Processes ch3

By

Lecturer Ameen A.Noor

Process Concepts

- A process is a program in execution. A process is more than the program code, which is sometimes known as the text section. It also includes the current activity, as represented by the value of the program counter and the contents of the processor's registers.

Process State

- The state of a process is defined in part by the current activity of the process. Each process may be in one of the following states:
  - **new**: The process is being created
  - **running**: Instructions are being executed
  - **waiting**: The process is waiting for some event to occur
  - **ready**: The process is waiting to be assigned to a processor
  - **terminated**: The process has finished execution

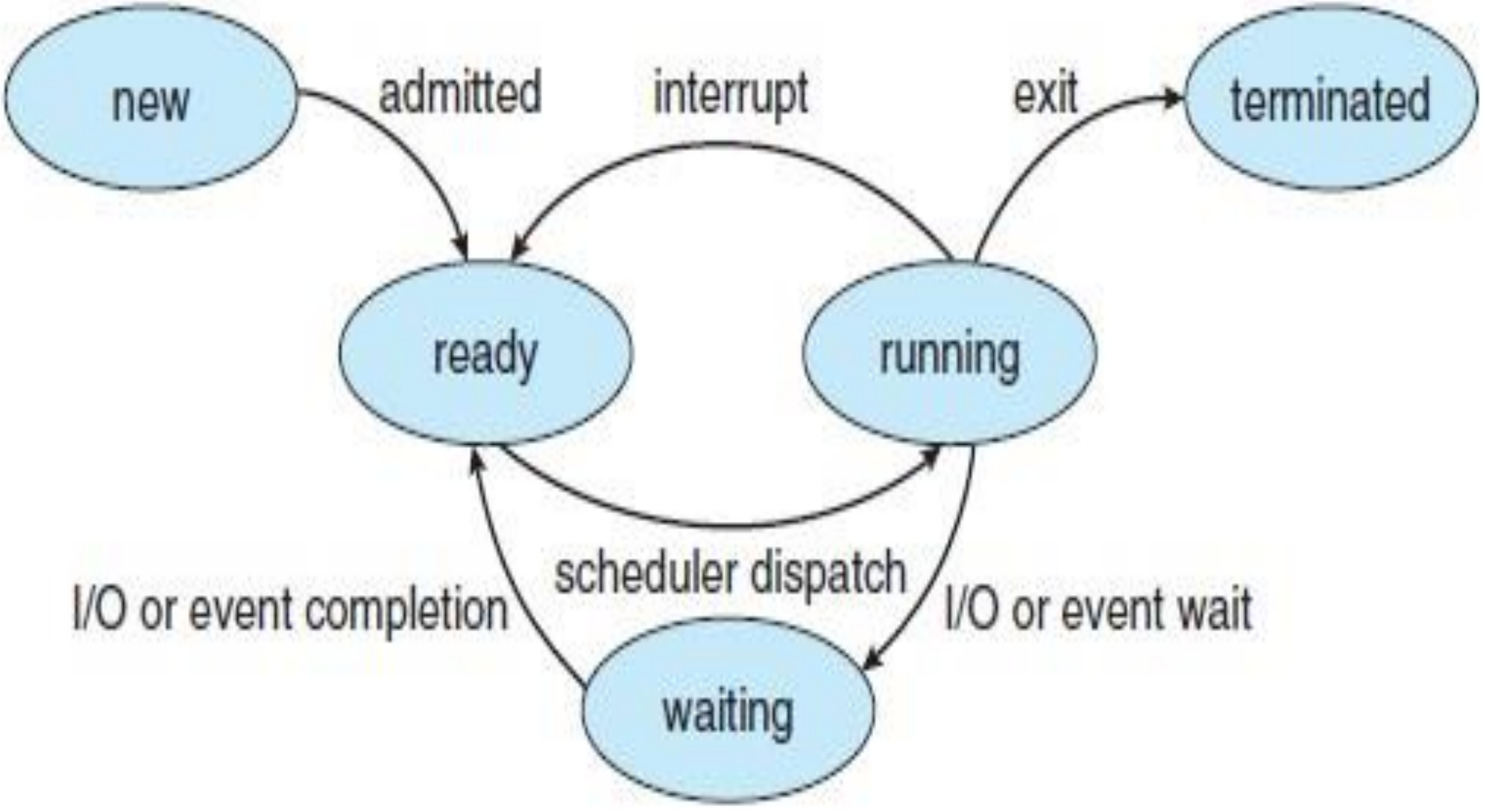


Figure 1: Diagram of Process State

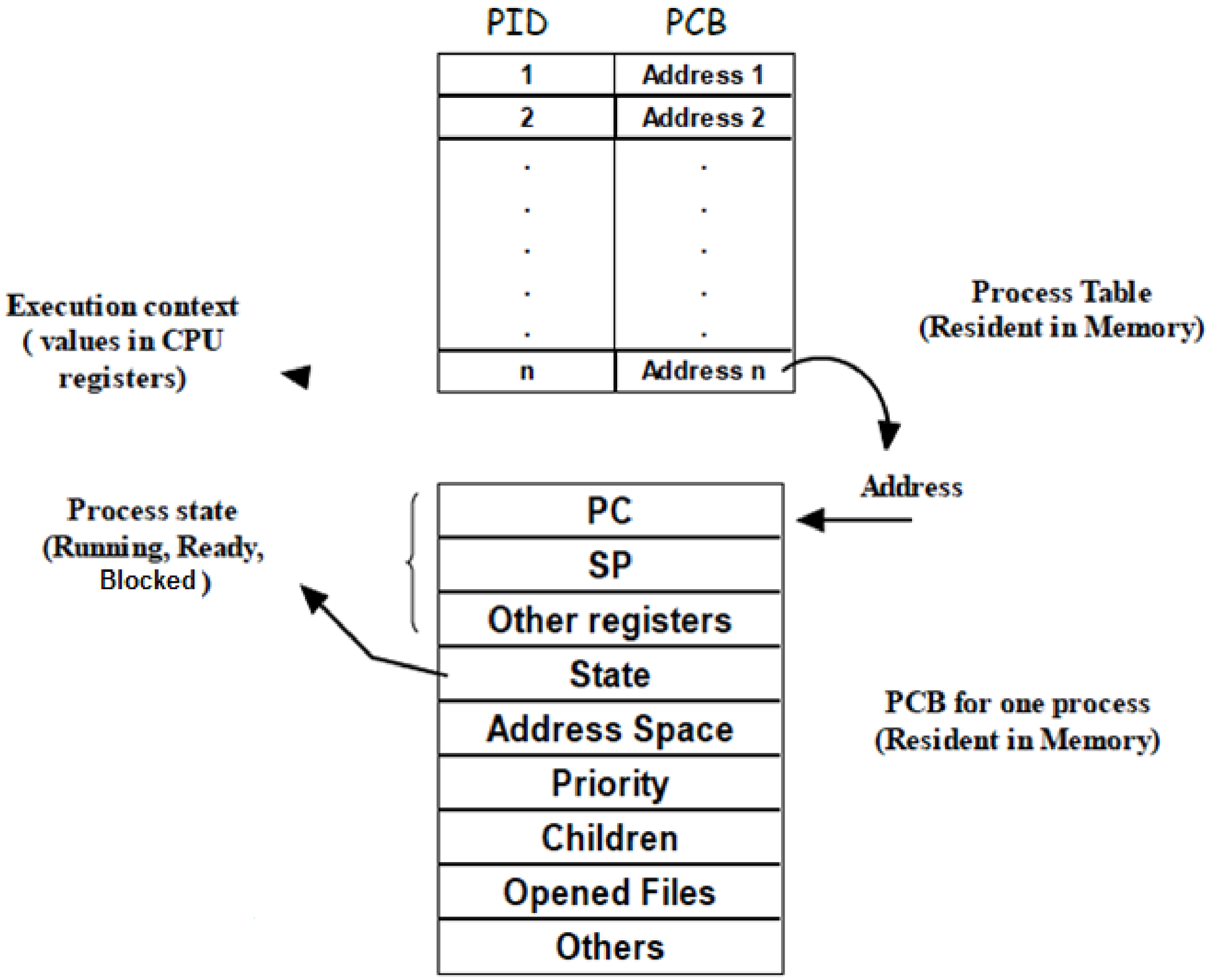
1. At any instant of time, there is only one process running i.e allocated CPU time.
2. Exit from Running state may occur as a result of any of following events:
   - Completion of process.
   - Request of I/O service by a process.
   - Time slice determined by interval timer has expired and hence an interrupt is activated which forces CPU to run OS instructions.
3. The transfer from Ready to Running state (dispatch) is carried out by OS according to certain criteria as will be shown later when studying "Processor Scheduling".
4. The term "Execution" means generally, "Ready", "Running", or "waiting".

