UNIT I

Operating Systems

- 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.
- The term OS denotes the program modules with in computer system that govern the control of resources such as processors, main storage, secondary storage, i/o devices and files.

Operating System (OS) is system software, which acts as an interface between a user of the computer and the computer hardware.

The main purpose of an Operating System is to provide an environment in which we can execute programs.

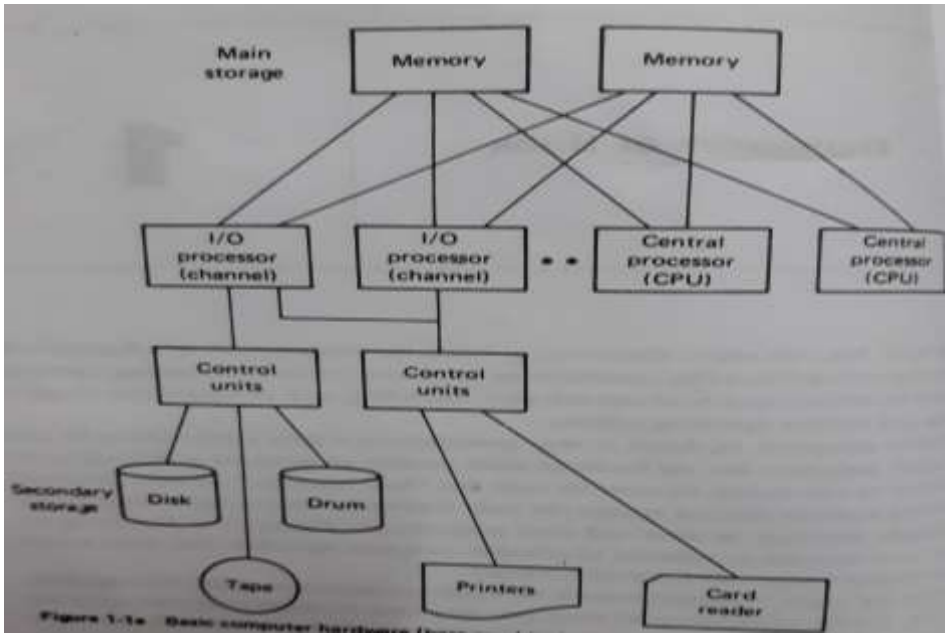
The main goals of the Operating System are:

(i) To make the computer system convenient to use,

(ii) (ii) To make the use of computer hardware in efficient way.
Operating System may be viewed as collection of software consisting of procedures for operating the computer and providing an environment for execution of programs.
It is an interface between user and computer. So an Operating System makes everything in the computer to work together smoothly and efficiently.

Basic concepts and Terminology

- **Computer hardware structure terminology**



- Many different h/w configurations exist.
- Instructions and data are stored in main storage or core memory in coded form.
- Connected to memory are various processors.
- A processor is a hardware device capable of executing the instructions and perform the indicated operations.
- A computer system may have one or more CPU.
- A CPU manipulates and perform arithmetic operations on data in memory.
- It also executes instructions to control other processors.
- For example, START I/O instruction is executed by CPU to initiate the I/o processor.
- Certain processors are specialized to perform specific tasks efficiently and inexpensively.
- The I/O processors are designed to control I/O devices and handle the movement of data from I/O and memory
- These processors execute several instructions simultaneously and control several I/O devices.
- Most common i/o devices such as card readers , printers etc require control circuits .
- For reasons of economy this hardware is sometimes separated out in to a separate device called control unit.

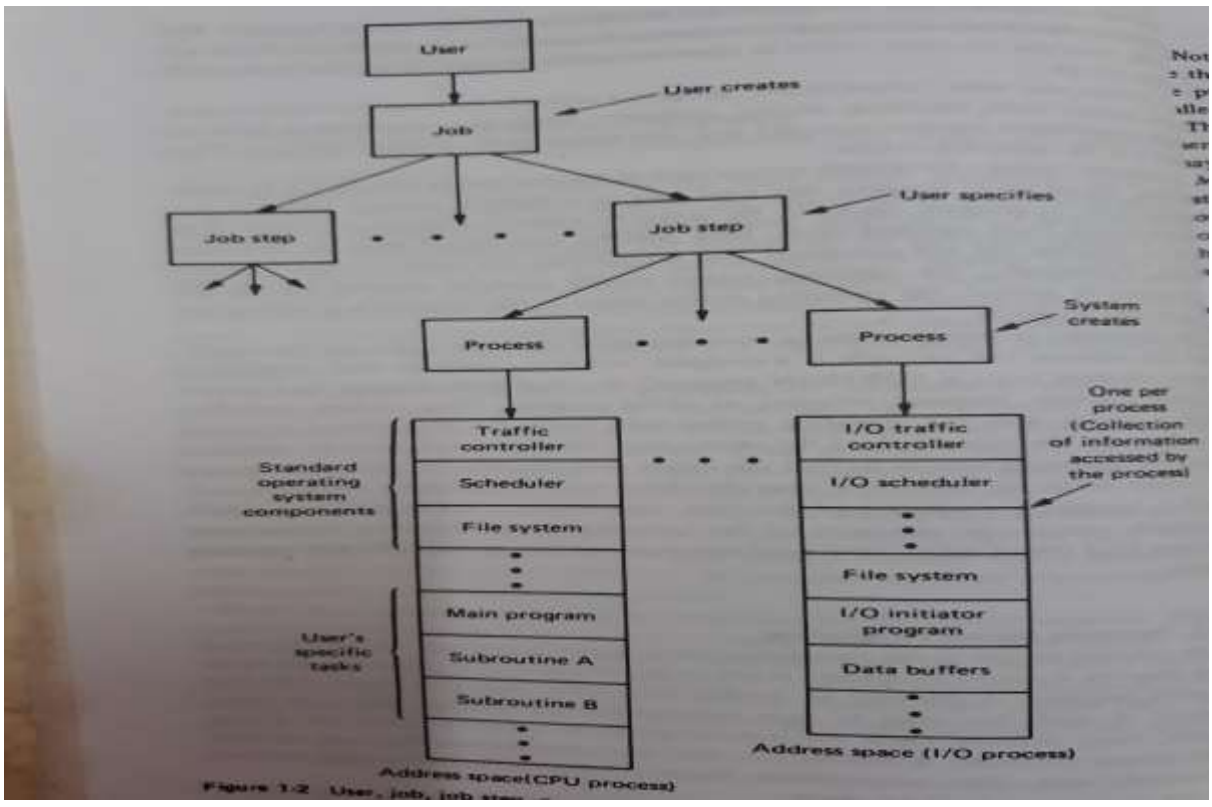
Programming terminology

- Software is a collection of program or data to perform certain tasks.

- A program is a sequence of instruction which are placed in memory and interpreted by the processor.
- Sometimes the program also stored in secondary storage device like disk, drum or tape.
- A system may also have prepacked program which are available to the users.
- In general the OS assists in construction and execution of user program.
- The OS also provides an environment in which particular software package can also execute.
- One such package is compiler program which translates the source program to object program.
- Most compilers interact closely with OS during compilation of source code and execution of object code.
- During compilation the compiler needs OS to read the source program.
- During execution the object program needs sophisticated run time environment which is provided by the OS.

