List the four cases (conditions) causing scheduling

1. Process moves from running → waiting
2. Process moves from running → ready (interrupt)
3. Process moves from waiting → ready
    a. Completion of child
    b. Receipt of signal
4. Process terminates

Non-preemptive OS

Schedules only conditions 1 and 4.

The element of cooperation had to be included by developers of software for these systems. Application programmers would use a system call to release the CPU so that others could use it. To be a certified product, had to play that game.

Examples: Windows 3.x, Mac OS through OS9

Preemptive OS

Schedules for conditions 1 through 4.

Preemption is initiated by device and timer interrupts (2 and 3)

Examples: All of the modern OSes (Windows 95 and onward, Mac OS X, Unix, etc.)

OS Scheduler	OS Dispatcher
Select process order Doesn't have to be fast – could run once a minute.	Context switcher Must be **fast** (**Dispatch latency** must be minimal – no more than 5% of the time slice)

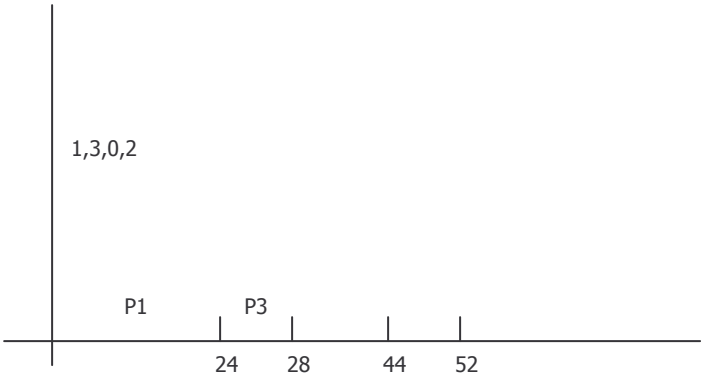
In many OSes, the scheduler and dispatcher are one in the same

Scheduling Algorithms

Prc	CPU Bur
0	16
1	24
2	8
3	4

First Come, First Served (FCFS) Scheduling

- Non-preemptive
- Uses a FIFO queue for the ready queue
    - Dispatcher simply removes what is at the head of the queue


1,3,0,2
P1
P3
24
28
44
52