# Process Scheduling

- Process scheduling is an essential part of a Multiprogramming operating systems. Such operating systems allow more than one process to be loaded into the executable memory at a time and the loaded process shares the CPU using time multiplexing.
- Multiprogramming system's objective is to allow processes run all the time so that CPU utilization is maximized. With CPU switching back and forth among the processes, the rate at which a process performs its computation will not be uniform.

- The scheduling mechanism is the part of the process manager that handles the removal of the running process from the CPU and the selection of another process on the basis of particular strategy.
- Aim is to assign processes to be executed by the processor in a way that meets system objectives, such as response time, throughput, and processor efficiency
- For a single-processor system, there will never be more than one running process. If there are more processes, the rest will have to wait until the CPU is free and can be rescheduled.

- Process may be in one of two states: Running and Not Running.
- When a new process is created by OS,  that process enters into the system in the running state. Processes that are not running are kept in queue, waiting their turn to execute. Queue is implemented by using linked list.

Use of dispatcher is as follows:

- When a process is interrupted, that process is transferred to the waiting queue.
- If the process has completed or aborted, the process is discarded.
- In either case, the dispatcher then selects a process from the queue to execute.

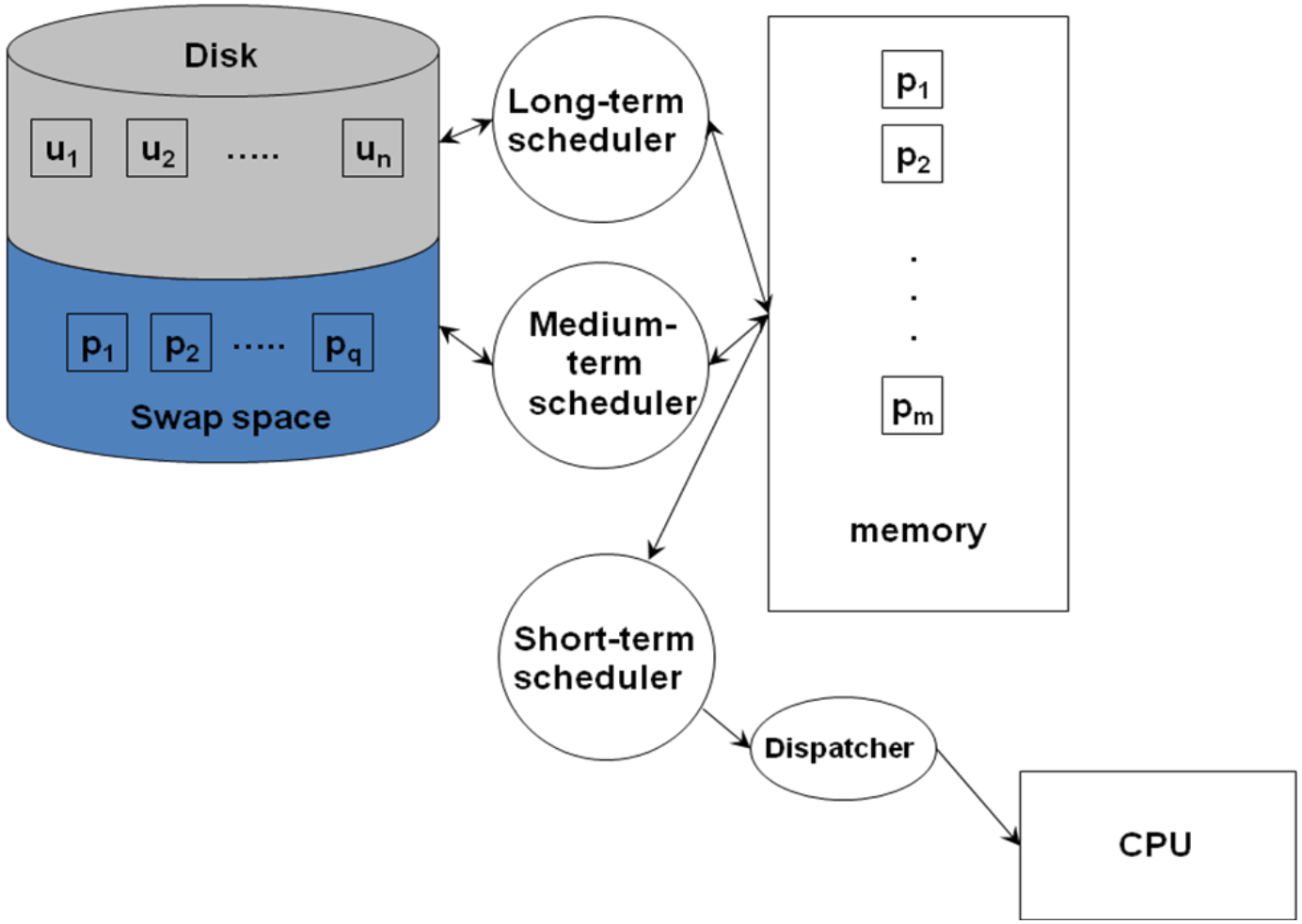
# When is Scheduling needed?

