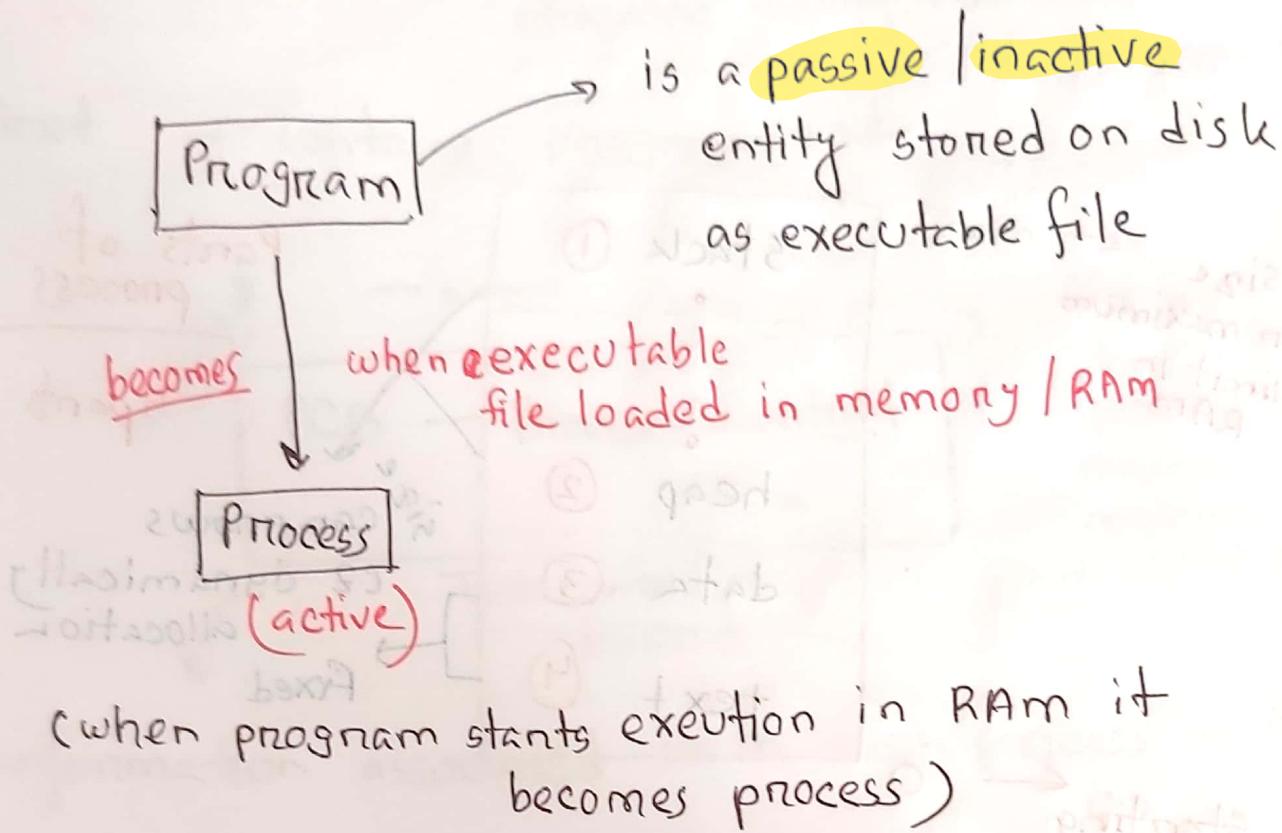


Chapter-3

Process



(one program can have several process)

