Process

Saptarshi Ghosh and Mainack Mondal

CS31202 / CS30002

This lecture

- What is a process?
- Structure of a process
- Process states
- Process control block
- Context switch

What is a process?

- A process is a program in execution
  - Recall multitasking
  - Several processes may be in various stages of execution at same time

What is a process?

- A process is a program in execution
  - Recall multitasking
  - Several processes may be in various stages of execution at same time

- CPU switches rapidly between several processes
  - We say that processes are executing concurrently
  - CPU multiplexed

Program vs. Process

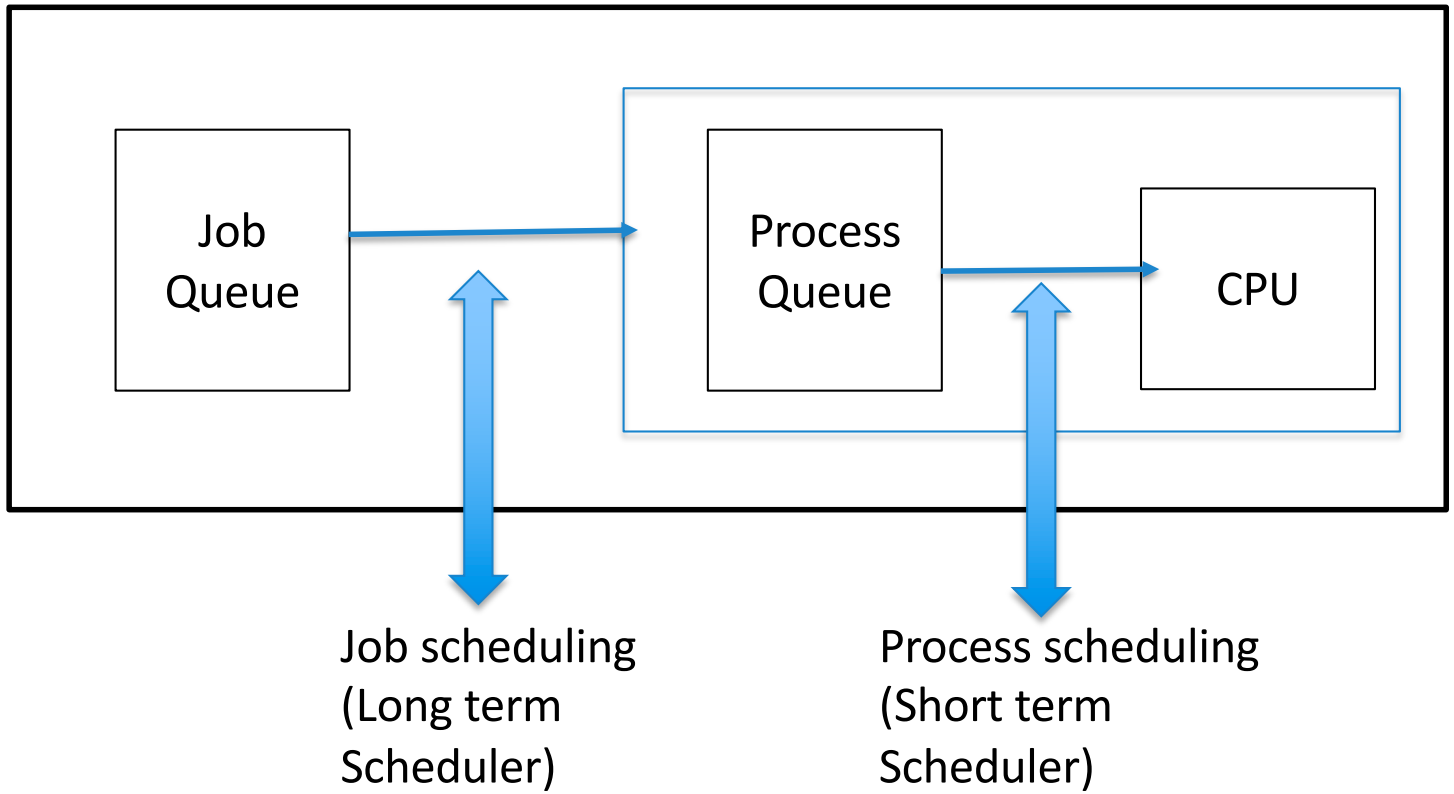
- Program is a passive entity
- Process is an active entity
  - Program becomes process when the code is loaded in the memory and ready to execute
- Each execution instance of the same program is a separate process

Job scheduling and process scheduling

- Job scheduling
  - In batch processing or multiprogramming, user programs are called jobs
  - Long term scheduler – loaded jobs into memory

- Process scheduling
  - Short term scheduler – allocated CPU to jobs

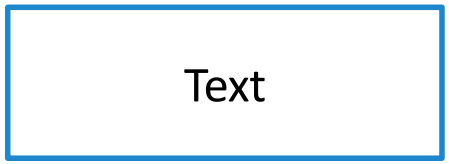
Job scheduling and process scheduling



This lecture

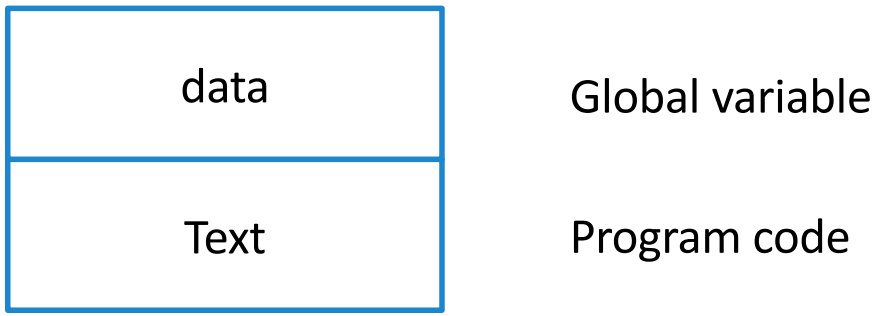
- What is a process?
- Structure of a process
- Process states
- Process control block
- Context switch

What are parts of a process (in memory)?

