Process

Indranil Sen Gupta (odd section)
and Mainack Mondal (even section)
CS39002

Spring 2019-20

Today's class

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

Job scheduling and process scheduling

- Job scheduling
  - In batch processing or multiprogramming user programs are called jobs
  - Long term scheduler

- Process scheduling
  - Short term scheduler

Job scheduling and process scheduling

- Job scheduling
  - In batch processing or multiprogramming user programs are called jobs
  - Long term scheduler

- Process scheduling
  - Short term scheduler

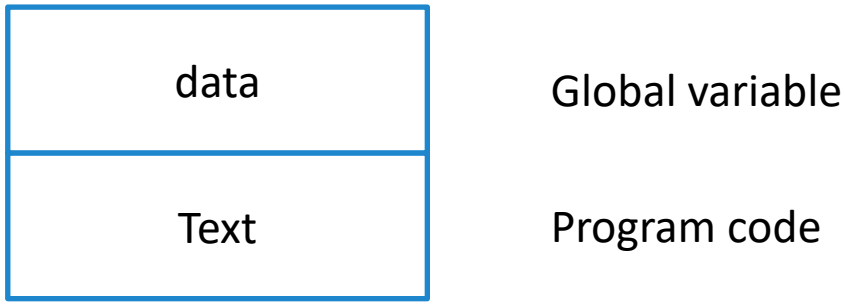
Check the board

Today's class

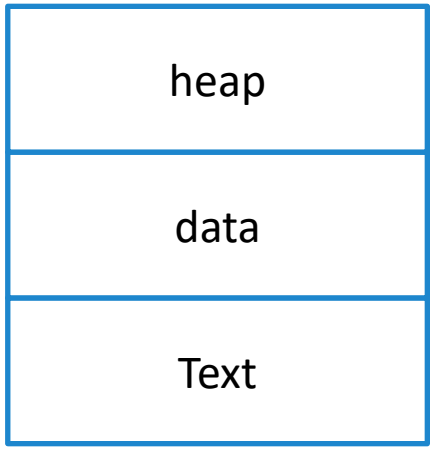
- What is a process?
- Structure of a process
- Process states
- Process control block
- Context switch

What are parts of a process?

What are parts of a process?



What are parts of a process?

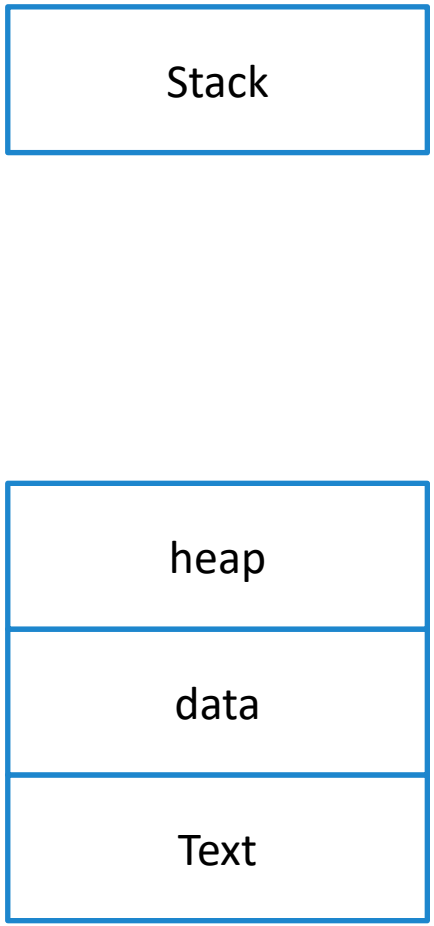


Dynamically allocated memory

Global variable

Program code

What are parts of a process?



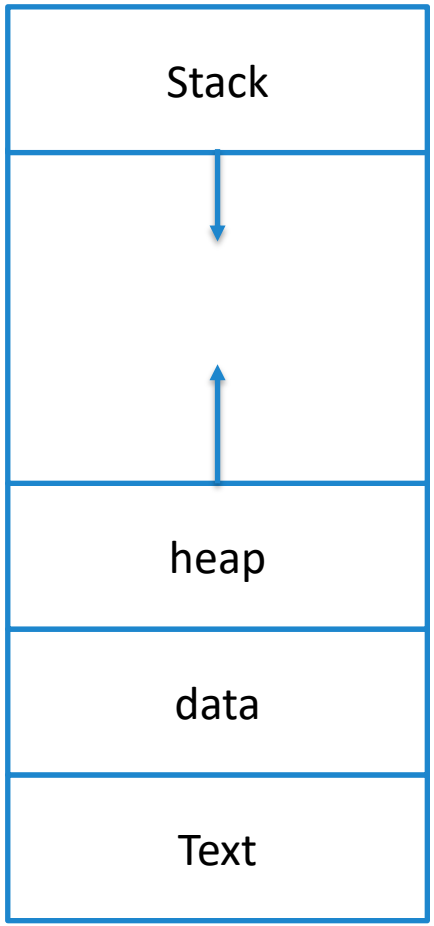
Temporary data,
function parameters,
local variables, return addresses

Dynamically allocated memory

Global variable

Program code

What are parts of a process?



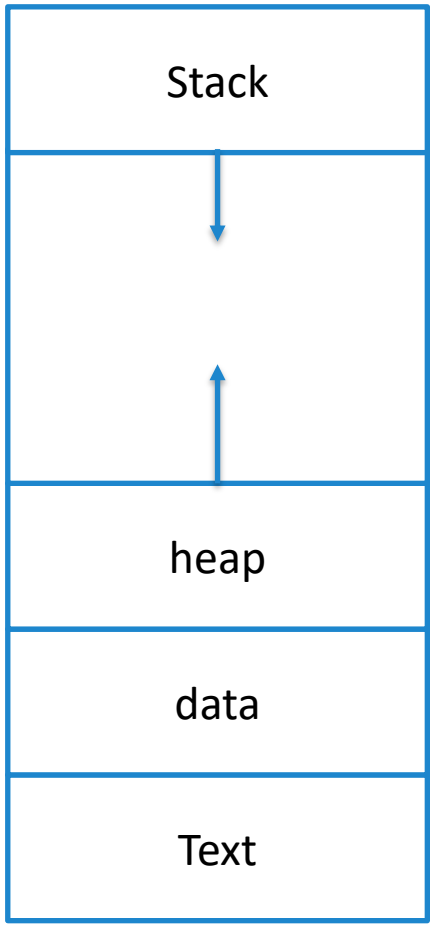
Temporary data,
function parameters,
local variables, return addresses

Dynamically allocated memory

Global variable

Program code

What are parts of a process?



