

What is an operating system?

It exploits the hardware resources of one or more processor to provide a set of services to system users. It acts as an interface between user application and hardware.

Basic elements of a computer?

Processor - which control the operation of computer and perform data processing

Main memory - stores data and programs.

I/O module - which move the data between computer and its external environment

System bus - It provides for communication among processor, main memory and I/O module.

Two main registers present in processor

MAR - Memory addressing register

MBR - Memory buffer register

MAR - specifies the addressing memory for the next read and write.

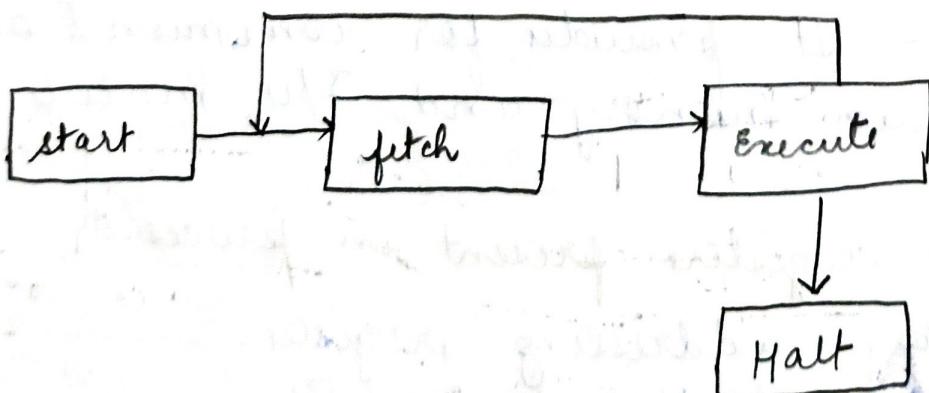
MBR contains the data to be written into the memory or receives the data read from memory.

Microprocessor - It is the fastest general purpose processor where a single processor is present on single chip. However each chip can contain multiple processor.

GPU - They provide efficient computation on a large amount of data using SIMD (single instruction multiple data) technique used for general numerical processing.

DSP - Digital signal processor. They deal with streaming signals such as audio or video used in embedded system like . They provide support for security through encryption technique.

SOC (System on a chip)
To satisfy the need of handheld devices the microprocessor is giving way to SOC. Components such as DSP, GPU, render and main memory in addition to processor are on the single chip.



Instruction Execution
A program consists of set of instructions stored in memory.

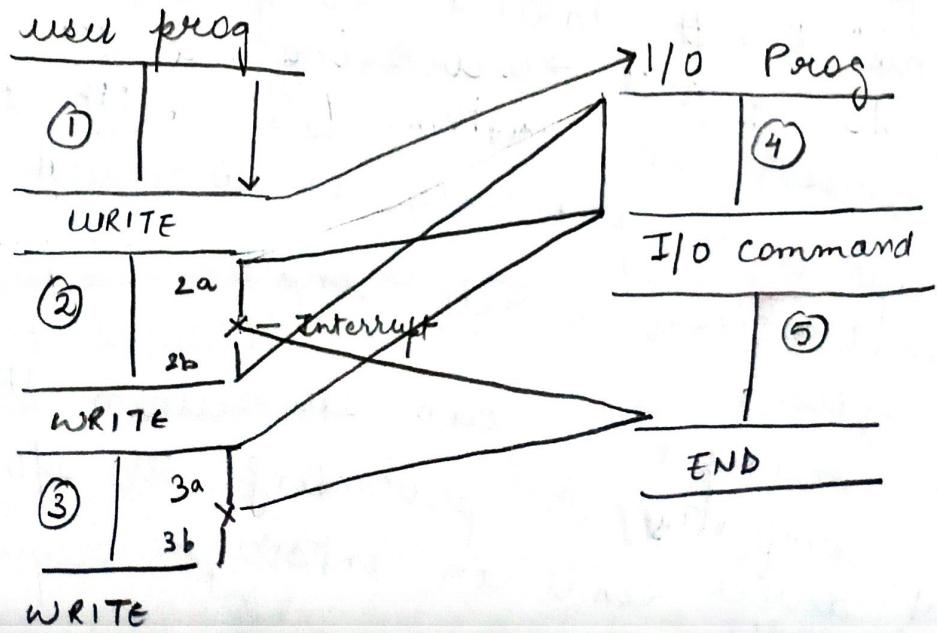
Instruction processing consist of 2 steps.

- i) Fetch
- ii) execute

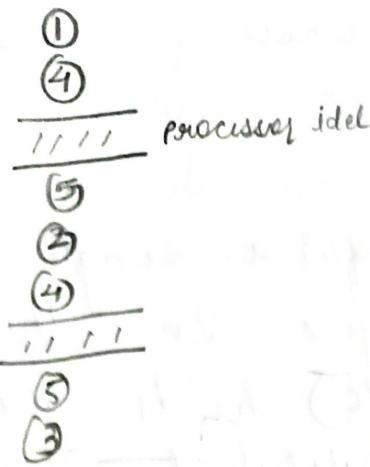
The processing require for a single instruction is called instruction cycle. In the instruction cycle Program counter (PC) holds the address of next instruction to be fetched. The processor always increments the PC after each instruction fetch so it will fetch the next instruction in sequence. The fetched instruction is loaded in instruction register. It, ^{Instruction} contains the bits that specifies the action processor has to take.

4 types of action

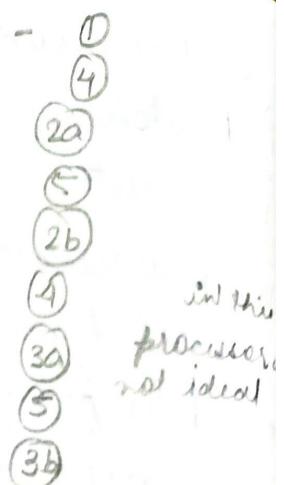
- i) Processor memory - Data may be transferred from processor to memory or vice versa.
- ii) Processor I/O - Data may be transferred to or from and peripheral device.
- iii) Data processing - The processor may perform arithmetical or logic logical operation on data
- iv) Control - Any instruction may specify the sequence of execution to be altered.



without interrupt -



with interrupt -



The mechanism is provided by all computers by which other modules may interrupt the normal sequencing of the processor. They are provided as a way to improve processor utilization.

There are 4 class of interrupt (Book)

with interrupt

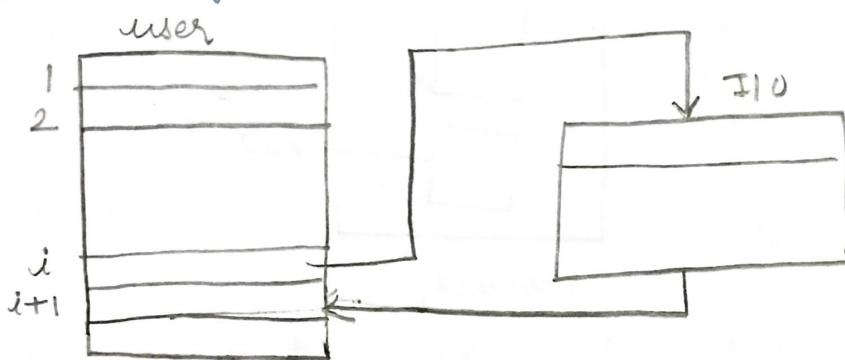
The user program performs the a series of end with processing code segments ①, ② and referred to instruction that don't need I/O. The ~~right wait cob calls are~~ are 2 and I/O routine which is system utility. The I/O program consist of 3 set instruction i) the preparation code to prepare for actual I/O ii) actual I/O command iii) the instructions to complete the operation.

without interrupt

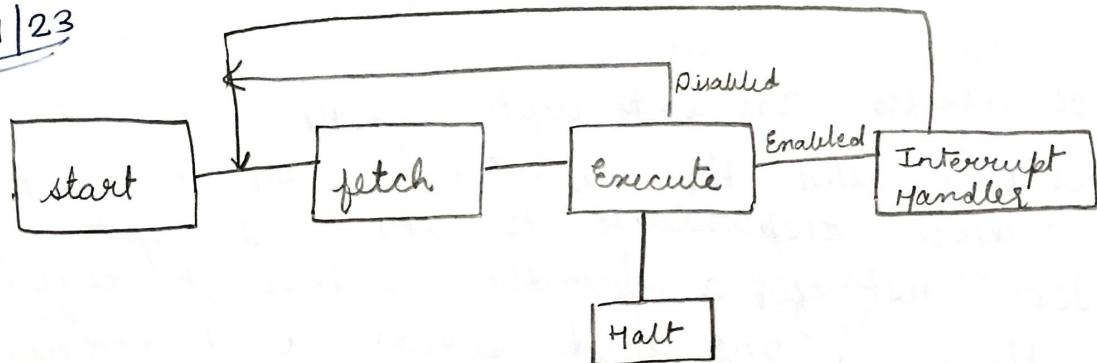
The processor has to wait until the I/O routine is finished. It takes a long time for the external device to finish processing until then the time of processor is wasted but with the use of interrupt the processor is engaged in executing other instruction while input-output operation is in progress.

In the example we can see when the external device completes its processing the I/O module for external device sends an interrupt request to signal the processor.

The I/O program consists of preparation code and actual I/O command. This way processor works concurrently with the I/O program & improves its utilisation.

