"TAHSIN
OS NOTE's
CSE325
"

L-1

Important Topics

→ to design and implementation of OS.

- interrupts
- Computer resource management : OS manages
  memory management, Processor management, I/O management, file management, Processes, security management.
- inter process communication.

→ Multithreaded OS, & Concurrent computer.

→ Others.

Intro

Components of a Computer system :

Banking, Airline, web Prog. } Application Prog.

Compiler, Editors, Command interpreter } System Prog.

OS

Languages (Machine), devices, Microarchitecture } Hardware.

- User
- Application
- OS
- Hardware

OS: A program that acts as an intermediary between a user & H/ware Hardware.

goal – convinent to use,
execute user programs & make solving user problem easier,
use H/W in an efficient manner.

- OS in a resourse allocator – [OS try to avoid wasting CPU time.]
  - → manages all resources.
  - → Decides between conflicting req. for efficent and fair resource use.
  (resource utilize করার জন্য) GW
- OS in a control program –
  - → Memory Protection: control execution of Program to prevent errors {fetch()} and improper use of computer.

Program (like: C, C++) ⇒ (compile) Assembly ⇒ (Assemble) Executable ⇒ Memory

(15)

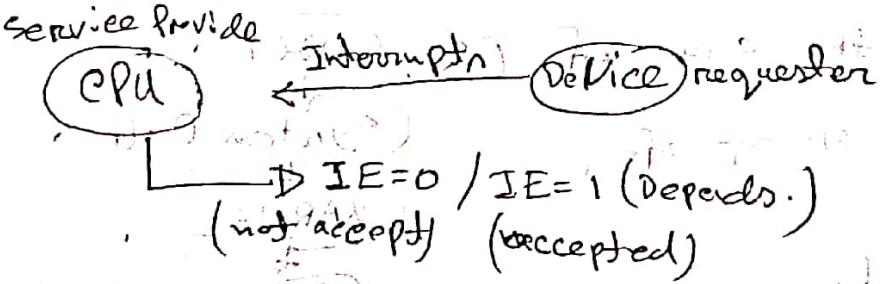
Polling Vs Interrupt: Both are methodes to notify Processor that I/O device needs attention.

→ Polling: simple, but slow;
- check regularly I/O devices (cpu waste 가능) without bells.

→ Interrupt: fast, but complicated; (signal 받으면 cpu 중단)
- it's better if procesor has other work to do;
- Procesor will be notified by I/O devices.

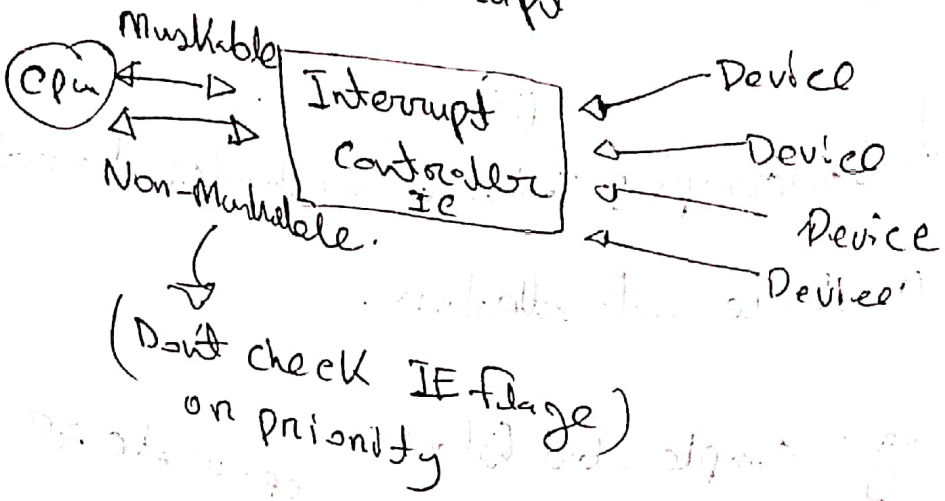
in short, interrupt is asynchronous, and polling is synchronous.