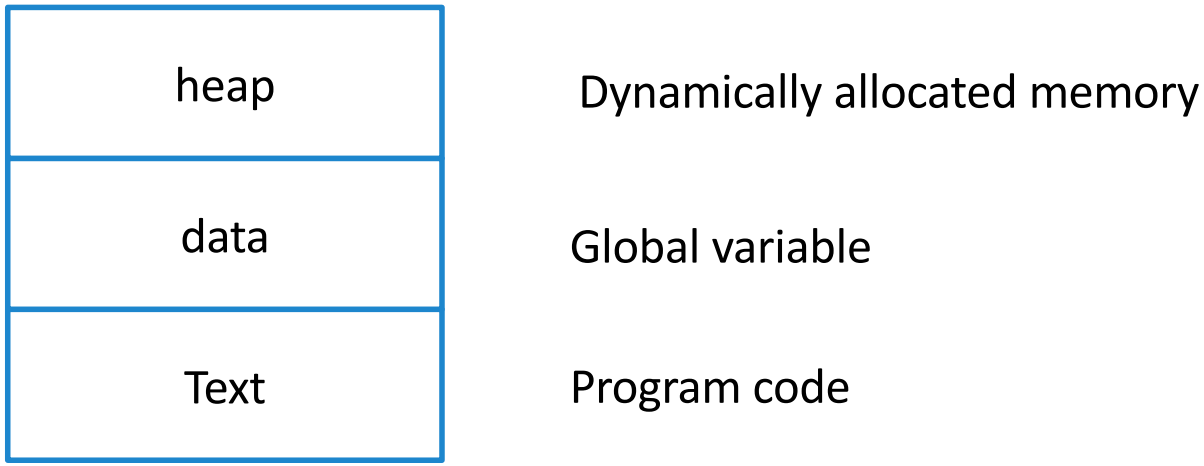


Program code

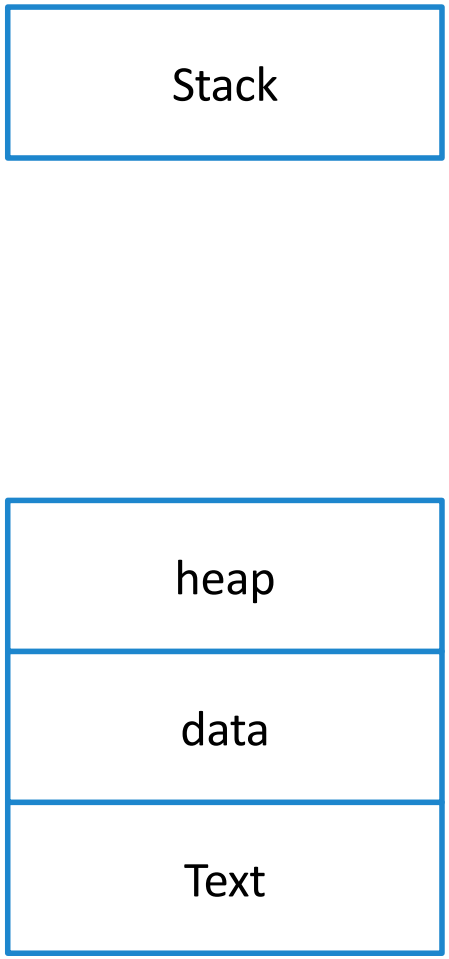
What are parts of a process?



What are parts of a process?



What are parts of a process?



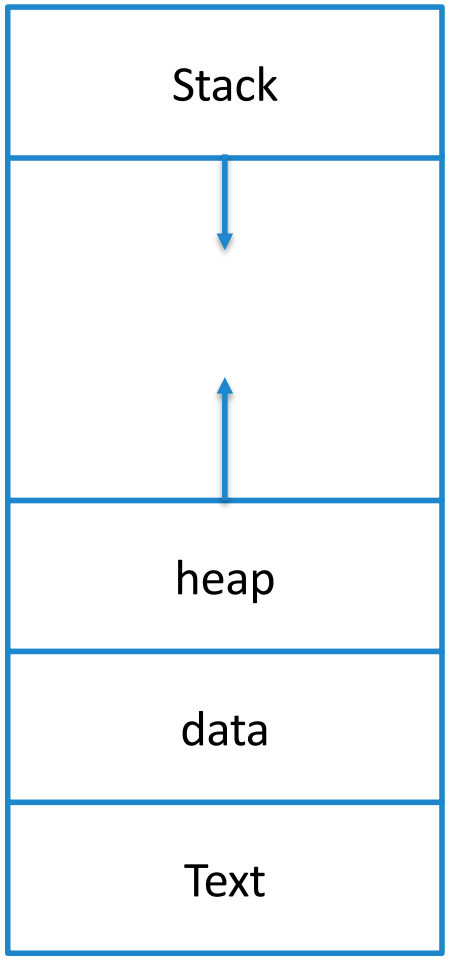
Temporary data,
function parameters,
local variables, return addresses

Dynamically allocated memory

Global variable

Program code

What are parts of a process?



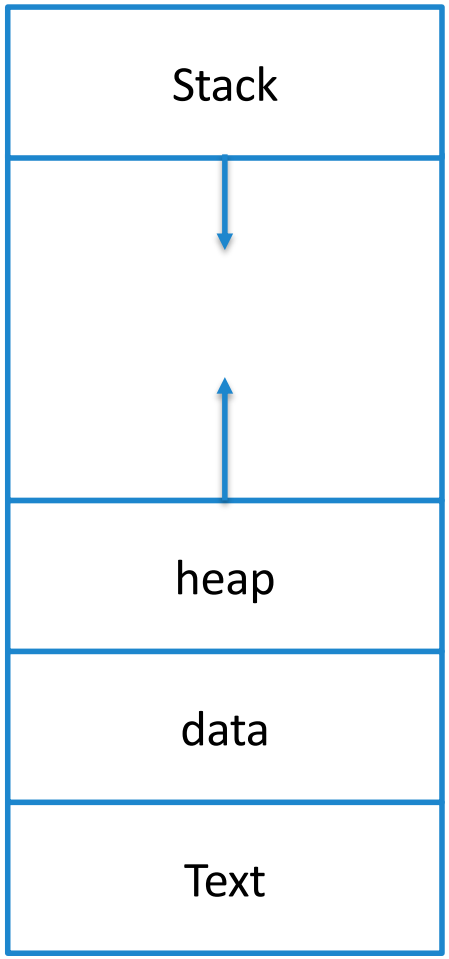
Temporary data,
function parameters,
local variables, return addresses

Dynamically allocated memory

Global variable

Program code

What are parts of a process?



Temporary data,
function parameters,
local variables, return addresses

Dynamically allocated memory

Global variable

Program code

Also
Program
counter (PC),
CPU registers,
open files

This lecture

- What is a process?
- Structure of a process
- Process states
- Process control block
- Context switch

A process has different states

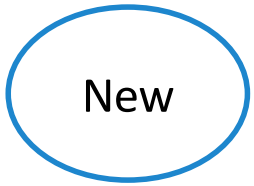
- At a particular point of time, a process may be in one of several states
- Processes change state based on certain events
- You can think of process execution as an automata

States of a process

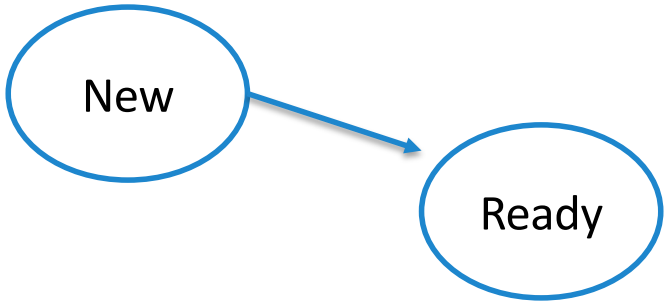
- States
  - **new:** The process is being created
  - **ready:** The process is waiting to be assigned to a processor
  - **running:** Instructions are being executed on the CPU
  - **waiting:** The process is waiting for some event to occur
  - **terminated:** The process has finished execution

- Note: there is a separate area in the disk: “swap space”
  - Swap-out and swap-in between main memory and disk