Temporary data,
function parameters,
local variables, return addresses

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

Dynamically allocated memory

Global variable

Program code

Characteristics of a 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

Today's class

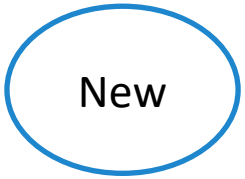
- What is a process?
- Structure of a process
- Process states
- Process control block
- Context switch

You can think process execution as an automata

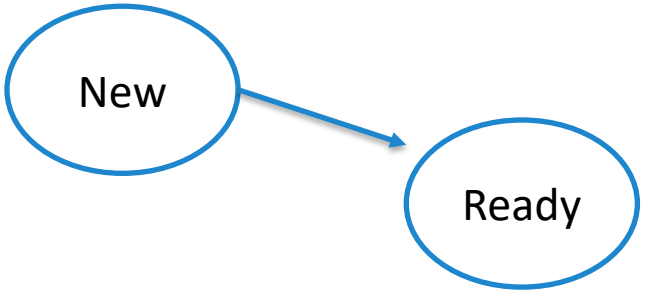
- States
  - **new:** The process is being created
  - **running:** Instructions are being executed
  - **waiting:** The process is waiting for some event to occur
  - **ready:** The process is waiting to be assigned to a processor
  - **terminated:** The process has finished execution

- Additionally there is “swap space”
  - Resides in the disk
  - Swap-out and swap-in between main memory and disk

