CODE: ....

"OS-2"

S. No......**3877**....

The LNM Institute of Information Technology, Jaipur

(Deemed to be University)

Instruction to Candidate (for examination)

1. Immediately on receipt of the Test Booklet the candidate will fill in the required particulars on the cover page with Ball Point Pen only.

2. Candidates shall maintain perfect silence and attend to their Question Paper only. Any conversation or gesticulation or disturbance in the Examination Room / Hall shall be deemed as misbehaviour. If a candidate is found using unfair means or impersonating, it shall be treated as breach of code of conduct and the matter dealt with accordingly.

3. No candidate, without the special permission of the Invigilator concerned, will leave his/her seat or Examination Room until the full duration of the paper is over. Candidate should not leave the room / hall without handing over their Answer Sheets to the Invigilator on duty.

4. During the examination time, the invigilator will check ID Card of the candidate to satisfy himself / herself about the identity of each candidate. The invigilator will also put his/her signature in the place provided in the Answer Sheet.

5. The Candidate shall fill the number of supplementary sheets attached, on the front page of the main answer sheet.

6. **Bringing cell phones /communication devices in the examination hall is strictly prohibited. Exam conducting authority will not be responsible for the custody of such articles. However, use of scientific calculator is permitted.**

Philosopher's

Name of the student: ...Abhishek. Sharma.

Roll No. : ...Lecture Notes

Name of Examination : ....

Subject : ....Embedded System...

Day & Date : ....

No. of Supplementary Sheets Attached: ....

Student's Signature   Invigilator's Signature

Question No.	Marks Obtained
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
Total Marks	

Atomic Operation

⇒ One that is guaranteed to terminate and is indivisible when applied to either examining a prog. Variable or modifying the state of such program

Keyword:- indivisible and irreducibility.

- The value of variable Count on which task accesses it and in which order.
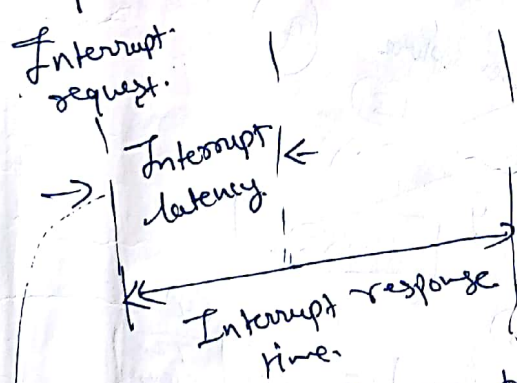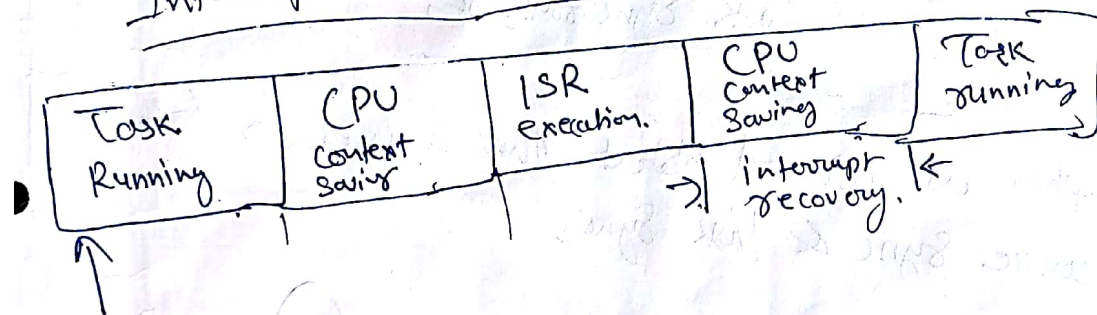- we want to prevent access by all other process.

Mutually exclusive access.

⟹ Sharing of resource or to common data, gives rise to one form of synchronization. That is called mutual exclusion. Synchronization.

⟹ Conditional Synchronization :- process delay or block until a specified condt. is true (or false)

↳ " target is to make sure that two processes are not in their critical sections at the same time.

· Interrupt Service Routine.

Task Running.	CPU Content Saving	ISR execution.	CPU Content Saving	Task running

→| interrupt |←
recovery.

↑

Interrupt request.

→ | Interrupt |← latency.

|← Interrupt response time.

