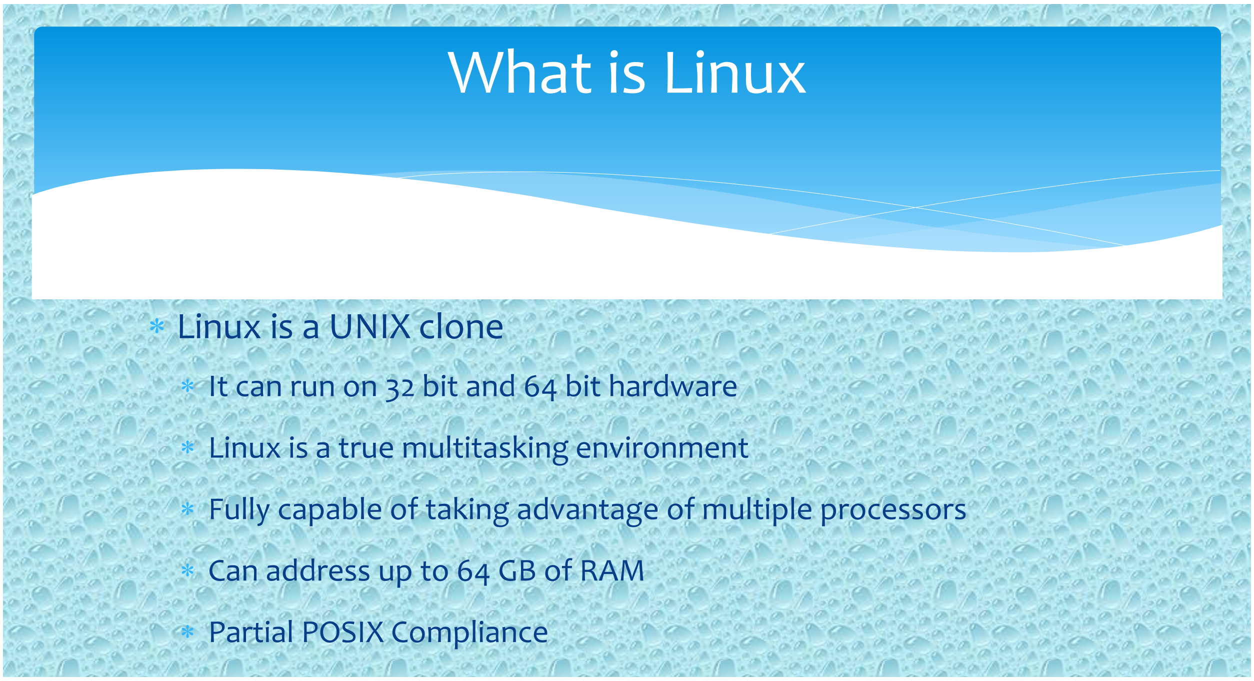


Internal File COMMANDS

1. Dir
2. Rename (Ren)
3. Delete (Del)
4. Copy
5. Type
7. Set Attributes (attrib)

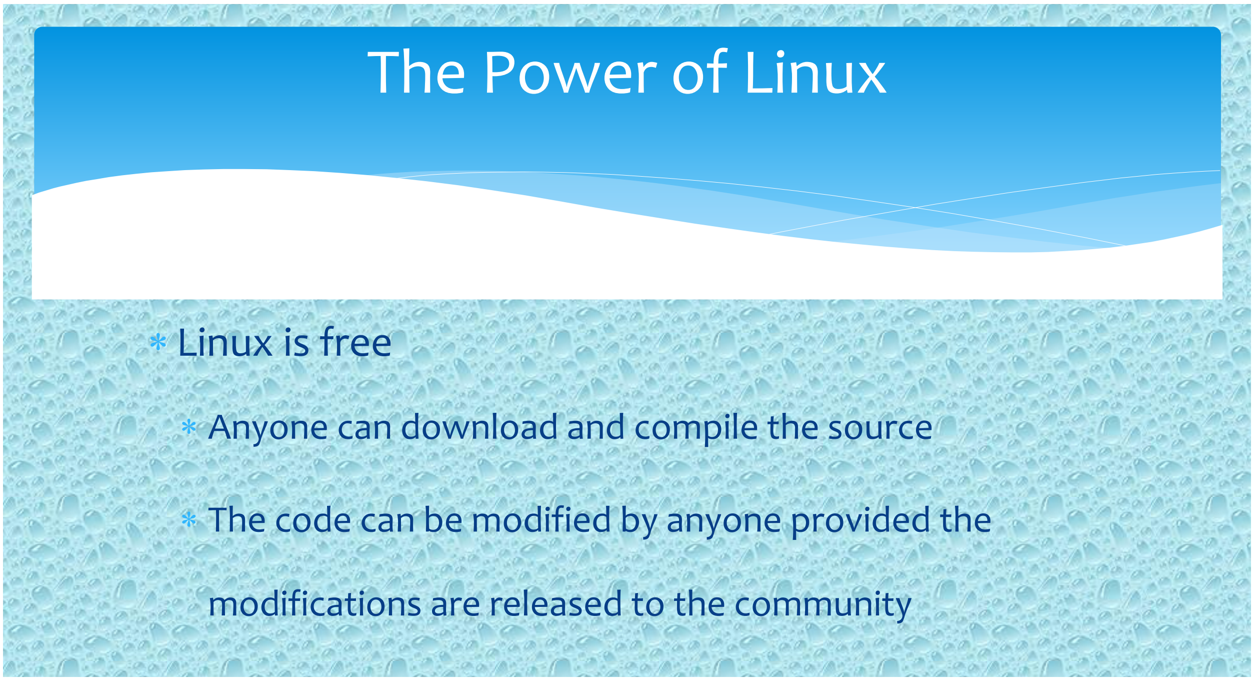


Introduction to Linux



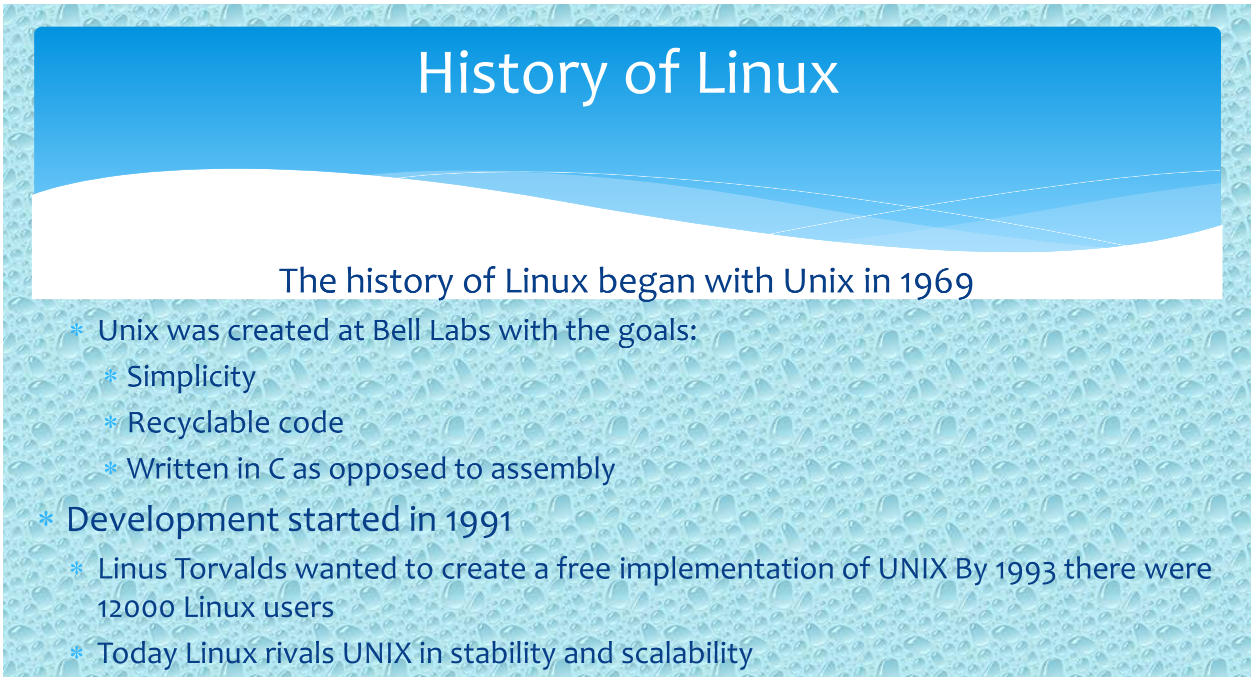
What is Linux

- Linux is a UNIX clone
  - It can run on 32 bit and 64 bit hardware
  - Linux is a true multitasking environment
  - Fully capable of taking advantage of multiple processors
  - Can address up to 64 GB of RAM
  - Partial POSIX Compliance



