ASSIGNMENT # 01

Submitted By

QASIM ALI (20P-0070)

Submitted to : Sana Jehaan

(INSTRUCTOR CS(OS))

DEPARTMENT OF COMPUTER SCIENCE

FAST NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES, PESHAWAR

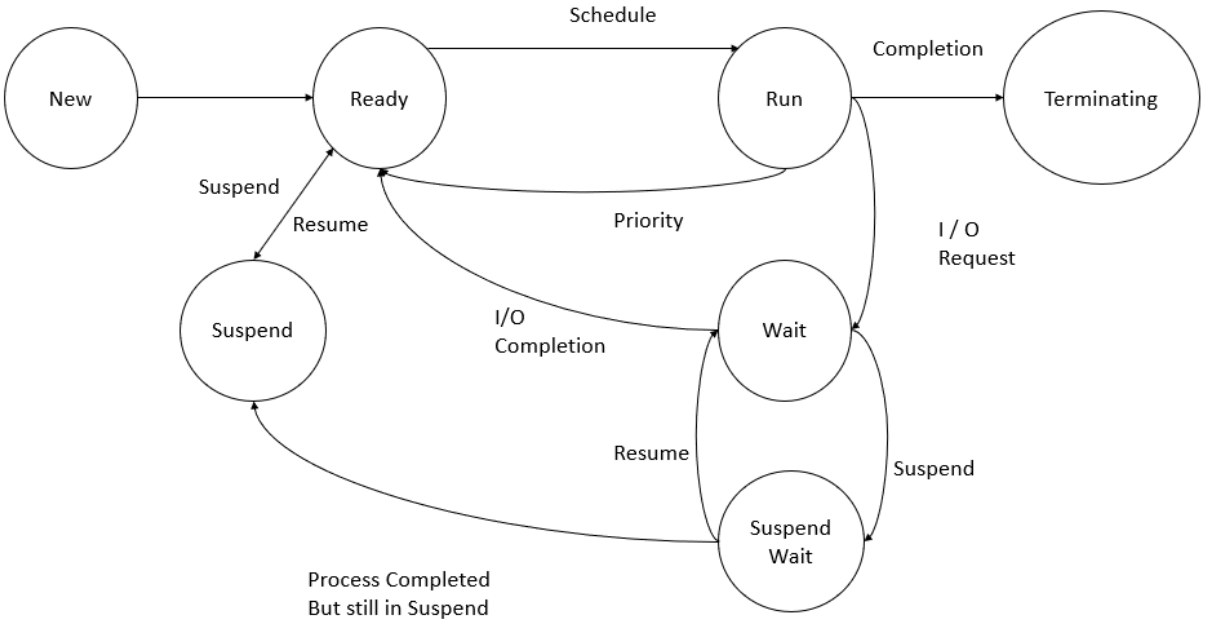
Session 2020-2024

QUESTION # 01

A) List all of the possible transitions and give an example of what could cause each transition?

Possible State Transitions:

- Null → New
- New → Ready
- Ready → Running
- Running → Exit
- Running → Ready
- Running → Blocked
- Blocked → Ready
- Ready → Exit
- Blocked → Suspend
- Suspend → Blocked
- Suspend → Ready
- Running → Terminate



Example:

1. **Null → New:**

   A new process is created for execution.

2. **New → Ready:**

   The new generated process that is prepared to run on the CPU and is awaiting its turn for execution is transitioned to the prepared state. Several processes can exist in this state concurrently.

3. **Ready → Running:**

   The chosen process is shifted to the Executing state, where it gains access to the CPU for performing its tasks. Only one process can occupy the Executing state at any given moment.

4. **Running → Exit:**

   The process whose execution gets over will be terminated and moved into the Exit state.

5. **Running → Ready:**

   When the process has exhausted its allotted Running time or when a higher-priority process becomes available for Running, the process currently residing in the Running state will be transitioned back to the Ready state.

6. **Running → Blocked:**

   When the process is waiting for some event to occur the process will be moved to the Blocked state from the Running state.

7. **Blocked → Ready:**

   A process will be moved back into the ready state when the event that it has been waiting for occurs.

