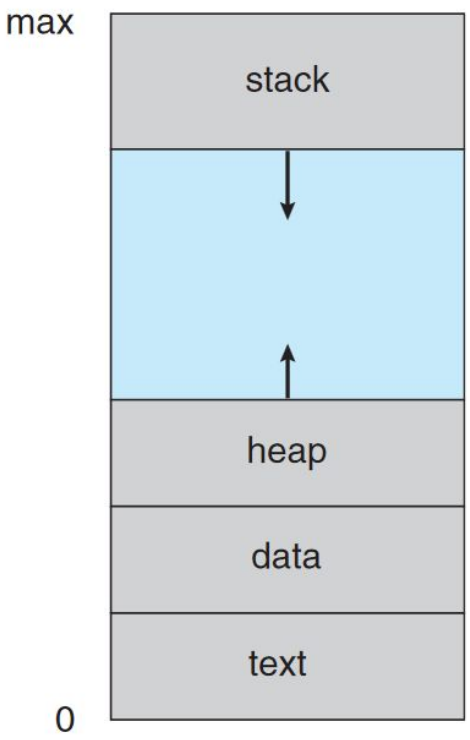


Figure 3.1 Process in memory.

Process States

- **New.** The process is being created.
- **Running.** Instructions are being executed.
- **Waiting.** The process is waiting for some event to occur (such as an I/O completion or reception of a signal).
- **Ready.** The process is waiting to be assigned to a processor.
- **Terminated.** The process has finished execution.

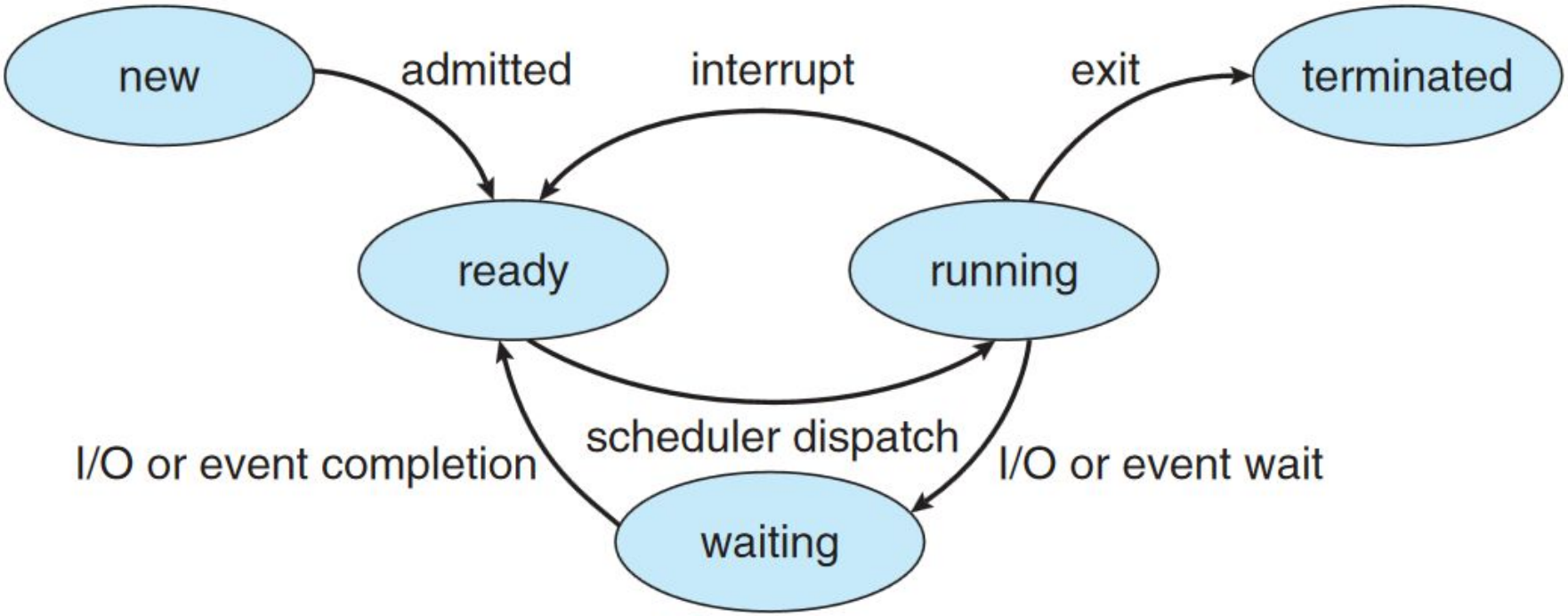


Figure 3.2 Diagram of process state.

