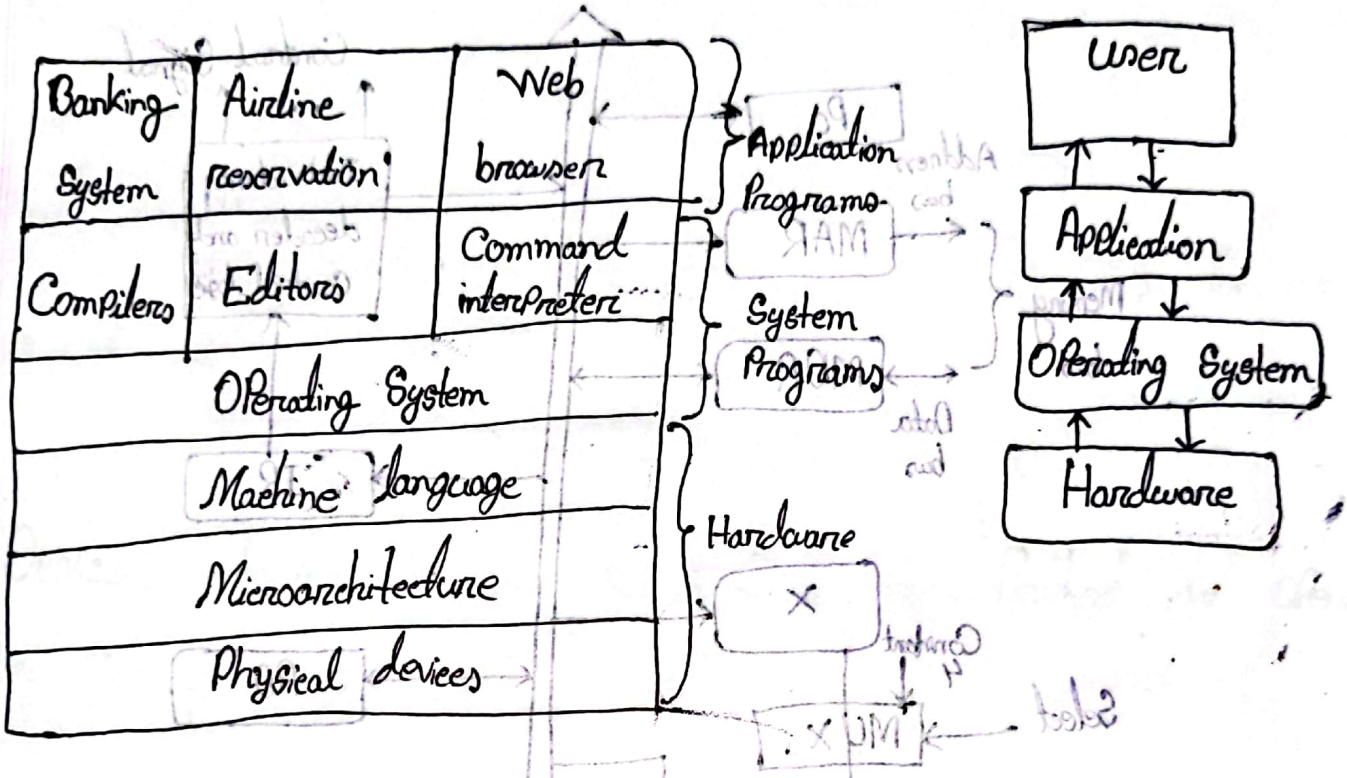
Components of a Computer System



Microarchitecture - Computer Organization:

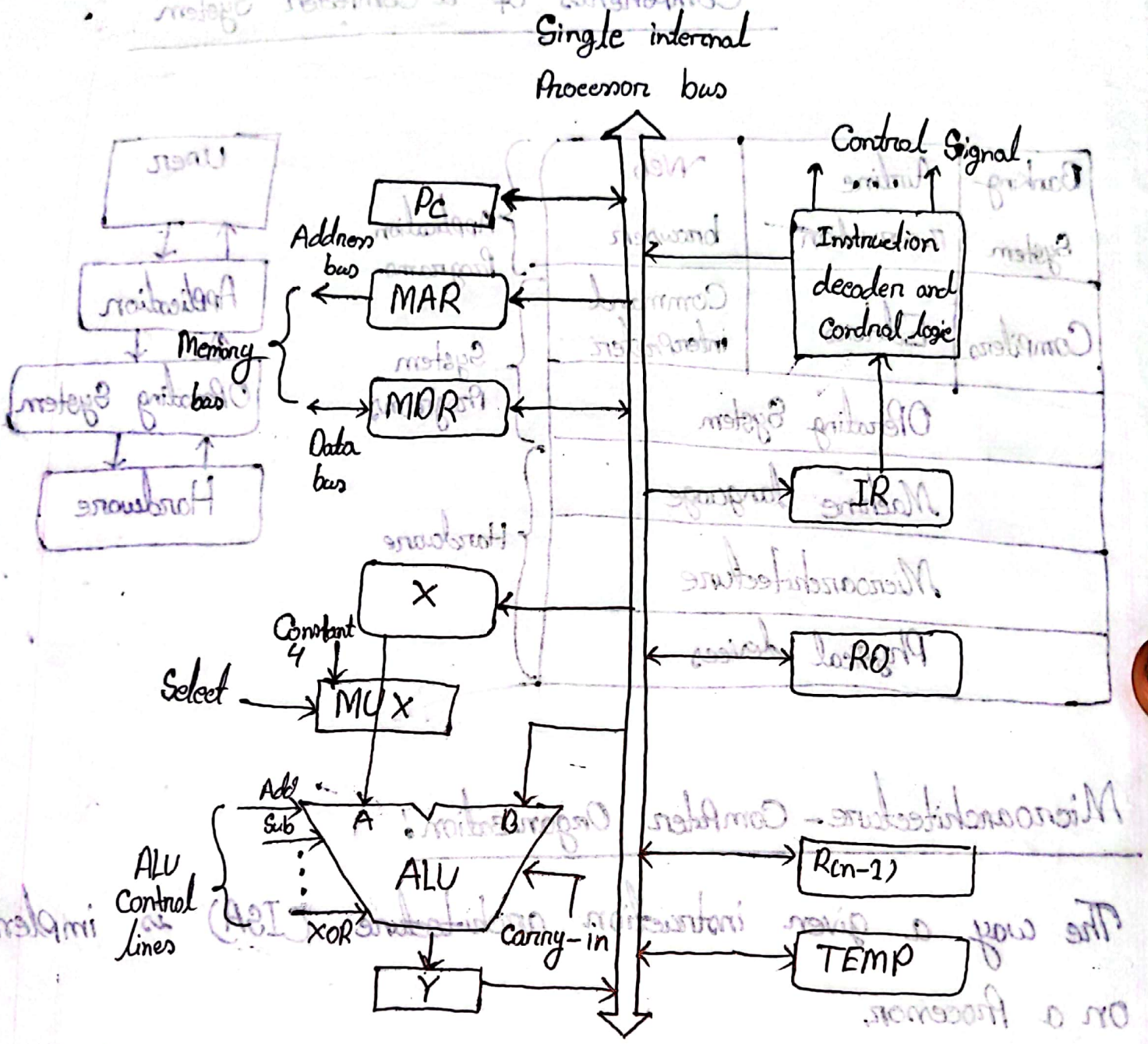
The way a given instruction architecture (ISA) is implemented on a processor.

