Lecture 4
Scheduling

I233E OPERATING SYSTEMS

RAZVAN BEURAN

Today’s Topics

Scheduling

- Why it is needed
- How it works
- Preemption

Scheduling algorithms

- First-come, first-served
- Shortest job first
- Shortest remaining time
- Priority scheduling
- Round-robin scheduling

Scheduling Decisions

Process creation

- Who to schedule? Parent? Child?

Process termination

- Which process to schedule next?
- What if no process is ready?

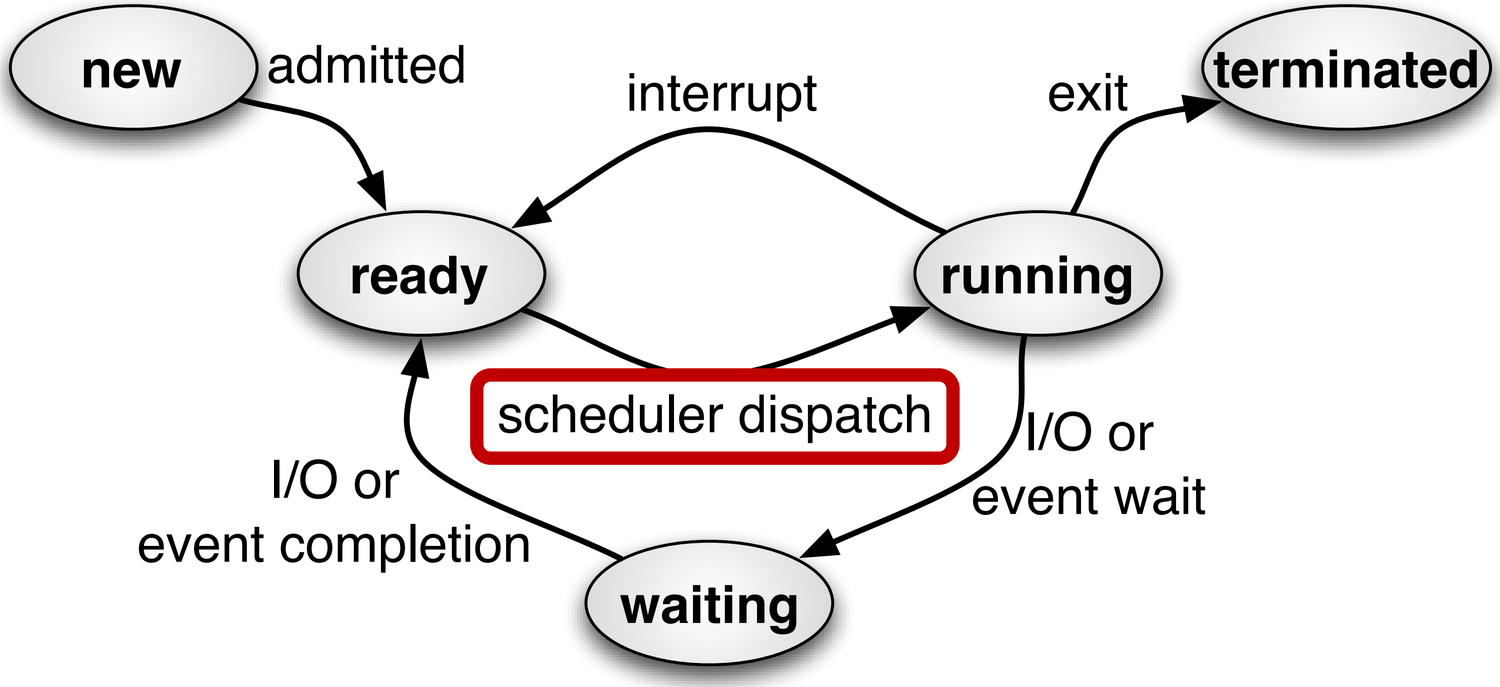
Process waiting (I/O, semaphore, wait)

- Which process to schedule next?

