☑ 1. Last time
☑ 2. Scheduling intro
☑ 3. Scheduling disciplines
- ☑ FIFO
- ☑ SJF
- ☑ RR
- ☑ incorporating I/O
- ☑ priority
- ☑ MLFQ
- ☑ Lottery

☑ 4. Scheduling lessons and conclusions

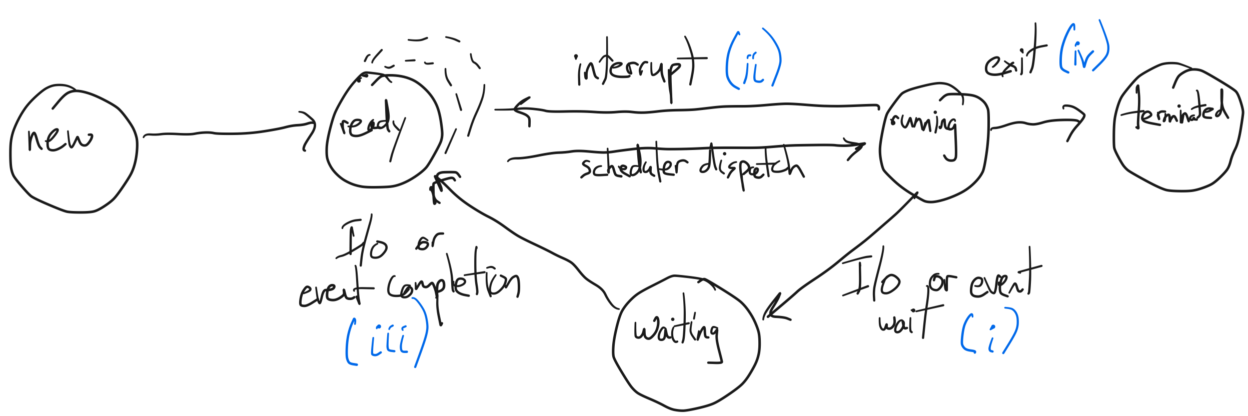
---

2. Scheduling intro

High-level problem: OS has to decide which process (or thread) to run

A. When scheduling decisions happen

process state/transitions:



scheduling decisons happen when process:
(i) Switches from running to waiting
(ii) Switches from running to ready
(iii) Switches from waiting to ready
(iv) Exits

preemptive scheduling vs. non-preemptive scheduling

B. Metrics and criteria

turnaround time : time for each process/thread to complete

waiting/response/output time : time spent waiting for something to happen

system throughput : # completed processes per unit time

fairness : often conflicts w/ efficiency

C. Context switching has a cost

CPU time in kernel

regs

addr. space switches

indirect costs

---

3. Scheduling disciplines

Assume first that processes/threads do not do I/O (unrealistic; we relax that later)

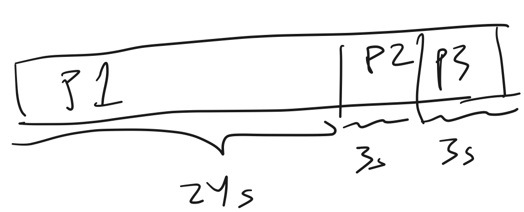
A. FCFS / FIFO

- run job until done

P1 : needs 24 s

P2 : needs 3 s

P3 : needs 3 s



throughput: 0.1 jobs/sec

avg tt: $\frac{1}{3}(24 + 27 + 30) = 27$ s.

| P2 | P3 | P1 |

$\frac{1}{3}(3 + 6 + 30) = 13$ s

B. SJF + STCF

SJF: choose job w/ shortest upcoming CPU burst

STCF: preemptive version of SJF

example:

process	arrival time	burst time
P1	0	~~7~~ 5
P2	2	~~4~~ 2
P3	4	1
P4	5	4

time	0	1	2	3	4	5	6	7	8	9	10	11	
	P1	P1	P2	P2	P3	P2	P2	P4	P4	P4	P4	P1	...