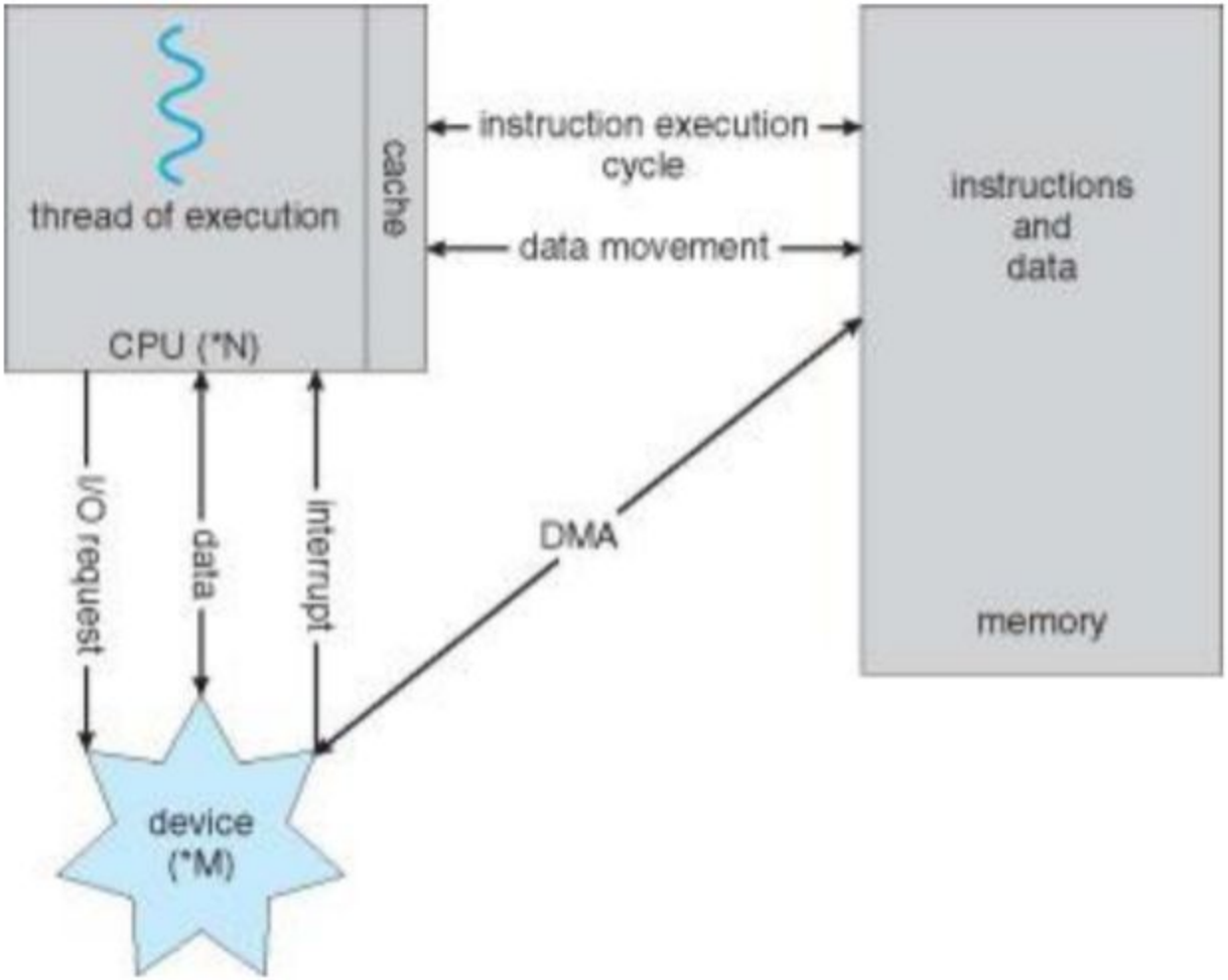
Device Controller

- The device controller is responsible for moving the data between the peripheral devices that it controls and its local buffer storage.
- Typically, operating systems have a **device driver for each device controller.**
- Whenever a process needs to read data from a file in the disk. The OS will instruct (via device driver) the disk controller to read the appropriate file.