- When a new process is created
- When a process exits
- When a process is blocked  and waiting (on an IO device, a semaphore)
- When an I/O interrupt occurs

# Schedulers

- Schedulers are special system software which handle process scheduling in various ways. Their main task is to select the jobs to be submitted into the system and to decide which process to run.
- Schedulers are of three types
  - Long-Term Scheduler
  - Short-Term Scheduler
  - Medium-Term Scheduler

# Scheduling Environments

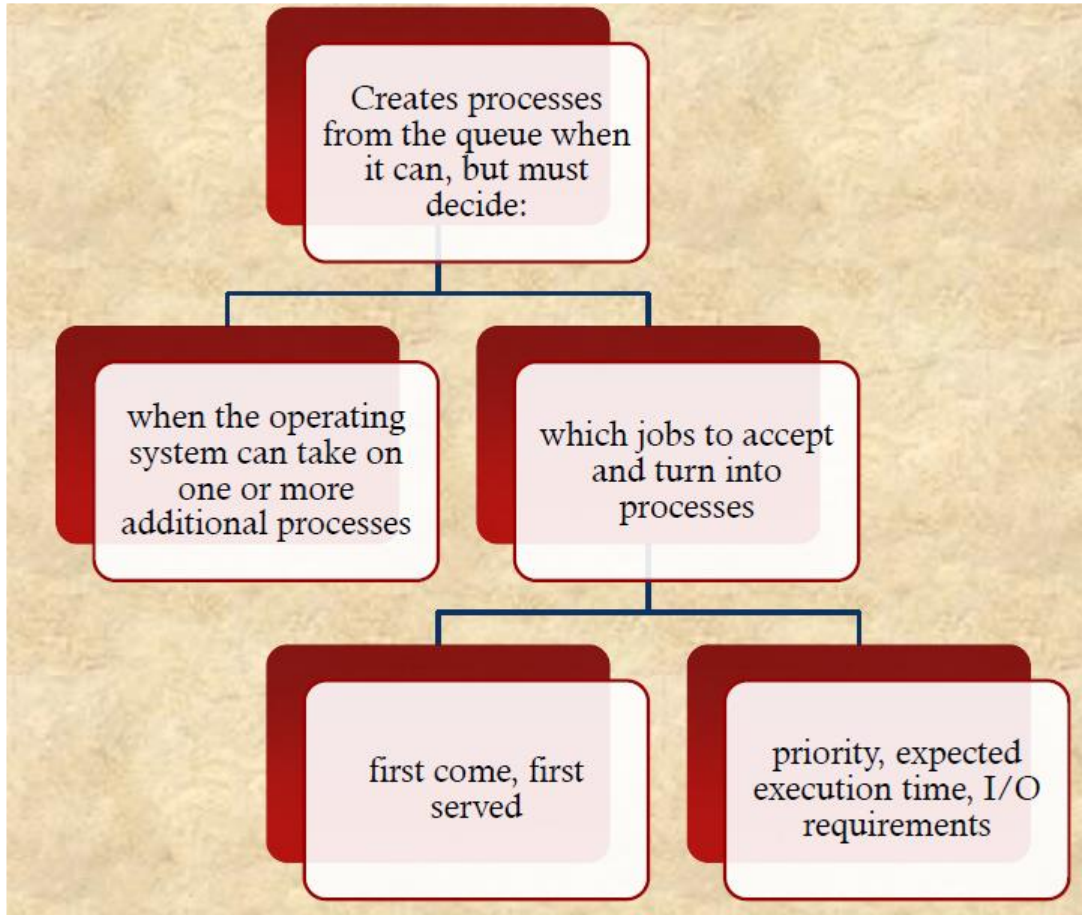


# Types of Scheduling

| | |
|---|---|
| **Long-term scheduling** | The decision to add to the pool of processes to be executed |
| **Medium-term scheduling** | The decision to add to the number of processes that are partially or fully in main memory |
| **Short-term scheduling** | The decision as to which available process will be executed by the processor |
| **I/O scheduling** | The decision as to which process's pending I/O request shall be handled by an available I/O device |

# Long-Term Scheduler

- Determines which programs are admitted to the system for processing
- Controls the degree of multiprogramming