Operating system terminology

- A user submits the job to the OS.
- A job is a collection of activities needed to do the required work.
- A job may be divided in to several steps.
- A job steps are units of work that must be done sequentially.
- Once the user job submitted to the OS, the OS create several processes.
- A process is a computation that must be done concurrently with other computation.



- The collection of program and data that are accessed in a process forms an address space.
- In the fig. there are two address space one for i/o process and other for CPU process.
- The code for file system portion of OS is same for both address space.
- The code which is shared is called pure code in which it cannot be modified by itself.
- The OS must map the address space of process in to physical memory.
- This is assisted by special hardware (e.g a paged memory) or by a software(e.g swapping).
- **Multiprogramming** systems are the one which consists of several processes in states of execution at the same time.
- If the computation of the process has started the it is in the state of execution.
- Such process may not stated to be completed or terminated.
- **Privileged Instructions**-The OS of some computer use special instruction or hardware which are not available to the ordinary user are called privileged instruction.

State of execution of Process

- Some contemporary computers have atleast two states of execution.
- One is problem state(user state , slave state) and other is supervisor state(executive state,master state).
- In supervisor state the process can execute privileged instruction.
- One privileged instruction may change the state of the machine.
- Other instruction may start i/o processor , change interrupt status of machine etc.
- Protection hardware- used to control access of the memory.
- The OS protects some parts of memory to be not writable.
- This hardware protects the memory from accidental write by the user.
- Interrupt hardware – allows the OS to coordinate operations going on simultaneously.
- It allows a non sequential flow of a program.
- An interrupt is a mechanism by which a processor is forced to take a note of an event.
- Hardware exist to mask the certain interrupts.
- Many configuration does not have all these special instructions or hardware.
- But some systems have even more elaborate hardware facilities than those referred above.

Operating System as a Resource Manager

The Operating System is a manager of system resources.

A computer system has many resources as stated above.

Since there can be many conflicting requests for the resources, the Operating System must decide which requests are to be allocated resources to operate the computer system fairly and efficiently.

Here we present a framework of the study of Operating System based on the view that the Operating System is manager of resources.

The Operating System as a resources manager can be classified in to the following three popular views: primary view, hierarchical view, and extended machine view.

The primary view is that the Operating System is a collection of programs designed to manage the system's resources, namely, memory, processors, peripheral devices, and information.

It is the function of Operating System to see that they are used efficiently and to resolve conflicts arising from competition among the various users.

The Operating System must keep track of status of each resource; decide which process is to get the resource, allocate it, and eventually reclaim it.

The major functions of each category of Operating System are.

Memory Management Functions

To execute a program, it must be mapped to absolute addresses and loaded into memory.

As the program executes, it accesses instructions and data from memory by generating these absolute addresses.

In multiprogramming environment, multiple programs are maintained in the memory simultaneously.

The Operating System is responsible for the following memory management functions:

- ✓ Keep track of which segment of memory is in use and by whom.
- ✓ Deciding which processes are to be loaded into memory when space becomes available. In multiprogramming environment it decides which process gets the available memory, when it gets it, where does it get it, and how much.
- ✓ Allocation or de-allocation the contents of memory when the process request for it otherwise reclaim the memory when the process does not require it or has been terminated.

Processor/Process Management Functions

A process is an instance of a program in execution.

While a program is just a passive entity, process is an active entity performing the intended functions of its related program.

To accomplish its task, a process needs certain resources like CPU, memory, files and I/O devices.

In multiprogramming environment, there will a number of simultaneous processes existing in the system. The Operating System is responsible for the following processor/ process management functions:

- ✓ Provides mechanisms for process synchronization for sharing of resources amongst concurrent processes.
- ✓ Keeps track of processor and status of processes. The program that does this has been called the traffic controller.
- ✓ Decide which process will have a chance to use the processor; the job scheduler chooses from all the submitted jobs and decides which one will be allowed into the system. If multiprogramming, decide which process gets the processor, when, for how much of time. The module that does this is called a process scheduler.
- ✓ Allocate the processor to a process by setting up the necessary hardware registers. This module is widely known as the dispatcher.
- ✓ Providing mechanisms for deadlock handling.
- ✓ Reclaim processor when process ceases to use a processor, or exceeds the allowed amount of usage.

I/O Device Management Functions

- ✓ An Operating System will have device drivers to facilitate I/O functions involving I/O devices.
- ✓ These device drivers are software routines that control respective I/O devices through their controllers.
- ✓ The Operating System is responsible for the following I/O Device Management Functions:
- ✓ Keep track of the I/O devices, I/O channels, etc. This module is typically called I/O traffic controller.
- ✓ Decide what is an efficient way to allocate the I/O resource. If it is to be shared, then decide who gets it, how much of it is to be allocated, and for how long. This is called I/O scheduling.
- ✓ Allocate the I/O device and initiate the I/O operation.

Reclaim device as and when its use is through. In most cases I/O terminates automatically.

Information Management Functions

- ✓ Keeps track of the information, its location, its usage, status, etc. The module called a file system provides these facilities.
- ✓ Decides who gets hold of information, enforce protection mechanism, and provides for information access mechanism, etc.
- ✓ Allocate the information to a requesting process, e.g., open a file.
- ✓ De-allocate the resource, e.g., close a file.

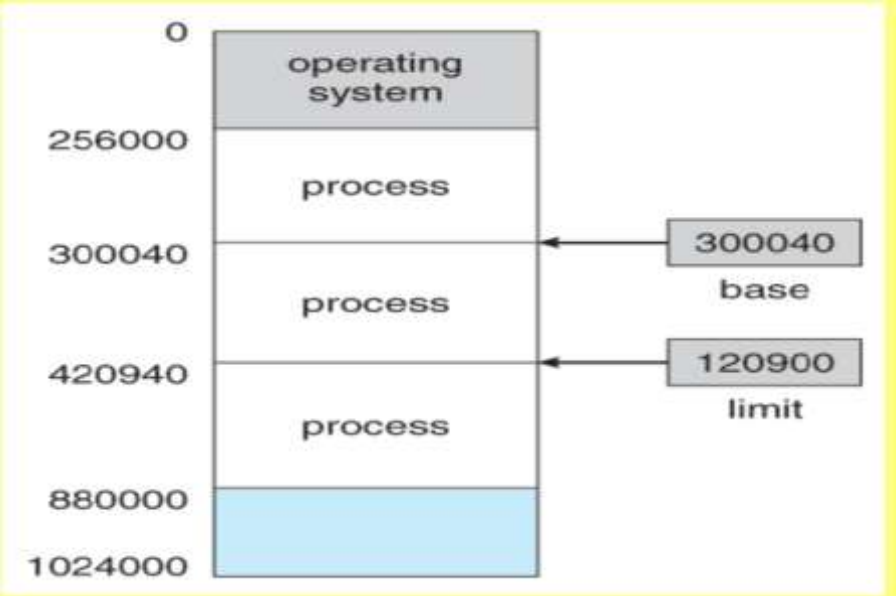
Network Management Functions

- ✓ An Operating System is responsible for the computer system networking via a distributed environment.