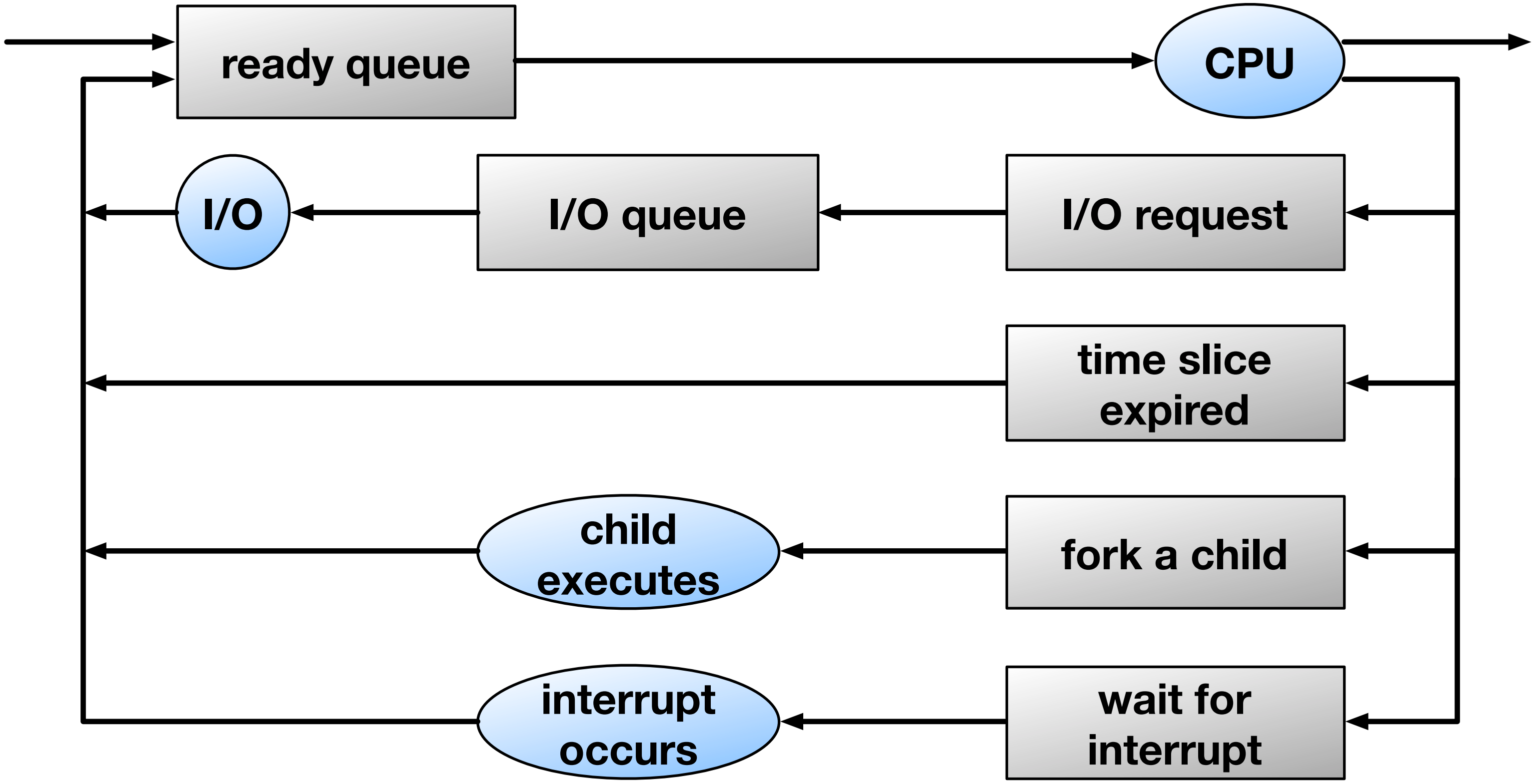
Interrupt (I/O, alarm)

- What if some process wakes up?