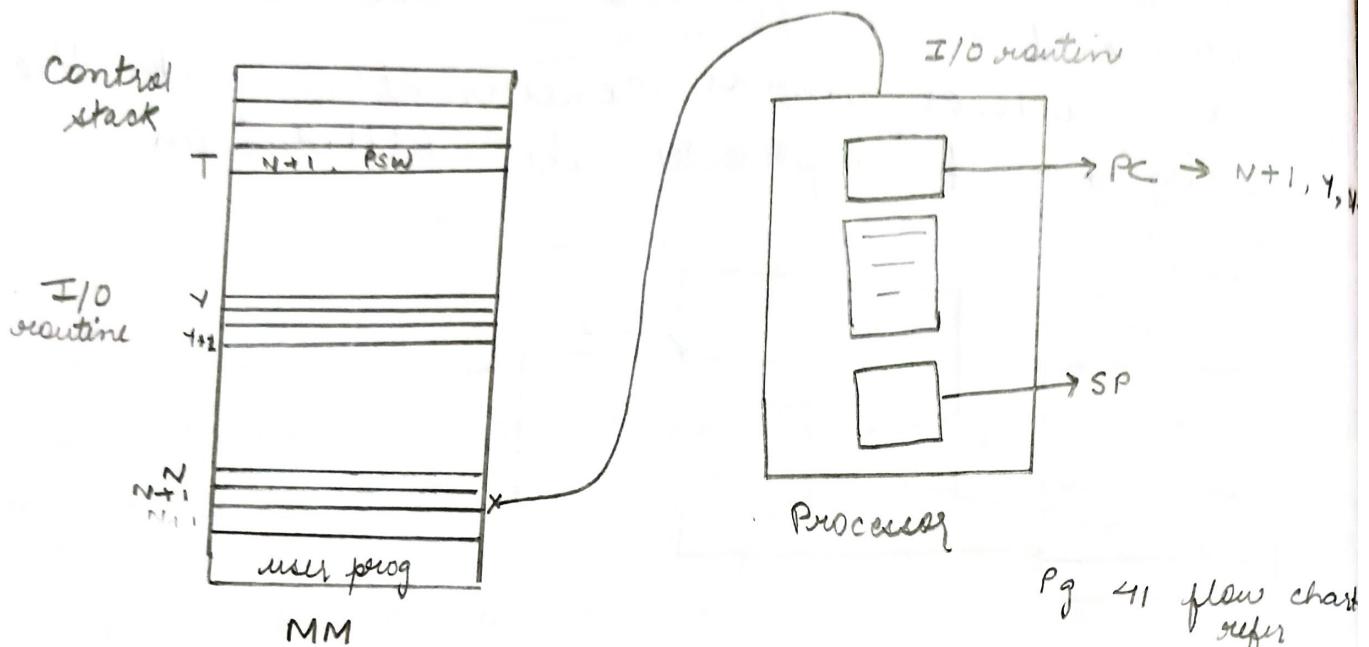


13/9/23



The user program does not have to contain any special code to accommodate interrupts. The processor and the OS are responsible for suspending the user program and then resuming it from the same point to accommodate the interrupt. An interrupt stage is added to the instruction cycle. In the interrupt stage, the processor checks to see if any interrupt has occurred. The interrupt signals are indicated by the presence of interrupt signals. If no interrupt is pending, the processor proceeds to the fetch stage and fetches the next instruction of the current program. If an interrupt is pending, the processor suspends the execution of the current program and executes the interrupt handler routine. This routine determines the nature of the interrupt and performs

whatever actions to be taken.



Pg 41 flow chart
refer
interrupt process

- The device issues an interrupt signals
- The processor finishes the execution of the current instruction before responding to the interrupt
- The processor test for a pending interrupt request determine there is one and sends an acknowledgement signals to the device that has issued an interrupt
- Now processor needs to prepare to transfer control to the interrupt routine. It saves the information needed to resume the current program which are PSW (program status word) and the location of next instruction to be executed. The processor then loads the program location of the interrupt handler routine that will handle the interrupt. Once the PC has been loaded next instruction cycle starts from interrupt handler routine. Now interrupt handler routine would need the general purpose registers so it starts by saving the all registers to the stack. The interrupt handler now proceed to process the interrupt. When interrupt processing is complete the saved val

is taken from control stack.

- The final act is to restore the value of PC and the program counter values from the stack

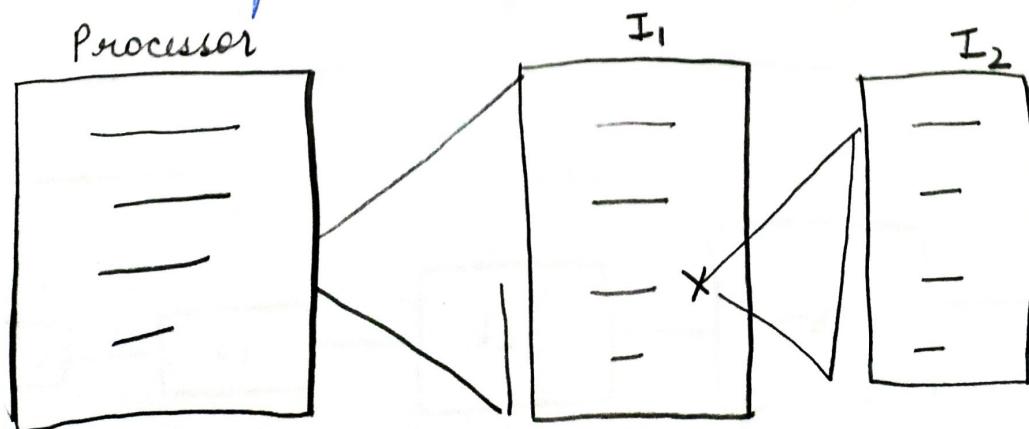
Multiple interrupts

One or more interrupts can occur while an interrupt is being processed. There are 2 approaches that can be taken in account to deal with multiple interrupt

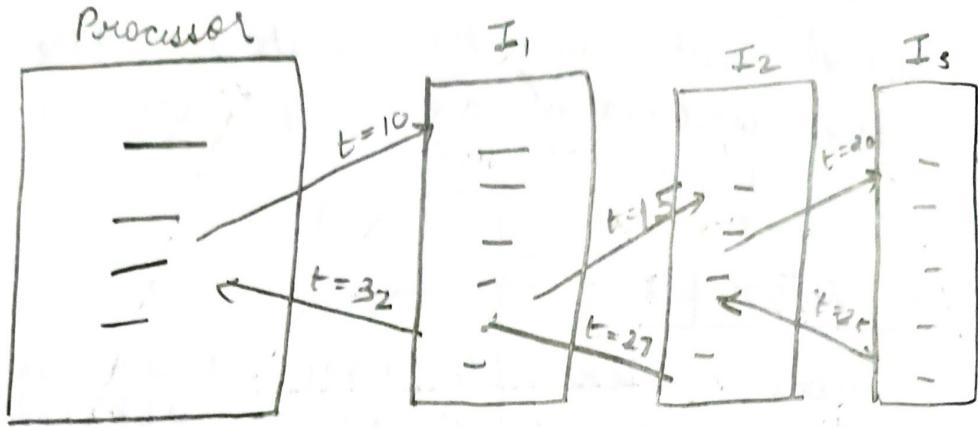
- To disable the interrupt when an interrupt is being processed. A disabled interrupt means that the processor will ignore any new interrupt signals if an interrupt comes during this time it will remain pending and will be checked by processor after the processor has enabled the interrupt. Thus if an interrupt occurs when a user program is executing the interrupt is disabled immediately.

Drawback

The drawbacks to the approach is it do not take in account relative priority and time critical needs to account. Eg:- when an input ~~arise~~ arise from communication its priority is higher if it needs to be serviced immediately but as interrupts are disabled it has to wait for a very long time until the previous interrupt has completed its execution

