

Bule Hora University

School of informatics

Department of Computer Science

Operating System Lecture

Prepared By: Firaol Bakana

12/18/25

Processes and Processes management

Process Concept (Cont.)

3

- In computing, a process is the instance of a computer program that is being executed by one or many threads
- A process is a program in execution.
- A process is an 'active' entity as opposed to program which is considered to be a 'passive' entity.
- It contains the program code and its activity.
- Depending on the OS, a process may be made up of multiple threads of execution that execute instructions concurrently

Process Concept (Cont.)

4

Program

- Static entity
- Exist in a single space and continue to exist.
- Programs contain instructions

Process

- Dynamic entity
- Exist in a limited span of time.
- Executes instruction

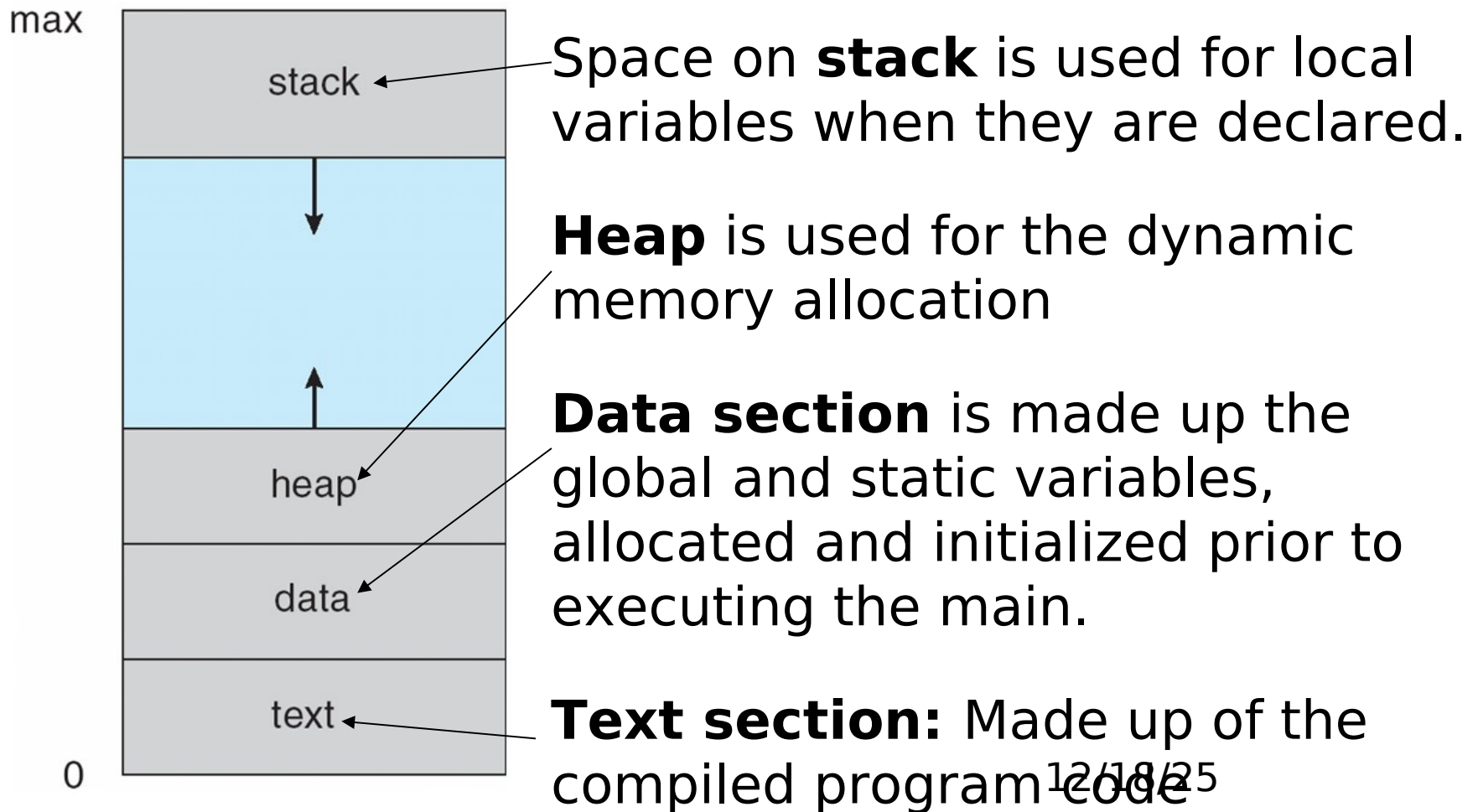
12/18/25

Program $\neq$ process

Process Concept (Cont.)

5

Sections of Process:



Process Concept (Cont.)

6

- Program is ***passive*** entity stored on disk (**executable file**),
- process is ***active***
- Program becomes process when executable file loaded into memory
- Execution of program started via GUI mouse clicks, command line entry of its name, etc
- One program can be several processes
(Consider multiple users executing the same program)