Device Controller

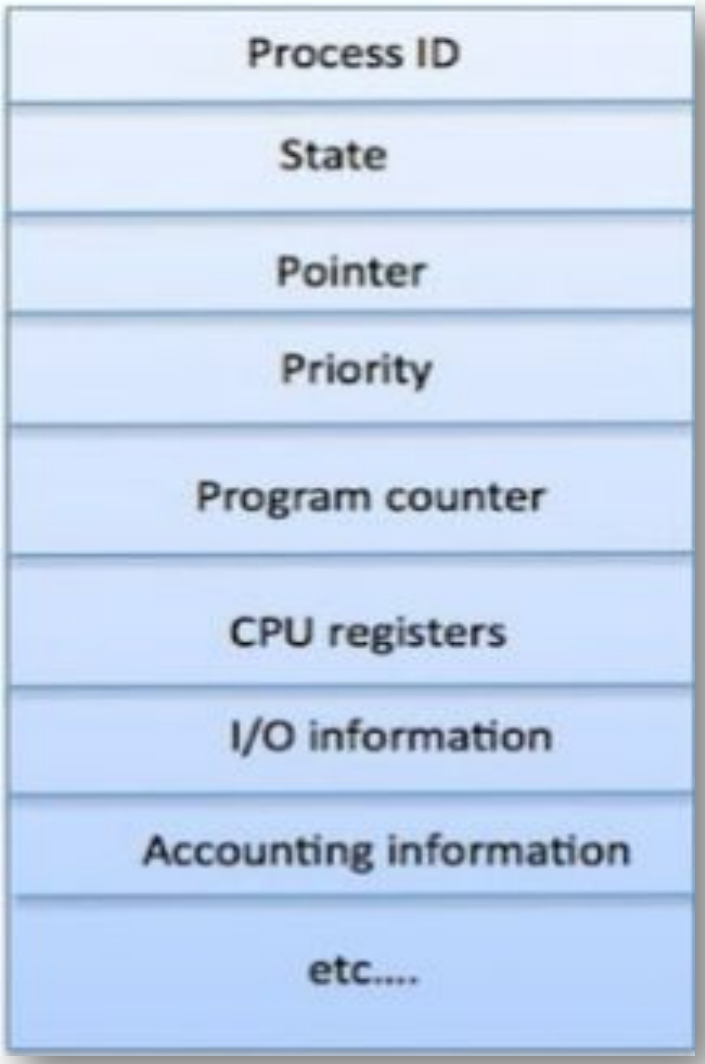
- The data will be read by the disk driver and stored in its buffer memory.
- Once the operation is complete, an interrupt will be generated to inform the system that data is ready.
- The CPU will then transfer the data from controller's buffer to RAM.
- CPU is still being wasted in this method in reading the data.
- Some architectures allow controllers to transfer the data directly to RAM. It is known as **Direct Memory Access (DMA)**.
- While the device controller is transferring the data to RAM, the CPU is available to accomplish other work.

Device Controller

- 

PCB (Process Control Block)

1. Process state
2. Program counter
3. CPU registers
4. CPU-scheduling information
5. Memory-management information
6. Accounting information
7. I/O status information


Process ID
State
Pointer
Priority
Program counter
CPU registers
I/O information
Accounting information
etc....

Process Queues Maintained by the Kernel

Job Queue:
As processes enter the system, they are put into a job queue. This queue consists of all processes in the system.

Ready Queue:
The processes that are residing in main memory and are ready and waiting to execute are kept on a list called the ready queue. This queue is generally stored as a linked list. A ready-queue header contains pointers to the first and final PCBs in the list. Each PCB is extended to include a pointer field that points to the next PCB in the ready queue.

Device Queue:
When a process is allocated the CPU, it executes for a while, and eventually quits, is interrupted or waits for a particular event, such as completion of an I/O request. In the case of an I/O request, the device may be busy with the I/O request of some other process, hence the list of processes waiting for a particular I/O device is called a device queue. Each device has its own device queue.