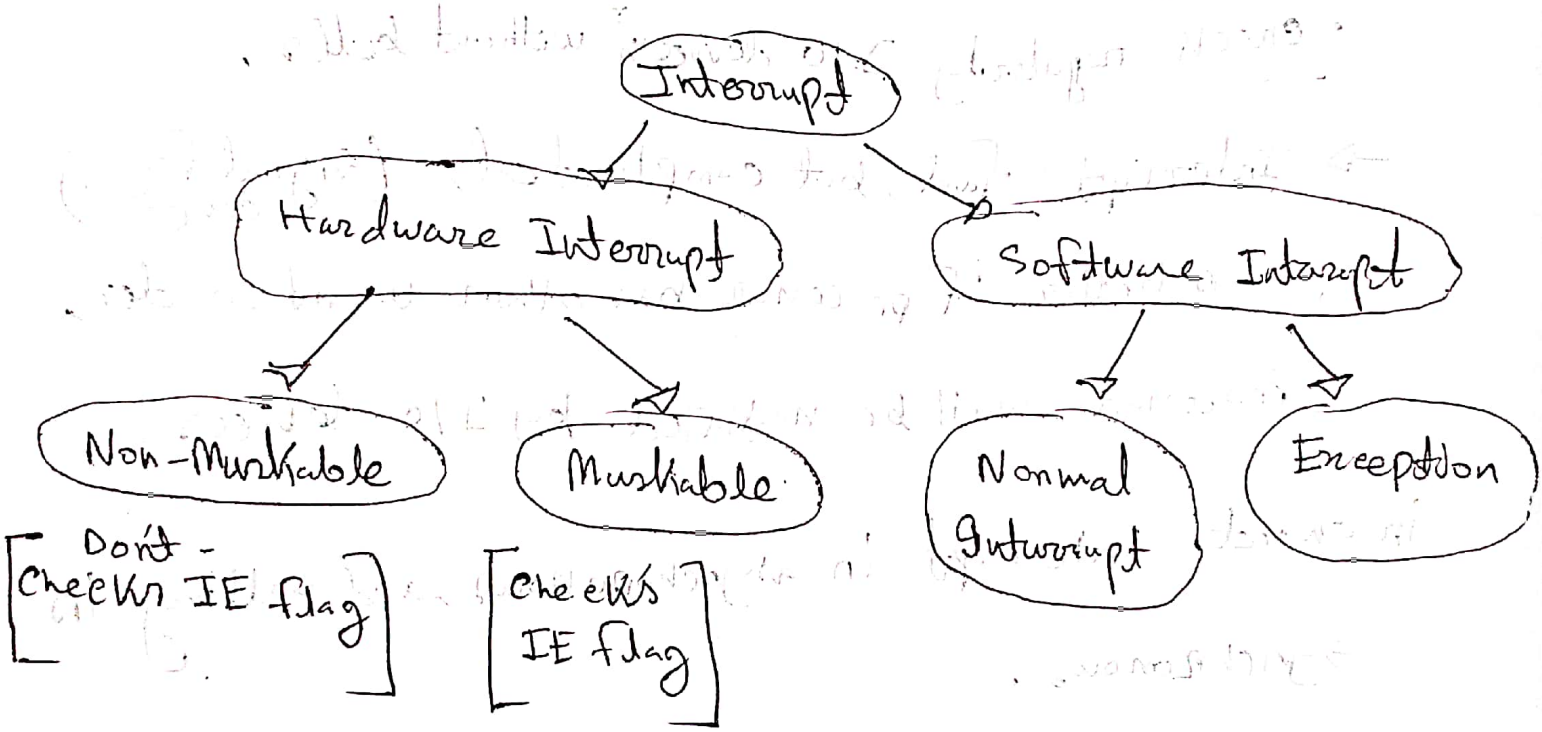
Interrupt working:



Interrupt Vector: (Alternative)

To control the interrupt



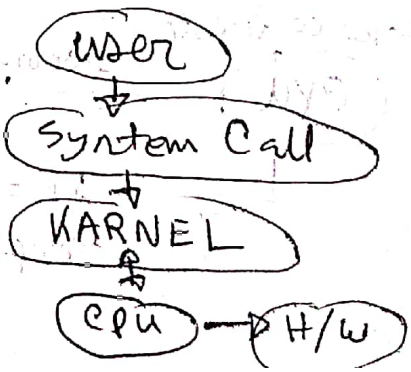


How does user communicate with OS:

(*) KERNEL is the core of OS.

(i) Monolithic.

(ii) Micro Kernel.



✓ OS has 2 parts – Kernel & Shell.

① Shell - watches each and every thing

ex - → (Hello Tahsin) Print ✓

out: Hello Tahsin

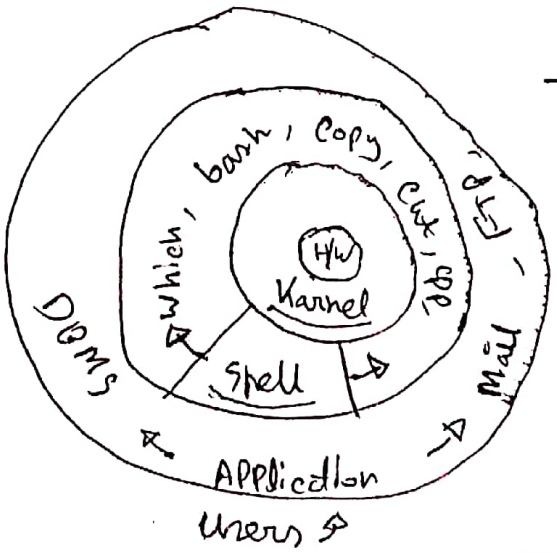
(Hello Tahsin) Print ✓

out: Hello Tahsin [Here, shell removes the spaces.]

✓ Shell is the interface which takes input from users and send it to Kernel.

- assembly language কে machine language এ convert করে।

② Kernel: is the heart of OS. It acts between application & Hardware. It lies in the centre of the centre of OS.



→ Copy, Paste এগুলো command shell এ আছে যায়।

→ Shell এবং Kernel এর language এ convert করে।

Tasks of Kernel : Memory management, Process Management, security, Networking - GUI.

Kernel Types :

(1) Monolithic Kernel : · All parts of a Kernel
→ Process management, memory management, interrupt etc.

· Advantage : faster
· Disadvantage : crash in secure, Porting Inflexibility
size explosion.

