CMPT 300
Operating System I

2.3 –Process I
Chapter 3

Dr. Hazra Imran

Admin notes

- Quiz 1 released

Learning goal

- To introduce the notion of a process -- a program in execution, which forms the basis of all computation

Process

A program in execution.

app create a new process

file program memory

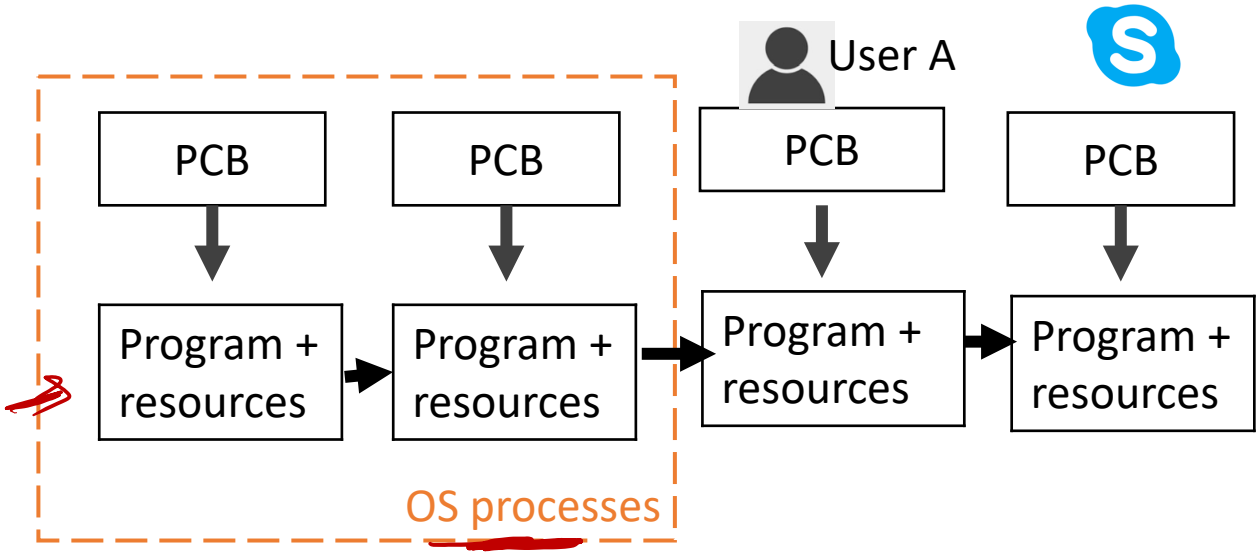
- Two essential elements of a process are:
  - Program code
  - A set of data/resources associated with that code

Main idea: processor begins to execute the program code, and we refer to this executing entity as a process.