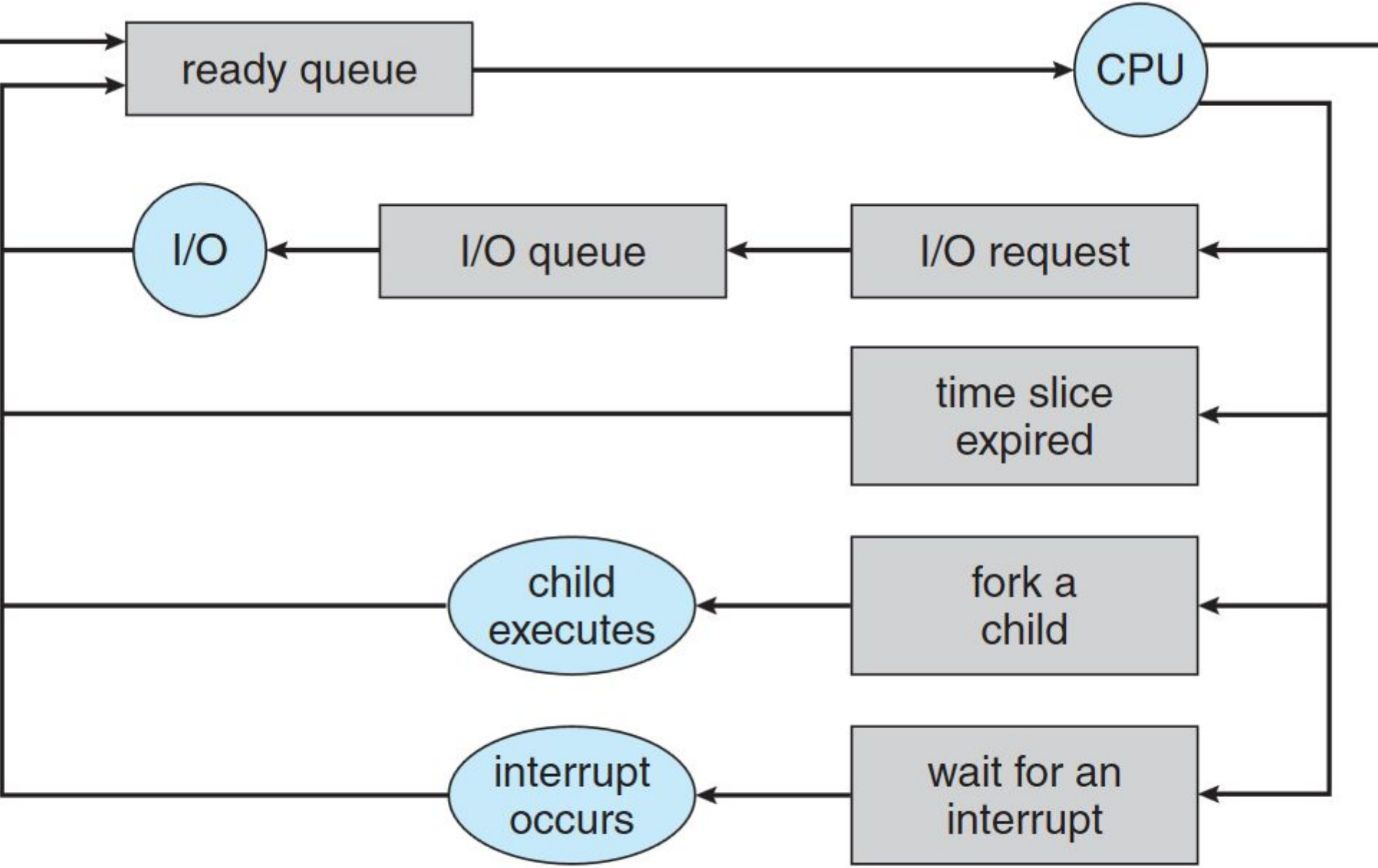


Figure 3.6 Queueing-diagram representation of process scheduling.

CPU Scheduler

- The short-term scheduler, or CPU scheduler, selects from among the processes that are ready to execute and allocates the CPU to one of them

Job Scheduler (Long Term Scheduler)

- More processes are submitted than can be executed immediately. These processes are spooled to a mass-storage device (typically a disk), where they are kept for later execution.

- The long-term scheduler, or job scheduler, selects processes from this pool and loads them into memory for Process Scheduling execution.

- In short, Job Scheduler decides which processes that have been submitted can be in memory and hence in the ready queue.

