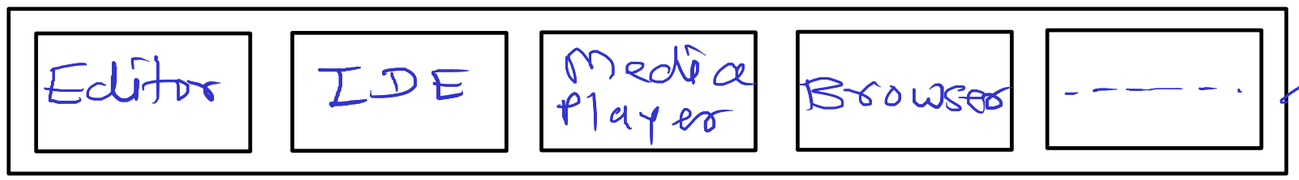
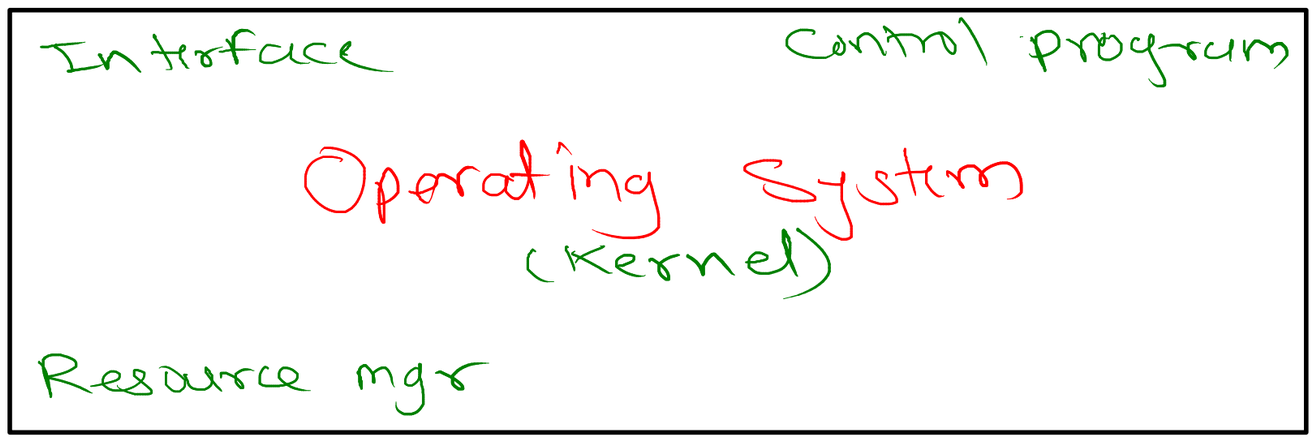
Operating System





- interface betn end user and hardware

- interface betn application softwares and hardware

- resource manager/ allocator which is managing all h/w resources

- control program which is controlling execution of all the programs which are running on the system

- CD/DVD/ISO - Core OS (Kernel) + Appln S/W + System Utilities

Functions of Operating System

(Kernel)

1) Process Management
2) CPU scheduling
3) Memory Management
4) File & IO Management
5) Hardware Abstraction

} Compulsory

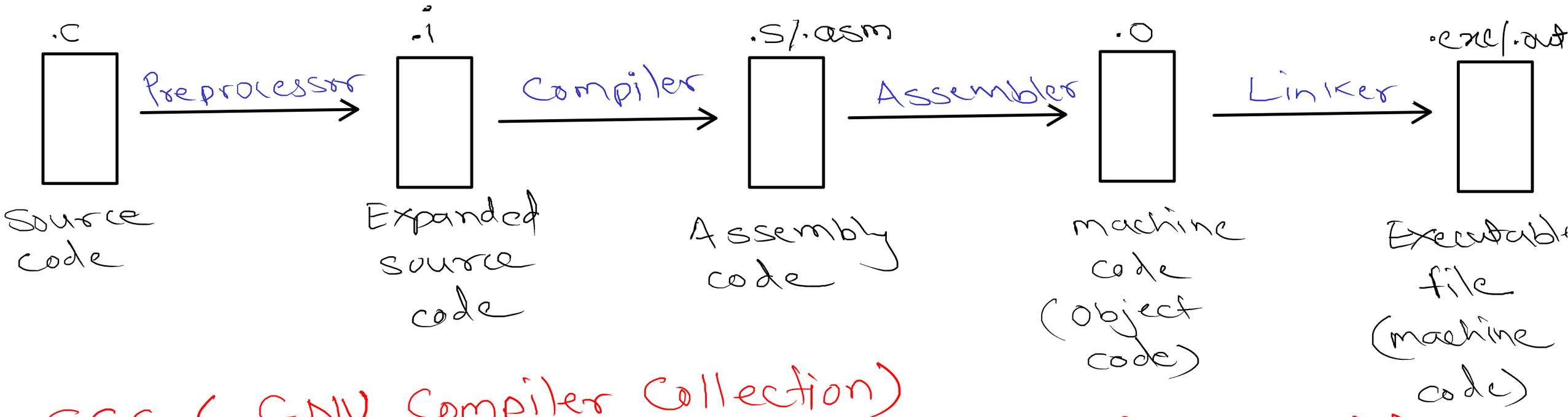
6) User Interfacing
7) Networking
8) Security and Protection