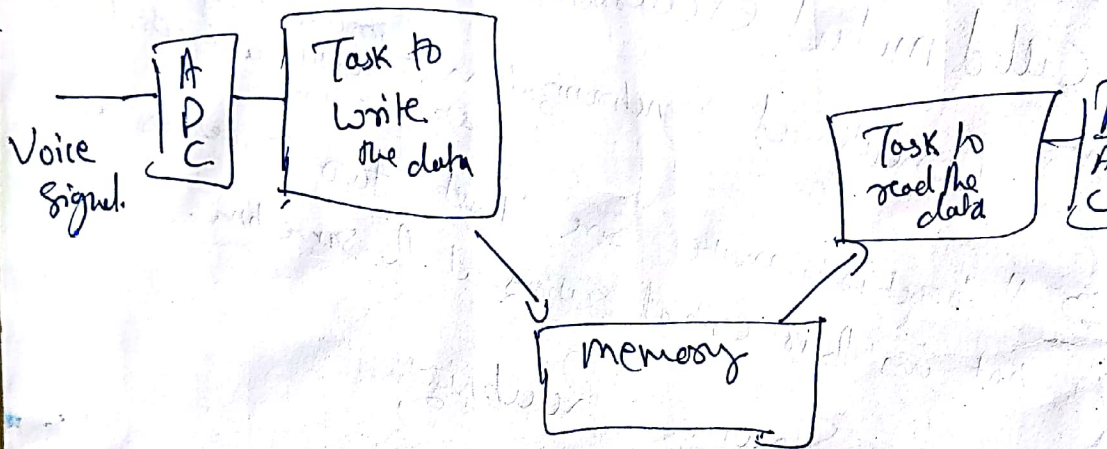IRT → time req for cpu to return to the interrupted code/highest priority task is called IRT recovery.

↳ IL → maxtime for which Interrupt are disabled, + time to start the exec^n of the 1st instruction in ISR

Interrupt response time :- Time b/w receipt of interrupt signal & starting the code that handles the Interrupt.
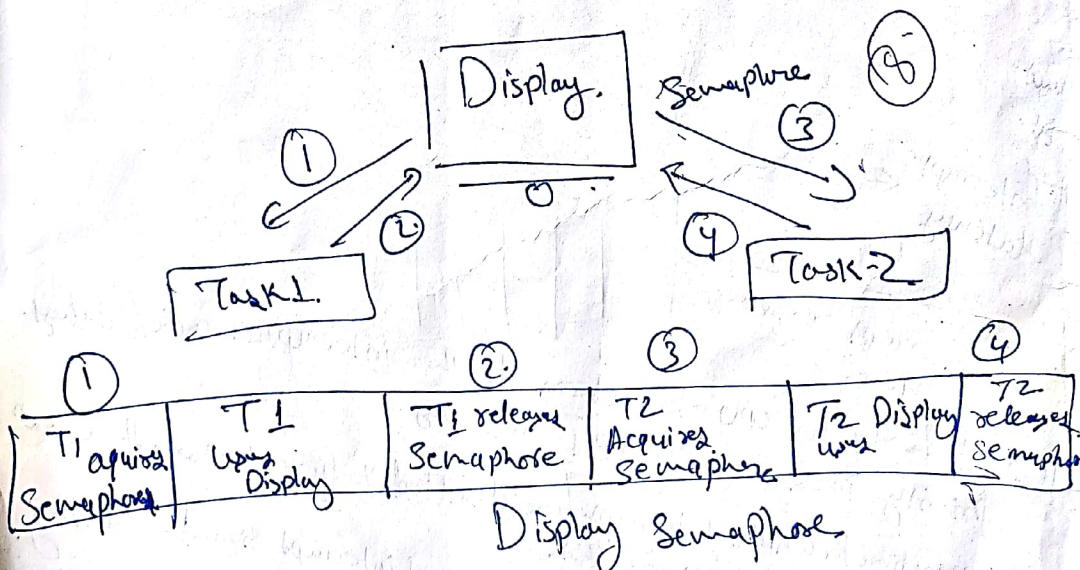
Semaphore

Display.

