CH-8

Operating System support

8.1 Operating system overview

(1) Operating system →

An operating system is a program that acts as a communication bridge (interface) between the user & computer hardware.

The purpose (objective) of an operating system is to provide a platform on which a user can execute programs conveniently & efficiently.

Ex → Linux os, Windows os, VMS, os/400, z/os etc.

(2) Aspects (functim) of OS :

There are two aspects of an OS

(I) The operating system as a user/computer interface

(II) The operating system as resource manager.

★ **(I) The OS as a user/computer interface →**

The user is not concerned with the computer's architecture. The OS provides the user, a convenient interface for using the system.

The following services provided by an OS:

1. program creation
2. program execution
3. Access to I/O devices
4. Controlled access to files.
5. System access.
6. Error detection & response.
7. Accounting.

① Program Creating →

The operating system provides a variety of facilities & services such as editors & debuggers to assist the user in creating program.

(2) Program execution →

The operating system provides a service like program execution. For program execution, it needs no of steps, like instruction & data must be loaded into main memory, I/O devices, & files must be initialized, & other resources must be prepared.

(3) Access to I/O devices →

Each I/O device requires its own specific set of instructions for operation. The OS takes care about it, so that the user can read & write the I/O devices.

(4) Controlled access to files →

With multiple simultaneous users, the OS can provide protection mechanism to control access to the files.

(5) System access →

In case of a shared or public system, the OS controls access to the system as a whole & to specific system resources.

(5) Error detection & response →

A varity of errors can occur while a computer system runs. These include internal & external hardware errors such as memory error, or a device malfunctim or failure, & various software error such as arithmetic overflow. These types of errors can be detected by the OS & also repoted to the applicatim.

(6) Accounting →

The OS collects the statistics for various resources & monitors performance such as response time.

Interfaces in a typical computer system:

(1) Instructim Set Architecture (ISA)

(2) Applicatim Binary Interface (ABI)

(3) Applicatim Programming Interface (API)

(i) THE OS AS A USER/COMPUTER INTERFACE

- ✓The end user is not concerned with the computer's architecture.
- ✓The application is expressed in a programming language.
- ✓Programs are referred as UTILITES.
- ✓The most important system program is the OS.
- ✓OS masks the details of the hardware from the programmer.
- ✓OS provides the programmer (user) a convenient interface for using the system.

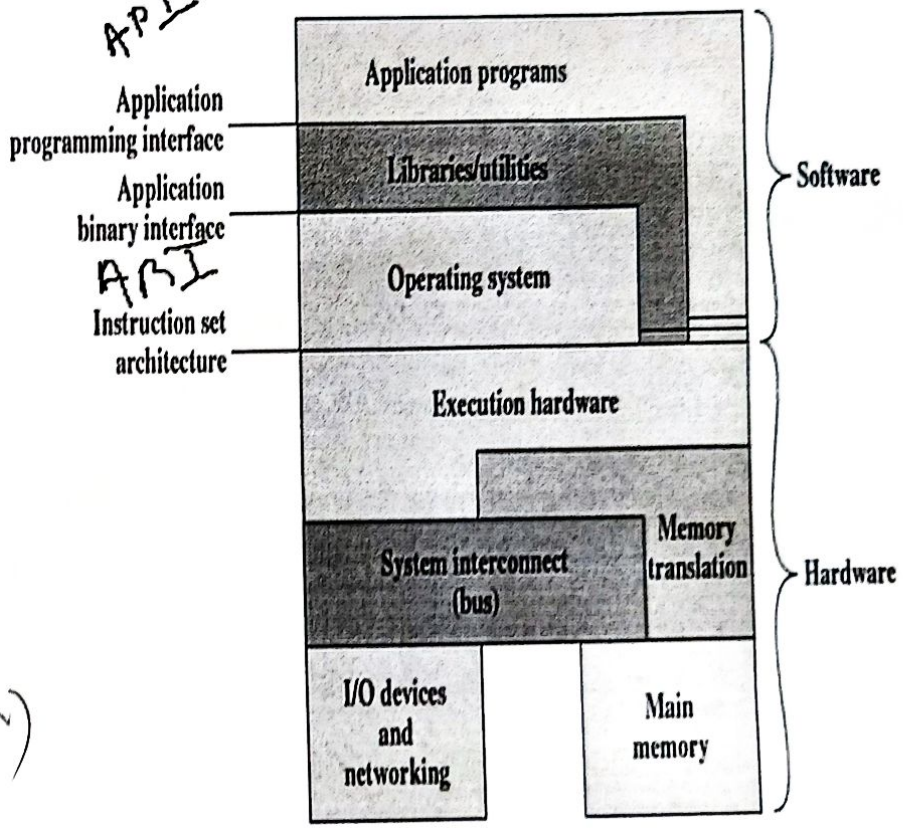

Fig. 1: Computer Hardware and Software Structure [Source: Computer Organization and Architecture by William Stallings]