file disk access

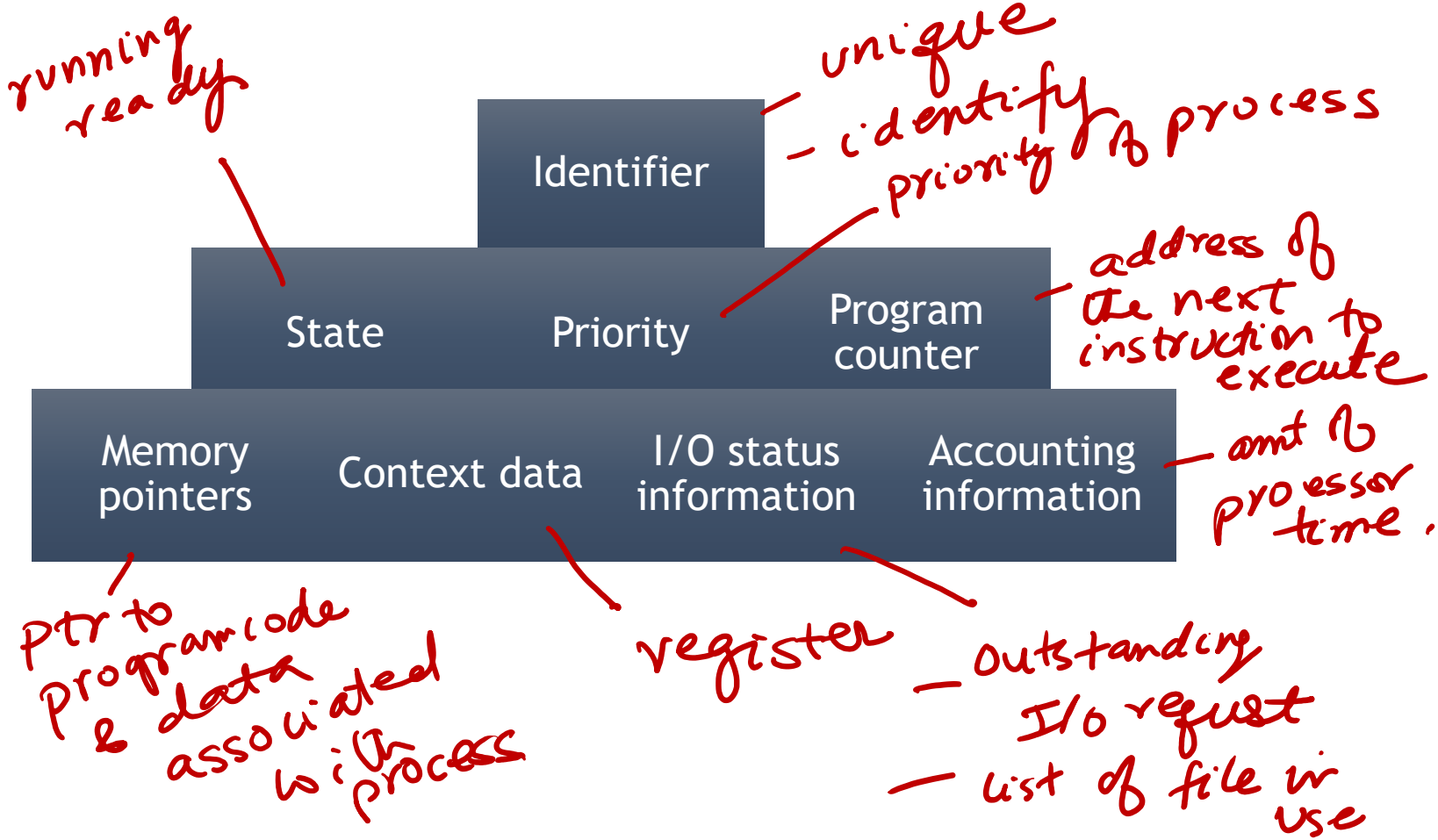
Process

OS → collection of programs.

