

topics so far: OS requirements / design, CPU (privileged) modes, interrupts, abstractions, process abstraction, PCB, process vs. program, process state, the process ABI — fork, exec, wait, pipe, ... signals.

① reaping a process (zombie status)

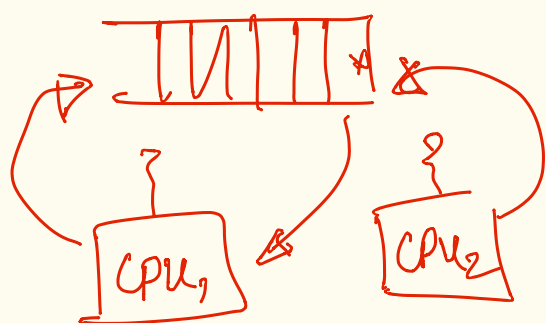
↳ waiting for parent to ack child termination consume status.

② ready status

↳ ready queue — list of all ready processes.

↳ input to the scheduler

↳ uses a scheduling policy to determine next process to execute on CPU.



↳ multi-CPU — synchronization

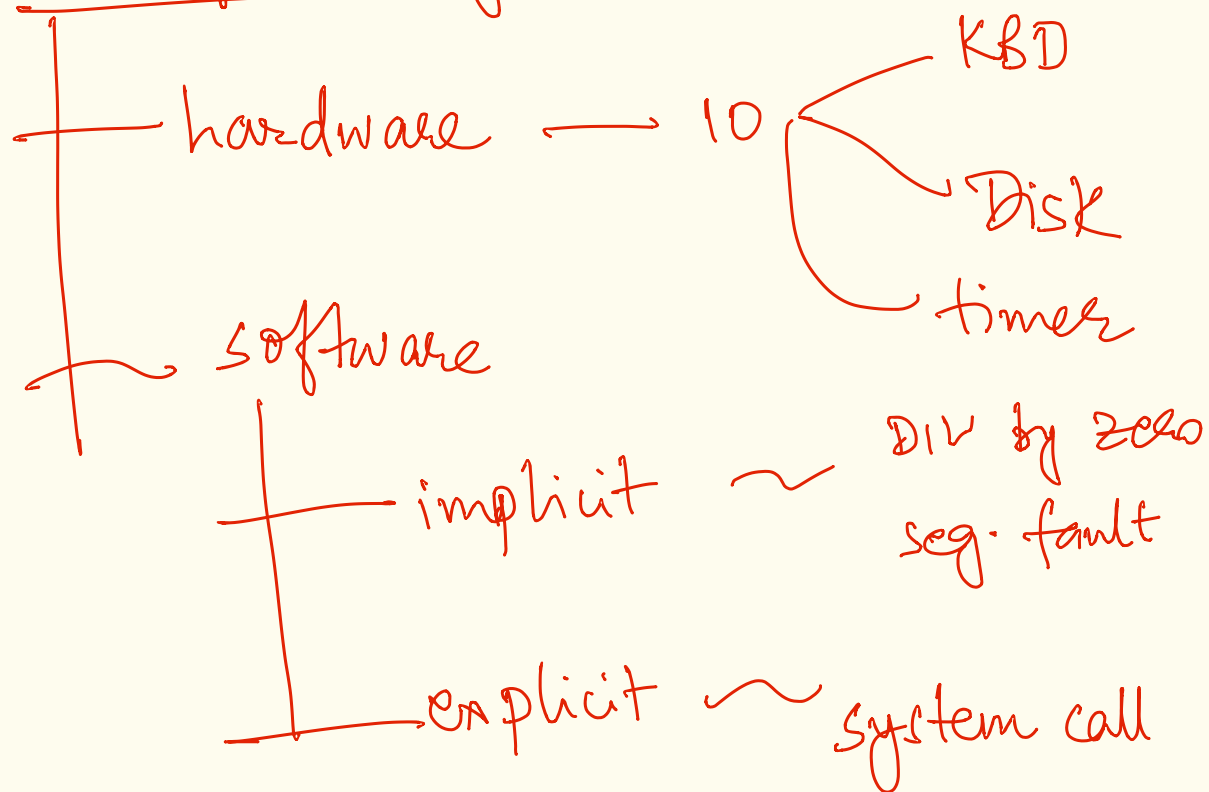
③ context switch

↳ switch out context on CPU of current process & load context on CPU of new process.

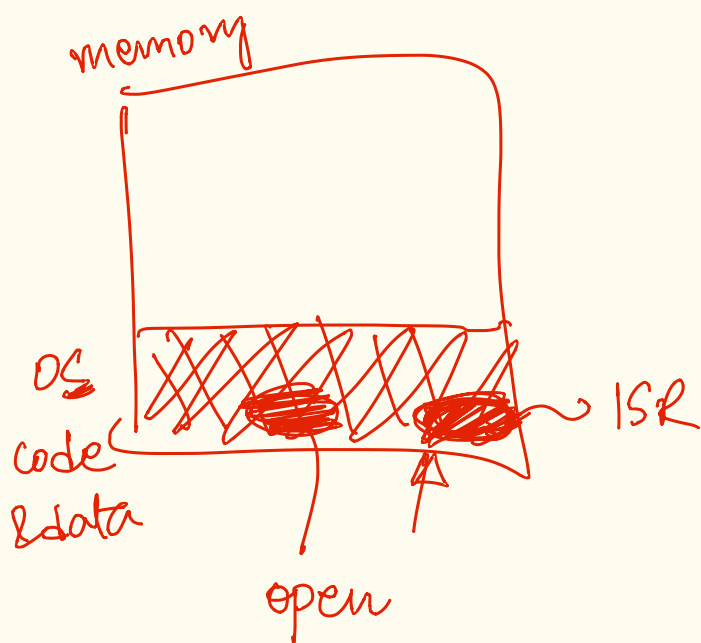
↳ store in PCB — multiple ready queues (one per CPU)

write ("Hello"); — CPU₁
write ("World"); — CPU₂

⑧ interrupts & system calls.



~ interrupt $\Rightarrow$ ISR interrupt service routine (interrupt handler)



⑧ system calls

mechanism to request OS functionality

~ via an ABI

consume functions/functionality from a binary/inst. seq. already in memory.

⑧ system call steps

- mechanism to switch CPU mode of execution
- indexing & identifying relevant location.
↳ system call number.
- mechanism to invoke the system call.