Task1 → Display ← Task2
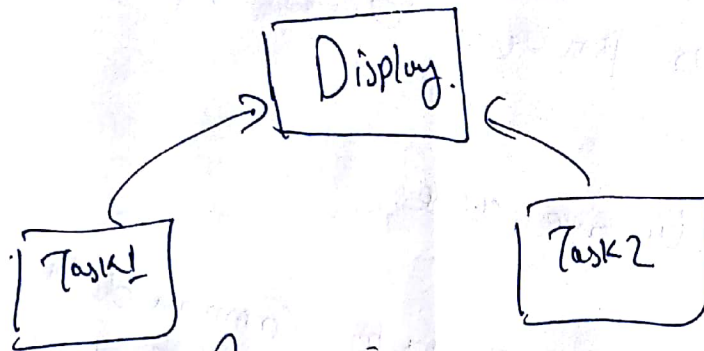
Resource Synchronization.

Voice Signal → [APC] → [Task to write the data] → memory ← [Task to read the data] ← [APC]

Task Synchronization

Semaphore is a Kernel object. that is used for both resource sync & task synch.

Display.   Semaphore (∞)

① Task1 → Display ← ③ Task-2
②   ④

①	②	③	④
T1 aquires Semaphore	T1 uses Display	T1 releases Semaphore	T2 Acquires Semaphore	T2 Display uses	T2 releases Semaphore

Display Semaphore

Counting Semaphore

→ binary Semaphore.
→ counting Semaphore.

Mutex ⟹ mutual exclusion
⟹ resource and task Synchronization.

mechanism:

① Disabling the scheduler :- Scheduler does the task switching
⟶ ⟹

② Disabling the interrupts ⟹
⟹ DI, execute critical section of code. for using shared resource
then EI

③ Test and set Operations:-
ₗₕ check the value of G·V to obtain the status of
Shared resource.
e.g if G·V x is set to ⓪ if resource can be used
1 if can't be used.

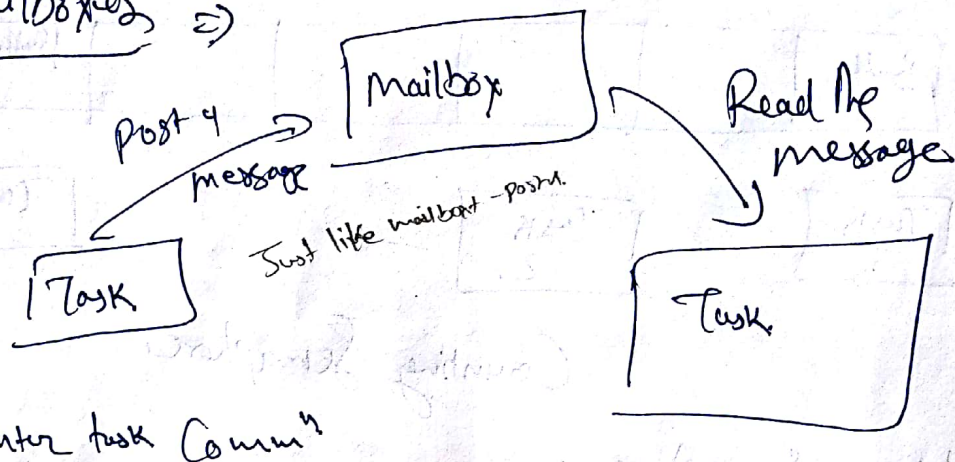Mutex are special binary Semaphore. Could be in clocker unlocked State.
→ It will have another
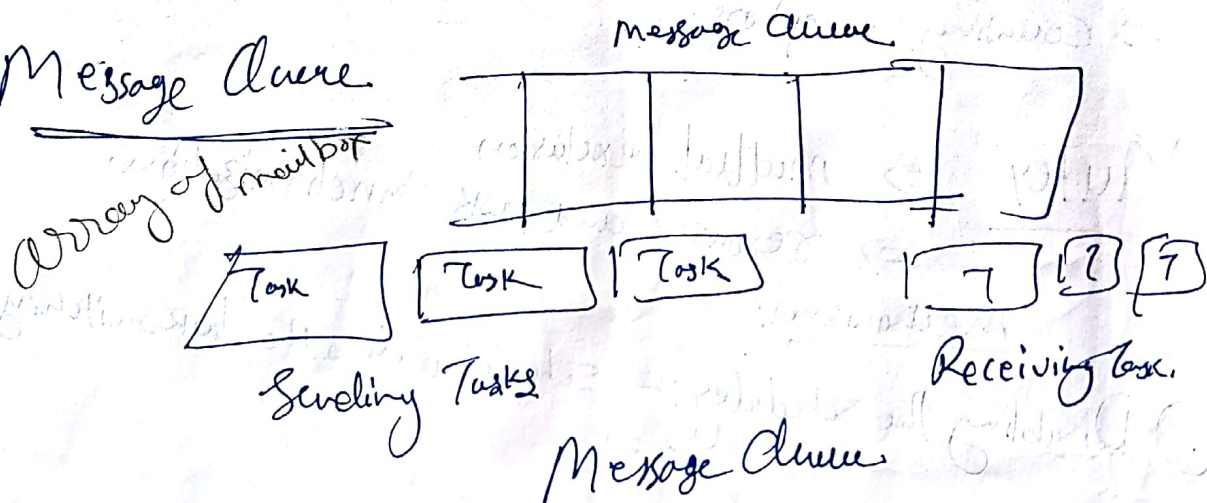→ Owner can aquire a mutex multiple times. in locked state.
→ A task owing. mutex can't be deleted
→ Mutex support Priority inheritance to avoid Priority inversion Prob