- the more processes that are created, the smaller the percentage of time that each process can be executed
- may limit to provide satisfactory service to the current set of processes


Creates processes from the queue when it can, but must decide:
when the operating system can take on one or more additional processes
which jobs to accept and turn into processes
first come, first served
priority, expected execution time, I/O requirements

# Medium-Term Scheduling

- Part of the swapping function
- Swapping-in decisions are based on the need to manage the degree of multiprogramming
- considers the memory requirements of the swapped-out processes

# Short-Term Scheduling

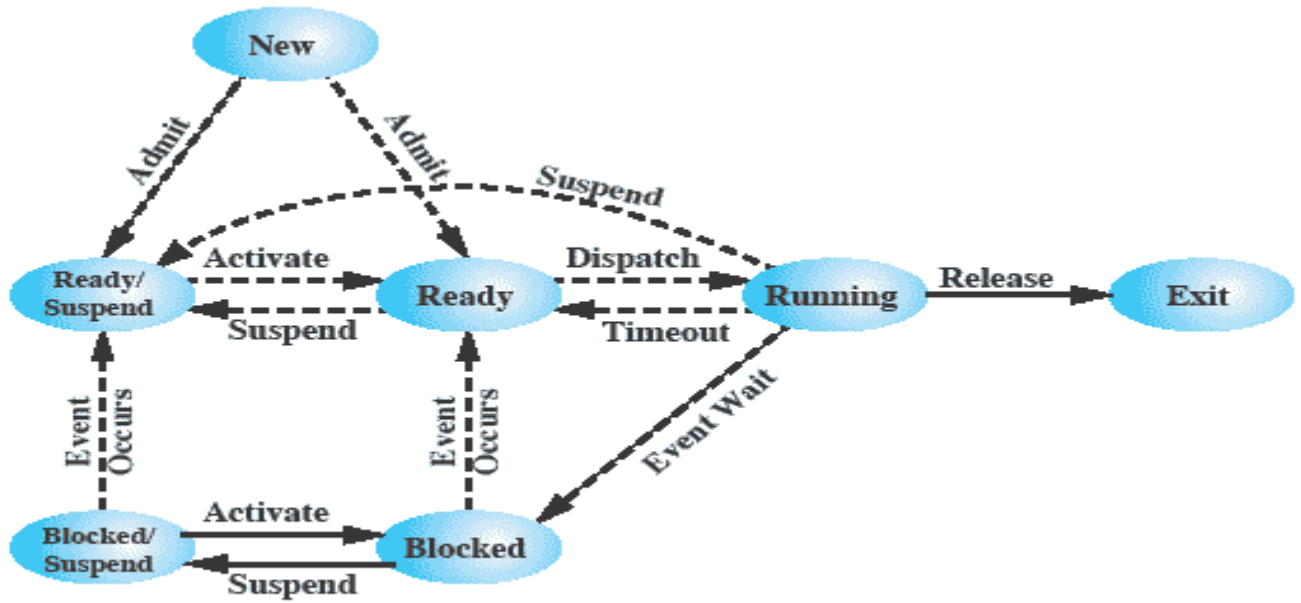
- Known as the dispatcher
- Executes most frequently
- Makes the fine-grained decision of which process to execute next
- Invoked when an event occurs that may lead to the blocking of the current process or that may provide an opportunity to preempt a currently running process in favor of another
- Examples:
  - Clock interrupts
  - I/O interrupts
  - Operating system calls
  - Signals (e.g., semaphores)

