Process Management

Multiprogramming vs Multiprocessing

- **Multiprogramming:** Running more than 1 processes at the same time on a single CPU. It achieves concurrency.
- More than 1 program are kept in the main memory.
- CPU is assigned to one of these processes at a time for certain amount of time. After that, the CPU is assigned to another process.

Multiprogramming vs Multiprocessing

- CPU is also assigned to another process when the already executing process requests for an I/O operation.

Multiprocessing: Using more than one cpu to run multiple processes. It achieves parallelism.

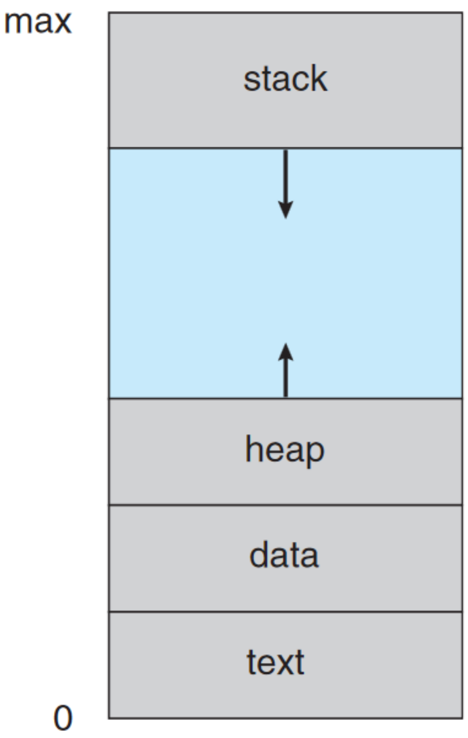

Figure 3.1 Process in memory.

Process Management

- Process management involves various tasks like process creation, process scheduling and termination of processes.

- A **process** is a program in execution.

- It decides which process should be given the CPU next.

- It assigns the CPU to a process for some amount of time. The process executes its instructions in that time.

Process Management

- After the allocated time has elapsed, the CPU manager takes the CPU from the process and gives the CPU to the next process.

- Deciding which process to select for CPU allocation is done by the **Scheduler**.

- After the process has been selected, the **context switching** takes place.

- **Context** refers to the data in the registers and program counter at a specific moment in time

- Context switching is done by the **dispatcher**.

Context Switch

- Context Switch is a method in which the CPU state (values of registers) of currently executing process is saved in the PCB and the CPU 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.

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.

Process Queues Maintained by the Kernel

- **Ready Queue**

List of Processes that are ready to execute.

- **Device Queue /Waiting Queue**

The list of processes waiting for a particular I/O device is called a device queue. Each device has its own device queue