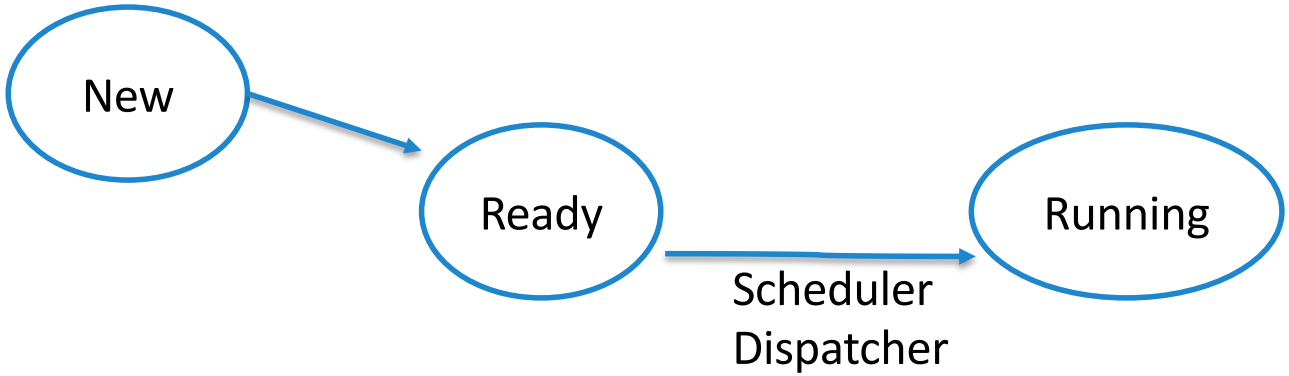
Process state diagram



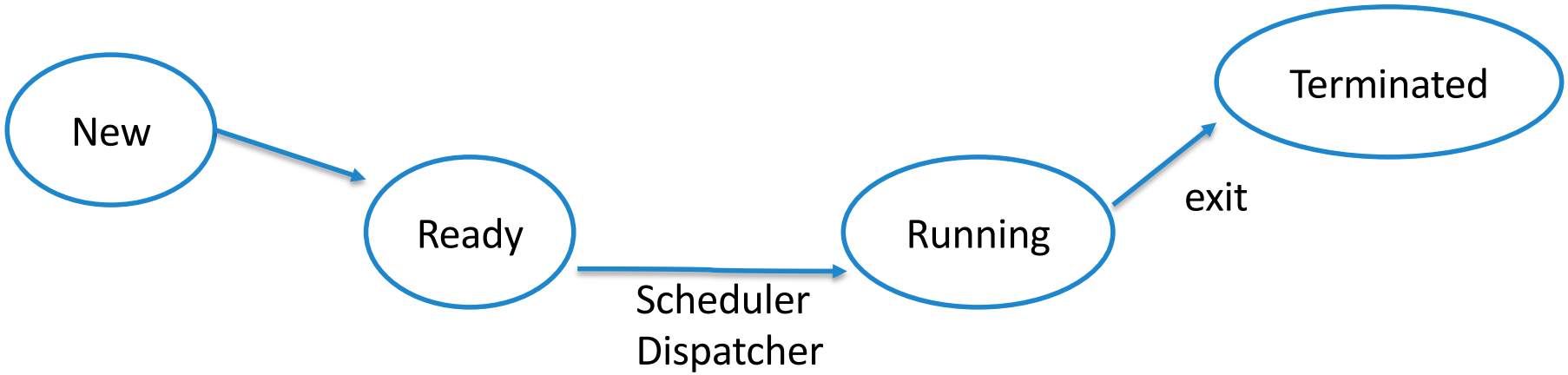
Process state diagram



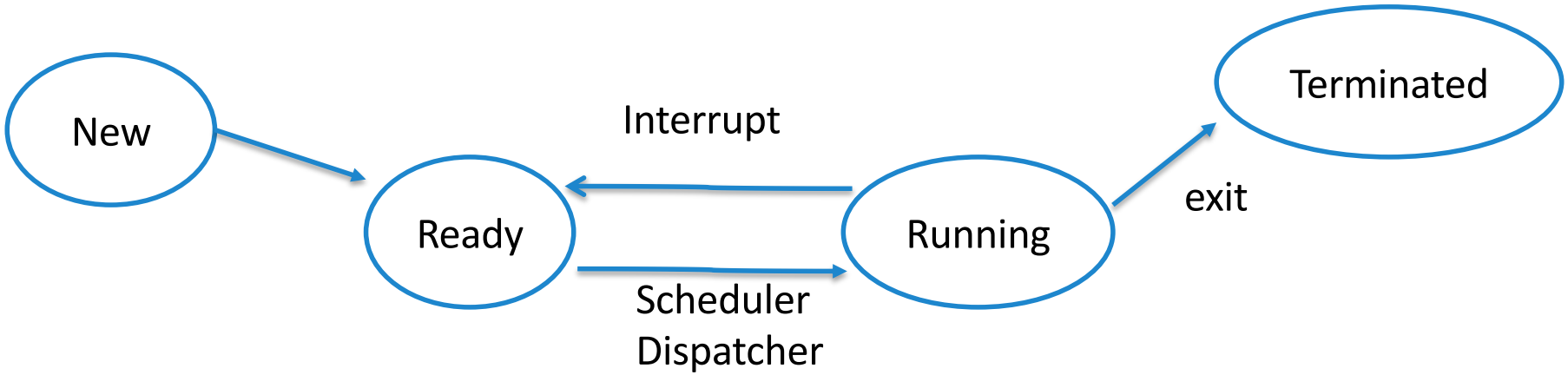
Process state diagram



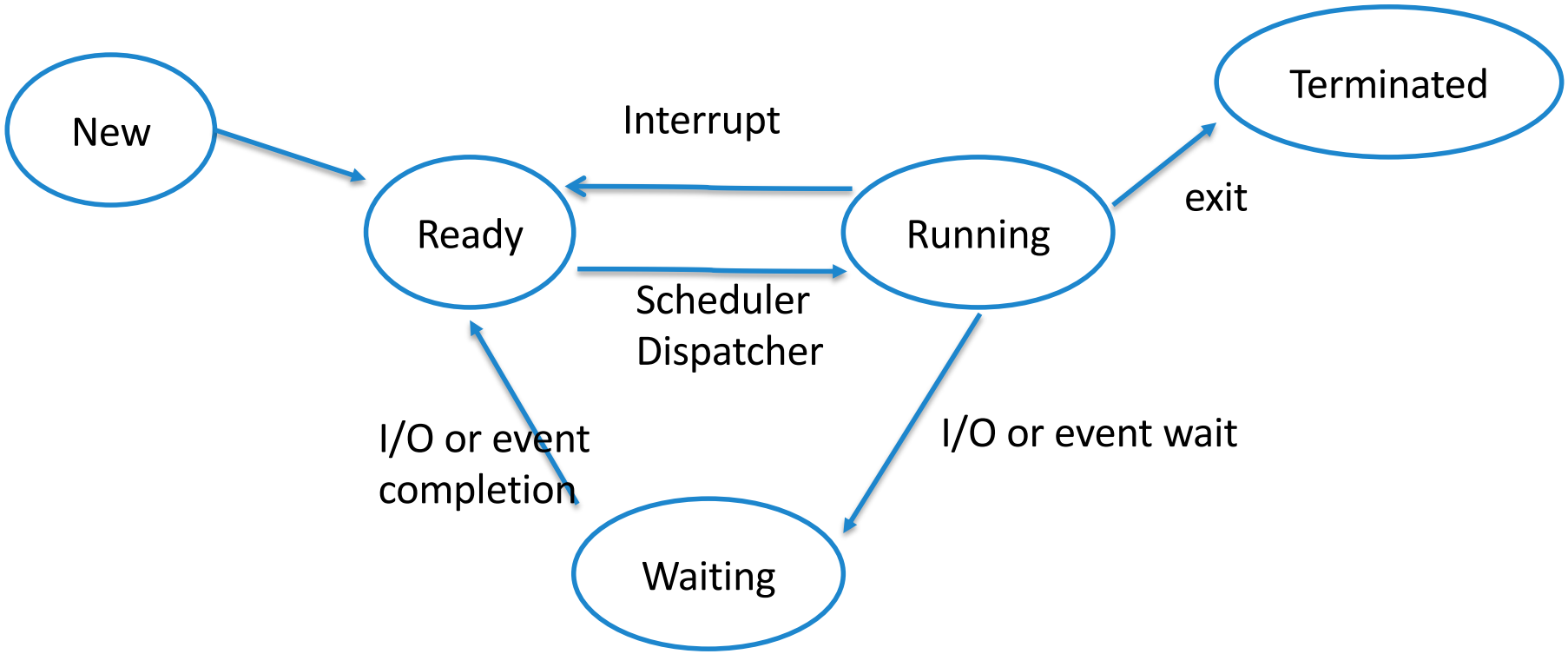
Process state diagram



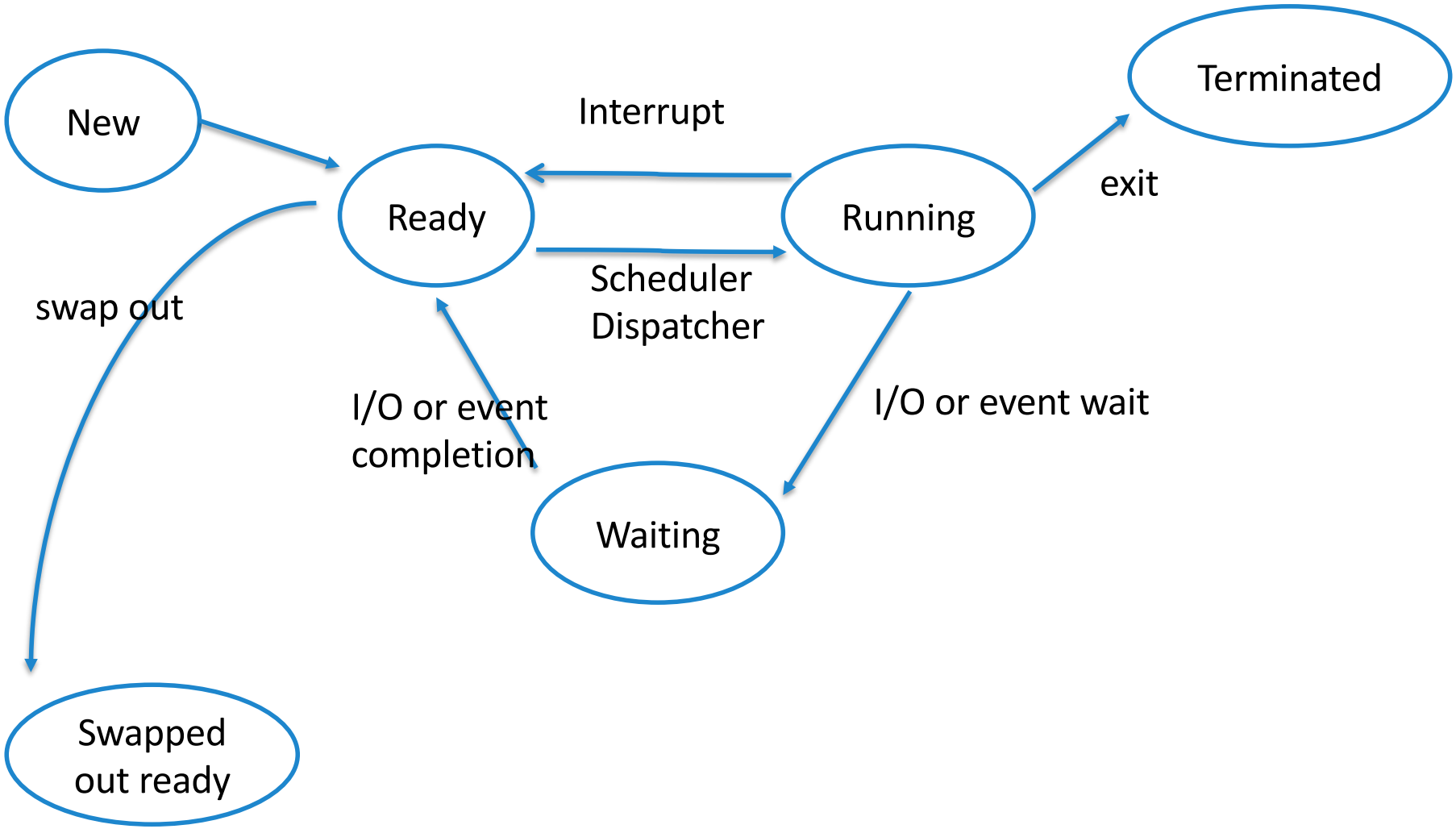
Process state diagram



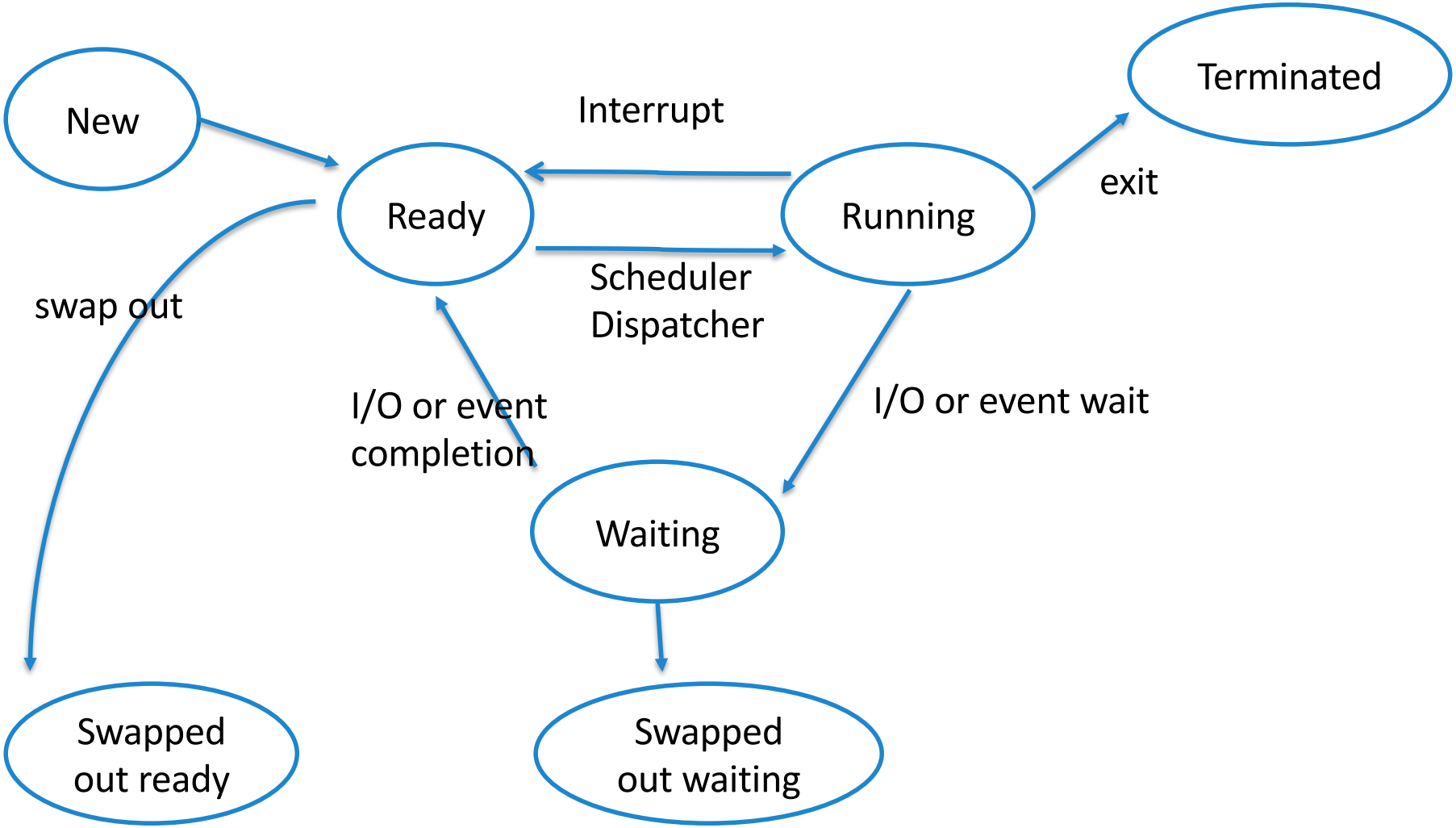
Process state diagram



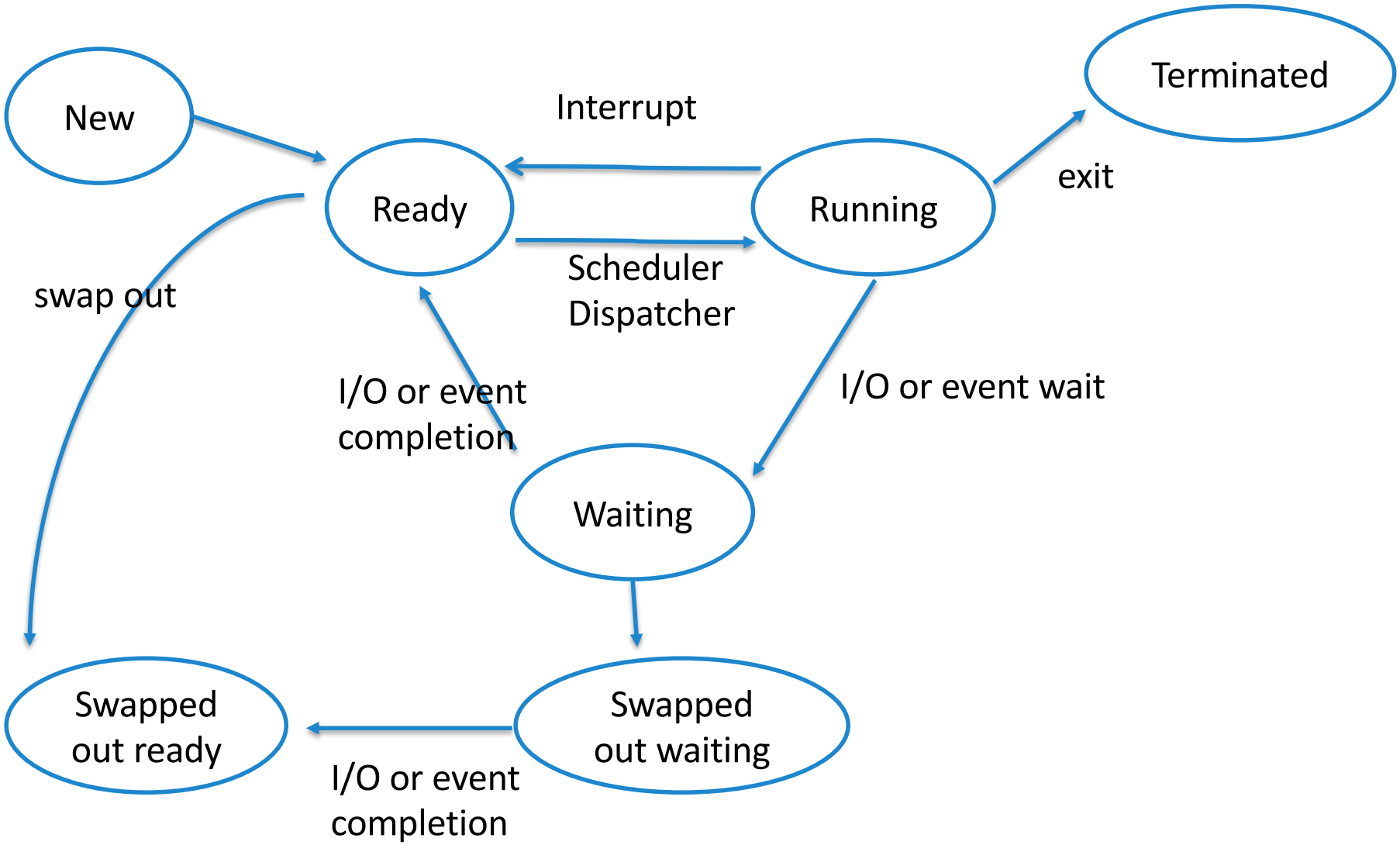
Process state diagram



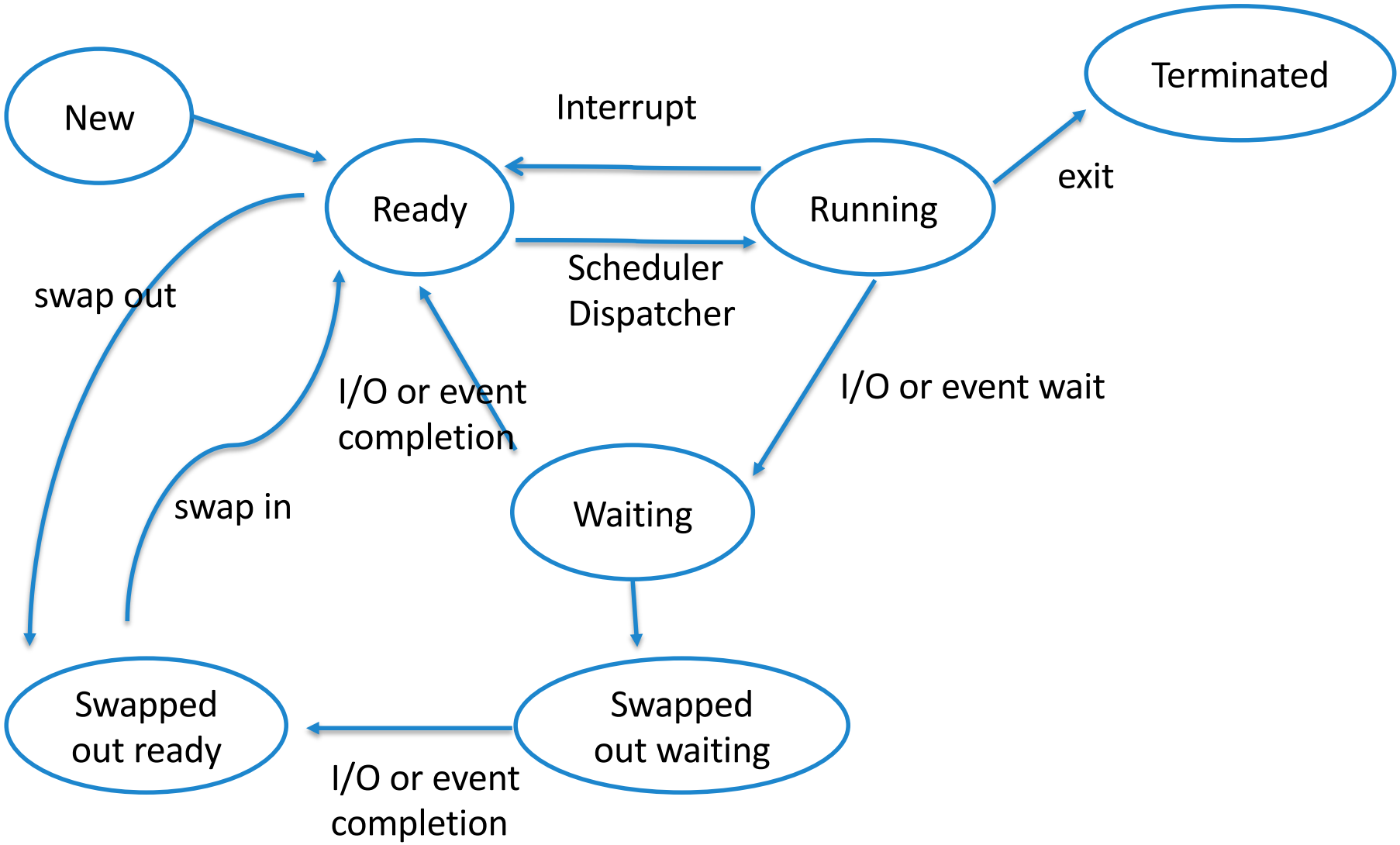
Process state diagram



Process state diagram



Process state diagram



This lecture

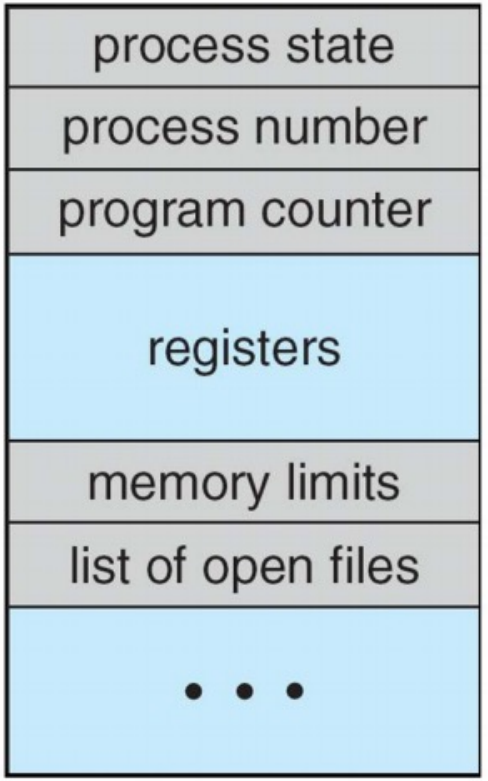