Process Concept (Cont.)

7

Process related goals of OS:

- Interleave the execution of existing processes to maximize **processor utilization**
- Provide reasonable **response times**
- **Allocate resources** to processes
- Support **inter-process communication** and user **creation of processes**

Process Concept (Cont.)

8

How are these goals achieved?

- ❑ *Schedule and dispatch* processes for execution by the processor
- ❑ Implement a safe and fair policy for *resource allocation* to processes
- ❑ Respond to requests by user programs
- ❑ *Construct and maintain tables* for each process managed by the operating system

Process Concept (Cont.)

9

Process Creation

When is the process Created?

1. System initialization
2. Execution of a process creation system call by a running process
3. A user request to create a process
4. Initiation of a batch job

Process Concept (Cont.)

10

Process Termination

When is the process Terminated?

1. Normal exit (voluntary)
2. Error exit (voluntary)
3. Fatal error (involuntary)
4. Killed by another process (involuntary)

Process State

11

As a process executes, it changes state

new: The process is being created

running: Instructions are being executed

waiting: The process is waiting for some event to occur

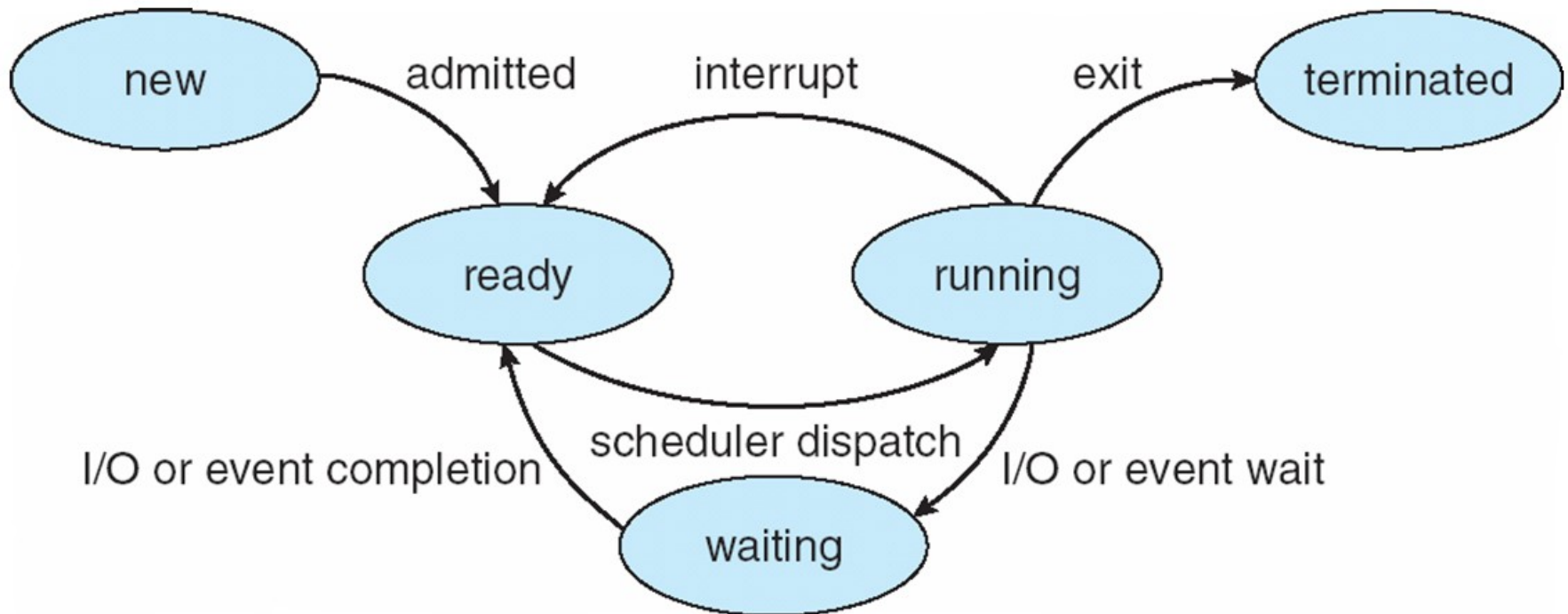
ready: The process is waiting to be assigned to a processor