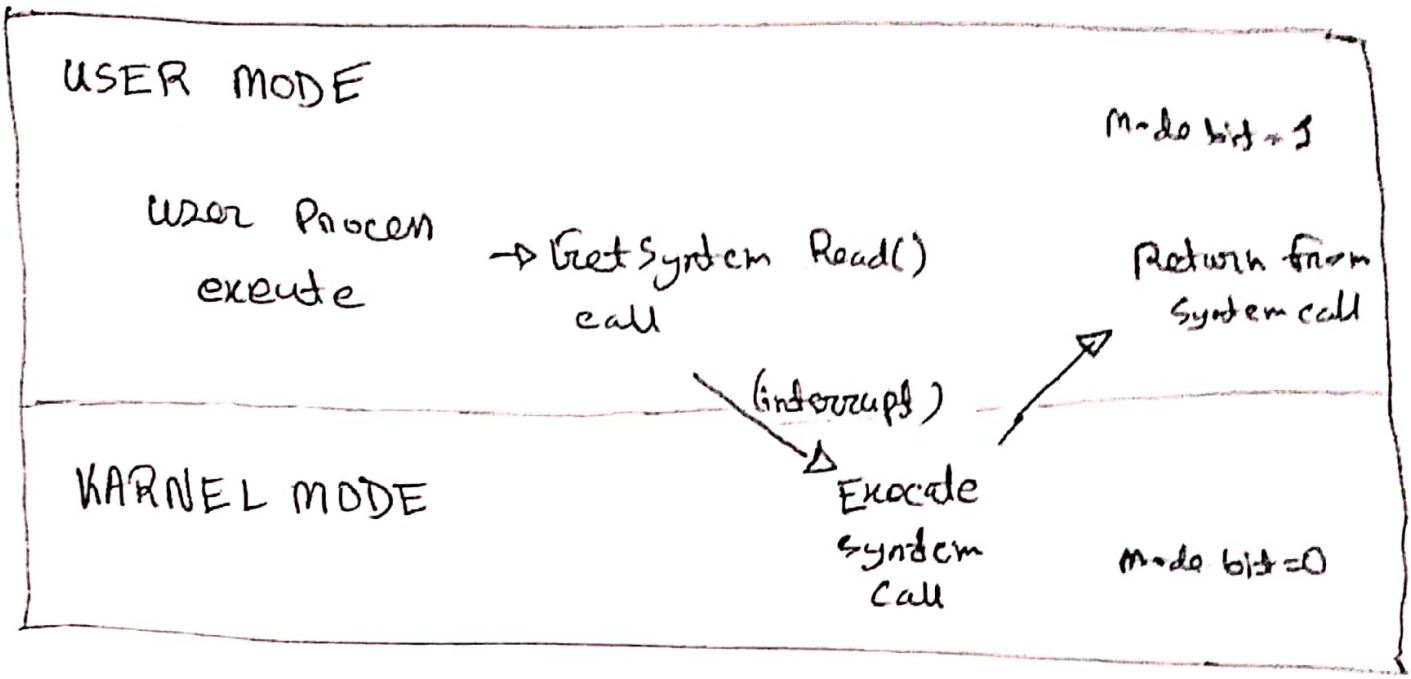
same devices: ▽ · Signal & Sockets to ensure IPC.

(11) Micro Kernel : · Only the important part in Kernel → IPC, basic memory, basic I/O Primitive
· Advantage : Crash resistant, smaller size, Portable, no re-compiling.
· Disadvantage : slower due to additional menage paring.

· भिन्न भिन्न devices. ↔ menage paring system to ensure IPC.

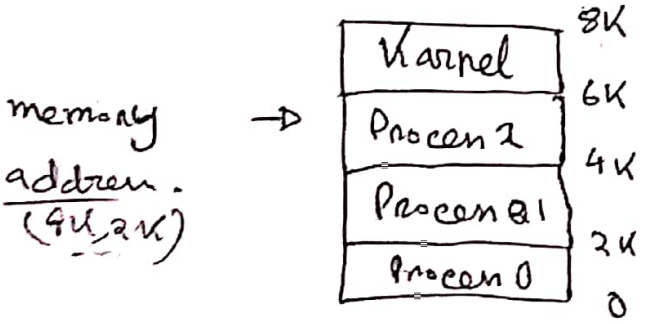
OS Protection

- **User mode Vs Kernel / System Mode:**



- ex→ যখন আমরা কিছি Printf() যাতে monitor এ display করে; তখন তা KARNE এ command টি প্রেরণ করে।
  - CPU both user mode & Kernel mode এ switch করতে পারে।

- **Memory Protection:** Programs, users & System should be given just enough privileges to perform their tasks.



nb:- high memory go on top.
- low " " " bottom.
- Process 2 cannot go outside of 6K & 4K.

✓ (Secure & Protection) ex.