8. **Ready → Exit:**

   This transition occurs only when the parent process of the current process is terminated, or when the child process explicitly requests termination.

9. **Blocked → Suspend:**

   When the Blocked queue becomes saturated with processes, a portion of the processes from the blocked queue will be transferred to the Suspended state. The Suspended state is located in secondary memory and is managed through the use of virtual memory.

10. **Suspend → Blocked:**

    When space becomes available within the Blocked state, the process previously placed in the Suspended state will be returned to the Blocked State.

11. Suspend → Ready:

When an event or condition occurs for a process that has been waiting in the Suspended state within secondary memory, it will be promptly transitioned to the Ready state.

12. Running → Terminate:

This transition takes place when a process encounters an error or an exceptional condition that cannot be handled, or when the process explicitly requests termination. In such circumstances, the process is shifted from the Executing state to the Terminate state, where it is flagged for termination, and its resources are released.

B) List all of the impossible transitions and explain why?

Impossible State Transitions:

Running → New

Blocked → New

Exit → New

Exit → Ready

Exit → Running

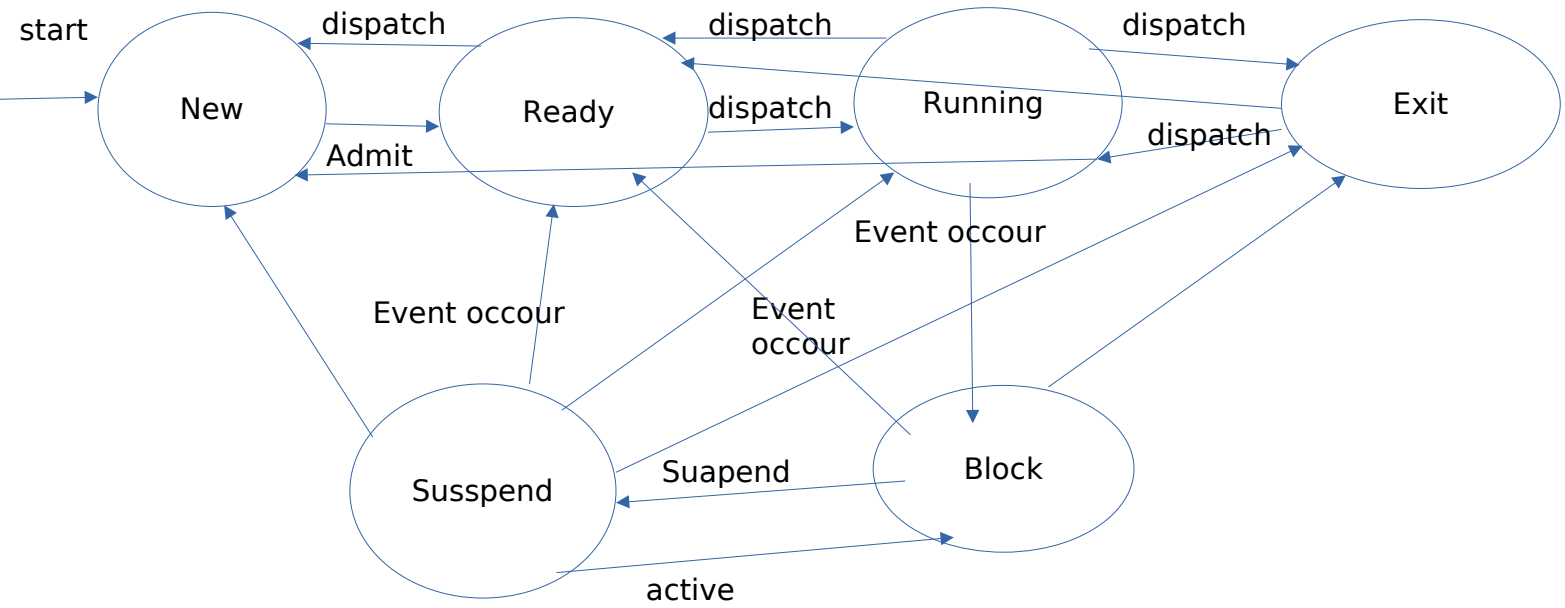
Suspend → New

Suspend → Running

Ready → New

Blocked → Exit

Suspend → Exit



Example:

Running → Error:

This transition occurs when a process encounters an unrecoverable error or exceptional condition. In the "Error" state, the process is flagged for termination, and its resources are released.

