Scheduling Policies: Introduction

Operating Systems

Baochun Li

University of Toronto

CPU scheduling: Revisiting our motivation

An OS scheduler decides when a thread should be run

Threads alternate between computation and I/O, called CPU and I/O bursts

- A CPU-bound thread has infrequent I/O bursts
- A I/O-bound thread has infrequent CPU bursts

When a thread performs I/O, CPU is not needed

It is the job of the scheduler to run another thread when a thread is waiting for I/O

- To keep the processor busy and improve its utilization

Assumptions (to be relaxed later)

Each job runs for the same amount of time

All jobs arrive at the same time

Once started, each job runs to completion

All jobs only use the CPU (no I/O)

The run-time of each job is known

Design objectives of scheduling policies

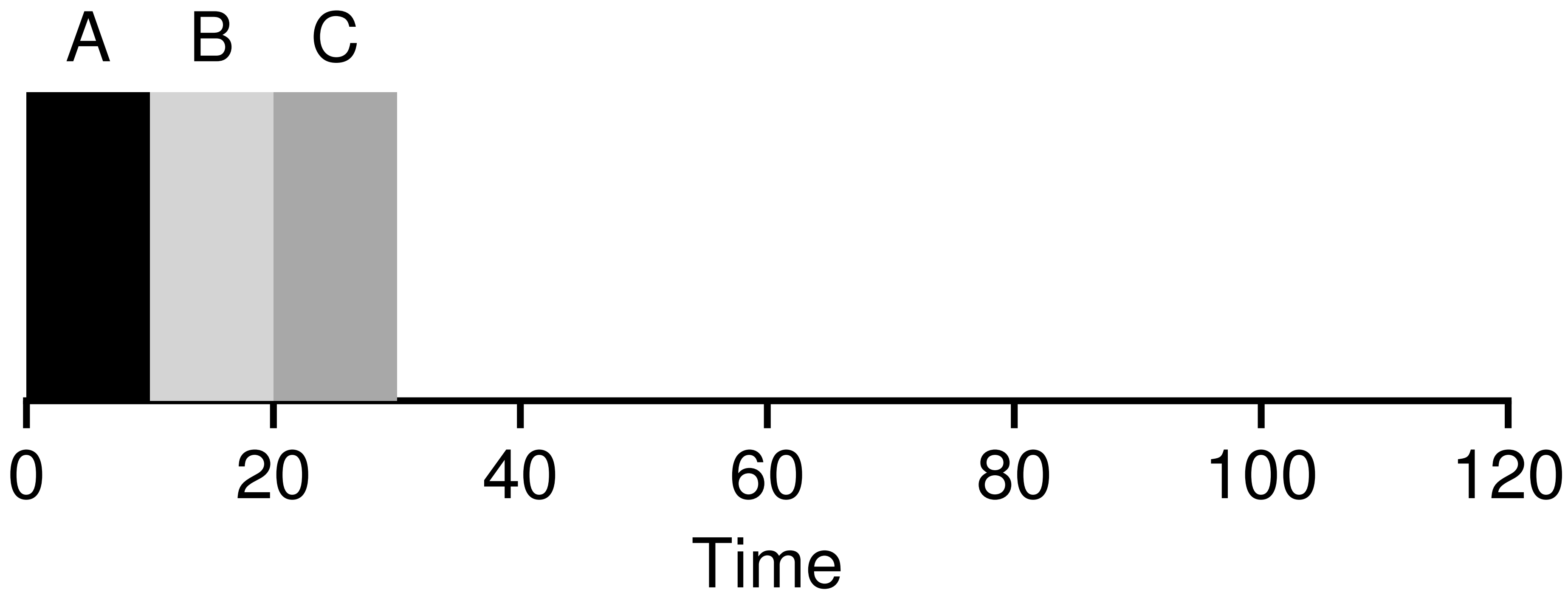
Turnaround time: total time needed to complete a job — $T_{completion}$ - $T_{arrival}$