THE OS AS A RESOURCE MANAGER

1. OS is responsible for managing the resources for the movement, storage and processing of data.
2. OS provides instructions for the processor.
3. OS directs the processor in the use of other system resources.
4. A portion of the OS is in the main memory which includes **KERNEL/NUCLEUS**.
5. The rest of the main memory contains the user programs and data.
6. OS decides when an I/O device can be used by a program in execution.

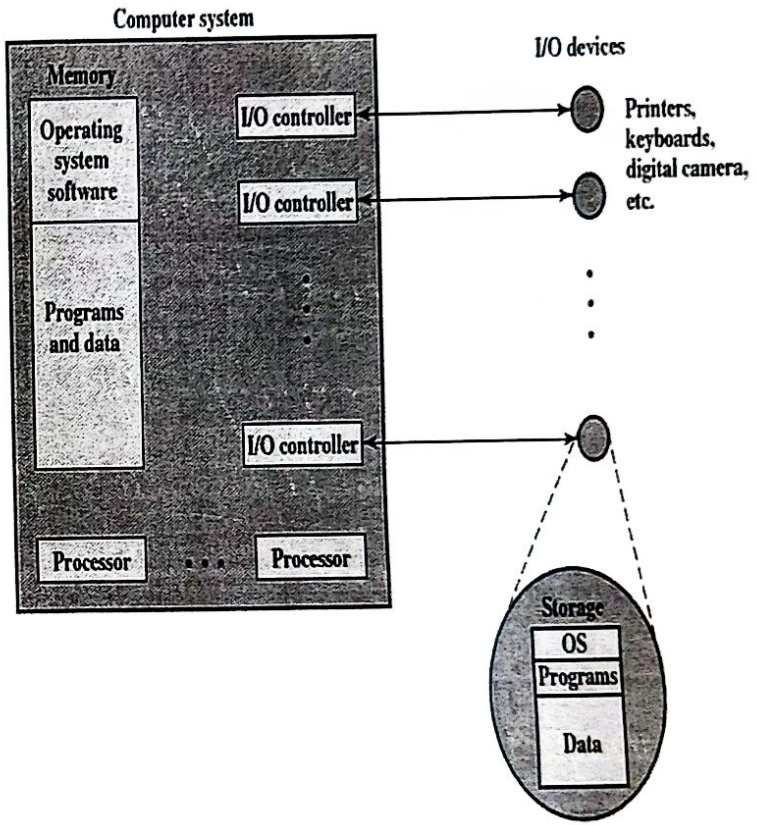

Fig.2: OS as Resource Manager [Source: Computer Organization and Architecture by William Stallings]

③ Types of Operating System (OS):

OS is various types:

① Interactive OS
② Batch operating systems
③ Multiprogramming OS
④ Uniprogramming OS

① Interactive operating system →

It gives the permission to the user to interact directly with the computer.

② Batch Operating system →

It allows the multiple users to use the computer at the same time without direct communication between them.

③ Multiprogramming OS →

It is one that can support to execute multiple programmes simultaneously on a single processor.

③ Uniprogramming Operating System →

It is one that can support to execute only single programme on a processor.

④ Early Systems →

With the earliest computers (before 1940):

① There was no OS
② The user interacted directly with the computer.
③ The processors were run from a console (consisting of display light, toggle switches, input devices, & printer)
④ programs in the processor code were loaded through the input devices (e.g. a card reader)
⑤ Error condition was indicated through lights.
⑥ The user checks the registers & main memory to determine the cause of error.

(7) Normal completion of the program appeared on the printer.

The Early systems presented two main problems:

(1) scheduling

(2) Setup time

(1) scheduling →

scheduling is the activity of process manager that handles the removal of the running process from the CPU & the selection of the another process on the basis of a particular strategy.

or

On the other hand, the user might run into problems, not finish in the allotted time, & be forced to stop before resolving the problem.

(2) Setup time →

Setup time is the interval needed to adjust the setting on a machine, so that it is ready to process a job.