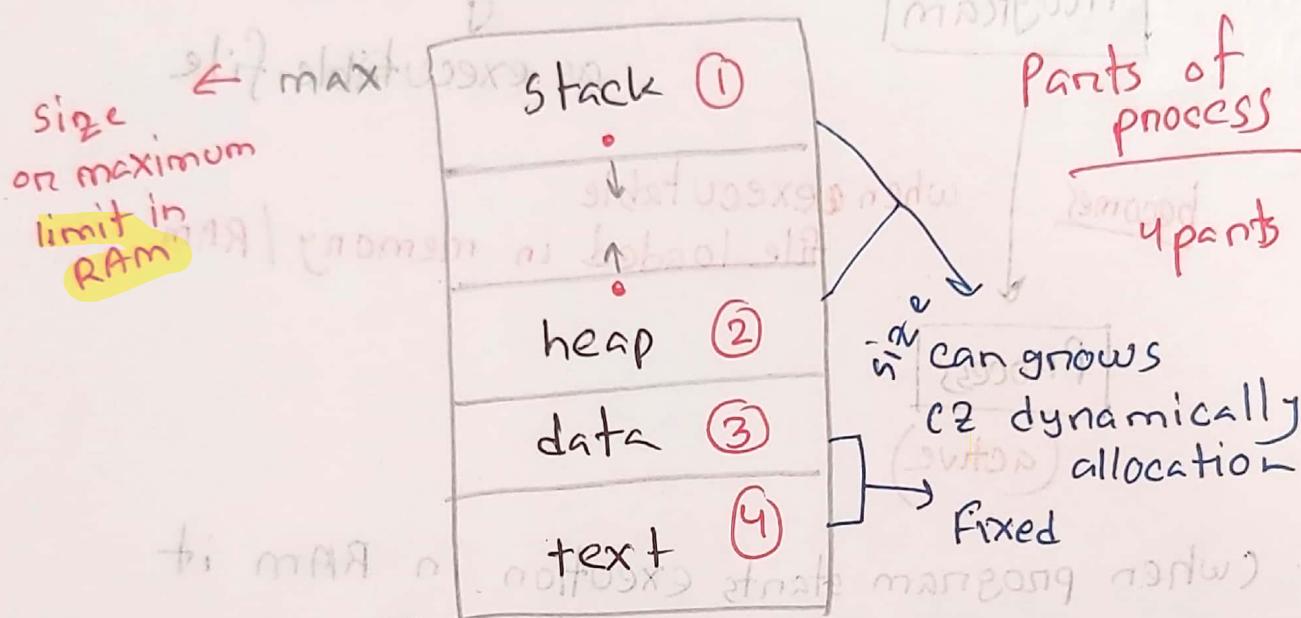
* To start a program you must bring it to ram

* program consists some instruction

Process

Components

→ Process is a program in execution



size can grows
c2 dynamically allocation

fixed

(How does 1 process look like in memory / RAM)

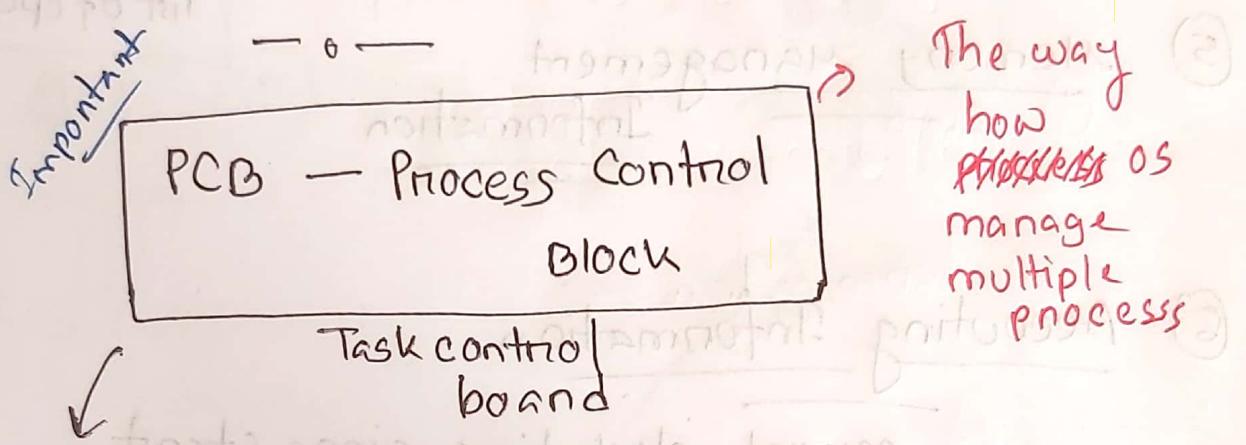
① stack → contains program code

→ contains temporary data

→ function parameters,
return addresses
Local variables

↓
inside in function

- ② Data Section → contain global variables
- ③ Heap → contain memory dynamically allocated during run time
- ④ Text → contains program code



information associated with each process in the

① Process state:

- i - (new)
- ii - (running)
- iii - (waiting)
- iv - (ready)
- v - (terminated)

② Program counter : Location of instruction to next execute the (where starts of execution)

③

CPU Registers:- content of all process centric registers

Process state
Process number
Program counter
Registers
memory limits
list of open file