Mailboxes =>

Post a message → **Mailbox** → Read the message

Just like mailbox - postal.

Task → → Task

" inter task Comm"

Message Queue

array of mailbox

Message Queue

Sending Tasks → Task | Task | Task ... → 7 | 3 | 7 → Receiving task.

Message Queue

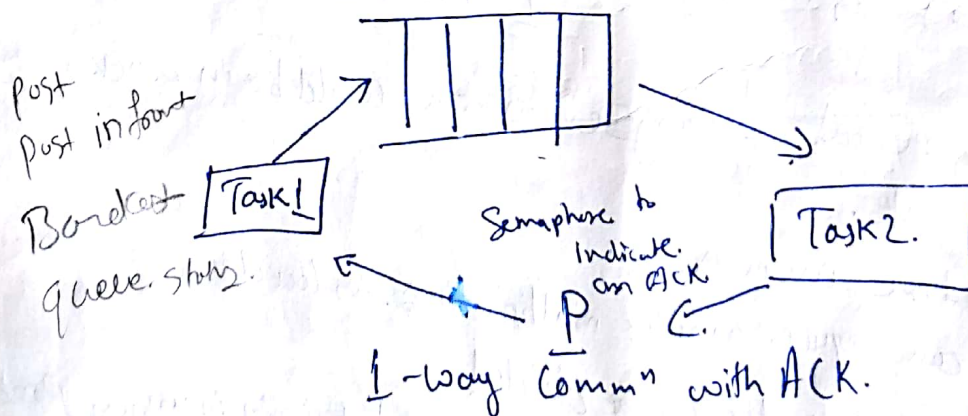Can be Considered as an array of mailboxes.

Task of ISR deposit msg in msg queue.

App^n :- Taking the I/P from keyboard

To display o/P.

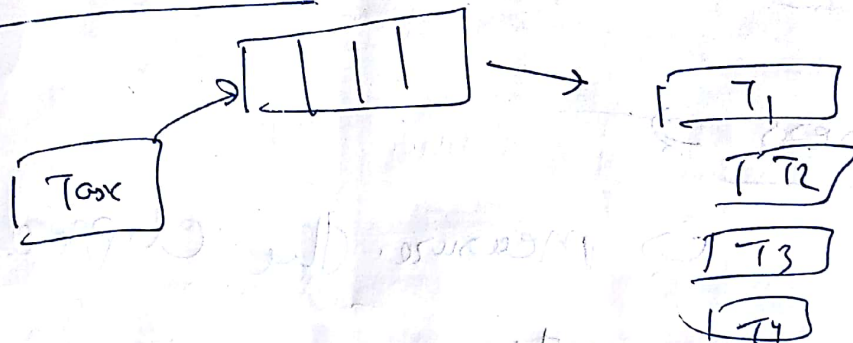Reading voltages from Sensors & transducers

Data packet transmission in a n/w ⊙

Post
Post in font

Broadcast
queue status.

Task1 → [queue] → Task2.

Semaphore to Indicate.
P an ACK

1-way Comm^n with ACK.

2 queue.

Task 1 · T2.