C. Round robin (RR)

- add a timer
- quantum

what if jobs are same length?

eg: 2 jobs of 50 time units each, quantum is 1

- avg. # ? 100
- but if we did FCFS? $\frac{(50+100)}{2} = 75$

quantum: 10-100ms.
context switches: μs 10s of μs

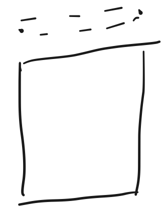
D Incorporating I/O

motivating example:

3 jobs

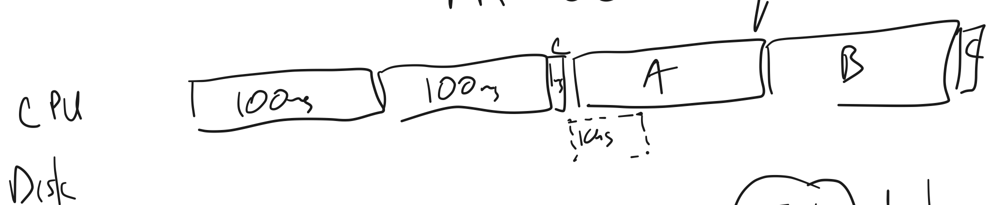
100% of CPU { A, B: CPU-bound, run for a week

90% of disk { C: I/O-bound, loop
1 ms of CPU
10 ms of disk I/O



what happens if we use FIFO?

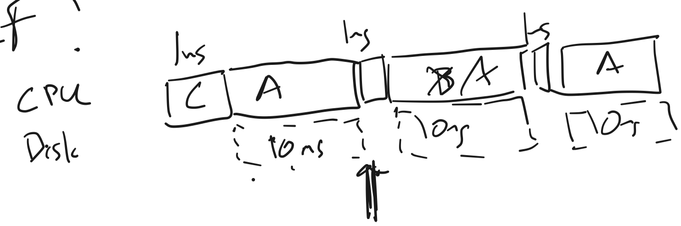
" RR 100msec quantum?



$\frac{10\text{ms}}{201\text{ms}} \approx$ 5% disk utilization

" " " RR 1ms quantum?
90% disk utilization

STCF?



disk utilization

Estimate:

(cardinal)

EWMA

t_n : length of a process's n^{th} CPU burst

τ_{n+1} : estimate for the $n+1^{st}$ burst

$0 < \alpha \le 1$

$$\tau_{n+1} = \alpha \cdot t_n + (1-\alpha)\cdot \tau_n$$

~~$(1-\alpha)(\alpha \cdot t_{n-1} + \alpha^2 \cdot t_{n-2} + \alpha^3 \cdot t_{n-3} \dots)$~~

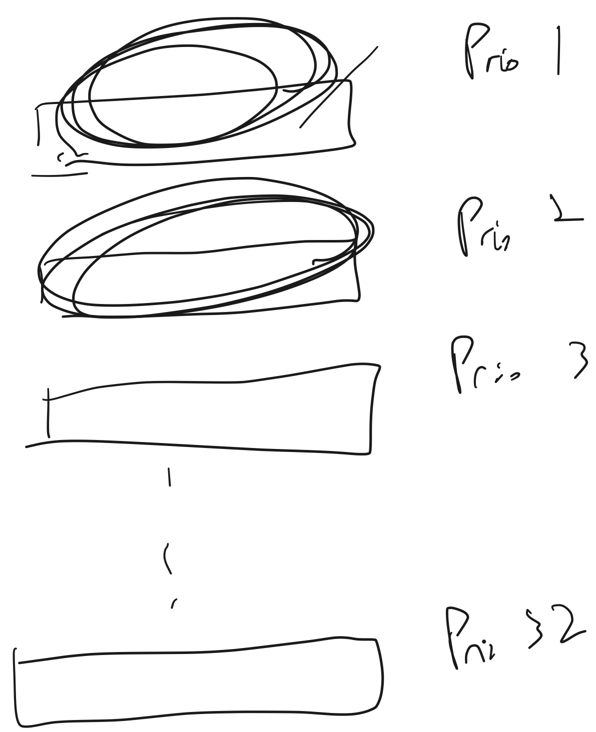
$$= \alpha t_n + (1-\alpha)\cdot\alpha t_{n-1} + (1-\alpha)^2\cdot\alpha t_{n-2} + (1-\alpha)^3\cdot\alpha t_{n-3} + \dots$$

Each older term is given exponentially less weight.

E. Priority scheme

SJF

F. MLFQ



approx STCF

disadvs:

G. lottery + stride scheduling (Chap. 9)

P_i will have b_i

$$B = \sum_{i=1}^{n} b_i$$

$$\frac{b_i}{B}$$

II. Linux: CFS

$\sim$ stride scheduling

---

4. Lessons metrics

(1) Know your goals
(2) Compare against optimal
(3) There are different schedulers that interact