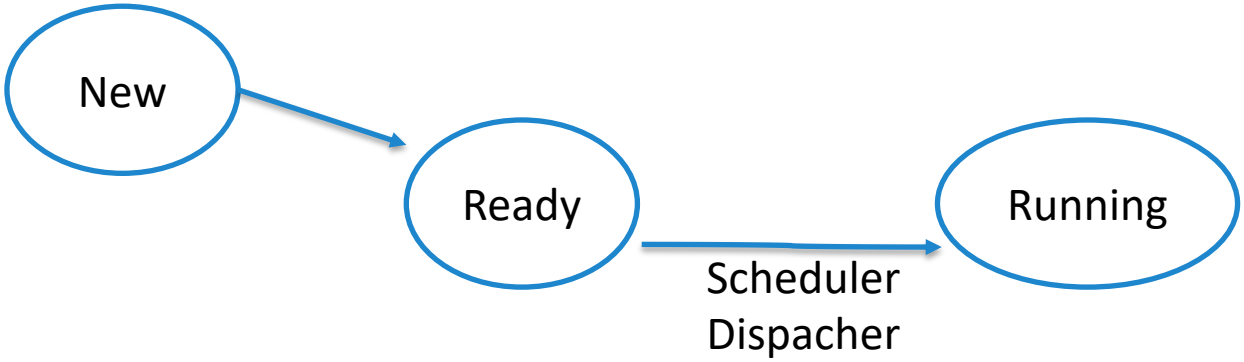
Process state diagram



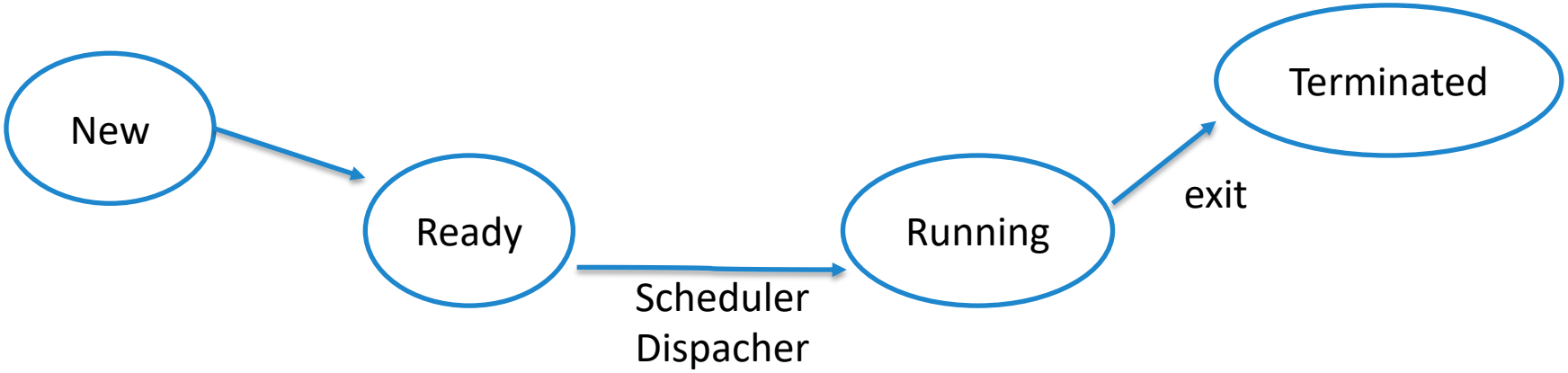
Process state diagram



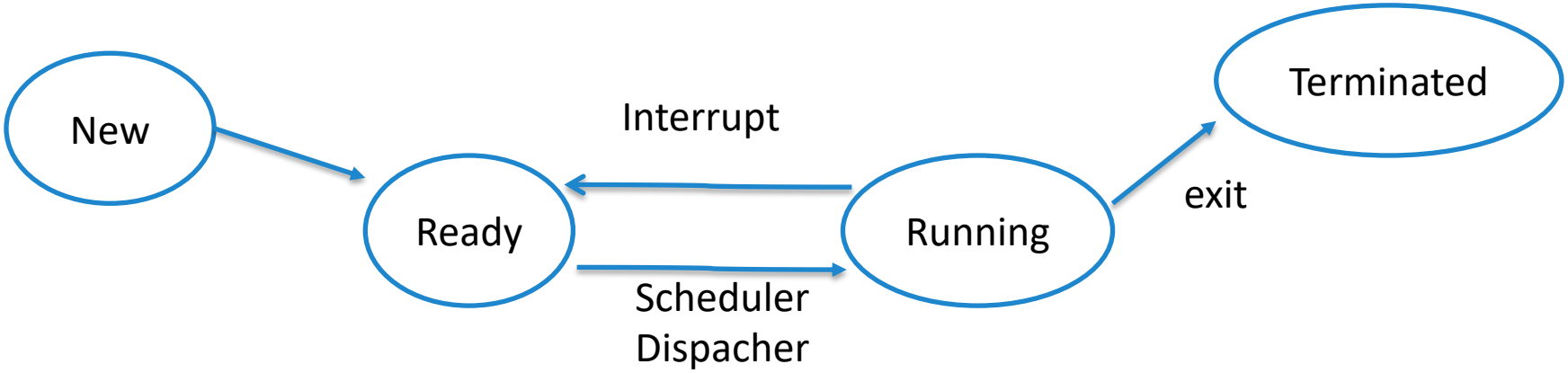
Process state diagram



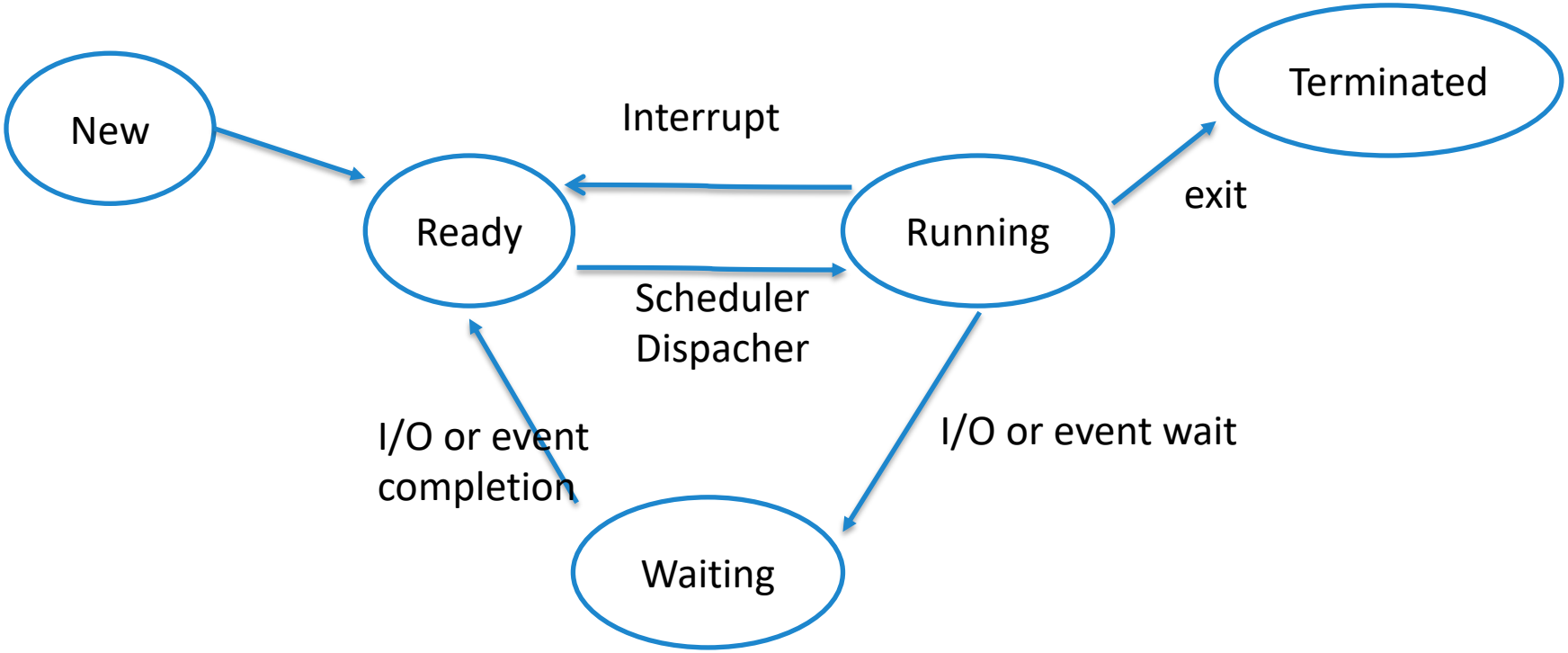
Process state diagram



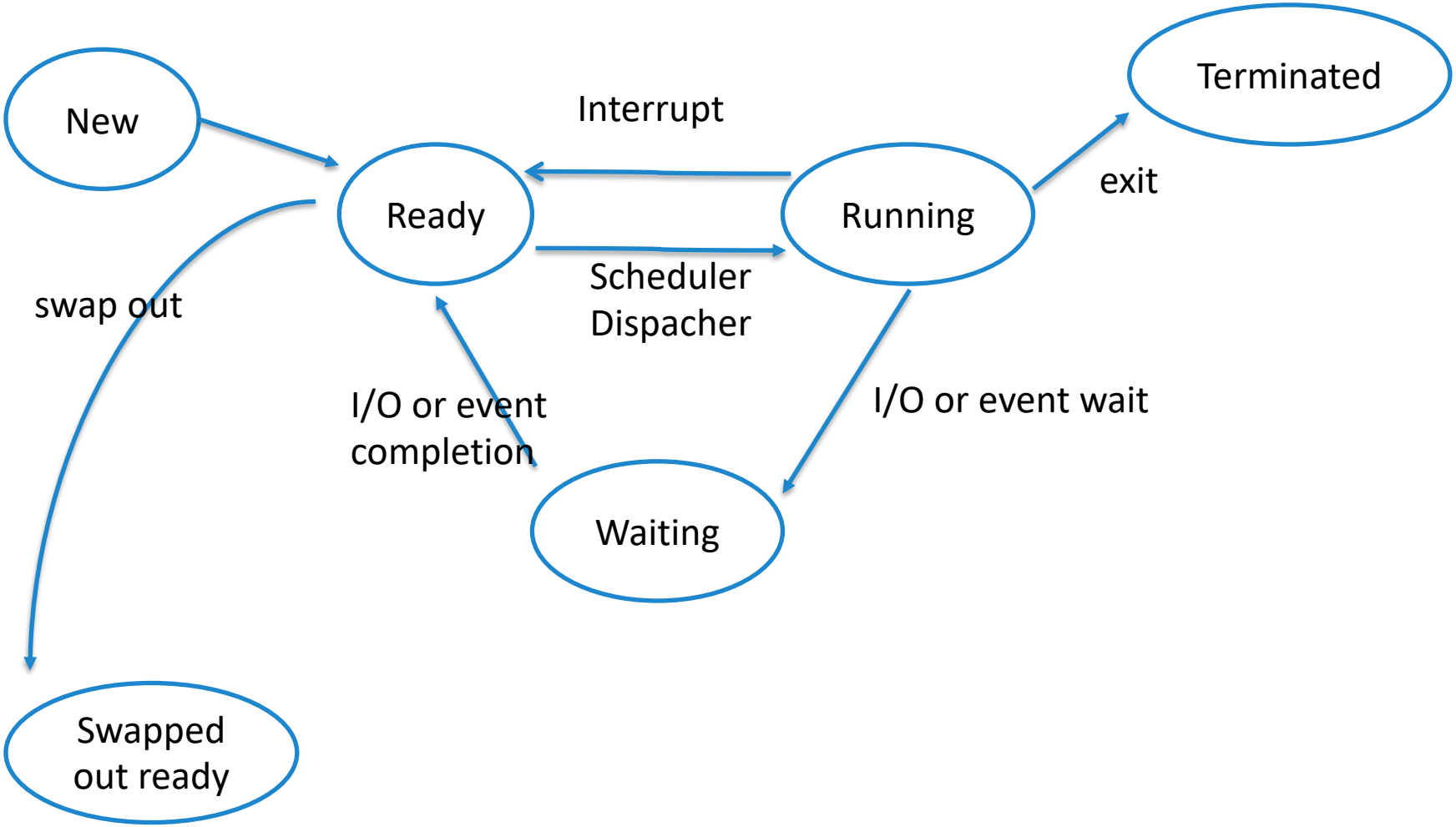
Process state diagram



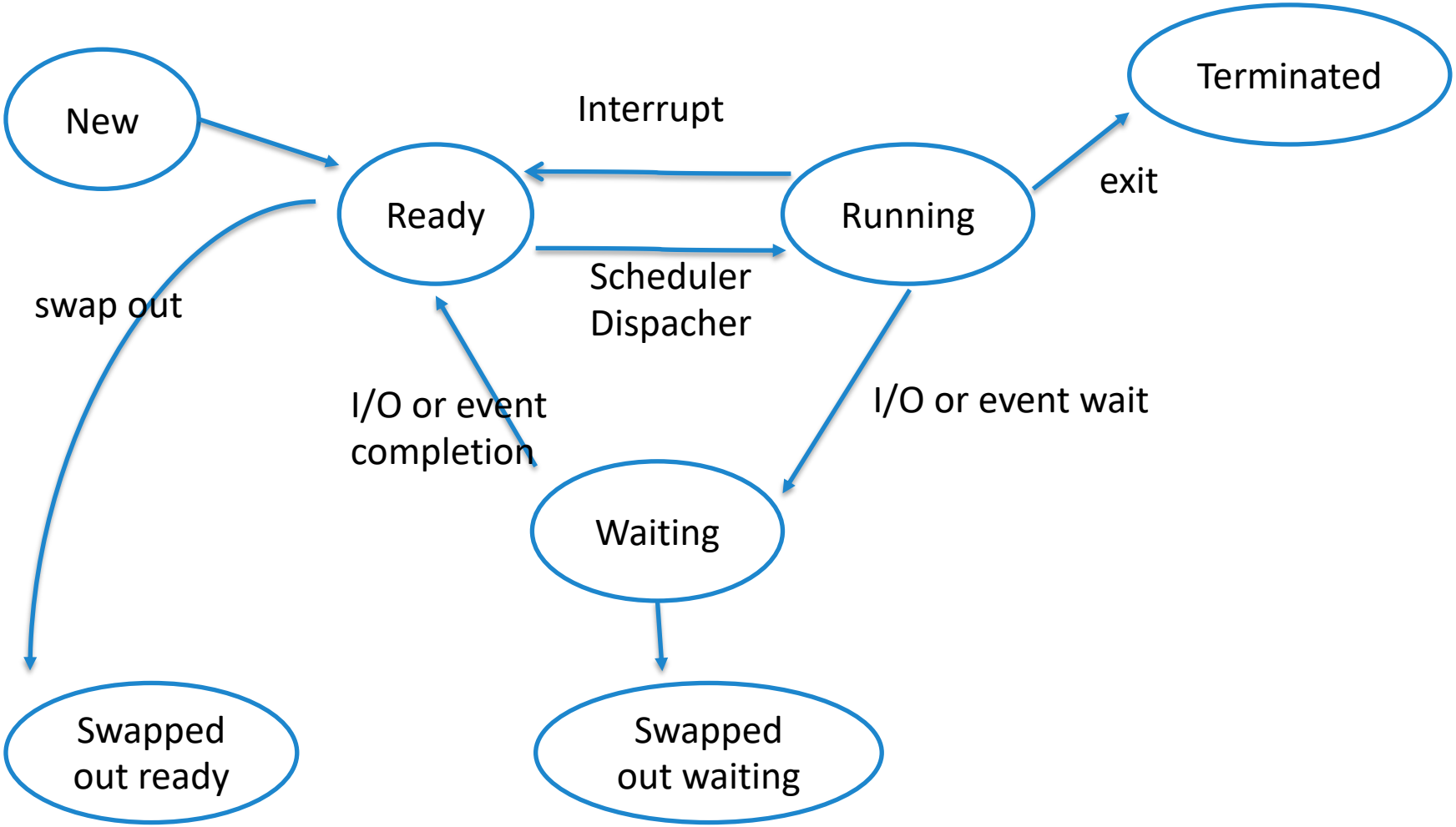
Process state diagram



