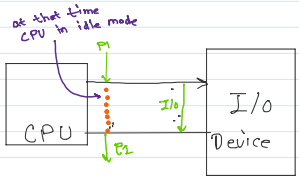
EOS Day -2

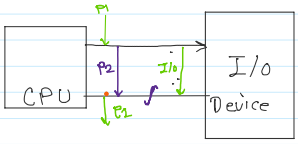
16 November 2023 19:06

* Syn IO



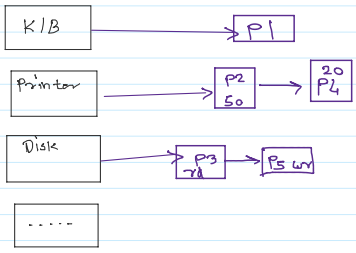
Sync I/o uses polling Technique.

* Asynch I/O



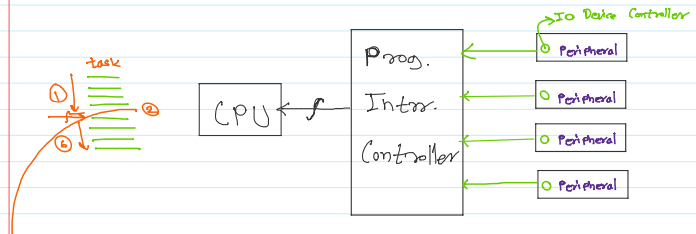
Async I/o using Interrupt Tech.

* Concept of device table



Device Status Table

* Interrupt Processing



Interrupt_Handler (3)

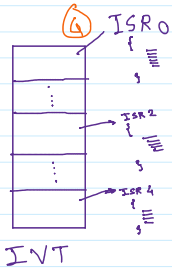
① Save cmc Context

② Get info about intr. from intr. Controller

③ Invoke ISR of Corresponding Intr

④ Restore Execution Context (in CPU) from Stack (5)

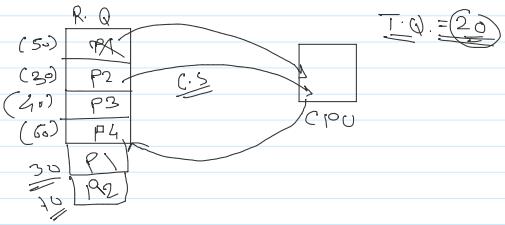
⑤ return from intr



* In 8086 & ARM Cortex A arch, intr handler internally invokes ISR

* In 8051, AVR, Cortex-M. arch intr Handler & ISR are same.

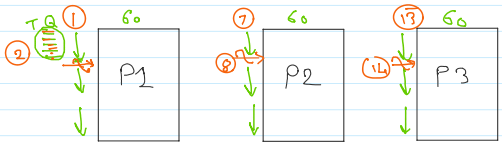
* Round Robin

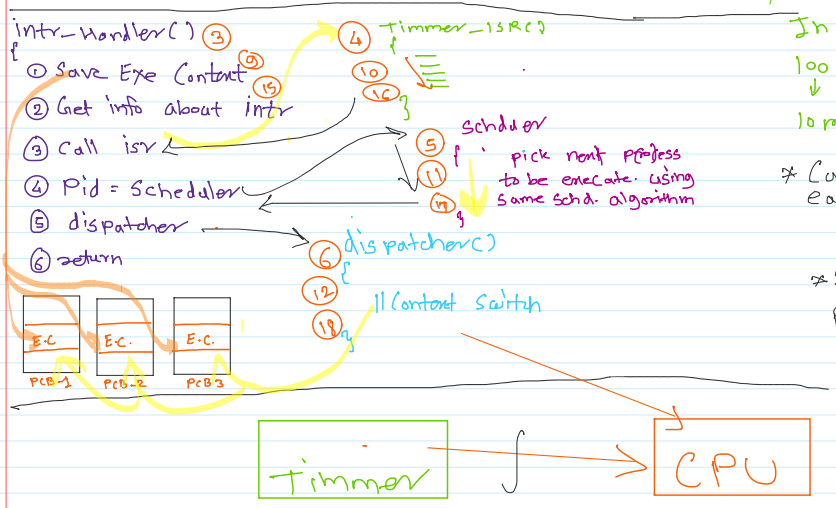


T.Q. = 20

T.Q = 20

RR





* Timmer HW used to generate intr periodicaly
It is refr as System timer or PIT (on x86 arch = 8253)
↑
Timer IC

* Usually timmer intr generated period is 1ms, 1ms or 100ms
In Linux Customization Config_HZ can be set to
100 , 250, or 1000 → (Num of intr/sec)
↓ ↓ ↓
10 ms 25ms 1ms

* Current process remaing time can be done in each tick intr by timmer ISR

```
Current_PCB -> time_left--;
```

* Sched ji check it Time Quatum of Current Process is Completed, if not it return same process id, otherwise it next Process.

```
If (PCB -> time_left > 0)
    return PCB -> Pid;
```

Context Switching

* one process goes off the CPU and next process comes on CPU.