(5) Simple batch systems –

* Early processors were very expensive, & hence it was important to maximize processor utilization.
* So simple batch systems were developed to improve processor utilization.
* It is also known as Monitor.
* The user has no direct access to the processor.
* The user submits the job on cards or tape to a computer operator, who batches the jobs together sequentially & places the entire batch on an input device (card reader or magnetic tape device) for use by the monitor.
* The monitor controls the sequence of events. Much of the monitor must always be in main memory & available for executing. That portion is known as the resident monitor. The rest of the monitor consists of utilities & common functions that are loaded as subroutines to the user program.

* The monitor reads in jobs one at a time from the input device.
* Now the current job is placed in the user program area & the control is passed to this job.
* After completion of the job the control is returned to the monitor which reads the next job.
* The result of each job are printed out for delivery to the user.
* The monitor handles the scheduling problem as well as the job setup time.
* With each job, instructions are included in a job control language (JCL). This is special type of programming language used to provide instructions to the monitor.
* Other desirable hardware features:
  ① Memory protection; ② Timer; ③ privileged instruction
  ④ interrupts.

Contd.

RESIDENT MONITOR

- The portion of monitor always present in the main memory and available for execution is known as resident monitor.
- The monitor reads in jobs one at a time from the input device.
- Now the current job is placed in the user program area and the control is passed to this.
- After completion of the job the control is returned to the monitor which reads the next job.
- The result of each job are printed out for delivery to the user.

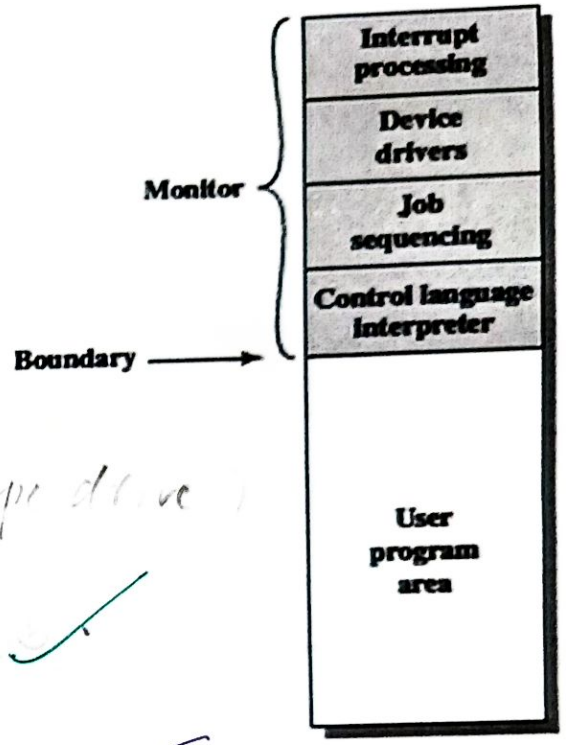

Fig.3: Memory Layout for a Resident Monitor [Source: Computer Organization and Architecture by William Stallings]

(6) Multiprogrammed Batch system :-

* In a simple batch system, processor utilization time is very less due to I/O devices are slow compare to the processor.

* Ex →

A program processes a file of records & executes 100 processor instructions per record. Computer spends over 96% of its time waiting for the I/O devices to finish transferring data.

Read one record from file	15 MS
Execute 100 instructions	1 MS
Write one record to file	15 MS
Total	31 MS

Percent CPU utilization $= \frac{1}{31} = 0.032 = 3.2\%$

∴ Hence 96.8% of its time waiting for the I/O devices.

* In case of multiprogrammed batch systems, the processor utilization time increases.

Contd.