# Short Term Scheduling Criteria

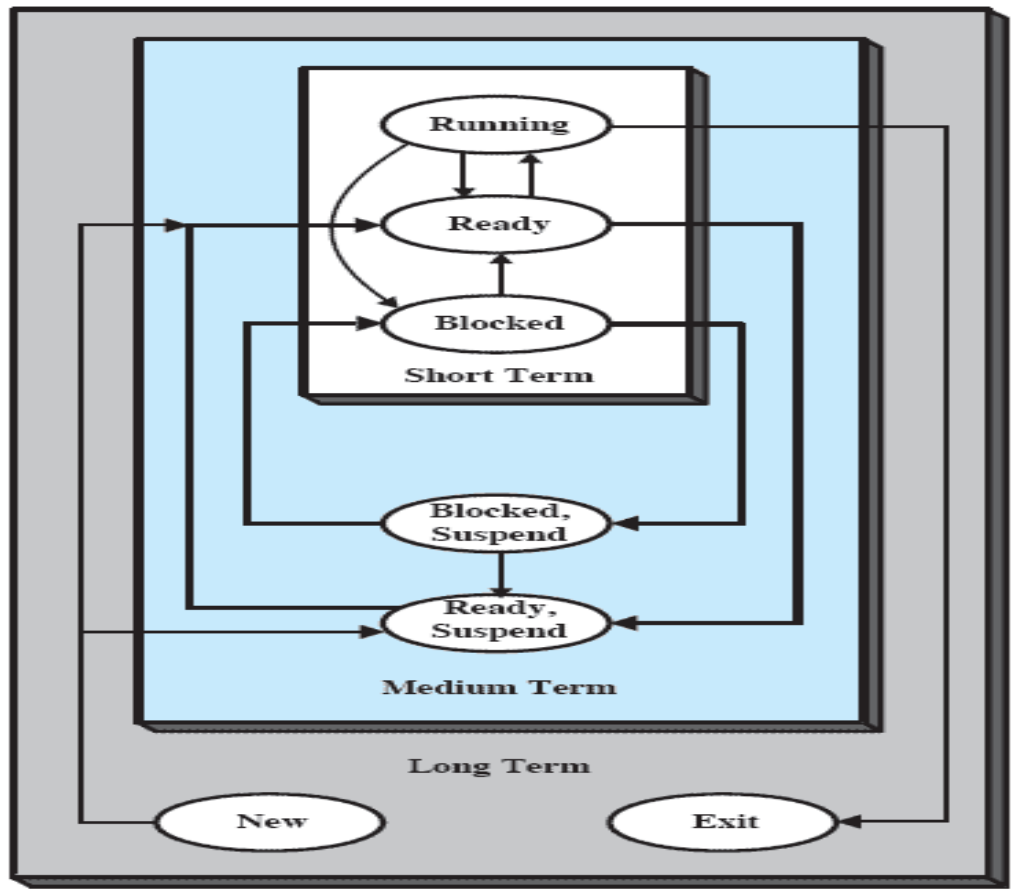
- Main objective is to allocate processor time to optimize certain aspects of system behavior
- A set of criteria is needed to evaluate the scheduling policy
- User-oriented criteria
  - relate to the behavior of the system as perceived by the individual user or process (such as response time in an interactive system)
  - important on virtually all systems
- System-oriented criteria
  - focus in on effective and efficient utilization of the processor (rate at which processes are completed)
  - generally of minor importance on single-user systems