The ISA includes:

- The execution model,
- Processor registers,
- address and data formats,

The microarchitecture includes:

- The parts of the processor &
- How these interconnect and
- interoperate to implement ISA.

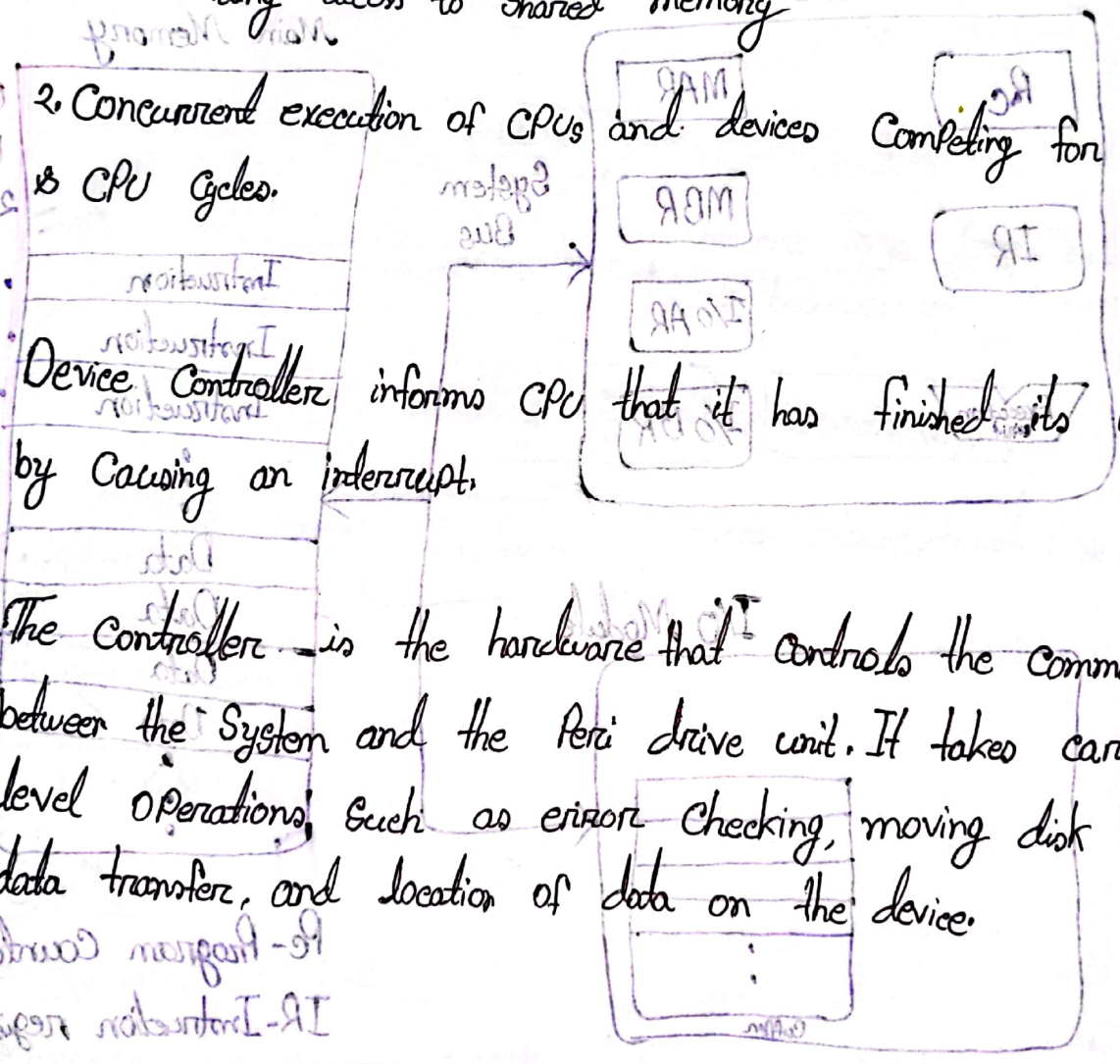


Single Bus Organization

Computer System Operation -

1. One or more CPUs, device controllers connect through common bus providing access to shared memory

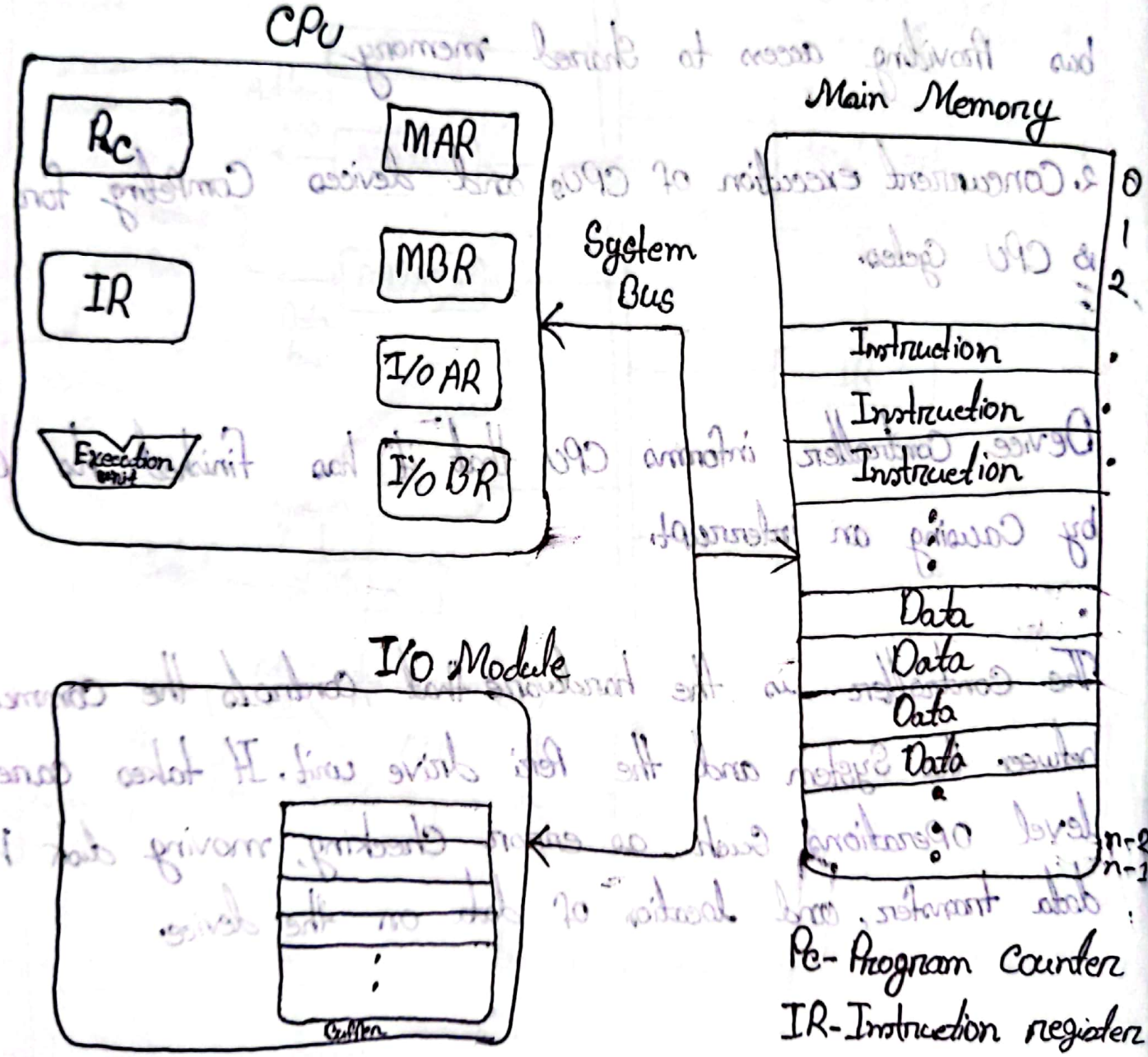
2. Concurrent execution of CPUs and devices competing for memory cycles.