- What is a process?
- Structure of a process
- Process states
- Process control block
- Context switch

Process control block (PCB)

- Each process is represented in the kernel as a PCB
  - Also called task control block
  - Contains many pieces of information associated with a specific process

Structure of a PCB

- Process state – running, waiting, etc
- Program counter (PC) – location of instruction to next execute
- Content of CPU registers
- CPU scheduling information- priorities, scheduling queue pointers
- Memory-management information –Base and limit registers, page tables
- Accounting information – CPU used, clock time elapsed since start, time limits, pid
- I/O status information – I/O devices, allocated to process, list of open files



Process representation in Linux

Represented by the C structure task_struct

```
pid t pid;    // process identifier

long state;    // state of the process

unsigned int time slice      // scheduling information

struct task struct *parent;   // this process's parent

struct list head children;    // this process's children

struct files struct *files;    // list of open files

struct mm_struct *mm;    // address space of this pro
```

This lecture

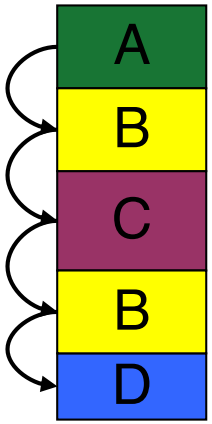
- What is a process?
- Structure of a process
- Process states
- Process control block
- Context switch

CPU runs multiple processes

- Multiprogramming of four programs
- Conceptual model
  - 4 independent processes
  - Processes run sequentially
- Only one program active at any instant!
  - That instant can be very short…

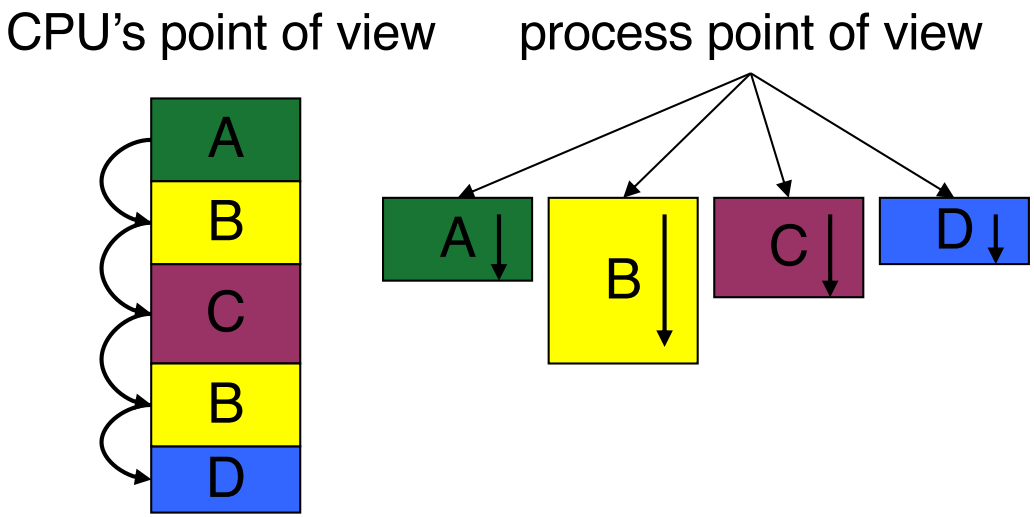
CPU runs multiple processes

CPU's point of view



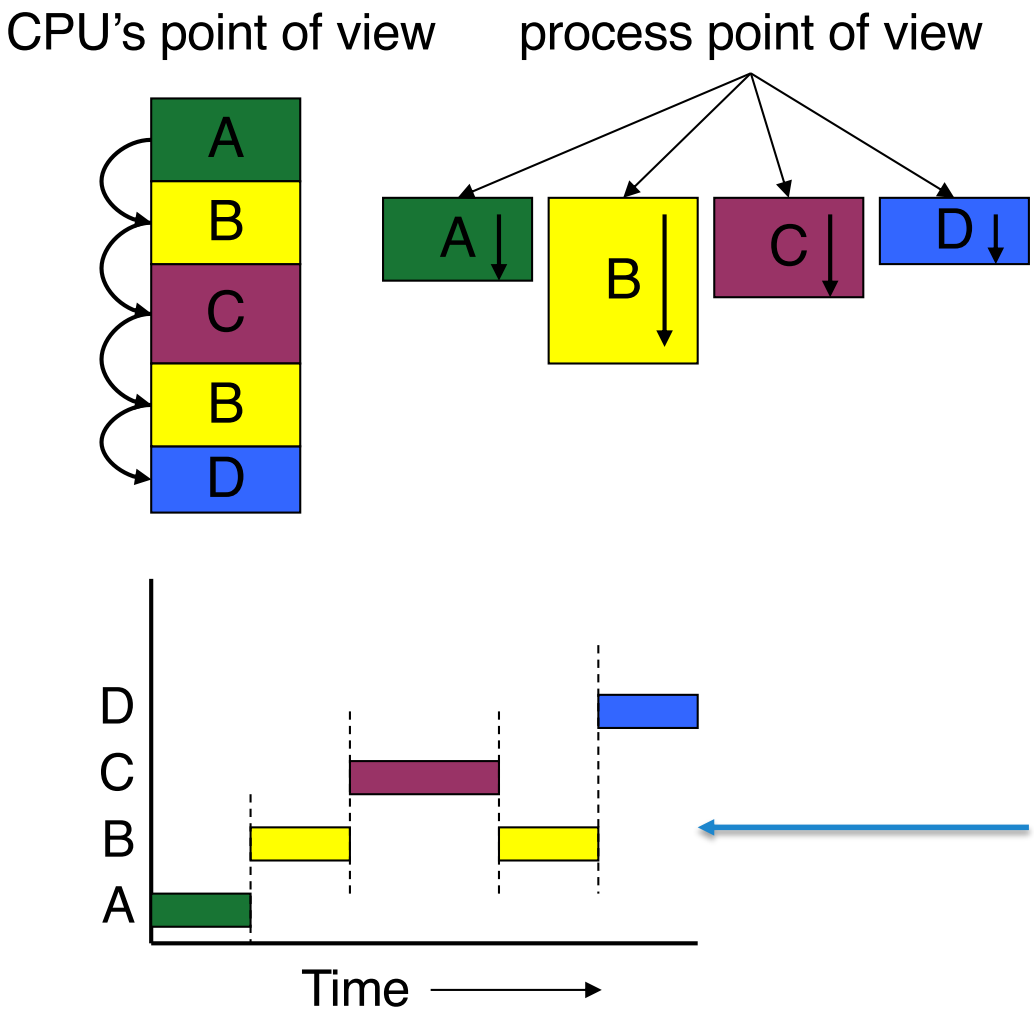
- Multiprogramming of four programs
- Conceptual model
  - 4 independent processes
  - Processes run sequentially
- Only one program active at any instant!
  - That instant can be very short…