} optional

Process Management

Process – Program in execution

Program – set of instructions to machine (CPU)



(Toolchain)

GCC (GNU Compiler Collection)

- collection of tools

1) Preprocessor
2) compiler
3) Assembler
4) Linker
5) Debugger
6) Utilities (make)

Program

Exe Header

- Magic Number (identity to file format)
  - first 2 or 4 bytes
  - .exe – Portable ~~Executable~~ (MZ)
  - .out – Executable Linking Format (& ELF)
- info about executable file
  - type – CLI / GUI / Library
- info about remaing sections (size, start, ends)
- addr of entry point function

Text (Code)

- instructions of your program in the form of machine code

Data – static & global variables (initialised) `int num1=10;`

BSS – static & global variables (uninitialised) `int num2;`

RO Data – Read only data (String constants) `char *ptr="dac";`

Symbol Table

- info about symbols of your programs

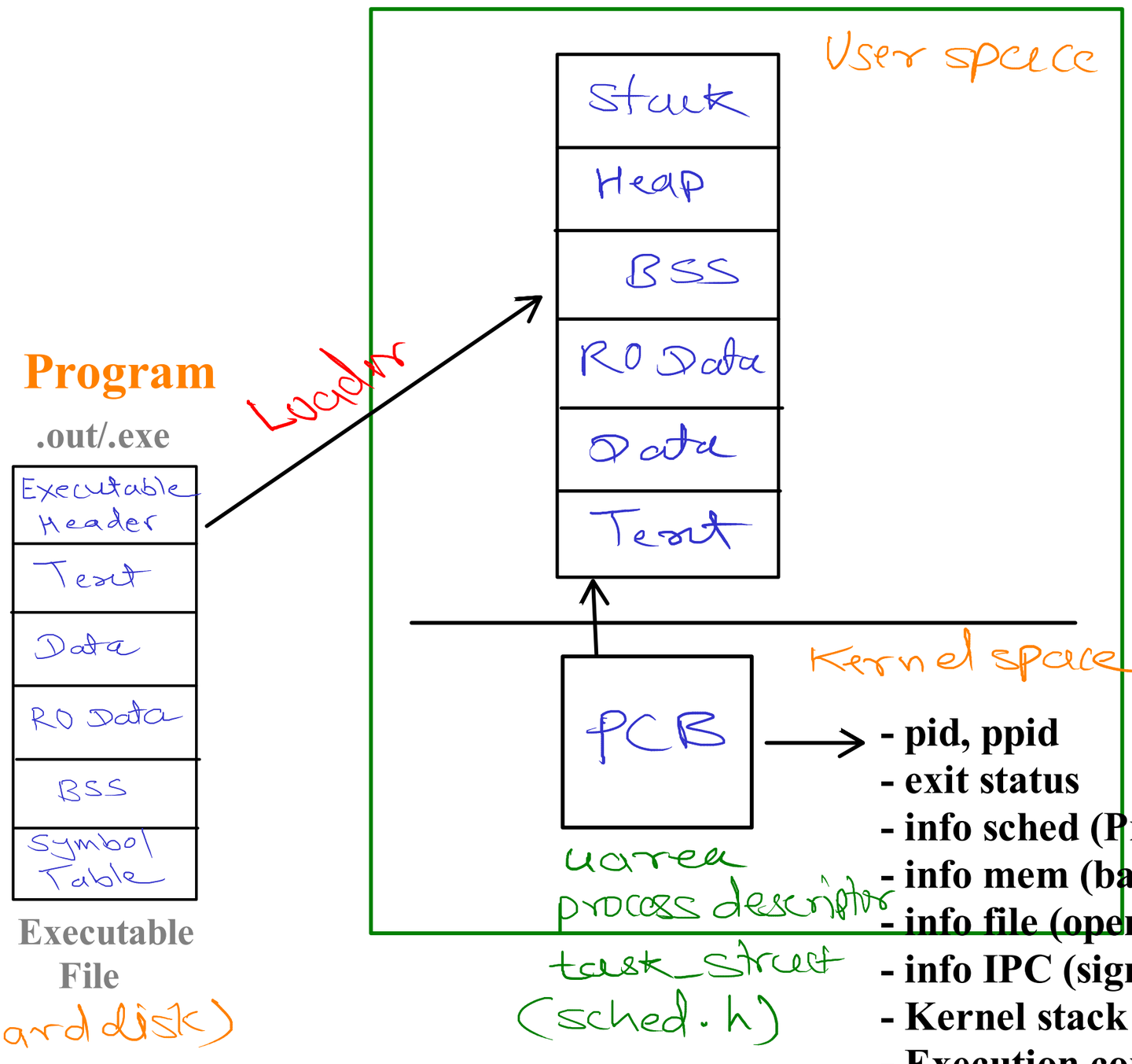
symbols
- variables – name, addr, section, type
- functions – name, addr, return type, no./type of args