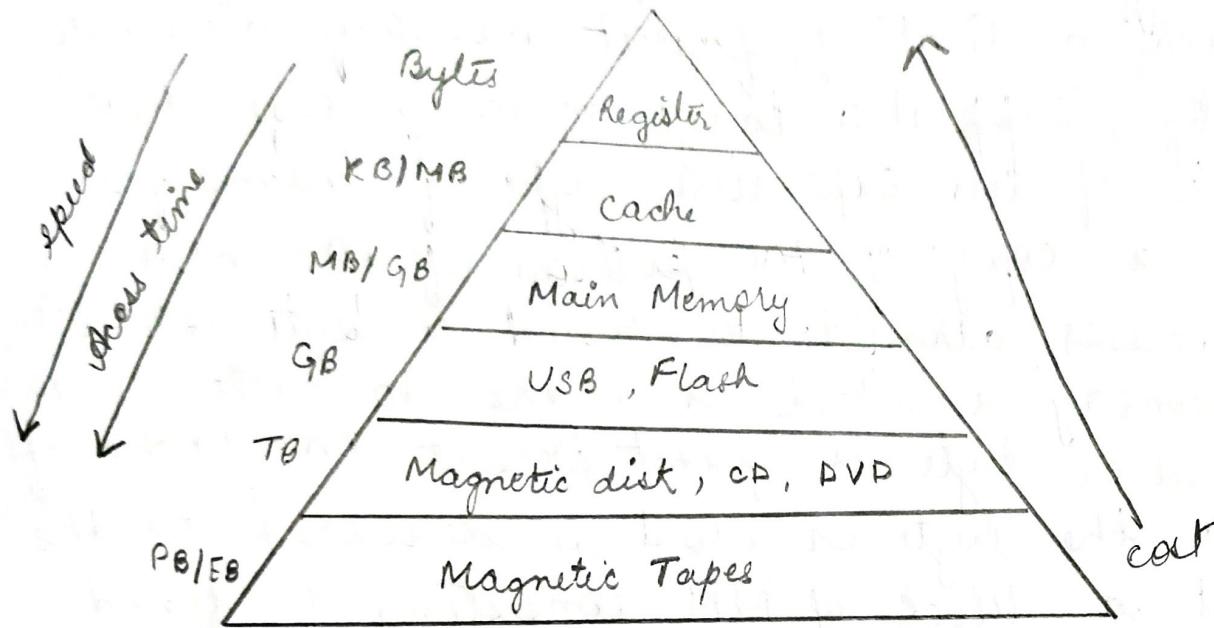


Priority	
	$t=0, 40$
2	$t=10, 10$
4	$t=15, 7$
5	$t=20, 5$

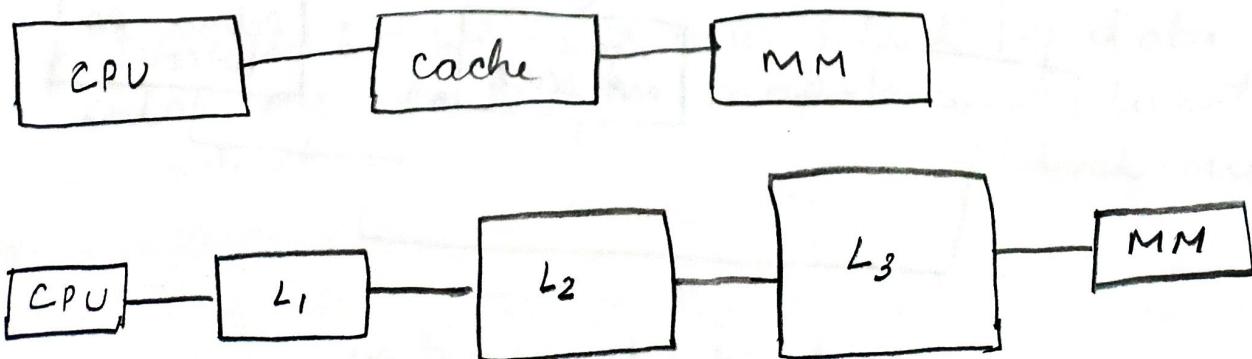


Suppose there are three I/O device a printer, a disk and a communication line with the priorities 2, 4, a user program begins at $t=0$, at $t=10$ a printer interrupt occurs the user information is stored in control stack & execution continues at the printer interrupt routine. while this routine is still executing another routine from communication line has higher priority the interrupt request of this I/O device is serviced at $t=20$ as the communication line has higher priority arises so the processor service the interrupt request of this device. when all the interrupts are handled previous process state is restored which is the execution of printer interrupt. After the interrupt of printer is completed the execution of user program continues.

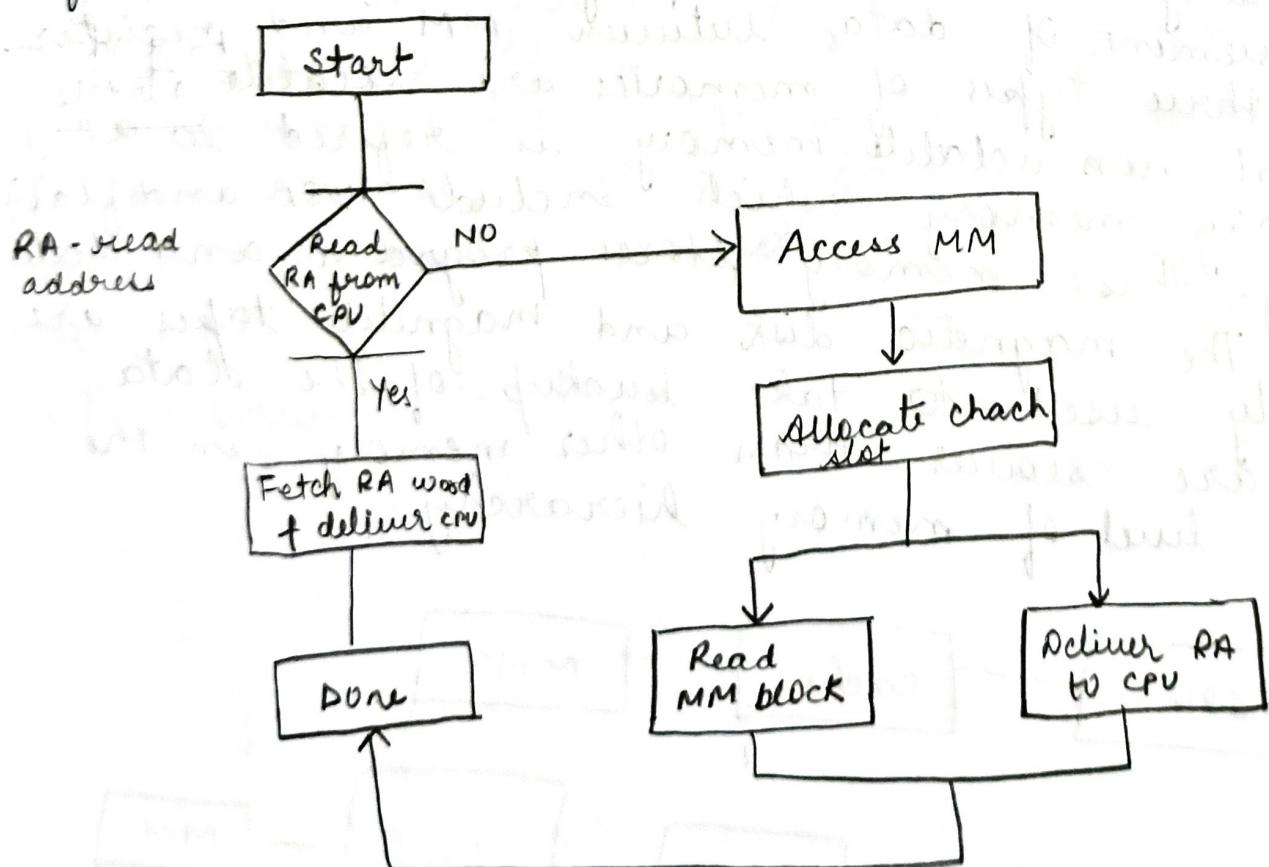
15/9/23



The computer system is divided in the form of levels. The smaller more expensive faster memories are supplemented by larger cheaper and slower memories. The smallest, fastest and the most expensive type of ^{memory} consist of register. Skipping to 2 levels MM is the principle internal memory of the computer. The cache is usually visible to the programmer for staging the movement of data between MM and registers. These three types of memories are volatile. External non volatile memory is referred to as secondary memory which includes USB and flash memory. These memory stores program and data files. The magnetic disk and magnetic tapes are usually used to take backup of the data and are slower than other memory in the above level of memory hierarchy.



Cache memory is intended to provide memory access time approaching that of fastest memory available. And secondly support a large memory size that has the price of less expensive type of memory. It contains a copy of the portion of the MM. When the processor attempts to read a byte or a word of memory a check is made to determine if the word or byte is first present in cache. If it is there the byte or word is delivered to the CPU, if not a block of MM consisting of fixed words is read into the cache then the byte or word is delivered to the processor because of concept of locality of reference. When a block of data is fetched into the cache to satisfy the single memory reference. It is likely the near future memory references will be to other bytes of the block.



Cache Read Operation

The figure illustrates the read operation. The processor generates the address (RA) of a word to be read if the word is contained in the cache it is delivered to the processor otherwise the block containing that word is loaded into cache and the word is delivered to the processor. Now there are design issues that must be addressed in dealing with virtual memory and cache design.

Design issues :-

1. Cache size
2. Block size
3. Replacement
4. Mapping function
5. Write policy

Cache size :- According to hierarchy small caches have significant impact on performance. Cache is closer to the CPU hence the size of cache is usually smaller.

Block size It is unit of data exchanged between cache and main memory.

Replacement Algo. ; - It chooses which block to replace when a new block is to be loaded into the cache and cache ~~are~~ already has all slots filled with other blocks. One of the way is replacing the block ie that is least likely to be needed again in the near future such as LRU (least

Mapping Function ; - When a new block of data is read into the cache the mapping function determines which cache location the block will occupy.

Mapping function If the contents of a block in cache are altered then it is necessary to write it back to the MM before replacing it. One way is writing can occur every time when the block is updated. Other way is writing occurs only when the block is replaced.

No. of cache level :- Multiple level of cache provides fast access time to the CPU and enhances the performance.

DATA Techniques to execute I/O operations

When the processor is executing a program & encodes an instruction related to I/O. Three techniques are possible for I/O of

- i) Programmed I/O
- ii) Interrupt driven I/O
- iii) Direct memory access

Programmed I/O

In case of programmed I/O the I/O module performs the requested action & sets the respective bits in I/O status register and takes no further action to alert the processor. The processor periodically checks the status of I/O module until it finds the operation complete.

Interrupt driven I/O

It is more efficient than programmed I/O and still requires active intervention of the processor to transfer data between memory & I/O module. Its drawback is the processor is busy in managing I/O transfer & its utilisation in running current program decreases. So, the solution to this problem is DMA.

When large volumes of data is to be moved DMA is more efficient technique. The DMA function can be performed

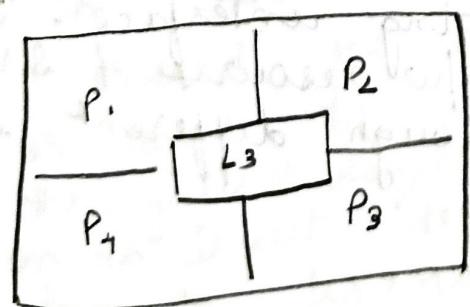
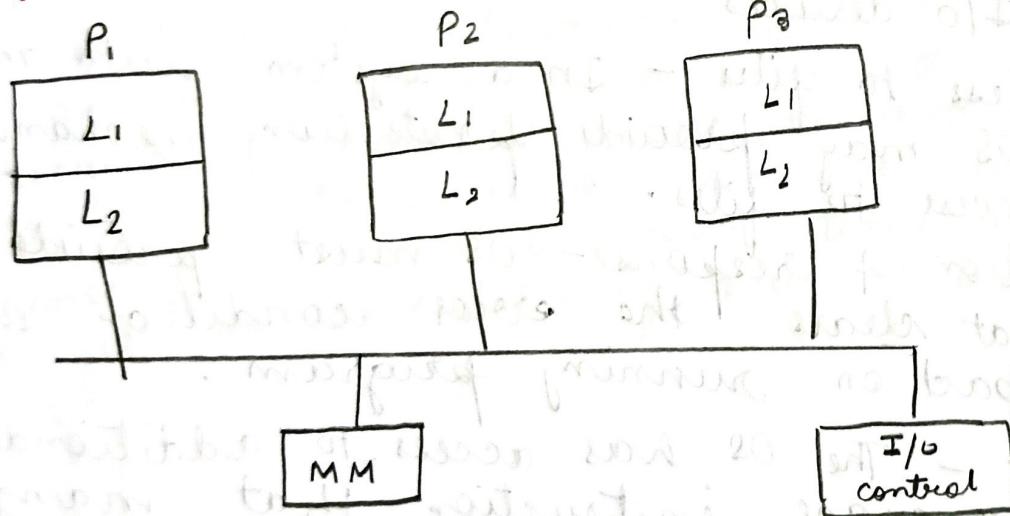
key a separate module or it can be incorporated or combined with I/O module. The processor issues a command to the DMA module by sending these four information :-

