OPERATING SYSTEM

The Process

Introduction

Process creation & termination

Process state diagram

Process scheduling & its criteria

Process

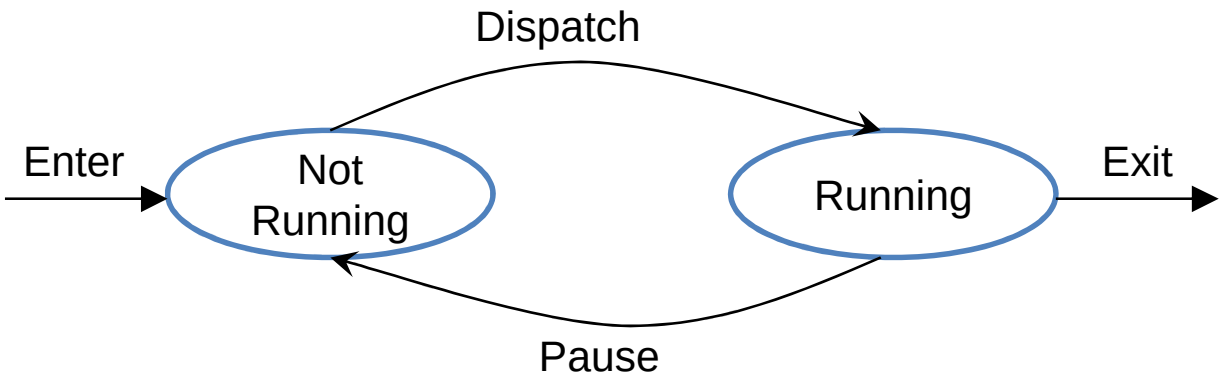
- The concept of process is fundamental to the structure of operating systems.
- Many definitions have been given for the term *process*, including:
  - A program in execution.
  - The “animated spirit” of a program.
  - The entity that can be assigned to and executed on a processor.
  - A program is a passive entity while process is a active entity

Process

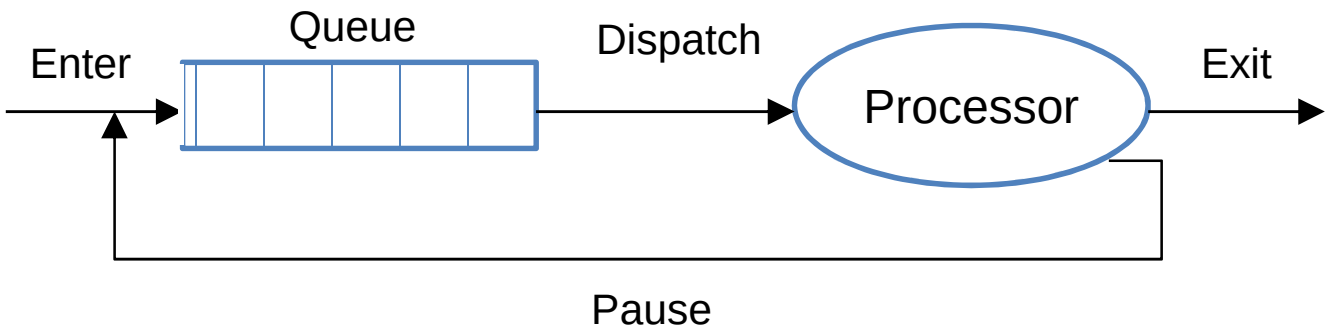
Many requirements that the operating system must meet can all be expressed with reference to process:

1. The operating system must interleave the execution of a number of processes to maximize processor use while providing reasonable response time.
2. The operating system must allocate resources to a process in conforming to a specific policy (e.g. certain functions or applications are of higher priority) while at the same time avoiding deadlocks.
3. The operating system may be required to support inter-process communication and user creation of processes, both of which may aid in the structuring of applications.

A Two-State Process Model



(a) State transition diagram



(b) Queuing diagram

The Creation and Termination of Processes

- Regardless of the model of process behavior used, the life of process is bounded by its creation and termination.

Creation of Processes

- When a new process is to be added to those that are currently being managed by the OS, the OS
  - Builds the data structures that are used to manage the process and
  - Allocates the address space to be used by the process.
- These actions constitute the creation of a new process.