Process

Process = All section of RAM + PCB

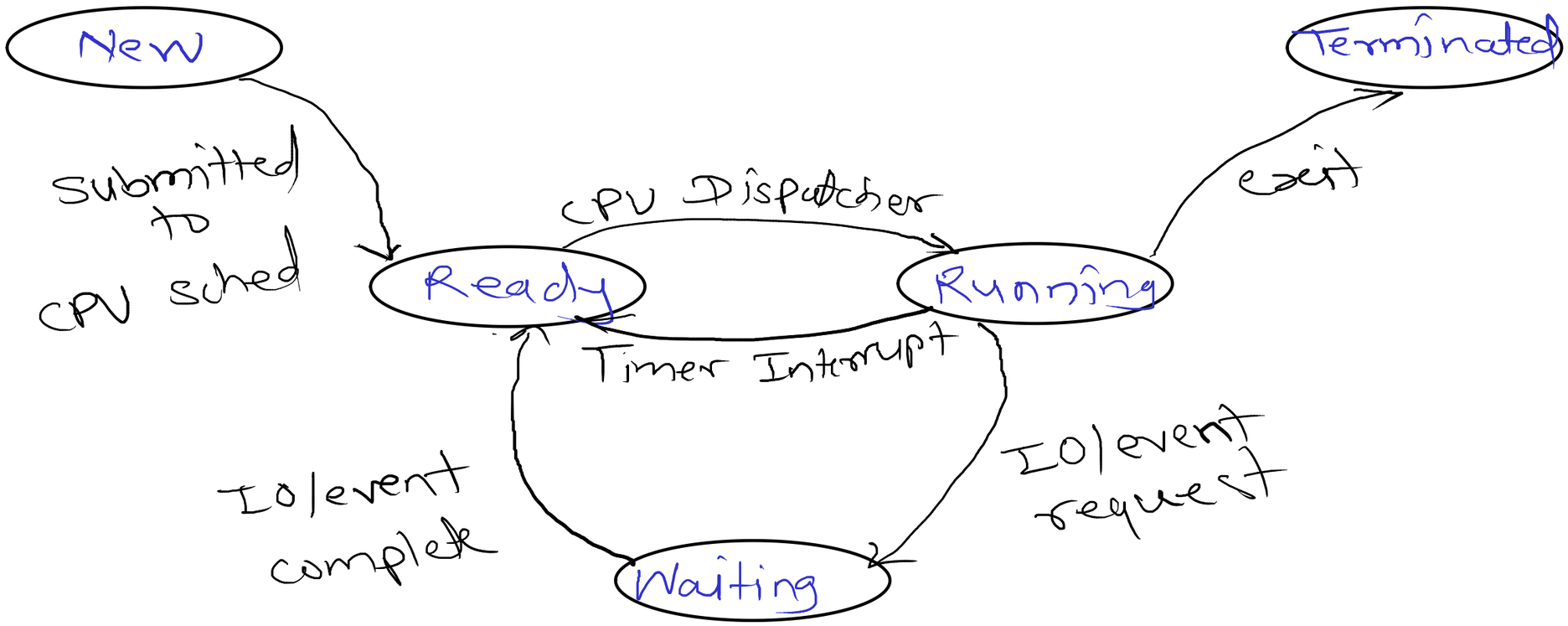
Stack
- FAR (local var)