CPU runs multiple processes



- Multiprogramming of four programs
- Conceptual model
  - 4 independent processes
  - Processes run sequentially
- Only one program active at any instant!
  - That instant can be very short…

CPU runs multiple processes



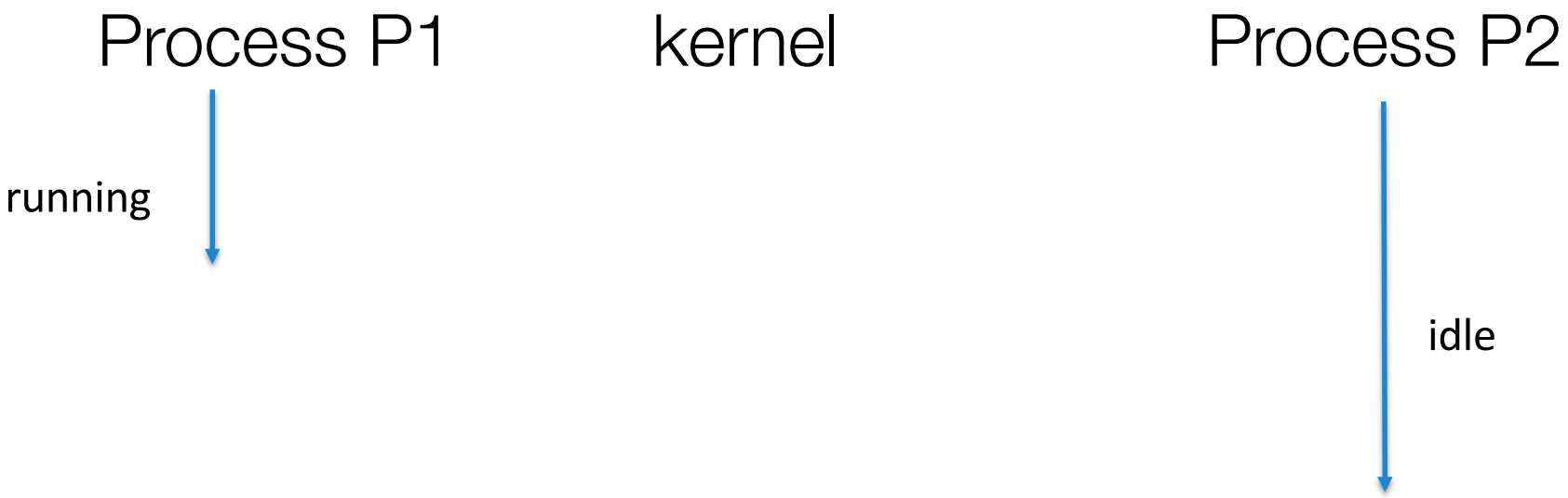
- Multiprogramming of four programs
- Conceptual model
  - 4 independent processes
  - Processes run sequentially
- Only one program active at any instant!
  - That instant can be very short…

Gantt chart for multiprogramming

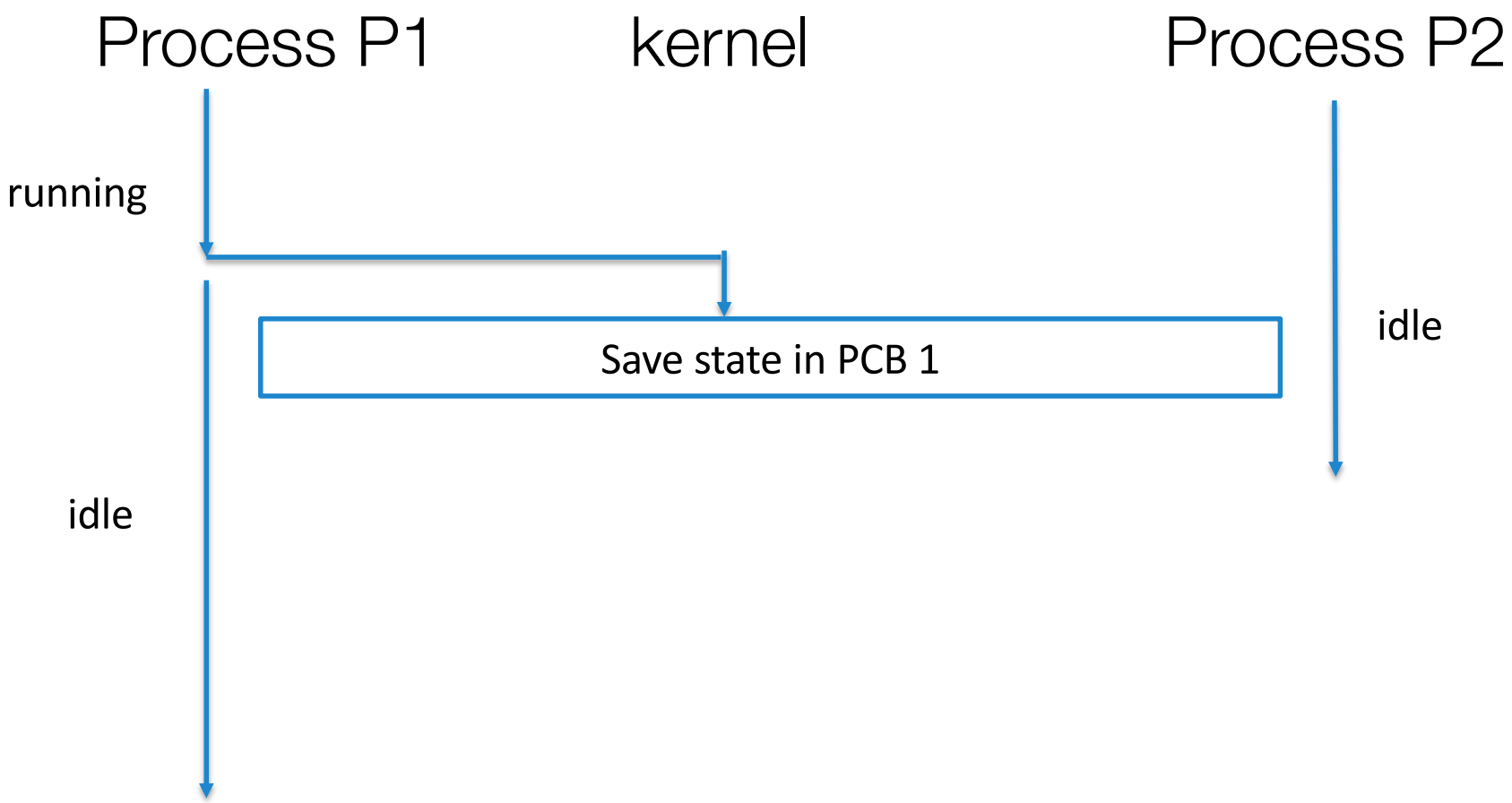
How to interleave processes?