Device Controller informs CPU that it has finished its operation by causing an interrupt.

The controller is the hardware that controls the communication between the system and the peri drive unit. It takes care of low level operations such as error checking, moving disk heads, data transfer, and location of data on the device.

Computer Components: Top level view



PC- Program Counter

IR- Instruction register

MAR- Memory address regis

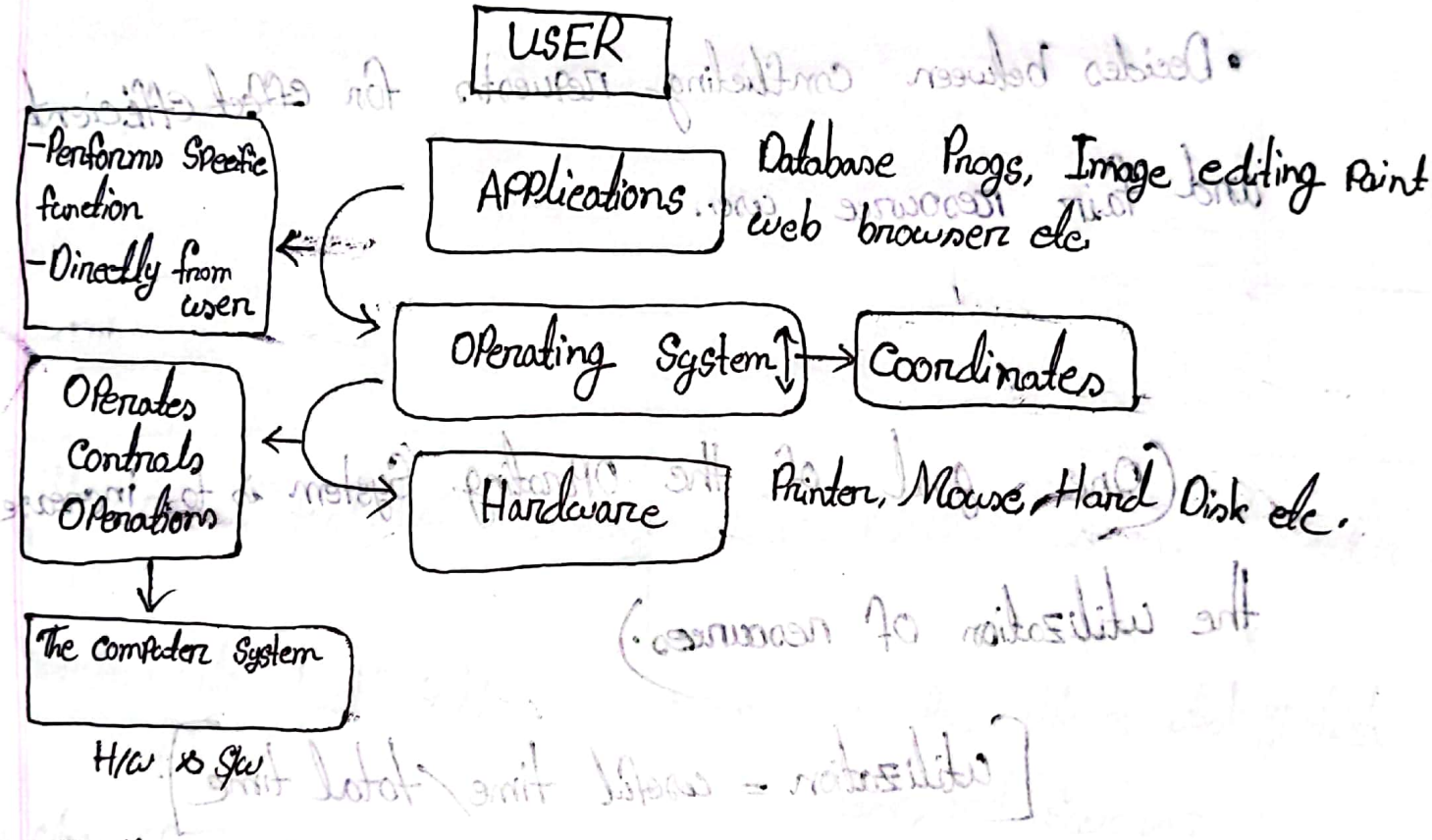
MBR- Memory Buffer addr

I/O AR- Input/output address

I/O BR= Input/output buffer regist

What is an Operating System?

