Chapter-3

What is Process? Describe the state of Process?

Process: Process is a program in execution.
It is the working unit under OS.
A process is an active entity.

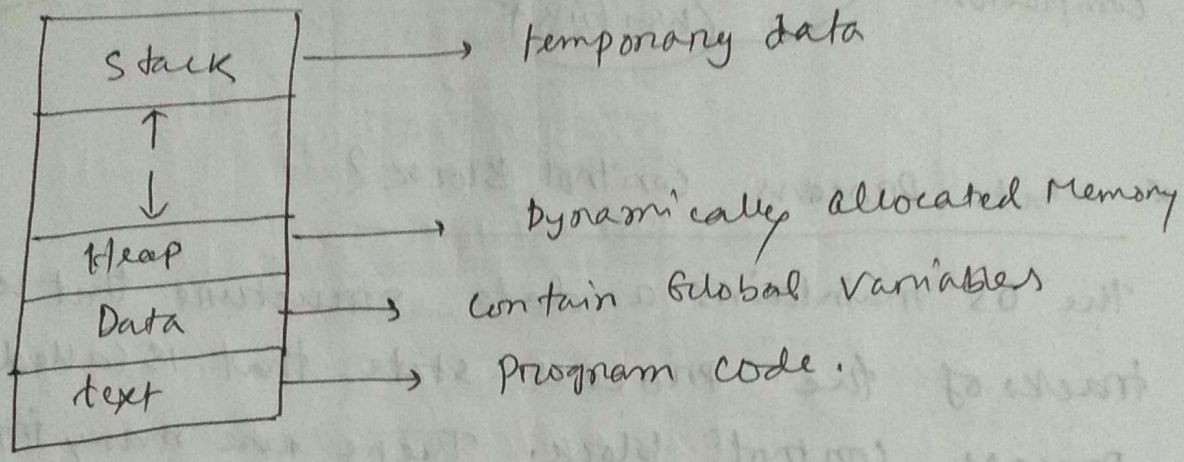
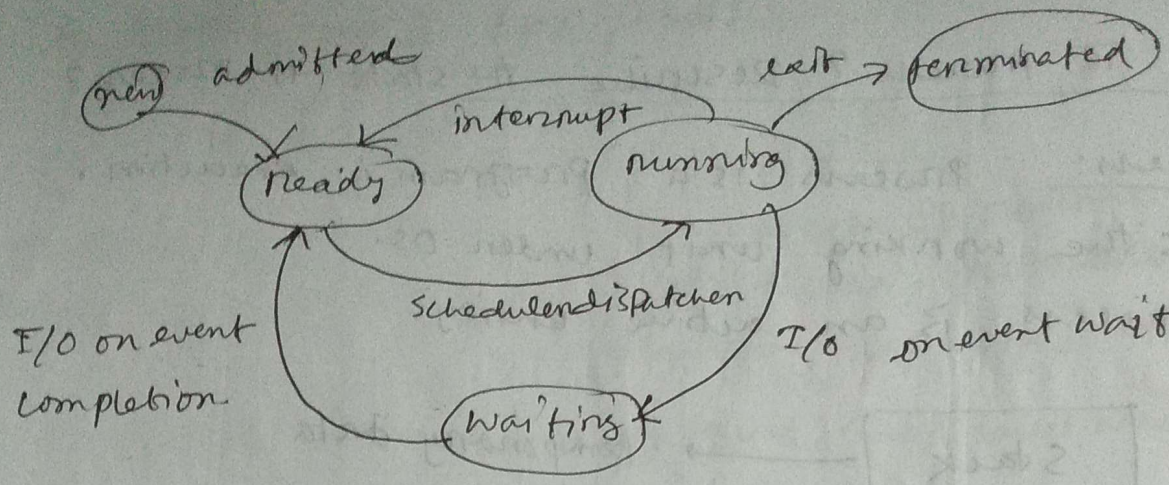


Fig: Process in memory

State of Process: As a process executes it changes it
A process may be one of the following states:--

* new: The process is being created.
* Ready: The process is waiting to be assigned in processor.
* Running: The instruction being executed.
* waiting: The process is waiting for some event to occur.
* Terminated: The process has finished execution.



* What Process control Block?

The OS maintains a data structure that keeps tracks of the process state that is called Process control block. There are many information associated to each process.