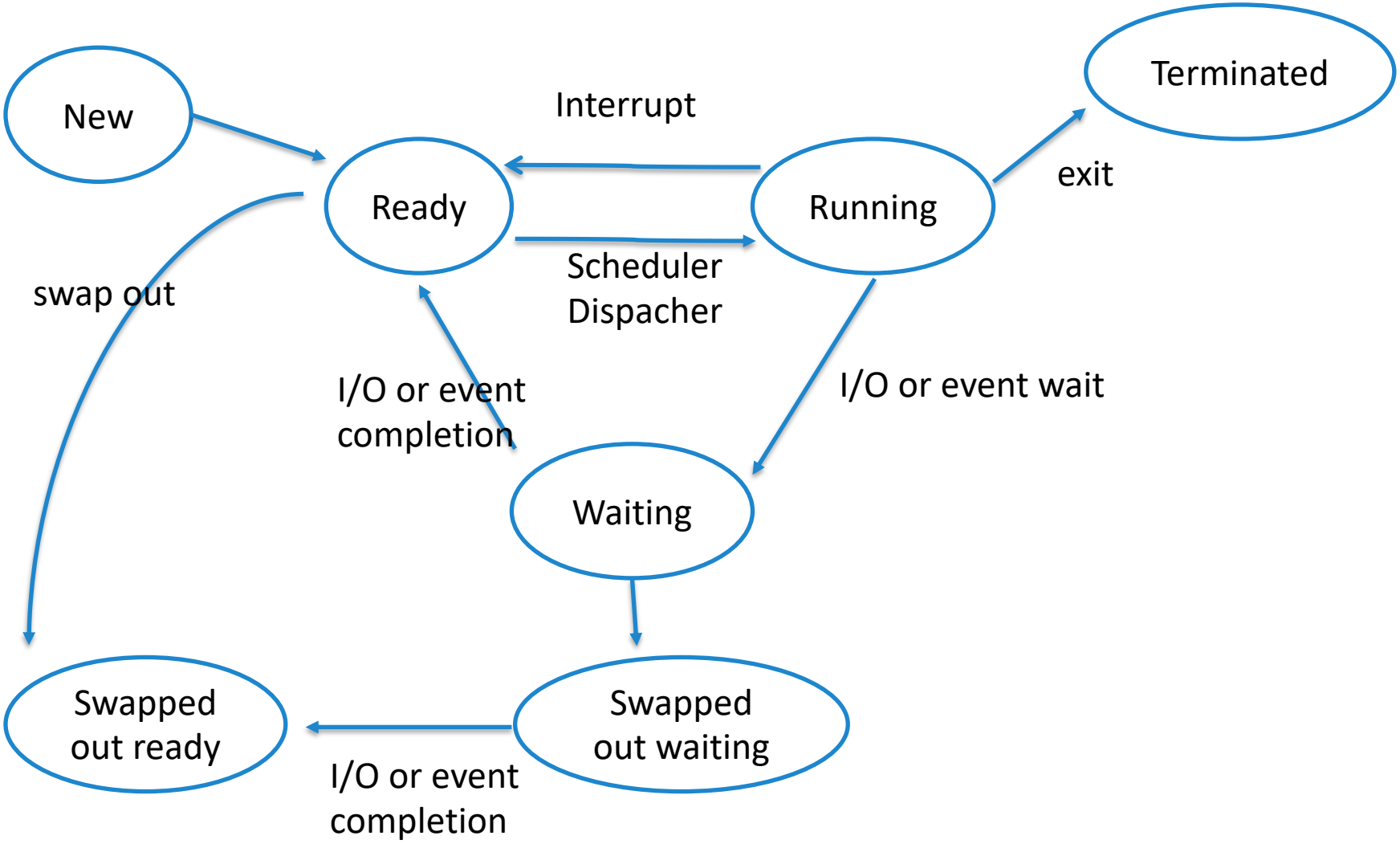
Process state diagram



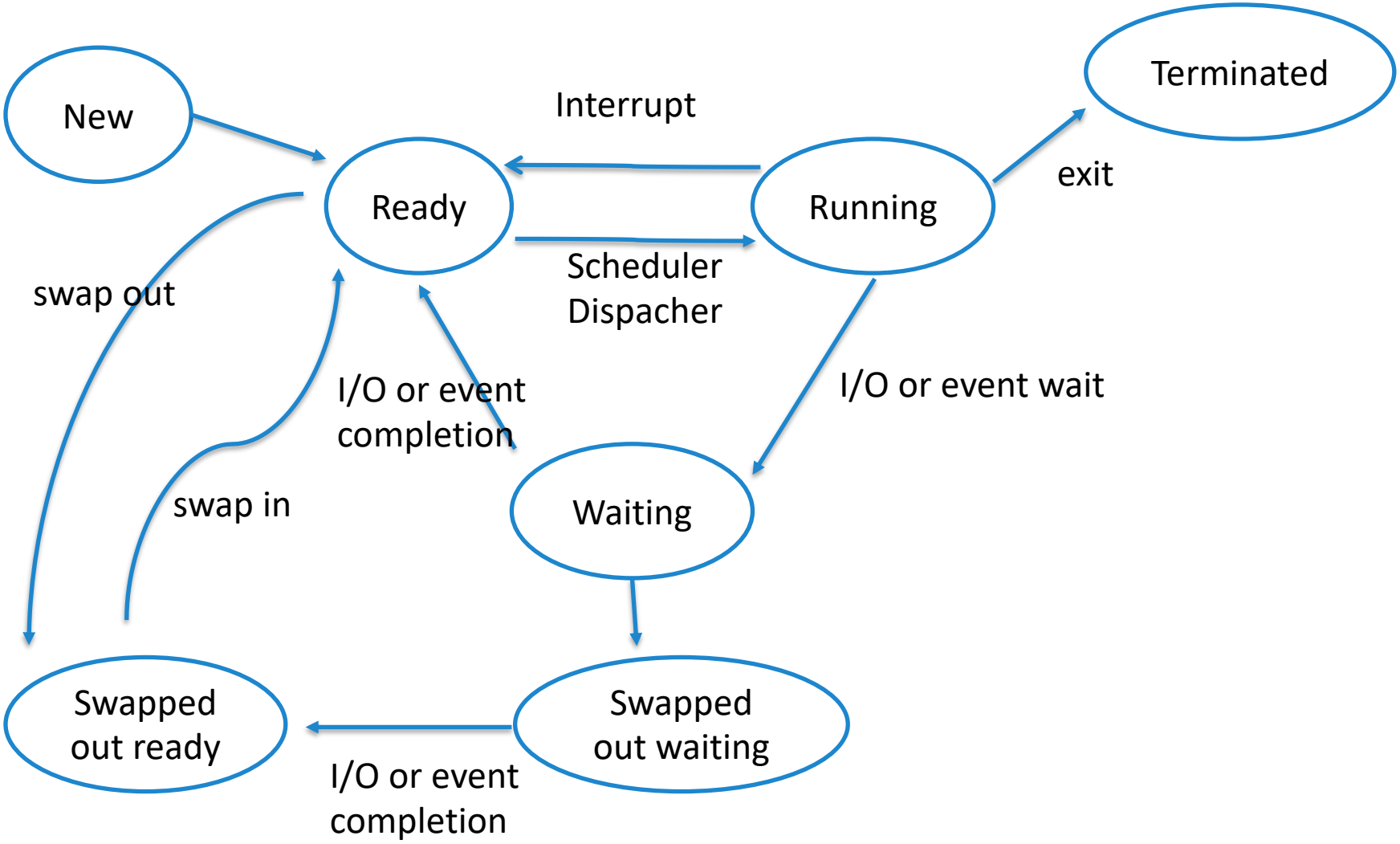
Process state diagram



Process state diagram



Process state diagram



Today's class

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

Process control block (PCB)

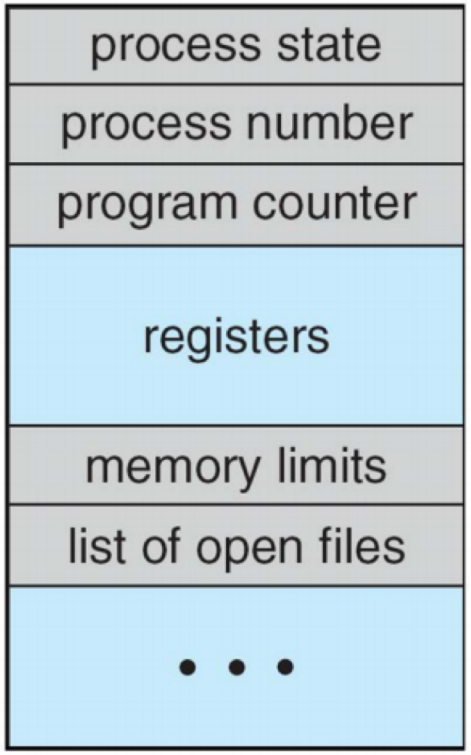
- Each process is represented in the kernel as a PCB
  - Also called task control block