Blocked → Available:

When space becomes free within the Blocked state, processes previously held in the Blocked state can transition to the "Available" state, where they await execution.

Exit → Completed:

After a process exits, it is marked as "Completed" to indicate that it has finished its execution and cannot return to the "New" state.

Exit → Waiting:

Processes in the "Exit" state have completed their execution and cannot move directly to the "Ready" state, which is reserved for processes awaiting CPU execution.

Exit → Execution:

A process that has exited has finished running and cannot return to the "Running" state.

Suspend → Load:

Processes in the "Suspend" state are stored in secondary memory and need to be explicitly "Loaded" into memory before transitioning to other states.

Suspend → Memory:

Processes in the "Suspend" state are in secondary memory and must be "Loaded" into memory before they can enter the "Running" state.

Ready → Active:

The "Ready" state represents processes prepared for CPU execution and cannot revert to the "New" state, which is reserved for newly created processes.

Blocked → Completed:

Processes in the "Blocked" state wait for events and cannot move directly to the "Exit" state, which signifies the completion of execution.

Suspend → Terminate:

Similar to the "Blocked" state, processes in the "Suspend" state need to be "Loaded" into memory and transition to other states before reaching the "Exit" state.

QUESTION # 02

ANS:

To determine the status of each task at times 22, 37, and 47, we can review the events and monitor the progress of each task. Here's the analysis:

1. Time 22:

Task A: Currently blocked while reading from disk unit 3 (event c). The operation has not yet been completed.

Task B: There is no information available regarding Task B's status in the events. We assume it remains in the ready state.

Task C: Currently blocked while reading from disk unit 2 (event d). It was interrupted at time 33 (event g), so it's still waiting for the read operation to complete.

Task D: There is no information available regarding Task D's status in the events. We assume it remains in the ready state.

Task E: Currently blocked while writing to disk unit 3 (event e). It was interrupted at time 44 (event k), so it's still waiting for the write operation to complete.

Task F: There is no information available regarding Task F's status in the events. We assume it remains in the ready state.

Task G: Currently blocked while writing to disk unit 3 (event l). The operation has not yet been completed.

Task H: Terminated at time 38 (event i).

2. Time 37:

Task A: Currently blocked while reading from disk unit 3 (event c). The operation has not yet been completed.

Task B: There is no information available regarding Task B's status in the events. We assume it remains in the ready state.

Task C: Successfully completed reading from disk unit 2 at time 33 (event g). It has no further events, so we assume it's in the ready state.

Task D: There is no information available regarding Task D's status in the events. We assume it remains in the ready state.

Task E: Currently blocked while writing to disk unit 3 (event e). It was interrupted at time 44 (event k), so it's still waiting for the write operation to complete.

Task F: There is no information available regarding Task F's status in the events. We assume it remains in the ready state.

Task G: Currently blocked while writing to disk unit 3 (event l). The operation has not yet been completed.

Task H: Terminated at time 38 (event i).

3. Time 47:

Task A: Successfully completed reading from disk unit 3 at time 36 (event h). It has no further events, so we assume it's in the ready state.

Task B: There is no information available regarding Task B's status in the events. We assume it remains in the ready state.

Task C: Successfully completed reading from disk unit 2 at time 33 (event g) and has no further events. We assume it's in the ready state.

Task D: There is no information available regarding Task D's status in the events. We assume it remains in the ready state.

Task E: Currently blocked while writing to disk unit 3 (event e). It was interrupted at time 44 (event k), so it's still waiting for the write operation to complete.

Task F: There is no information available regarding Task F's status in the events. We assume it remains in the ready state.

Task G: Successfully completed writing to disk unit 3 at time 48. It's now back to the ready state.

Task H: Terminated at time 38 (event i).

Since the status of **Task B** and **Task D** is not specified in the events, we assume they are in the ready state.