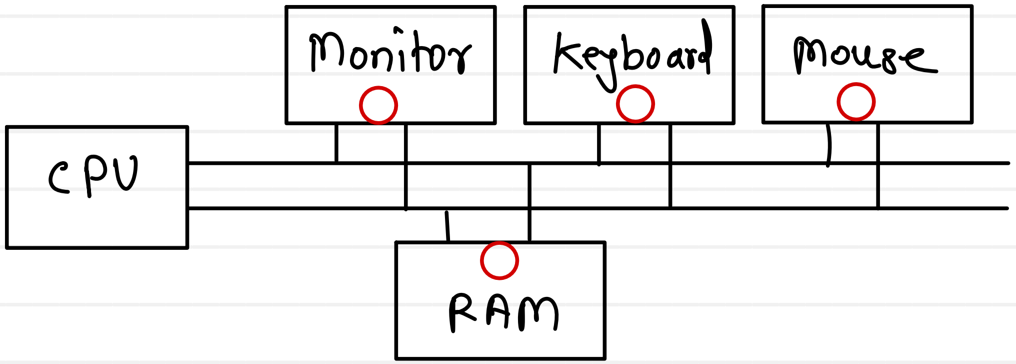
Sunbeam Institute of Information Technology
Pune and Karad

Module - Embedded Operating System

Trainer - Devendra Dhande

Email – devendra.dhande@sunbeaminfo.com

Types of I/O

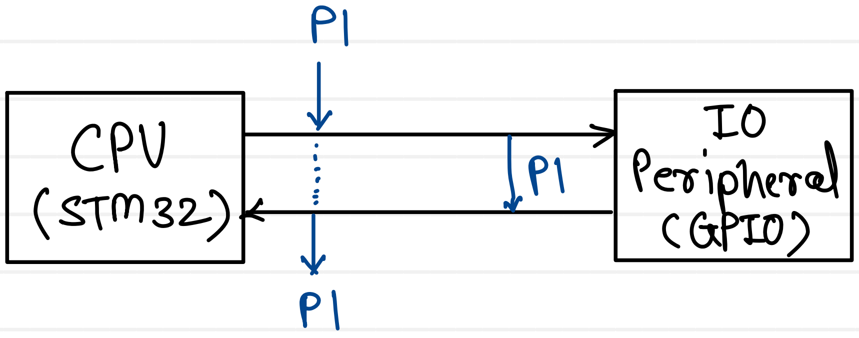


IO Device table

Devices	Status (Idle/Busy)	waiting queue
Mouse	Idle	-----
Keyboard	Busy	....

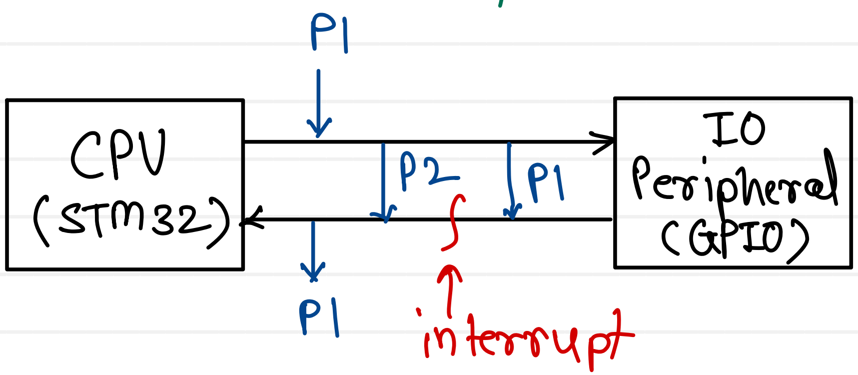
Synchronous (single task)

H/w technique : Polling

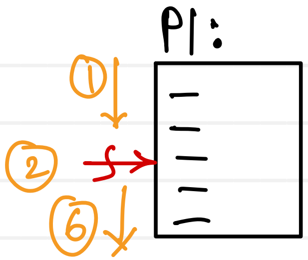


Asynchronous (multi-tasks)