2 way Comm^y

Broadcast Comm^y

Task

T1
T2
T3
T4

Event Register's

1 2 3 4 | 5 6 7 8 9 10 | 12 | 13 | 16

Task #2

Task #5 · Task 4 · Task #2

Task #2
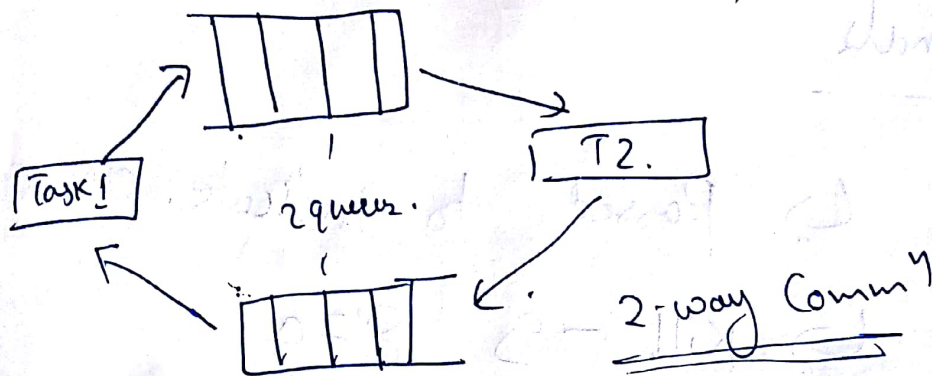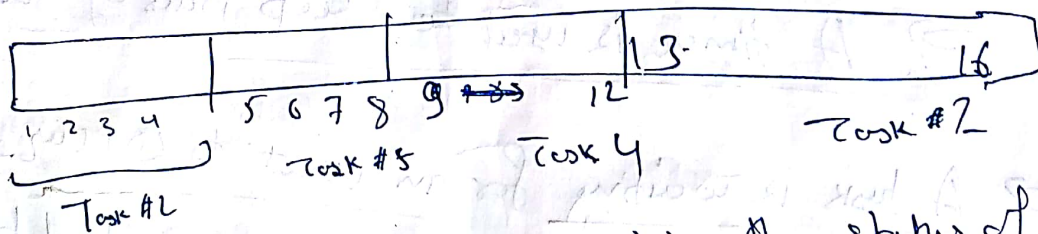
each bit can be obtain the status of.
an event.
Task can have event Reg & set/clear the bit

Pipe.

Task.
writes data to pipe

Task 2 Task Comm.

Task Ready data from pipe

Task 2

Task 2
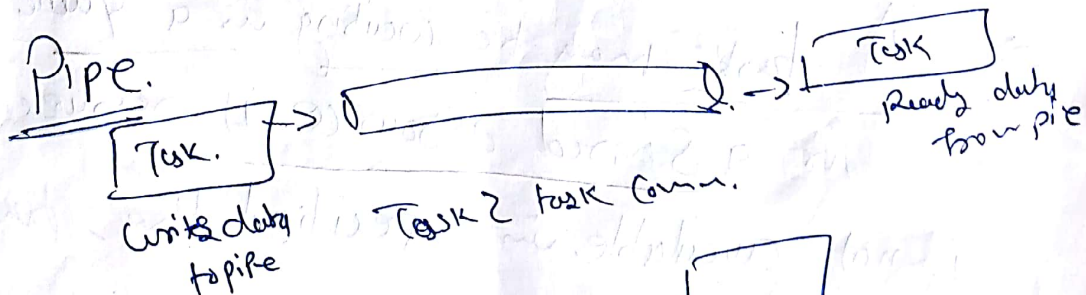
Signals

↳ Passed to indicate an event

↳ Kill -9 879
   Signal no.   → process ID

Timers →

↳ measure the elapsed time of an event.

In Kernel (keep tracks of different times)

→ A timer is used to keep track of Periodicity.