terminated: The process has finished execution

Process State (cont.)

12

Diagram of Process State



12/18/25

Process State (cont.)

13

Process state transition

- ❑ **Running** - actually using the CPU
- ❑ **Ready** - runnable, temporarily stopped to let another process run
- ❑ **Blocked** - unable to run until some external event happens

A process can ***block itself***, but not “run” itself

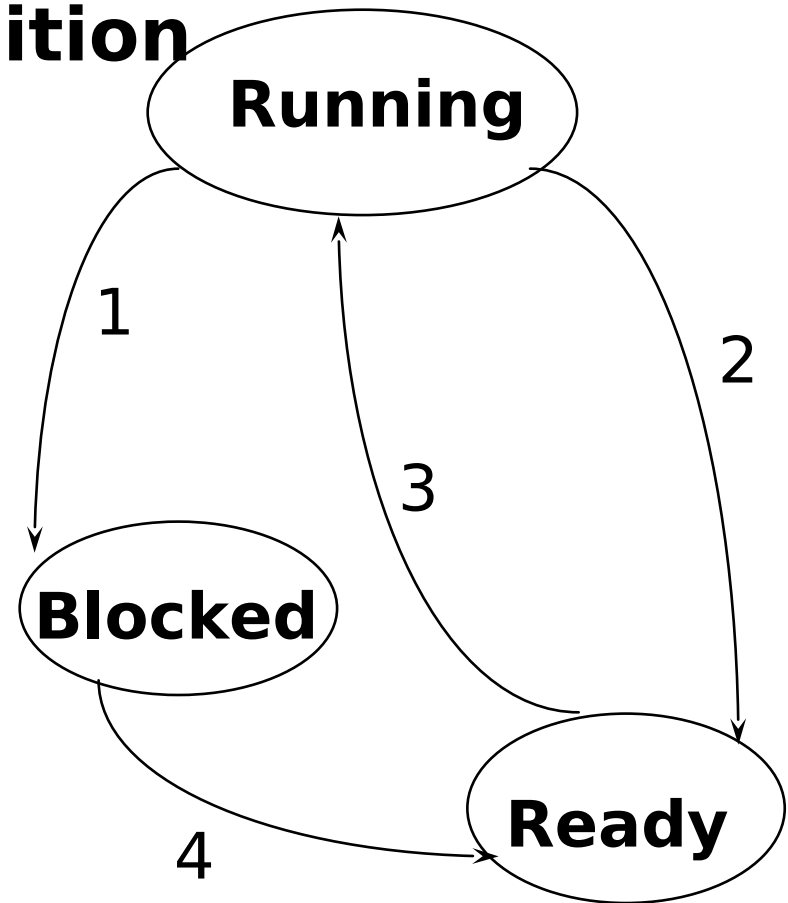
Process State (cont.)