① Process State: Running, waiting etc.

② Program counter: location of instruction to next execute.

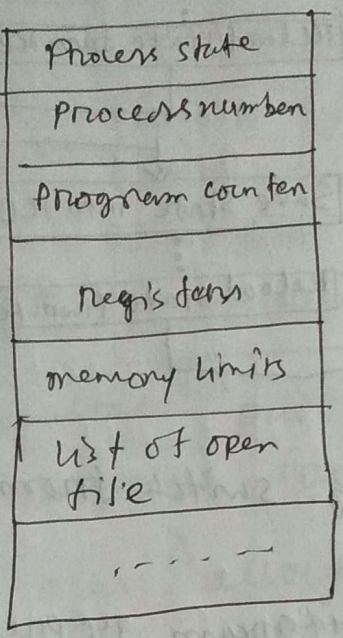
③ CPU registers: contents of all process centric registers.

④ CPU scheduling information: Priorities, scheduling Queue pointers.

⑤ Memory Management info: memory allocated to the program.

⑥ Accounting info: CPU used, clock time elapsed since start, time limits.

⑦ I/O status information: I/O devices allocated to process.



Context switch: ~~When a process switches fr~~ When CPU switches from one process to another the system must save the state of old process & load the saved state of new process. This technique is called context switch.

The time required to switch from one process to another is called context switch time. context switch time is pure overhead because system do nothing useful when switching.

