NATIONAL UNIVERSITY
OF COMPUTER & EMERGING SCIENCES

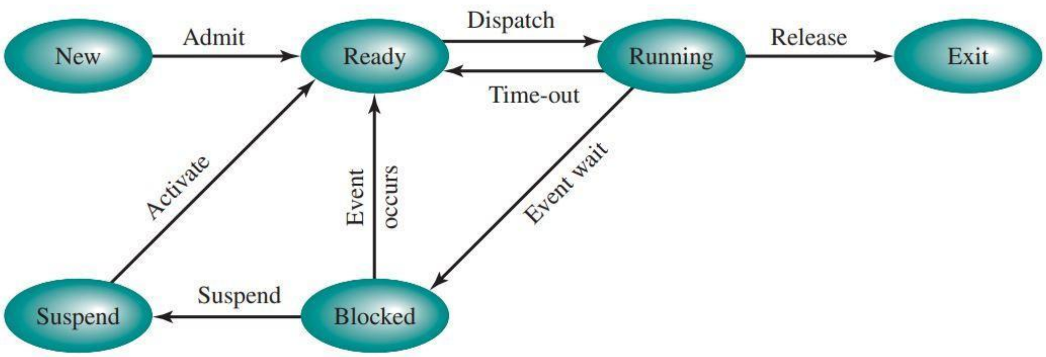
CS2006 – Operating System

Fall 2023

ID: ________________ **Name:**____________________
Assignment #1 **Instructor :** Sanaa Jeehan
Marks: 20 **Date:** 14th September, 2023

Question #1 - 10 Marks

When all of the processes in main memory are in the Blocked state, the OS can suspend one process by putting it in the Suspend state and transferring it to disk. The space that is freed in main memory can then be used to bring in another process.



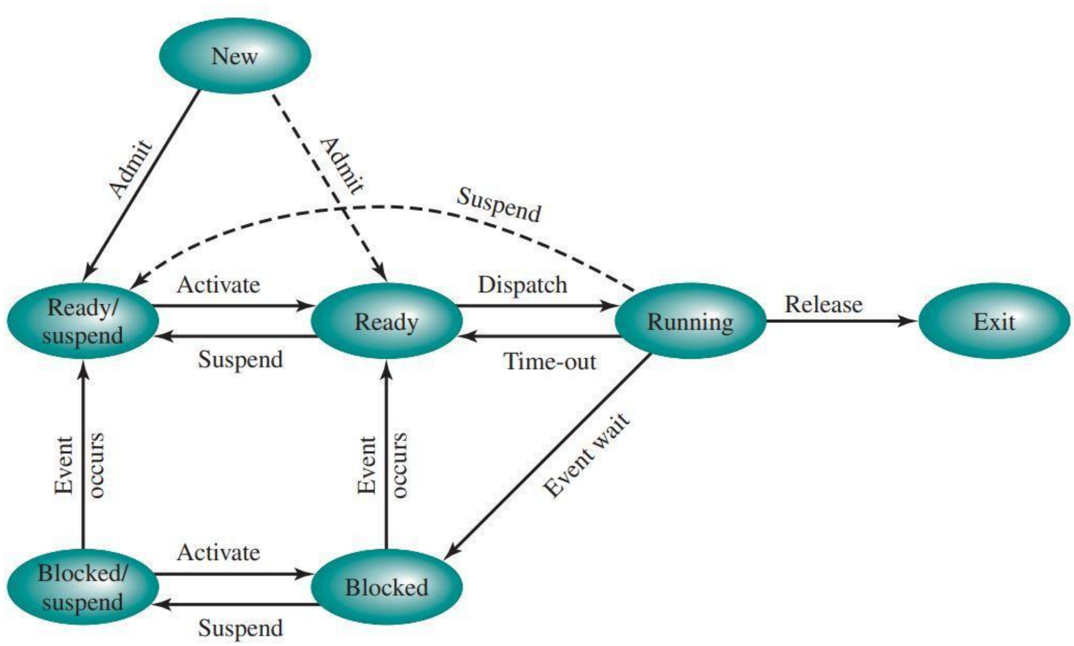
When the OS has performed a swapping-out operation, it has two choices for selecting a process to bring into main memory: It can admit a newly created process or it can bring in a previously suspended process. It would appear that the preference should be to bring in a previously suspended process, to provide it with service rather than increasing the total load on the system.

But this line of reasoning presents a difficulty. All of the processes that have been suspended were in the Blocked state at the time of suspension. It clearly would not do any good to bring a blocked process back into main memory, because it is still not ready for execution. Recognize, however, that each process in the Suspend state was originally blocked on a particular event. When that event occurs, the process is not blocked and is potentially available for execution.

Therefore, we need to rethink this aspect of the design. There are two independent concepts here: whether a process is waiting on an event (blocked or not) and whether a process has been swapped out of main memory (suspended or not). To accommodate this 2 * 2 combination, we need four states:

- Ready: The process is in main memory and available for execution
- Blocked: The process is in main memory and awaiting an event. Event occurs
- Blocked/Suspend: The process is in secondary memory and awaiting an event.
- Ready/Suspend: The process is in secondary memory but is available for execution as soon as it is loaded into main memory.

The resulting figure contains seven states shown as follows:



In principle, one could draw a transition between any two states, for a total of 42 different transitions.

a) List all of the possible transitions and give an example of what could cause each transition.
b) List all of the impossible transitions and explain why.

Question #2 - 10 Marks

Assume that at time 5 no system resources are being used except for the processor and memory. Now consider the following events:

a) At time 5: P1 executes a command to read from disk unit 3.
b) At time 15: P5's time slice expires.
c) At time 18: P7 executes a command to write to disk unit 3.
d) At time 20: P3 executes a command to read from disk unit 2.
e) At time 24: P5 executes a command to write to disk unit 3.
f) At time 28: P5 is swapped out.
g) At time 33: An interrupt occurs from disk unit 2: P3's read is complete.
h) At time 36: An interrupt occurs from disk unit 3: P1's read is complete.
i) At time 38: P8 terminates.
j) At time 40: An interrupt occurs from disk unit 3: P5's write is complete.
k) At time 44: P5 is swapped back in.
l) At time 48: An interrupt occurs from disk unit 3: P7's write is complete.

For each time 22, 37, and 47, identify which state each process is in. If a process is blocked, further identify the event on which it is blocked.