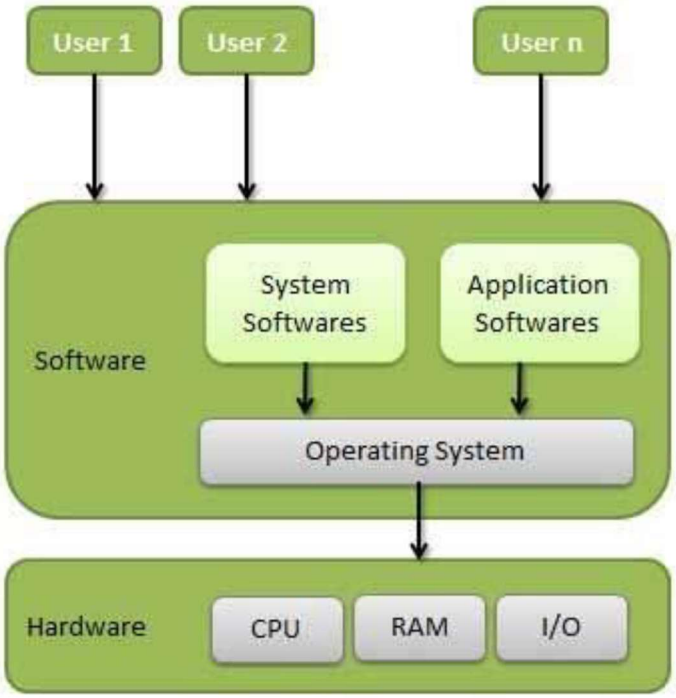
# Operating System

An Operating System (OS) is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.
Some popular Operating Systems include Linux Operating System, Windows Operating System, VMS, OS/400, AIX, z/OS, etc

## Definition

An operating system is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs.



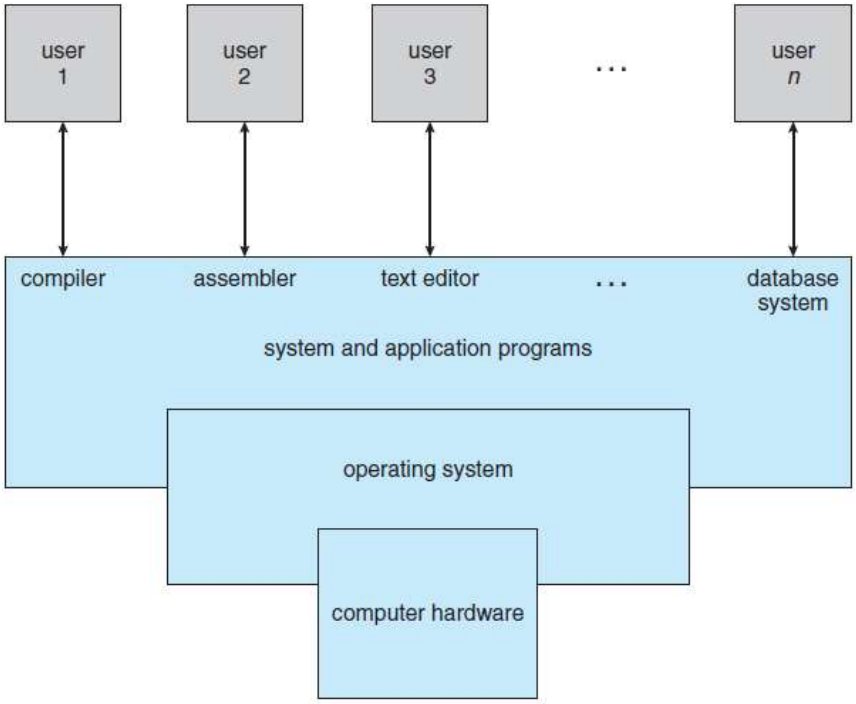



Figure 1.1 Abstract view of the components of a computer system

A computer system can be divided roughly into four components: the hardware, the operating system, the application programs, and the users (Figure 1.1).
The hardware—the central processing unit (CPU), the memory, and the input/output (I/O) devices—provides the basic computing resources for the system. The application programs—such as word processors, spreadsheets, compilers, and Web browsers.

Following are some of important functions of an operating System.

1. Memory Management
2. Processor Management
3. Device Management
4. File Management
5. Security
6. Control over system performance
7. Job accounting
8. Error detecting aids
9. Coordination between other software and users

## Memory Management

Memory management refers to management of Primary Memory or Main Memory. Main memory is a large array of words or bytes where each word or byte has its own address.
Main memory provides a fast storage that can be accessed directly by the CPU. For a program to be executed, it must in the main memory. An Operating System does the following activities for memory management:

1. Keeps tracks of primary memory, i.e., what part of it are in use by whom, what part are not in use.
2. In multiprogramming, the OS decides which process will get memory when and how much.
3. Allocates the memory when a process requests it to do so.
4. De-allocates the memory when a process no longer needs it or has been terminated.