→ A task is waiting for an event to occur. if event does not occur, for specified time it has to take appropriate action.
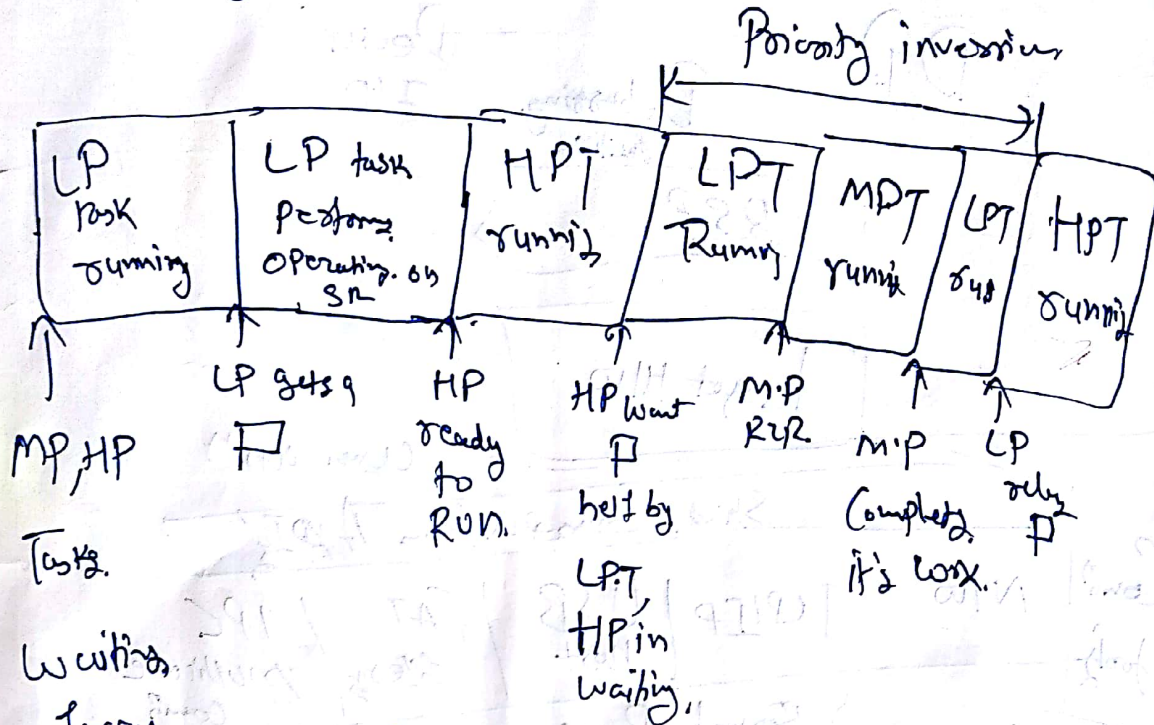
→ A task may be waiting in a queue for a shared resource. if resource is not available in specified time, take appropriate action.
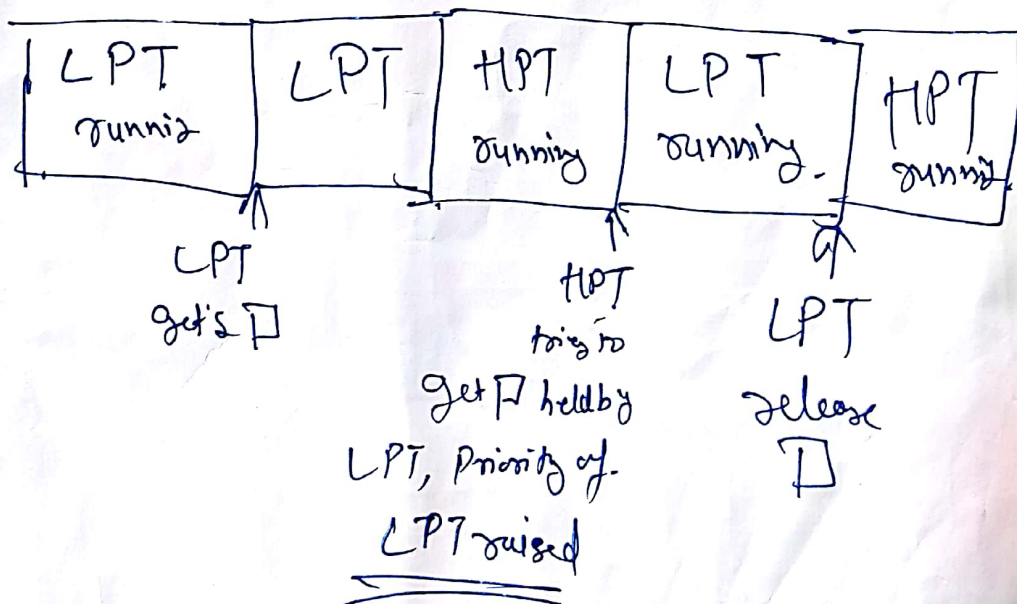
Memory. ③

Priority. Inversion. Problem

Priority inversion

LP task running	LP task performs Operating. on SR	HPT running	LPT Running	MDT running	LPT Sus	HPT running

MP,HP
Tasks.

