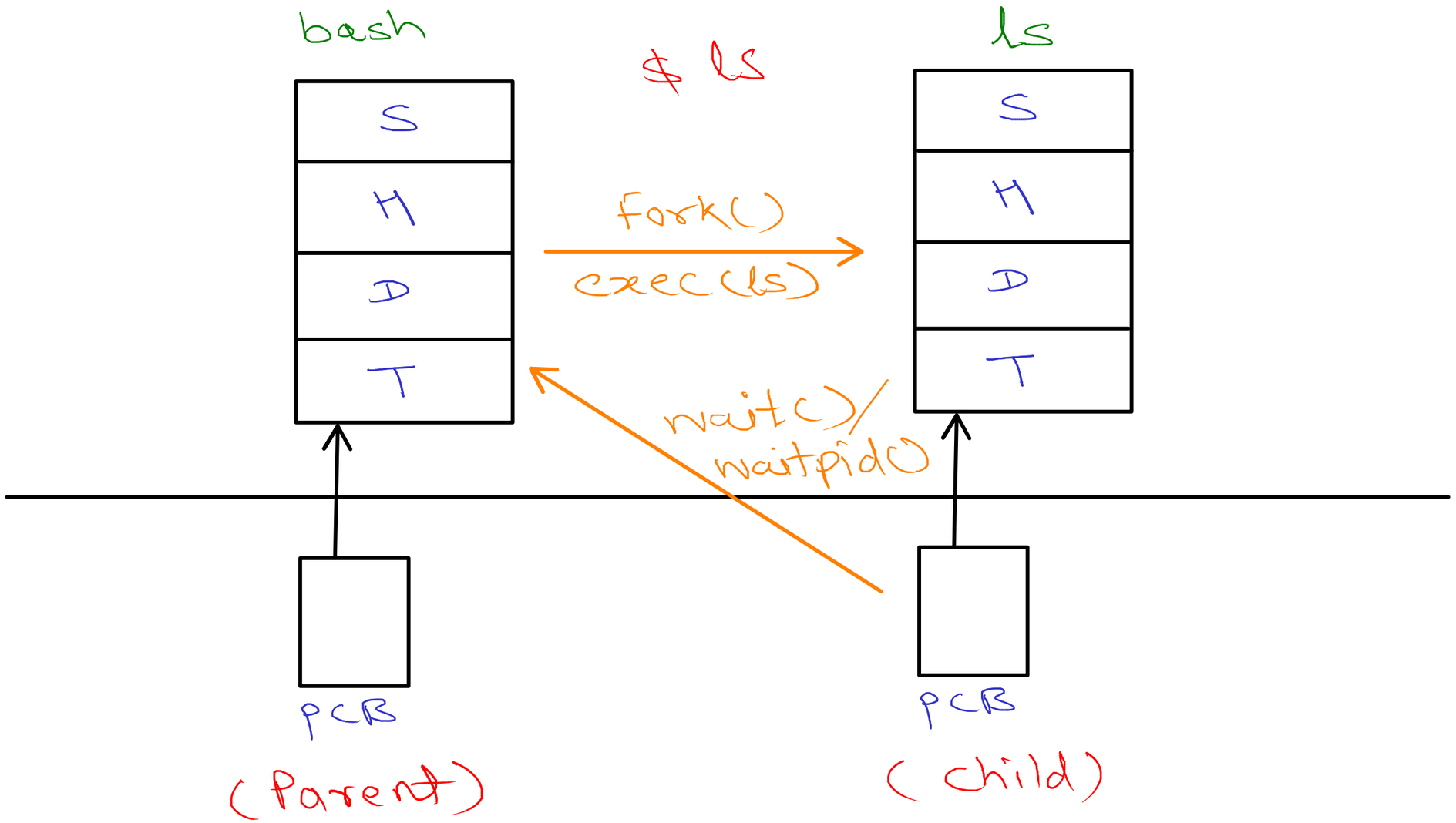
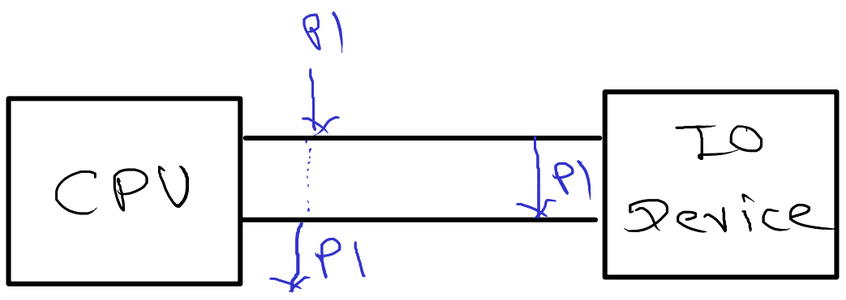

bash
$ ls
ls
S
H
D
T
S
H
D
T
fork()
exec(ls)
wait()/
waitpid()
PCB
PCB
(Parent)
(child)