Heap
- DMA

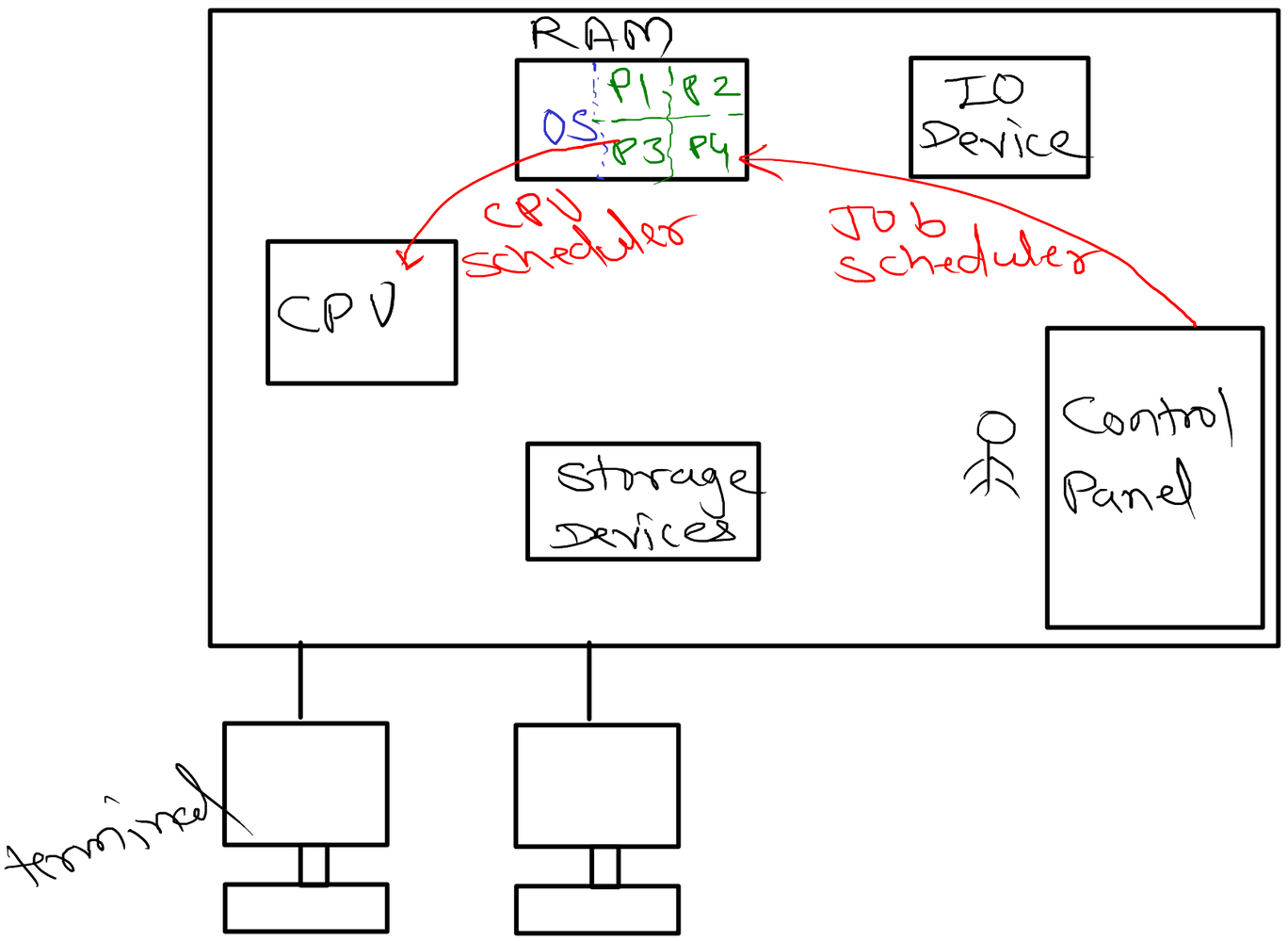


- pid, ppid
- exit status
- info sched (Priority, Algo, state)
- info mem (base and limit, segment/page table)
- info file (opened file, current directory)
- info IPC (signal...)
- Kernel stack
- Execution context

Process Life Cycle



Types of Operating System



1) Resident Monitor
2) Batch System
3) Multiprogramming system
   - degree of Multiprograming
     ↳ no. of programs loaded into RAM.
- CPU time/burst – time spent on CPU
- IO time/burst – time spent on IO
- CPU time > IO time – CPU bound
- IO time > CPU time – IO bound

4) Time sharing system/ Multitasking :–
   Response time < 1 sec

5) Multiuser system
   multiple users can access/control single computer

6) Multiprocessing system
   1) Symmetric multiprocessing
   2) Asymmetric multiprocessing

1) Process based
2) Thread based (Multithread)

OS's Data Structures

1) Job Queue / Process List
   - All the processes which are loaded into RAM

2) Ready Queue
   - All processes which are ready for execution on CPU
   - CPU scheduler always select process from ready queue

3) Waiting Queue
   - All processes which are waiting for some IO or event.
   - multiple (per device one) waiting queues are there

Types of Scheduling

1) Running → Terminated
2) Running → Waiting
3) Running → Ready
4) Waiting → Ready

