

## Part 2

# 2.3 PROCESS STATES AND IMPLEMENTATION OF A PROCESS



# PROCESS STATES

Ø When a process executes, it passes through different states. These stages may differ in different operating systems, and the names of these states are also not standardized.

# PROCESS STATES

**1. Running:** It means a process that is currently being executed. Assuming that there is only a single processor in the below execution process, so there will be at most one processor at a time that can be running in the state.

**2. Ready:** It means a process that is prepared to execute when given the opportunity by the OS.

**3. Blocked/Waiting:** It means that a process cannot continue executing until some event occurs like for example, the completion of an input-output operation.

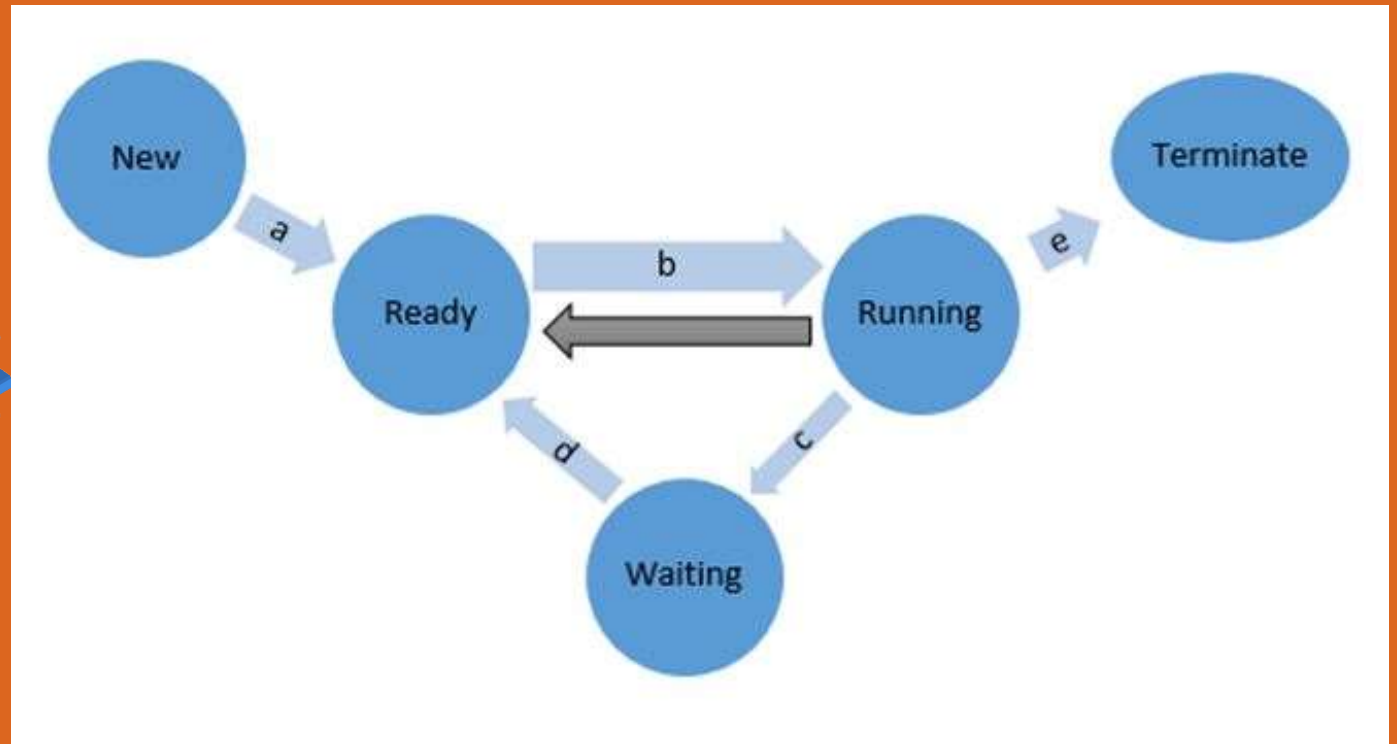
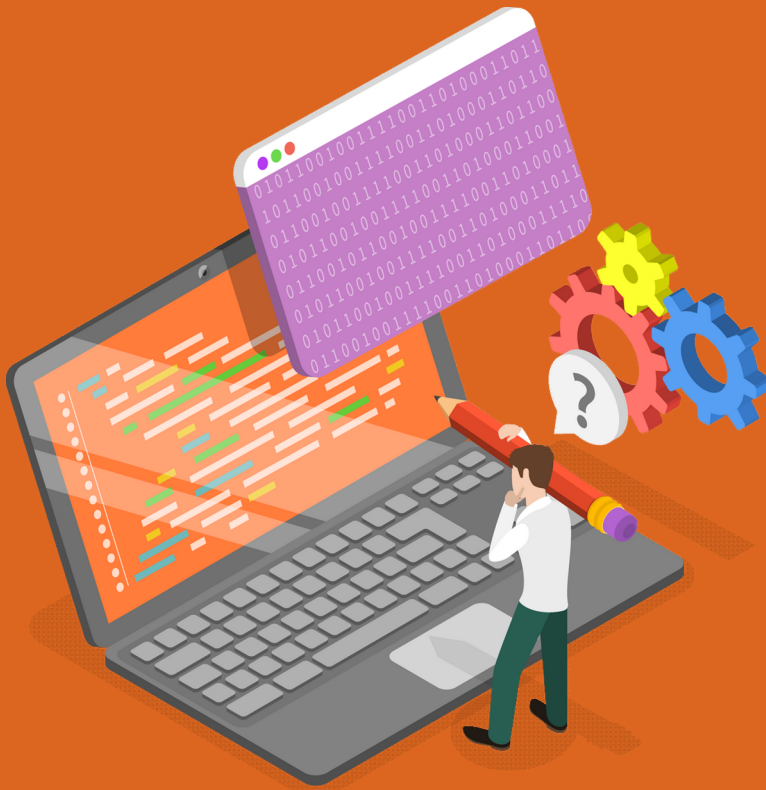