The Power of Linux

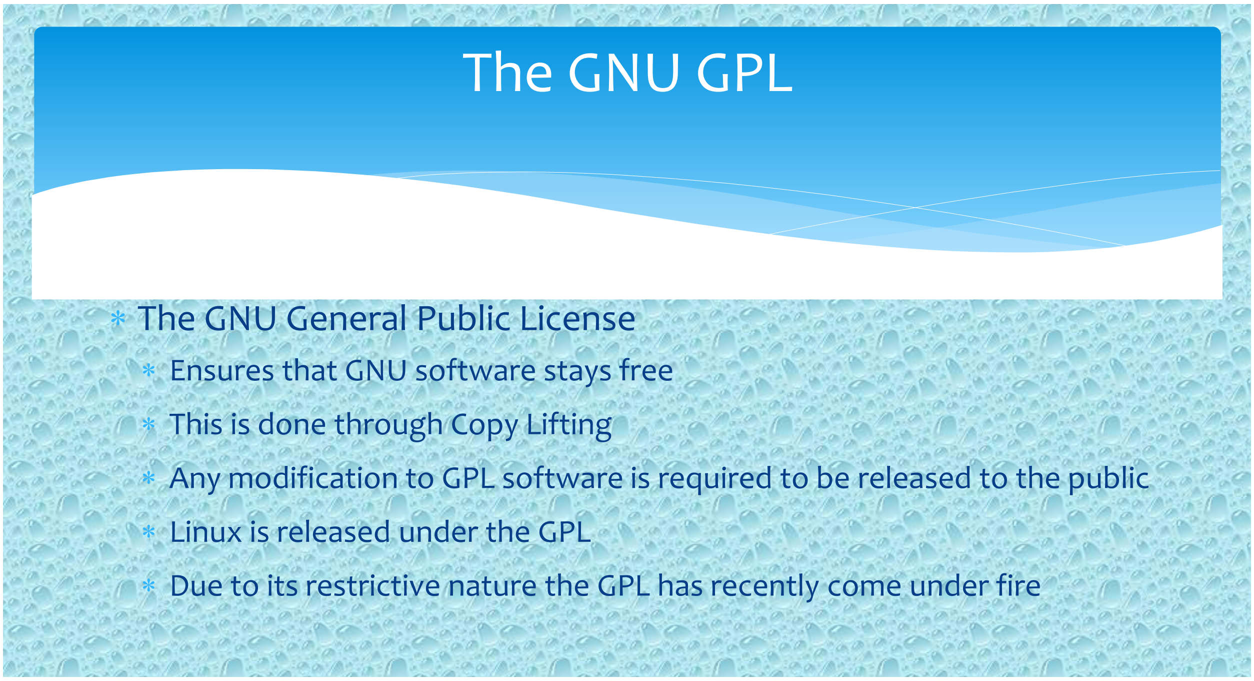
- Linux is free
  - Anyone can download and compile the source
  - The code can be modified by anyone provided the modifications are released to the community



History of Linux

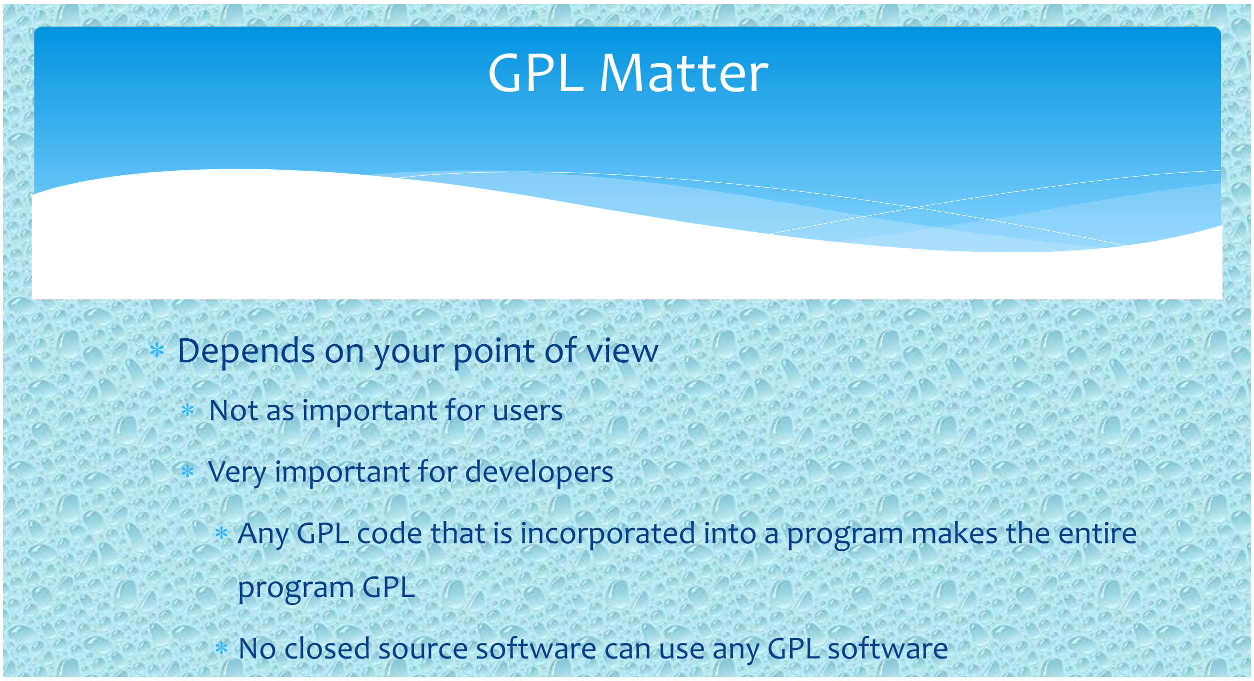
The history of Linux began with Unix in 1969