A Program that acts as an intermediary bet^{n} a user & H/w



GOALS

→ Convenient to use

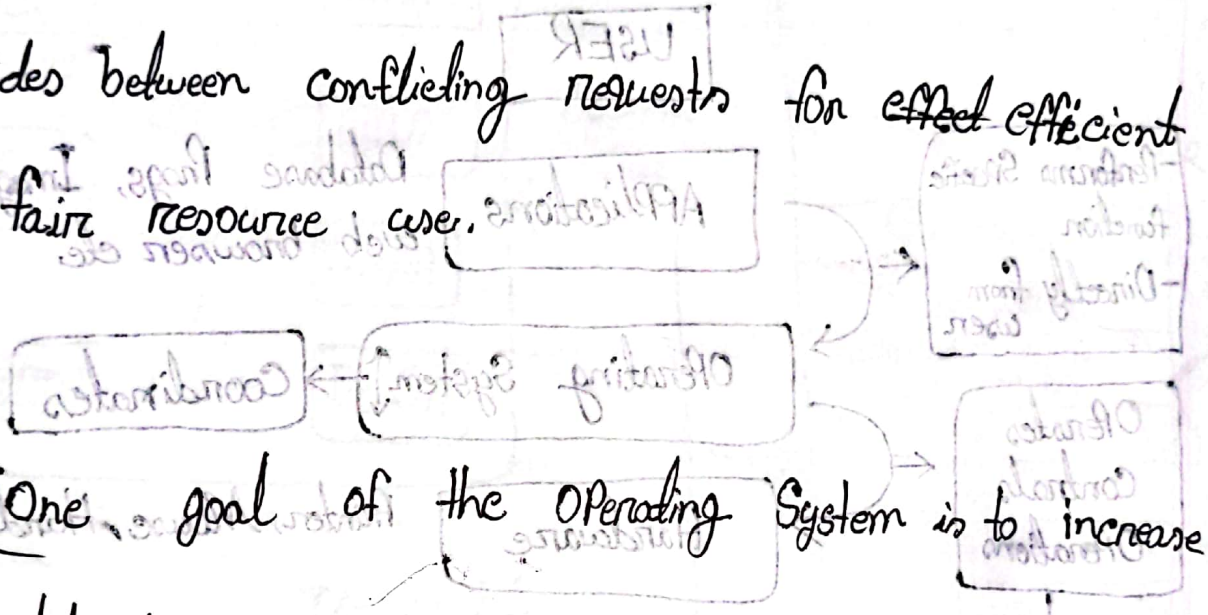
→ Execute user Programs & make Solving user Problems easier

→ use H/W in an efficient manner

Operating System Defination -

OS is a resource allocator

- Manages all resources
- Decides between conflicting requests for effect efficient and fair resource use.



(One goal of the Operating System is to increase the utilization of resources.)

[utilization = useful time / total time]

For example, the OS should avoid wasting CPU time bec the disk is rotating, or wasting switching among tasks waiting for I/O.

OS is a Control Program

- Controls execution of Programs to Prevent errors and improper use of the computer. e.g Memory Protection

An example comparing life with/without OS:

Life with an OS	Life without an OS
file = open ("test.txt", O_WRONLY); Write (file, "test", 4); close (file);	• Blocks, Platter, track, and Sector • Where is this file on disk? Which Platter, track, and Sectors? • code needs to change on a different System.

*Kernel & Shell

Why interrupt

The kernel and the shell

An Operating System has two Parts: Kernel and shell

KERNEL (resides in main memory):

1. Is the heart and Soul of an OS. useful applications and utilities are added over the kernel, then the complete Package becomes an OS.

2. directly controls the computer hardware by performing OS Services.

3. hides the hardware details from the users and application Programs.

Example-

Linux is a kernel as it does not include applications file-System utilities, windowing Systems and graphical desktops

An Operating System

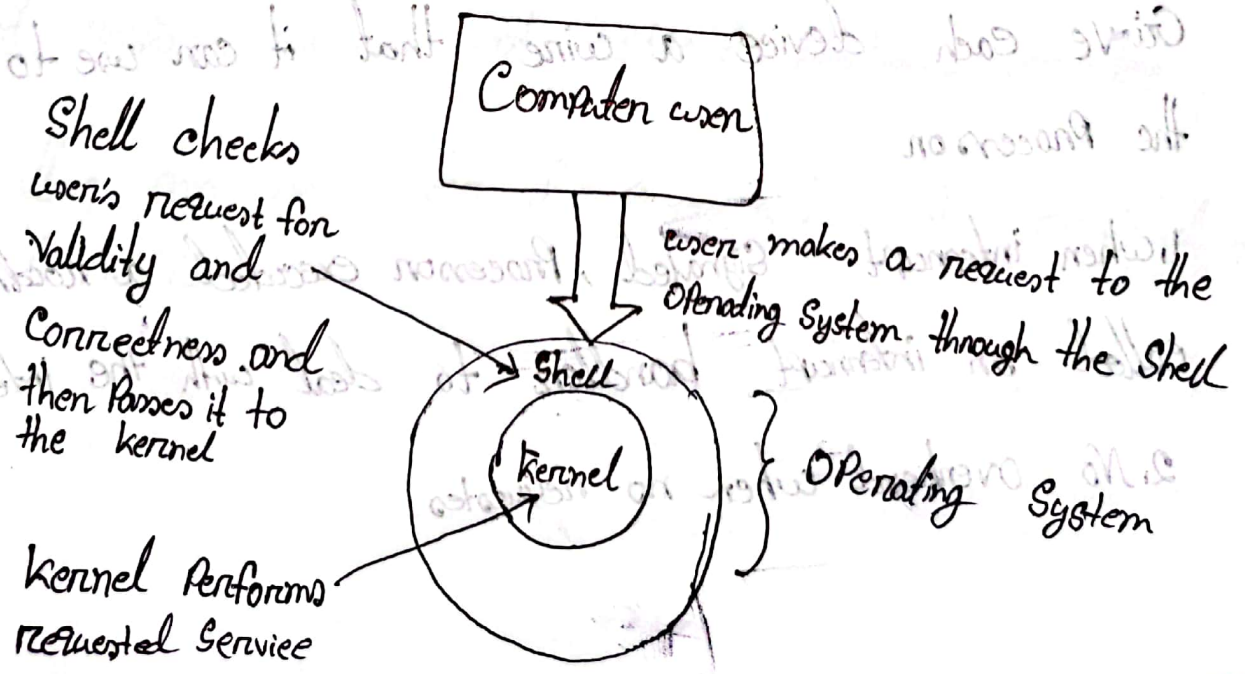
Shell (Command interpreter): Part of OS

1. Serves as the interface between user and kernel.
2. Users enter commands via shell in order to use the servic of kernel.