Lec-2

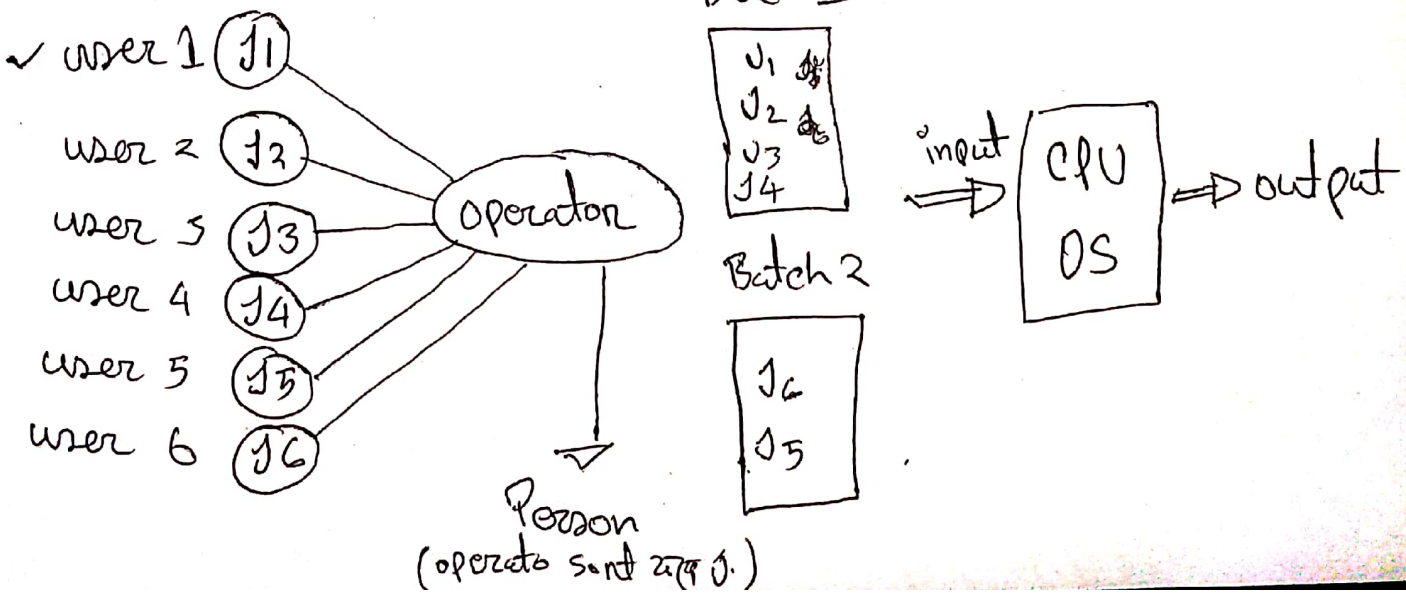
gate Smashers

- major goal of OS is Convenience.
  - ↳ user এর জন্য easier করে command to hardware.
- functionality of OS →
  - (i) Resource Mgmt. (কতটুকু জায়গা দরকার তা OS Provide করে)
  - (ii) Storage Mgmt (file System)
  - (iii) Process mgmt. (~~file System~~)
  - (iv) Memory mgmt.
- Types of OS: (1) Batch: (We are making a batch of similar kind of job and giving computer so it can execute it.)

✓ Job

✓ Punch card → Job

Program
I/O data
Control instruction



(ii) **Multi-Programming :** different from Batch operation.

ex → [P1 , P2 , P3] → P1 → [CPU]

when P1 done then -

[P1, P2, P3] → P2 → [CPU]

whe P2 done, then P3 ;

→ P1 হলে 30 min লাগলে CPU 30 min কাজ করবে then P2 এর কাজ করবে। কিন্তু Batch এ Same types Job গুলোকে একত্রে নিয়ে execute করতো।

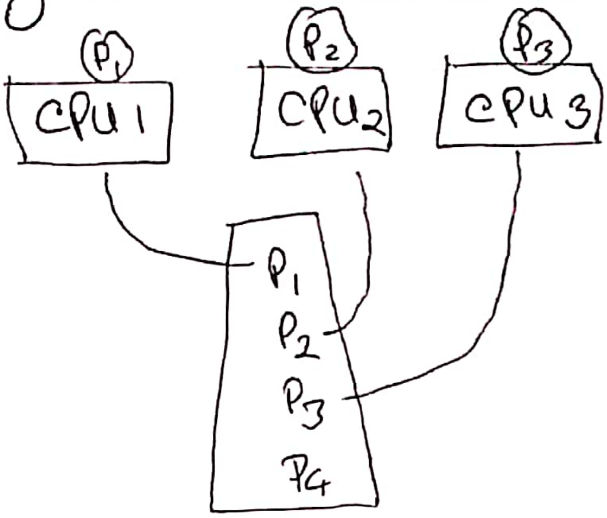
(iii) **Multi-tasking OS :**

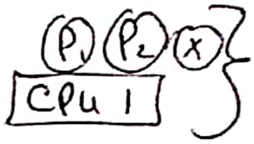
[P1 , P2 , P3] → [CPU]

- Multi-Programing এ P1 এর কাজ শেষ হলে P2, P3 এ আসে। কিন্তু Multi-tasking এ forcefully P1 কাজ Complete না করেই P2 / P3 এর কাজ করা যায়।

** NOTE : CPU cannot run multiple Process in same time. It can run only 1 process, though it looks like multiple process are running.

(iv) Multiprocessing OS: ✓ multiple CPU's.



Nb: 

→ Within one [Processor/CPU] computer multiple Processor's are there is called multi-Processing. Such as; dual (2) core processor → 2 core's, quard core → 4 cores processors.

(v) Real time OS: ✓ Time bound ✓ 2 types –

(i) Soft Realtime: the processes should ~~must~~ be completed within time bound. হবে।

(ii) Hard RT: exect time এ Process complete হতে হবে।

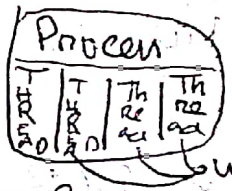
→ Time bomb, missile launching.

Lec-(4-5)

Process Management

✓ **Process**: A process can be thought of a program in execution.

✓ **Thread**: A Thread is the unit of execution within a process. A process can have anywhere from just one thread to many thread.



