Betriebssysteme

Tutorium 3

Peter Bohner

14. November 2024

ITEC - Operating Systems Group

Scheduling basics

Scheduling - About

What is a Scheduler? Why do we even need it?

Scheduling - About

What is a Scheduler? Why do we even need it?

- Maps processes to resources

Scheduling - About

What is a Scheduler? Why do we even need it?

- Maps processes to resources
- Ideally: Every process gets what it needs some day

Scheduling - About

What is a Scheduler? Why do we even need it?

- Maps processes to resources
- Ideally: Every process gets what it needs some day

What Schedulers do you know?

Scheduling - About

What is a Scheduler? Why do we even need it?

- Maps processes to resources
- Ideally: Every process gets what it needs some day

What Schedulers do you know?

- CPU-Scheduler: The classic

Scheduling - About

What is a Scheduler? Why do we even need it?

- Maps processes to resources
- Ideally: Every process gets what it needs some day

What Schedulers do you know?

- CPU-Scheduler: The classic
- Disk-Scheduler: Why have one?

Scheduling - About

What is a Scheduler? Why do we even need it?

- Maps processes to resources
- Ideally: Every process gets what it needs some day

What Schedulers do you know?

- CPU-Scheduler: The classic
- Disk-Scheduler: Why have one? Multiplexing but also efficiency!

Scheduling - About

What is a Scheduler? Why do we even need it?

- Maps processes to resources
- Ideally: Every process gets what it needs some day

What Schedulers do you know?

- CPU-Scheduler: The classic
- Disk-Scheduler: Why have one? Multiplexing but also efficiency!
- Network I/O:

Scheduling - About

What is a Scheduler? Why do we even need it?

- Maps processes to resources
- Ideally: Every process gets what it needs some day

What Schedulers do you know?

- CPU-Scheduler: The classic
- Disk-Scheduler: Why have one? Multiplexing but also efficiency!
- Network I/O: When to send packets, which packets to drop, QoL,...

Scheduling - Long- and Short-Term Scheduler

What are the differences? When are they used?

Scheduling - Long- and Short-Term Scheduler

What are the differences? When are they used?

- LTS: Decide which processes to put in the *run queue*

Scheduling - Long- and Short-Term Scheduler

What are the differences? When are they used?

- LTS: Decide which processes to put in the *run queue*
- STS: Decide which process runs on the *CPU*

Scheduling - Long- and Short-Term Scheduler

What are the differences? When are they used?

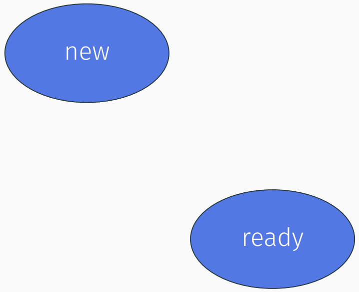
- LTS: Decide which processes to put in the *run queue*
- STS: Decide which process runs on the *CPU*
- MTS: Temporarily removes processes from main memory (and e.g. writes them out to disk)

⇒ Reduce degree of multiprogramming, make room in memory (and a few other reasons)