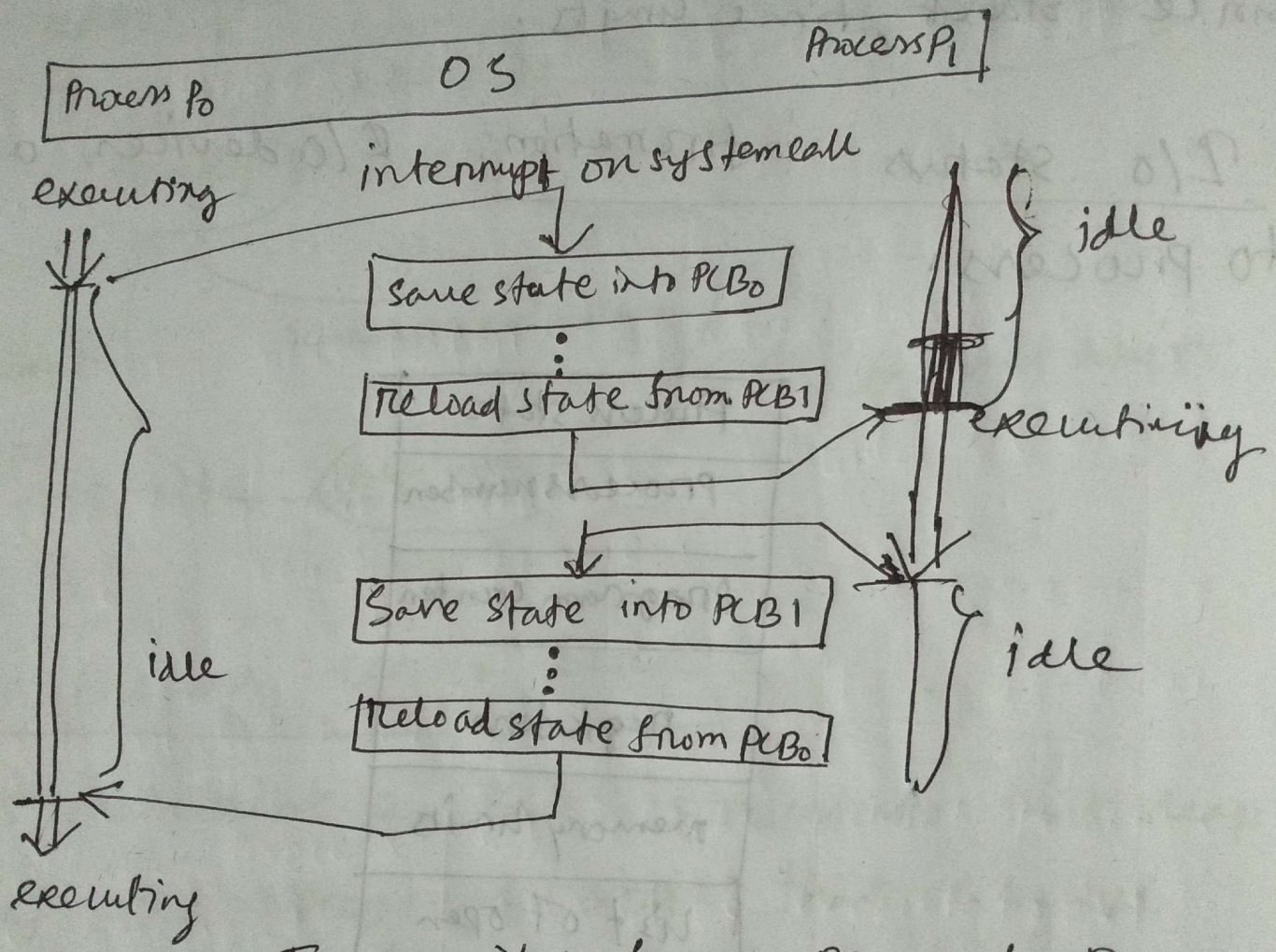


Fig: switch from process to process.

Short term scheduler / CPU scheduler

Selects which process should be executed next & allocates CPU.

Long term scheduler / job scheduler:

Selects which process should be brought into the ready queue.

Long term scheduler	Short term scheduler
(1) Select which process should be brought into ready queue	(1) Selects which process should be executed next
(2) executed less frequently.	(2) executed more frequently
(3) controls degree of multiprogramming	(3) Less control over degree of multiprogramming
(4) Alternatively referred as job scheduler	(4) alternatively referred as CPU scheduler
(5) may or may not be present in both batch & sharing system	(5) present in both batch & sharing system.

There are two types of process:-

(i) I/O bound process: spend more ~~time~~ of its time in generating I/O requests than doing computation.

(ii) CPU bound Process: spend more of its time doing computation than generating I/O requests.

It is important for long term scheduler to select a good mix of I/O bound process & CPU bound process.

(i) If all the processes are I/O bound then the ready queue is empty.

(ii) If all the processes are CPU bounded then I/O waiting queue almost always empty.

If any of these occurs system will be unbalanced so we need to select a good mix of I/O & CPU bound processes.

What is IPC? Describe the methods of IPC?

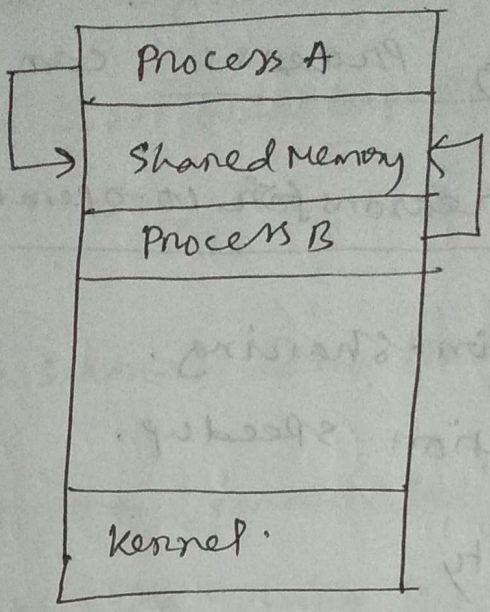
IPC: Interprocess communication is a mechanism that allow Processes to communite with each other & synchronize their actions.

Shared Memory:

(1) A part of memory is shared among the Processes that ~~wites~~ wishes to ~~to to~~ communicate.

(2) The communication is under control of the ~~users~~ processes not the OS.

(3) The major issue is to synchronize the action of processes after accessing shared memory.



(2) Message Passing:

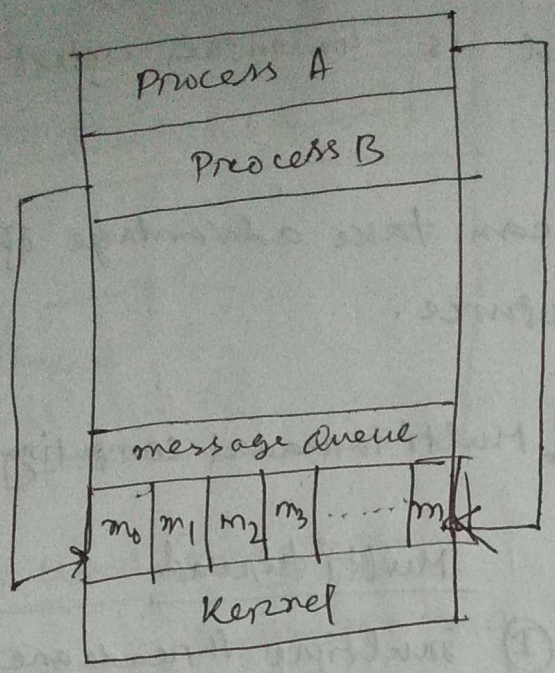
(i) Process communicate with each other with a message system without resorting to shared variable.