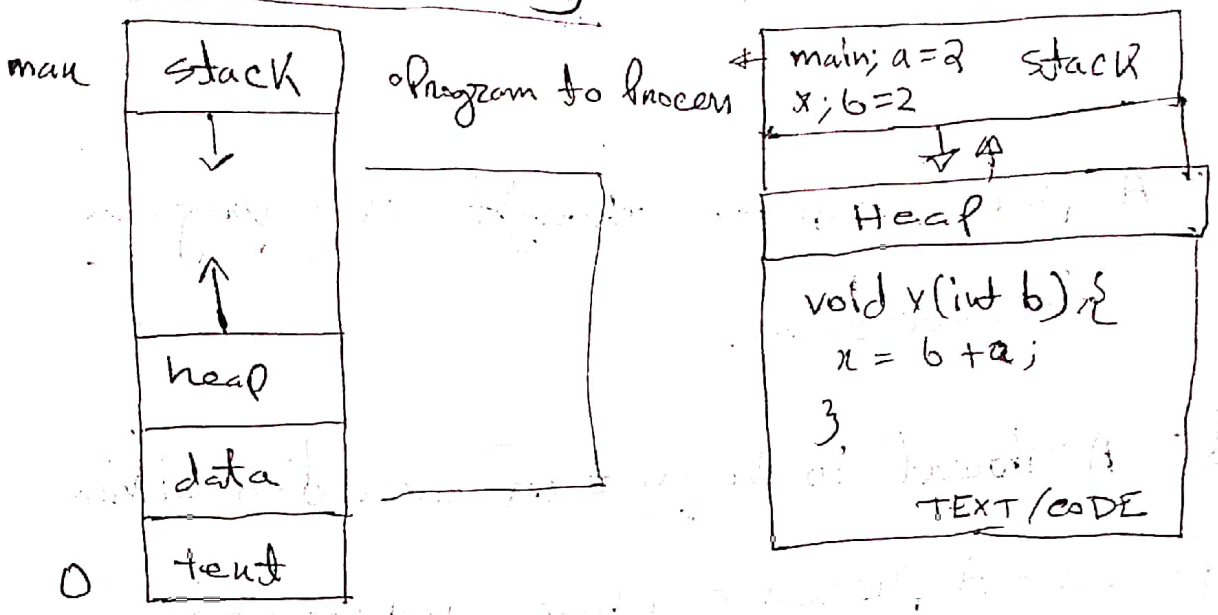
✓ Memory Organization for an Executed Program:

When a program is loded in memory, user space in organized 4 regions called segments.

(I) code/text (II) data (III) stack (IV) heap segments.

- (I) code/text → Compiled code of the program
- (II) data → Data area contains [Static Variable]
- (III) stack → Life - Stack, local variable.
- (IV) heap → Dynamic variable are allowed here. Dmelloc.

Procen in memory –



✓ Procen control Block (PCB):

- PCB used to represent a particular procen in OS.

(#) Procen state: The state of a Procen is defined in part by the current activity of that procen. As a procen executes, it changes state.

- each process maybe in one of the following states:

(i) New state: The process is being created.

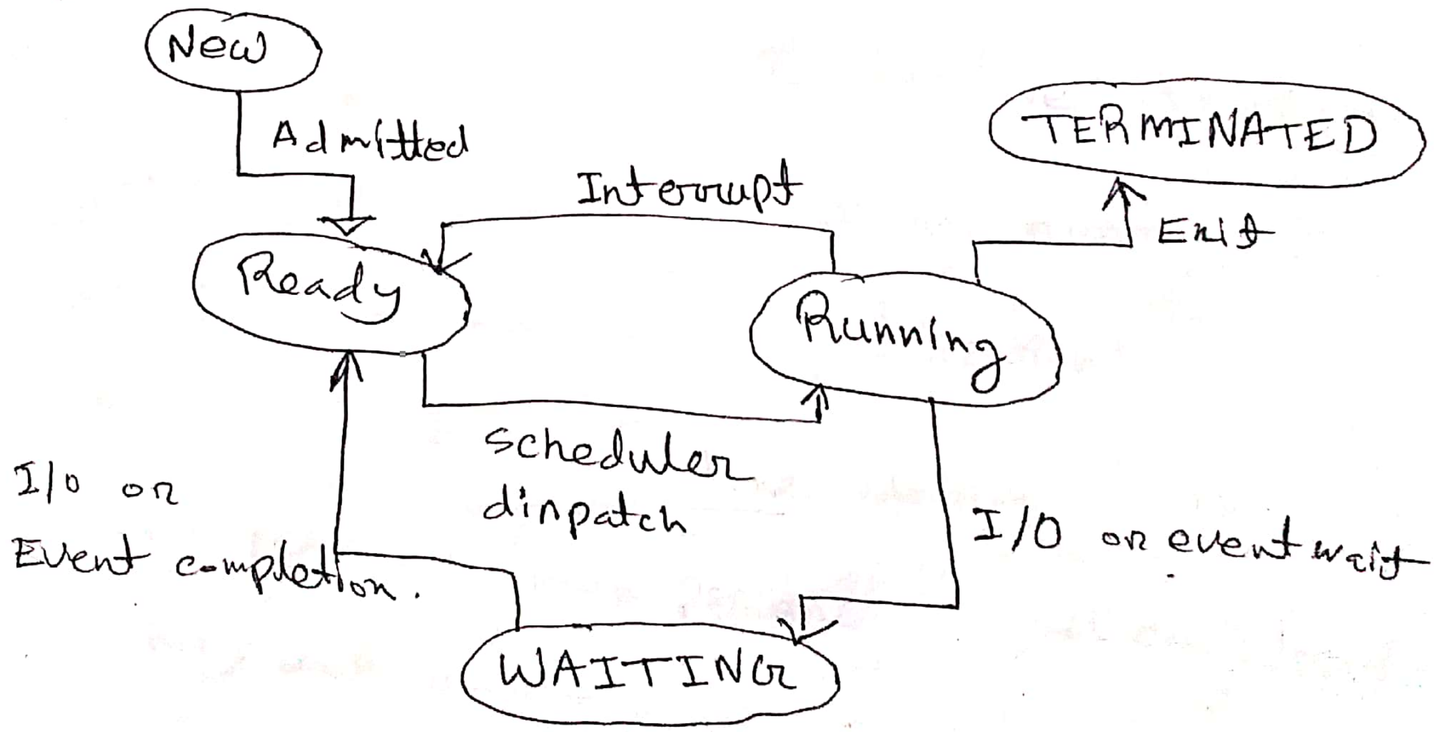
(ii) Running state: Instructions are being executed.

