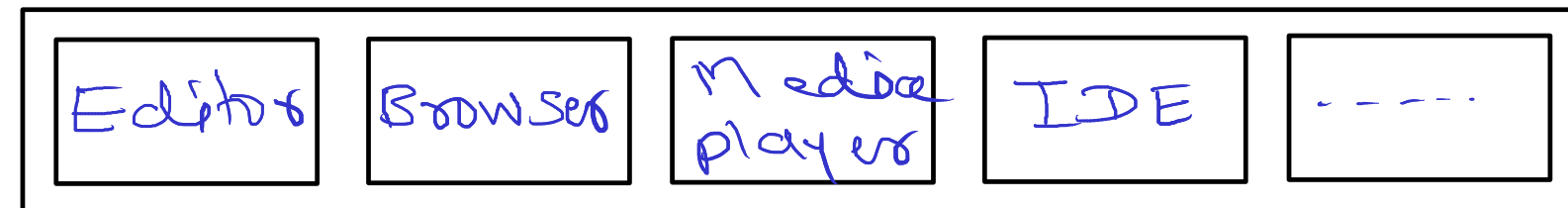


Operating System

End user

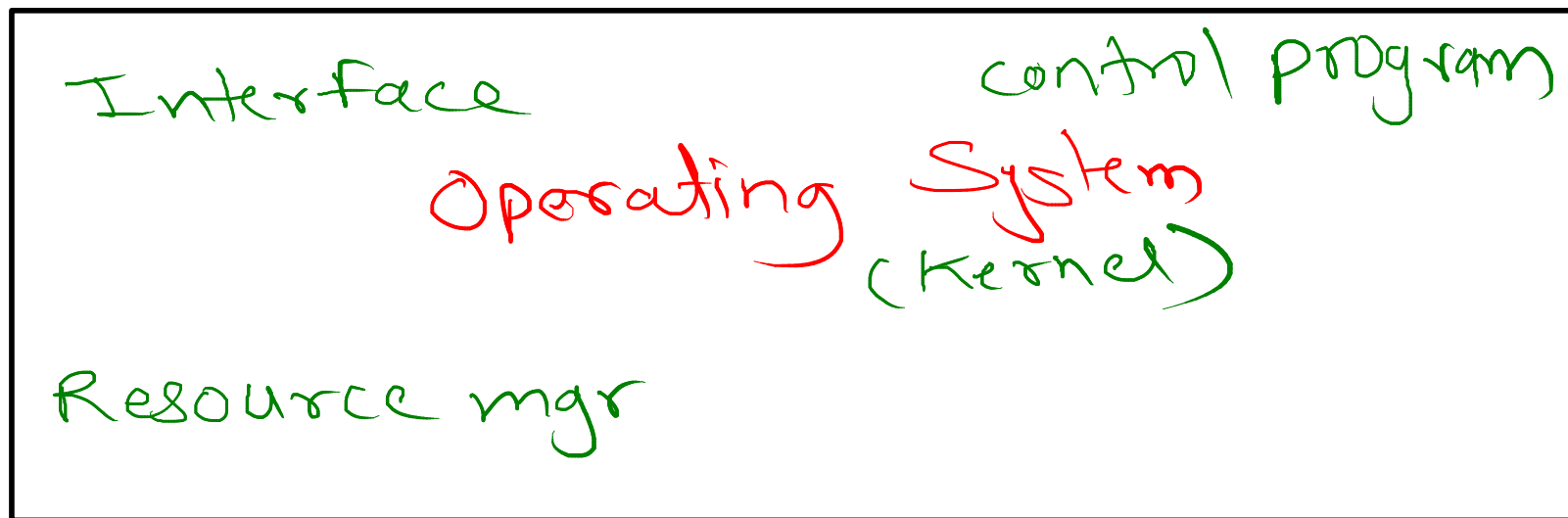


Appⁿ
S/W

— interface betⁿ end user and hardware

— interface betⁿ Appⁿ S/W (programs) and hardware.