  - Contains many pieces of information associated with a specific process

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

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

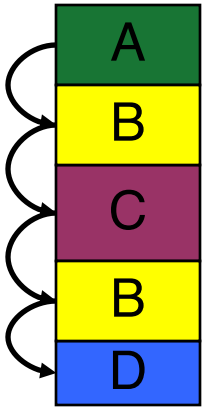


Today's class

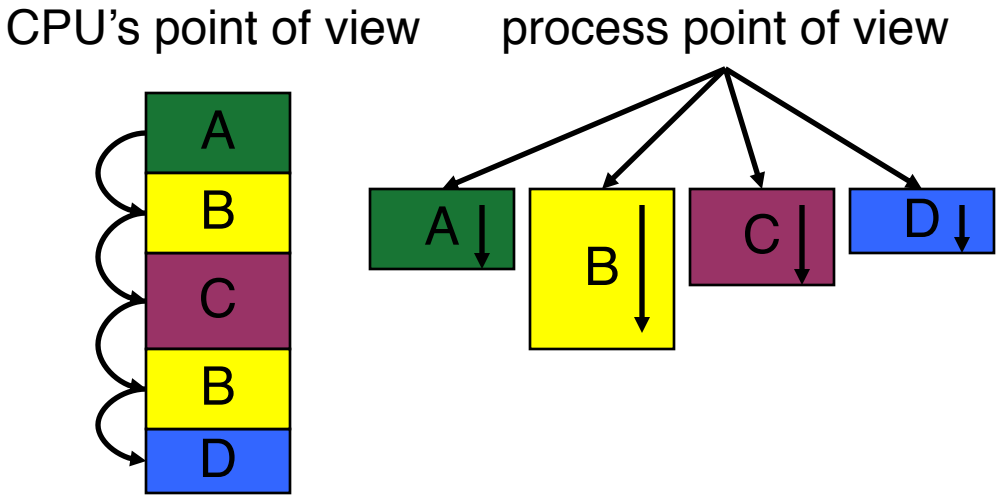
- What is a process?
- Structure of a process
- Process states
- Process control block
- Context switch