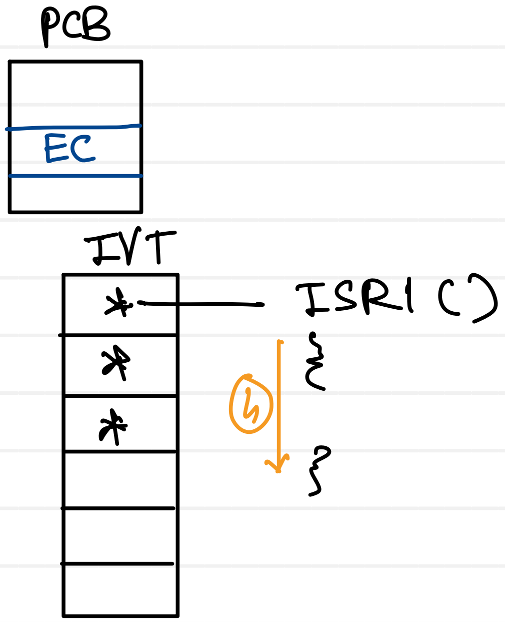
H/w technique : Interrupt

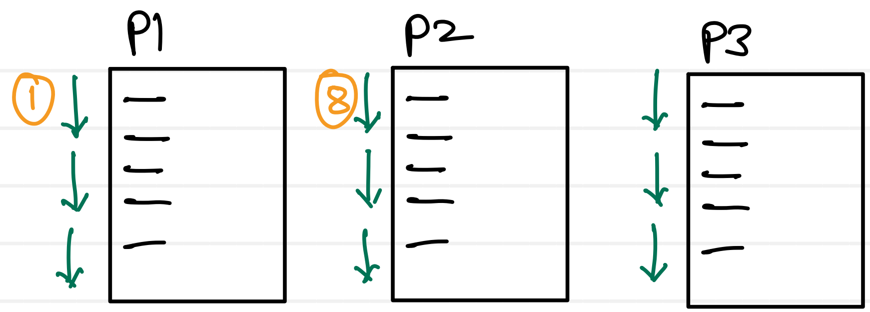


Interrupt handling




```
interrupt_handler( ) {
    1) Save execution context of current
       running process into its PCB
    2) find address of ISR from IVT
    3) call ISR
    4) restore execution context of paused
       process from its PCB
}
```



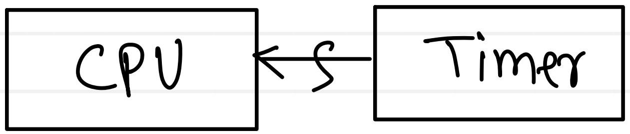



```
interrupt_handler( ) {
    1> Save execution context of running
       process in its PCB.
    2> Find address of ISR from IVT
    3> call ISR
    4> pid = CPU_Scheduler( )
    5> CPU_Dispatcher(pid)
}
```

2 4 6

```
ISR( )
{
}
```

3



- timer peorid can be configured at the time kernel customization into kernel configuration parameter

CONFIG_HZ = 100/250/300/1000

1000 interrupts/sec

```
int CPU_Scheduler( ) {
    if Left time > 0 ;
        select same process to
        execute on CPU
    else
        select next process to
        execute on CPU
    return pid;
}
```

5

```
void CPU_Dispatcher(pid) {
    restore execution context
    of selected process on CPU
}
```