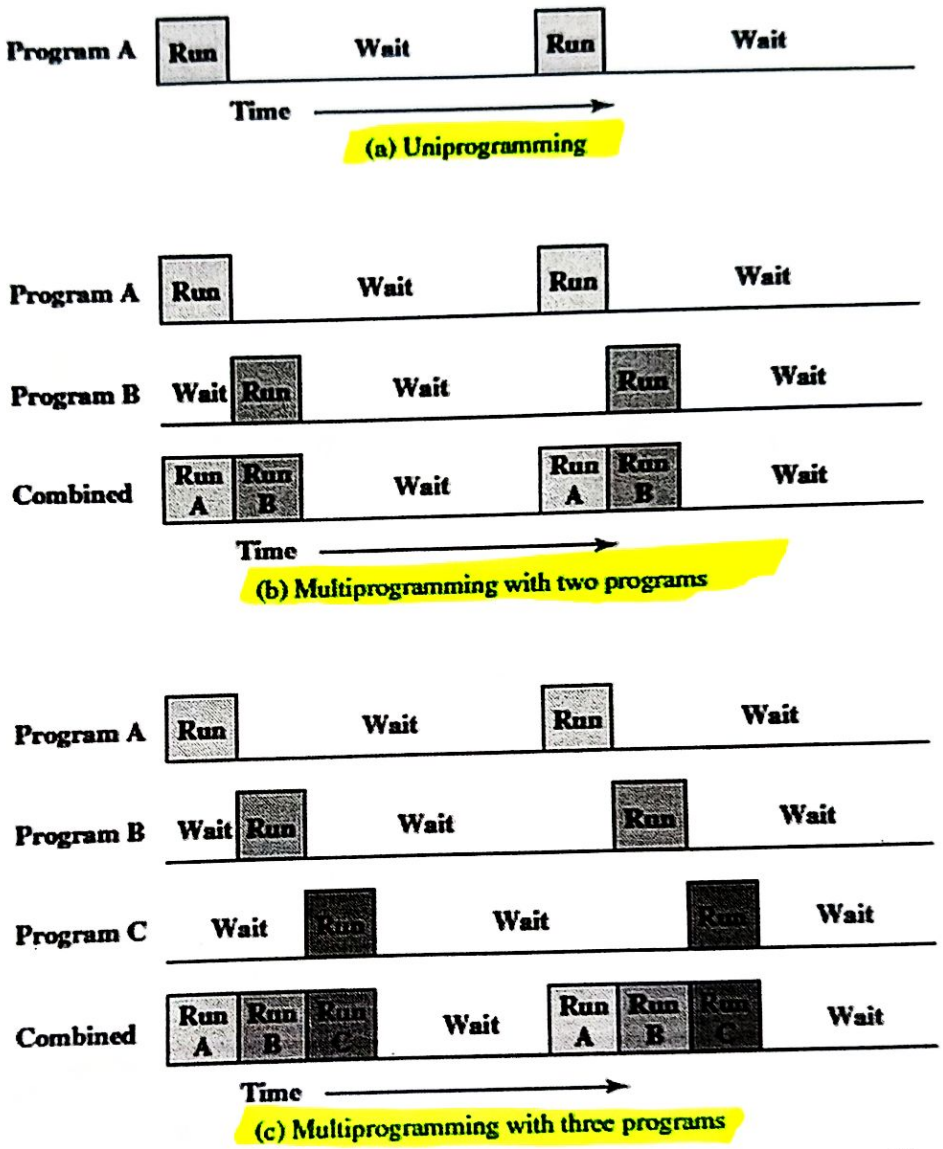

Fig. 5: Multiprogramming example [Source: Computer Organization and Architecture by William Stallings]

(7) Time-sharing Systems →

* processor's time is shared among multiple users.
* Multiple users can simultaneously access the system through terminals.
* Ex→ If there are n users actively requesting service at one time, each user will see on the average $(\frac{1}{n})$ of the effective computer speed.

Table 3: Batch Multiprogramming versus Time sharing

	Batch Multiprogramming	Time sharing
principal objective	Maximize processor use	Minimize response time
source of directives to operating system	Job control language commands provided with the job.	commands entered at the terminal.

8.2 Scheduling →

Types of scheduling

1. Long-term scheduling
2. Medium-term scheduling
3. short-term scheduling.
4. I/o scheduling.

① Long-term scheduling →

* The long-term scheduler determines which programs are to be admited to the system for processing.
* Thus, it controls the degree of multiprogramming (number of processes in memory)

② Medium-term scheduling →

* Medium-term scheduling is a part of the swapping function.

* The swapping-in decision is based on the need to manage the degree of multiprogramming.

(3) short-term scheduling →

* The short-term scheduling also known as dispatcher, executes frequently makes the fine-grained decision of which job to be executed.

Contd.

(2) PROCESS STATES

- ✓ During the life-time of a process its status changes a number of times.
- ✓ Its status at any point of time is known as a **state.**