The Creation and Termination of Processes

Reasons for Process Creation

1. A process is created in response to the submission of a job/task.
2. Interactive Log On initiates process(es)
3. OS creates process to Provide a Service on behalf of a user program
4. Spawned by Existing Processes for purposes of modularity or to exploit parallelism.
   - The spawning process is called the parent process and the spawned process is called the child process.
   - Related processes need to “communicate” and “cooperate” with each other.

The Creation and Termination of Processes

Termination of Process

- A process should acknowledge OS about its completion for resource de-allocation
- A batch job should indicate a “Halt” instruction, which generates an interrupt to alert the operating system for process completion
- User actions can indicate when the process is completed in interactive applications

Reasons of Process Termination

1. Normal Termination: The process executes an OS service call to indicate that it has completed running.
2. Time Limit Exceeded: The process has run longer than the specified total time limit.
   - Types of process time measuring
     - Total time elapsed
     - Amount of time spent executing
     - In user-interactive processes, the amount of time since the user provided input

The Creation and Termination of Processes

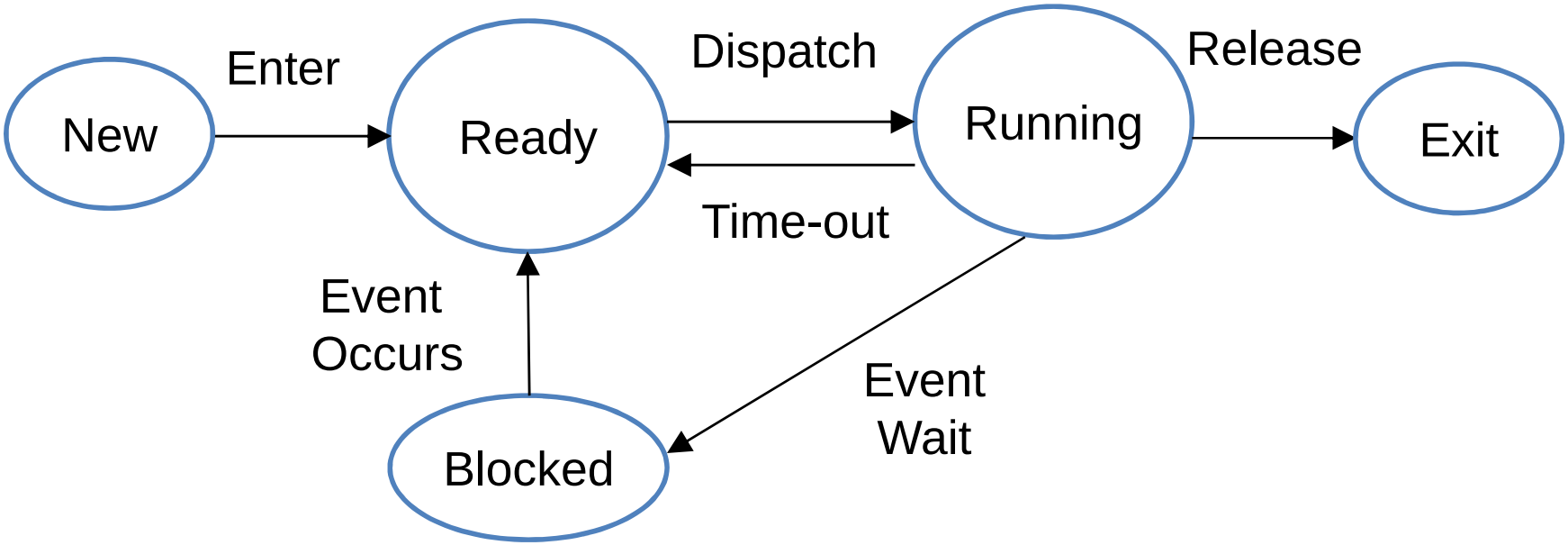
3. Required Memory Unavailable
4. Bounds Violation: The process tries to access memory locations that it is not allowed to access.
5. Protection Error: Process violates protection policies
6. Arithmetic Error: The process tries a prohibited computation
7. Time Overrun: The process has waited longer than specified time for an event to occur
8. An I/O Failure, such as:
   - inability to find a file
   - failure to read or write after specified number of tries
   - invalid operation

The Creation and Termination of Processes

9. Invalid Instruction: The process attempts to execute a non-existent instruction
10. Privileged Instruction: The Process attempts to use an instruction reserved for the operating system.
11. Use of invalid data
12. User or the operating system has terminated the process
13. Parent Termination: When a parent terminates, the operating system may be designed to automatically terminate all the offspring of that parent.
14. Parent Request: A parent process typically has the authority to terminate any of its spawn.