} Non-preemptive
} Preemptive

Algorithms:
1) FCFS
2) SJF
3) Priority
4) RR
5) Fair share

CPU Scheduling Criterias

1) CPU Utilization: (Max)
   - desktop systems — 70%
   - server systems — 90%
2) Throughput: (Max)
   - amount of work done in unit time
3) Waiting time: (Min)
   - time spent by process in ready queue
4) Response time: (min)
   - time from arrival of process into ready queue upto first time scheduled on CPU
5) Turn Around Time: (Min)
   - total time spend by process in RAM(memory)

$$TAT = \text{CPU waiting} + \text{CPU time} + \text{IO waiting} + \text{IO time}$$

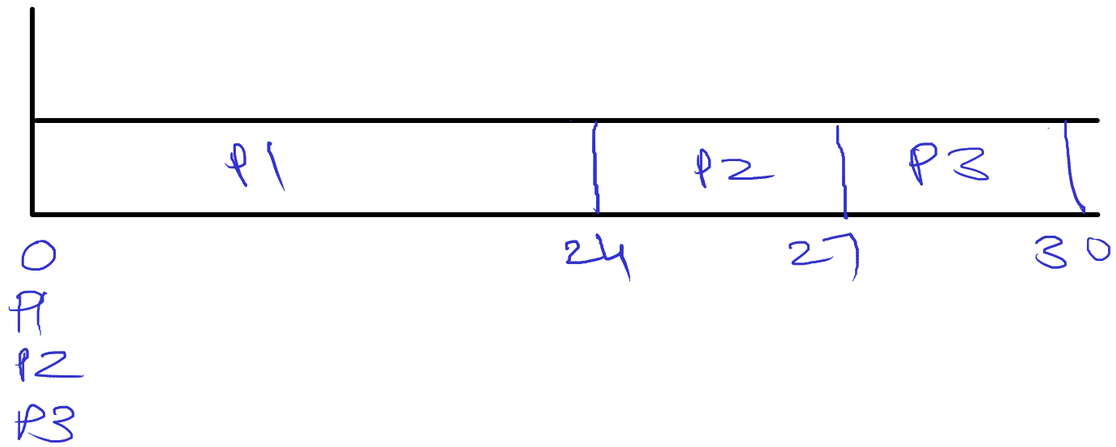
FCFS (First Come First Serve)

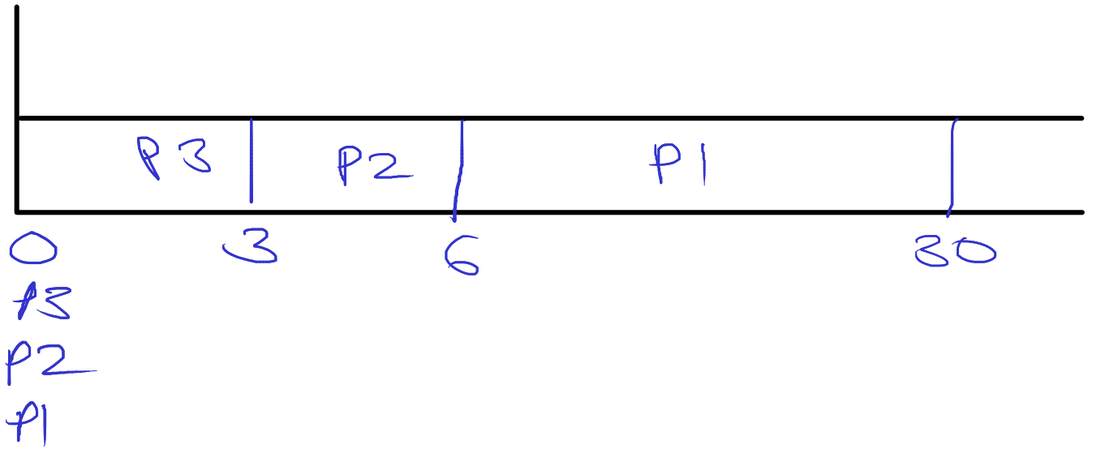
(Non Preemptive)

Process	Arrival	CPU Burst	WT	RT	TAT
P1	0	24	0	0	24
P2	0	3	24	24	27
P3	0	3	27	27	30