Fairness: give each thread its fair share

Response Time: the time from when the job arrives to the first time it is scheduled — $T_{firstrun}$ - $T_{arrival}$

First-Come First-Served

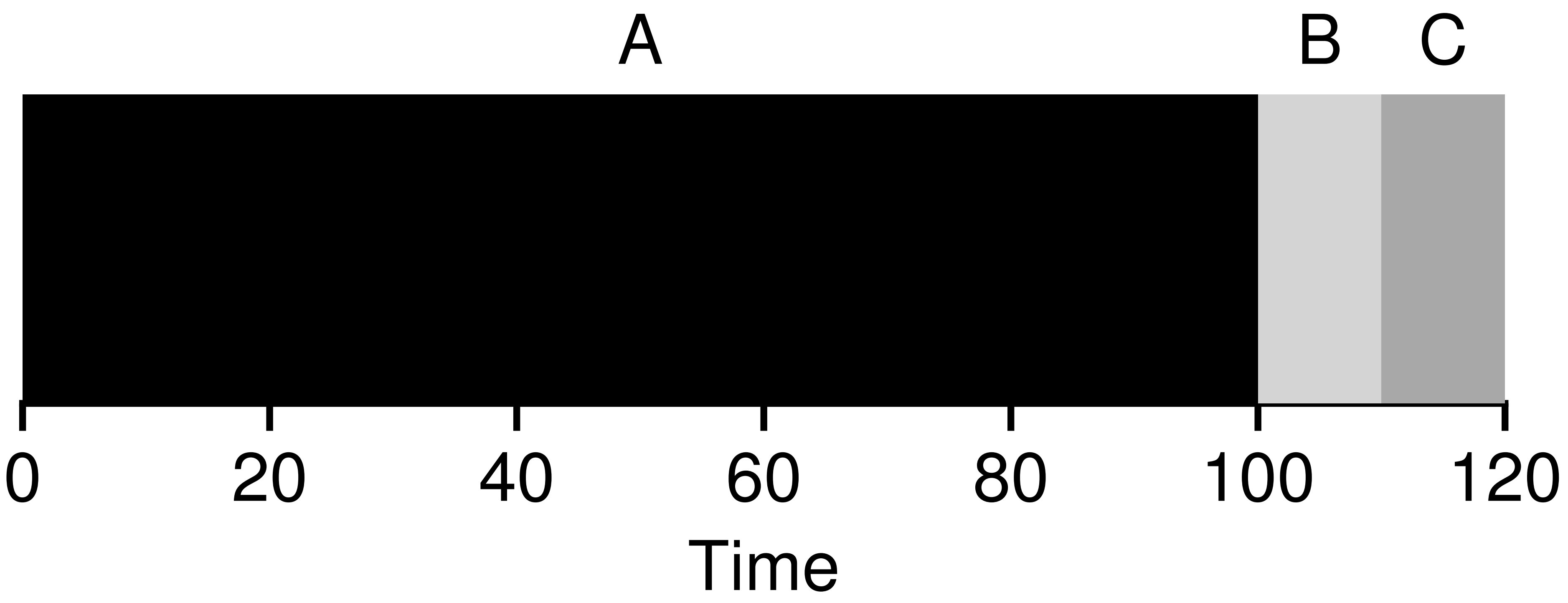
First-Come-First-Served (FCFS) example



What’s the average turnaround time?