Five-State Process Model

- The Not-Running state is further divided into:
  - Ready State: Processes are ready for execution
  - Blocked State: Processes waiting an I/O operation or some event to occur
- Dispatcher selects processes for execution form ready queue
- Increases efficiency of system

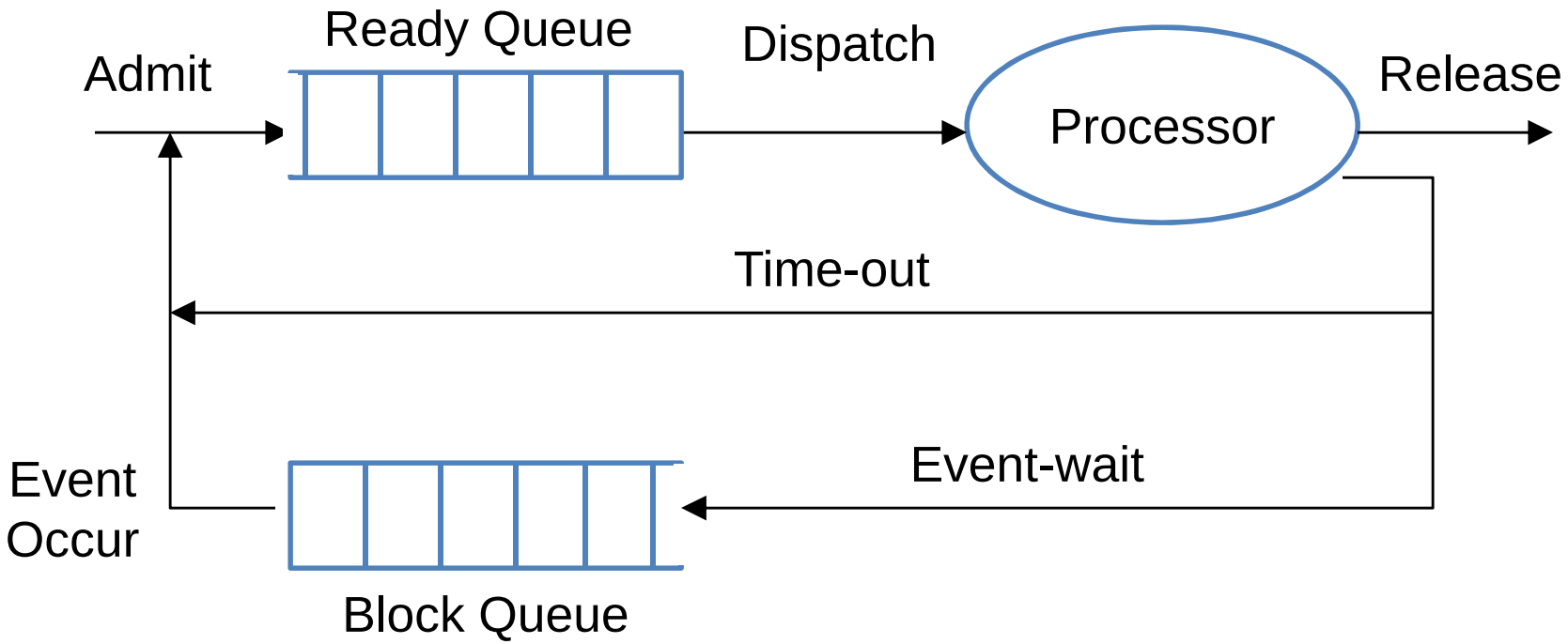


A Five-State Process Model

The five states in this model are

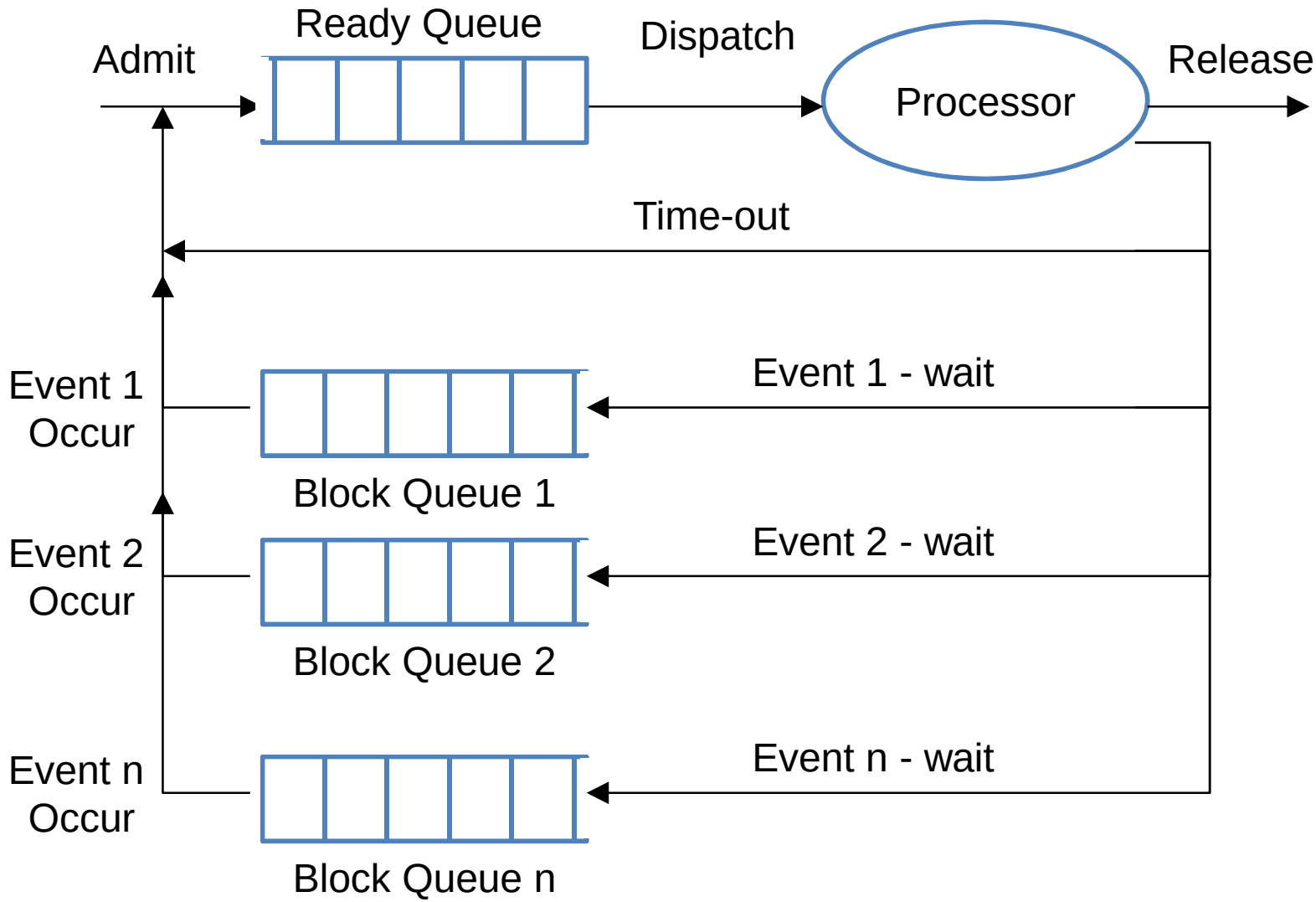
- **Running:** The process is currently being executed.
- **Ready:** Processes that prepared to execute when given opportunity.
- **Blocked:** A process that cannot execute until some event occurs
- **New:** A newly created process that has not yet been admitted to the pool of executable processes by the OS
- **Exit:** A process that has been released from the execution cycle due because it is halted or aborted