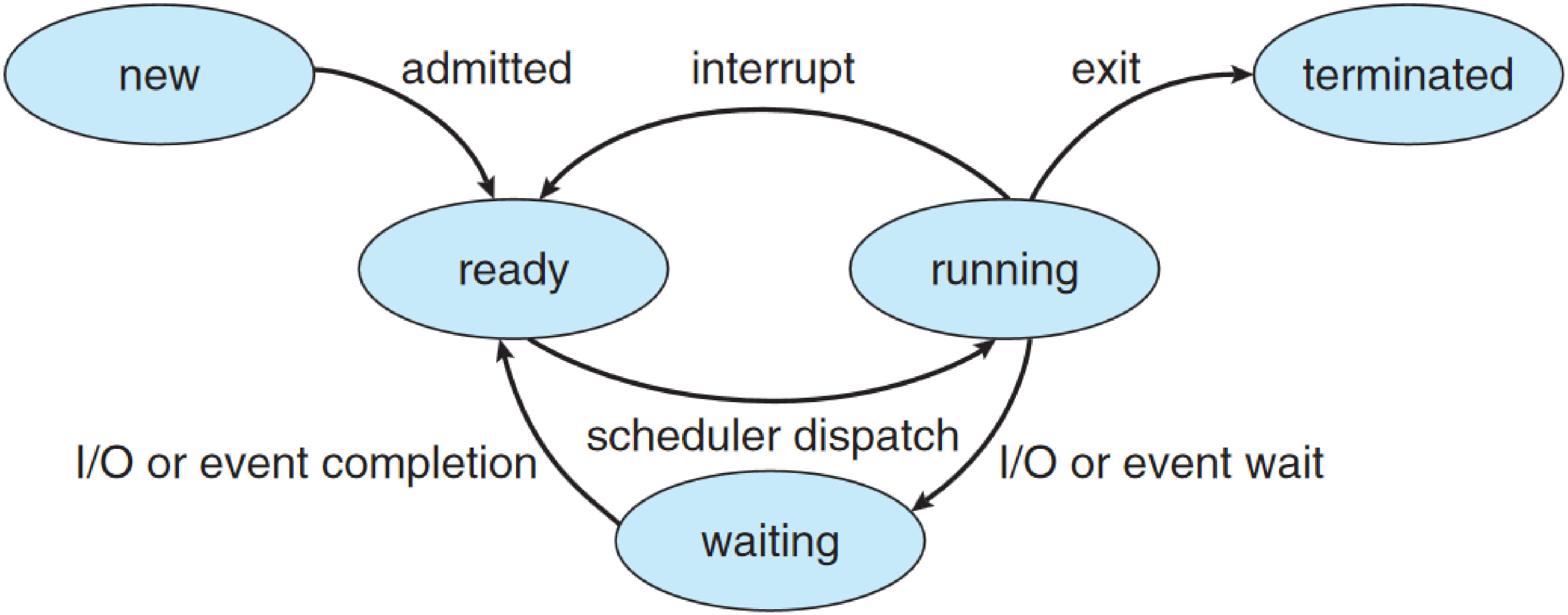

Figure 3.2 Diagram of process state.

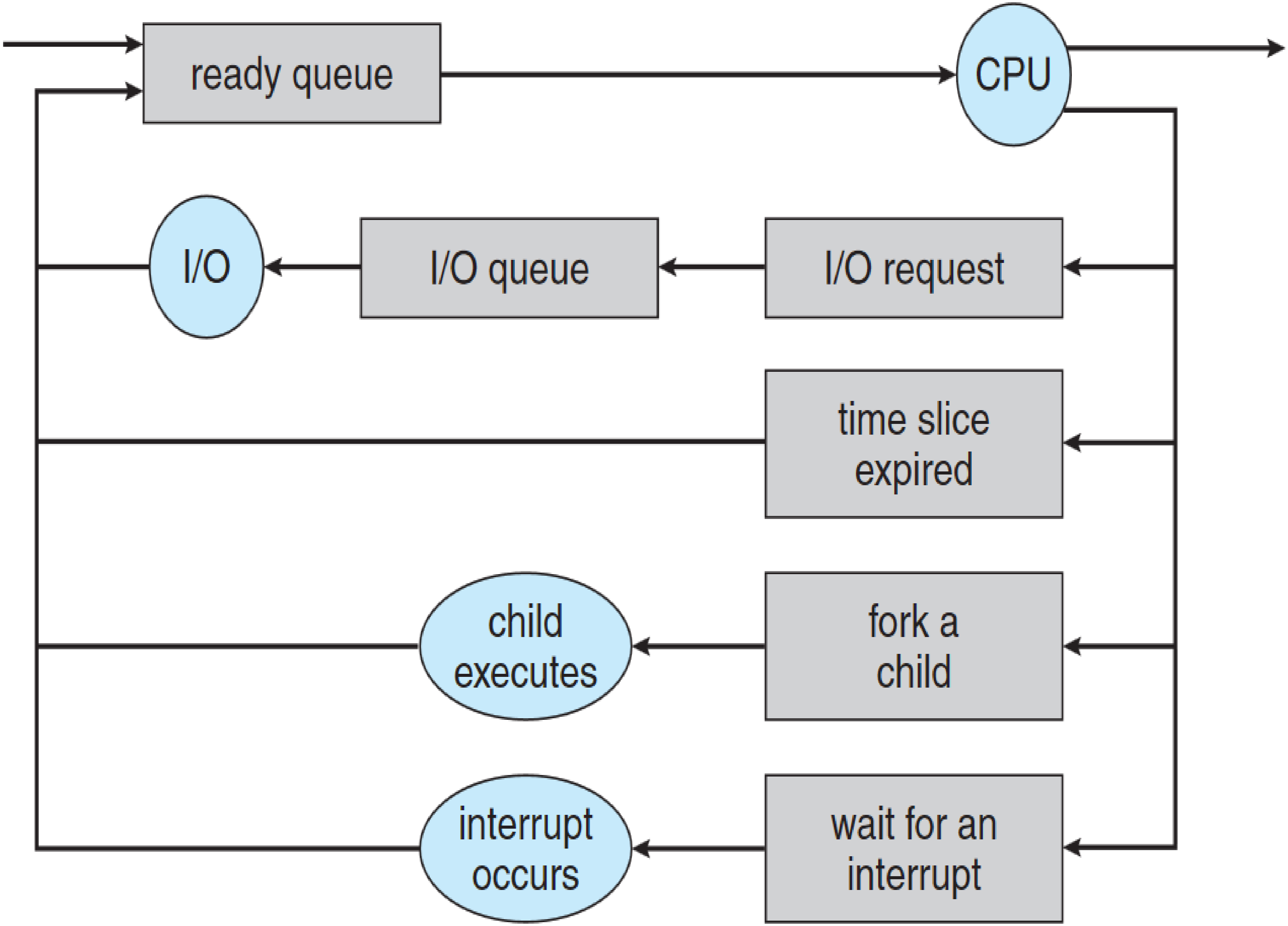

Figure 3.6 Queueing-diagram representation of process scheduling.

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

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

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, and spends 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.

I/O-Bound Process vs CPU-Bound Process

- 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. Hence the CPU will be idle most of the time.

- 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.

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.