Relax assumption 1: jobs take the same amount of time

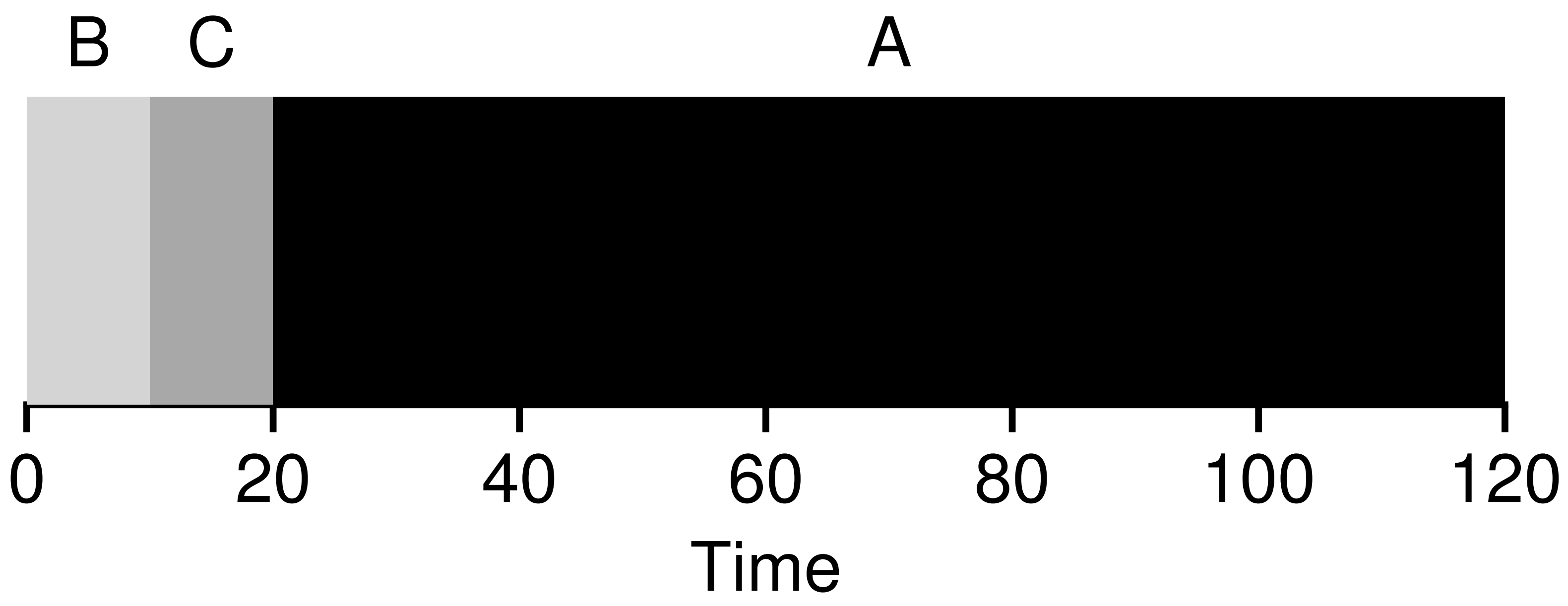
FCFS: The convoy effect



What's the average turnaround time?