(ii) A massage Passing system provide two operation.

Send (message)

Receive (message)

(iii) The message size is either fixed or variable.



Ch-4
Threads.

Threads: ~~It is the fundamental unit of CPU utilization & is used to form the~~ It is the fundamental unit of CPU utilization and is used to form the basis ~~of~~ of multithread computer system. It consists of thread id, program counter, a register set & a stack.

Benfits of thread/Multithreaded programming:-

(i) **Responsiveness:** may allow continued execution if part of processer is blocked, especially important for user interfaces.

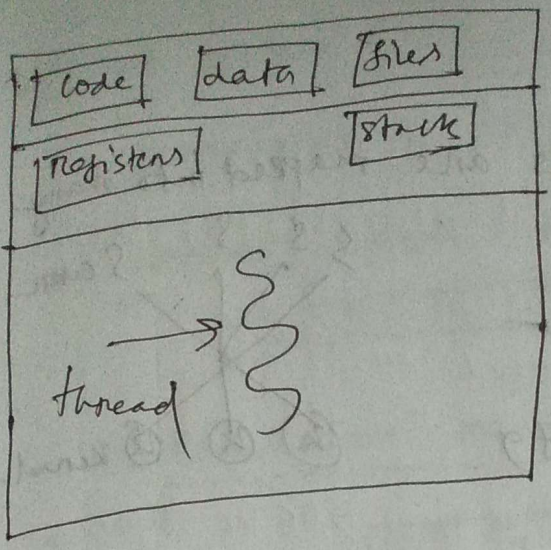
(2) **Resource sharing:-** threads share resources of process, easier than shared memory or message pas

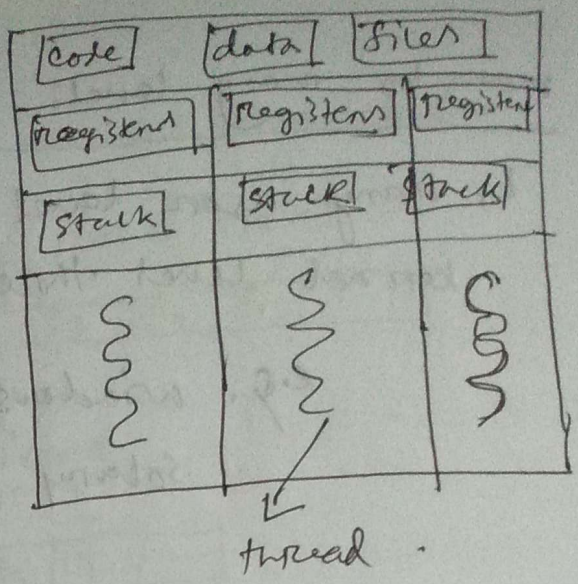
③ Economy: cheaper than process creation, thread switching overhead is lower than context switching.

④ Scalability: Process can take advantage of multiprocessor architecture.

Difference between single & Multithreaded computing:-

Single thread	Multithread
① This type of programming there is only one ~~thread~~ thread is running at a time	① multiple threads are allowed to run concurrently.
② CPU time is wasted	② CPU time is never wasted.
③ The CPU can be idle	③ CPU is never idle.
④ If thread is blocked for some resources, other threads wait for it to resume.	④ If a thread is blocked other threads continue executing.
⑤ Results in a less efficient Program	⑤ It result in a more efficient program.
⑥ It uses event loop with polling	⑥ It doesn't use event loop.





User level thread:

Thread supported by a user level library is called user level thread.

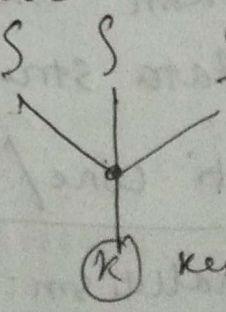
Kernel level thread:

Threads supported by a kernel level library are called Kernel level threads.