(iii) Waiting state: Waiting for some event to occur.
 → I/O operation.

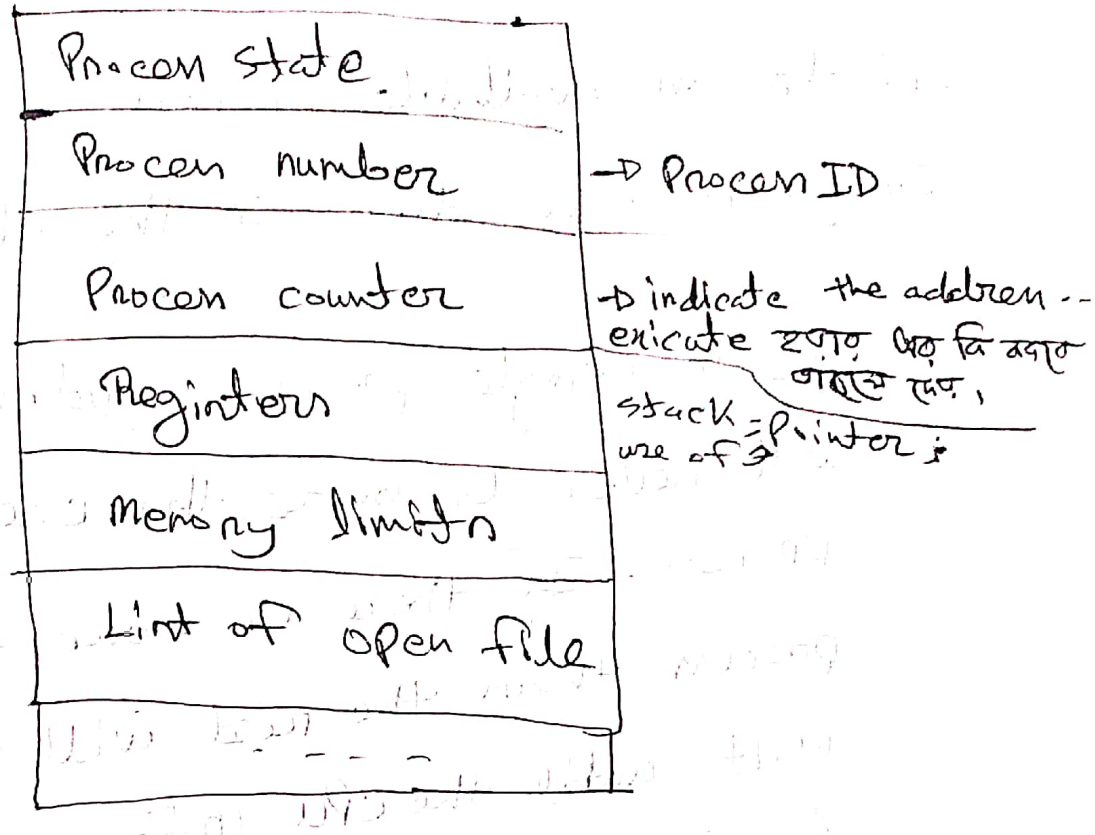
(iv) Ready: The process is waiting to be assigned to a processor.

(v) Terminated: The process has finished execution.

✓ Diagram:



PCB : ek : There represents the different aspects of a particular process -



✓ Process Scheduling : ◦ The objective of multiprogramming is to have some process running at all times, to maximize CPU utilization.

◦ The objective of time sharing is to switch CPU among process so user can interact with each program while it's running.

- So, to meet these objective, the process scheduler

Select an available process for Program execution on the CPU.

ie: in a single-processor system, there will never be more than one running process. If there are more than one process than the rest will have to wait until the CPU in free.

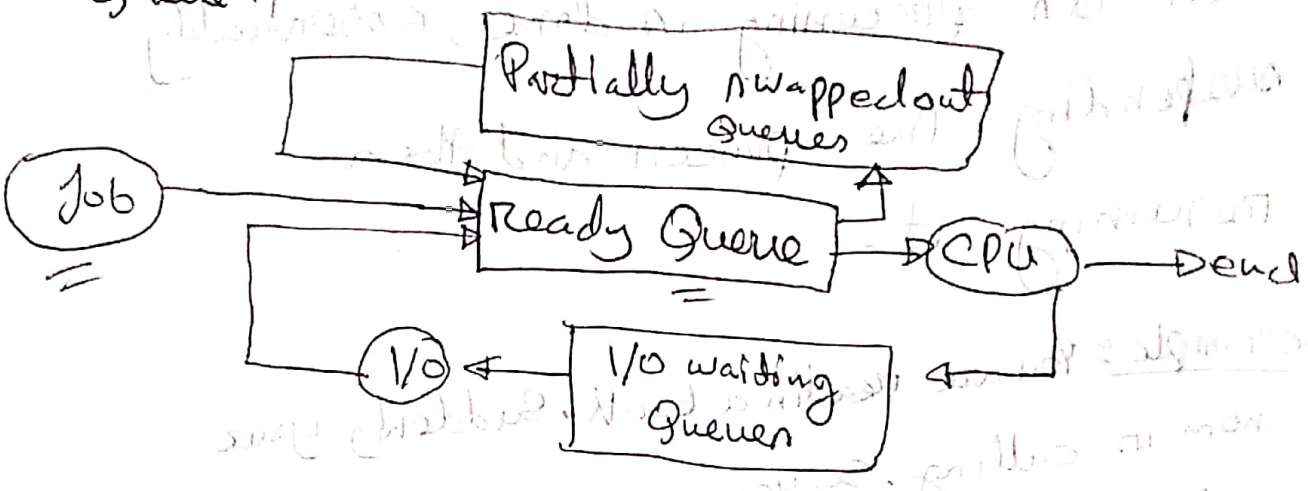
↳ [illegible] Process [illegible] [illegible] Priority -wise Process Scheduler decide करता है,

✓ Scheduling Queues:

① Job Queues: As processes enter the system, they are put into job queue. it contains all the processes in the system.