Process States



Scheduling Diagram



Preemption

Preemptive scheduling

- Process runs for a fixed time interval
- Process can be interrupted anytime (anywhere)

Non-preemptive scheduling

- Process runs until it blocks
- Process can explicitly “yield”
- If interrupted, come back to the same process

Scheduling Goals

All systems

- Fairness
- Policy enforcement
- Balance

Batch execution

- Throughput
- Turnaround time
- CPU utilization

Interactive tasks

- Response time
- Proportionality

Real-time execution

- Meeting deadlines
- Predictability

Scheduling Algorithms

First-Come, First-Served

Idea

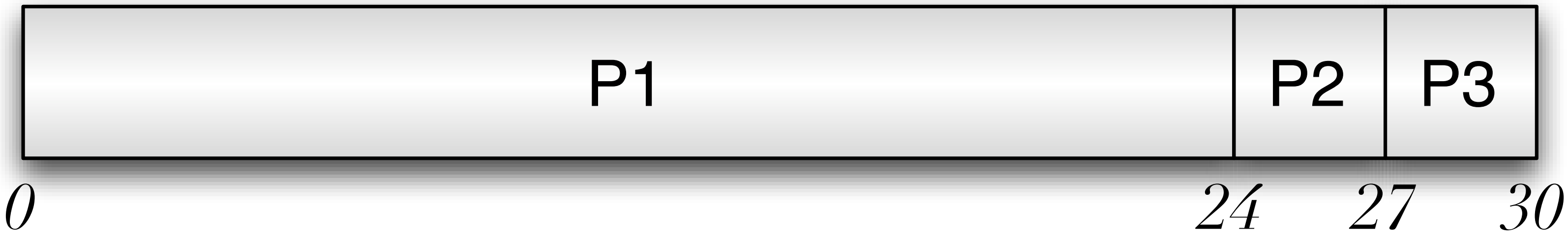
- Jobs scheduled in order of arrival

Characteristics

- Simplest policy
- Average waiting time can be high

Process	Burst
P1	24 ms
P2	3 ms
P3	3 ms

[Arrival = 0 ms]



Avg. waiting time = 17 ms

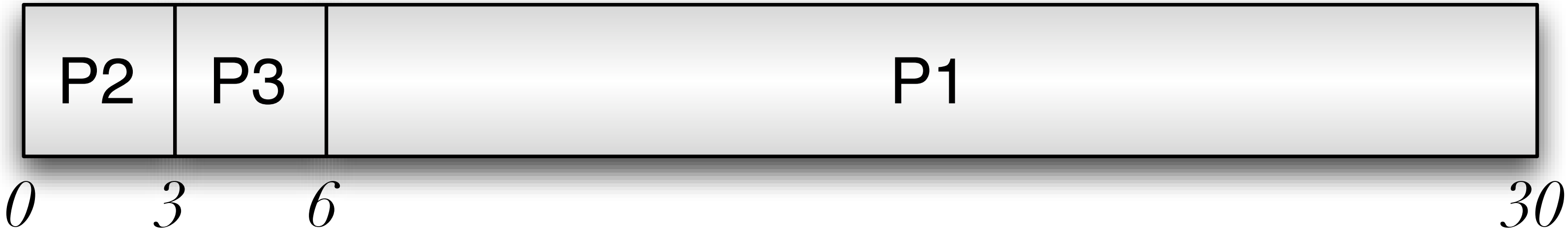
First-Come, First-Served (cont.)

Average waiting time

- Important performance metric for scheduling algorithms
- Other scheduling order leads to other waiting time

Process	Burst
P2	3 ms
P3	3 ms
P1	24 ms

[Arrival = 0 ms]



Avg. waiting time = 3 ms!