Process	Arrival	CPU Burst	WT	RT	TAT
P3	0	3	0	0	3
P2	0	3	3	3	6
P1	0	24	6	6	30

Gantt's chart





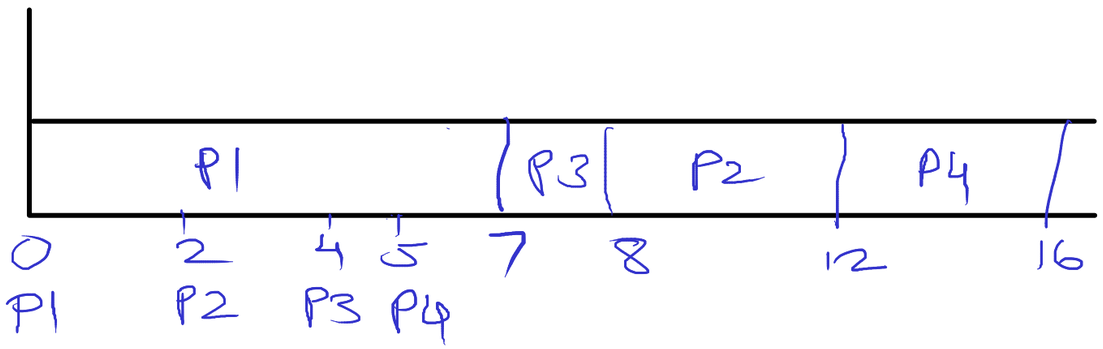
Convoy Effect:

due to longer process all other processes has to wait for longer time

SJF (Shortest Job First)

(Non-Preemptive)

Process	Arrival	CPU Burst	WT	RT	TAT
P1	0	7	0	0	7
P2	2	4	6	6	10
P3	4	1	3	3	4
P4	5	4	7	7	11

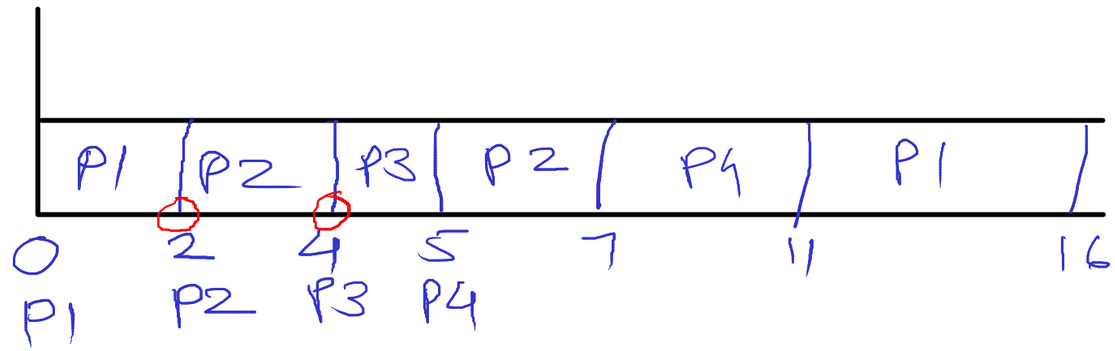


(Preemptive)
(Shortest Remaining Time First)

Process	Arrival	CPU Burst
P1	0	7
P2	2	4
P3	4	1
P4	5	4

5
2
X
9

WT	RT	TAT
9	0	16
1	0	5
0	0	1
2	2	6
	0.5	



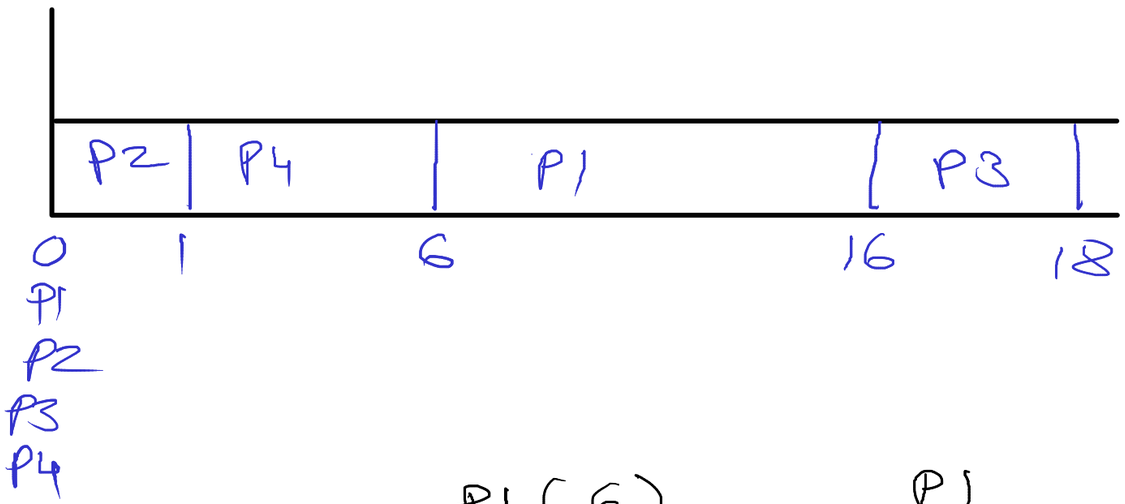
Starvation:

due to longer CPU time/burst, process will not get enough CPU time for execution.