- Unix was created at Bell Labs with the goals:
  - Simplicity
  - Recyclable code
  - Written in C as opposed to assembly
- Development started in 1991
  - Linus Torvalds wanted to create a free implementation of UNIX By 1993 there were 12000 Linux users
  - Today Linux rivals UNIX in stability and scalability



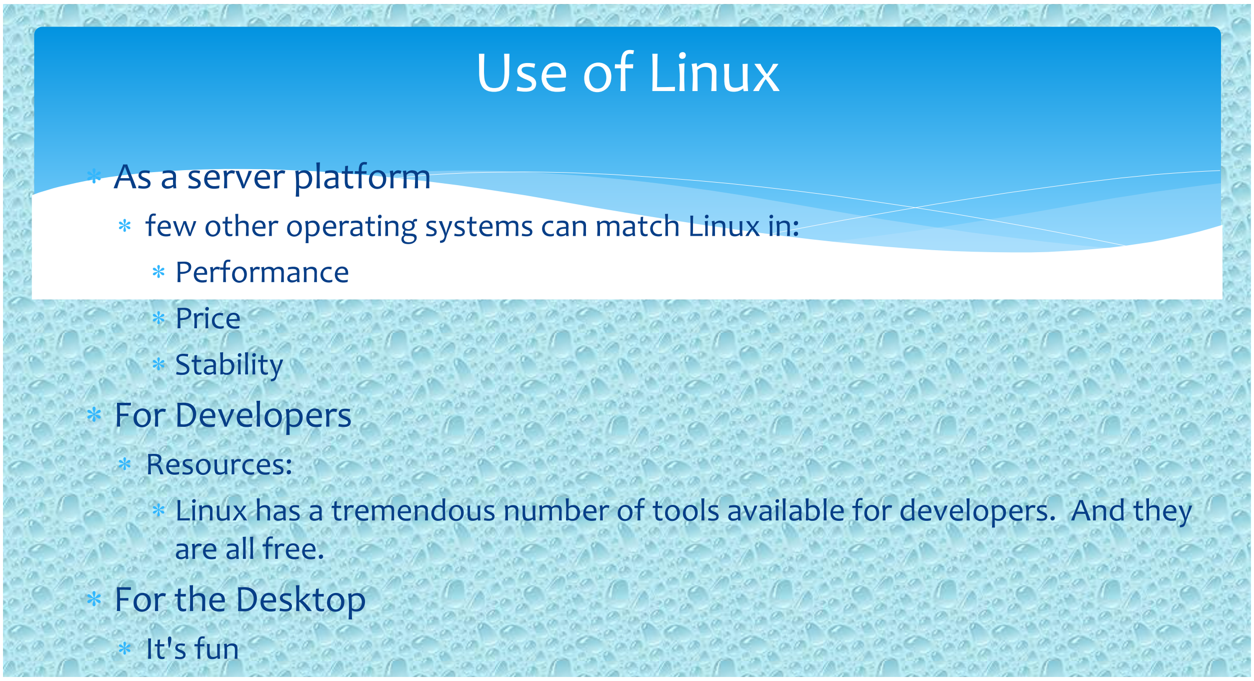
The GNU GPL

- The GNU General Public License
  - Ensures that GNU software stays free
  - This is done through Copy Lifting
  - Any modification to GPL software is required to be released to the public
  - Linux is released under the GPL
  - Due to its restrictive nature the GPL has recently come under fire