There 3 Multi-threading model.

Many to one model: many user level threads are mapped to one kernel level thread.

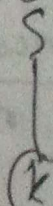
e.g. Solaris green thread,
GNU Portable thread.



one to one model:

Each user level thread is mapped into one kernel level thread.

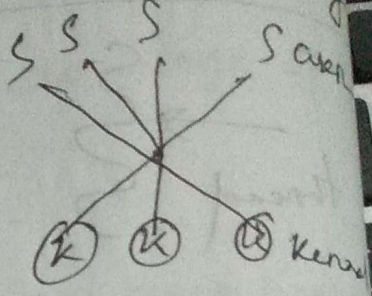
e.g. windows/XP/2000,
Linux, Solaris 9 & later



many to many level:

↳ many user level threads are mapped into many kernel level thread.

e.g. windows /xoo
solary previous of g



Diff^n between user level & kernel level

(1) user level threads are unknown by the kernel but kernel's aware of kernel level threads.

(2) user level threads must be associated with a process but kernel level threads are not associated.

(3) Kernel level threads are difficult to maintain than user level threads because of kernel data structure.

multi core / multi processor

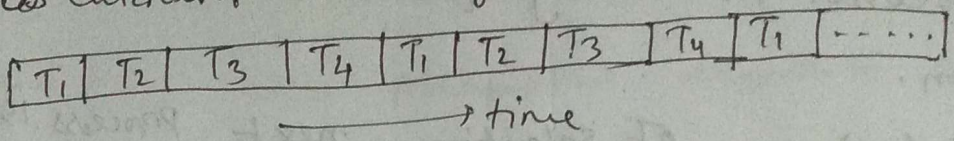
Parallelism: It specifies that a system can run multiple tasks simultaneously.

Concurrency: It specifies a system that supports more than one tasks making progress.

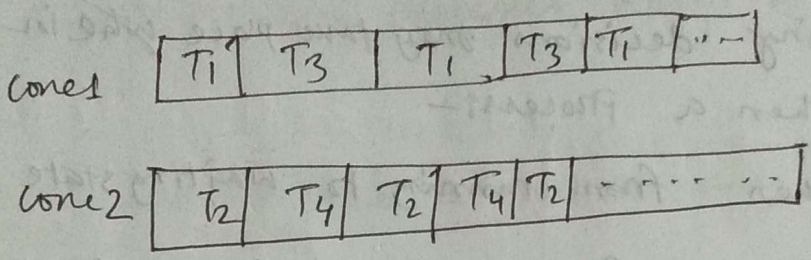
Diff → ২টা কাজ

↳ concurrency is possible without Parallelism

concurrent on single core system:-



Parallelism on a multicore



Two types of Parallelism:

Data Parallelism: Distributes subset of the same data across multiple cores, same operation on each.

Task parallelism: distributing threads across cores, each thread performing unique operation

Thread Drawback

① Read lock may occur.

② Thread crashes a process.

Ch-6

CPU scheduling.

CPU scheduler: selects ~~which~~ processes from among the processes in ready queue & allocates CPU to one of them.

The technique of selecting next process is refer as CPU scheduling.

CPU scheduling decisions may take place ~~whe~~ in following situation. when a process:-

1. Switches from running to waiting state.
2. switches from running to ready state.
3. switches from waiting to ready.
4. Terminates.

Dispatcher module:

Dispatcher module gives the control of the CPU to the process selected by short term scheduler.

This involves

(i) switching context

(ii) switching to user mode

(iii) jumping to the proper location in the user program to restart that program.

Dispatcher latency: The time required to stop a process & running a new process by the dispatcher is called dispatcher latency.

CPU-I/O Brust cycle:

Process execution consist of a cycle of CPU execution & I/O wait

Non Preemptive: Non preemptive algorithm are designed so that once a program enters running state it will not remove from processes until it has completed its service.

Preemptive: Preemptive algorithm are designed by the notation of prioritized computation. The process with highest priority will alway be the currently using processor.

VVI Scheduling criteria

* CPU utilization - Make the CPU as busy as possible.
* Throughput - # no. of processes that complete their execution per time unit.
* Turn around time: amount of time to execute a particular process.
* Waiting time: amount of time a process has been waiting in the ready queue.
* Response time: amount of time it takes from when a request is submitted until first response is produced.

