

Systems 3

Scheduling

Marcel Waldvogel

(Handout)

Department of Computer and Information Science
University of Konstanz

Winter 2019/2020

How many active processes are running?

- 1 The program in foreground
- 2 Mail daemon
- 3 Update checker
- 4 SSH client
- 5 Antivirus program
- 6 ...

Chapter Goals

- How do processes (and threads) use the CPU?
- Why do we need scheduling?
- What are the different scheduling options? What are their pros and cons?
- Can we achieve fairness?
- What is the difference between a thread and a process?
- What are the advantages of threads?

Different process behavior

- **compute-bound**

spend most of their time computing

- **I/O-bound**

spend most of their time waiting for I/O

When to Schedule

When scheduling is absolutely required:

- 1 When a process exits.
- 2 When a process blocks on I/O or a mutual exclusion mechanism.

When scheduling usually done (though not absolutely required)

- 1 When a new process is created.
- 2 When an I/O interrupt occurs.
- 3 When a clock interrupt occurs.

Why? When?

Goals of scheduling algorithms

- All systems
 - Fairness
 - Policy enforcement
 - Balance
- Batch systems
 - Throughput
 - Turnaround time
 - CPU utilization
- Interactive systems
 - Response time
 - Proportionality
 - User happiness
- Real-time systems
 - Avoiding event loss
 - Avoiding data loss
 - Predictability

Basic algorithms for batch systems

FCFS First-Come First-Serve (nonpreemptive)

SJF Shortest Job First (nonpreemptive)

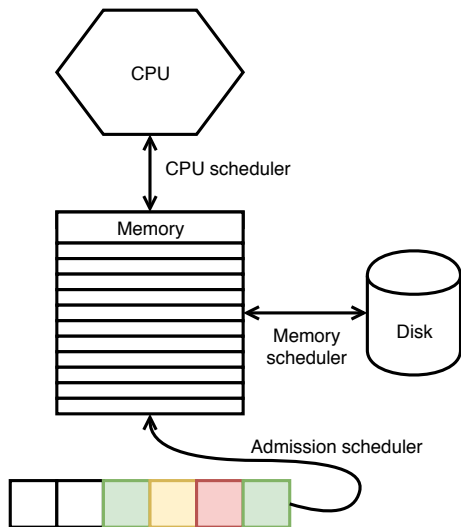
SRT Shortest Remaining Time Next (preemptive)

Three Level Scheduling

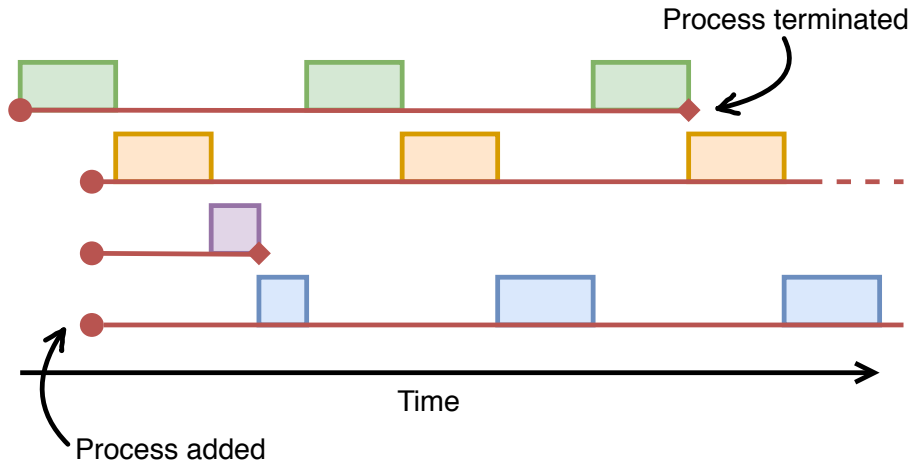
Criteria for deciding which process to choose:

- How long has it been since the process was swapped in or out?
- How much CPU time has the process had recently?
- How big is the process? (Small ones do not get in the way.)
- How important is the process?

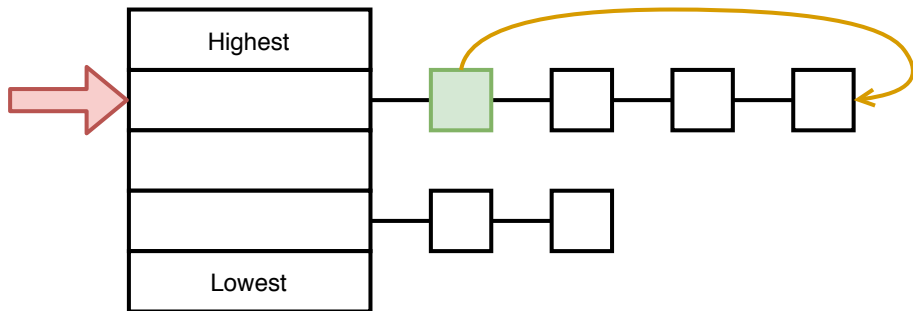
How determined?



RR: Round-Robin Scheduling

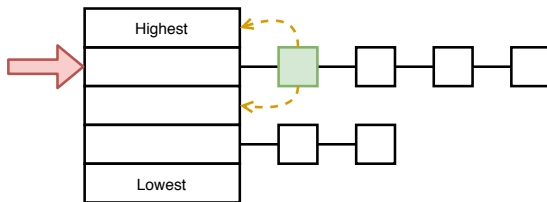


Priority Scheduling



Starvation!

Dynamic Priorities



- Processes are associated with priorities
- Scheduling as in Priority Scheduling
- Additionally:
 - When a process uses up its quantum, it's priority is reduced
 - When a process does not use up its quantum, it's priority is increased

What does this achieve?

Real-time Systems

- time limit
- hard¹ vs. soft² real time
- processes with predictable behavior
- processes (or actions) are generally short lived

¹Something bad is going to happen (e.g., brake system)

²The value of the result to be computed is reduced or zero (e.g., video playout)

What is used?

System	Goals	Scheduler
Real-time	React to events in time	Strict Priority
Server	Fast reaction to many requests	Dynamic priority
HPC	Finish simulations fast	Don't care/Admission
Desktop	Fast reaction to user inputs	Dynamic priority