

# OPERATING SYSTEMS

## INTRODUCTION

### OPERATING SYSTEM DEFINITIONS

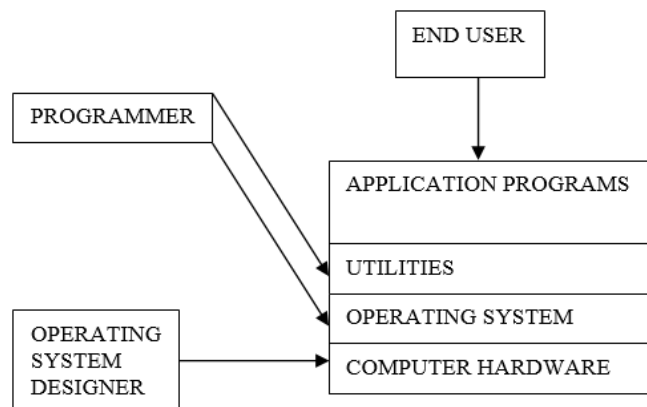
1. An operating system is program which acts as an interface between a user and the computer.
2. The operating system controls and coordinates the use of the hardware among various application programs for various users.
3. An operating system is similar to a government, the basic resources of a computer are provided by its hardware, software and data. The operating system provides the proper use of these resources.
4. An operating system is a resource allocator.
5. The operating system acts as the manager of the resources.
6. The primary goal of an operating system is to make the computer system convenient to use. Its secondary goal is to use the computer hardware in an efficient manner.

- |       |                                       |
|-------|---------------------------------------|
| Goals | 1. Convenience for the user           |
|       | 2. Efficient operation                |
|       | 3. Optimal use of computing resources |

To summarize:

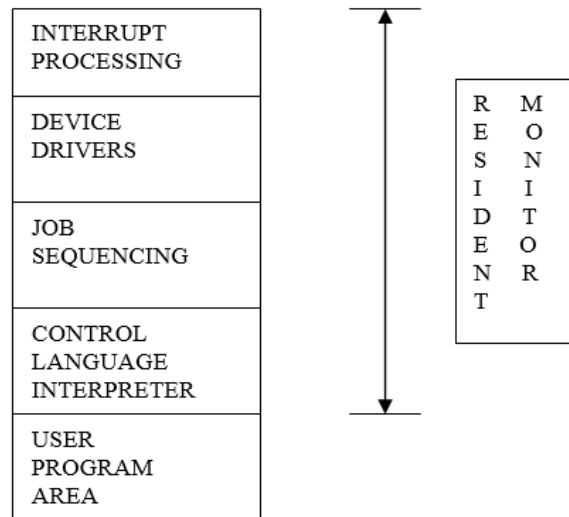
The common functions of controlling and allocating resources when brought together in piece of software may be called '*operating system*'.

## OPERATING SYSTEM AS A USER / COMPUTER INTERFACE



The user of an application is called the end user and generally is not concerned with the computer architecture. The end user views a computer system in terms of an application. The application can be expressed in a programming language and is developed by an application programmer. If one were to develop an application program as a set of machine instructions that is completely responsible for controlling the computer hardware, one would be faced with an overwhelmingly complex task. To ease this task a set of systems program is provided, some of these programs are referred to as Utilities and they implement frequently used functions that assist in program creation, the management of files, and the control of I/O devices. A programmer makes use of these facilities in developing an application, and the application, while it is running invokes the utilities to perform certain functions. The most important system program is the OS. The operating system masks the details of the hardware from the programmer and provides the programmer with a convenient interface for using the system. It acts as mediator, making it easier for the programmer and for application programs to access and use those facilities and services.

## RESIDENT MONITOR



With the earliest computers, from the late 1940s to the mid-1950s, the programmer interacted directly with the computer hardware, there was no operating system

These early systems presented two main problems:

1. **SCHEDULING:** Most installations used a sign-up sheet to reserve machine time. Typically, the user could sign up for a block of time in multiples of a half-hour or so. User might sign up for hour and finish in 45 minutes; this would result in wasted computer idle time.
2. **SETUP TIME:** A single program, called a **job**, could involve loading the compiler plus the high-level language program (source program), into memory saving the compiled program (object program), and then loading and linking together the object program and common functions. Each of these steps could involve mounting or dismounting tapes or setting up card decks. A considerable amount of time was spent just in setting up the program to run. This mode of operation could be termed **SERIAL PROCESSING**

### SIMPLE BATCH SYSTEMs

Early machines were very expensive and therefore it was important to maximize machine use. The wasted time caused by scheduling and setup

time was unacceptable. To improve use, the concept of a batch operating system was developed. The first batch operating system (and the first operating system of any kind) was developed in the mid-1950s by General Motors for use on an IBM 701.

The central idea behind the simple batch processing scheme was the use of a piece of software known as the *MONITOR*, with the use of this type of operating system, the user no longer has direct access to the machine.

It should be clear that the monitor handles the scheduling problem, batch jobs is queued up, and jobs are executed as rapidly as possible, with no intervening idle time. How about the job setup? The monitor handles this as well; with each job, instructions are included in a primitive form of JOB CONTROL LANGUAGE (JCL), which is special type of programming language used to provide instructions to the monitor.