Types of Shell:

1. Graphical user Interface (GUI)
Window/Mac OS
2. Command line Interface (CLI)
Dos / Linux

How user communicate?



Why interrupt?

People like connecting devices

- A computer is much more than the CPU.
- keyboard, mouse, Screen, disk drives.
- Scanner, Printer, SoundCard, Camera etc.

These devices occasionally need CPU Service

- But we can't Predict when

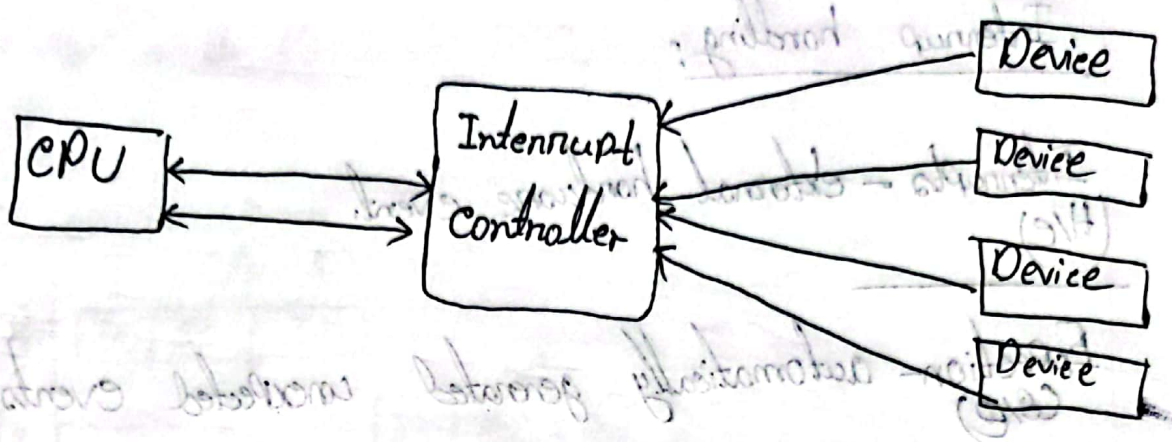
External events typically occur on a macroscopic timesca

- we want to keep the CPU busy between events

Alternative: Interrupt vector

Give each device a wire that it can use to signal the Processor

1. when interrupt Signaled, Processor executed a routine called an interrupt handler to deal with the interrupt
2. No overhead when no requestes



Polling vs Interrupt vector

Polling is like picking up your phone every few seconds to see if you have a call. Interrupts are like waiting for the phone to ring.

1. Interrupts win if Processor has other work to do and event response time is not critical. ~~Polling can be~~

2. Polling can be better if Processor has to respond to an event ASAP.

Interrupt handling:

Interrupts – external hardware event. (H/E)

Exception – automatically generated unexpected events occur in response to some exceptional condition. (S/E)

Trap – 1. a Programmer initiated and expected transfer of control to a special handle routine (S/I)

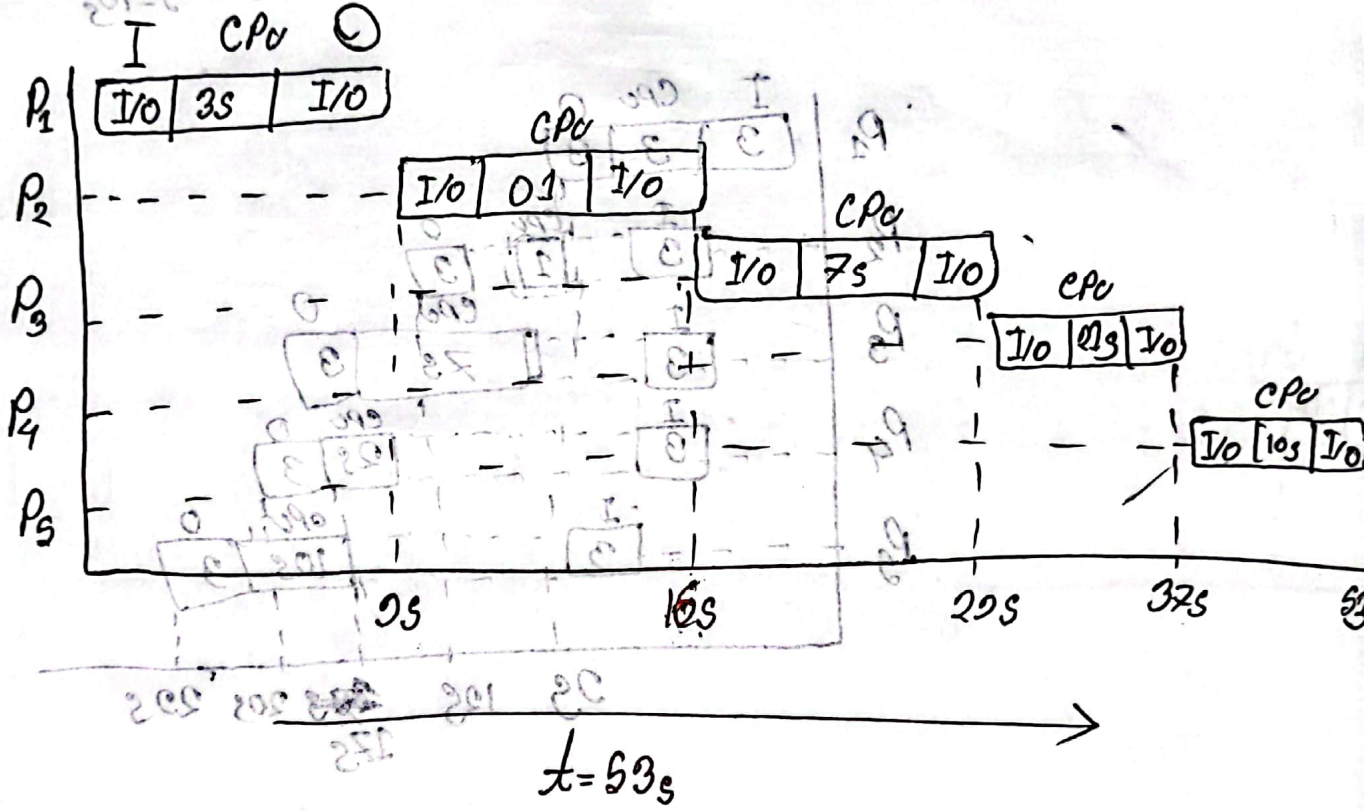
2. User Program can ask for an OS Service.