- ✓ A distributed system is a collection of processors, which do not share memory, clock pulse or any peripheral devices.
- ✓ Instead, each processor is having its own clock pulse, and RAM and they communicate through network.
- ✓ Access to shared resource permits increased speed, increased functionality and enhanced reliability.
- ✓ Various networking protocols are TCP/IP (Transmission Control Protocol/ Internet Protocol), UDP (User Datagram Protocol), FTP (File Transfer Protocol), HTTP (Hyper Text Transfer protocol), NFS (Network File System) etc.

An Operating System- Process View Point

- ✓ An operating system consists of collection of programs to manage resources.
- ✓ The below picture depicts three processes coexisting in a multiprogrammed system.



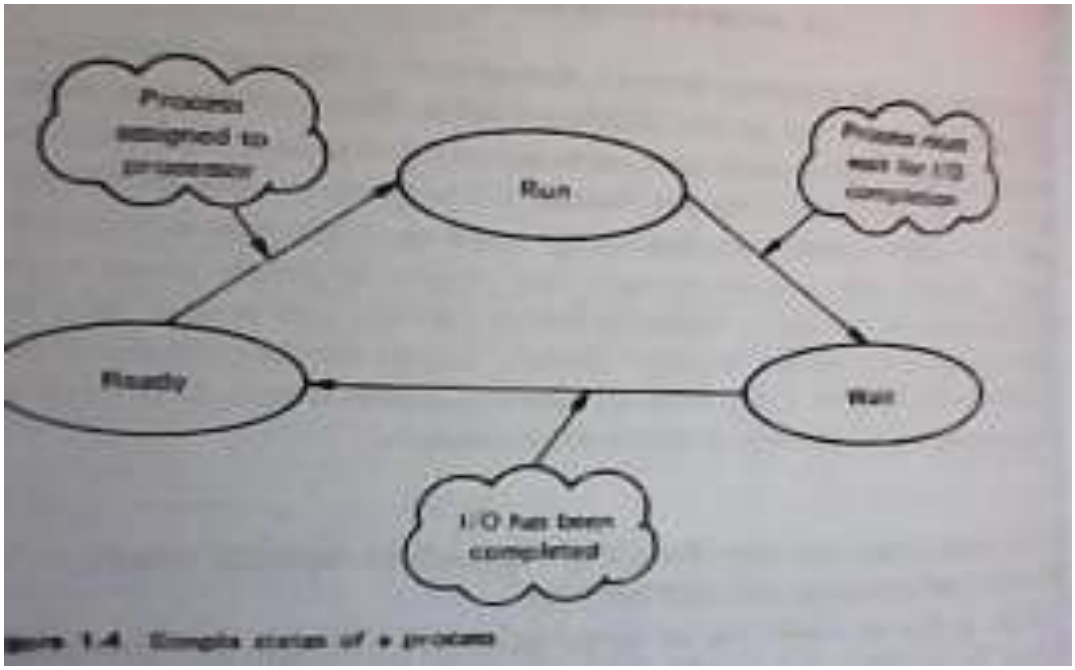
Life cycle of a process

The life cycle of a process can be represented by the transitions between the three states.

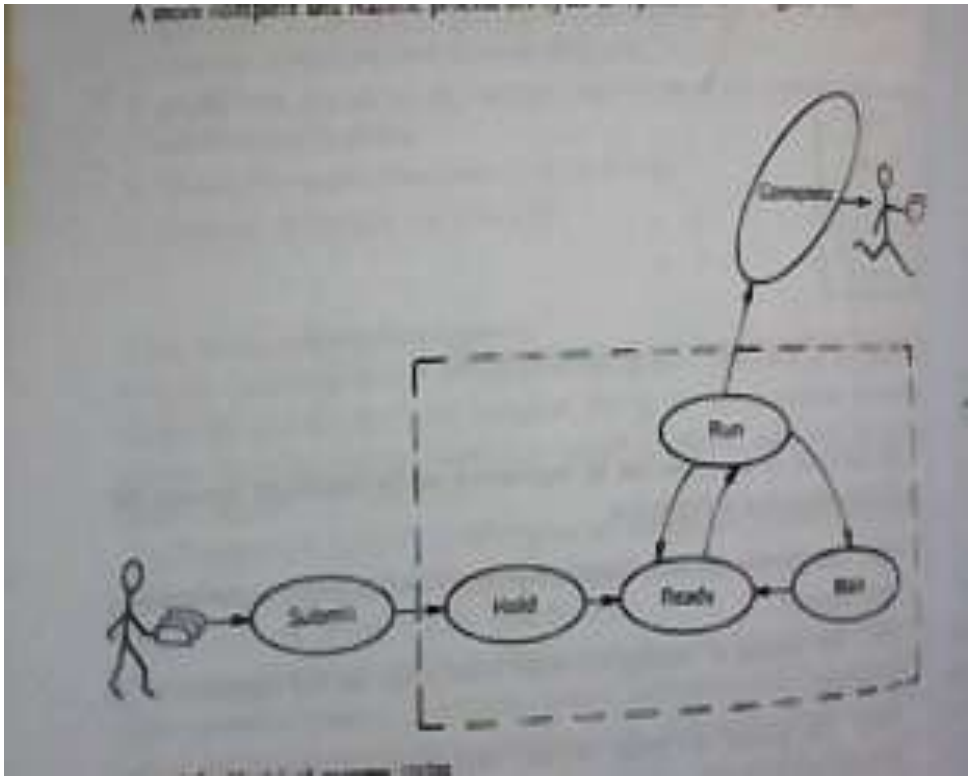
Run state- The process has been assigned a processor and its program are presently being executed.

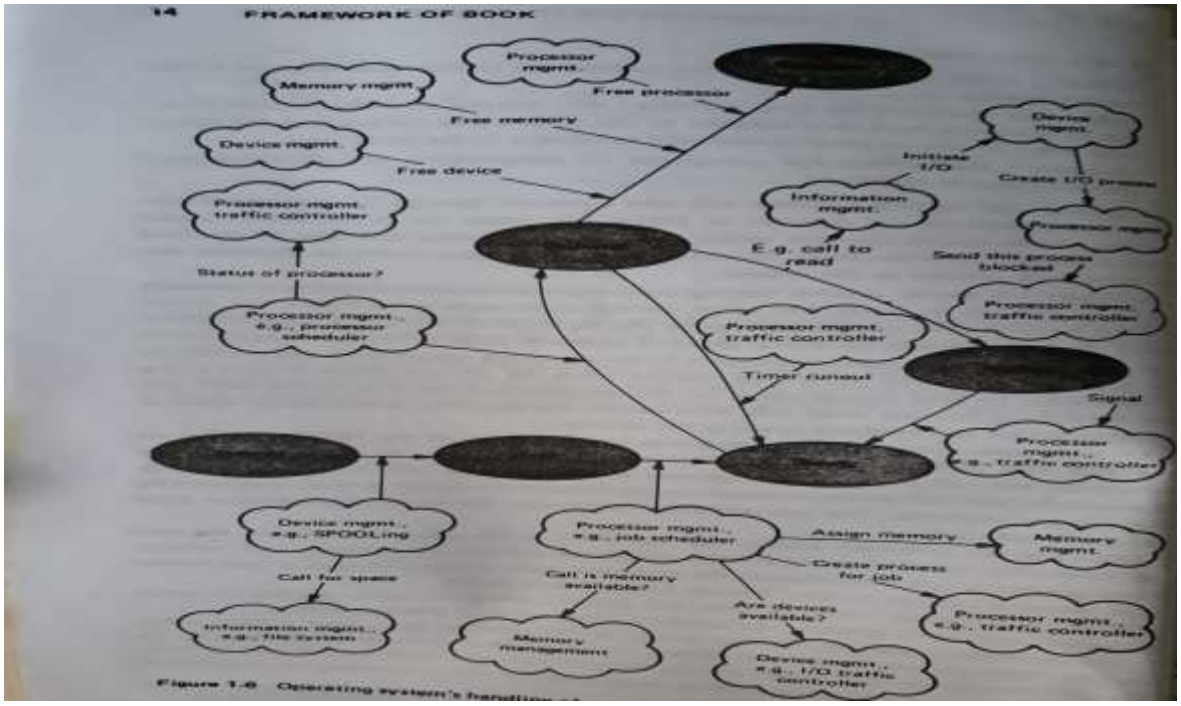
Wait state- The process is waiting for some event.