14

Process state transition

When do these transitions
occur

1. Process **blocks** for input or waits for an event
2. End of time-slice, or preemption
3. Scheduler switches back to this process
4. Input becomes available, event arrives



12/18/25

Process Control Block (PCB)

15

PCB: Information associated with each process

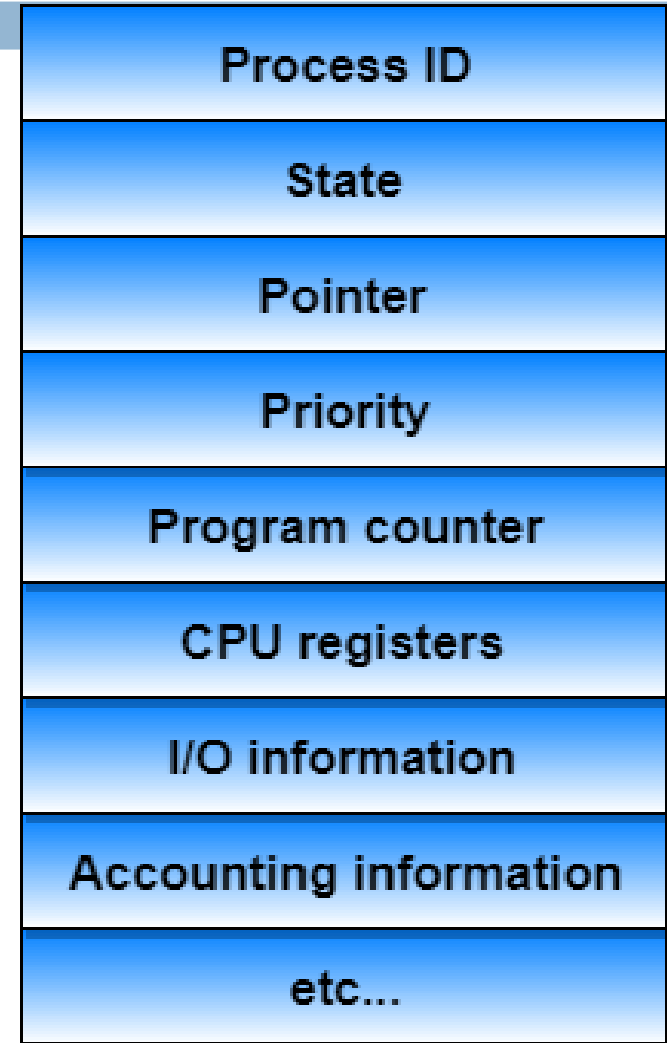
There is a PCB for each process, enclosing all the information about the process.

It is a data structure, which contains the following:

Process ID and the parent process ID.

Process State: It can be running, waiting etc.

CPU registers and **Program Counter**.



12/18/25

Process Control Block (PCB) (cont'd)

16

Program Counter holds the address of the next instruction to be executed for that process.

CPU Scheduling information: Such as priority information and pointers to scheduling queues.

Memory Management information: For example, page tables or segment tables.

Accounting information: The User and kernel CPU time consumed, account numbers, limits, etc.

I/O Status information: Devices allocated, open file tables, etc.