IO Techniques

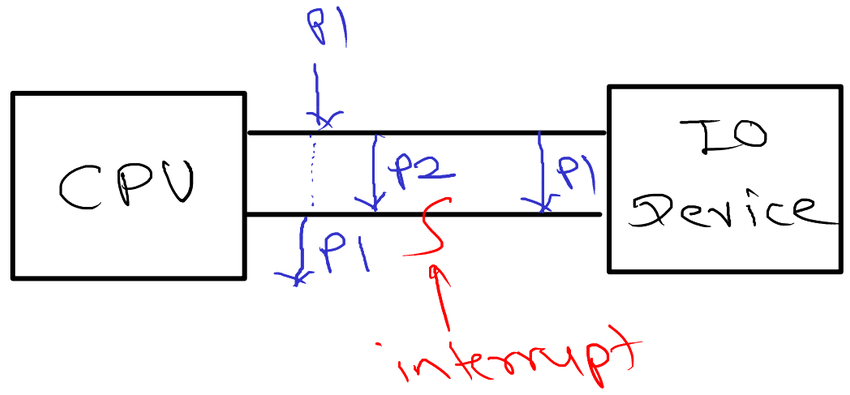
Synchronous IO

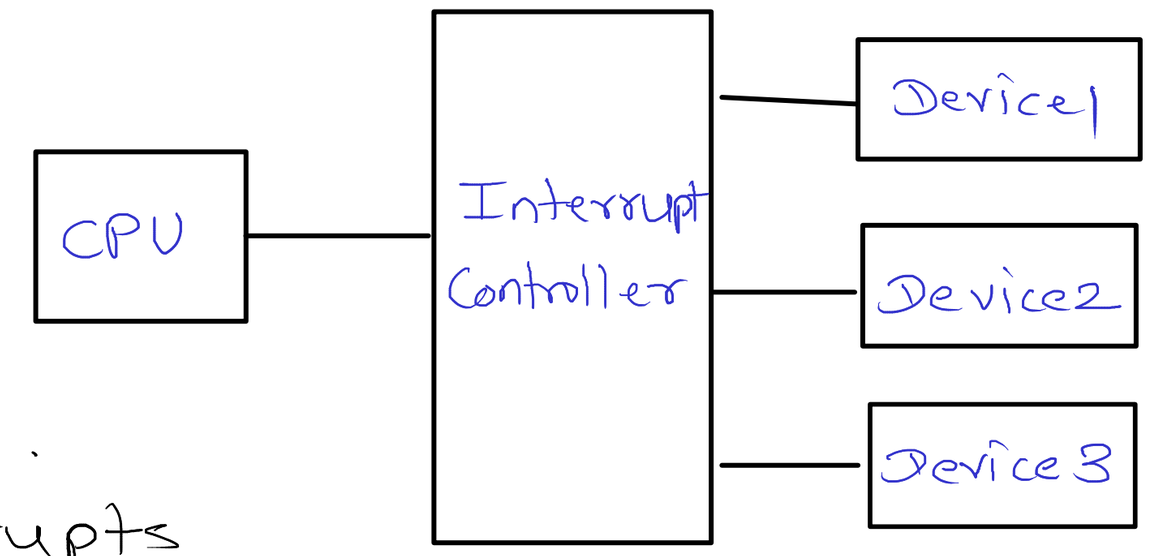
Hardware Technique : Polling



Asynchronous IO

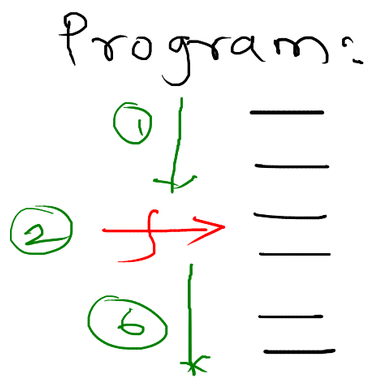
Hardware Technique : Interrupt



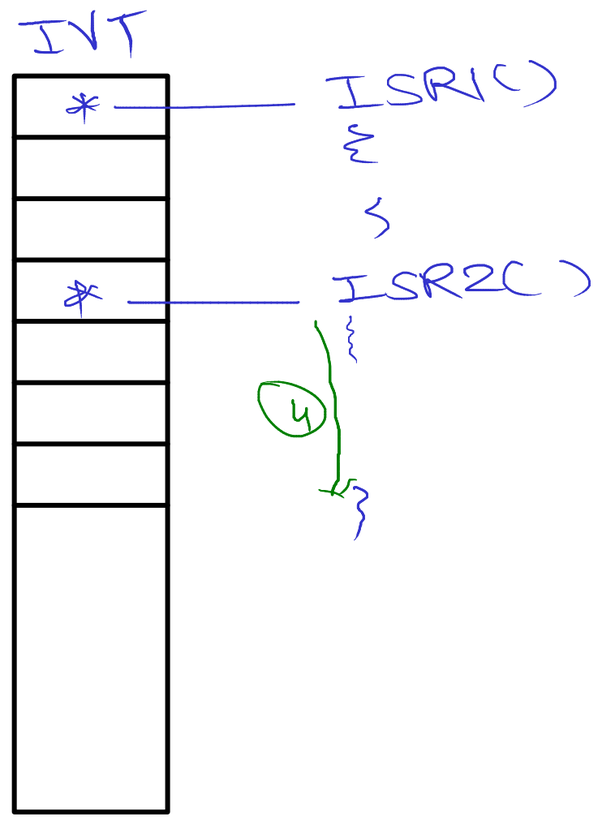


- inform interrupts to CPU.
- decide priority of interrupts

Interrupt Handling



```
interrupt_handler()
{
    1> save execution context of current Process (PCB)
    2> find address of ISR from IVT.
    3> call ISR
    4> Restore execution context of paused process.
}
```



Dual Mode Operations

→ few instructions & CPU registers are not accessible

mode
- 1 User mode / unprivileged
- 0 Kernel mode / system mode / privileged

mode 1

Program:

① ↓ ___

___

___

syscall_api ( ) ;

⑥ ↓ ___

___

___

---