Ready state- The process is ready to run but there are more processes and its is waiting for its turn.



- The above fig. is a simplified view.
- The more complete and realistic process life cycle is represented as follows.
- submit- The user submit the job to the system.The system must respond to the user's request.
- Hold-The user's job is converted to machine readable form but no resources have been assigned to the process.
- Complete- The process has been completed its execution and reclaim its resources





The OS will take its process through all its states.

The process management plays an important role in sequencing the process through all its states.

The user submits the job to the system(submit state).

The job consists of several job steps of programs preceded by job control cards.

The job control card pass information to the OS as to what resources the job needs.

The SPOOLing routine reads job and place it in disk(Hold state).

To find the storage space for user job the SPOOLing routine call information management.

The information management must keep track of all available space in the system.

The job scheduling routines scans all the SPOOLed file on disk and picks a jobto be admitted to the system.

In picking a job the scheduler call memory management to find sufficient memory is available and also call device management to find whether devices requested by the job are available.

- The job scheduler determine the device requirements from the user’s job cards.
- Once the job scheduler decides that the job is to be assigned resources the traffic controller is called to create the associated process information and memory management is called to allocate the necessary main storage.

- The job is then loaded in to the memory and then the process is ready to run.(ready state).
- When processor becomes free the process scheduler scans the list of ready processes, chooses a process and assigns a processor to it.(running state).
- If the running process requests access to information (e.g reading a file) the information management would call device management to initiate the reading of file.
- Device management would initiate the I/O operation and then call process management to indicate that the process requesting the file is awaiting the completion of I/O operation.(wait state).
- When the I/O is completed the I/O hardware sends signal to traffic controller in process management which places process back in ready state.
- If the process complete its execution the it is placed in completed state and all allocated resources are freed.
- The fig. represents one possible view of life cycle of a process.
- In more complex system, there may be even more states.

Operating System-Hierarchical and Extended Machine view

- In process view we discussed about various system resources which are used by operating system.
- But we need know where these resources logically reside and how they are called.
- This section presents an hierarchical view of OS to show how these modules are related.
- The hierarchical view results in a simple, clean structure that is easy to analyze and implement.

Extended Machine Concept

- In this section we introduce the basic hardware features of contemporary computers.
- This is called bare machine- a computer with out software.
- This environment is not liked by most programmers.
- The program which is written by the programmers involve interaction with the key system resources such memory and i/o.
- The instruction to perform these resource management function are provided by the OS.
- The user program can request the services by issuing special supervisor call instruction.
- These instruction transfer the control to OS.
- The OS provides many instruction in addition to the hardware instruction

- The sum of these instruction are called instruction set of extended machine.
- The kernel of OS runs on the bare machine and user program runs on entended machine.

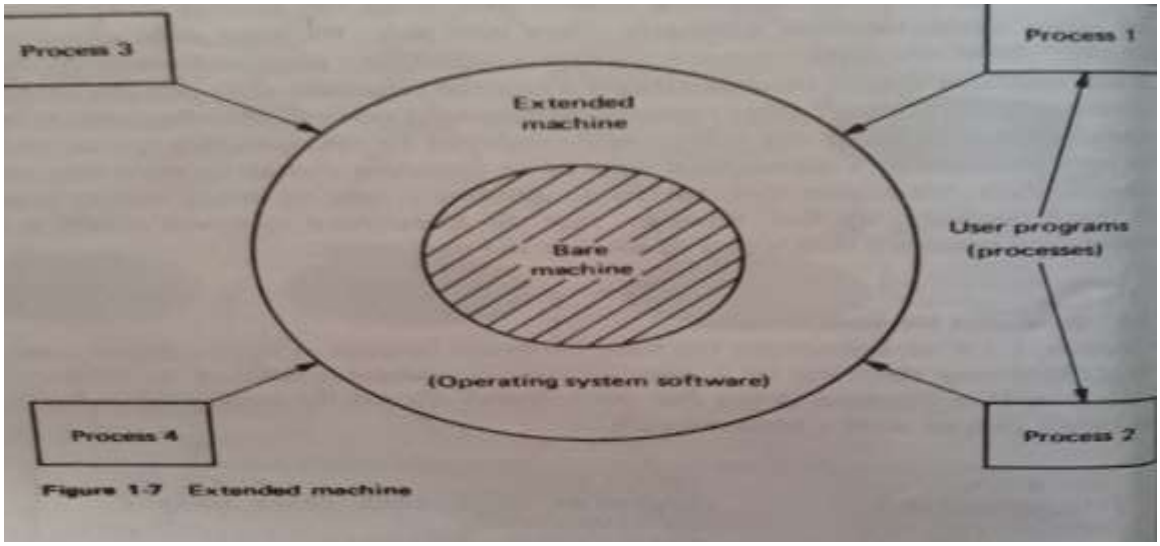


Figure 1-7 Extended machine

Hierarchical Machine view

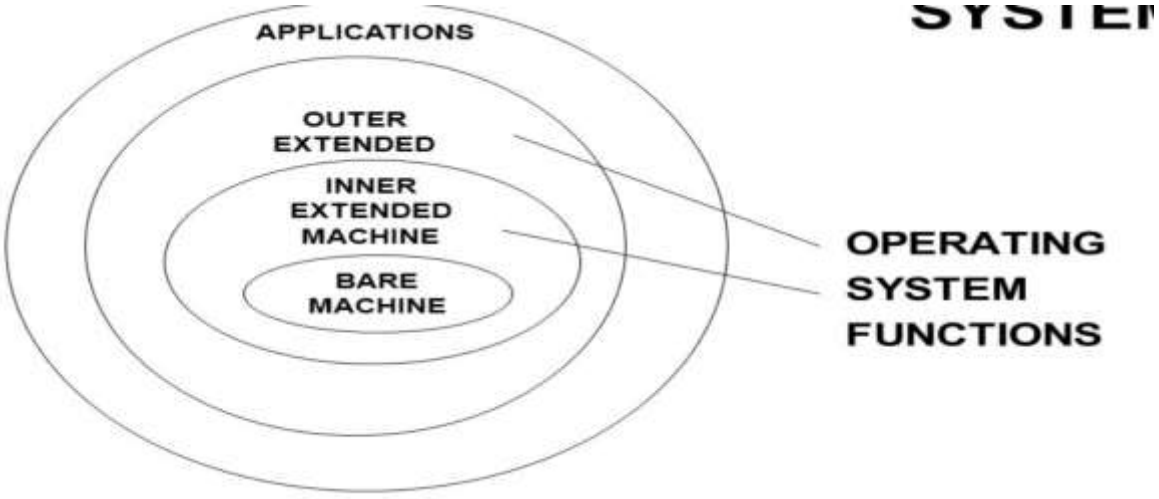
* Most of the early OS consists of a one big program.