Process Control Block (PCB)

- Each process is represented by a process control block (PCB). PCB is a data structure describing a process and resident in memory each process has its own PCB in memory and assigned a process Identification Number (**PID**). The locations of PCBs is kept in a special table called "Process Table" which is resident in memory and used by O.S.

Process Control Block (PCB)

- A PCB contains many pieces of information associated with a specific process, such as:
  - Process states
  - Program counter
  - CPU registers
  - CPU scheduling information
  - Memory management information
  - Accounting information
  - I/O status information


PID
PCB
1
Address 1
2
Address 2
n
Address n
Process Table
(Resident in Memory)
Execution context
(values in CPU
registers)
Address
Process state
(Running, Ready,
Blocked)
PC
SP
Other registers
State
Address Space
Priority
Children
Opened Files
Others
PCB for one process
(Resident in Memory)

Process Scheduling

- A uniprocessor system can have only one running process. If more processes exist, as in multiprogramming system, there will be only one process running and the rest must wait until the CPU is free and can be rescheduled.

Scheduling Queues

- A new process as enter the system is put in a queue called ready queue. It waits in the ready queue until it is selected for execution. Once the process is assigned to the CPU and it is executing, one of the several event could occur:
  - The process could issue an I/O request, and then be placed in an I/O queue.
  - The process could create a new subprocess and wait for the termination.
  - The process could be removed forcibly from the CPU, as a result of an interrupt and be put back in the ready queue.

Scheduler

- A process migrates between the various scheduling queues throughout its lifetime. The operating system must select processes from these queues in some fashion. The selection process is carried out by the appropriate scheduler. There are two types of scheduling algorithms categorized according to the frequency of their execution.
  - Long term scheduler (job scheduler) which selects a process from the job pool and load them into the MM.
  - Short term scheduler (CPU scheduler) which select a process from the ready queue and allocate it to the CPU.

Context Switch

- Switching the CPU to another process requires saving the state of the old process and loading the saved state for the process. This task is known as a context switch.

Operation on Processes

- The process in the system can execute concurrently, and they must be created and deleted dynamically.

1. **Process Creation:**

   A process may create several new processes during the course of execution. The creating process is called a parent process, whereas the new processes are called the children. When a process is created it obtains various resources and initialization values that may be passed along from the parent process to the child process.

Operation on Processes

2. **Process Termination:**

   A process terminates when it finishes executing its final statement and asks the operating system to delete it. At that point the process may return data to its parent process and the OS deallocate all the physical and logical resources that are previously allocated to that process.

Cooperating Processes

- The concurrent process executing in the operating system may be either independent processes that does not share any data or cooperating that affects each other's.
- We may provide an environment that allows process cooperation for several reasons:
  - Information sharing
  - Computation speedup
  - Modularity
  - Convenience

Inter process Communication (IPC)

- The cooperating processes can communicate in a shared memory environment. The scheme requires that these processes share a common buffer pool. Another way to achieve the same effect for the operating system is provided via an interprocess communication (IPC).
- IPC provides a mechanism to allow processes to communicate and synchronize their actions without sharing the same address space. This technique is useful for distributed systems. IPC is provided by a message passing system.

Interprocess Communication (IPC)

- IPC is sometimes necessary but it presents two main problems:

1. Address violation problem: IPC means sharing some data (access common locations in memory). The shared data will be outside the address space of at least one process which, in turn, creates address violation problem. This problem may be solved by using "System Calls" for shared variables.
2. Write Access Problem: If the shared variable is of type Read/Write then another problem has to be solved in order to keep data integrity.