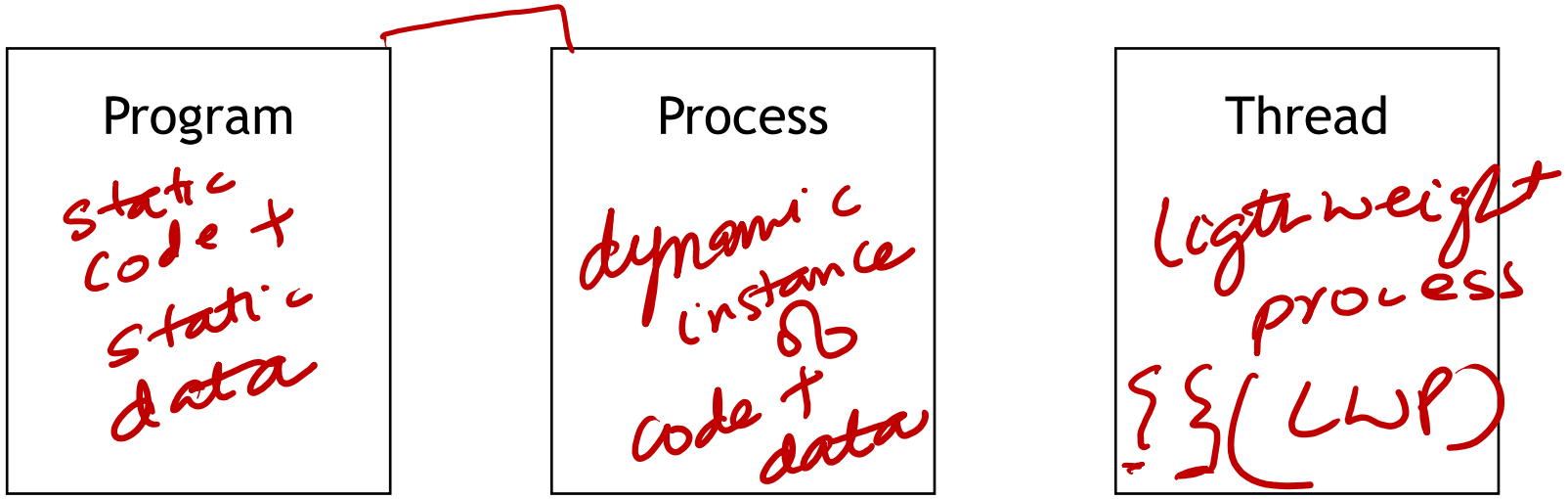


Process

- While the program is executing, the process can be uniquely characterized by a number of elements, including:



Program vs Process vs Thread



multiple processes

program → P1, P2, ..., Pn

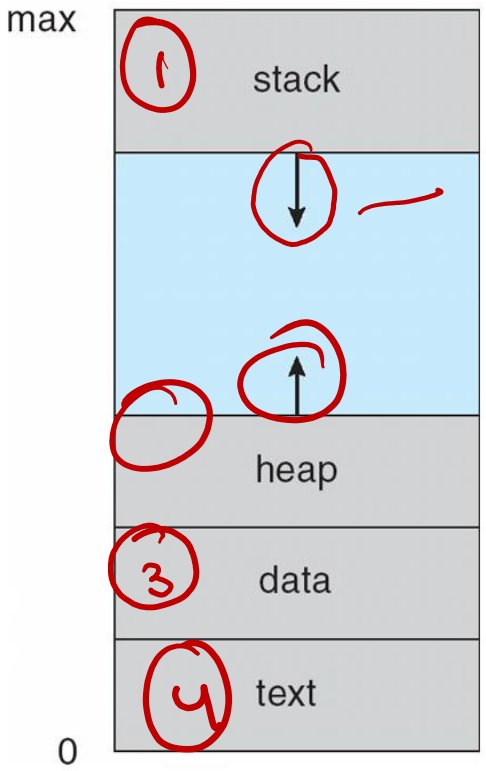
process → T1, T2, T3

Process Execution

OS	Program
• Create entry for process list • Allocate memory from program • Load program into memory • Set up stack with argc and argv • Clear registers • Execute call main()	
	• Run main() • Execute return from main
• Free memory from process • Remove from process list	

Processes and Memory

On process creation, the process is effectively given its own memory space



Stack: local variable storage

Heap: dynamically allocated space

Data: global variables (preallocated space)

Text: storage of code

Clicker

The PCB is created by the process when execution starts.

(A) True

(B) False

OS

The PCB becomes part of the program being executed by a process.

(A) True

(B) False

independent ds

created/ mainted by OS

process terminate

What do we need to track about a process?

Process States

A process is in exactly one state at any instant in time:

new **running** **waiting** **ready** **terminated**

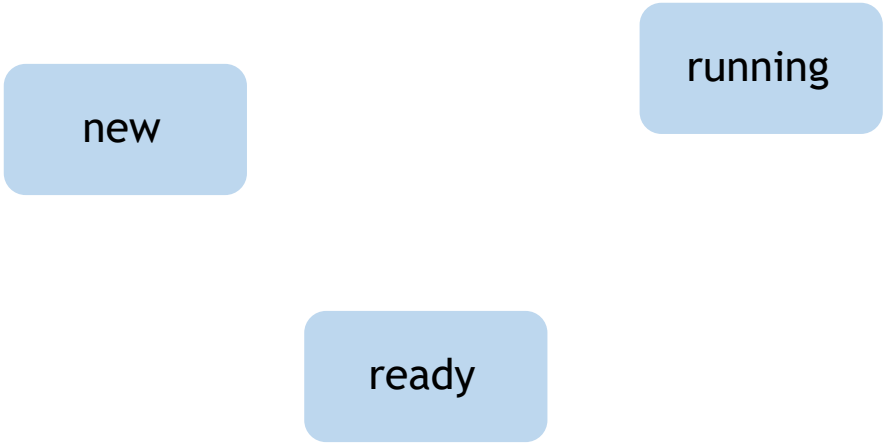
- **new:** The process is being created

Process States

A process is in exactly one state at any instant in time:

new running waiting ready terminated

- **new:** The process is being created
- **running:** Instructions are being executed by the CPU
- **ready:** The process is waiting to be assigned to a processor