CPU runs multiple processes

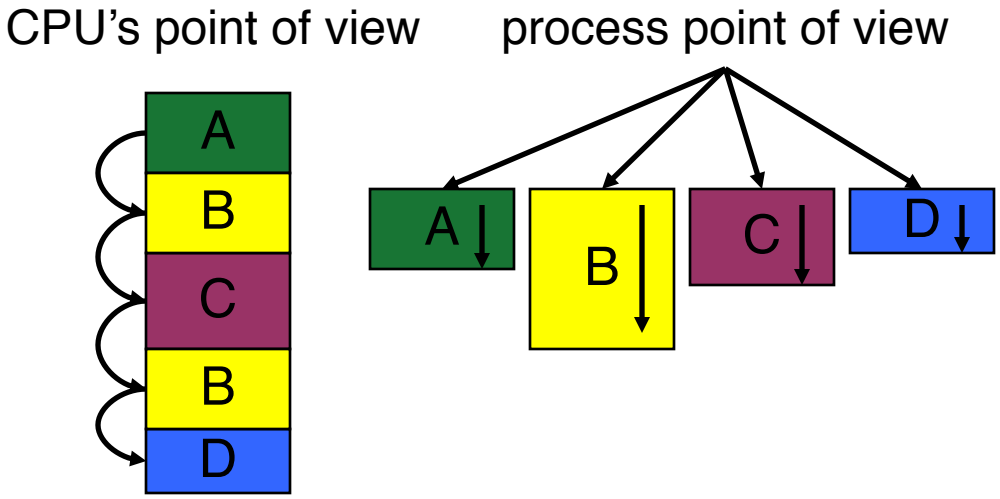
CPU's point of view



CPU runs multiple processes

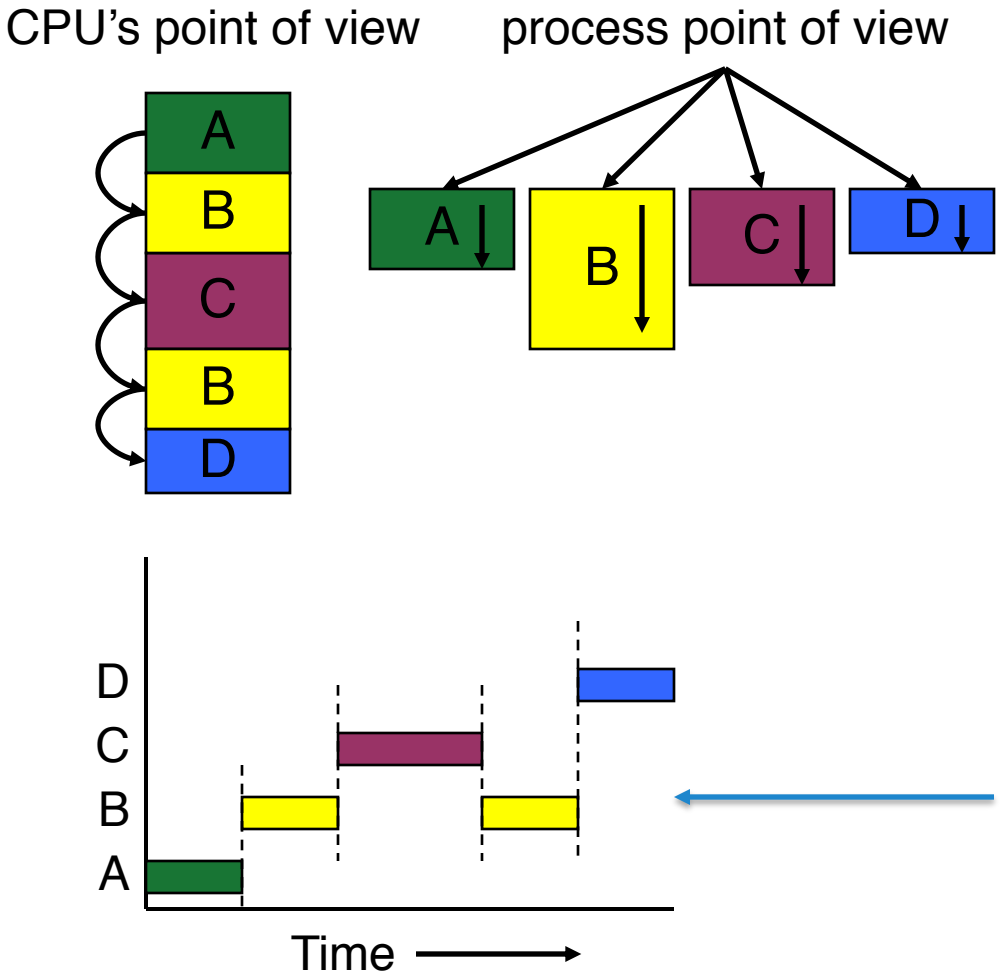


CPU runs multiple processes



- 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



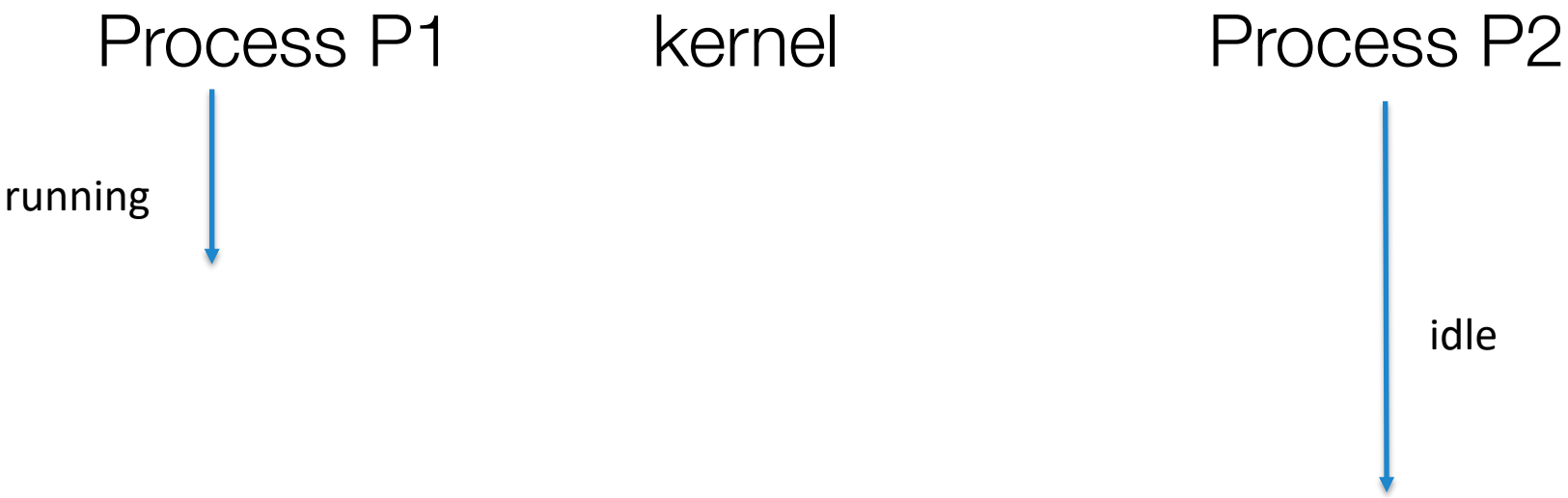
- Multiprogramming of four programs
- Conceptual model
  - 4 independent processes
  - Processes run sequentially
- Only one program active at any instant!
  - That instant can be very short…

Gantt chart for multiprogramming

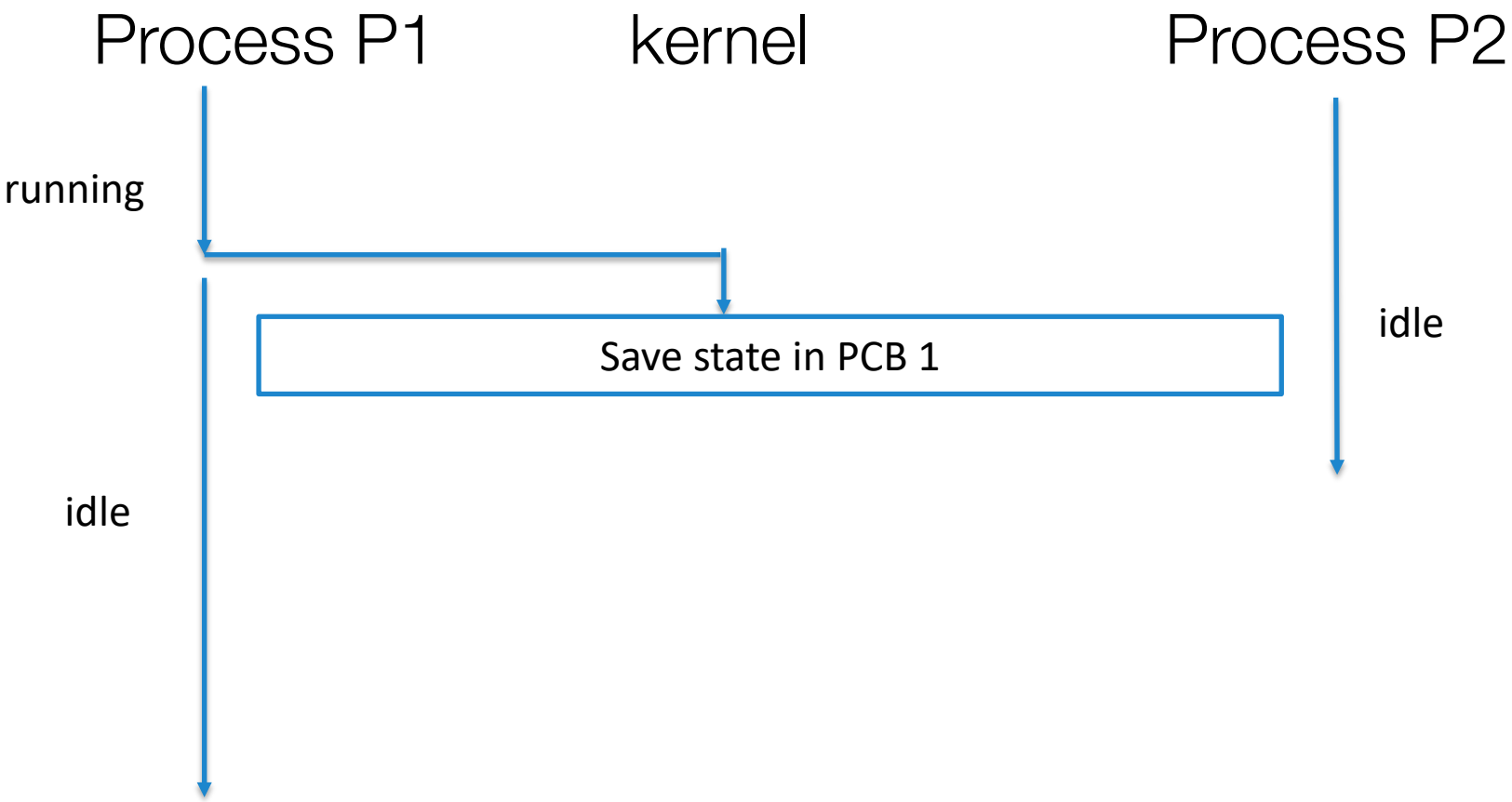
How to interleave processes?