④

CPU scheduling Information

Priorities, scheduling pointers

⑤

Memory management

Information

⑥

Accounting Information

cpu used, clock time since start

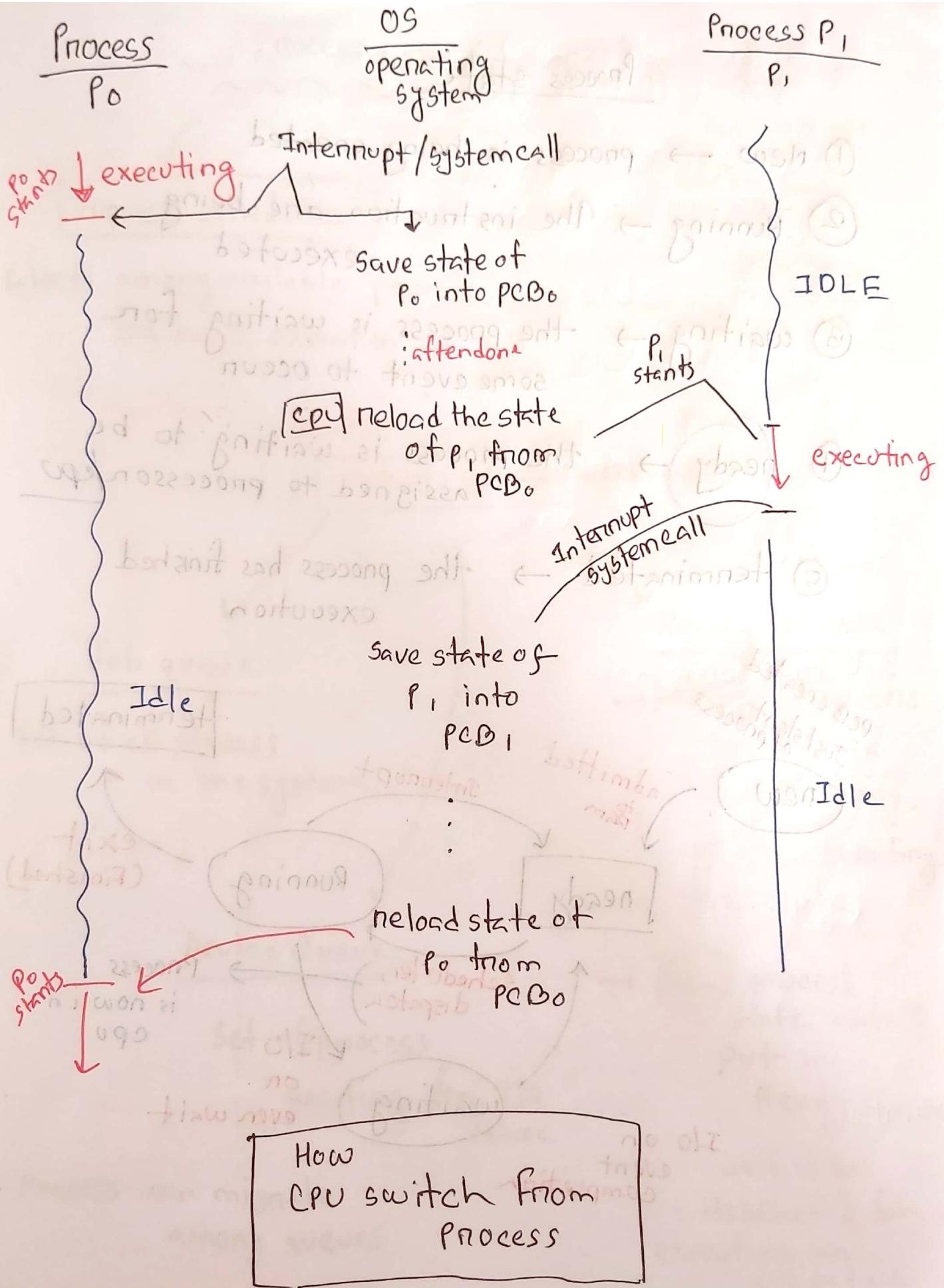
⑦

I/O status Information

I/O device allocated to process,
list of open files

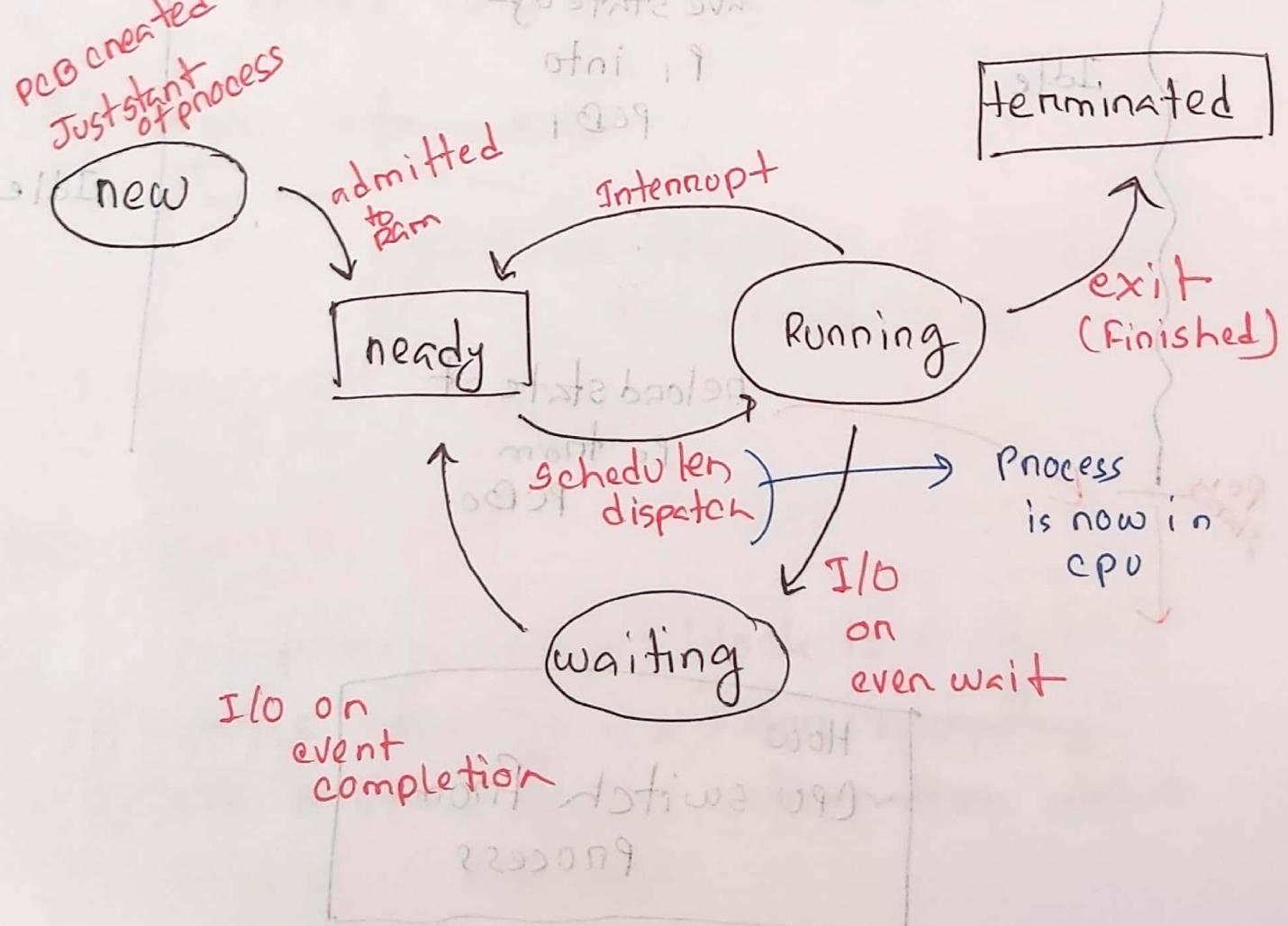
PCB definition

a process control block is a data structure used by computer operating system to store all the information about a process



Process state

- ① New → process is being created
- ② Running → The instruction are being executed
- ③ waiting → the process is waiting for some event to occur
- ④ ready → the process is waiting to be assigned to processor/cpu
- ⑤ terminated → the process has finished execution