* As years passed this type became unmanageable.

* The fig. is an extension of the previous fig.

* Here the main key function of the OS are placed in to the inner extended machine.

* And remainder of the key modules are placed in outer extended machine.



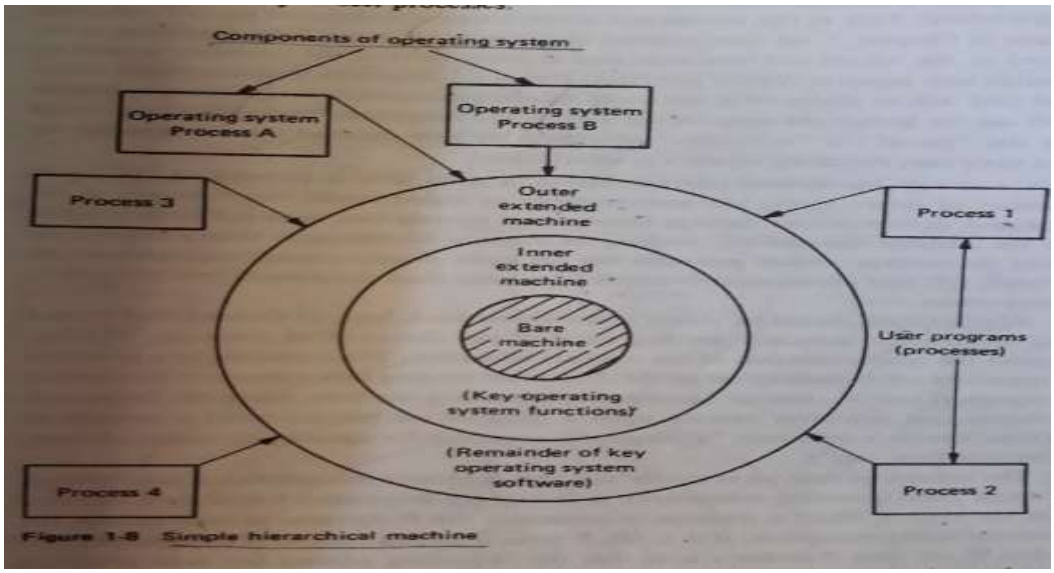
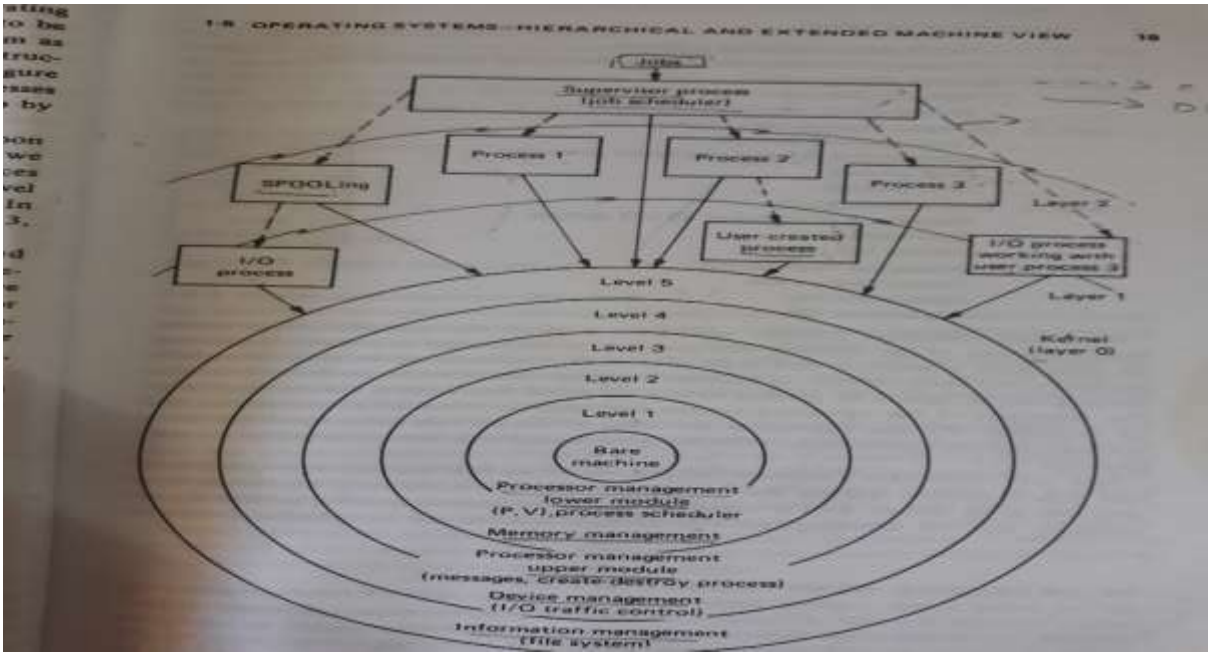


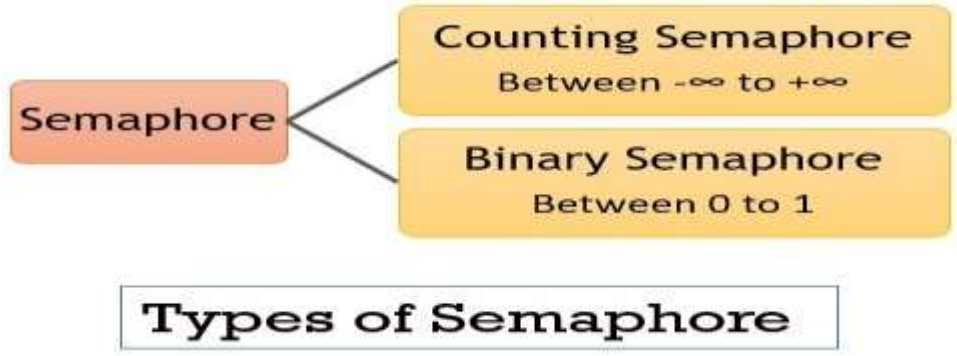
Figure 1-8 Simple hierarchical machine

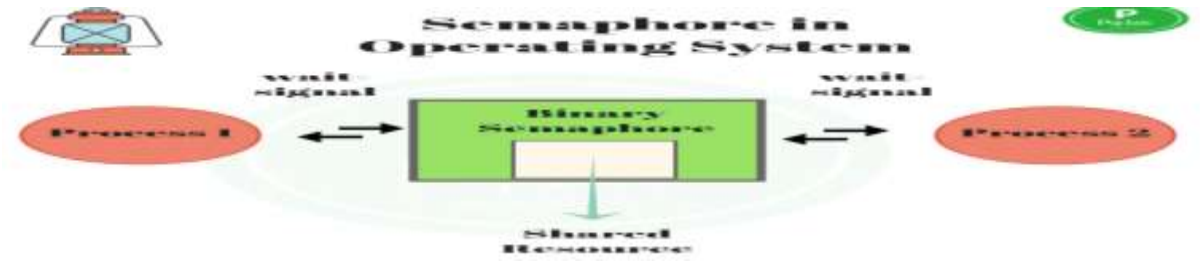
- In this hierarchical view the user did not know where the modules of OS resides.
- It may be in inner extended machine or in outer machine or as a process.
- The inner and outer machine concept can be generalized in to levels of extended machines.
- The OS processes can be generalized in to layers of processes.
- All the modules in the extended machine and those that operate as process layers are collectively called the kernel of the operating system.
- At present there is no firm rule to indicate how many layers should be used and what modules be placed in which levels , what should be in kernel, etc.
- Two designs based on hierarchical concept was developed.
- One by Dijkstra(1968) and other by Madnick(1969).