— resource manager/ allocator which is managing all limited h/w resources.



computer
h/w

— control program which is controlling execution of all the programs

— CD/DVD/ISO : core OS + User Appⁿ + System Utilities
(Kernel)

Functions of Operating System (Kernel)

- ✓ 1> Process Management
- ✓ 2> CPU Scheduling
- ✓ 3> Memory Management
- 4> File and IO Management
- 5> Hardware Abstraction

Compulsory

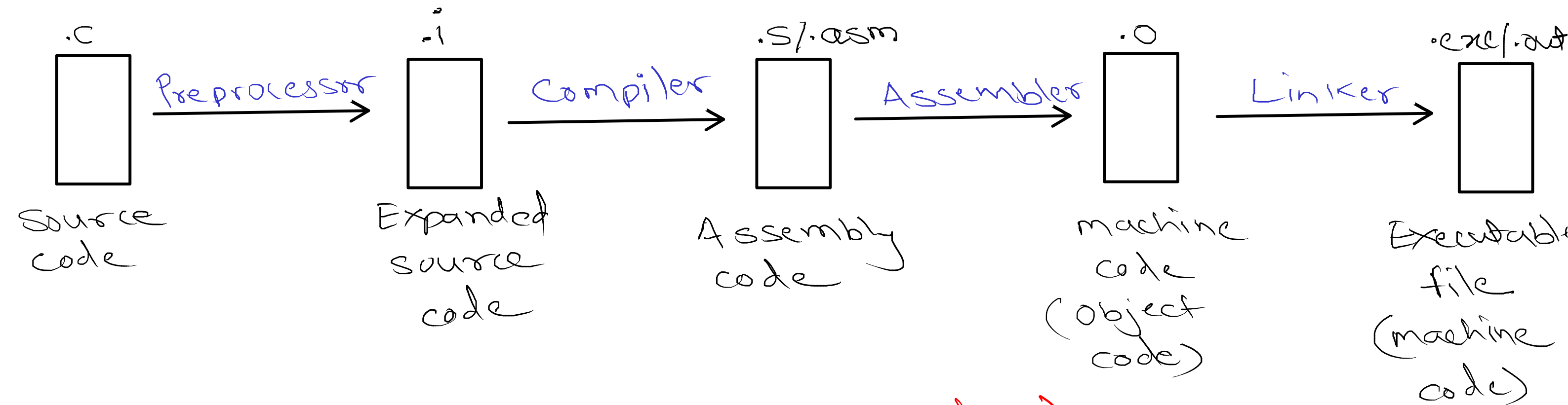
- 6> User Interfacing
- 7> Networking
- 8> Security and Protection

optional

Process Management

Process – Program in Execution

Program – set of instruction for machine (CPU)



GCC (GNU Compiler Collection) (Toolchain)

– collection of tools

1) Preprocessor

2) compiler

3) Assembler

4) Linker

5) Debugger

⑥ Utilities (make)

Program

.out/.exe

Executable Header
Text
Data
RO Data
BSS
Symbol Table

Executable
File

Exe Header

- Info about executable file
- magic number (identity to file format)
- (2/4 bytes) • exe - Portable Executable (MZ)
- out - Executable Linking Format (ELF)
- type of application
(CLI based / GUI based / library)
- info remaining sections of exe (size, start, end)
- address of entry point function

Text (Code)

- instructions are stored in machine code format.