7

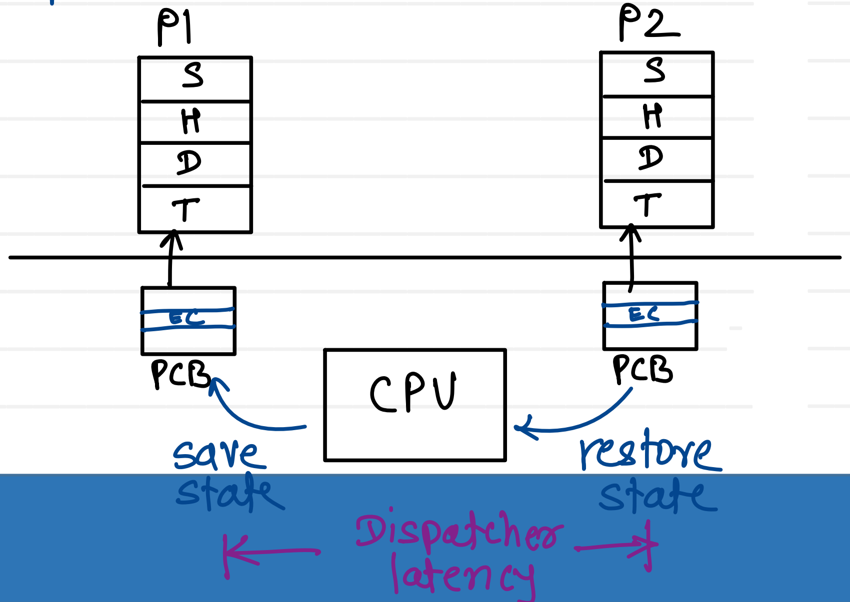
Context switching

Execution context :

- values of CPU registers
- Execution context is stored into PCB of each process

Context switching :

- changing the process of CPU
- save execution context of current running process & restore execution context of next process, is called as context switching.
- CPU Dispatcher does the context switching



- time required to change the process of CPU is called as kernel response time

- Dispatcher latency :
  time from save state to restore state.

System call

Program :

—
—
exit(5)
—
—

user mode

library functions

_exit(5) {
  mov r0, #5
  mov r7, #1
  swi
}

system call API

library (libc)

swi_handler( ) {
1) save execution context of running process.
2) get system call number from r7 register & find address of actual system call from system call table.
3) call system call implementation
4) restore execution context of running process
}

Kernel node

system call table

sys_exit()
{
}

system call implementation

- system mode is always changed on interupts

User mode —interrupt→ kernel mode

ARM : 5 mode bits
7 modes
6 – priviledged
1 – un privi ledged

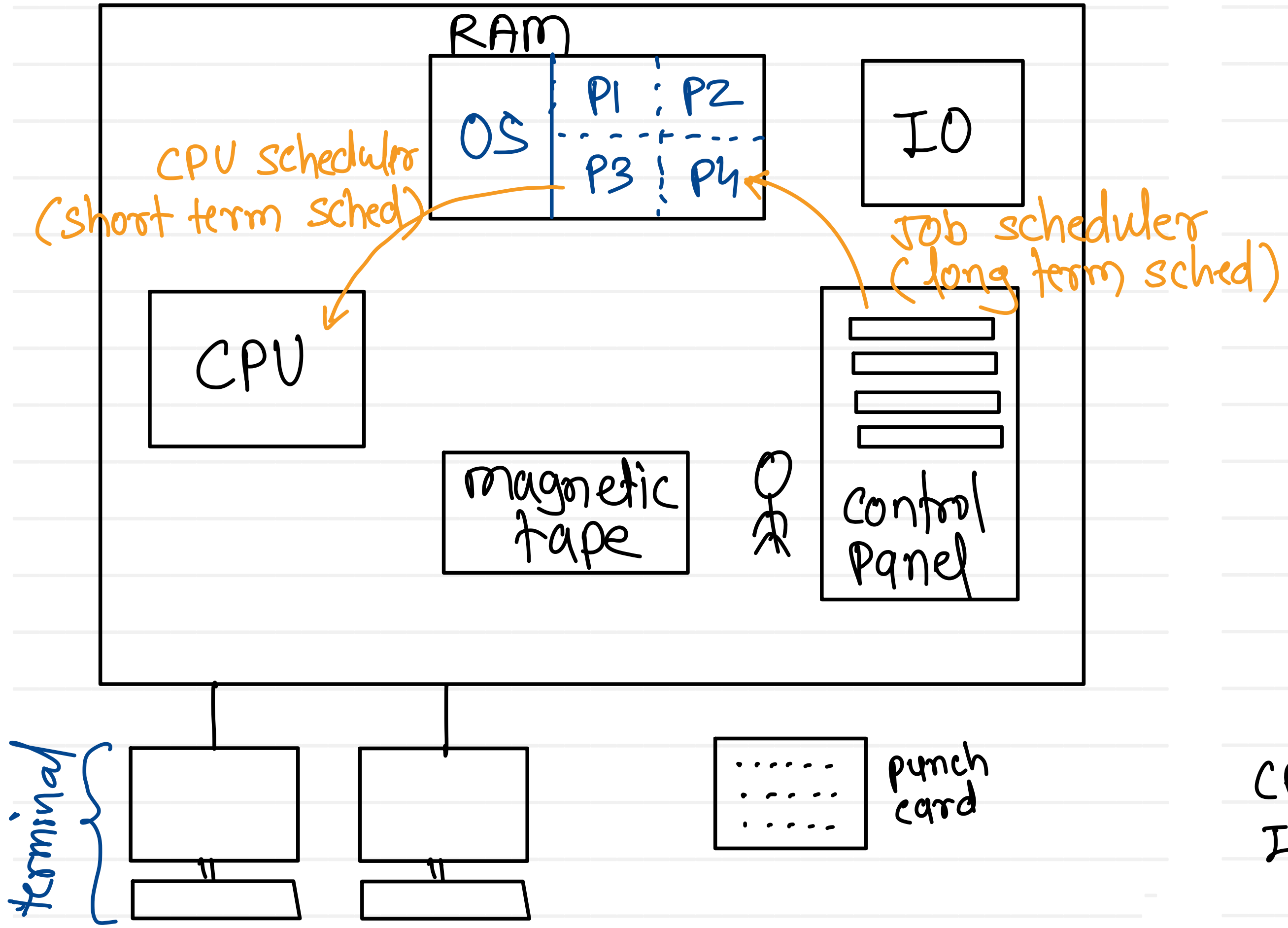
x86 : 2 mode bits
00 – ring 0 more
01 – ring 1
10 – ring 2 less
11 – ring 3 – unpriviledged