3. Unconditional – control always transfer to Pre-defined Procedure

System call – Simply sets up some register with the need System call / OS Service number; and execute the trap / Software interrupt. (S/I)

I/O = 3s, $C_1 = 3$, $C_2 = 1$, $C_3 = 7$, $C_4 = 21$, $C_5 = 10$

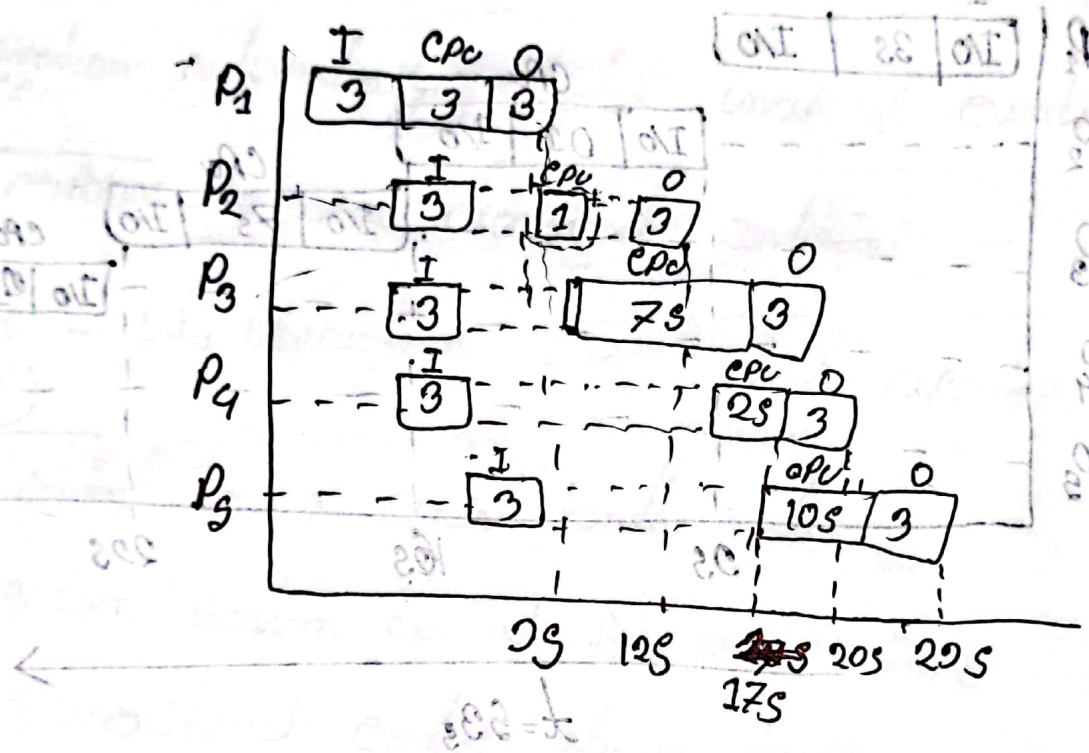
Batch:



CPU Idleness = 18s

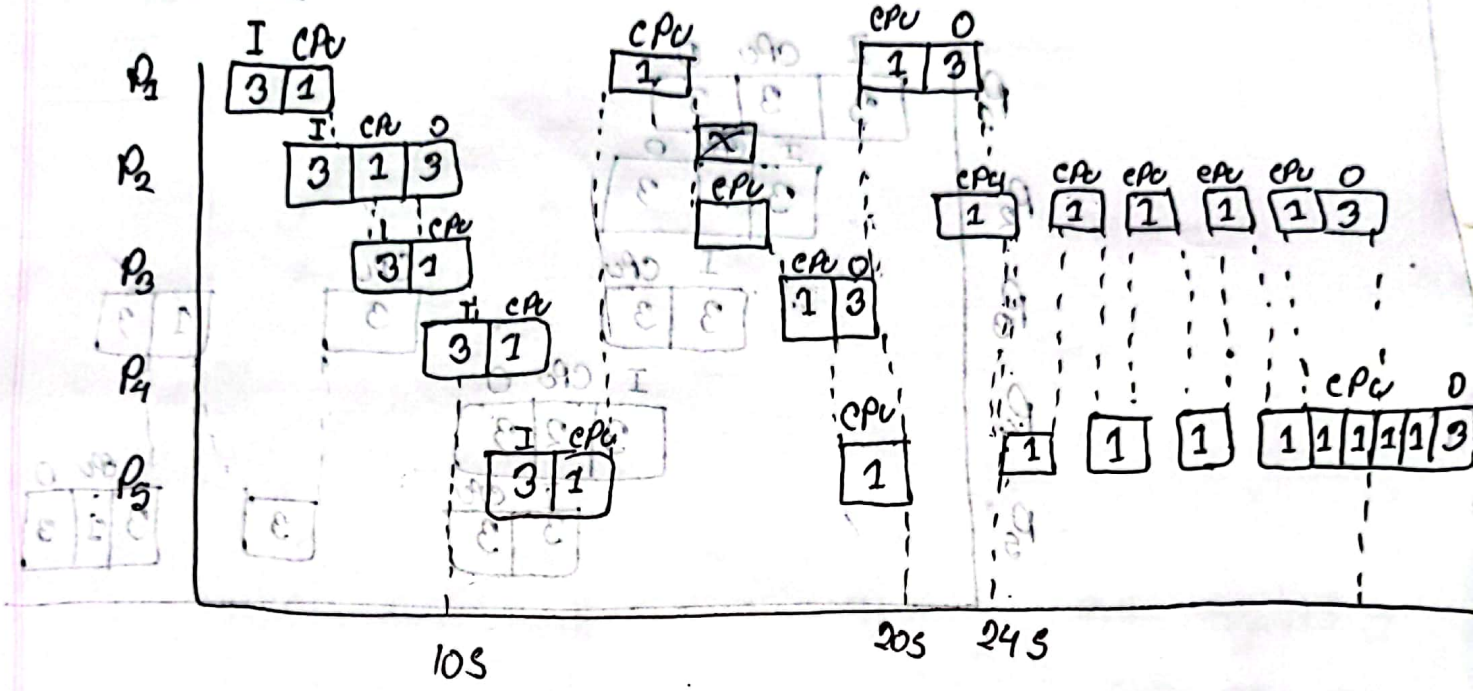
Multiprogrammed:

I/O = 3s, C_1 = 3s, C_2 = 1s, C_3 = 7s, C_4 = 2s, C_5 = 10s



CPU - Idleness = 3s

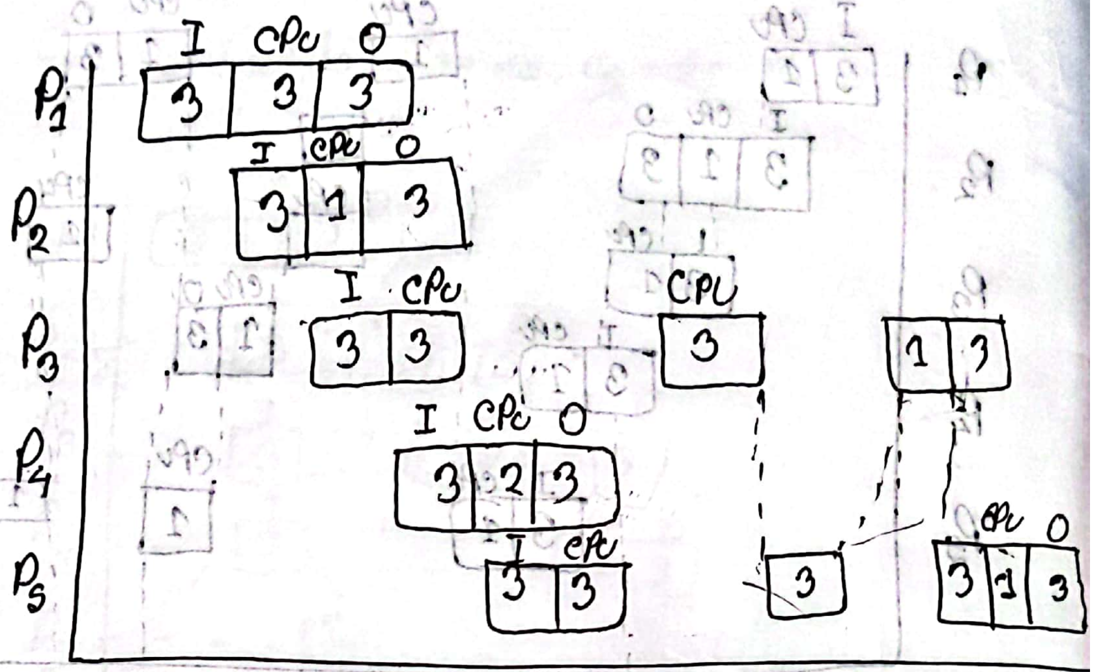
Time Sharing: I/O = 3, $C_1 = 3$, $C_2 = 1$, $C_3 = 7$, $C_4 = 2$, $C_5 = 10$



CPU Idleness = 8s

Process	—	Execution time
P_1	–	24 s
P_2	–	10s
P_3	–	33s
P_4	–	22 s
P_5	–	37 s

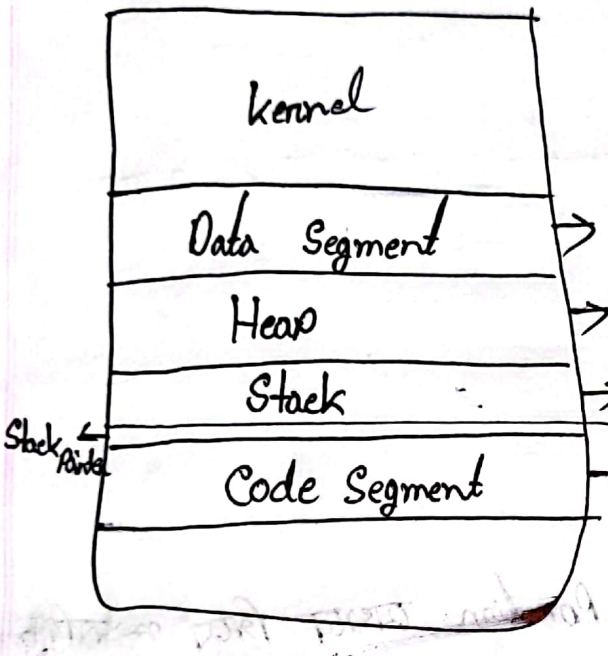
If we do it with '3' See time



না
ধামাধরাদের-
আবার-
দলের- চিন্তা
ভাবে নাকি!

অন্যায়েও
যারা নাটাইয়ে
দেয়- না

Topic: The Process



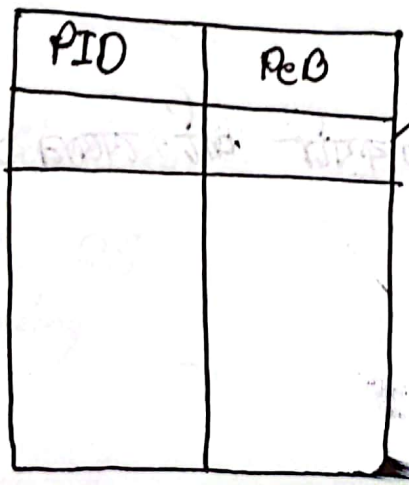
কোড Segment-এ লেখার করা যায় না

যতবার function recursive হবে ততবার Stack Pointer ঘুরবে [Stack Pointer এর মধ্যে বারে

রিকার্সন যদি বেইজ রেজ না পায় Stack Pointer ঘুরতে থাকবে [call করতেই থাকবে]

Process Table:

(যে Process টাকে আমরা Protect করব Kernel Space এ রাখবে)



PCB - Process control Block

CPU — Pc, IR, MBR, Ax

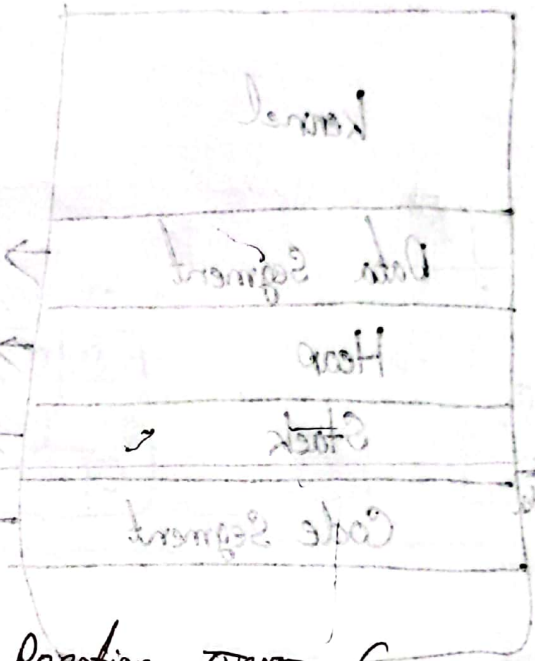
Pc → Instruction address রাখে

IR → Instruction run করে

Ax → যদি কোন যোগ, বিয়োগ, গুণ জাতীয় কাজ করে সেগুলোই রেখে দেয়

CPU স্ক্র্যাচপ্যাড · Address

memory map →



Process control block – memory portion যেখানে গিয়ে সবকিছু store করে রাখে

OS দুভাগে – OS কে ২টা ভাগে ভাগ করে (অর্থাৎ ১ম part যা থাকবে ২য় part তে থাকবে না)