LP gets P

LP gets q

HP ready to Run.

HP want P held by LPT, HP in waiting,

M·P R2R

M·P Completes it's work.

LP only P

Waiting for an event to occur.

Priority inversion coz priorites of LPT & HPT are inverted.

Priority ⊄ Inheritance

LPT running	LPT	HPT running	LPT running.	HPT running

LPT get's P

HPT trig to get P heldby LPT, Priority of. LPT raised

LPT release P

N/w protocol.   File System   Other Components

C/C++ libraries   Kernel

DD   Debugging facilities   Device I/O

BSP

Target H/W

User app^n

Stand., Cross Platform API's

| Comm^n tools | N/w | TCP/IP | USB host | FAT file sys | IPC multicore comm^n |

Kernel BIOS

Debug & Instrumentation.

EMAC, USB, SD/MMC, UART, I²C SPI

H/W

Bit Rate

(1)
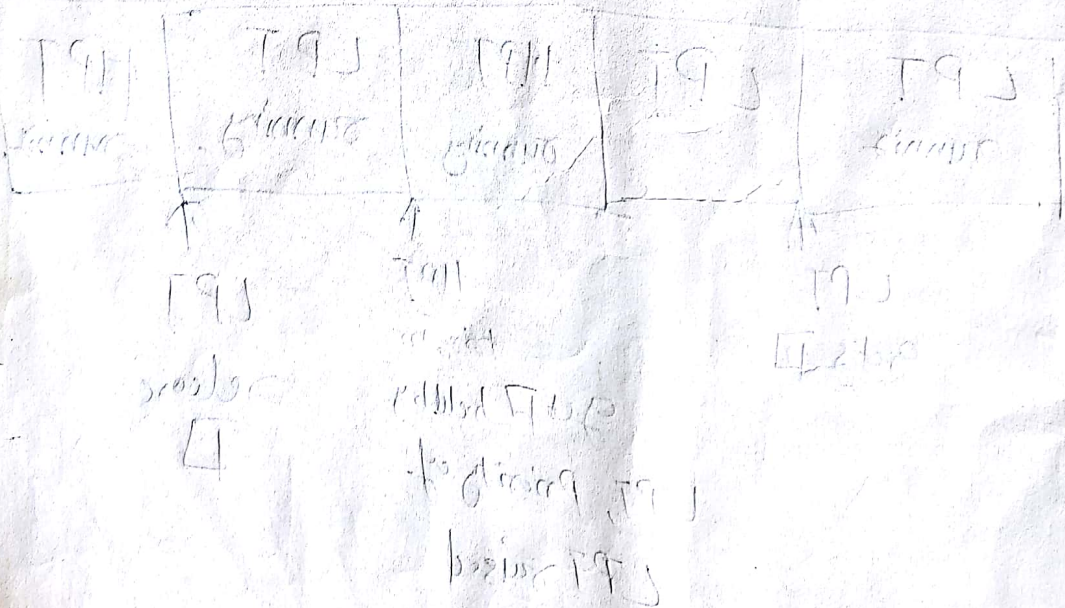
Seep of data. in bPS.

Baud Rate ⇒ french engineer. Emile Baudot

5 - bit teletype. Code.

Baud rate Afer's to the no. of Signal or

Symbol Changes. that occur per Second.

e-g 2400 Baud rate. means.

That the channel. can change state 2400 'ing
per second.

Non-preemptive. Once a processor A. execute till termination

Preemptive. — Process. Can be interrupted by another process. in mid of its execution

This is what happened !!

Priority (H) > Priority (M) > Priority (L)

② H can't run since it is waiting for L to relinquish the lock X, it sleeps

④ M finishes and L runs (H can't run since L has the lock X)

① L runs and aquires the lock X

③ M pre-empts L (since it is not waiting for lock X) and has higher priority than L

⑤ L finishes (releases lock X) and now H can acquire it