- The fig illustrates the hierarchical structure.
- All the processes use the kernel and share all the resources of the system.
- Some of the processes are parent or controller of others.
- This is denoted by wavy lines.
- In strictly hierarchical structure a given level is allowed to call services of lower level but not the higher level.
- What function is placed in level 1? The one that is used by all the resource manager.
- This will keep track of all the allocated resources.
- This requires the synchronization of the resource allocation.
- The corresponding primitive operations frequently called the P operator (noting that the resource is seized or requested) and V operator(noting that the resource is released).
- These synchronization primitives operate on software mechanism called a semaphore.
- The semaphore may be a simple switch(binary) or a integer value.
- A Semaphore is associated with each resources of the system.
- When a resource is requested then the P operator is used test the corresponding semaphore.
- If it is off, P turns it on and returns.
- If semaphore is already on the P makes a note that requesting process has been placed in WAIT state .

- Later V operator used by other process releases the resource turns the semaphore off and awaken any process waiting for the resource.
- Since most of the modules of OS use this P and V then these functions are placed in core of kernel.





- Examples of primitive function in various levels of kernel are

Level 1 : Processor Management Lower Level

The P Synchronization primitive

The V Synchronization primitive

Process scheduling – the mechanism of multiprogramming

Level 2 : Memory management

Allocate memory

Release memory

Level 3 : Processor Management – Upper level

Create / Destroy process

Send / receive messages between process

Stop process

Start process

Level 4 : Device Management

Keep track of status of all I/O devices

Schedule I/O

Initiate I/O

Level 5 :Information Management

Create / Destroy file

Open / close file

read / write file

The kernel has routines to support the operation of process.

The resource management function can be performed independently and concurrently or serially.

The functions such as job scheduling, remote terminal handling and I/O SPOOLing may execute concurrently with the user process.

Each user may have separate I/O process for each device.

In a flexible system , the system allow the user to create as many number of processes.

This is useful for parallel numerical computation and for handling real time process control and time sharing functions.

I/O programming

Early computers like ibm 360,1401 have 3 basic components.

1.cpu 2. main storage 3. I/o devices

These components are inter connected.

Early systems have all these components.

They also contain instructions like read a card , punch a card and print a line.

Add instruction take 1ms of cpu time read a card instruction take500 ms

As computers evolved performance was also upgraded

CPU speed and size of memory was also increased

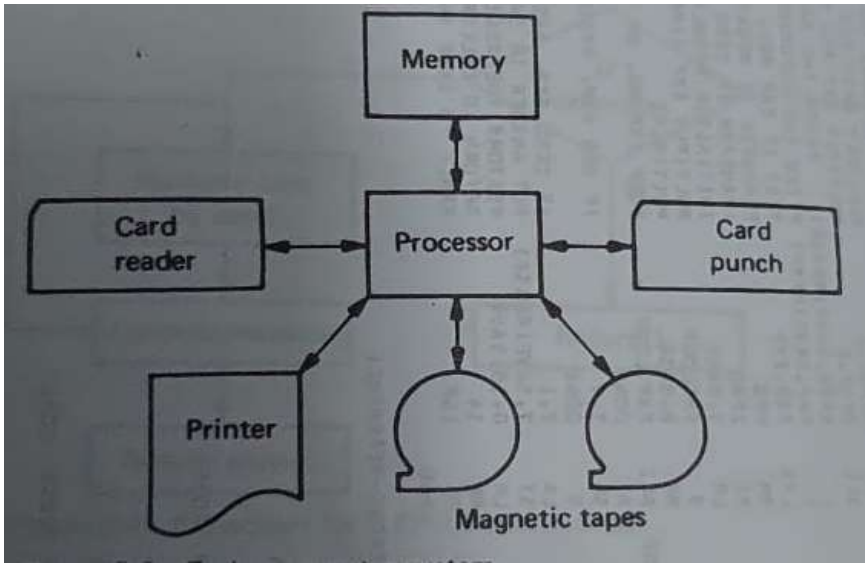
The 1000 bytes with 1ms per access was increased to 1 million bytes with 1micro second per access.

There is a disparity in speed of CPU and I/O devices

So I/O processors was developed.

The I/O channels provide a path for I/O device with the memory

The I/O processors are Specialised devices to operate on I/O devices



The I/O processors are simple and less expensive then CPU

All the I/O execution is through the i/o channel And the CPU is free perform other high speed computation without wasting the time such as reading a card

It is possible that several channels operate at the same speed

OS must supervise all these operations.

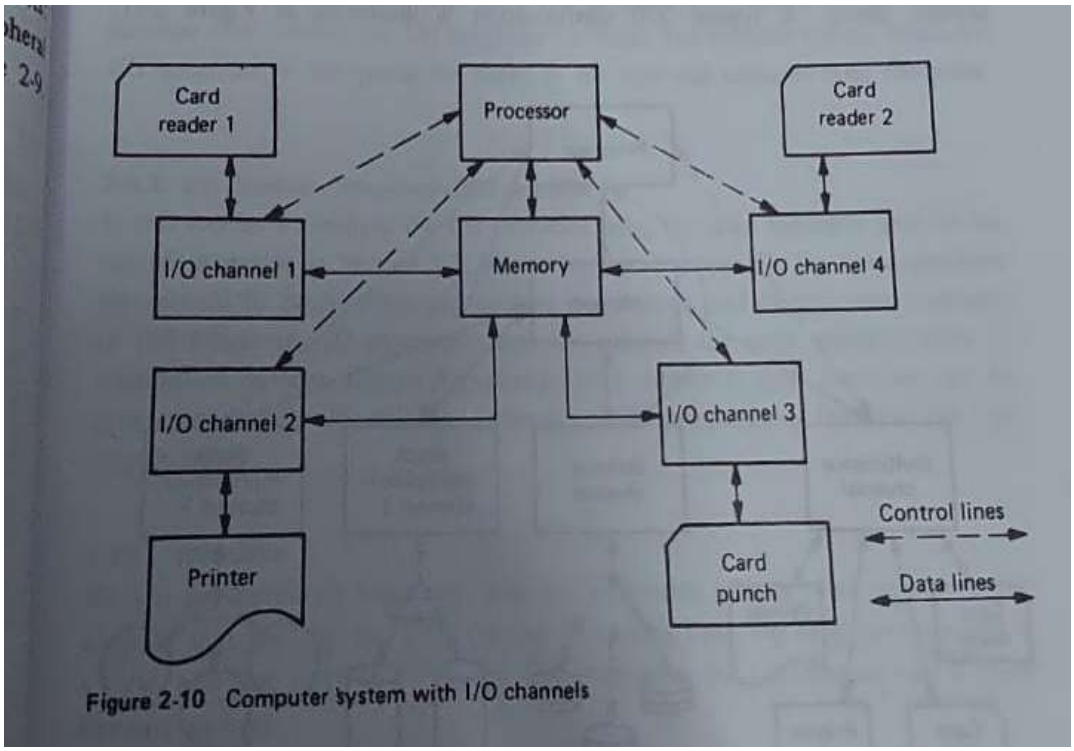


Figure 2-10 Computer system with I/O channels

Types of I/O channels

I/O channels comes in various size, range and shape

though it is less expensive it is important to use or separate channel for each i/o device