② Ready Queue: The processes are residing

in main memory and are ready and waiting to execute are kept on a list in called ready Queue.



Content Switch:

- Interrupts cause the operating system to change a CPU from its current task and to run a kernel routine.
- Such operations happens frequently on general-purpose system.

— So, when an interrupt occurs, the system needs to save current Context of the

Note: Contents: Current info about particular process.

Process currently running on the CPU so that it can restore that content when it's processing is done, essentially suspending the process and then resuming it.

Example: You are reading a book. Suddenly your mom is calling. So, you marked the part/point. After helping your mom, you continue reading that book.

NB: The content is represented in PCB of the Process. (PCB)

- State Save
- State Restore.

Threads

Thread: A basic unit of **CPU utilization / execution.**

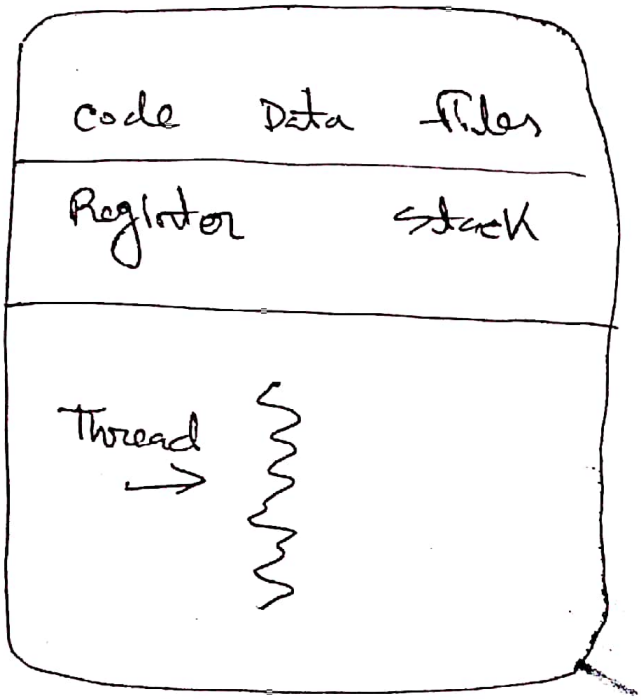
It comprises → Thread Id

Program counter

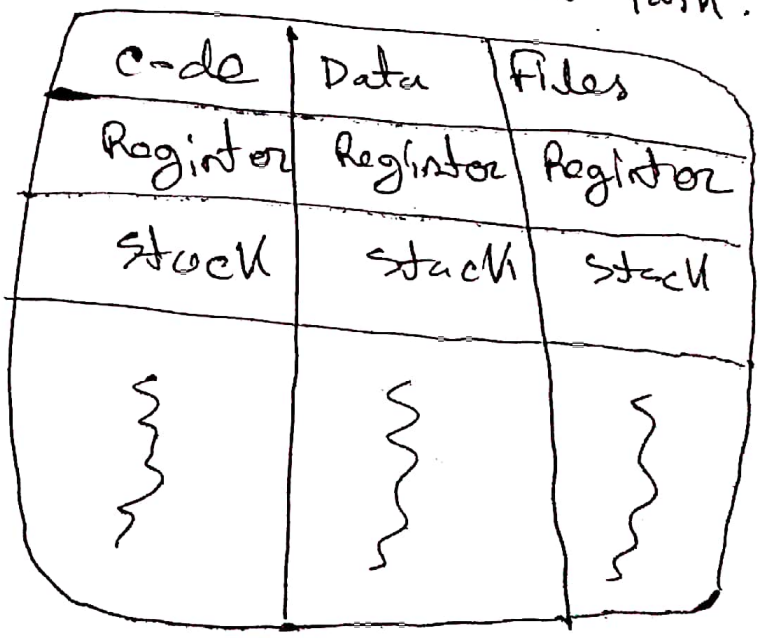
A register set

A stack

- **A process** can contains **different threades.**
- A traditional process has a **single thread** but if a process has multiple threads than it can perform more than one task.



Single thread Process.



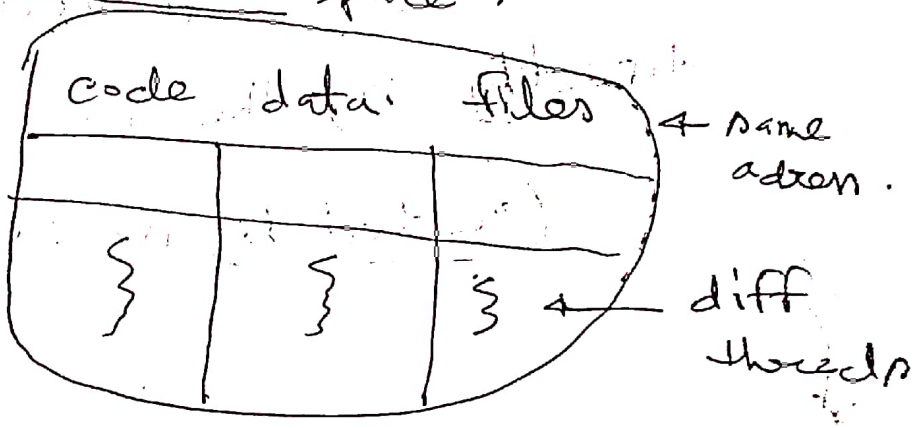
Multi threades Process.