Data - static/global variables (initialised) `int num1 = 10;`

BSS - static/global variables (uninitialised) `int num2;`

RO Data - Read Only Information (string constants)

Symbol Table

- info about symbols

symbols {
- variables (name, addr, type, default value, section)
- functions (name, addr, return type, type/no. of arguments)

Process

Process = All section of RAM + PCB

RAM

user space

Stack
- FAR

Heap
- Dynamically allocated memory

area
process descriptor
task_struct
(sched.h)

Program

.out/.exe

Executable Header
Text
Data
RO Data
BSS
Symbol Table

Executable
File

(Harddisk)

Loader

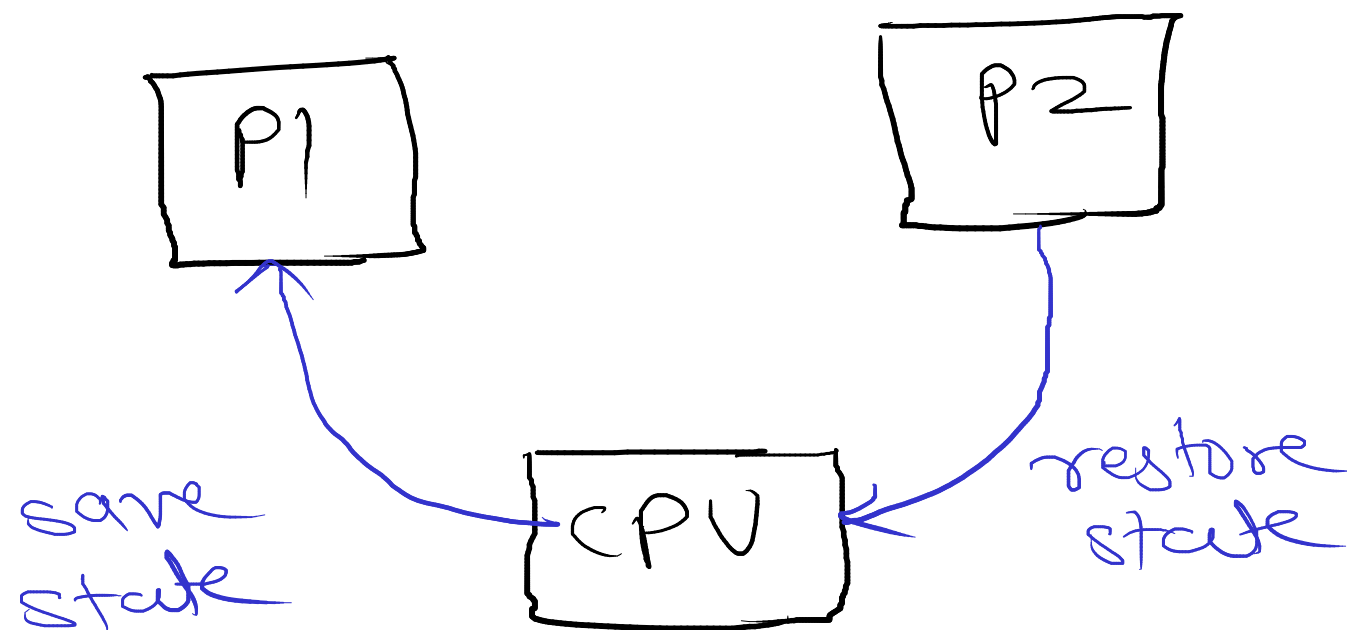
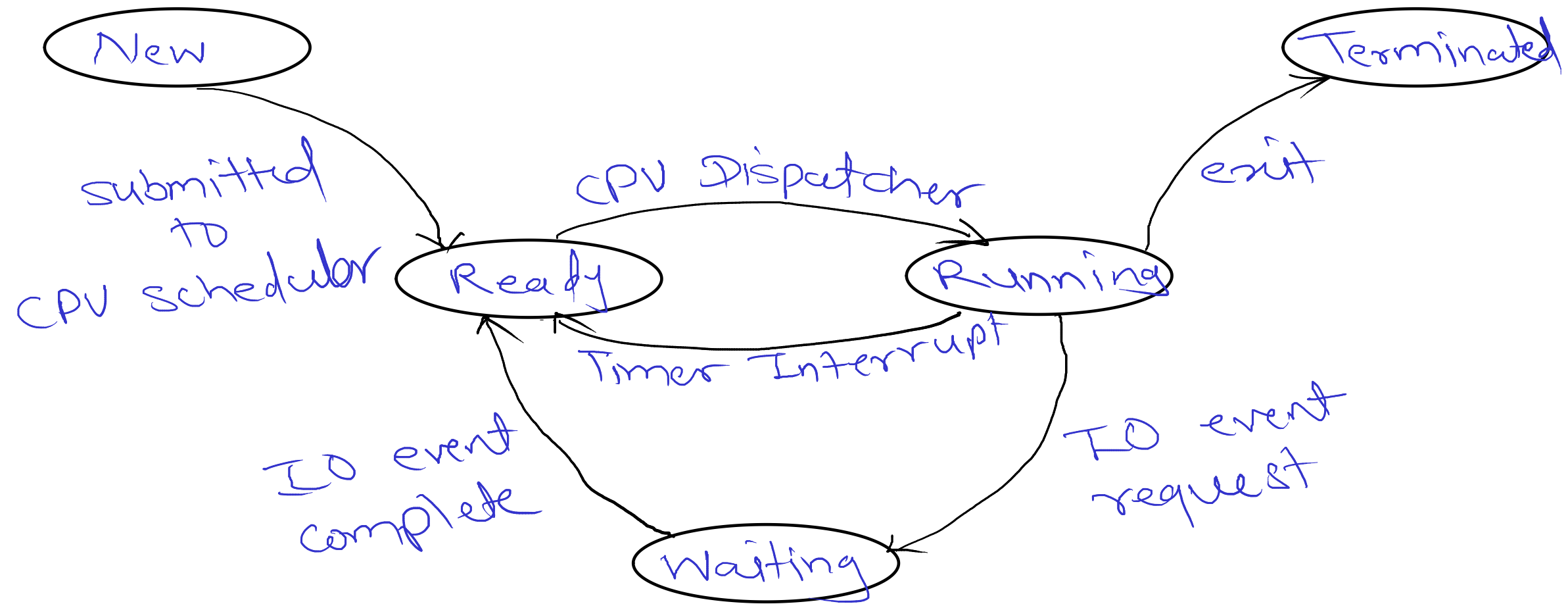
Stack
Heap
RO
BSS
Data
Text