CPU Switch (context switch)

17

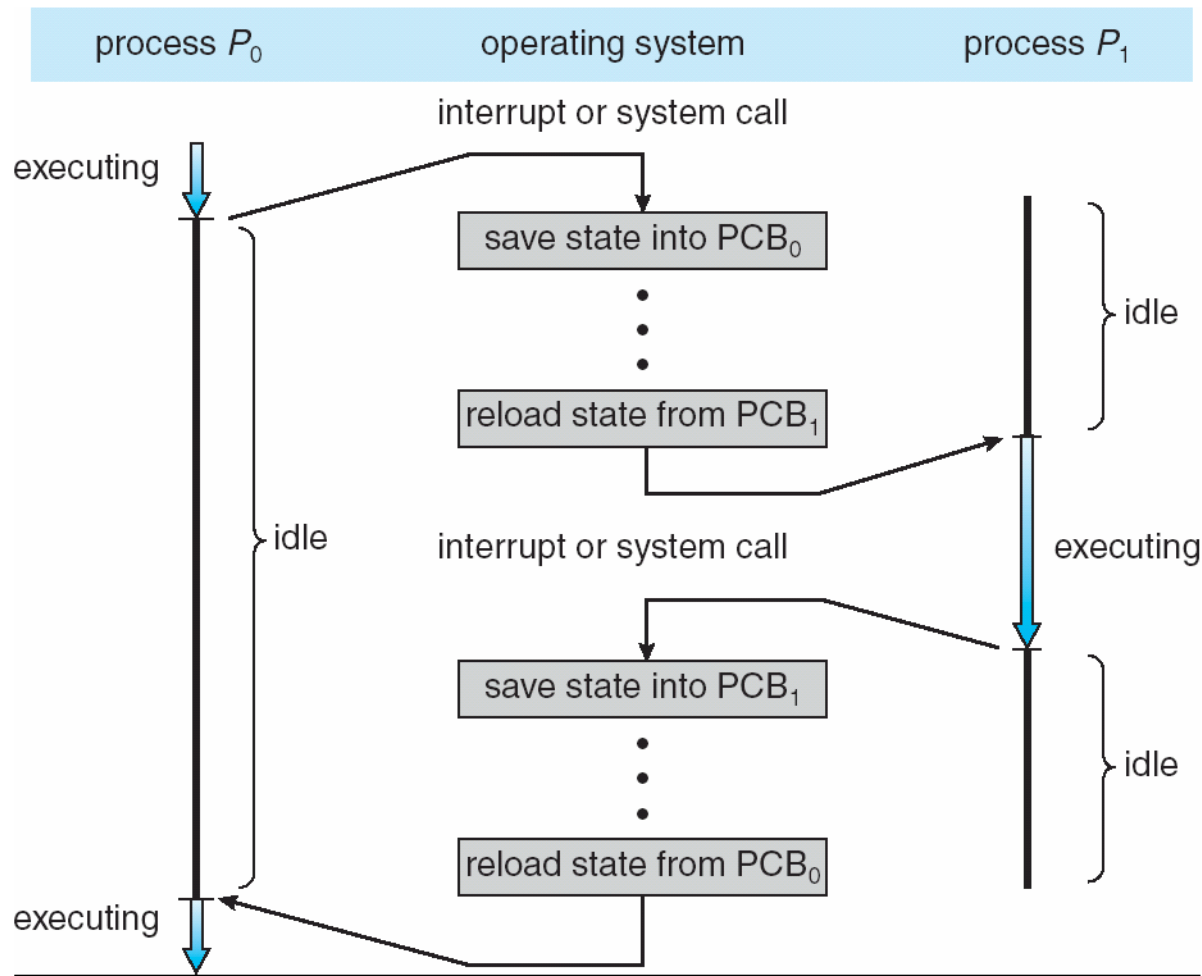
- When CPU switches to another process, the system must **save the state** of the old process and load the **saved state** for the new process via a **context switch**

Context of a process represented in the PCB

- Context-switch time is overhead; the system does no useful work while switching
 - The more complex the OS and the PCB -> the longer the context switch
- Time dependent on hardware support
 - Some hardware provides multiple sets of registers per CPU -> Multiple contexts loaded at once

12/18/25

CPU Switch (context switch) (cont'd)



. . . to be cont'd