Priority

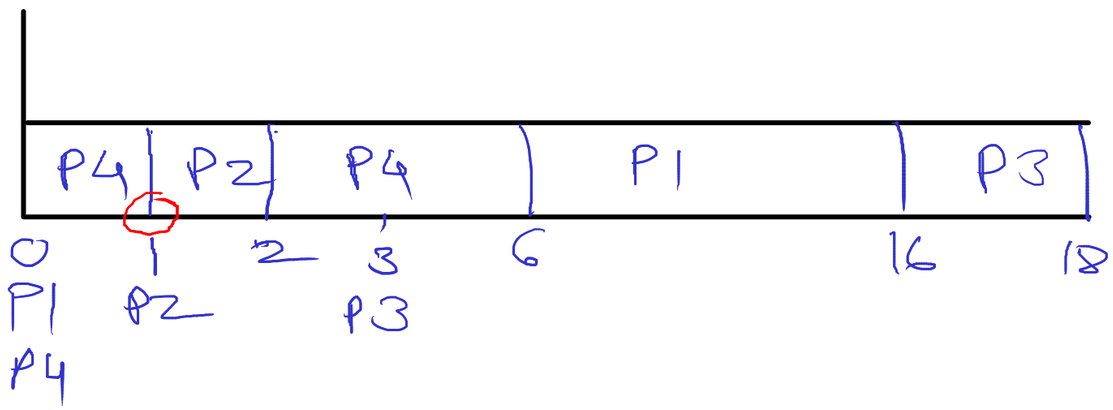
(Non Preemptive)

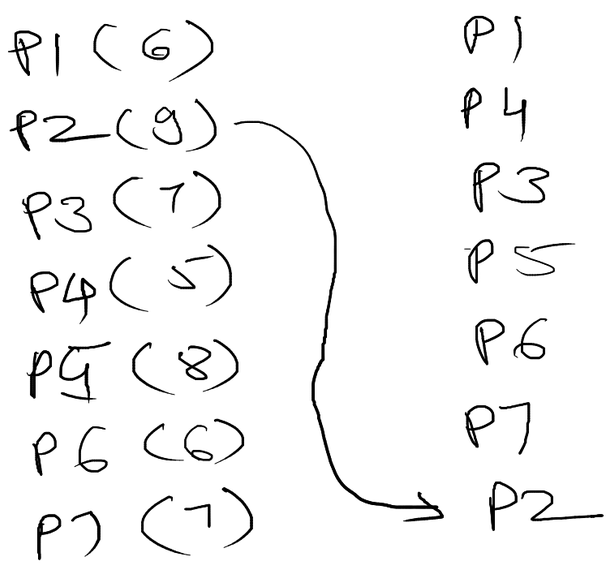
Process	Arrival	CPU Burst	Priority	WT	RT	TAT
P1	0	10	3	6	6	16
P2	0	1	1 (H)	0	0	1
P3	0	2	4 (L)	16	16	18
P4	0	5	2	1	1	6



(Preemptive)

Process	Arrival	CPU Burst	Priority	WT	RT	TAT
P1	0	10	3	6	6	16
P2	1	1	1	0	0	1
P3	3	2	4	13	13	15
P4	0	5	2	1	0	6





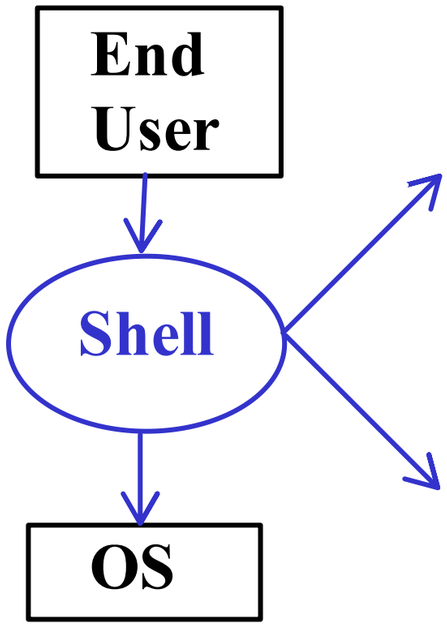
Starvation:
due to less priority process is not getting enough CPU time for execution

Aging:
increase priority of low priority process gradually

User Interfacing

Shell - intermediate between End user and OS

Shell - Command Interpretor



CLI (Command Line Interface)