Kernel space

PCB

- 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



{ Dispatcher Latency }

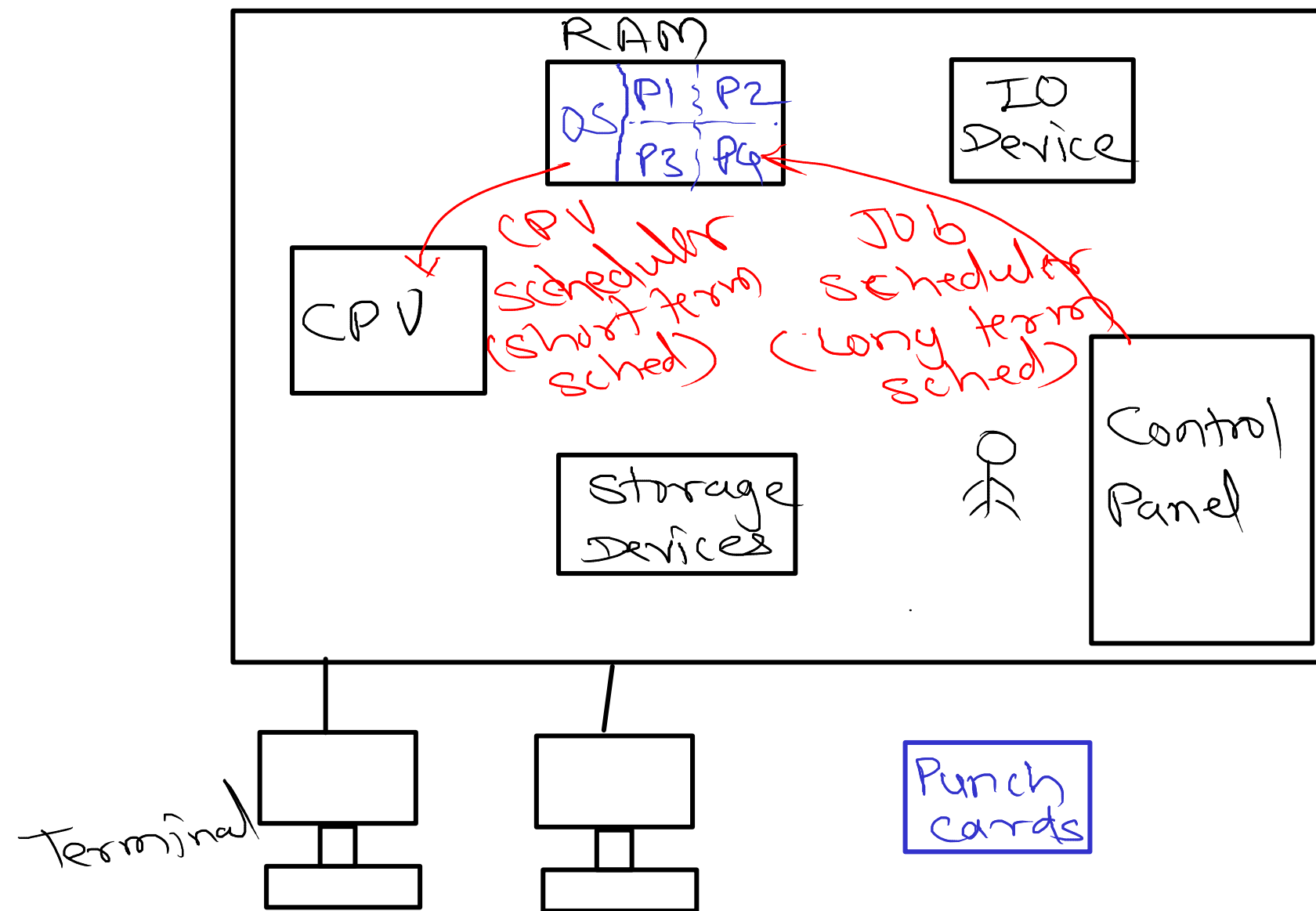
Context switching :

- save current running process and load newly selected process by scheduler on CPU.

Execution context :

- state of process
- values of CPU registers

Types of Operating System



⇒ multiuser systems
⇒ multiprocessing system

- 1) Resident Monitor
- 2) Batch system
- 3) Multiprogramming system
 - multiple programs are loaded into RAM.
 - degree of multiprogramming
 - $\text{CPU burst/time} - \text{time spent on CPU}$
 - $\text{IO burst/time} - \text{time spent on IO}$
 - $\text{CPU burst} > \text{IO burst} - \text{CPU bound}$
 - $\text{IO burst} > \text{CPU burst} - \text{IO bound}$

4) Time Sharing systems/
multitasking systems:

Response time $< 1\text{sec}$

- 1) Process based multitasking
- 2) Thread based multitasking (multithreading)

OS's Data Structures

1) Job Queue / Process List :

— All processes of system

2) Ready Queue

— processes which are ready for execution

3) Waiting Queue

— there are multiple waiting queues

— processes which are waiting for I/O are kept in to its respective waiting queue.

Types of Scheduling

- | | | |
|-------------------------|---|----------------|
| 1) Running - Terminated | } | Non-Preemptive |
| 2) Running - Waiting | | |
| 3) Running - Ready | } | Preemptive |
| 4) Waiting - Ready | | |

Algorithms:

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

CPU Scheduling Criteria

- 1) CPU Utilization (Ideal : Max)

- server systems : 90%
- Desktop systems : 70%

- 2) Throughput (Ideal : Max)

- amount of work done in unit time

- 3) Waiting time (Ideal : min)

- time spent by process in ready queue

- 4) Response time (Ideal : min)

- time from arrival of process into ready queue upto first time getting scheduled.