Shortest Job First

Idea

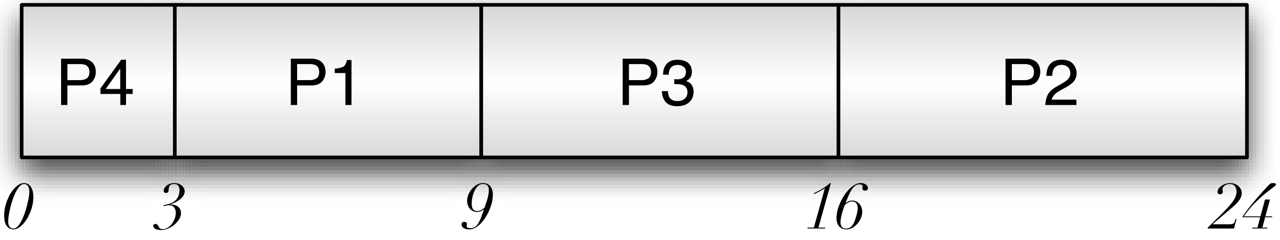
- Jobs scheduled in the order of running time (burst)

Characteristics

- Shorter average waiting time
- How to estimate burst duration?

Process	Burst
P1	6 ms
P2	8 ms
P3	7 ms
P4	3 ms

[Arrival = 0 ms]



Avg. waiting time = 7 ms (vs. 10.25 ms for FCFS)

Shortest Remaining Time

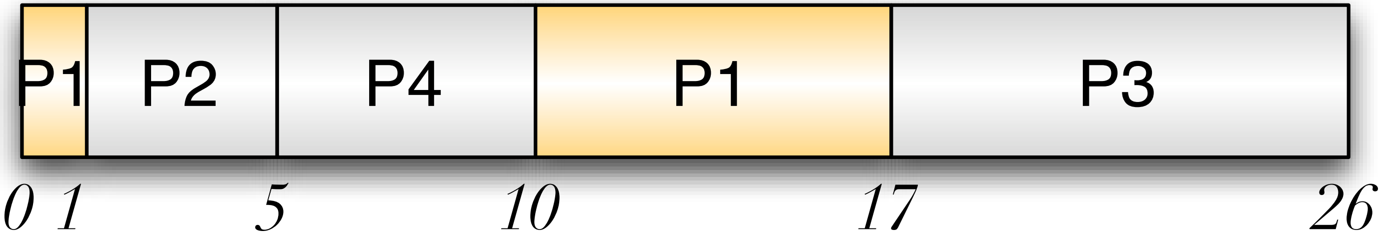
Idea

- Shortest Job First with **preemption**

Characteristics

- Shortest (optimal) waiting time

Process	Arrival	Burst
P1	0 ms	8 ms
P2	1 ms	4 ms
P3	2 ms	9 ms
P4	3 ms	5 ms



Avg. waiting time = 6.5 ms (vs. 7.75 ms for non-preemptive SJF)

Priority Scheduling

Idea

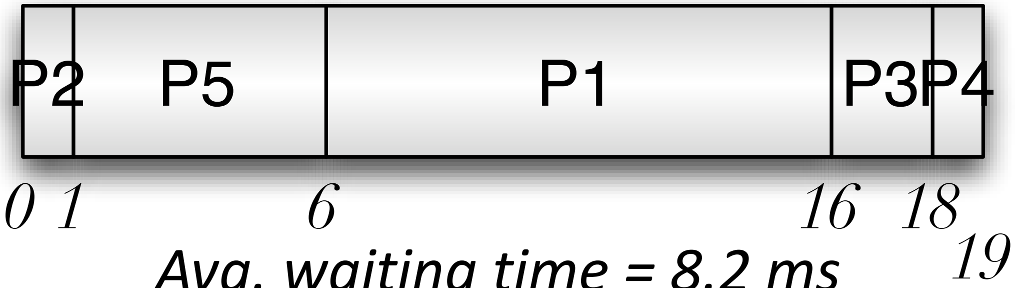
- Processes have “priority”
- High priority (small level number) processes are scheduled first
- Equal priority → FCFS

Characteristics

- Prone to starvation

Process	Burst	Priority
P1	10 ms	3
P2	1 ms	1
P3	2 ms	4
P4	1 ms	5
P5	5 ms	2

[Arrival = 0 ms]



Avg. waiting time = 8.2 ms

Round-Robin Scheduling

Idea

- Basically, FCFS with preemption
- Use *time quantum* (10 – 100 ms)