- CPU switches to another process
  - the system saves the state of the old process and load 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

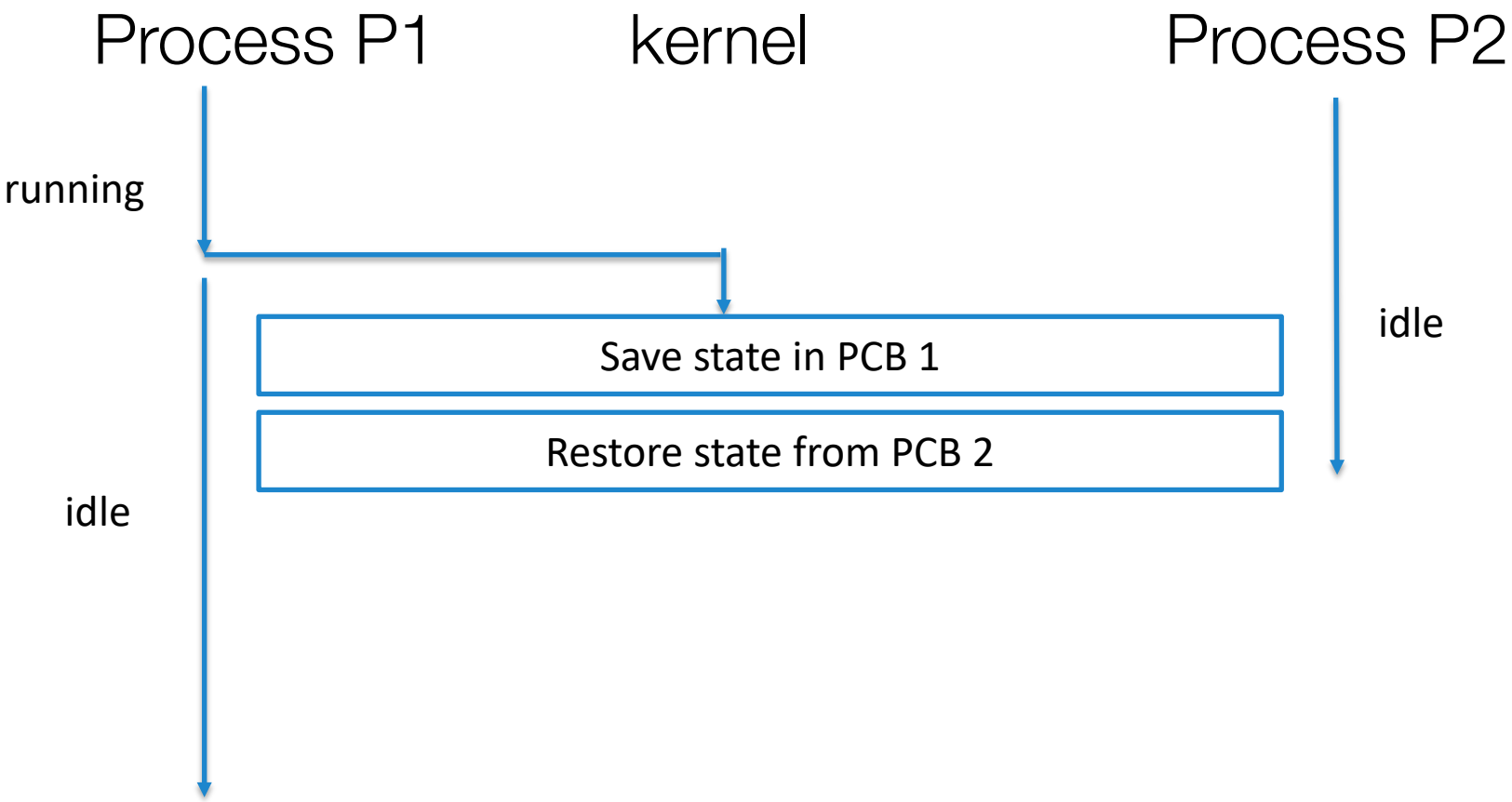
Context switch



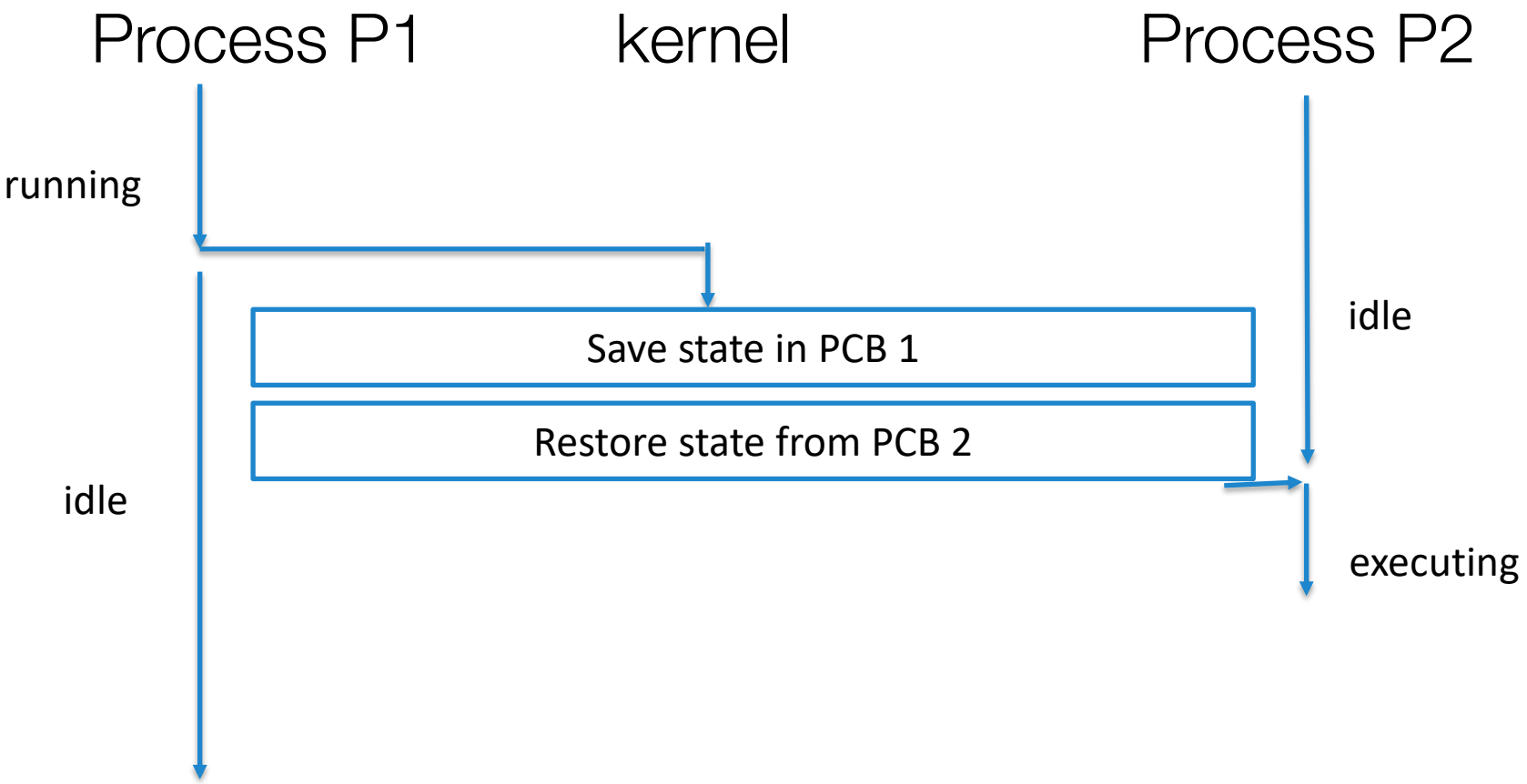
Context switch



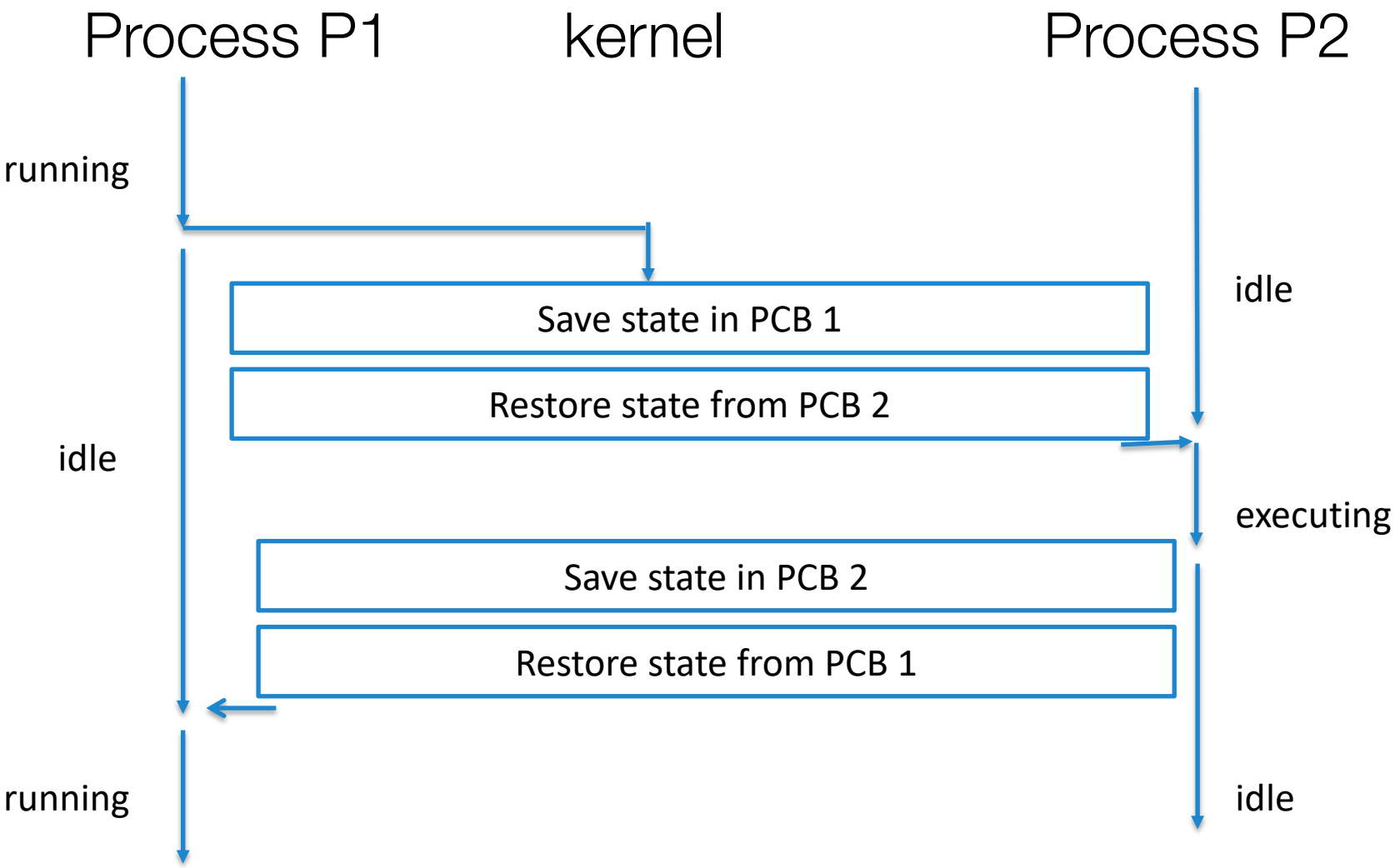
Context switch



Context switch



Context switch



Today’s class

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

Process scheduling and operations

- We will look at process scheduling the next day
- Today let’s look at an operation

pipes()

- Acts as a medium to allow two processes to communicate
  - Communication can be uni/bi directional
  - Must there exist a relationship (i.e., parent-child) between the communicating processes?
  - Can the pipes be used over a network?

Ordinary pipes

- A message passing medium between related processes
  - Cannot be accessed from outside the process
  - Typically, a parent process creates a pipe and uses it to communicate with its child process
  - Pipes behave like FIFO queues
  - Read-write in pipe == producer-consumer
  - Producer writes to one end (the write-end of pipe)
  - Consumer reads from the other end (the read-end of pipe)
  - Unidirectional

Producer consumer in pipes

Producer

```
message next_produced;
while(TRUE){
        <produce and put data in next produced>
        ...
        send( next_produced);
}
```

Consumer

```
message next_consumed;
while(TRUE){
        receive(next_consumed);
        …
        <data in next consumed>
}
```

Named pipes

- Accessed as files by processes
  - No parent child relation is necessary
  - Still behave like FIFO queues (even called fifo)
  - Several processes can use the named pipe

Named pipes

- Accessed as files by processes
  - No parent child relation is necessary
  - Still behave like FIFO queues (even called fifo)
  - Several processes can use the named pipe

```
char myfifo = '/tmp/myfifo';
mkfifo (myfifo, 0666); // creates the fifo or named pipe
…
fd = open(myfifo, O_WRONLY);  // Process A
write(fd, …);                 // Process A
close(fd);                    // Process A
…
fd = open(myfifo, O_RDONLY);  // Process B
read(fd, …);                  // Process B
close(fd);                    // Process B
```