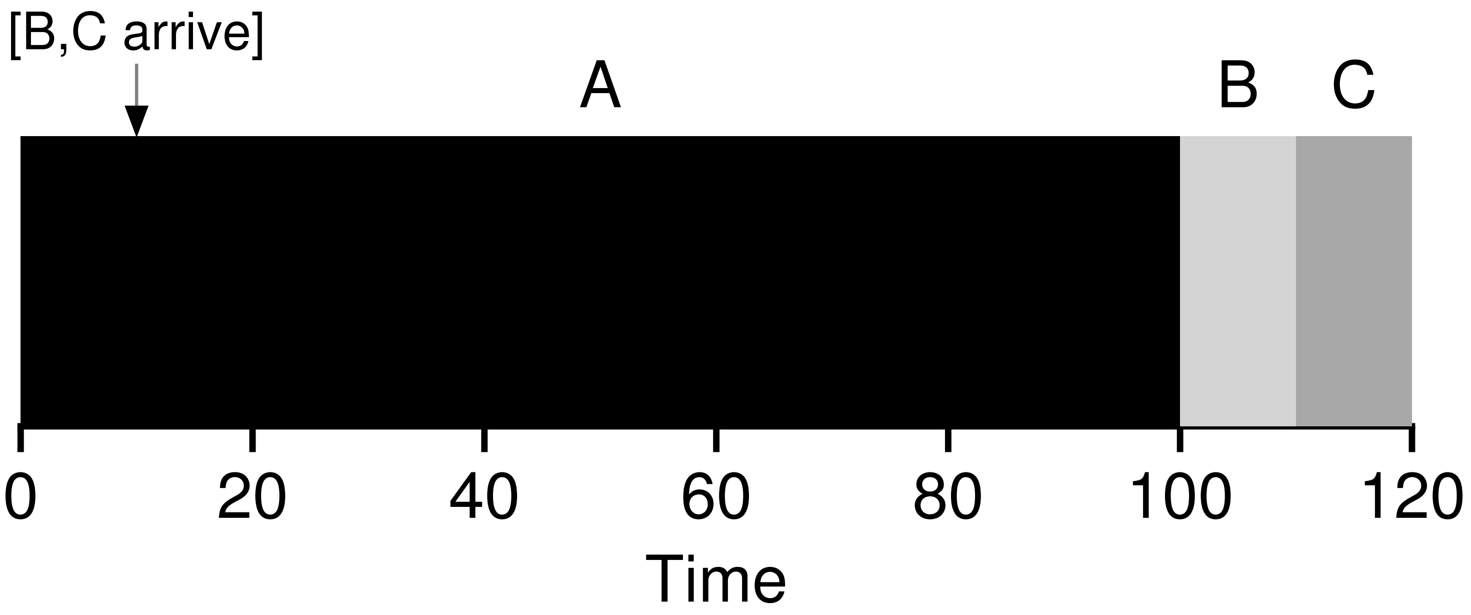
Shortest Job First

SJF: an optimal scheduling policy



What's the average turnaround time?

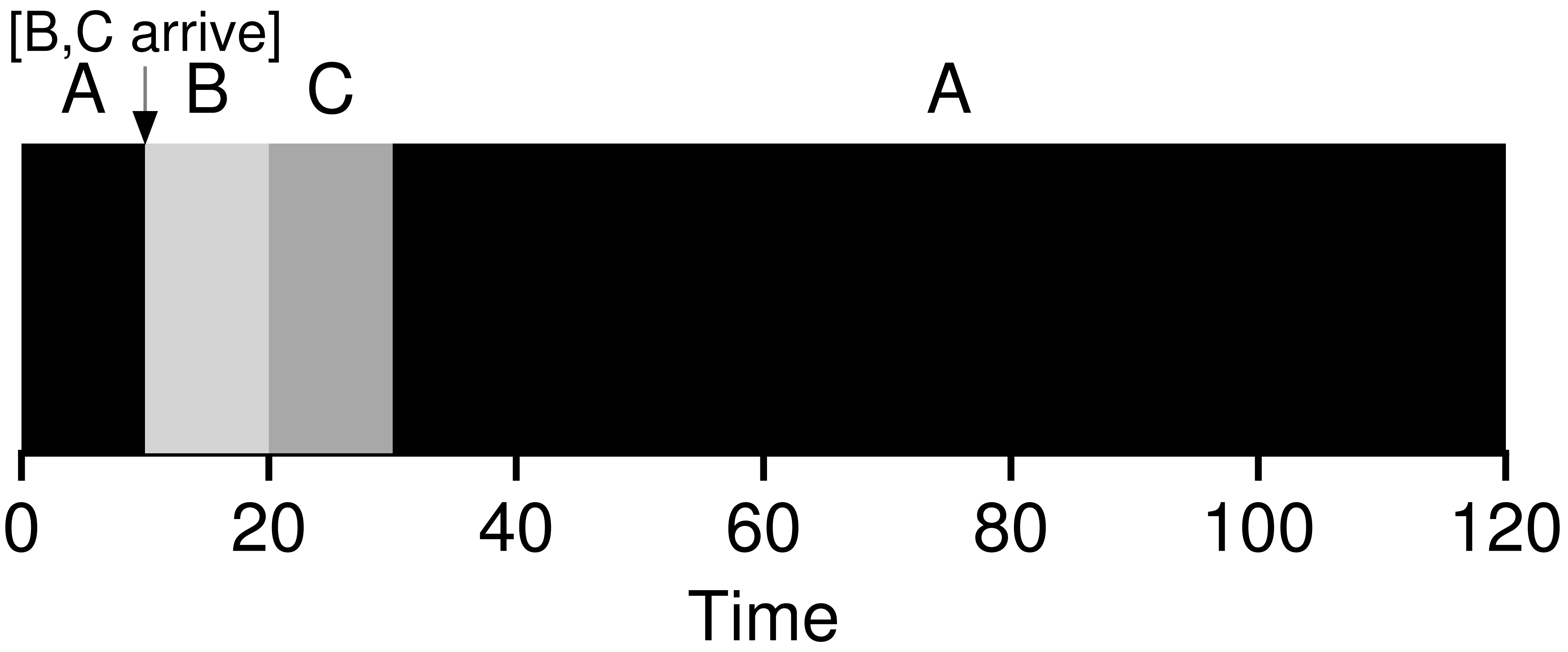
Relax assumption 2: jobs can now arrive at any time