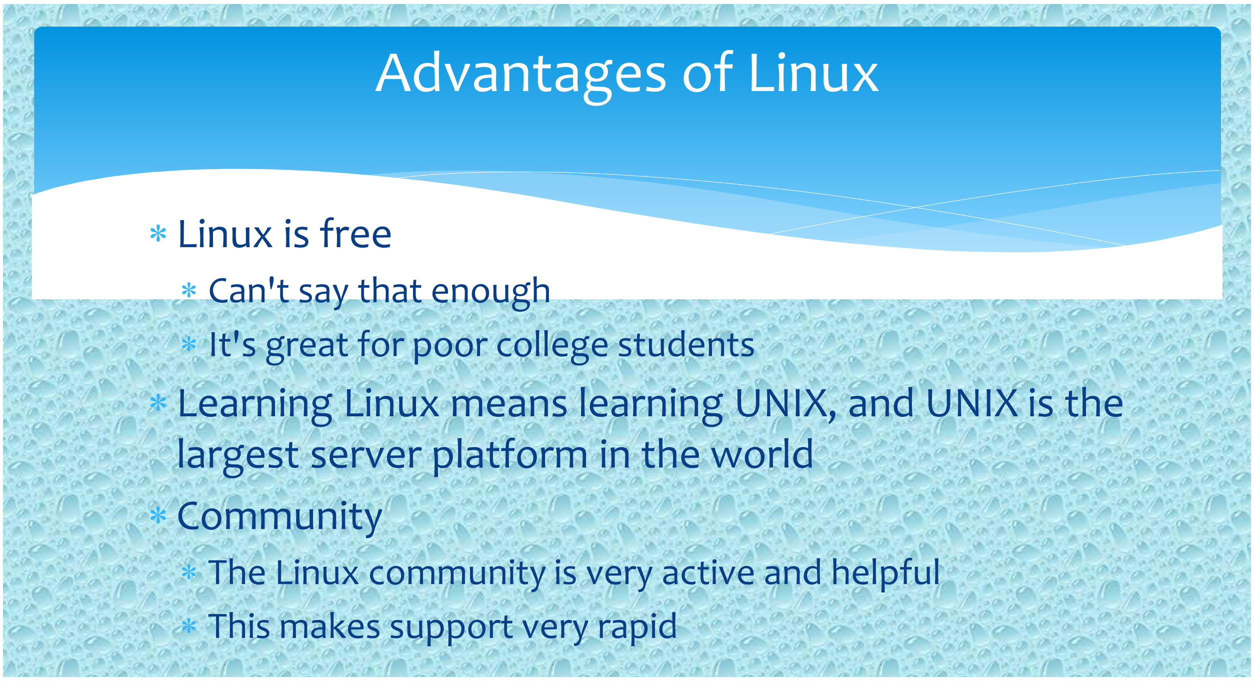
GPL Matter

- Depends on your point of view
  - Not as important for users
  - Very important for developers
    - Any GPL code that is incorporated into a program makes the entire program GPL
    - No closed source software can use any GPL software



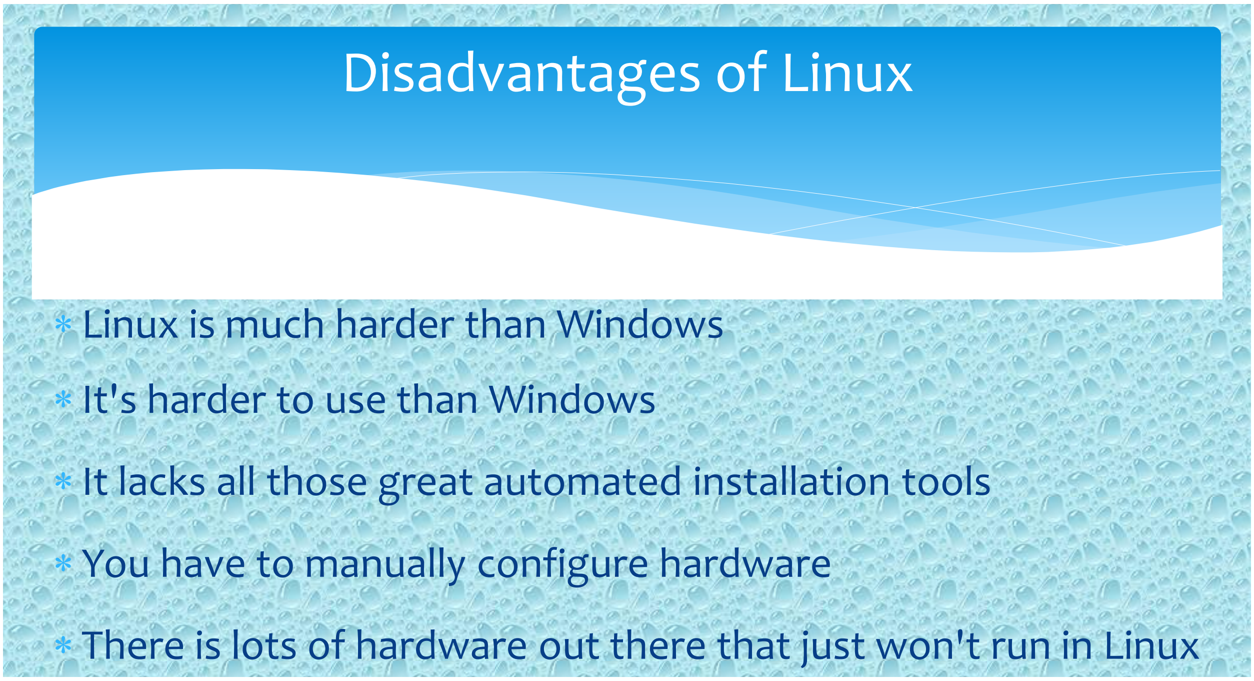
Use of Linux

- As a server platform
  - few other operating systems can match Linux in:
    - Performance
    - Price
    - Stability
- For Developers
  - Resources:
    - Linux has a tremendous number of tools available for developers. And they are all free.
- For the Desktop
  - It's fun