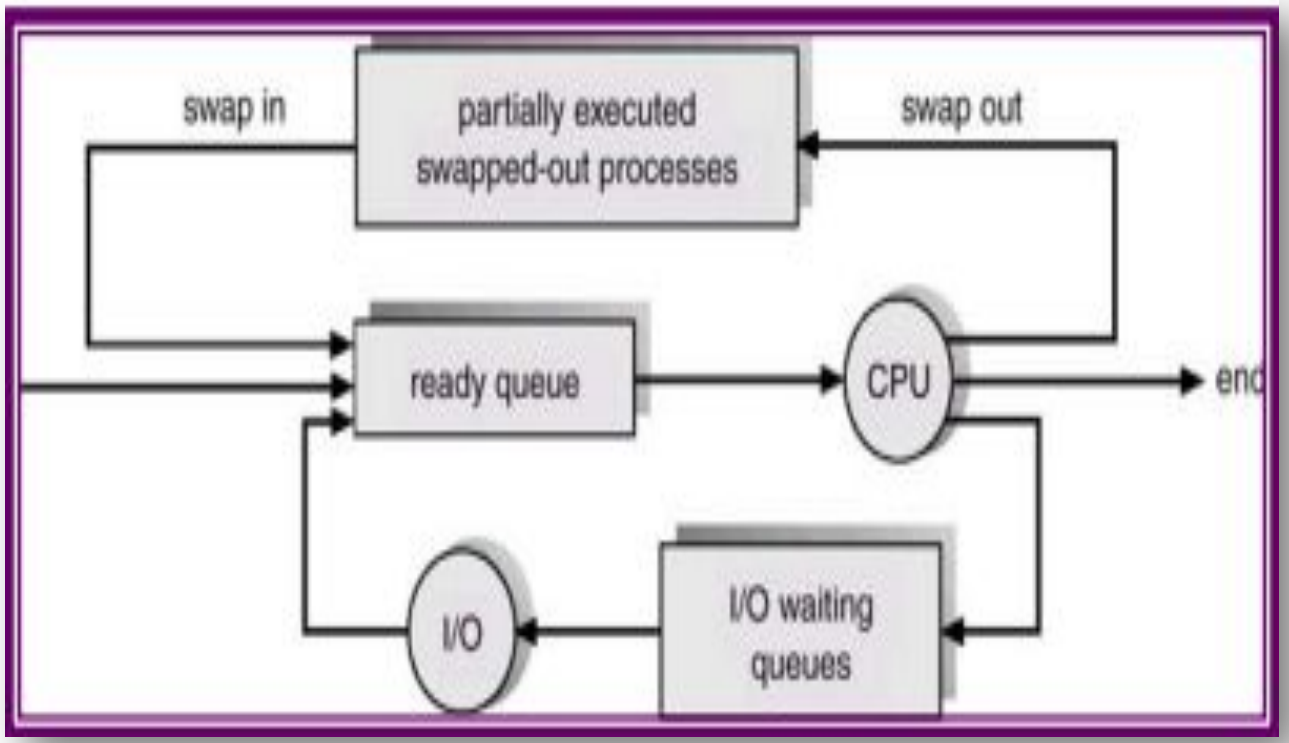
- CPU scheduler runs more often than job scheduler.

Other Types of Schedulers

- I/O Schedulers such as disk scheduler

Other Types of Schedulers

- Some operating systems such as time-sharing systems may introduce a medium-term scheduler, which removes processes from memory and thus reduces the degree of multiprogramming.
- At some later time the process can be reintroduced at some later stage, this scheme is called swapping.
- The process is swapped out and is later swapped in by the medium-term scheduler.
- Swapping may be necessary to improve the job mix, or because a change in memory requirements has over-committed available memory, requiring memory to be freed up.
- The work carried out by the swapper to move a process from the main memory to disk is known as swap out and moving it back into the main memory is called swap in.
- The area on the disk where swapped-out processes are stored is called the swap space.


swap in
partially executed
swapped-out processes
swap out
ready queue
CPU
end
I/O
I/O waiting
queues

I/O-Bound Process vs CPU-Bound Process

- I/O-bound process is one that spends more of its time doing I/O than it spends doing computations.
- A CPU-bound process, in contrast, generates I/O requests infrequently, using more of its time doing computations.

It is important that the long-term scheduler select a good process mix of I/O-bound and CPU-bound processes.

- If all processes are I/O bound, the ready queue will almost always be empty, and the short-term scheduler will have little to do.
- If all processes are CPU bound, the I/O waiting queue will almost always be empty, devices will go unused, and again the system will be unbalanced.
- The system with the best performance will thus have a combination of CPU-bound and I/O-bound processes.

Context Switch

- Context Switch is a method in which the CPU state of currently executing process is saved in the PCB and the state of another process in the ready queue is loaded in the CPU from the PCB.
- Context switching is done by the dispatcher.
- Context switching also wastes some time of the CPU.

Interactive Process vs Non-interactive Process

- An interactive process is a process which communicates with the user frequently via I/O devices.
- A non-interactive process runs in the background without any user intervention.

- Generally, interactive processes are given more priority.