What's the average turnaround time?

Shortest Time-to-Completion First (or Preemptive Shortest Job First)

STCF: optimal policy with different job arrival times



Round-Robin Scheduling

Round-Robin Scheduling

Enable interactivity by limiting the amount of time a thread can run at a time

Time slice: amount of time the scheduler gives a thread before choosing another thread

- Requires timer interrupts

Effectiveness of Round-Robin Scheduling

The number of jobs

More jobs -> slower response times

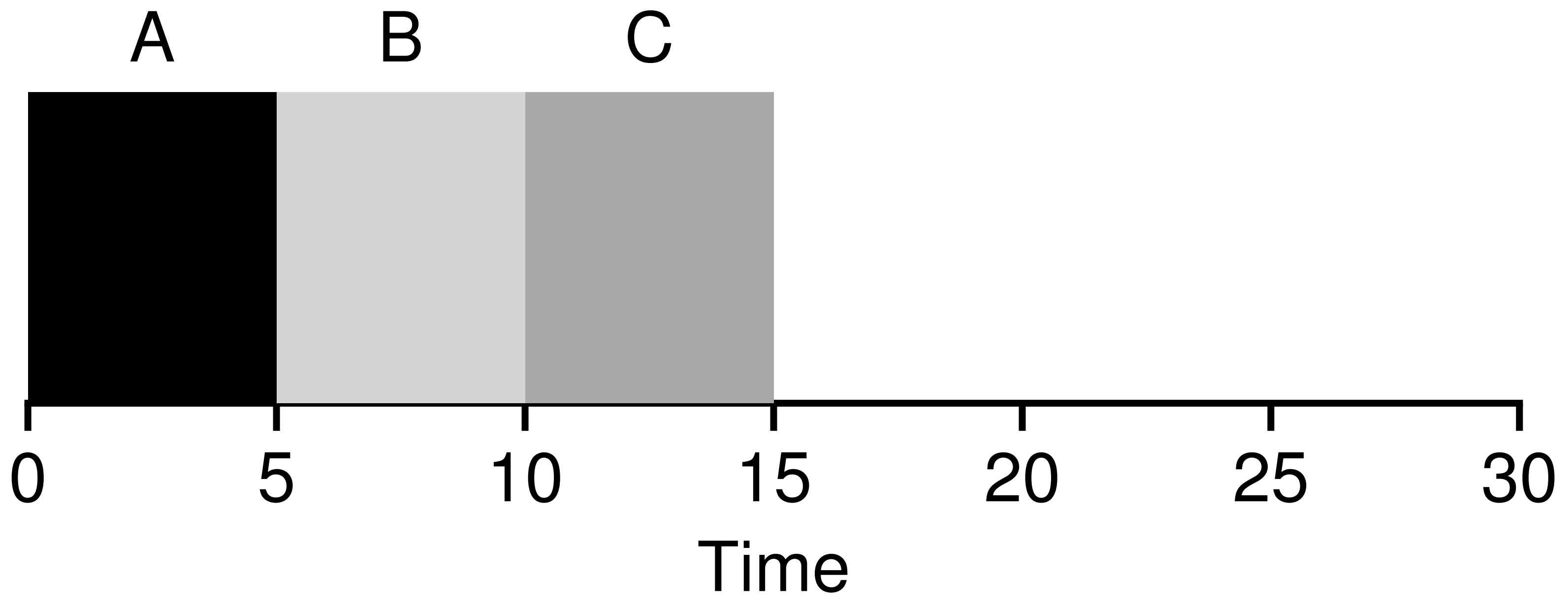
The length of the time slice (scheduling quantum)