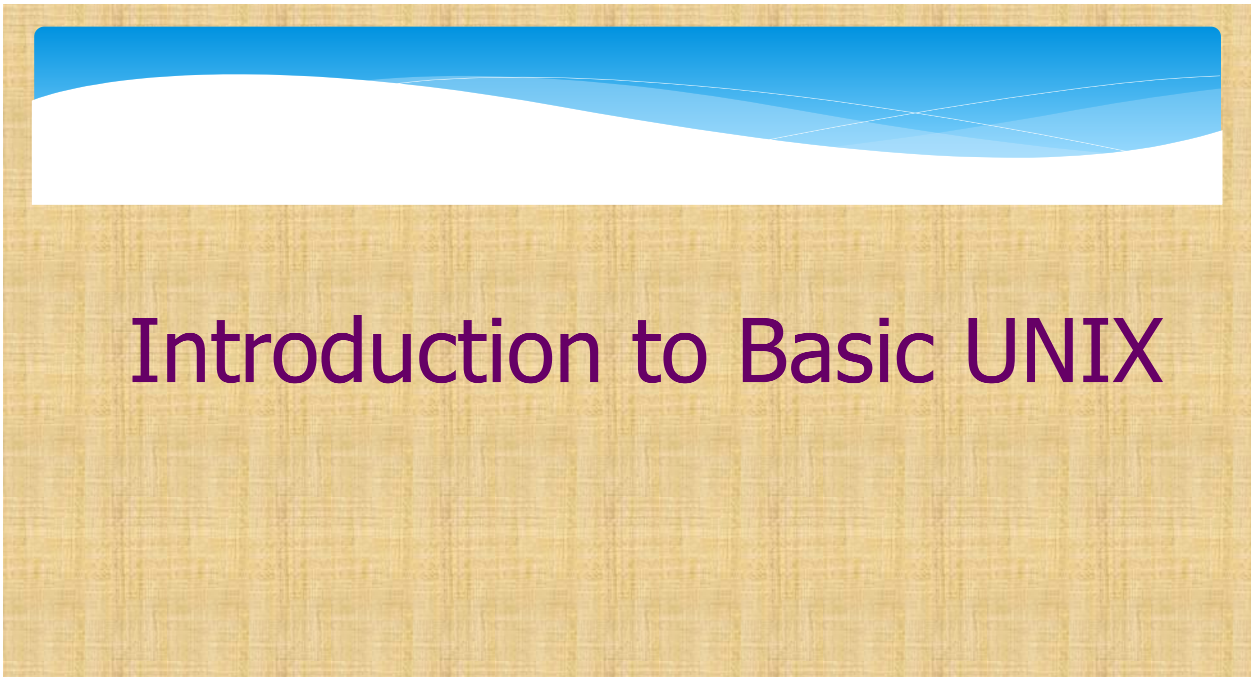
Advantages of Linux

- Linux is free
  - Can't say that enough
  - It's great for poor college students
- Learning Linux means learning UNIX, and UNIX is the largest server platform in the world
- Community
  - The Linux community is very active and helpful
  - This makes support very rapid