Five State Process Model



Queuing diagram for 5-state process model

Five State Process Model



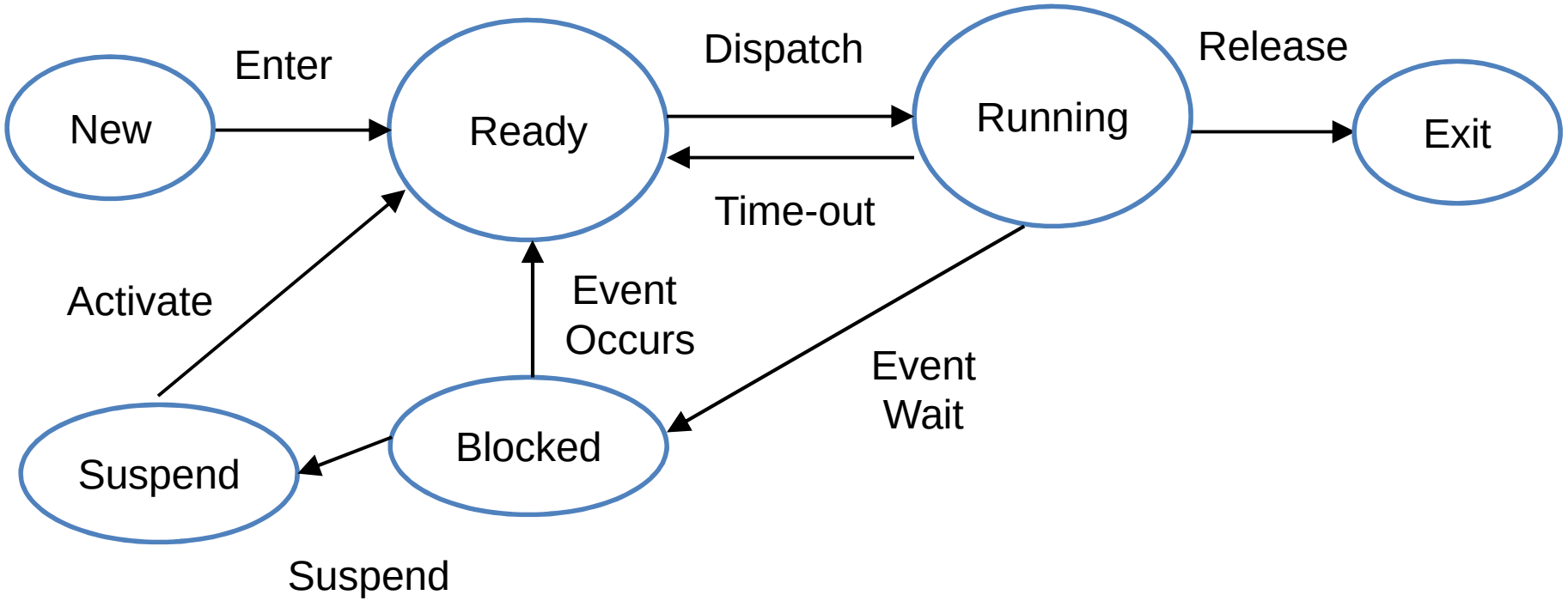
Queuing diagram with multiple blocked queue

Suspended Processes

The Need for Swapping

- The blocked state doesn't solve problem completely
- Memory holds multiple processes and the processor can move to another process when one process is waiting.
- While, processor is so much faster than I/O which is common for all processes residing in memory
- Thus even with multi-programming, a processor could be idle most of the time
- It will be better to perform swapping (virtual memory) among processes

Suspended State



Process state model with one suspend state

Suspended Processes

- Choices to bring a process into main memory:
  1. admit a newly created process
  2. bring in a previously suspended process.
- Previously suspended processes should have higher priority than newly created processes
- In other case, the load on system may increase