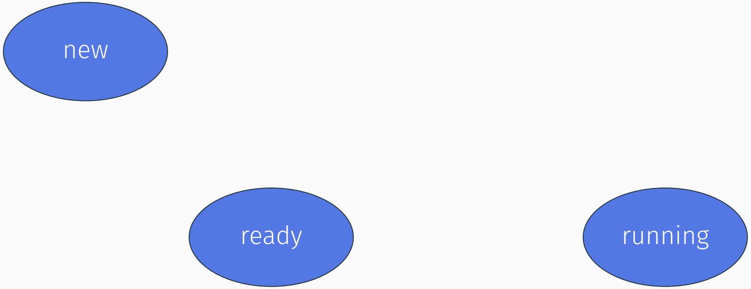
Process states

Process states

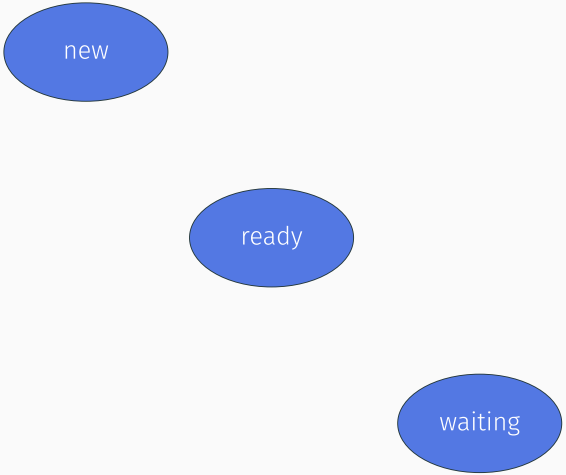
Process states



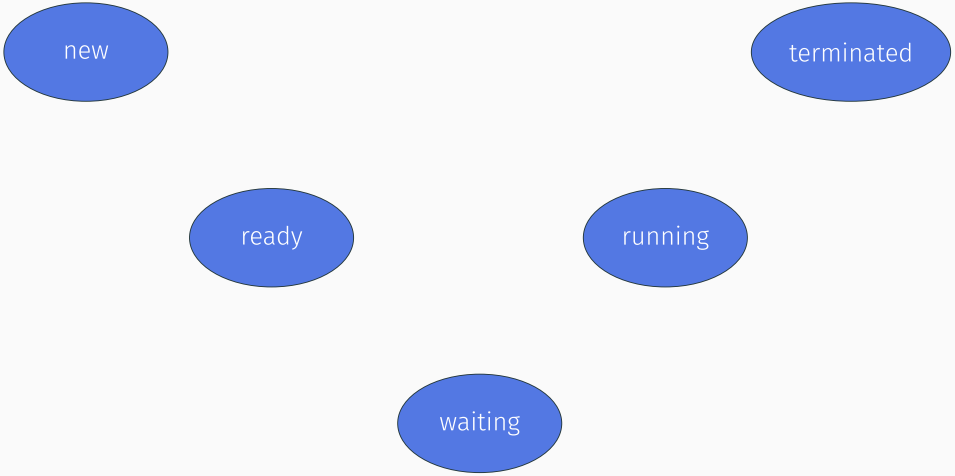
Process states



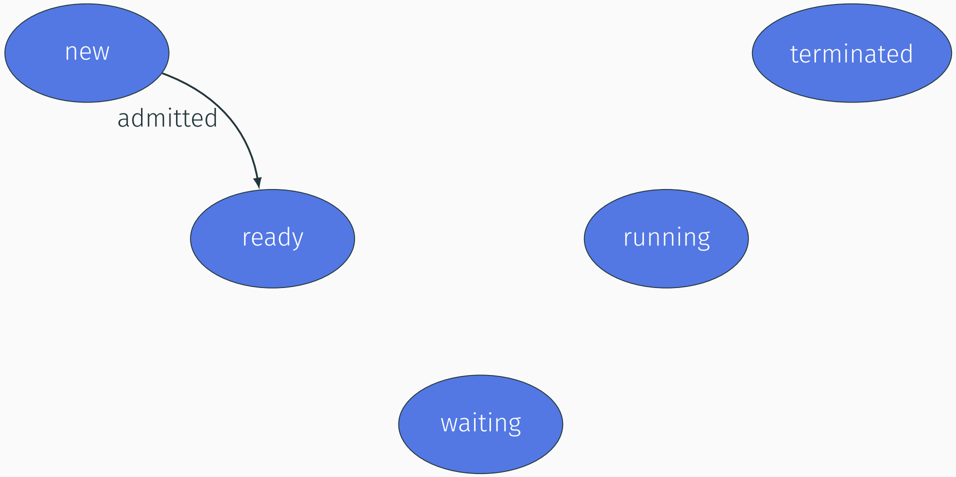
Process states



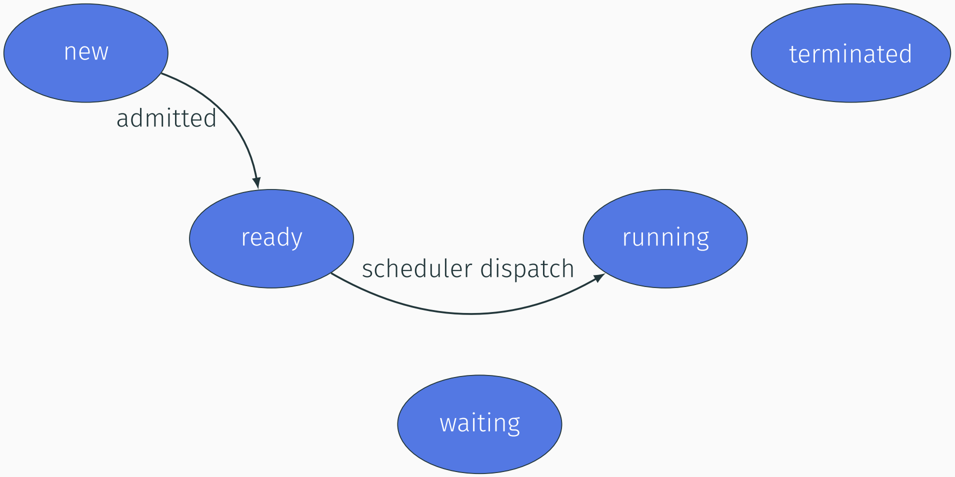
Process states



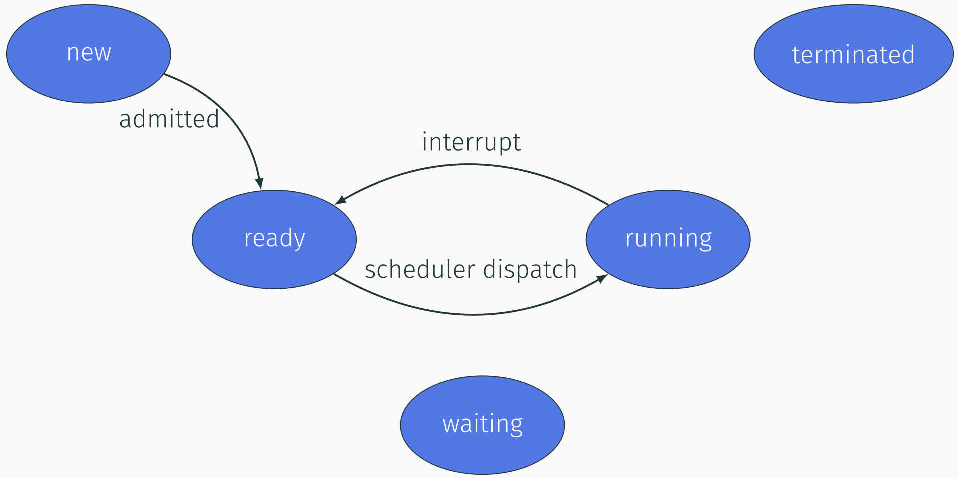
Process states



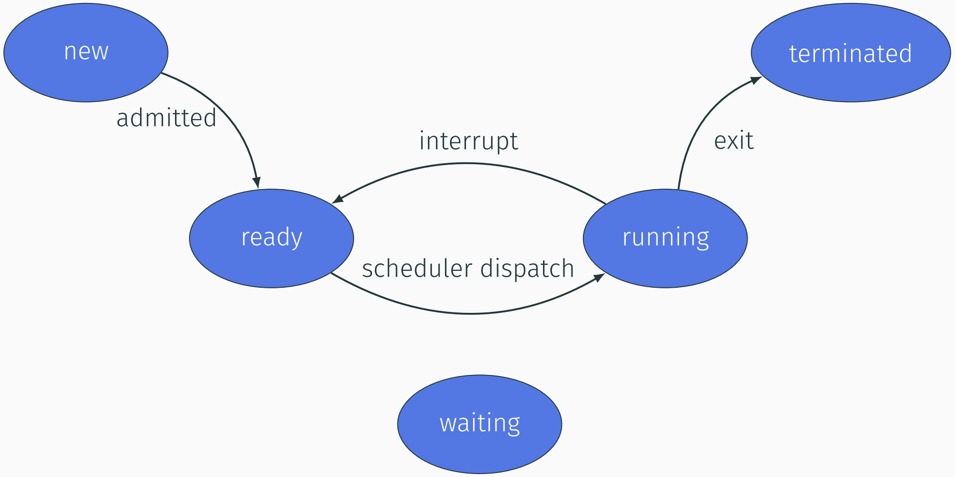