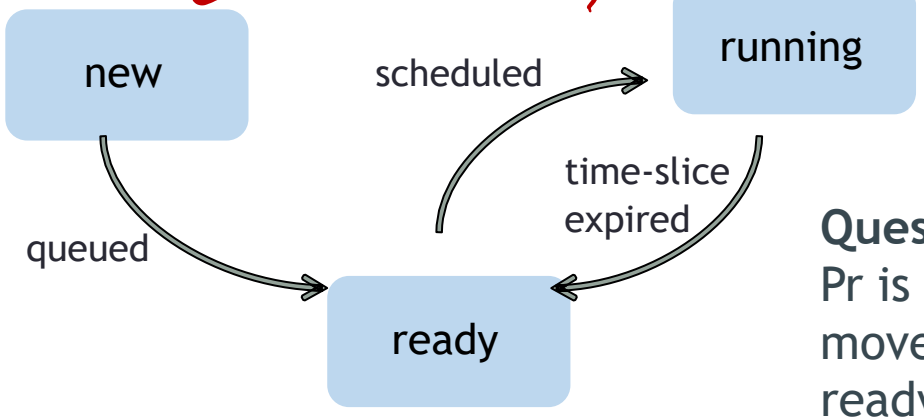


Process States

A process is in exactly one state at any instant in time:

new running waiting ready terminated

- **new:** The process is being created
- **running:** Instructions are being executed by the CPU
- **ready:** The process is waiting to be assigned to a processor



Ques. When a newly created process Pr is ready to compete for the CPU, Pr moves itself from the new to the ready state.

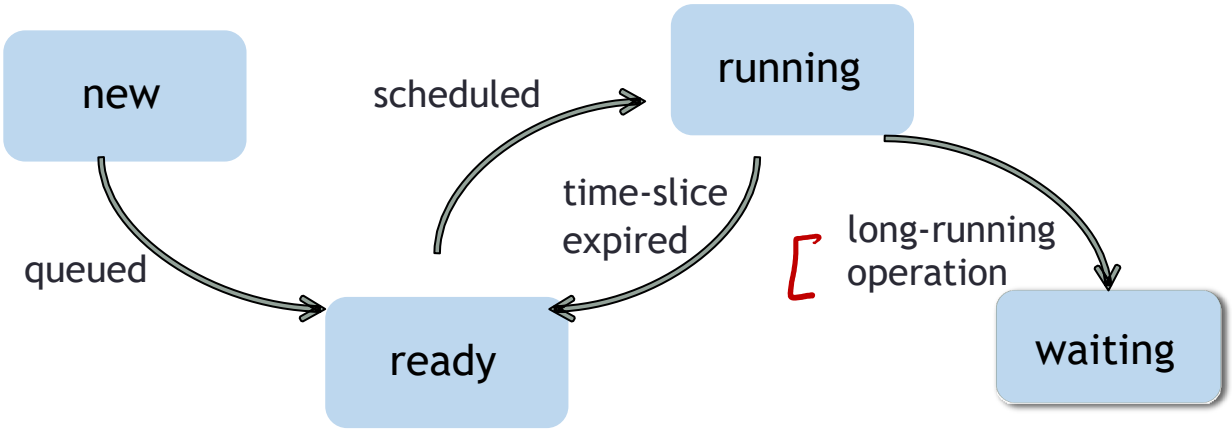
(A) True (B) False

Process States

A process is in exactly one state at any instant in time:

new running waiting ready terminated

- **waiting:** The process is waiting for some event to occur

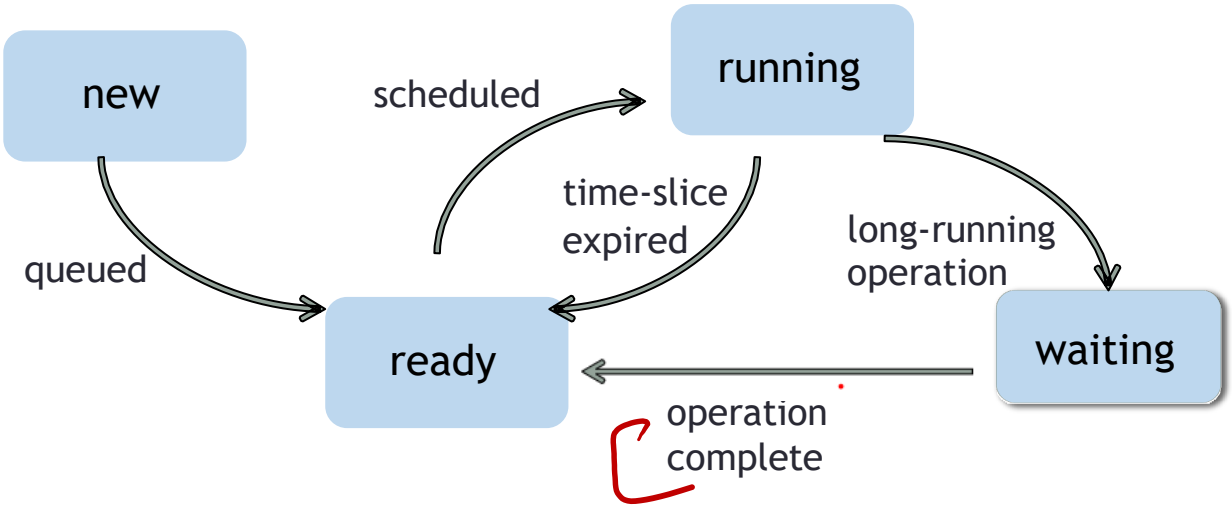


Process States

A process is in exactly one state at any instant in time:

new running waiting ready terminated

- **waiting:** The process is waiting for some event to occur

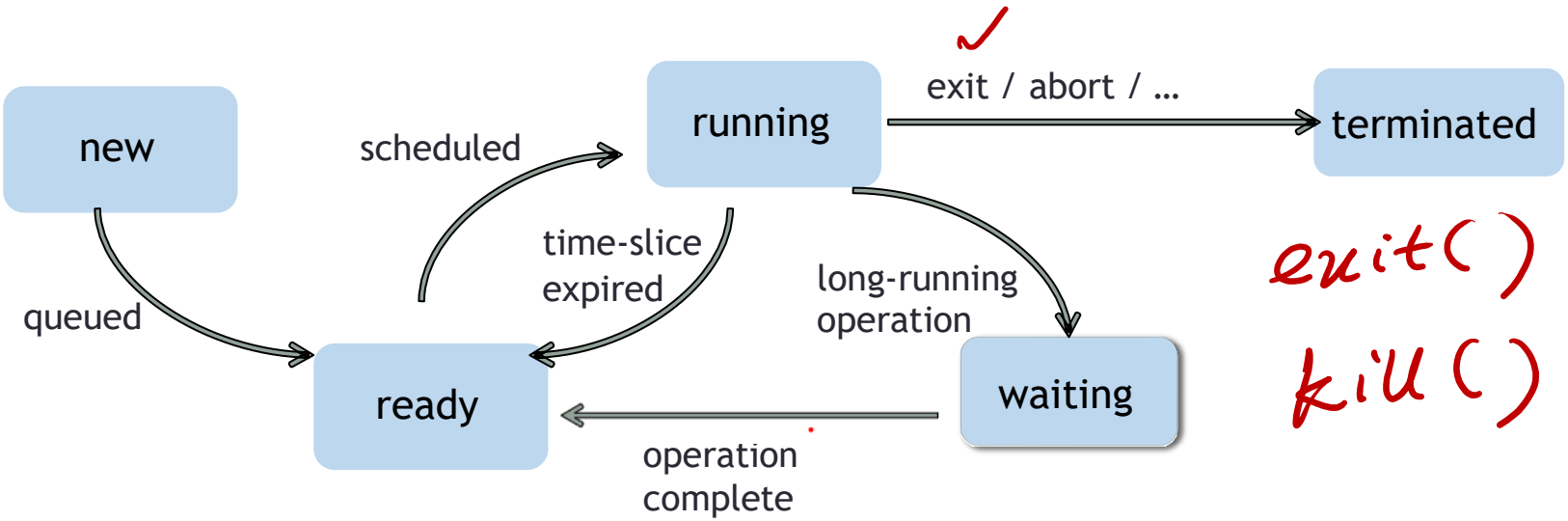


Process States

A process is in exactly one state at any instant in time:

new running waiting ready terminated

- **terminated**: The process has finished execution



Clicker

Ques. The transition (ready → running) of a process P is caused by _____.

A) OS

B) the process P itself

C) Some other process

(not in running state)

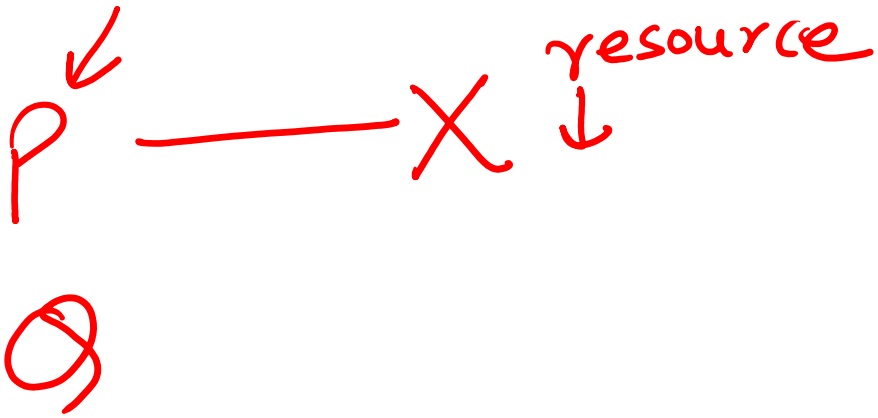
Clicker

Ques. The transition (running → waiting) of a process P is caused by _____.

A) OS

B) the process P itself

C) Some other process



Clicker

Ques. The transition (running → ready) of a process P is caused by _____.

A) OS

B) the process P itself

C) Some other process

✓

X

OS

stop a running process

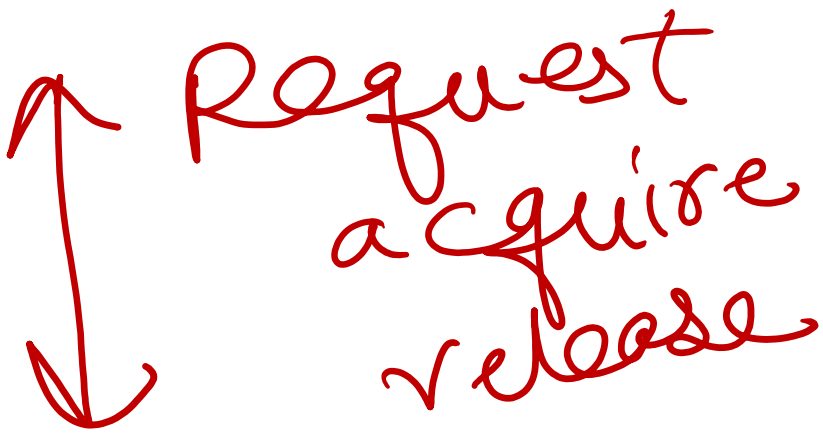
Clicker

Ques. The transition (waiting → ready) of a process P is caused by _____

A) OS

B) the process P itself

C) Some other process



Questions?

- How are a process' resources managed and reclaimed?
- How are processes in waiting are managed by the OS?
- How do we switch what process is currently running?
  - i.e. how do we perform a **context switch**?
- How does the OS choose what process should run next?
  - i.e. how does **process scheduling** work?