Difficulty

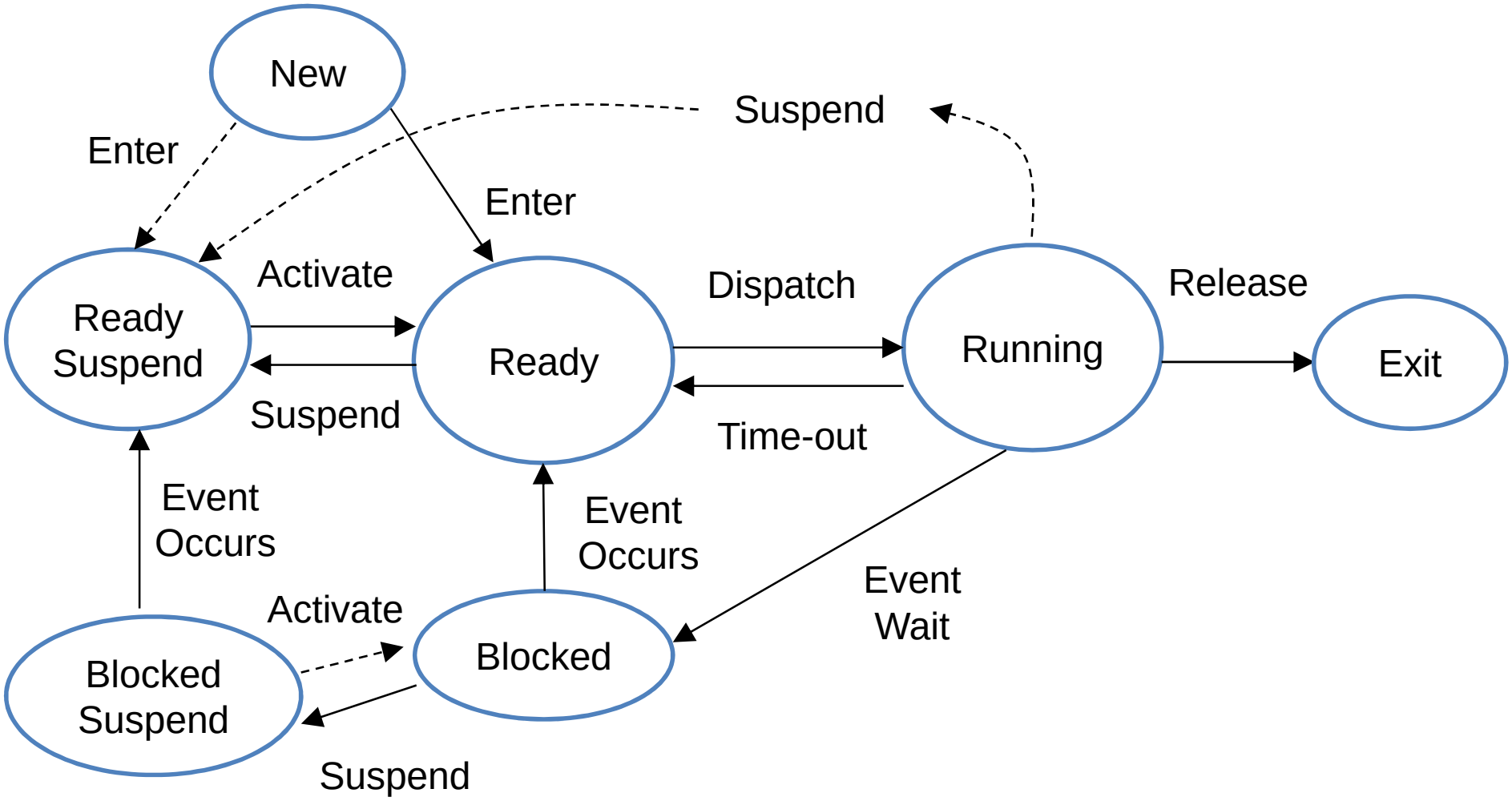
- All the processes that have been suspended were in the Blocked state at the time of suspension
- Bring a suspended process back into main memory will decrease throughput (process might be still waiting)
- Process transits from blocked/suspended state to Ready state particular event occurs and is potentially available for execution.

Suspended Processes

Therefore, there is need of an axillary state, as:

- Ready: The process is in main memory and available for execution.
- Blocked: The process is in main memory and awaiting an event.
- 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.

Seven State Process Model



Process state model with two suspend states

Reasons of Suspension

- Swapping: OS need to release sufficient main memory to bring in a process that is ready to execute,
- Interactive user request
- Timing: Process may be executed periodically and may be suspended while waiting for the next time interval.
- Parent process request
- Other OS reason: The OS may suspend a background or utility process or a process that is suspected of causing a problem.