## Processor Management

In multiprogramming environment, the OS decides which process gets the processor when and for how much time. This function is called process scheduling.
An Operating System does the following activities for processor management:

1. Keeps tracks of processor and status of process. The program responsible for this task is known as traffic controller.
2. Allocates the processor (CPU) to a process.
3. De-allocates processor when a process is no longer required.

## Device Management

An Operating System manages device communication via their respective drivers.
It does the following activities for device management:

1. Keeps tracks of all devices. The program responsible for this task is known as the I/O controller.
2. Decides which process gets the device when and for how much time.
3. Allocates the device in the most efficient way.
4. De-allocates devices.

## File Management

A file system is normally organized into directories for easy navigation and usage. These directories may contain files and other directions.

An Operating System does the following activities for file management:

1. Keeps track of information, location, uses, status etc. The collective facilities are often known as file system.
2. Decides who gets the resources.
3. Allocates the resources.
4. De-allocates the resources.
5. Other Important Activities
6. Following are some of the important activities that an Operating System performs:
7. Security -- By means of password and similar other techniques, it prevents unauthorized access to programs and data.
8. Control over system performance -- Recording delays between request for a service and response from the system.
9. Job accounting -- Keeping track of time and resources used by various jobs and users.
10. Error detecting aids -- Production of dumps, traces, error messages, and other debugging and error detecting aids.
11. Coordination between other software and users -- Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems.

## Word definition

A program loaded into memory and executing is called a process. When a process executes, it typically executes for only a short time before it either finishes or needs to perform I/O.
If several jobs are ready to be brought into memory, and if there is not enough room for all of them, then the system must choose among them. Making this decision involves job scheduling
A trap (or an exception) is a software-generated interrupt caused either by an error (for example, division by zero or invalid memory access) or by a specific request from a user program that an operating-system service be performed.

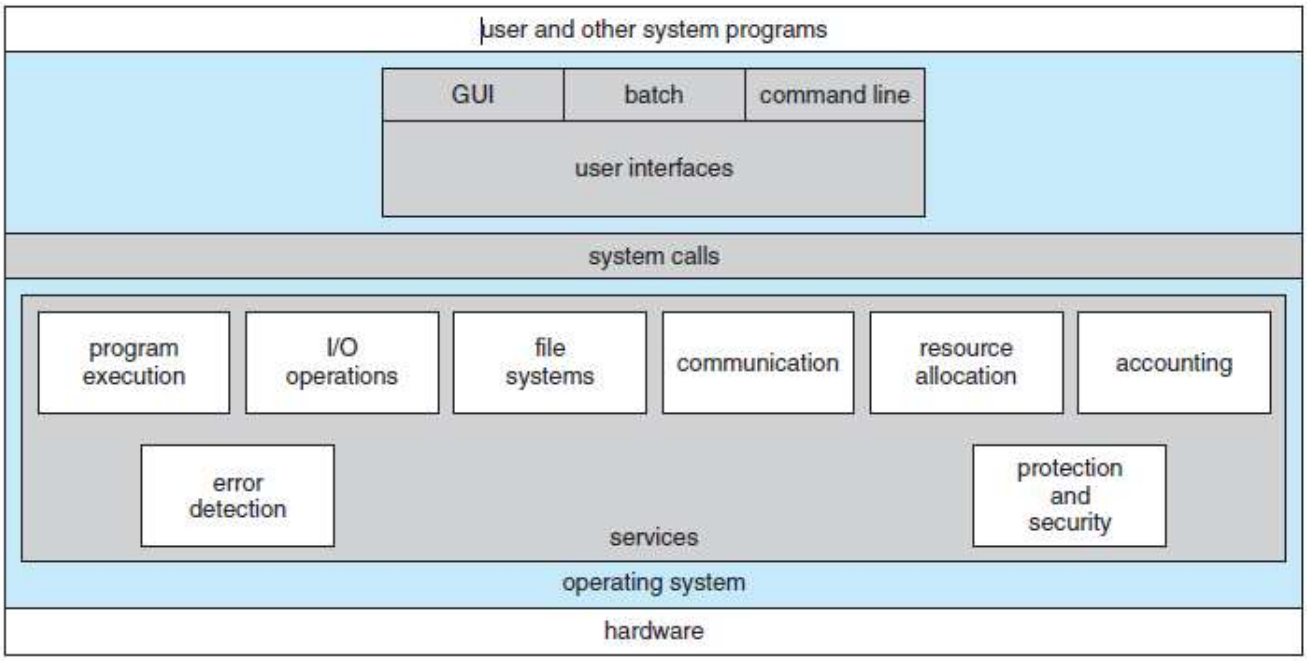



Figure 2.1 A view of operating system services.