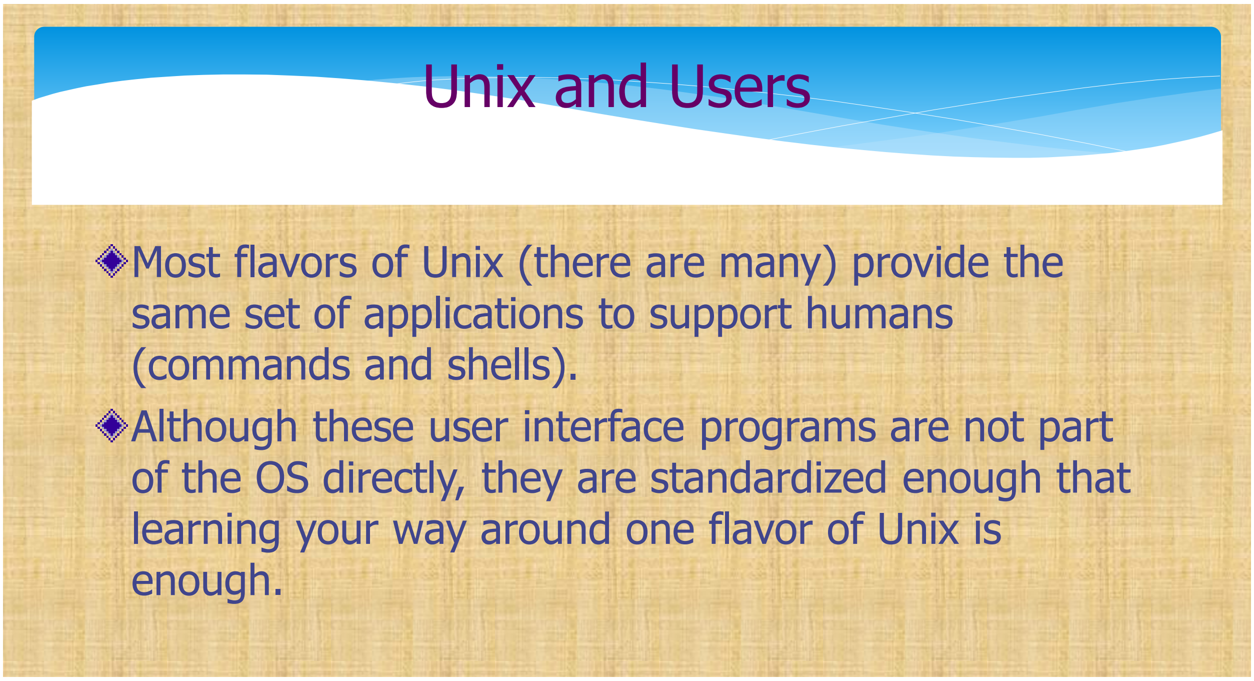


Disadvantages of Linux

- Linux is much harder than Windows
- It's harder to use than Windows
- It lacks all those great automated installation tools
- You have to manually configure hardware
- There is lots of hardware out there that just won't run in Linux

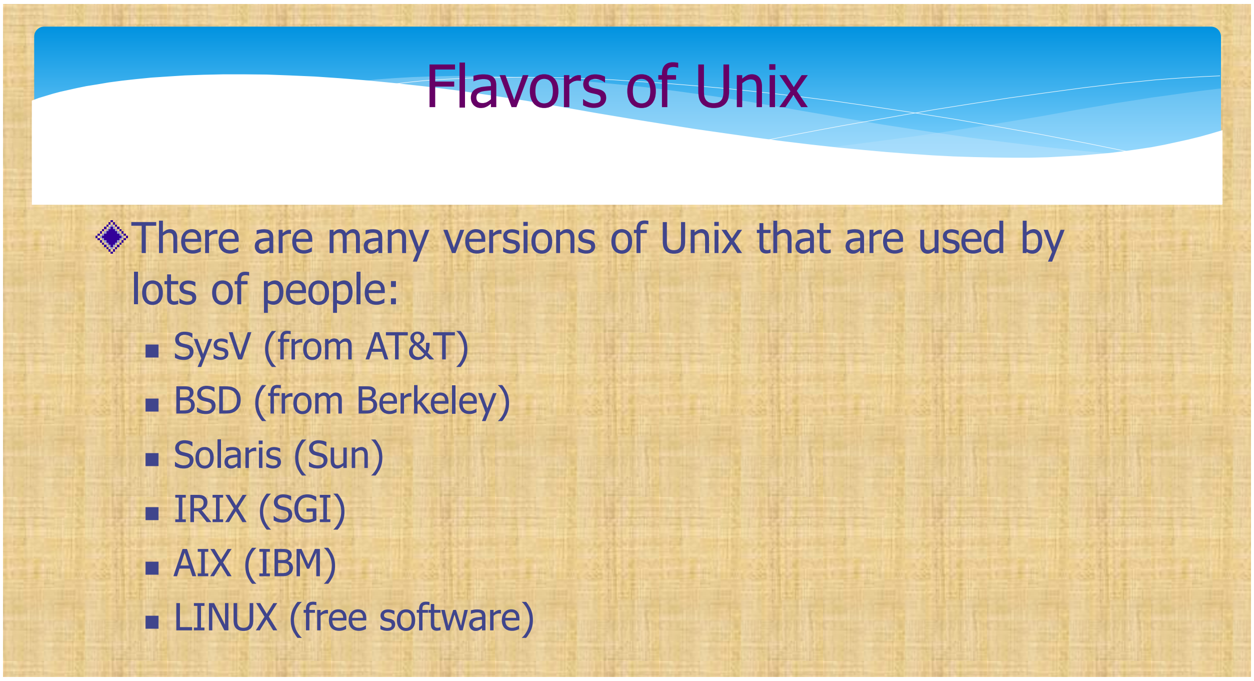


Introduction to Basic UNIX



Unix and Users

- Most flavors of Unix (there are many) provide the same set of applications to support humans (commands and shells).
- Although these user interface programs are not part of the OS directly, they are standardized enough that learning your way around one flavor of Unix is enough.