Convoy effect: In FCFS scheduling if long process execute before short process then average T.T & W.T will increase. This pheromena is called convoy effect.

Starvation: In priority scheduling, low priority process will have to wait untill high priority process execute.

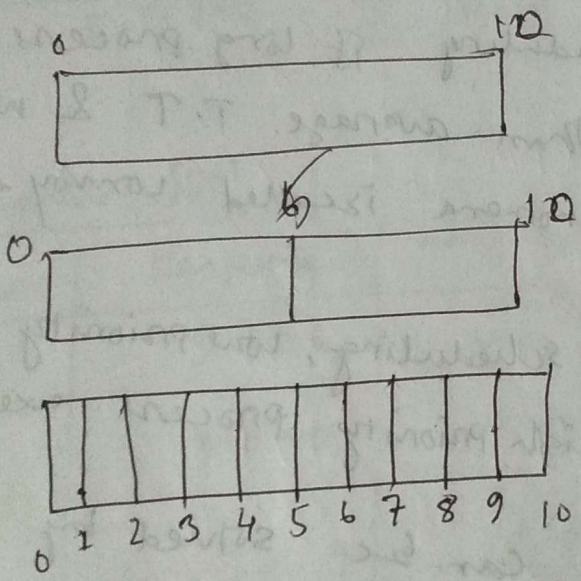
Aging: This problem can be solved by aging. As time progresses, increases the priority of the process.

Time quantum: Each process get a small amount of CPU time for execution. it is called time quantum The time spend in selecting process from ready queue is called context time. In context switch time no execution is happened.

* Performance of Round Robin:

Relation between time quantum & context switch

The Performance of Round Robin algo heavily depends on time quantum. If the time quantum size is large, context switch will require less. Hence the context switch overhead will decrease and it will decrease the average waiting & Turnaround time.



quantum

10

5

1

context switch

0

1

9.

If time quantum is decreased, number of context switch will increase & hence context switch overhead will increase, it will increase average turnaround time & waiting time.

Difficulty in implementing SJF?

The most difficulty with SJF is knowing the length of the next CPU burst.

Soln For long term scheduling user specifies time limit for each process. The time limit specification can be used as next CPU Burst. The lower value of this CPU burst will execute first.

For short term we predict the CPU Burst via prediction process. Next CPU burst can be calculate from previous actual length of CPU Burst & previous prediction.

$$T_{n+1} = \alpha t_n + (1-\alpha) T_n$$

$\alpha = \frac{1}{2}$

t_n = actual length of n^{th} CPU Burst

T_{n+1} = Predicted value for next CPU Burst.

T_n = " " n^{th} " "

2012
3(b)

We know $T_{new} = \alpha t_n + (1-\alpha) T_n$.

Hence given $T_0 = 10 \quad \alpha = 0.1$

$$T_1 = \alpha t_0 + (1-\alpha) T_0$$

$$= 0.1 \times 10 + (1-0.1) \times 10$$

$$= 10$$

$$T_2 = 0.1 \times 16 + (1-0.1) \times 10$$

$$= 10.6$$

$$T_2 = 0.1 \times 4 + (1-0.1) \times 10.6$$

$$= 9.94$$

$$T_3 = 0.1 \times 20 + (1-0.1) \times 10 = 10.95$$

$$T_4 = 0.1 \times 12 + (1-0.1) \times 10.952 = 11.05$$

$[10, 10.6, 9.94, 10.95, 11.05]$.

Multi level Queue:

In multilevel queue process permanently given a queue where they can enter the system.

In multilevel feedback queue, A process can move between queue. Aging can be implement in this way.

