

Scheduling Review

FIFO:

- + simple
- - short jobs can get stuck behind long ones; poor I/O device

RR:

- + better for short jobs
- - hard to select right time slice
- - poor turnaround time when jobs are the same length

Scheduling Review Cont.

SJF:

- + minimal average waiting time
- - hard to predict the next CPU burst length
- - unfair

Multi-level feedback:

- + approximate SJF (gives preference to short jobs)
- + establishes the nature of a process quickly
- - unfair to long running jobs

Example 1

Thread	Arrival Time	Burst Length
A	0	10
B	1	5
C	3	2

- FIFO turn around time

$$A: (10-0) = 10$$

$$B: (15-1) = 14$$

$$C: (17-3) = 14$$

$$(10+14+14)/3 = 12.66$$

- FIFO wait time

$$A: 0$$

$$B: (10-1) = 9$$

$$C: (15-3) = 12$$

$$(0+9+12)/3 = 7$$

Example 2

Thread	Arrival Time	Burst Length
A	0	10
B	1	5
C	3	2

- SJF turn around time

$$B: 8 - 1 = 7$$

$$C: 5 - 3 = 2$$

$$A: 17 - 0 = 17$$

$$(17 + 2 + 7) / 3 = 8.67$$

- SJF wait time

$$B: 2$$

$$C: 0$$

$$A: 2 + 2 + 3 = 7$$

$$(2 + 0 + 7) / 3 = 3$$

Priority Inversion

- Have three processes
- P1: Highest priority; P2: Medium; P3: Lowest
- P1 and P3 have this code:

```
P(mutex);  
<critical section>  
V(mutex);
```

- P2 is a long-running task
- P3 acquires mutex; preempted
- P1 tries to acquire mutex; blocks
- P2 enters the system at medium priority; runs
- P3 never gets to run; P1 never gets to run!!
- This happened on Mars Pathfinder in 1997!
- Solutions?