- i) whether read or write is requested
- ii) The address of I/O device is involved
- iii) The starting location in memory to read data from and write data to
- iv) The no. of words to be read or to be written

The processor then continues ^{with} the other work

20/9/23

SMP :-



Multicore Processor

19/9/23 CH:2 - Operating System Overview

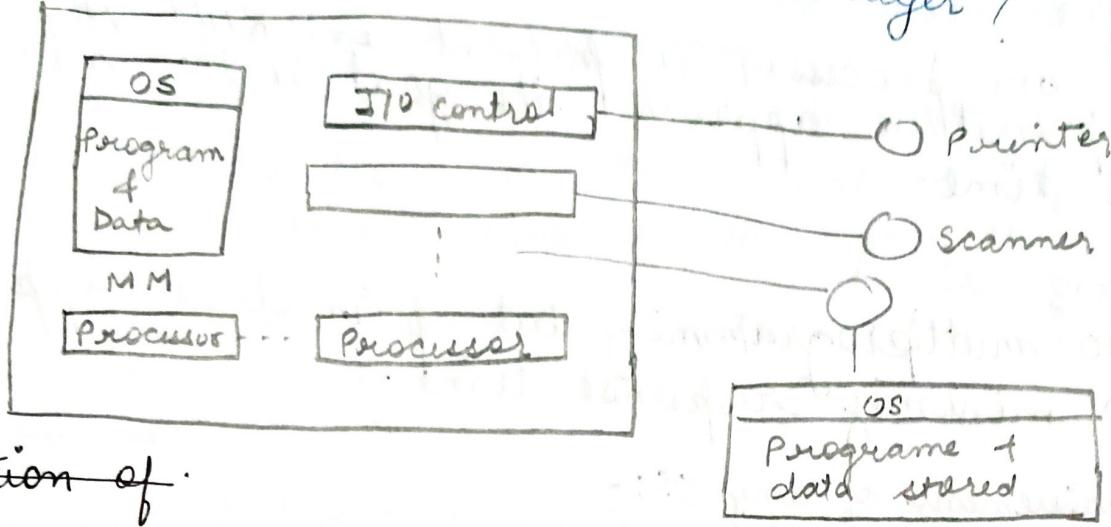
Goals

- i) Convenience
- ii) Efficiency
- iii) Ability to evolve

How OS acts as user interface

- i) Program development provides services of editor, debugger
- ii) For the execution of program instruction and data much be loaded I/O devices & file must be initialized & other resources must be prepared & handled by OS
- iii) Access to I/O devices
- iv) Control access to files - In a system with many users the OS may provide protection mechanism to control access to files.
- v) Error detection & response - OS must provide a response that clears the error condition with least impact on running program.
- vi) ISA support - The OS has access to additional machine language instructions that manage system resources
- vii) API - application programming interface. It gives a program access to the I/W resources & services available in the system through different service calls.

Q) How OS acts as resource manager?



A portion of:

A portion of OS is in MM which includes the Kernel which contains the most frequently used function in the OS. The remainder of MM contains user and utility programs and data. The OS and the MM management hardware h/w in the processor, jointly controls the allocation of MM. The OS decides when an I/O device can be used by a program in execution & controls access to and use of files. The processor itself is a resource and OS determines how much processor time is to be given for the execution of any program.

Types of OS

i) Batch OS :-

In batch OS the user has no direct access to the processor instead the programs are submitted on cards or tapes to a computer operator who batches the job sequentially & places an entire batch on an input device each program is constructed to branch back to the monitor when it completes processing at which point the monitor automatically loads the next program. The programs are executed in system mode or kernel mode.

ii) Multiprogramming OS

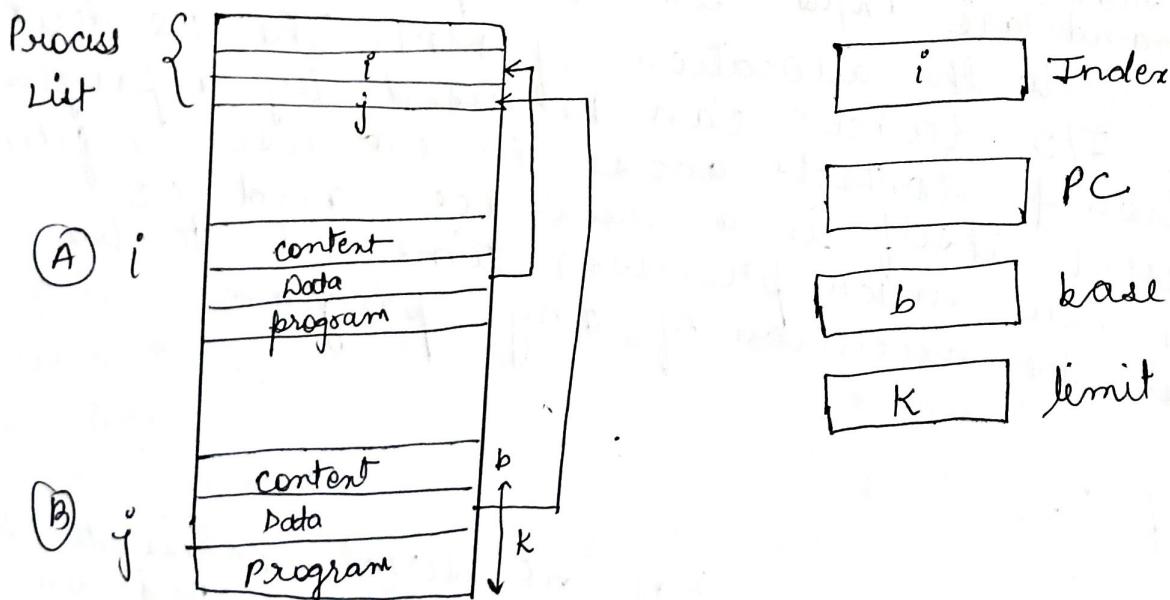
More than one processor is present in RAM this is a non primitive approach the goal is to minimize CPU ideal time.

iii) Multitasking OS

Similar to multiprogramming but primitive approach goal is to minimize response time.

Major achievements of any OS:-

- 1) Processes
- 2) Memory management
- 3) Information & security measures
- 4) Scheduling



i) Process :- A program in execution or the instance of a program currently running on the computer.

In the dig two processes A & B exist in portion MM each process is recorded in a process list built and maintained by OS. There are 3 components of a process:-

- i) Executable program

ii) Associated data

iii) Execution context of the program

The process list contains 1 entry for each process which includes a pointer to the location of the block of memory that contains the process.

The process index register contains the index into the process list of the process currently controlling the processes.

The base & limit registers define the region in memory occupied by the process. The base register is the static starting address of the region of the memory & limit is the size of region

Q i) Process isolation

ii) Automatic allocation & memory management

support of modular programming.

iv) Long term storage

v) Protection & access control

A ₁		B ₅
B ₁	A ₂	B ₃
	A ₃	
		B ₂
A ₄	A ₅	

MM

non contiguous
memory location
(not in order)

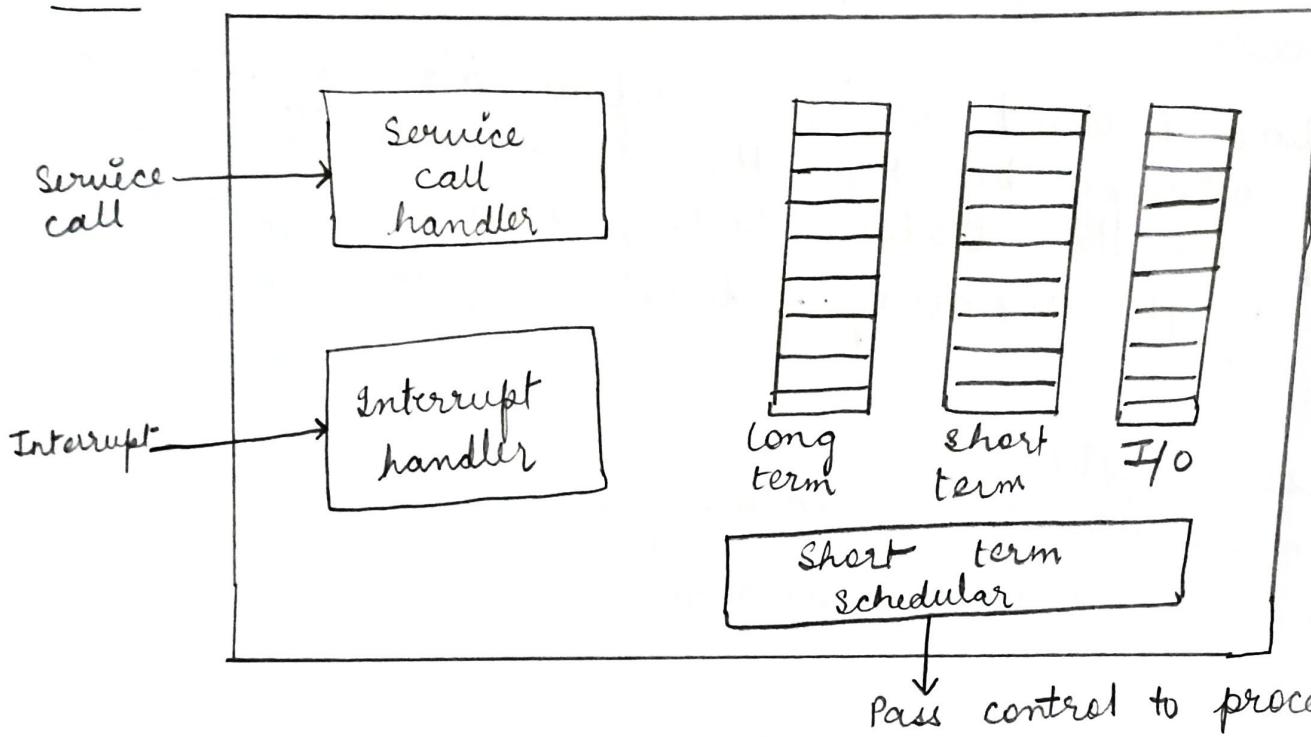
A	B
1	1
2	2
3	3
4	4
5	5

SM (secondary memory)

Processes consist of fixed size block called pages which are stored in secondary memory. MM consist of no. of frames each equal to the size of the page for a program to execute some of all pages should be in MM.