| S.N. | Long-Term Scheduler | Short-Term Scheduler | Medium-Term Scheduler |
|---|---|---|---|
| 1 | It is a job scheduler | It is a CPU scheduler | It is a process swapping scheduler. |
| 2 | Speed is lesser than short term scheduler | Speed is fastest among other two | Speed is in between both short and long term scheduler. |
| 3 | It controls the degree of multiprogramming | It provides lesser control over degree of multiprogramming | It reduces the degree of multiprogramming. |
| 4 | It is almost absent or minimal in time sharing system | It is also minimal in time sharing system | It is a part of Time sharing systems. |
| 5 | It selects processes from pool and loads them into memory for execution | It selects those processes which are ready to execute | It can re-introduce the process into memory and execution can be continued. |

# Scheduling and Process State Transitions




(b) With Two Suspend States

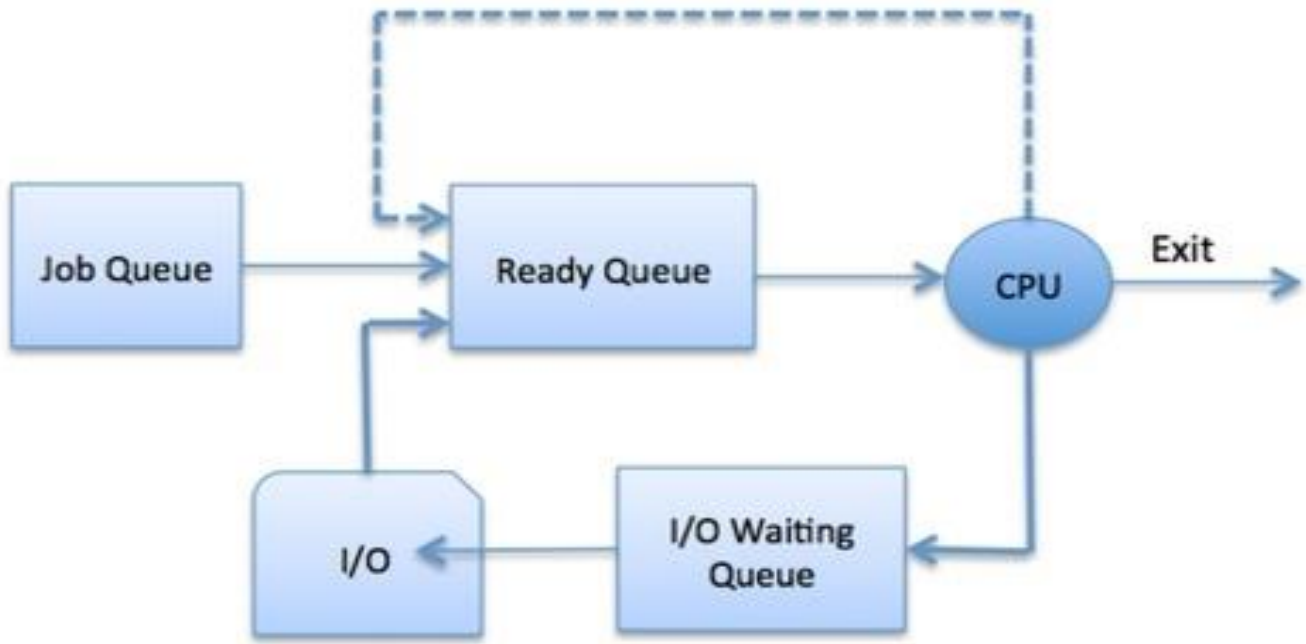

Running
Ready
Blocked
Short Term
Blocked,
Suspend
Ready,
Suspend
Medium Term
Long Term
New
Exit

# Process Scheduling Queues

- The OS maintains all PCBs in Process Scheduling Queues.
- The OS maintains a separate queue for each of the process states and PCBs of all processes in the same execution state are placed in the same queue.
- When the state of a process is changed, its PCB is unlinked from its current queue and moved to its new state queue.