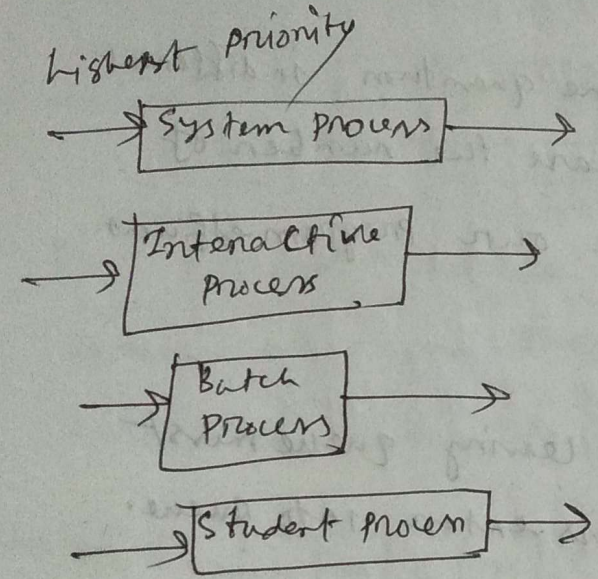


Fig: Multilevel Queue

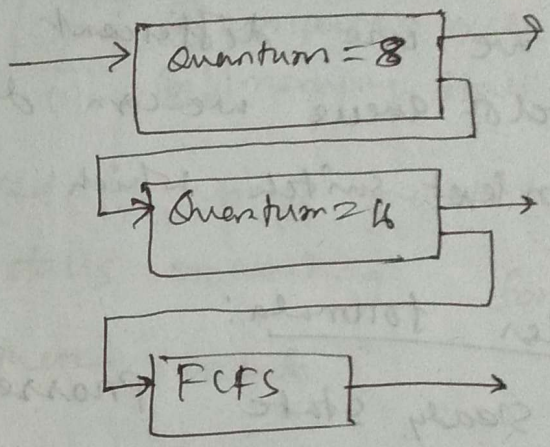


Fig. Multilevel Feedback Queue.

Multiprogramming	Multitasking	Multiprocessing	Multithreading
① It's the ability of OS to execute more than one program on a single processor.	① It's the ability of OS to execute more than one task simultaneously on a single processor machine.	① It's the ability of an OS to execute more than one process simultaneously on multiprocessor machine	① It's the ability of an OS to execute different parts of a program called thread at same time on a single processor machine.
② Here one program as a whole keeps running	② A task in multitasking OS is not a whole application.		
③ Time sharing is not that good as whole application takes much time.	③ In multitasking. Time sharing is be manifested as each running processes taken only a fair equation of CPU time.		
④ Older OS	④ It's widely used in Modern OS. Linux, & Unix. Windows.		

Diff^n between Program & process

Program	Process
① Program is a set of instructions which is Prepared to Perform a sepecific ~~instructions~~ assignment if executed by a computer.	① OS creates process from Program. It's created to a Program using computer utility.
② This Program is not active entity. completely passive	② It's an active entity.
③ It's need'nt to be on-Line can be stored in a flash memory	③ It's to be on line stored in memory
④ A Program may contain several Process	⑭ A Process must be ~~execute~~ execute under a single Program

⇒ CPU scheduling.

(iv) If processes don't fit in memory, swapping moves them in & out to run.

(v) virtual memory allows execution of processes not completely in memory.

Operating system operations:

① Interrupt driven: (Hardware & software).

(i) Hardware interrupt by one of the devices.

(ii) Software interrupt (exception & on trap).

↳ Software errors.

↳ Request for OS services.

↳ other process problems include infinite loop, ~~processing~~ processes modifying each other on the OS.

② Describe Dual Mode Operation:-

Dual mode operation allows OS to protect itself & other system components. An OS operates in two separate modes:-

(i) User mode

(ii) kernal mode (monitor mode/ system mode/ Priviliged mode /supervisor mode).

(i) A bit called mode bit, is added to the hardware to indicate the current mode: Kernal (0) or user (1).

(ii) With the mode bit, we are able to distinguish between a task that is executed on behalf of the OS & one that is executed on behalf of the user.

(iii) When the computer is executing on behalf of the user application, the system is in user mode.

(iv) Some instructions designated as Previliged, only executable in kernal mode.