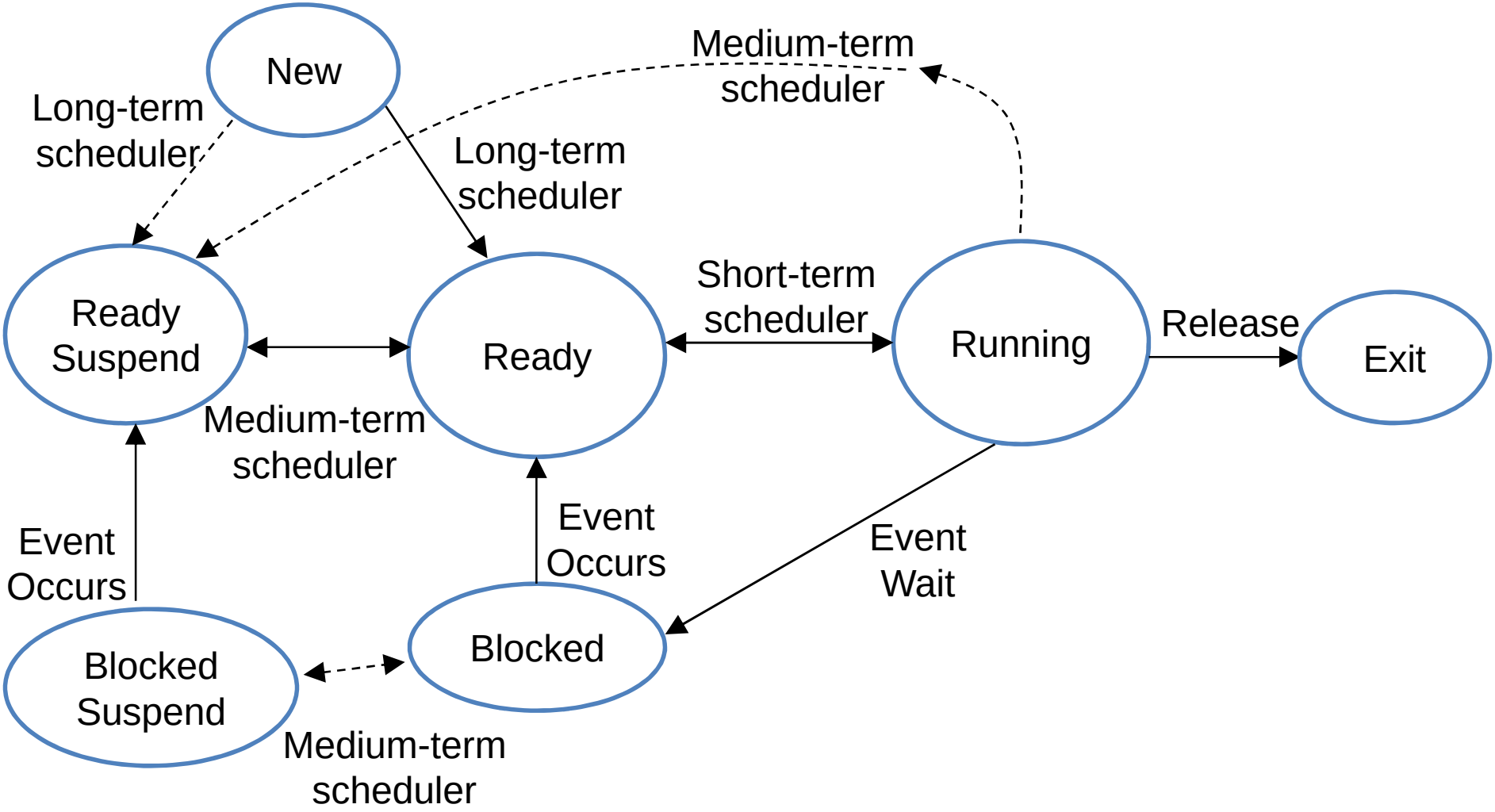
Process Scheduling

- Determines the processes to run when there are multiple run-able processes.
- Efficient resource utilization and increase in overall performance of the system

Types of Scheduling

- The scheduling components (schedulers) are categorized w.r.t relative frequency of their usage:
  - Long-term scheduler: Decides that whether or not to take on a new process and which one to take.
  - Medium term scheduler: Responsible for swapping decision.
  - Short-term scheduler or dispatcher: decides that which process should execute next.

Process Scheduling



Scheduling and Process state transitions

Scheduling Criteria

(1) User-Oriented, Performance-Related Criteria

- Response Time: Time between submission and first response of request. Technique should achieve low response time and maximize the number of interactive user processes receiving acceptable response time.
- Turnaround Time: Time interval (average time!) between the submission & completion of a process, i.e. actual execution time plus waiting time
- Deadlines: The scheduler should meet maximum percentage of deadline policies if defined.

(2) User-Oriented, Other Criteria

- Predictability: The technique minimizes the waiting time regardless of the load to execute a processes in expected time.

Scheduling Criteria

(3) System-Oriented, Performance-Related Criteria

- Throughput: Number of processes completed per unit time. Depends on the average length of a process, but is also influenced by the scheduling policy.
- Processor-Utilization: Time in percentage that the processor is busy. Technique should focus to increase this measure.

(4) System-Oriented, Other Criteria

- Fairness: All processes should be treated similarly, if policy isn't provided or processes of same policy should be treated in same way
- Enforcing Priorities: Scheduling policy should favor higher-priority processes if priority is assigned/provided
- Balancing Resources: Processes which uses busy resources should be favored

Questions