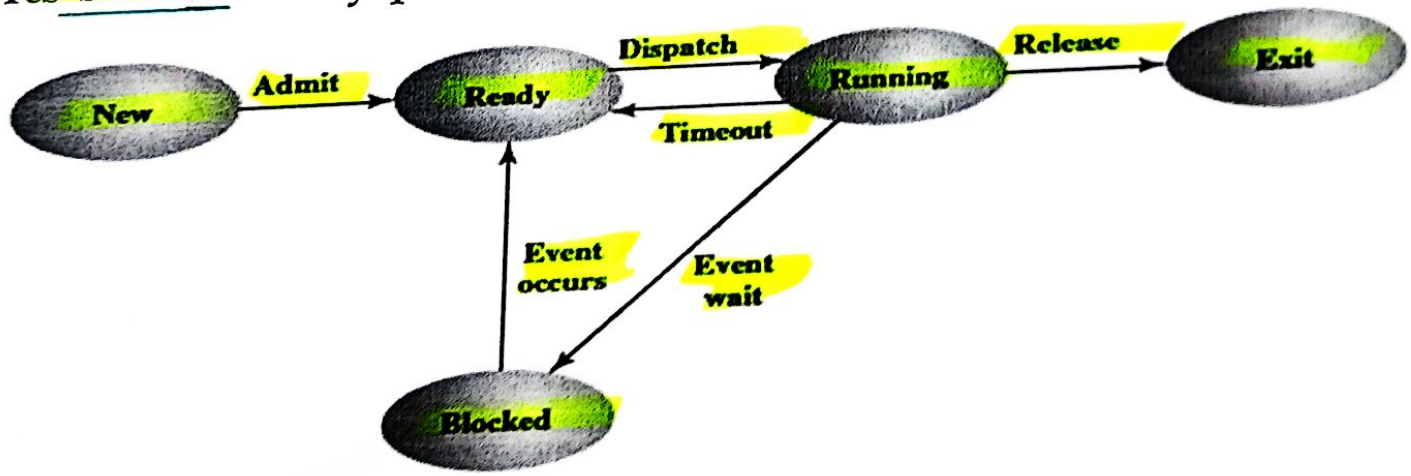

Fig.7: Five-State Process Model [Source: Computer Organization and Architecture by William Stallings]

* Ready → The process is ready to execute & is awaiting access to the processor.

* Running → The process is being executed by the processor.

process control block —

* For each process in the system, the OS must maintain information indicating the state of the process & for process execution. For this purpose, each process is represented in the OS by a process control block.

* process control block contains:
  1. Identifier → Each current process has a unique identifier.
  2. State → The current state of the process (new, ready, & so on)
  3. Priority → Relatively priority level.
  4. PC → It holds the address of the next instruction to be executed.
  5. Memory pointer → The starting & ending locations of the process in memory.

Contd.

PROCESS CONTROL BLOCK

- ✓ For each process in the system, the OS must maintain information indicating the state of the process and for process execution.
- ✓ Each process is represented in the OS by a process control block.
- ✓ When the scheduler accepts a new job for execution it creates a blank process control block and places the associated process in the new state.

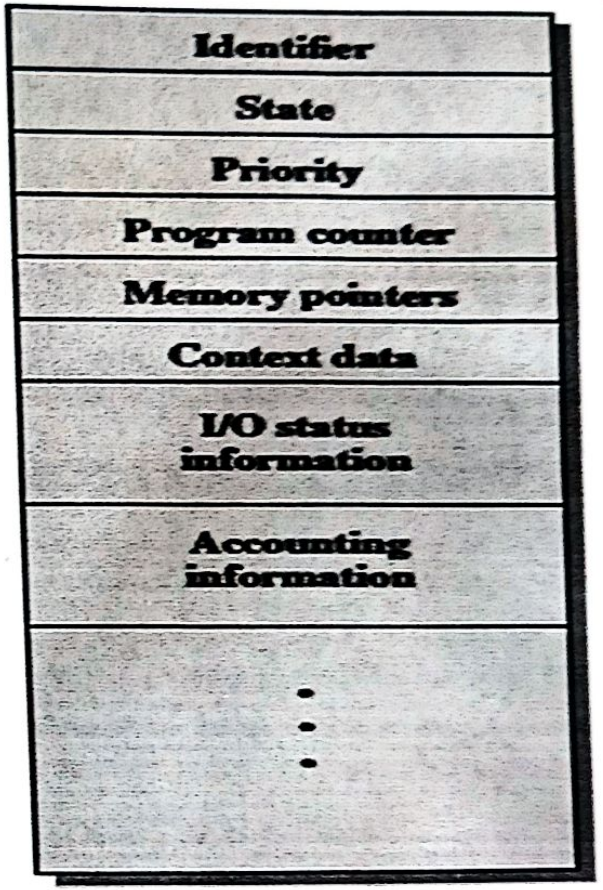

Fig.8: Process Control Block

[Source: Computer Organization and Architecture by William Stallings]

⑥ Context data → These are data that are present in registers in the processor while the process is executing.

* When the scheduler accepts a new job for executing, it creates a blank process control block & places the associated process in the new state.