Benifits :

4 catagories : (multi threading)

(i) Responsiveness : allows to continue running even if part of it is blocked.

(ii) Resource Sharing : it will allows an app to have several different threads of activity with in same adress space.



(iii) Economy : Allocating memory and resources for process creation is costly. Because threads share resources of the process to which they belong, it more economical to create and context-switch threads.

(iv) Utilization of multiprocessor :

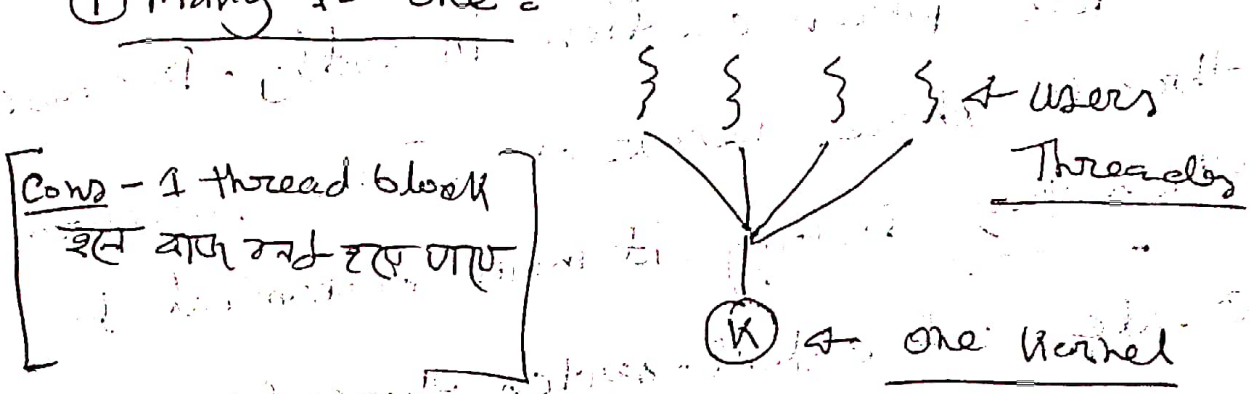
⊛ Types of threades :

① USER Threades : Support above the Kernel & managed without Kernel support.
(Created by the users.)

② Kernel Threades : Supported and managed directly by the Os.
(created by the OS)

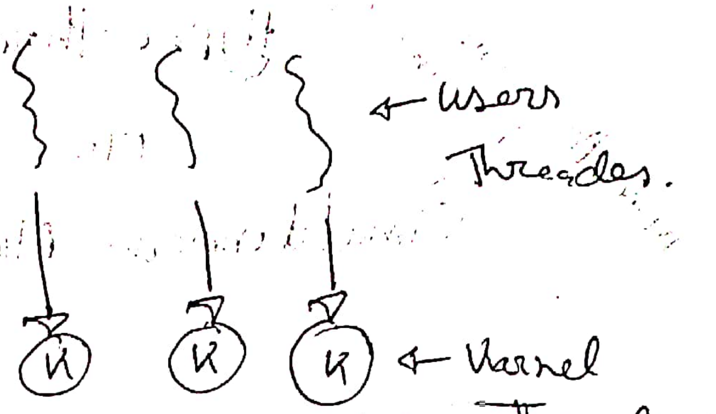
⊛ There are three common ways to establish

① Many to one :



✓ Many Threades (user) → one Kernel (managed)

(ii) One to one Model:



Pros-

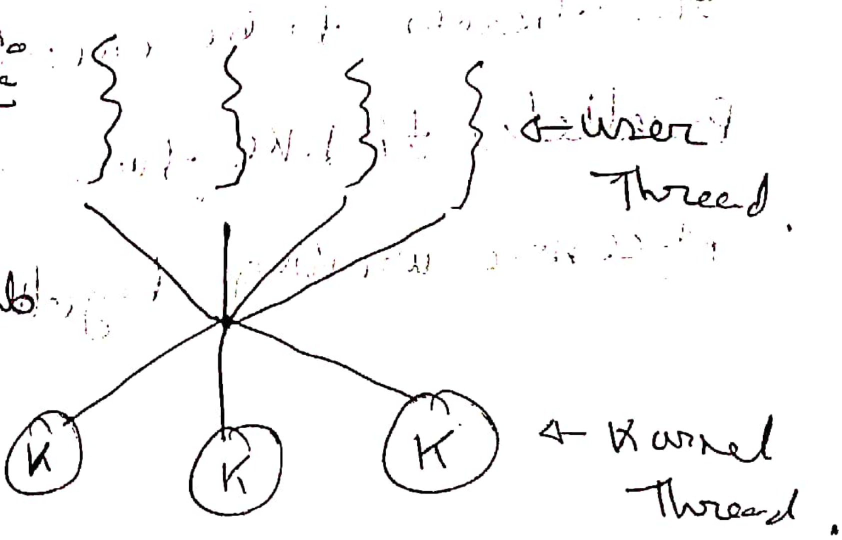
Note: • 1টি user thread block হলেও system off হবে না।

• more efficient.

Cons: • 1টি user এর জন্য 1টি kernel লাগে তাই costly হয়- hardware.

(iii) Many to Many:

✓ user thread numb >= Kernel threads



more than one threading in same system.

called by intel

hyper-threading

OR

multiple threads same times

Simultaneous Multithreading (SMT)

→ many of this multithreading going in same time.

→ enables the processor to execute two or more threades, at a same time. Since hyper-threading allows two streames to be executed in parallel, it's like two seperate processor working together.