x=10, y, z
y=x
z=x+y

Scheduling – related information

Process control Block এ কয়টা part? যেকোন ২টা explain

Scheduling

Not-Running Process in Queue

*CPU Switch From Process to Process — explain

Process Scheduling- (Device queues কে ~~কাজ~~ waiting Queue বলা হয়)

~~OS~~ PCB মা জানে OS এর

↙

Note Book এর মতো

Priority interrupt

fork a Child

(fork একটি system call বা Instruction

*Ready Queue and various I/O Device Queue

*Representation of Process Scheduling.

***Scheduler (Long term vs Short term)

Short Time Scheduler

→ Time Slice 2S

→ I/O or waiting

→ Interrupt

→ Terminate

যেখানে Multitasking use করে

Multiprogramming - একটি CPU এ একাধিক

Multitasking

Degree of Multiplanning * যদি বাড়ে বা কমে তাহলে কী সমস্যা হতে পারে।

Zombie Process is created when Parent Proc Doesn't wait that of a child Process for termination of that Particular child Proces. Lat After termination Parent Process their child Proces Still Rem in the Compaten System (Occupying) Allocating re resources.

* Process Termination

* Process creation

Leeluen - 3

* * *The Soul of a Process

*Processes and threads

*why used child Process

*Parent Process ও dependency থাকে থাকবে

*Single and Multithread

Memory inefficiency

*Process and Threads

includes threads

CPU cycle

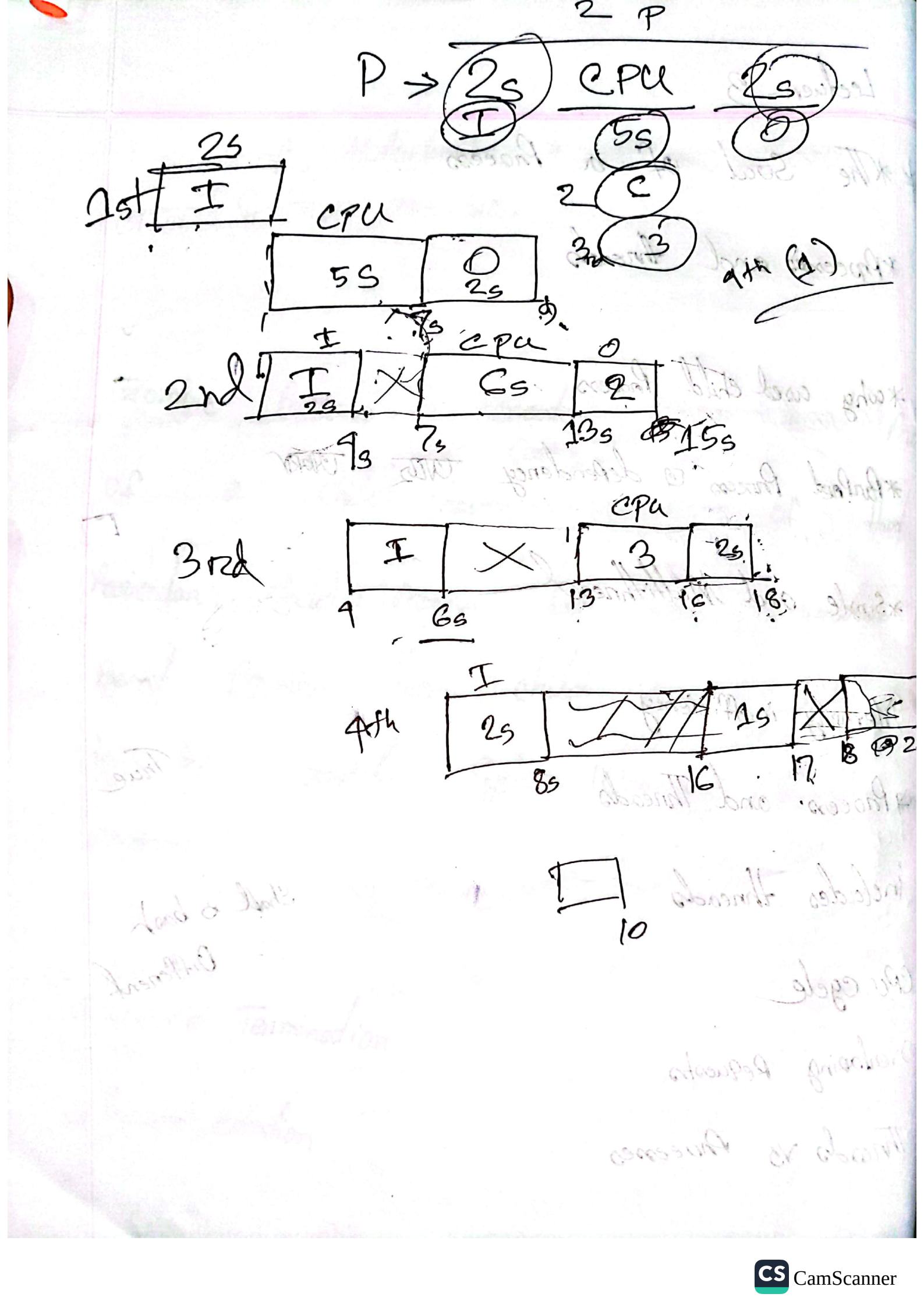
Overlapping Requests

Threads vs Processes

True

Shall ও bool

= Different


2 P
P → 2s I
CPU 5s
2s O
2 C
3 3
1st 2s I
CPU
5s
2s O
9th (1)
2nd I
I 2s
CPU
6s
O 2
9s 7s 13s 15s
3rd
CPU
I
3
2s
9 6s 13 16 18
4th I
2s
1s
8s 16 17 18 2
10

time sharing

cpu work — 2s times — Limit

I/O = 2s

P_1 → CPU = 5s

P_2 → CPU = 3s

P_3 → CPU = 7s