Process states



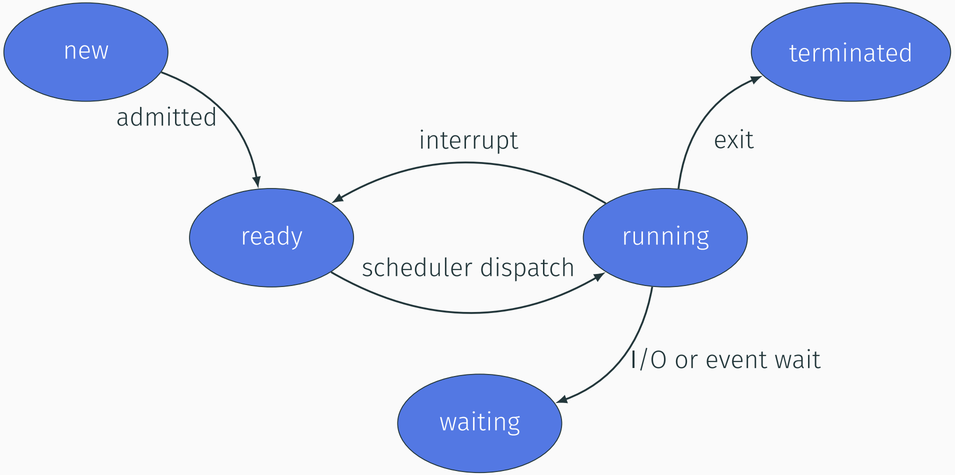
Process states



Process states



Process states



Process States - Waiting

„I/O or event wait“? When does a process move from ready to waiting?

- Network / Disk I/O

Process States - Waiting

„I/O or event wait“? When does a process move from ready to waiting?

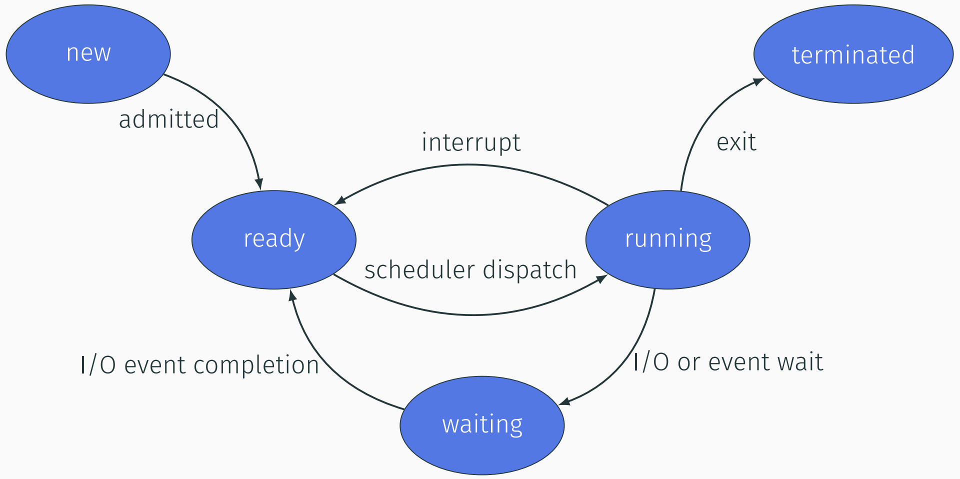
- Network / Disk I/O
- Mutex or other inter-process synchronisation

Process States - Waiting

„I/O or event wait“? When does a process move from ready to waiting?

- Network / Disk I/O
- Mutex or other inter-process synchronisation
- Sleepyness

Process states



Scheduling - Scheduler Worldcup

What makes a good Scheduler good?

Let’s play scheduler!

Scheduling - Scheduler Worldcup

What makes a good Scheduler good?