# PROCESS STATES

**4. New:** It means a new process that has been created but has not yet been admitted by the OS for its execution. A new process is not loaded into the main memory, but its process control block (PCB) has been created.

**5. .Exit/Terminate:** A process or job that has been released by the OS, either because it is completed or is aborted for some issue.

# PROCESS STATE



# IMPLEMENTATION OF A PROCESS

- The model works when any new job/process occurs in the queue, it is first admitted in the queue after that it goes in the ready state.
- Now in the Ready state, the process goes in the running state. In the running state, a process has two conditions i.e., either the process goes to the event wait or the process gets a time-out. and at last it is terminated.

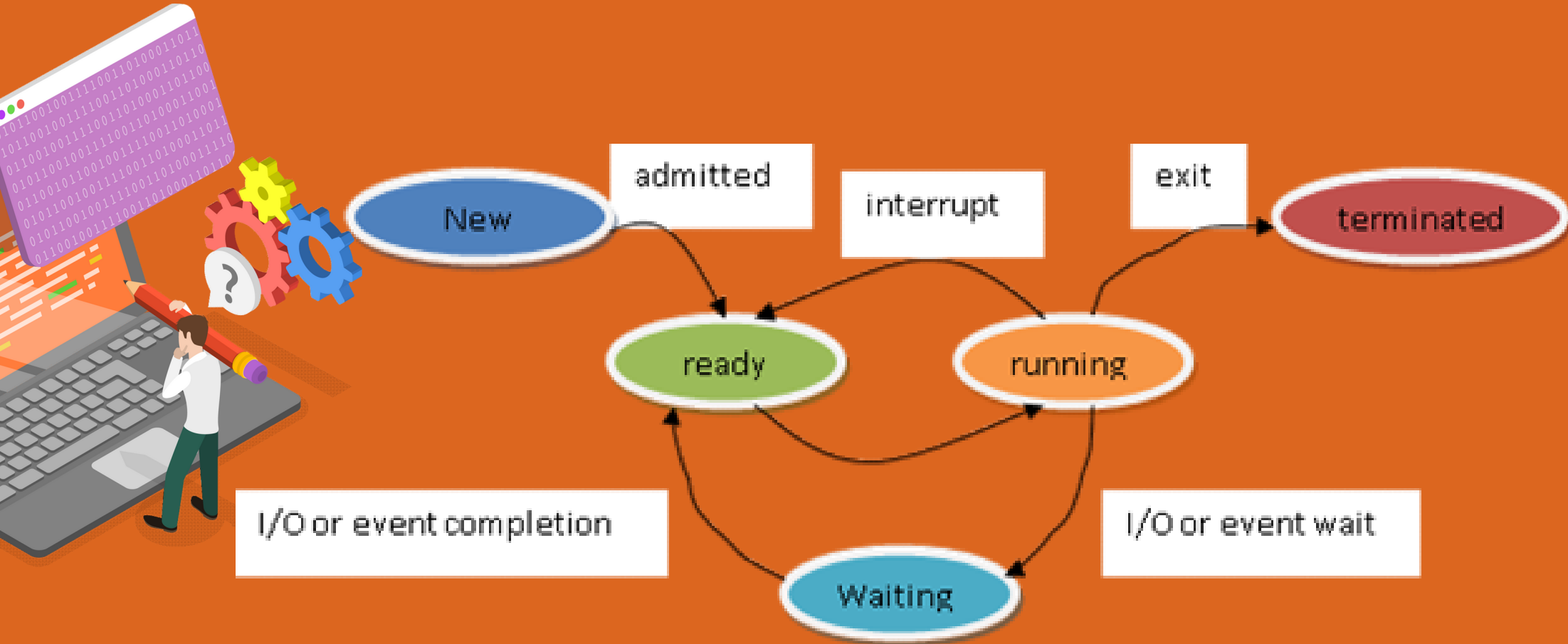


# IMPLEMENTATION OF A PROCESS

- If the process has timed out, then the process again goes to the ready state as the process has not completed its execution.
- If a process has an event wait condition then the process goes to the blocked state and after that to the ready state.
- If both conditions are true, then the process goes to running state after dispatching after which the process gets released and at last it is terminated.



# IMPLEMENTATION OF A PROCESS





# POSSIBLE STATE TRANSITIONS

- 1. Null -> New:** A new process is created for the execution of a process.
- 2. New -> Ready:** The system will move the process from new to ready state and now it is ready for execution. Here a system may set a limit so that multiple processes can't occur otherwise there may be a performance issue.
- 3. Ready -> Running:** The OS now selects a process for a run and the system chooses only one process in a ready state for execution.



## POSSIBLE STATE TRANSITIONS

**4. Running -> Exit:** The system terminates process if the process indicates that is now completed or if it has been aborted.

**5. Running -> Ready:** The reason for which this transition occurs is that when the running process has reached its maximum running time for uninterrupted execution. An example of this can be a process running in the background that performs some maintenance or other functions periodically.



## POSSIBLE STATE TRANSITIONS

**6. Running -> Blocked:** A process is put in the blocked state if it requests for something it is waiting. Like, a process may request some resources that might be available at the time or it may be waiting for an I/O operation or waiting for some other process to finish before the process can continue.

**7. Blocked -> Ready:** A process moves from blocked state to the ready state when the event for which it has been waiting.

**8. Ready -> Exit:** This transition can exist only in some cases because, in some systems, a parent may terminate a child's process at any time.



# QUESTIONS?



# NEXT.....



PREPARED BY:  
Ms. MARIEFEL T. BASIBAS