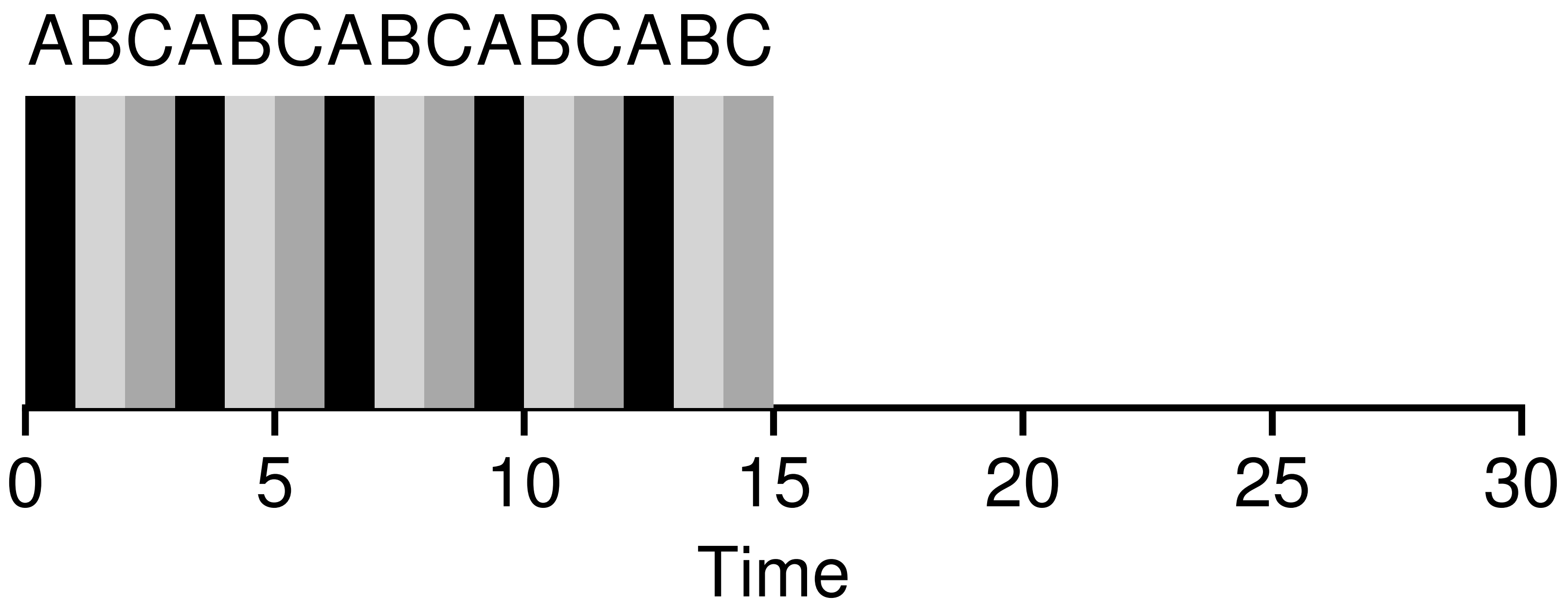
Longer time slice -> slower response times

Shorter time slice -> more overhead

10 ms to 100 ms is often a reasonable compromise

Round-Robin: Example





What we've covered so far

Three Easy Pieces, Chapter 7 (Scheduling: Introduction)