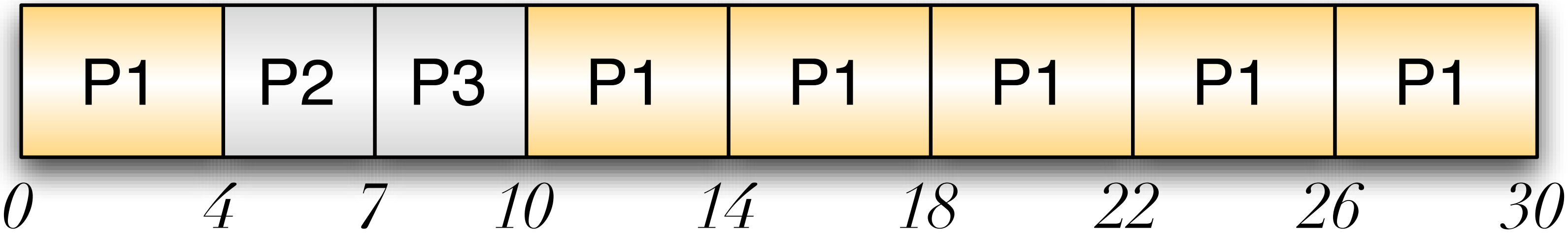
Characteristics

- Fairness between processes
- Short quantum → increased overhead
- Average waiting time can be long

Process	Burst
P1	24 ms
P2	3 ms
P3	3 ms

[Arrival = 0 ms]

Quantum = 4 ms



P1 | P2 | P3 | P1 | P1 | P1 | P1 | P1

0 4 7 10 14 18 22 26 30

Avg. waiting time = 5.67 ms

Multi-level Queue Scheduling

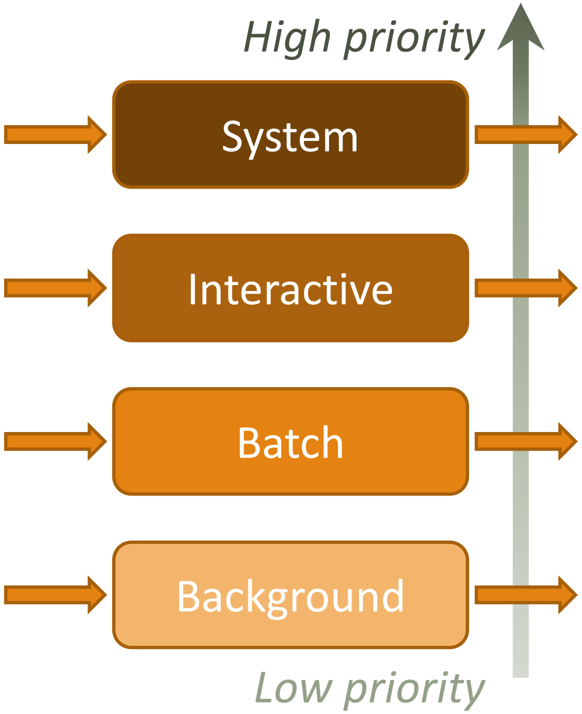
Groups of processes with common characteristics

- System processes
- Interactive processes
- Batch processes
- Background processes

Between classes → priority scheduling

Within each class → some scheduler

- Preemptive
- Non-preemptive



Dynamic Priorities

Idea

- Priority scheduling paradigm
- If CPU time use is high → process priority decreases

Characteristics

- Favors processes that do not use CPU
  - I/O-bound processes
  - Interactive processes
- Prevents starvation

Real-Time Scheduling

Hard vs. soft deadlines

- Important in real-time operating systems

Schedulability

- Can a process be scheduled?

Scheduling algorithms

- Static
- Dynamic

Priority inversion

- Priority inheritance

Summary

Scheduling principles

- Scheduling is needed to decide which process should be executed next
- Preemption → processes can be interrupted after running for a certain time interval

Scheduling algorithms

- Various algorithms exist: FCFS, SJF, SRT, etc.
- Average waiting time is an important metric
- Fairness can be an issue for some algorithms
- Scheduling also used in other fields, such as networking

Next Time

Synchronization

- Race conditions

Classical problems

- Mutual exclusion
- Producer-consumer