Process scheduling

Process scheduler

selects among available processes
for next execution on CPU

maintains

scheduling

queues of processes

Job queue

Set of all processes
in the system

Device Queue

Set of processes

waiting for I/O
device

maximize CPU use
quickly switch
CPU for time sharing

Ready Queue

→ Set of all processes
residing in main memory,
ready and waiting to execute

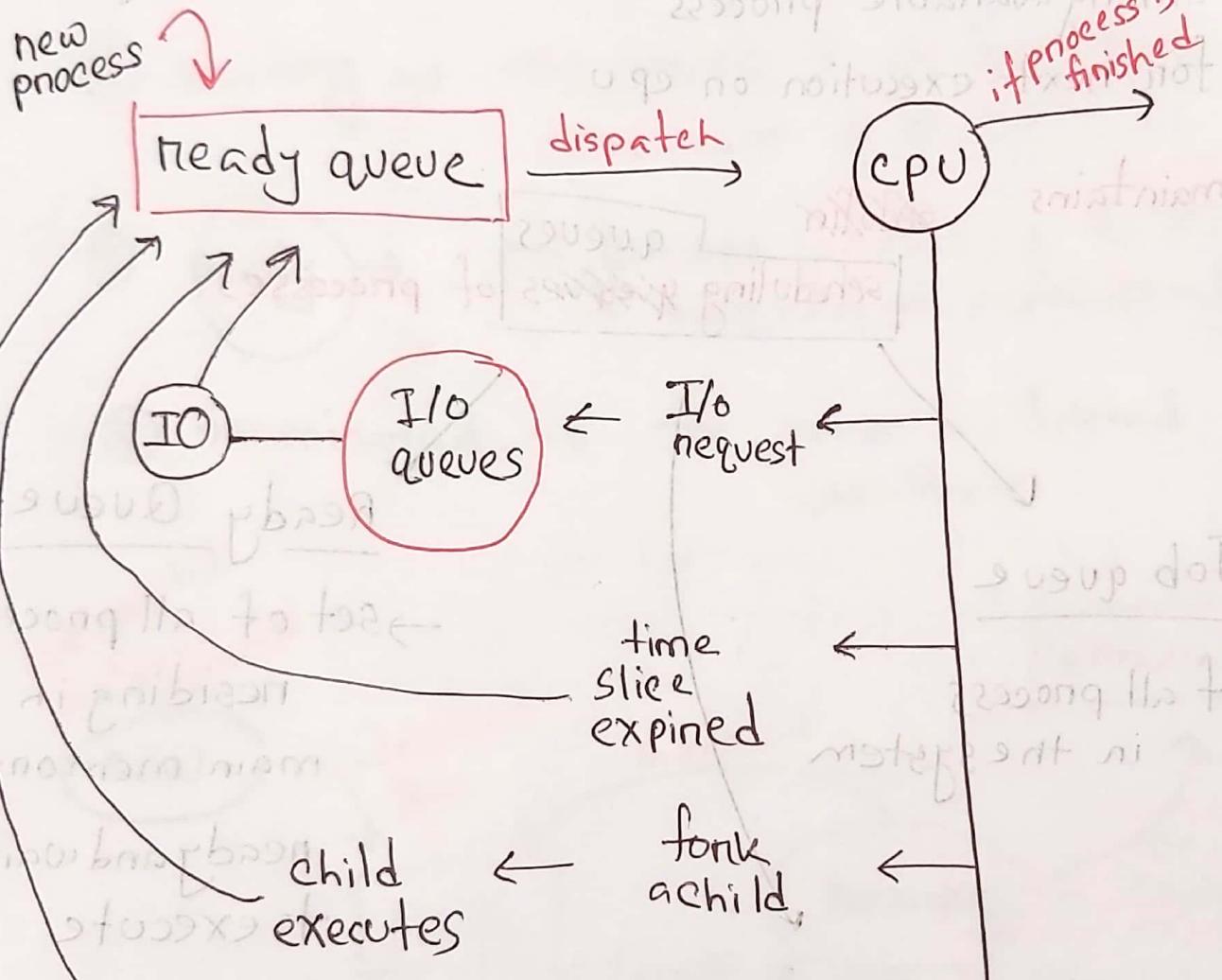
→ New process state initially put in Ready Queue

wait until its selected for execution on dispatch

* Process can migrate
among queues

Representation of Process scheduling

Queueing diagram



Schedulers

Short term scheduler

- CPU scheduler
- which sends a process to CPU
- select which process should be executed next and allocated to CPU
- must be very fast (is invoked / called frequently) (milliseconds)

Long term scheduler

- Job scheduler
- selects which processes should be brought into the ready queue in main memory
- Strives for good process mix
- may be slow (is invoked / called infrequently (seconds, minutes))

→ controls the degree of multiprogramming

definition ↓

how many programs should allow at same time on admitted to main memory

Types of process

① I/O bound process :-

- spend more time doing I/O than computations
- many short cpu bursts,
bursts

② CPU-bound process

- spend more time doing computations
- few very long cpu bursts,
bursts

Medium-term scheduler :

- remove process from memory
- store ⁱⁿ disk
- bring back in RAM to continue execution

Context Switch

(definition important)

① the system must **save** the state of old process

and **load** the saved state for new process via context switch

when a cpu switches

② to another process

it is called context switch

(context of process represented in PCB)

while context switch system does not do useful work

definition :- ① + ②

Process Creation

Parent process create **child** process which in turn create other process , forming a tree of processes