scall_api ( ) {

② ↓ - necessary setting

- s/w interrupt

---

mode 0

interrupt_handler ( )

{

③ ↓ 1) save execution context of current process

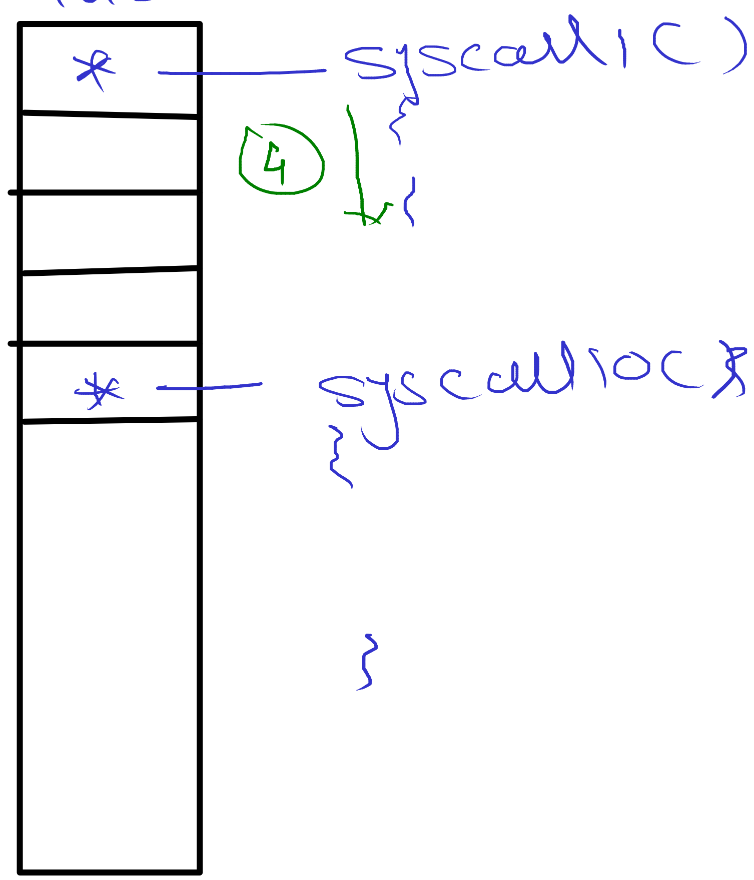
2) Find addr of actual system call implementation from system call table

3) call system call impl^n

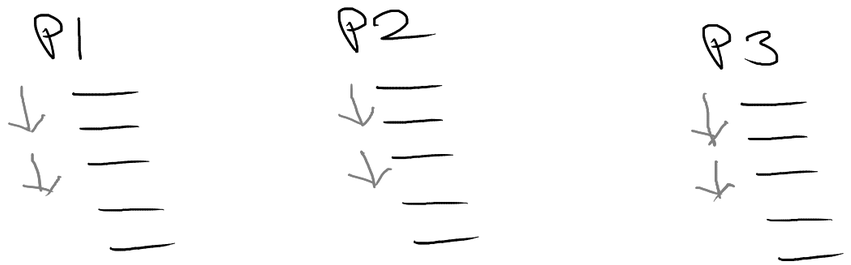
⑤ ↓ 4) Restore execution context of paused process.

}

System call Table



CPU Scheduler




```
interrupt_handler()
{
    1) save execution context
    2) find ISR from IVT
    3) call ISR
    4) pid = scheduler()
    5) cpu_dispatcher(pid)
}

ISR()
{
    ...
}

scheduler()
{
    if(remaining-time > 0)
        select same process
    else
        select another process
    return pid;
}

dispatcher(pid)
{
    restore execution
    content of pid process
}
```