- CPU switches to another process
  - The kernel saves the state of the old process and loads the saved state for the new process via a context switch
  - Context of a process == PCB
  - More complex the OS and PCB, longer to switch

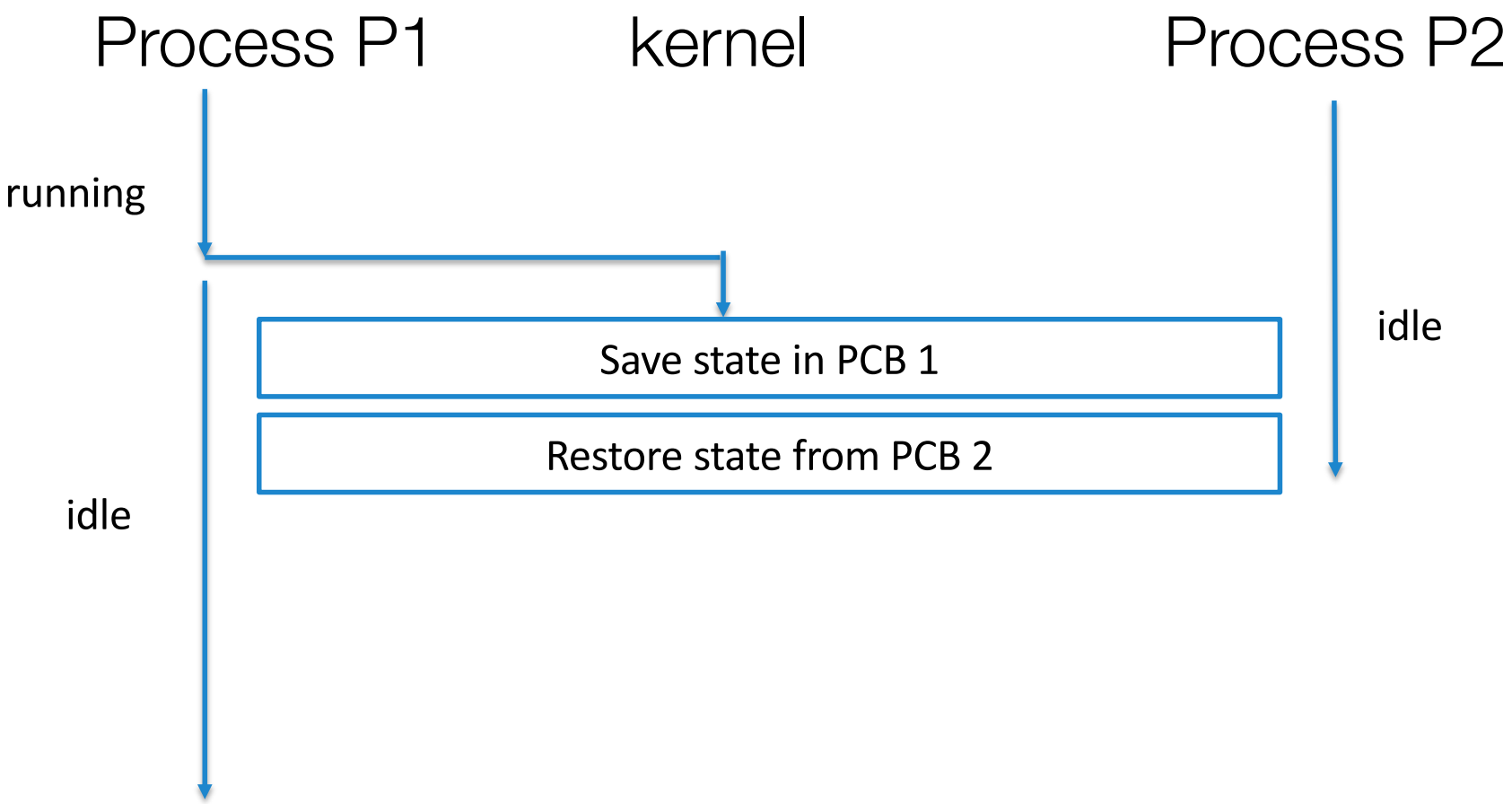
Context switch



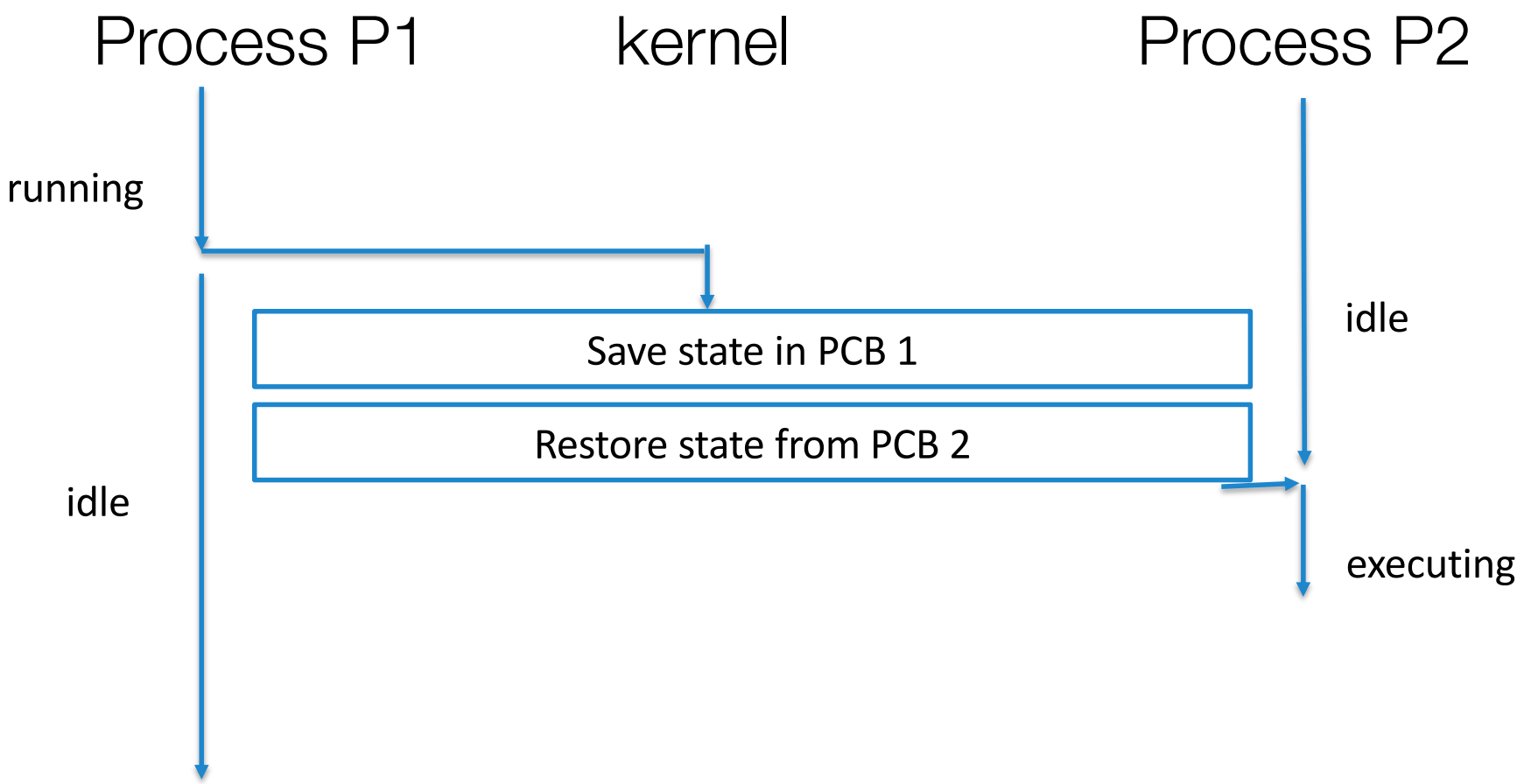
Context switch



Context switch



Context switch



Context switch