- 5) Turn Around time (Ideal : min)

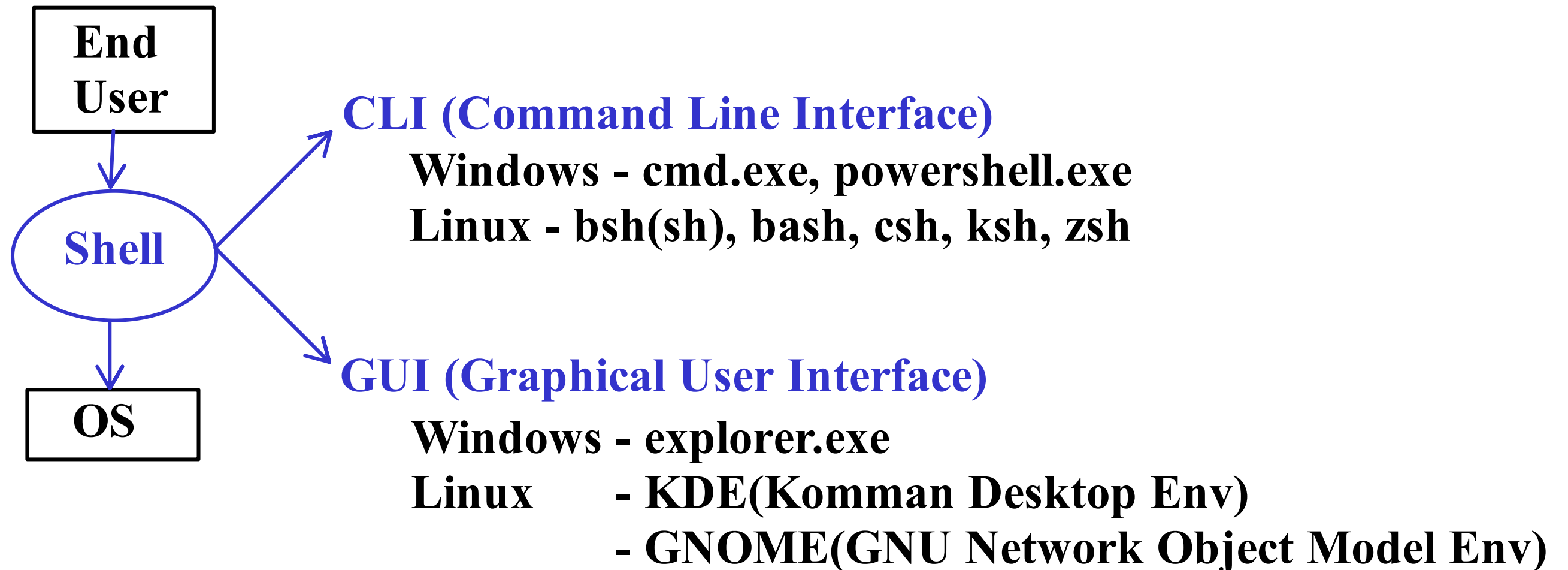
- time spent by process in memory (RAM)