20/9/23

Scheduling and resource management



The OS maintains a number of queues that contain a list of processes waiting for some resource. The short term queue consists of processes that are in main memory and are ready to run as soon as processor is made available. The long term queue is the list of new jobs waiting to be processed. The OS adds the jobs to the system by transferring a process from the long term queue to the short term queue. There is a queue for each I/O device more than one process may request the use of same I/O device. The interrupt handler receives control of processor if the interrupt is raised or if a service call is initiated from any application.

- Information + security
- i) Availability
 - ii) Confidentiality
 - iii) Data integrity
 - iv) Authentication

Development that lead to modern OS :-

- i) Microkernel architecture (explanation from book)
- ii) Multithreading
- iii) SMP
- iv) Distributed environment
- v) Object oriented design

Features of SMP :-

- i) Performance
- ii) Incremental growth
- iii) scaling
- iv) Availability - even after one processor shuts down

Faults :-

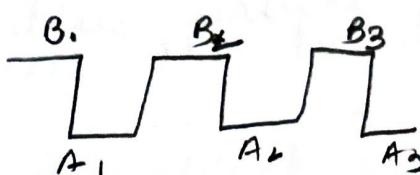
Fault is a hardware or software state resulting from component failure. Operator error, design error, data structure error or physical interference from the environment.

There are 2 types of error :-

- i) Permanent
- ii) Temporary

Fault tolerance :-

- i) Reliability
- ii) MTTF - ~~minimum~~ ^{mean (avg)} time to failure
- iii) MTTR - ~~minimum~~ ^{mean} time to repair



A - downtime
B - up time

$$MTTF = B_1 + B_2 + B_3 / 2$$

$$MTTR = A_1 + A_2 + A_3 / 3$$

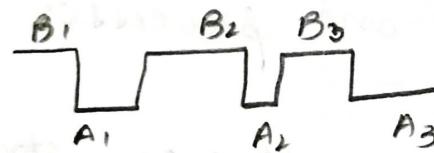
Fault tolerance & tolerance refer to availability of system to continue normal operation despite presence of h/w, s/w faults

Three basic measures are:-

Reliability - It is the probability of current operation upto time t given that the system was operating correctly at $t=0$;

MTTF - It is the average time available of the system when it was executing correctly

MTTR - The average time it takes to repair or replace a faulty element. The time during which system is not available is called downtime when system is available it is up time.



Design consideration for multiprocessor:-

A multiprocessor OS must provide all the functions of multiprogramming system. Some of the design considerations are:-

- i) Simultaneous concurrent processes or threads
- ii) Scheduling
- iii) Synchronisation
- iv) Memory management - It must be guaranteed that old data can no longer be accessed & needs to be updated if changed by any process
- v) Reliability & fault tolerance.

26/9/11 Multicore OS Consideration

i) Parallelism

- Many applications can be subdivided into multiple task that can execute in parallel. It is the duty of developer how to split up the application task into independent executable task.

ii) GCD is multicore support capability in which the OS maps the tasks onto the threads.

iii) Virtual machine approach.

One or more cores are dedicated to a particular process. Then processor denotes all its effort to the process. It is the duty of OS that assigns an application a processor and some memory and the program itself using meta data generated by ~~compit~~ compiler.

Windows

Windows 1983



Kernal mode component

- 1) Executive Service
- 2) Micro Kernel
- 3) HAL
- 4) Device Drivers
- 5) Windows graphic interface



User

User mode component

- 1) Special processing
- 2) Environment subsystem
- 3) User application

Kernal mode

- 1) Executive service
It makes the low level of Kernal mode. There are different modules such as "I/O manager", "cache manager", "object manager", "security manager", "power", "virtual memory", & "process manager".

i) Microkernel

It is the heart of the kernel and it is central element of new technology. Its primary task is to schedule all active system threads while maintaining a procedure operating environment at its highest performance level.

ii) HAL (Hardware abstraction layer)

It maps between generic hardware commands and responses of those unique to a specific platform.

iv) Device Drivers

These include hardware device drivers that translate user I/O function calls into specific hardware device I/O request.

v) Windows graphic interface

Implements the GUI functions such as user interface controls, user mode.

i) Special system processes

They manage the systems such as session manager, authentication subsystem, service manager and log in process.

ii) Environment subsystems

It includes the subsystem process. That deals with responsiveness of application.

iii) User application

Executable (.exe) DLL's that provides the functionality users run to make use of systems.

Client-Server Model

A client which can be the application program or another server program request a service by sending a message. The message is routed through executive to the appropriate server. Here server is subsystem of process that provide the services. The server perform the requested operation & returns the results or status information by means of another message ~~the~~ which is routed through the executive back to the client.

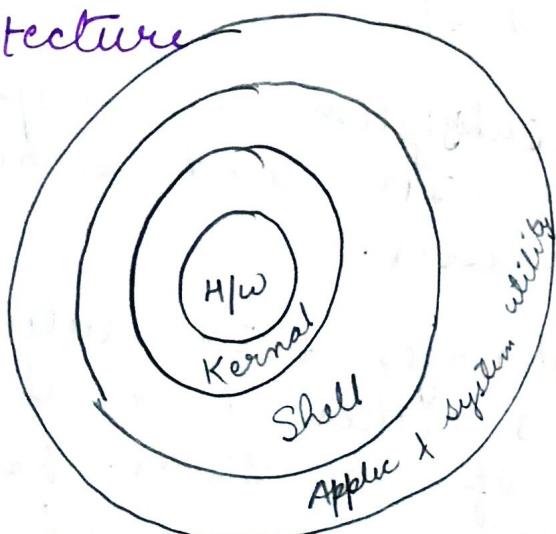
Advantage

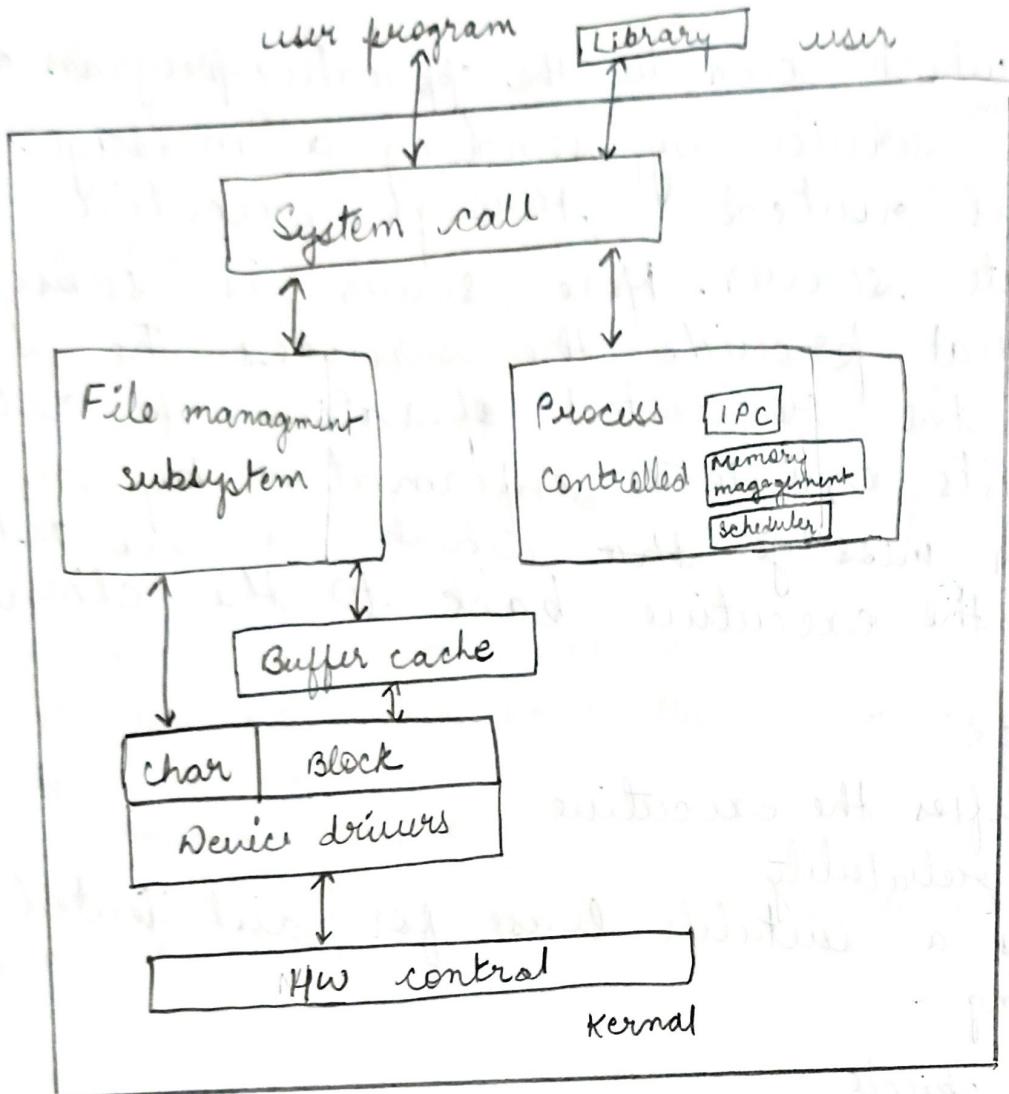
- i) It simplifies the executive
- ii) Improve reliability
- iii) Provides a suitable base for distributed computing -

Windows objects

- i) Encapsulation - The only way to access the data in an object by invoking one of the object services
- ii) Inheritance
- iii) Object class and instance
- iv) Polymorphism

Unix Architecture





H/w level

User programs can invoke OS services either directly or through library programs. The system call interface is the boundary with the user and allows software to gain access to specific kernel functions. System has 2 main parts

i) Process control

ii) File management and I/O

The process control subsystem is responsible for memory management, IPC and scheduling and dispatching of processes.