① Chrome Browser is a multiprocess with 3 different type of process

④ Browser process : manage user interface, disk, network I/O

② Render process : render web pages, deals with HTML and javascript

③ Plug in process : for each type of plugin

⑤ + ① - ; nothing

— o —

Based no Interprocess communication

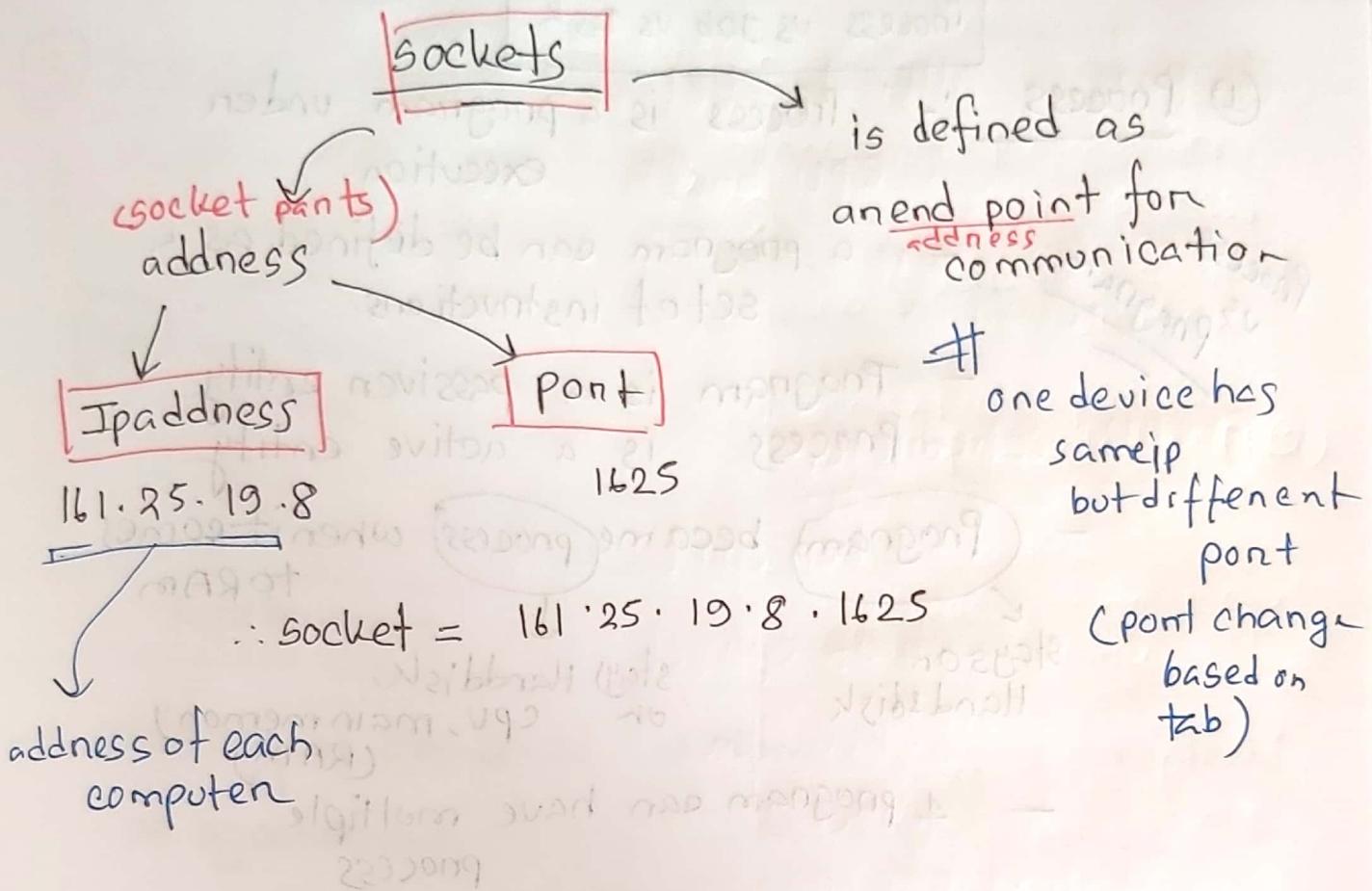
Process type

① Independent process

(not affected by other process)

② Cooperating process

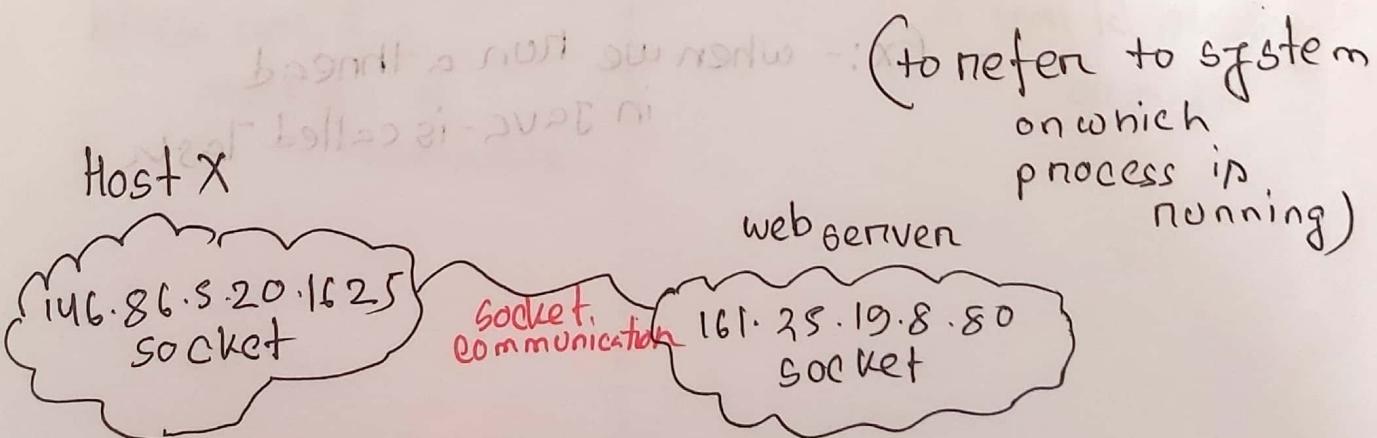
(affected by another process)



communication
consists between a pair of sockets

port < 1024 → well known ports
(used for standard
Internet service)

special IP address: 127.0.0.1



Process vs job vs Task

① Process

Process is a program under execution.