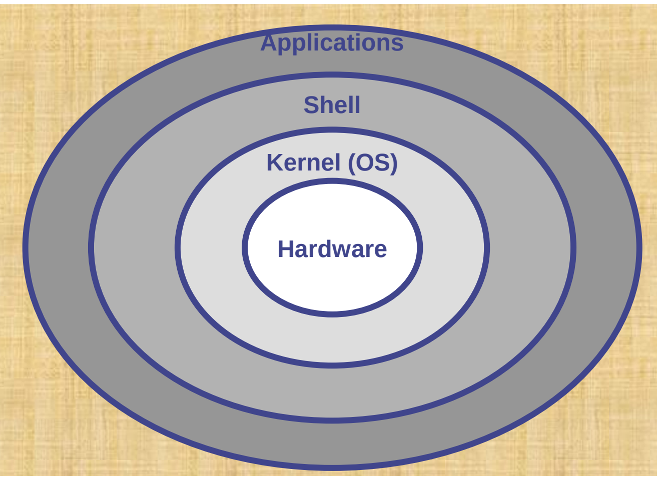
Flavors of Unix

- There are many versions of Unix that are used by lots of people:
  - SysV (from AT&T)
  - BSD (from Berkeley)
  - Solaris (Sun)
  - IRIX (SGI)
  - AIX (IBM)
  - LINUX (free software)

The power of Unix

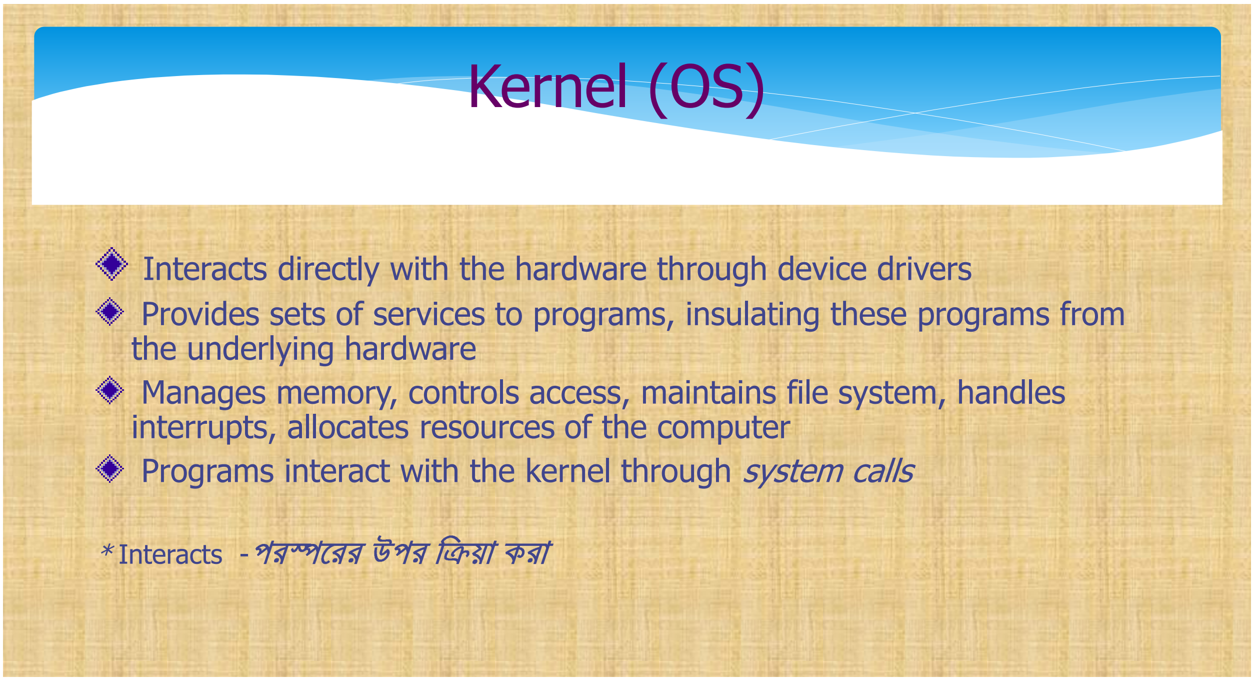
- Open source, portability
- You can extend the basic functionality of Unix:
  - customize the shell and user interface.
  - string together a series of Unix commands to create new functionality.
  - create custom commands that do exactly what we want.

Structure of the UNIX system



There are many standard applications:

- file system commands
- text editors
- compilers
- text processing



Kernel (OS)

- Interacts directly with the hardware through device drivers
- Provides sets of services to programs, insulating these programs from the underlying hardware
- Manages memory, controls access, maintains file system, handles interrupts, allocates resources of the computer
- Programs interact with the kernel through *system calls*

* Interacts - *পরস্পরের উপর ক্রিয়া করা*