The file system exchanges data between memory and external devices either as a string of character or ~~or~~ ^{log} of in blocks. For this variety of device drivers are used.

- Modern unix system
- i) SVR4
 - ii) BSD
 - iii) Solaris 11

Linux

Linux follows modular structure. The modules can be loaded and unloaded on demand.

They have two main characteristics:

- i) Dynamic linking
- ii) Stackable modules

Components of Linux:-

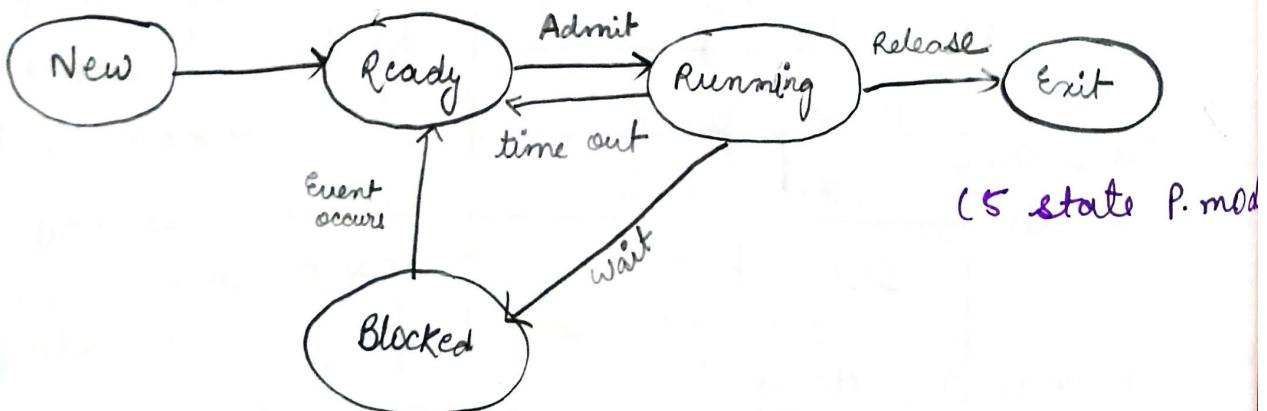
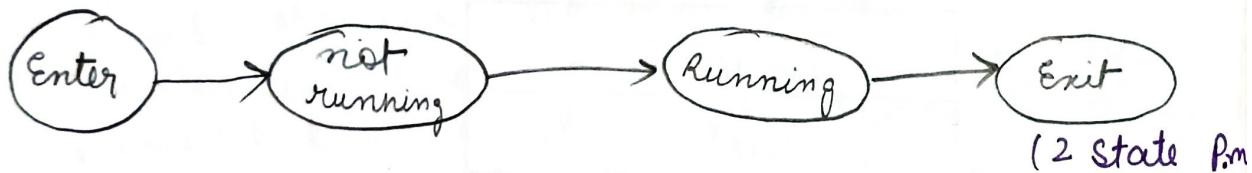
- i) Signals - The kernel uses signals to call into a process
- ii) System call - It is the means by which a process requests a specific kernel service.
- iii) Process & scheduler
- iv) Virtual memory

Process & process control block:-

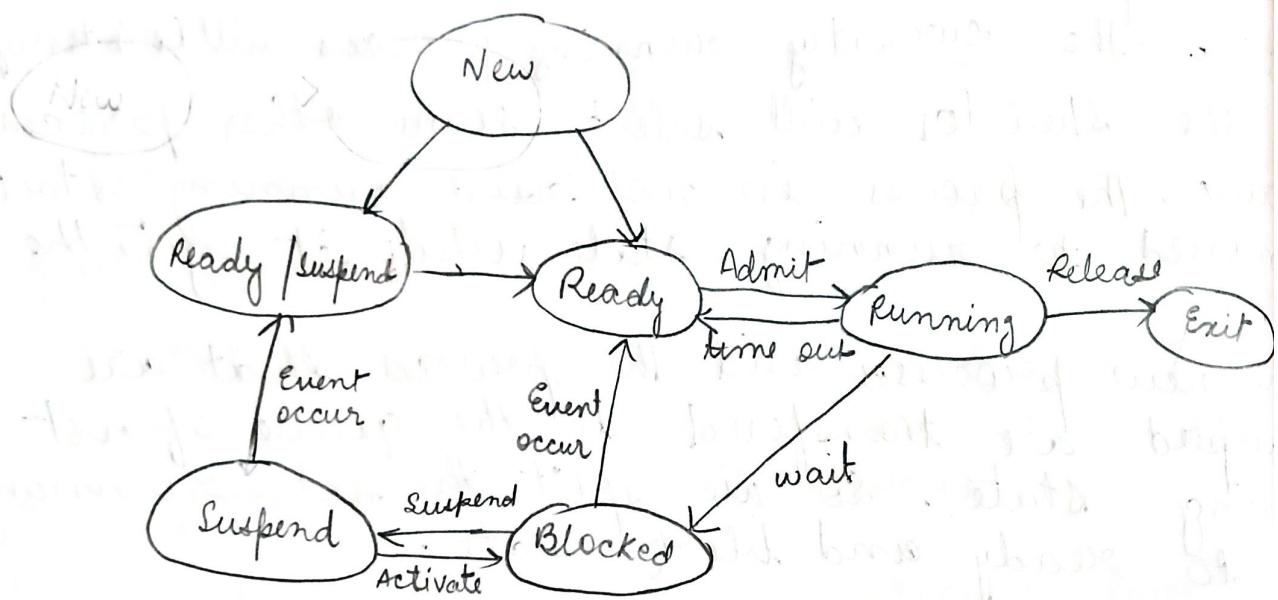
Identifier
state
Priority
Memory pointer
context data
I/O status information
Accounting information
PC

The process can be uniquely characterised by a number of elements.

- i) Identifier - unique id associated with process
- ii) The state of the process
- iii) Priority - level relative to other processes
- iv) Memory pointers - pointers to the programme code & data associated.
- v) Content data - Data that are present in registers while the process is executing
- vi) I/O status info. - includes outstanding I/O request. I/O device assigned to the process a list of files is used by the process
- vii) Accounting info - Amount of processor time & clock time or time limits for this process
- viii) PC - The address of next instruction



* While a process is in new state. Info. reg. the process is maintained in control table in MM. However the process itself is not in MM.



The processor is much faster than I/O that will be common for all the processes in memory waiting for I/O. Hence processor will be idle for most of the time. When all the processes in MM are in blocked state. The OS can suspend one process by putting it in suspended state. Transferring it to secondary memory.

Read - The process is in MM & ready for exec

Block - The " " " " " awaiting an event

Suspend - The process is in secondary memory, awaiting an event

Ready / Suspend - The process is in secondary memory but is available for execution as soon as it is loaded in MM.

Two state process Model

when OS creates a new process. It creates a process control block for the process & enters that process into the system into not-running state. The process exists and it waits for its turn to execute. From time to time the currently running process will be interrupted and the scheduler will select some other process to run. The process in not over running starts turn.

All the new processes and the processes that are interrupted are transferred to the queue of not-running state. So we split the not running state to ready and block state.

5-state model Process model

New - A process that has just been created but has not been admitted into the pool of executable processes by OS. It means the process is not loaded in the main memory but its PCB is created (Process control block) (process)

Ready - A process that is prepared to execute when given the opportunity

Running - The process that is currently being executed.

Block - A process that cannot execute until some event occurs such as completion of I/O.

Exit - A process that has completed the execution & have been released by the process

Process creation

When a new process is added to those currently being executed the OS builds the data structure used to manage the process and allocates address space in the MM to the process.

Reasons:-

- i) New application or new batch job is created
- ii) Created by OS to provide a service.
- iii) Process spawning - A currently managed process can create a new process.
- iv) Interactive log-on - A user in terminal log-on(s) to the system