— a program can be defined as a set of instructions.

— Program is a passive entity
Process is an active entity

- Program became process when it comes to RAM
- 1 program can have multiple process

stays on Harddisk

stays Harddisk,
or CPU, main memory
(RAM)

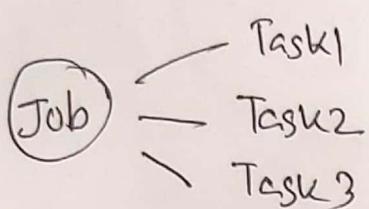
② Task :- Task is a unit of work being executed

Task in OS may be synonymous with processes

→ Task is a sub part of job

Ex:- when we run a thread in JAVA - is called Task

③ Job :- → a job is a complete unit of work under execution



- a job consists many task which in turn consists of many process

Synchronous

Asynchronous

→ Blocking is considered as synchronous

→ Nonblocking is considered as asynchronous

→ Blocking send has the sender block until the message is received

→ Non blocking send has the sender send the message and continue

→ Blocking receive has the receiver block until a message is available

→ Non blocking receive has the receiver receive a valid message or null

④ Process alternates between CPU bursts and I/O bursts

⑤ CPU scheduler:-

when CPU becomes idle, it is the job of CPU scheduler / short term scheduler to select another process from the ready queue to run next on CPU.

⑥ Scheduler / dispatcher :- is a module that gives the control of CPU to the process selected by the scheduler.

Types of sockets

actually
types

- ① Datagram socket
(connectionless)
- ② Stream socket
(connection-oriented)
- ③ Raw socket
- ④ Sequenced packet socket

access
Prot col

UDP
(UDP)

User Datagram
(TCP)

Transfer control
Protocol
(ICMP)

Internet
control message
protocol
(ICMP)

(IDP)

Internet
date protocol

Degree of multiprogramming

Degree of multiprogramming describes the maximum number of processes that a single processor system can accommodate efficiently.

Process

Thread

- | | |
|---|---------------------------------------|
| ① Process means any program is in execution | ① Thread means a segment of a process |
| ② Process takes more time to terminate | ② Thread takes less time to terminate |
| ③ Takes more time for creation | ③ Less time for creation |
| ④ More time in context switching | ④ Less time for context switching |
| ⑤ Less efficient in communication | ⑤ More efficient in communication |
| ⑥ A system call is involved in it | ⑥ No system call is involved |
| ⑦ Process has Process control block | ⑦ Thread has Thread control block |

Process scheduling

is the activity of process manager that handle the removal of the running processes from CPU and the selection of another process on the basis of a particular strategy.

Loop back address

A loop back address is distinct reserved IP address 127.0.0.1 to refer to system on which process is running.

what is API?

Application program Interface (API)

is code that allows two software programs to communicate with each other.

The principle of least privilege

the principle of least privilege states that a subject should be given only those privileges needed for it to complete its task.

- In UNIX, user is domain. True or False?

- true

what is kernel

→ is a central component of OS.

that manages operation of computer and hardware

→ acts as a bridge between applications and data processing performed

at hardware level using

IPC and system calls

Objectives

- To control disk, memory, task management

- To decide state of incoming process

- To establish communication between user level application and hardware

① what is system call ? explain different type of system call.

→ **System call** is a way for programs to interact with operating system.

5 types system call

② Process call :-

- deals with process such as creation & termination

Example:-

(Window)

CreateProcess()

Exit Process()

(Linux)

fork()

exit()

wait()

()

② File management :-

- responsible for file manipulation
creating, reading, writing into file

Example:-

(x00)

()

(window)

• CreateFile()

ReadFile()

WriteFile()

CloseHandle()

(zwlobn)

• open()

read()

write()

close()

(Linux)

• open()

read()

write()

close()

③ Device management :-

- responsible for device manipulation
- reading / writing device buffer

Example :-

(Windows) ~~int main()~~ (Linux)

SetConsoleMode()

ioctl()

ReadConsole()

read()

④

Information Maintenance

- handle information and its transfer

(Xunix) between OS and user programs

Example :-

(Windows)

(Linux)

(Windows)

(Linux)

SetTimer()

alarm()

sleep()

(Windows)

(Linux)

⑤ Communication

→ useful for inter process communication

Example :-

(Xunix)

(Windows)

(Linux)

(Windows)

CreatePipe()

pipe()

(Windows)

MapViewOfFile()

mmap()

(Windows)

(Windows)

(Linux)