this led to the development of Selector and multiplexer channel

Multiple devices connected to each channel

The selector channel can service only one of its device at a time

One device is selected first service

Used for high speed i/o device like Magnetic tape ,Magnetic drum and disc

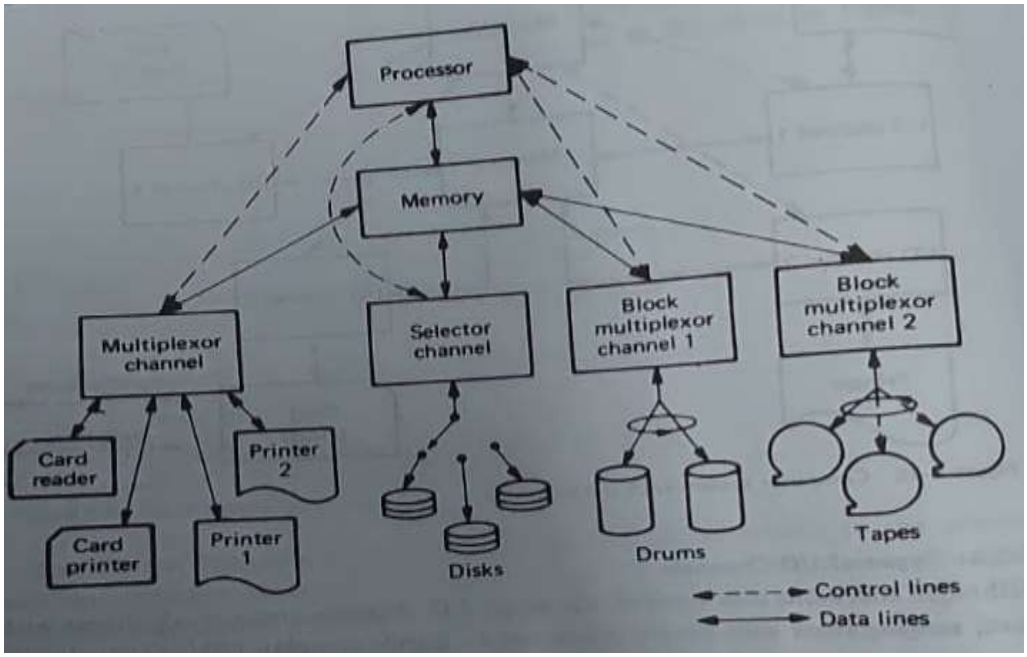
The multiplexer channel services up to 256 devices at a time

It is used for slow devices like card reader, card punch, and printer.

The block-multiplexer channel merges the features of both the multiplexer and selector channels.

It supports a connection to several high-speed devices, but all I/O transfers are controlled with a full block of data as distinguished to a multiplexer channel, which can share only one byte at a time.

The CPU interacts directly with the channels through dedicated control lines and indirectly through reserved storage areas in memory.



I/O programming concepts

The OS coordinates all the processors.

CPU communicate i/o through some instruction like START I/O and HALT I/O

The I/O communicate CPU by means of Interrupt

The I/O processor interprets its own instruction set

The I/O channel instructions are called I/O command.

CPU has also its own instruction

Programs written with I/O commands are called I/O programs or I/O programming

Programmers don't write I/O programs but can call all system supplied functions that provide I/O programming

I/O Processor Structure IBM 360

Memory

The basic unit of memory is byte.

The original architecture of System/360 provided for up to 2^{24} = 16,777,216 bytes of memory.

For addressing the I/O channel uses 24 bit absolute address.

Registers

The I/O channel has no explicit registers but have instruction register and data counter.

Some I/O devices have internal registers similar internal CPU working registers

Data

Only logical and character data can be handled.

Instructions

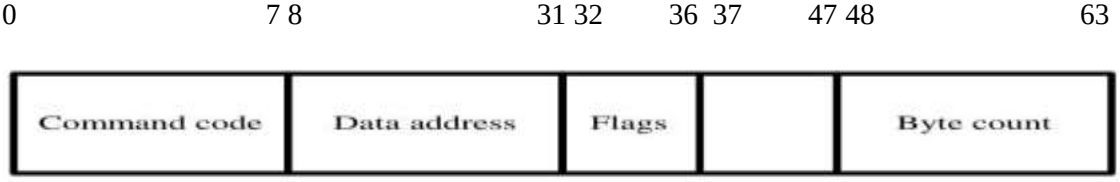
there are three basic grouping of I/O commands.

1. data transfer - read, read backwards, write and sense

2. device control - control(page, eject, tape rewind etc)

3. Branching- transfers of control within the channel program

the channel fetches the channel commands ccw from memory and decode them according to the following format



The opcode bits indicate the command to be performed it consists of two parts.

2 to 4 are operation bits. 4 to 6 are modifier bits.

the data address specifies the beginning location off the data field

the count field specifies byte length of the data field

the flag bits further specialize the command .The principal flags are

1. the command chain flag(bit 33)

2. the data chain flag(bit 34)

3. Suppress length indication flag(bit 35)

4.The skip flag(bit 36)

5.Programmed controlled interruption flag(bit 37)

The group of ccw link together by command chains, data chains or transfers is called a channel program.

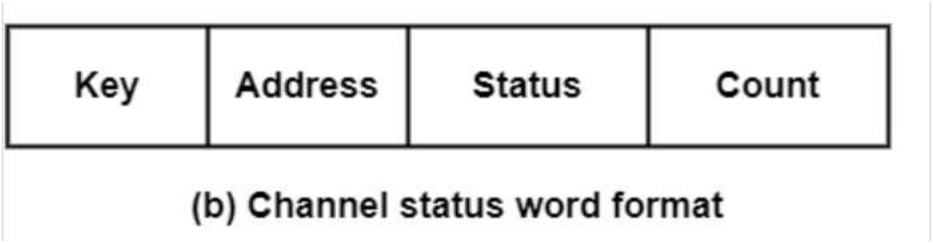
Status

The channel has an internal register that acts as instruction register

Three specific words of memory are used for or status information

The channel address word(CAW) contains the address of first instruction to be executed by the channel

The channel status word(CSW) and coding instruction indicates status of the channel



(b) Channel status word format

Key- protection key used by the channel

Address – address of next channel command

Status - status information

Count- still how many bits are need to process.