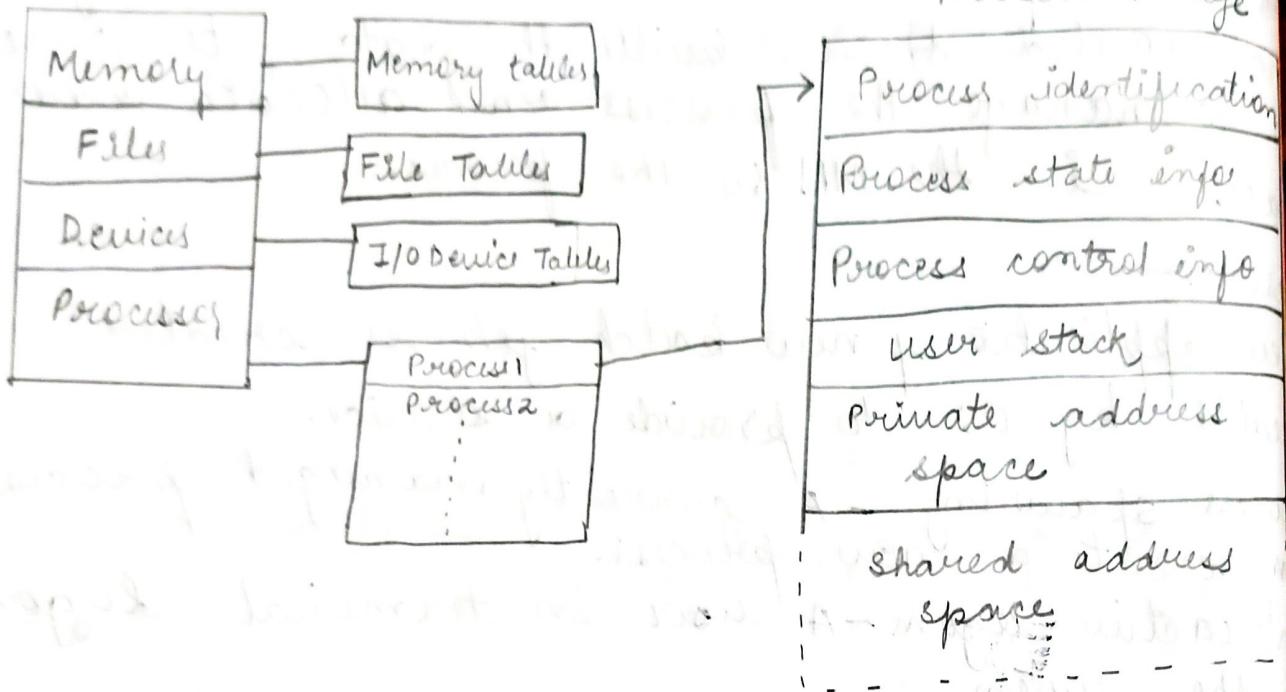
Process termination :-

Reasons:-

- i) Normal completion
- ii) Requires more memory than the system can provide
- iii) Bounds violation - The process tries to access a memory location that is not allowed to access
- iv) Arithmetic error
- v) I/O failure
- vi) Invalid instruction - The process attempts to execute a non-existent instruction.
- vii) Parent process termination (parent terminates so child also terminates)
- viii) Time limit exceeded

29/9/23

Process distribution



If OS wants to manage processes & resources must have info about current status of each process and resource. The OS manages this in four categories:-

- i) Memory tables - These are used to keep track both main & secondary memory. The memory t include following info.
 - a) allocation of MM to processes
 - b) " of secondary memory to processes
 - c) Any information needed to manage virtual m
 - d) Any protection attributes of blocks of MM
- ii) File Tables - provide information about the existence of files. Their location on secondary m & their current status.
- iii) I/O tables - are used to manage the I/O devic & channels of computer system. The OS need know the status of I/O operation & the locatio in MM being used as source or destination of

I/O transfer.

- iv) Process tables - are used to manage process. There is a primary process table with one entry for each process. Each entry contains a pointer to process image. Process image consist of
- Process identification - unique process identifier.
 - Processors info - It consists of content of processor register which contain the information of a running process in the registers.
 - Process control information - This consist of different flags such as:- resume flag, interrupt enable flag, interrupt all flag and others to control the process.
 - Private and shared address space - It also consist private & shared address space.

Scheduling Algorithm

Primitive

Non- Primitive

SRTF shortest remaining time first

SFCFS first come first serve

LRTF largest " "

SJF shortest job first

*Priority level based

LTF largest "

*Round - Robin

Multilevel Queue

HRRN highest response ratio next

Selecting a process from ready queue and putting it on processor for running. There are multiple processes in ready queue which is present in MM.

1) Arrival time start time

2) Burst time time to complete the processes (inside process)

3) completion time Total time to complete the process (also include I/O process time)

- 4) Turn around time \rightarrow Completion time - arrival time
 5) waiting time \rightarrow TAT - burst time
 6) Response time \rightarrow first time when process gets to the CPU) - Arrival time

Arrival time - The time at which the process enters a ready queue or

Burst time - Time required by a process to get executed on CPU (time duration)

Completion time - Time at which process complete execution

Turn around time = completion time - arrival time

Waiting time - TAT - burst time

Response time - first time the process gets the CPU - arrival time

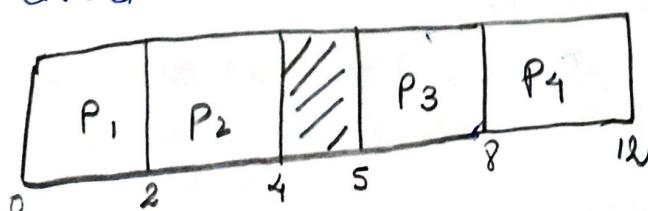
3/10/23

FCFS

in non primitive WT =

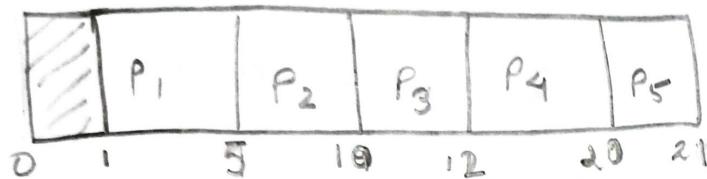
P NO	AT	BT	CT	TAT	WT	RT
P ₁	0	2	2	2	0	0
P ₂	1	2	4	3	1	1
P ₃	5	3	8	3	0	0
P ₄	6	4	12	6	2	2

Gantt Chart



$$\text{Avg TAT} = \frac{14}{4} = 3$$

PNO	AT	BT	CT	TAT	WT	RT
P ₁	1	4	5	4	0	0
P ₂	3	5	10	7	2	2
P ₃	4	2	12	8	6	6
P ₄	7	8	20	13	5	5
P ₅	8	1	21	13	12	12

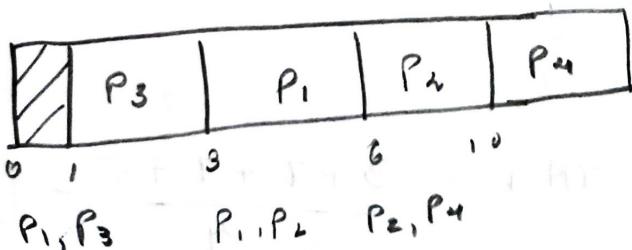


$$\text{Avg TAT} = \frac{4+7+8+13+13}{5} = \frac{45}{5} = 9$$

$$\text{Avg WT} = \frac{0+6+5+12+0}{5} = \frac{25}{5} = 5$$

PNO	AT	BT	CT	TAT	WT	RT
P ₁	1	3	6	5	2	2
P ₂	2	4	10	8	4	4
P ₃	1	2	3	2	0	0
P ₄	4	4	14	10	6	6

If burst time is same then use AT



$$\text{Average TAT} = \frac{5+8+2+10}{4} = \frac{25}{4} = 6.25$$

$$\text{Average WT} = \frac{0+4+0+6}{4} = \frac{12}{4} = 3$$

PNO AT BT CT TAT WT RT

P_1	2	3	5	3	0	0
P_2	0	4	12	12	8	8
P_3	0	2	2	2	0	0
P_4	3	3	8	5	2	2

P_3	P_1	P_4	P_2
0	2	5	8

P_2, P_3 P_1, P_2 P_2, P_4 P_3, P_4

$$\text{Avg CT} = \frac{5+12+2+8}{4} = \frac{27}{4}$$

$$\text{Avg TAT} = \frac{3+12+2+5}{4} = \frac{22}{4}$$

$$\text{Avg WT} = \frac{0+8+0+2}{4} = \frac{10}{4}$$

FIFO

SJT

PNO AT BT CT TAT WT RT

P_1	0	3	3	3	0	0
P_2	1	5	8	7	2	2
P_3	3	2	10	7	5	5
P_4	9	4	14	5	1	1

P_1	P_2	P_3	P_4
0	3	8	10

$$\text{Avg TAT} = \frac{3+7+7+5}{4} = \frac{22}{4}$$

$$\text{Avg WT} = \frac{0+2+5+1}{4} = \frac{8}{4}$$

P NO	AT	BT	CT	TAT	WT	RT
P ₁	0	3	3	3	0	0
P ₂	1	5	10	9	4	4
P ₃	3	8	15	12	10	10
P ₄	9	4	14	5	1	1

P ₁	P ₃	P ₂	P ₄
0	3	5	10

P₁ P₂ P₃ P₄

$$\text{Avg TAT} = \frac{3+9+8+5}{4} = 6.75$$

$$= \frac{19}{4} = 4.75$$

$$\text{Avg WT} = \frac{0+4+10+1}{4} = 3.75$$

FCFS

P NO	AT	BT	CT	TAT	WT	RT
P ₁	3	1	14	11	10	10
P ₂	1	4	10	9	5	5
P ₃	4	2	16	12	10	10
P ₄	0	6	6	6	0	0
P ₅	2	3	19	17	8	18

$$\text{Avg TAT} = \frac{1+7+6+16+17}{5} = 9.4$$

P ₄	P ₂	P ₅	P ₁	P ₃
0	6	10	13	14

$$\text{Avg WT} = \frac{10+5+10+0+8}{5} = \frac{33}{5} = 6.6$$

SJF

P NO	AT	BT	CT	TAT	WT	RT
P ₁	3	1	7	4	3	3
P ₂	1	4	16	15	11	11
P ₃	4	2	9	5	3	3
P ₄	0	6	6	6	0	0
P ₅	2	3	12	10	7	7

P ₄	P ₁	P ₃	P ₅	P ₂
0	6	7	9	12

$$\text{Avg TAT} = \frac{4+15+5+6+10}{5} = \frac{40}{5}$$

P ₄	P ₁ , P ₂ , P ₃ , P ₅	P ₂ , P ₃ , P ₅	P ₂ , P ₅	P ₂
0	6	7	9	12

$$\text{Avg WT} = \frac{3+11+3+7}{5} = 4.8$$

10/23

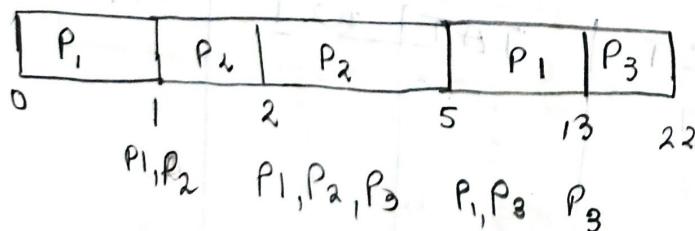
SRTF

It is a primitive SJF that will prevent the currently executing process whereas a primitive SJF will allow currently running process to finish its CPU time or CPU burst.