⑧ How does communication take place in client server system? Explain

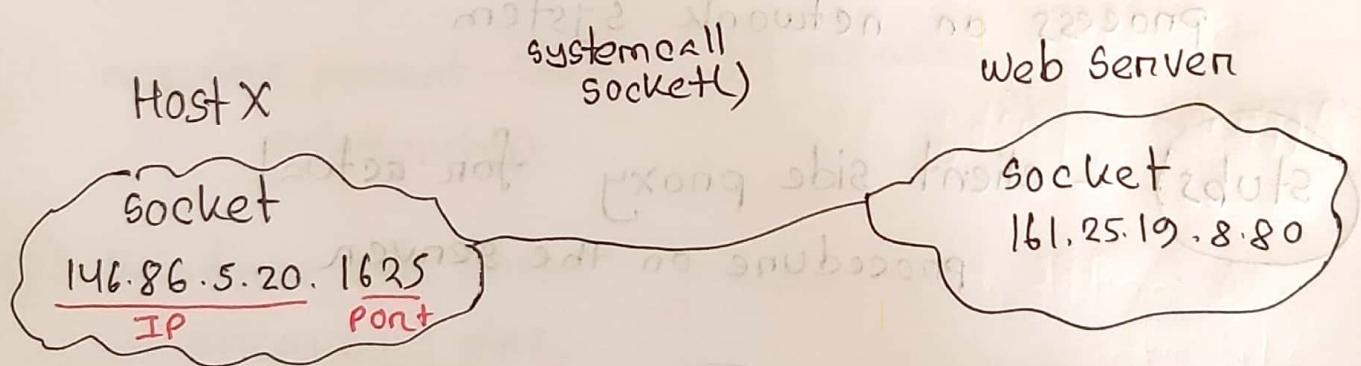
→ Client-server communication has two components client and server. The client sends request to the server and the server responds to the client request.

Three methods of client-server communication

① **socket** → low level communication channel

→ a socket is an endpoint for communication

→ a socket is identified by an IP address and a port number



Two major form

① Connection Oriented / TCP (Transmission Control Protocol)

→ guaranteed that all packets will arrive in good condition at other end and in order receive

② Connectionless (UDP - user data protocol)
→ no guarantee that packets will delivered
with undamaged and will receive in
particular order
→ UDP ~~for~~ Transmission > TCP

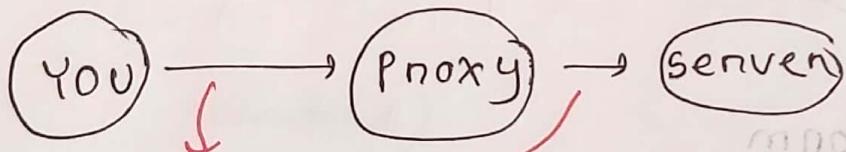
②

RPC - Remote Procedure Calls

- * in socket we communicate with process
 - * In RPC "functions / procedure"
- RPC abstracts procedure calls between processes on network system

Stubs

client side proxy for actual procedure on the server



③ Pipe

- acts as a conduit between two process
to communicate

Ordinary Pipe

- ① ordinary pipe allows communication in standard producer consumer style

- ② unidirectional

- ③ produces write to one end and consumer needs at other end

- ④ requires parent-child relationship

- ⑤ windows calls these anonymous pipe

Named pipe

- ① more powerful than ordinary pipe

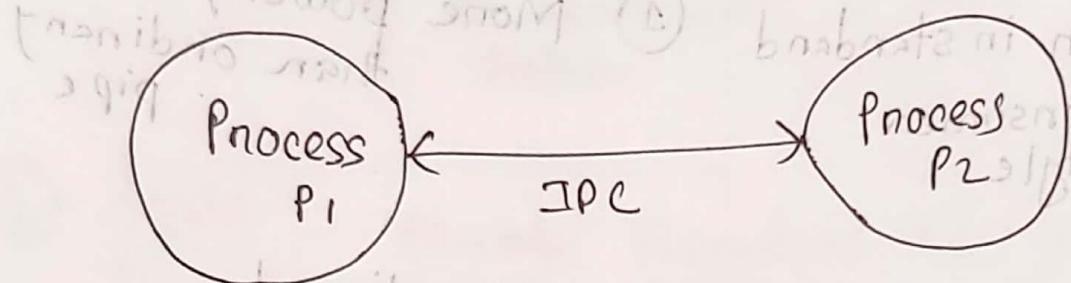
- ② bidirectional

- ④ no need parent child relationship

IPC - Interprocess Communication

— is the mechanism provided by

OS that allows processes to communicate with each other



two types of process

- ① Independent process
- ② Co-operating process

Applic.

why IPC?

IPC helps achieving

- ① Computational speed up
- ② Modularity
- ③ Information & data sharing
- ④ Privilege separation
- ⑤ Process can communicate with each other and synchronize their action

Approaches for Interprocess Communication