priviledged

CPU

mode
0 kernel/system/monitor mode (privileged) – all instructions are allowed can access all CPU registers
1 user mode (unpriviledged)

Types of Operating system



1) Resident monitor
2) Batch system
3) Multiprogramming system:
 - multiple programs are loaded into memory.
 Degree of multiprogramming
 - no. of programs loaded inside memory.

CPU burst : time spend by process on CPU
IO burst : time spend by process on IO

CPU burst > IO burst : CPU bound process
IO burst > CPU burst : IO bound process

- mixture of CPU bound & IO bound processes is loaded inside RAM

Types of Operating system

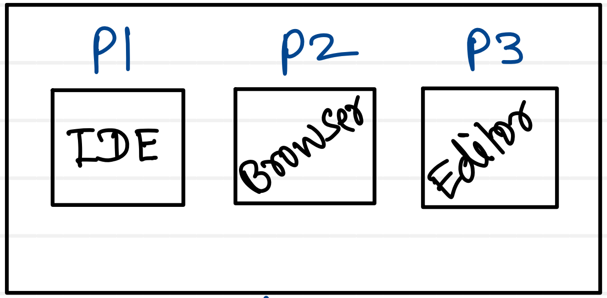
4) Time sharing system: (Multitasking)

- CPU time ~~is~~ shared in all the processes of memor

response time < 1 sec

- there are two types of multitaskings

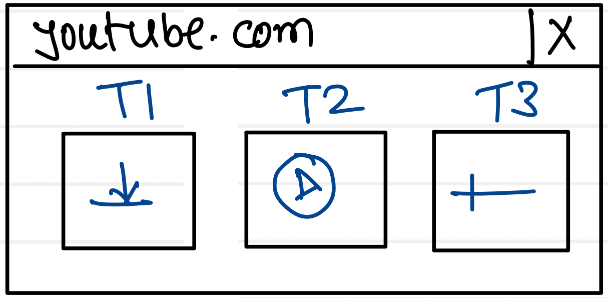
1. Process based Multitasking



System

- within system multitasking

2) thread based multitasking (multithreading)



Browers

- within process multitasking

Types of Operating system

5> Multiuser system :

- multiple terminals are connected to system

(Keybord + Monitor)

6> Multiprocessing system

- multiple CPUs are putted together in single chip. such chips are called as multiprocessor / multicore

- OS can schedule multiple processes for multiple cores, means multiple instructions will be processed parallel

1> Symmetric multiprocessing

2> Asymmetric multiprocessing

SUNBEAM

Thank you!!!

Devendra Dhande

devendra.dhande@sunbeaminfo.com