Let's play scheduler!

Some metrics

- Processor utilization: Percentage of working time
- Throughput: How many jobs do you finish?
- Turnaround time: Wallclock-time from submission to finish
- Waiting time: How long did it spend in the ready queue
- Response time: Time between submission of a request and first response (e.g. key press to echo on screen)

Scheduling - Preemption the third

What does your hardware need to support to allow non-cooperative scheduling?

Scheduling - Preemption the third

What does your hardware need to support to allow non-cooperative scheduling?

Timer Interrupts! Waiting for a cosmic ray to hit, a network package to arrive, a system call or any other random interrupt gets old fast :)

Scheduling - When to interrupt

Any guesses for how long a timeslice usually is?

Scheduling - When to interrupt

Any guesses for how long a timeslice usually is?

2ms - 200ms

- On windows it depends on the configuration (favor foreground / background processes)

Scheduling - When to interrupt

Any guesses for how long a timeslice usually is?

2ms - 200ms

- On windows it depends on the configuration (favor foreground / background processes)
  Which setting has the longer timeslice?

Scheduling - When to interrupt

Any guesses for how long a timeslice usually is?

2ms - 200ms

- On windows it depends on the configuration (favor foreground / background processes)
  Which setting has the longer timeslice?
  20ms to 30ms for foreground, 180ms to 200ms for background

Scheduling - When to interrupt

Any guesses for how long a timeslice usually is?

2ms - 200ms

- On windows it depends on the configuration (favor foreground / background processes)
  Which setting has the longer timeslice?
  20ms to 30ms for foreground, 180ms to 200ms for background
- Linux’s „Completely Fair Scheduler“ adjusts them dynamically based on the priority, number of processes, ...

Scheduling - When to interrupt

Any guesses for how long a timeslice usually is?

2ms - 200ms

- On windows it depends on the configuration (favor foreground / background processes)
  Which setting has the longer timeslice?
  20ms to 30ms for foreground, 180ms to 200ms for background
- Linux's „Completely Fair Scheduler“ adjusts them dynamically based on the priority, number of processes, ...

Benefits of shorter/longer timeslices?

Scheduling - When to interrupt

Any guesses for how long a timeslice usually is?

2ms - 200ms

- On windows it depends on the configuration (favor foreground / background processes)
  Which setting has the longer timeslice?
  20ms to 30ms for foreground, 180ms to 200ms for background
- Linux’s „Completely Fair Scheduler“ adjusts them dynamically based on the priority, number of processes, ...

Benefits of shorter/longer timeslices?

- Short: High interactivity, higher overhead

Scheduling - When to interrupt

Any guesses for how long a timeslice usually is?

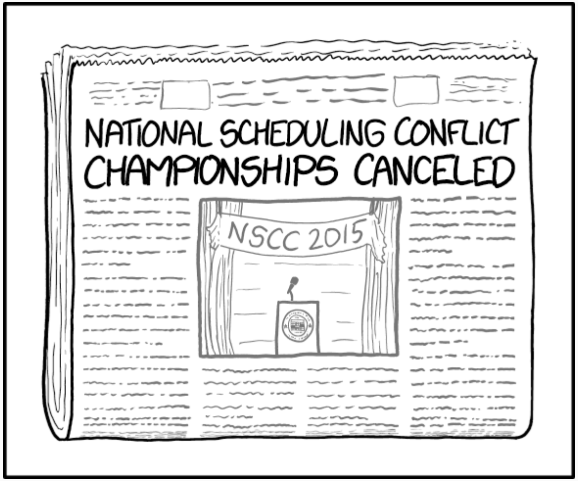
2ms - 200ms

- On windows it depends on the configuration (favor foreground / background processes)
  Which setting has the longer timeslice?
  20ms to 30ms for foreground, 180ms to 200ms for background
- Linux's „Completely Fair Scheduler" adjusts them dynamically based on the priority, number of processes, …

Benefits of shorter/longer timeslices?

- Short: High interactivity, higher overhead
- Long: Lower interactivity, smaller overhead

Ende



XKCD 1542 - Scheduling Conflict

F R A G E N?

Bis nächste Woche :)