PNO	AT	BT	CT	TAT	WT	RT
P ₁	0	5843	9	9	4	0
P ₂	1	3210	4	3	0	0
P ₃	2	4	4	3	0	0
P ₄	4	01	13	11	7	7
			5	1	0	0



PNO	AT	BT	CT	TAT	WT	RT
P ₁	0	88	13	13	5	0
P ₂	1	43	5	4	1	0
P ₃	2	9	22	20	11	11



$$\text{Avg TAT} = \frac{13 + 4 + 20}{3} = \frac{37}{3} = 12.3$$

$$\text{Avg WT} = \frac{5 + 1 + 11}{3} = \frac{17}{3} = 5.6$$

same using SRTF

PNO	AT	BT	CT	TAT	WT	RT
P ₁	1	5 4 3 2	7	6	1	0
P ₂	2	6	20	18	12	12
P ₃	2	4	14	12	6	6
P ₄	4	1	5	1	0	0
P ₅	6	3	10	4	6	1

	P ₁	P ₁	P ₁	P ₄	P ₁	P ₁	P ₅	P ₃	P ₃	P ₂
0	P ₁	P ₁ , P ₂ , P ₃	P ₁ , P ₂ , P ₃ , P ₄	P ₁ , P ₂ , P ₃ , P ₄	P ₁ , P ₂ , P ₃ , P ₅	P ₁ , P ₂ , P ₃ , P ₅	P ₂ , P ₃ , P ₅	P ₂ , P ₃	P ₂	20

$$\text{Avg WT} = \frac{2+15}{5} \quad \text{Avg TAT} = \frac{6+18+12+1+4}{5} = \frac{41}{5} = 8.2$$

Priority (priority scheduling priority)

Higher NO, higher priority

(same priority then see arrival time)

Priority	PNO	AT	BT	CT	TAT	WT	RT
2	P ₁	0	4 3	15	15	11	0
3	P ₂	1	3 2	12	11	8	0
4	P ₃	2	1 0	3	1	0	0
5	P ₄	3	5 / 4 3	8	5	0	0
5	P ₅	4	2	10	6	4	4

	P ₁	P ₂	P ₃	P ₄	P ₄	P ₅	P ₂	P ₁
0	1	2	3	4	8	10	12	15

P₁, P₂, P₃ P₁, P₂, P₄ P₁, P₂, P₄, P₅

In this technique a priority is associated to the process and CPU is allocated to the process with highest priority. Equal priority processes are scheduled in FCFS

Priority	P NO.	AT	BT	CT	TAT	WT	RT
1	P ₁	0	4	16	16	12	0
4	P ₂	1	5, 8	9	8	7	0
2	P ₃	3	1	13	10	9	9
3	P ₄	4	3	12	8	5	5
5	P ₅	5	3	18	3	0	0

P ₁	P ₂	P ₂	P ₂	P ₂	P ₅	P ₂	P ₄	P ₃
0	1	2	3	4	5	6	7	8

P_1, P_2 P_1, P_2 P_1, P_2, P_3 P_1, P_2, P_3, P_4 P_1, P_2, P_3, P_4, P_5 P_1, P_2, P_3, P_4 P_1, P_2, P_3, P_4

$$\text{Avg TAT} = \frac{16+8+10+8+3}{5} =$$

$$\text{Avg WT} = \frac{12+7+9+5+6}{5} =$$

PID	P NO	AT	BT	CT	TAT	WT	RT
4	P ₁	1	5	6	5	0	0
2	P ₂	2	6	15	13	7	7
2	P ₃	2	4	19	17	13	13
1	P ₄	4	1	20	16	15	15
3	P ₅	6	3	19	3	0	0

	P ₁	P ₅	P ₂	P ₃				
0	1	2	3	4	5	6	7	8

P_1, P_2, P_3 P_1, P_2, P_3 P_1, P_2, P_3, P_4 P_1, P_2, P_3, P_4, P_5 P_2, P_3, P_4, P_5 P_2, P_3, P_4 P_2, P_4 P_4

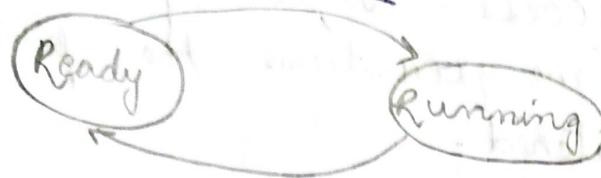
$$\text{Avg TAT} = 54/5 = 11.2$$

$$\text{Avg WT} = 35/5 = 7$$

priority same \Rightarrow then arrival \rightarrow PID

6/15

Round Robin (preemptive)



PID	AT	BT	CT	TAT	WT	RT
P ₁	0	5	12	12	7	0
P ₂	1	4	11	10	1	1
P ₃	2	2	6	4	2	2
P ₄	3	1	9	5	3	4

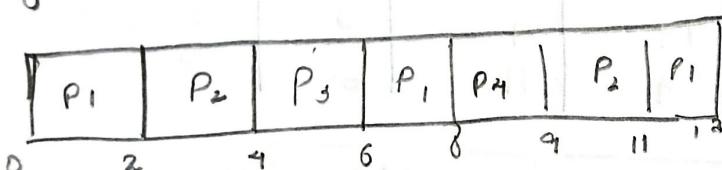
$q = 2$

Ready

P ₁	P ₂	P ₃	P ₁	P ₄	P ₂	P ₁
----------------	----------------	----------------	----------------	----------------	----------------	----------------

$$\text{Avg TAT} = \frac{31}{4}$$

Gantt chart



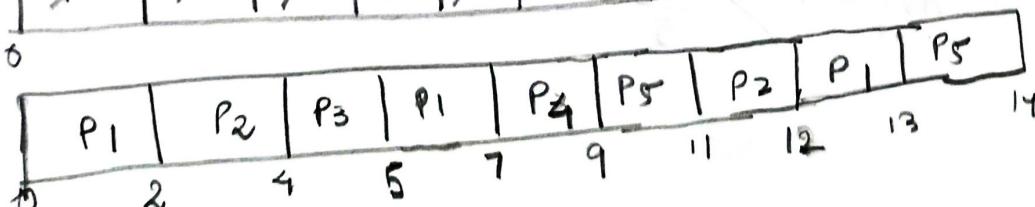
PID	AT	BT	CT	TAT	WT	RT
P ₁	0	5	13	13	8	0
P ₂	1	3	12	11	8	1
P ₃	2	1	5	3	2	2
P ₄	3	2	9	6	4	4
P ₅	4	3	14	10	7	5

$$q = 2$$

Ready

P ₁	P ₂	P ₃	P ₁	P ₄	P ₅	P ₂	P ₁	P ₅
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

Gantt



$$\text{Avg WT} = \frac{29}{5}$$

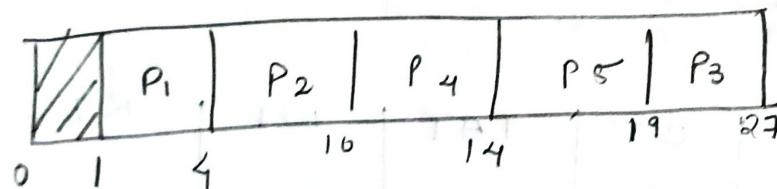
$$\text{Avg TAT} = \frac{43}{5} = 8.6$$

A time quantum q is maintained if it is used for choosing the next process. It is primitive. It is no i.e. after the time quantum the process will be ~~next~~ context switched.

HRRN (higher response ratio next) (non-primitive)

$$\text{Response ratio} = \frac{WT + BT}{BT}$$

PID	AT	BT	CT	TAT	WT	RT
P ₁	1	3	4	3	0	0
P ₂	3	6	10	7	1	1
P ₃	5	8	27	22	14	14
P ₄	7	4	14	7	3	3
P ₅	8	5	19	11	6	6



at $t = 10$

$$P_3 = \frac{5+8}{8} = 1.6$$

$$P_2 = \frac{3+4}{4} = 1.7$$

$$P_5 = \frac{2+5}{5} = 1.4$$

at $t = 14$

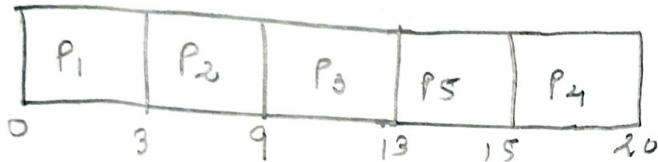
$$P_3 = \frac{9+8}{8} = 2.12$$

$$P_5 = \frac{6+5}{5} = 2.2$$

$$\text{Avg TAT} = \frac{50}{5} = 10$$

$$\text{Avg WT} = \frac{24}{5}$$

PID	AT	BT	CT	TAT	WT	RT
P ₁	0	3	3	3	0	0
P ₂	2	6	9	7	1	1
P ₃	4	9	13	9	5	5
P ₄	6	5	20	14	9	9
P ₅	8	2	13	7	5	5



at t = 9

$$P_3 = \frac{5+4}{4} = \frac{9}{4} = 2.25$$

$$P_4 = \frac{3+5}{5} = \frac{8}{5} = 1.6$$

$$P_5 = \frac{1+2}{2} = \frac{3}{2} = 1.5$$

at t = 13

$$P_4 = \frac{7+5}{5} = \frac{12}{5} = 2.4$$

$$P_5 = \frac{5+2}{2} = \frac{7}{2} = 3.5$$

$$\text{Avg TAT} = \frac{40}{5} = 8$$

- HRRN - In this algorithm we consider the response ratio = $\frac{WT + BT}{BT}$ using SJF algorithm we face the problem of starvation i.e. the process with longer time don't get their chance to get executed on the CPU. Hence, in HRRN we consider waiting time as null at burst time. Its mode is non primitive in nature.