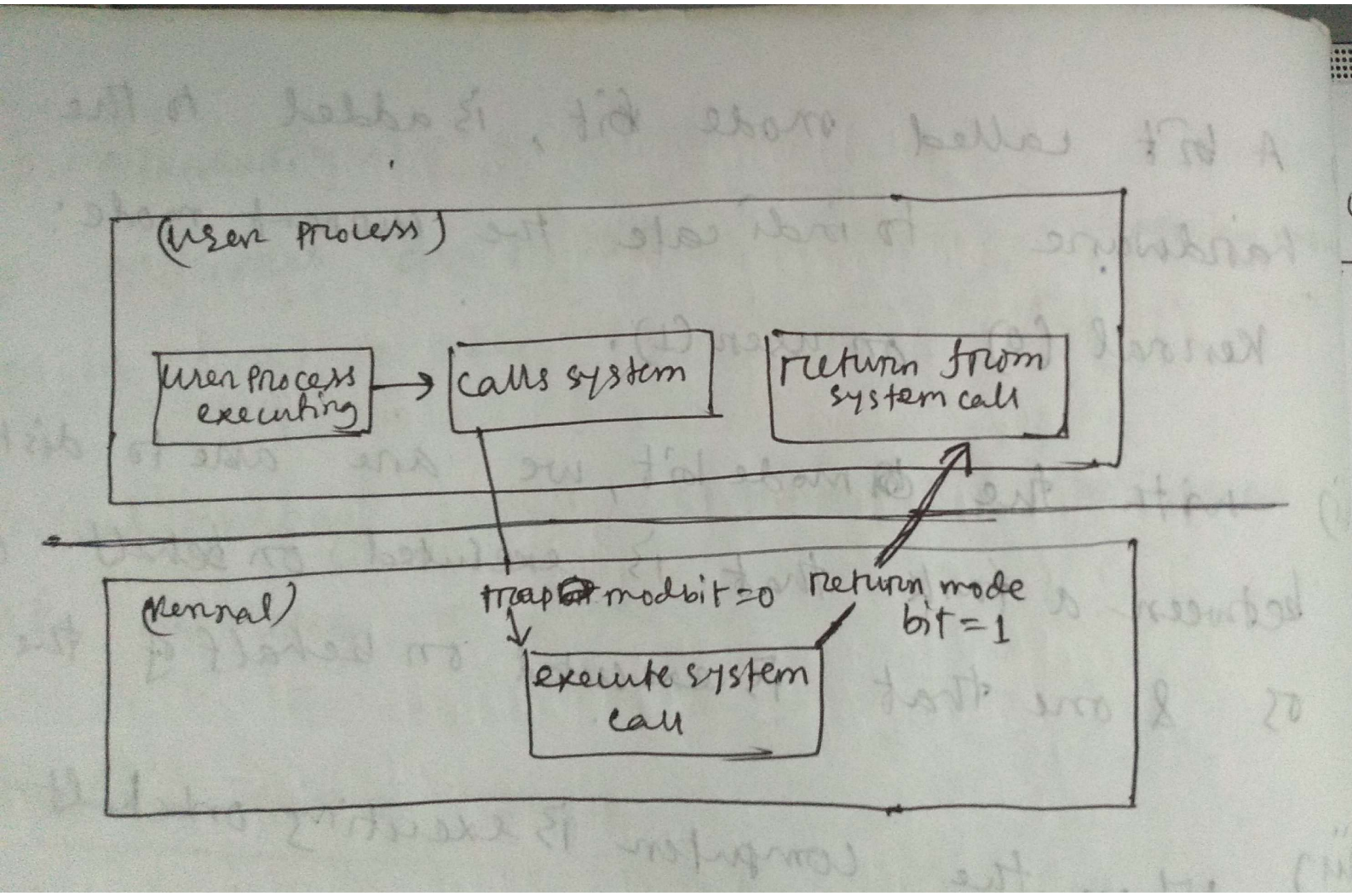
(v) system call changes mode to kernal, return from call resets it to users.

Transition from user to kernal mode:-

(i) When a user application requests a service from the OS, it must transition from user to kernal mode to fulfill the request.

(ii) at system boot time the hardware starts in kernal mode. The OS is then loaded & starts user application in user mode.

(iii) When a trap or interrupt occurs, the hardware switches from user to kernal mode.



What is operating system?

An operating system is a program that manages computer hardware. It acts an an intermediary between a user of a computer and the computer hardware.

Why OS called Goverment? Draw abstract view of OS.

Ans: An OS is a program that manages computer hardware. It provides a basis for application program & acts as an intermediary between a user and the computer hardware. Without OS a computer would be useless hunk of metal & plastic. The OS provides the basic rules, conventions, protection & services necessary for functioning of application program.

Goverment plays a similian ~~role~~ role in the functioning of modern society. Goverment furnishes rule, conventions & basic services necessary for the smooth functioning ~~for~~ of the people of a country.

So OS called goverment.