Communication between the CPU and the channel

The purpose of having a channel is to free the CPU from having to control detailed i/o operations.

the CPU on the channel are usually in a master slave relationship.

the CPU tells the channel when to start and commands it to stop or change what it is doing

the channel cannot start any operation on its own

there are two types of communication between CPU and channel

1. CPU to channel initiated by the CPU

2. channel to CPU interrupt initiated by channel

all CPU I/O instruction has the following format

The channel and the device number are specified by some of the contents of register B1 and D1

Bits 16 to 31 contains device on the channel

There are three main CPU instructions

1. Start I /O - two items are needed to start I/O

the channel and device number and the beginning address of candle program

2, test I/O- Used to test the addressed channel

3,Halt I/O - execution of current operation at the addressed and channel terminated

The CPU indicate the state of the addressed channel and the device by using Condition Code(CC).

The CC can either be

8-OK

2-busy

1-Not operational

4-verify CSW for status

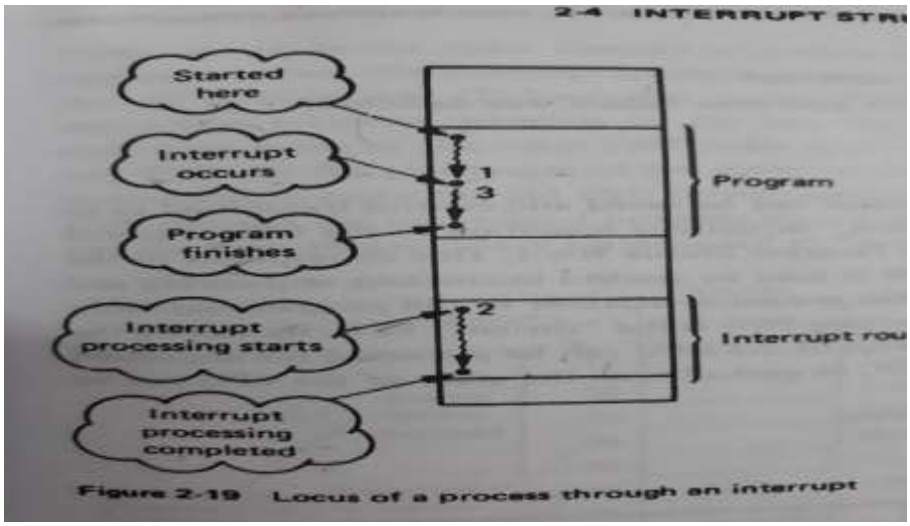
Interrupt Structure and processing

An interrupt is a response to an asynchronous or exceptional event that automatically saves the CPU status to allow later to restart and causes an automatic transfer to specified routine.

Fig

In the fig. upto point 1 the program was executed then the interrupt has occurred.

The processor transfer the control to interrupt routine 2.After the termination of the interrupt the processor resumed processing the original program.



In the fig. upto point 1 the program was executed then the interrupt has occurred.

The processor transfer the control to interrupt routine 2.After the termination of the interrupt the processor resumed processing the original program.

Interrupt types

The interrupt is divided in to five types.

1.Input/Output interrupt

a.Invalid I/O command

b. I/O cahnnel end

c.I/O device end

2.Program Interrupt

a.Invalid CPU instruction

b.Fixed point arithmetic overflow

c.Storage protection violation

3.Supervisor Call Interrupt

4.External Interrupt

a.Interval timer going off

b.Operator’s interrupt button

c.CPU to CPU communication interrupt

5.Machine check interrupt

Interrupt Mechanism

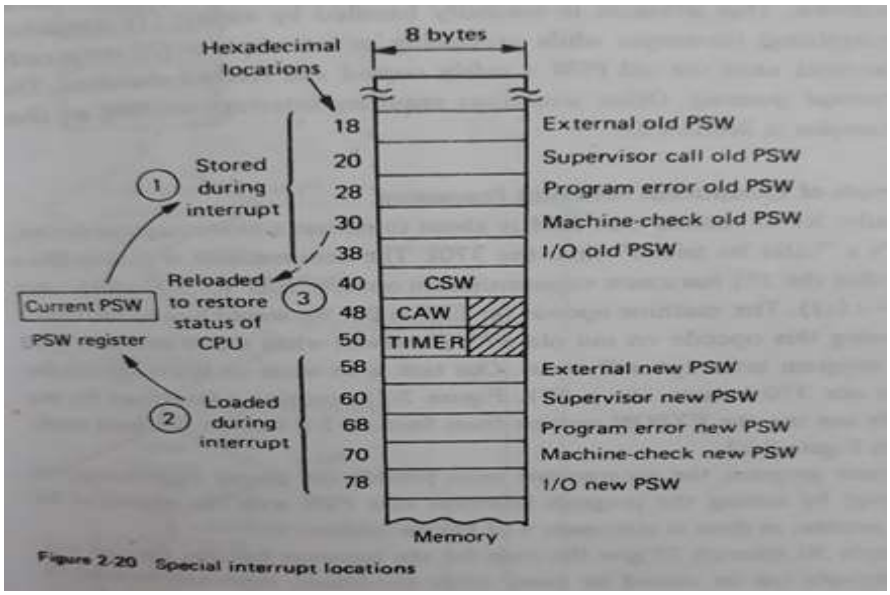


Figure 2-20 Special interrupt locations

The state and cuuent conditions of the CPU is stored in a double word register called Program Status Word(PSW)

This is same as CSW of I/O processor.

processor status word (PSW) A word that describes fully the condition of a processor at each instant.

It indicates which classes of operations are allowed and which are forbidden, and the status of all interrupts associated with the processor.

It will also contain the address of the instruction currently being executed. The PSW is held in a register known as the *processor status register*.

By storing the PSW during interruption the status of the CPU is preserved.

By loading a new PSW or part of one, the state of the CPU can be changed.

Bits in PSW may be used to mask certain interruptions.

When masked I/O, external and machine check interruption may be temporarily inhibited but kept pending.

The other program interruptions and supervisor call interruptions cannot be masked.

There are various status switching instruction

It include Load PSW(LPSW), Set Program Mask(SPM), Set System Mask(SSM), Supervisor call(SVC), and Set Storage Key(SSK).

Most of these are privileged instructions and executed in supervisor mode.

Each of the interrupts is associated with two double words called “old” and “new”.

When an interrupt occurs the interrupt hardware mechanism stores the current PSW in old position and loads the current PSW in new position.

The control transfers to interrupt handler routine specified

After completion of interrupt routine there is usually an LPSW instruction to make old PSW current again.

The interrupt routine access the old PSW to find the condition that cause the interrupt.

This contains the code and the location of the program to be executed when interrupt occurred.

The interrupt code is unique – 01 means invalid operation, 02 means privileged operation, 08 means fixed point overflow.

The PSW interrupt code also indicate the channel and device causing the interrupt.

The CSW also set at that time indicates the cause of the interrupt.

CSW bit 36=1 means cahnnel end bit 37=1 device end.

There are various circumstances in which it is possible to prohibit interrupt.

For example when an I/O interrupt occurred for a device like card reader then current status of the device is stored in old PSW .

During processing if another interrupt occurs for other device such as printer then its staus is stored in old PSW by destroying its previous contents.

The original PSW is lost and it is impossible to restore.

This situation can be handled by

1. Completely masking interrupts while processing an interrupt.
2. Temporarily masking the interrupt until storing the old PSW safely copied and stacked else where. This is called Interrupt queing