# Processes

A question that arises in discussing operating systems involves what to call all the CPU activities. A batch system executes jobs, whereas a time-shared system has user programs, or tasks. Even on a single-user system, a user may be able to run several programs at one time: a word processor, a Web browser, and an e-mail package. And even if a user can execute only one program at a time, such as on an embedded device that does not support multitasking, the operating system may need to support its own internal programmed activities, such as memory management. In many respects, all these activities are similar, so we call all of them processes.

Informally, as mentioned earlier, **a process** is a program in execution. A process is more than the program code, which is sometimes known as the *text* section. It also includes the current activity, as represented by the value of the program counter and the contents of the processor's registers. A process generally also includes the process *stack*, which contains temporary data (such as function parameters, return addresses, and local variables), and a *data section*, which contains global variables. A process may also include a *heap*, which is memory that is dynamically allocated during process run time. The structure of a process in memory is shown in Figure 3.1.

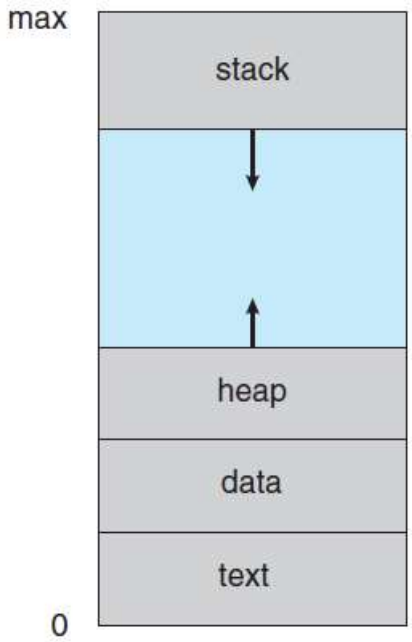



Figure 3.1 Process in memory

A system consists of a collection of processes: operating system processes executing system code and user processes executing user code. Potentially, all these processes can execute concurrently, with the CPU (or CPUs) multiplexed among them. By switching the CPU between processes, the operating system can make the computer more productive.

## Process State

As a process executes, it changes state. The state of a process is defined in part by the current activity of that process. A process may be in one of the following states:

- **New.** The process is being created.
- **Running**. Instructions are being executed.
- **Waiting.** The process is waiting for some event to occur (such as an I/O completion or reception of a signal).
- **Ready**. The process is waiting to be assigned to a processor.
- **Terminated**. The process has finished execution.

These names are arbitrary, and they vary across operating systems. The states that they represent are found on all systems, however. Certain operating systems also more finely delineate process states. It is important to realize that only one process can be running on any processor at any instant. Many processes may be ready and waiting, however. The state diagram corresponding to these states is presented in Figure 3.2.

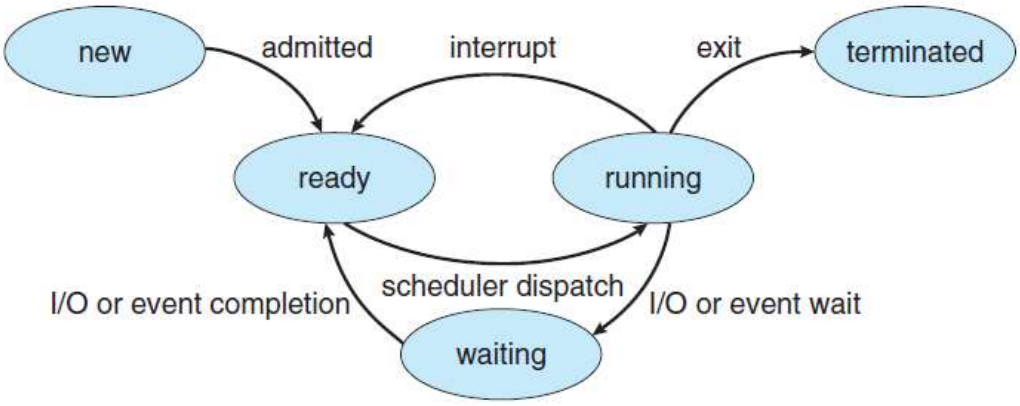



Figure 3.2 Diagram of process state.