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

User Interfacing

Shell - intermediate between End user and OS

Shell - Command Interpreter

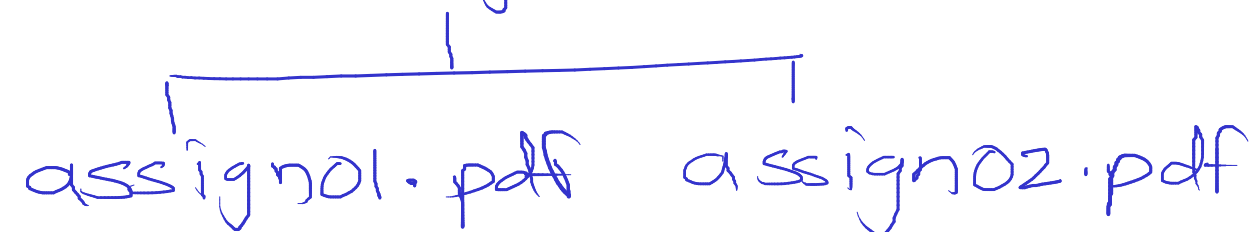
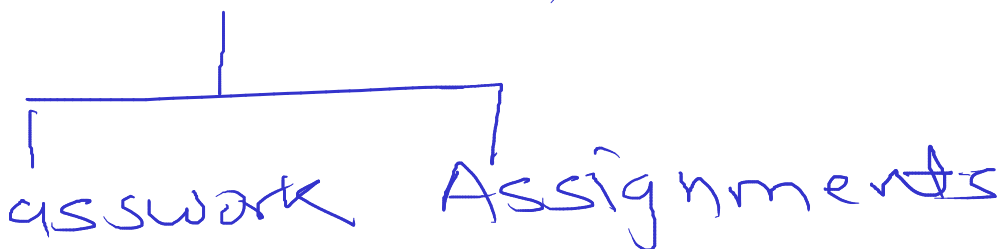
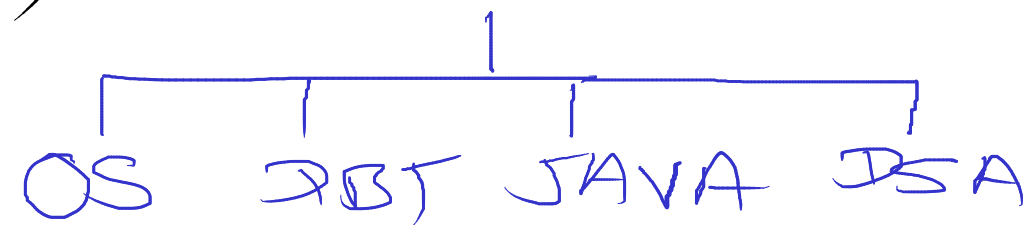
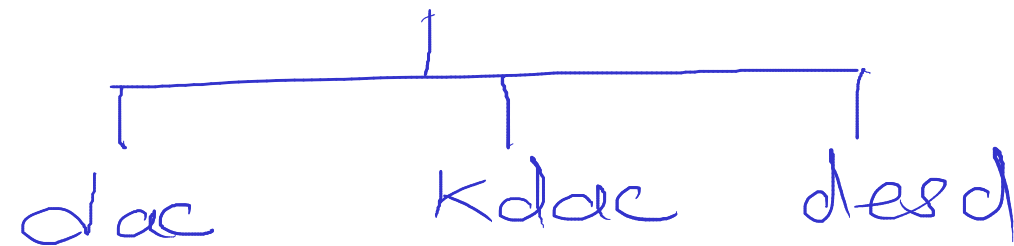
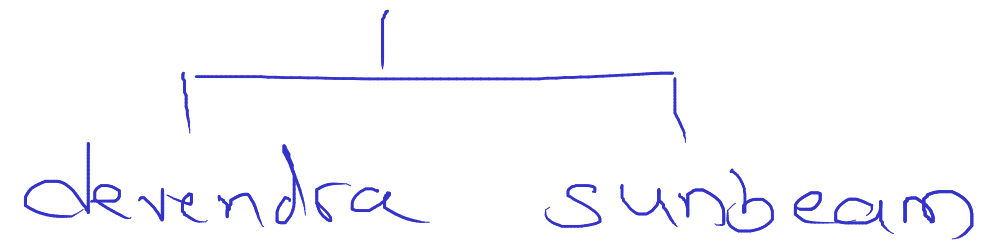


- In Linux, default shell is bash(Bourne Again Shell)
- echo \$SHELL
- to change shell - chsh

Linux File Structure

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

Virtual
File
System



Absolute path:

/home/sunbeam/dac/

os/Assignments/
assign01.pdf

Relative path: classwork Assignments

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