The Operating System maintains the following important process scheduling queues

- **Job queue** − This queue keeps all the processes in the system.
- **Ready queue** − This queue keeps a set of all processes residing in main memory, ready and waiting to execute. A new process is always put in this queue.
- **Device queues** − The processes which are blocked due to unavailability of an I/O device constitute this queue.


Job Queue
Ready Queue
CPU
Exit
I/O
I/O Waiting Queue

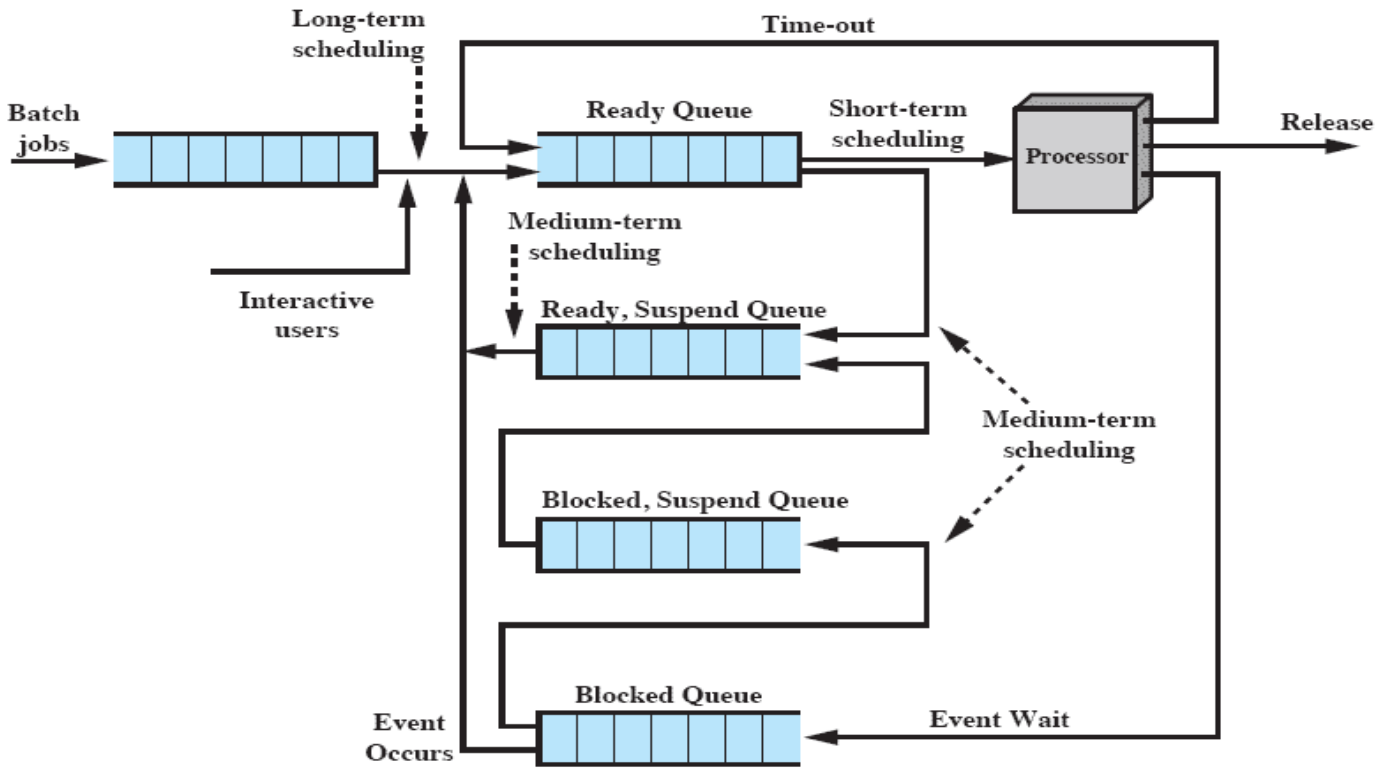



**Figure 9.3 Queuing Diagram for Scheduling**