Windows - cmd.exe, powershell.exe
Linux - bsh(sh), bash, csh, ksh, zsh

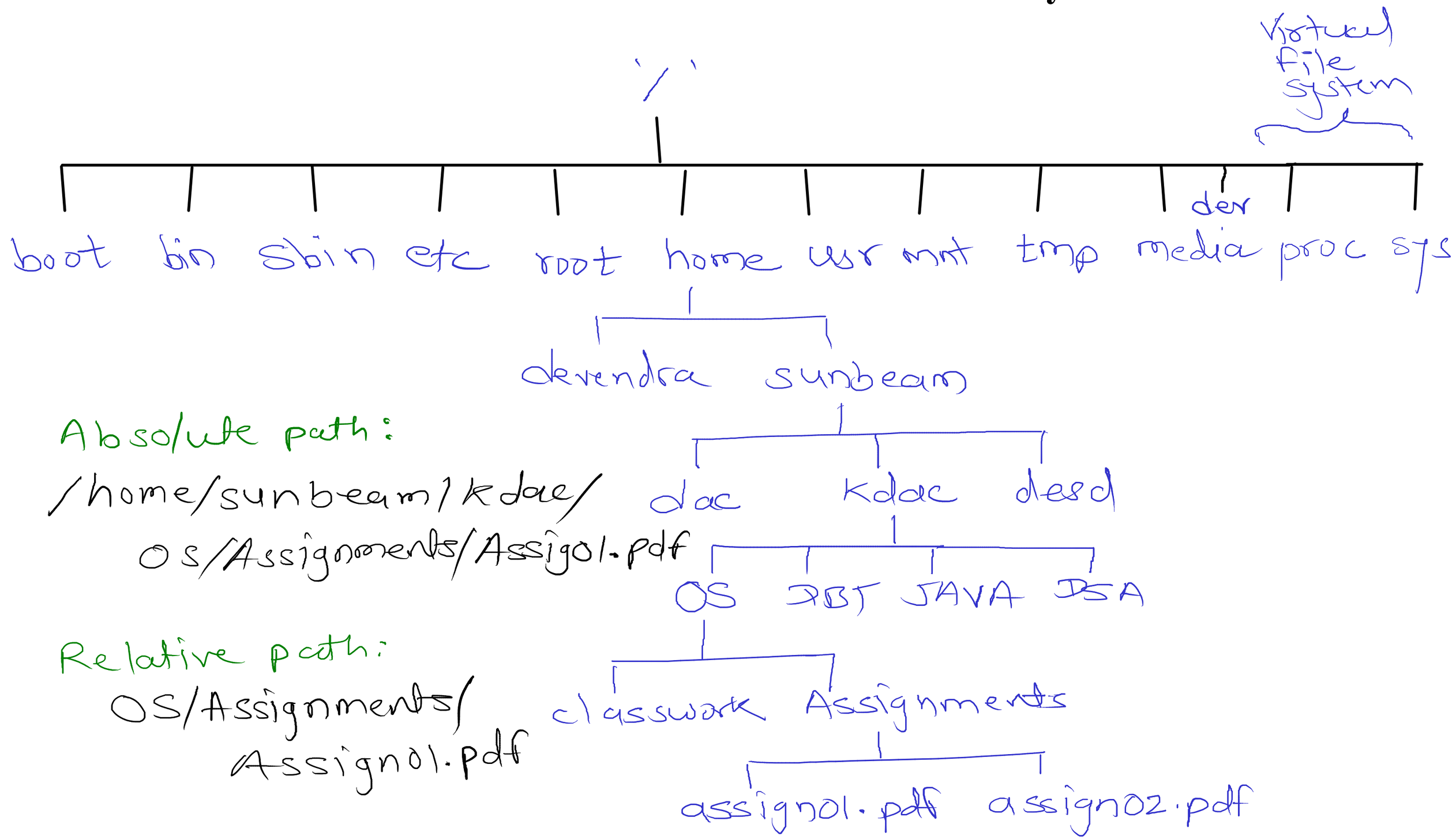
GUI (Graphical User Interface)

Windows - explorer.exe
Linux - KDE(Komman Desktop Env)
- GNOME(GNU Network Object Model Env)

- In Linux, default shell is bash(Bourne Again Shell)
- echo $SHELL
- to change shell - chsh

Linux File Structure

- Linux follows root "/" file structure
- In Linux file -> file and folder -> directory



Absolute path:

/home/sunbeam/kdac/OS/Assignments/Assign01.pdf

Relative path:

OS/Assignments/Assign01.pdf

Disks and Partitions - Naming Conventions

Disk:
- Windows – disk0, disk1 ----
- Linux – /dev/sda, /dev/sdb ----

Partition:
- Windows – c:, d:, e:
- Linux – /dev/sda1, /dev/sda2, /dev/sda3