Process Management

Jayesh Deep Dubey
Technical Officer
C-DAC Patna

What is a Process?

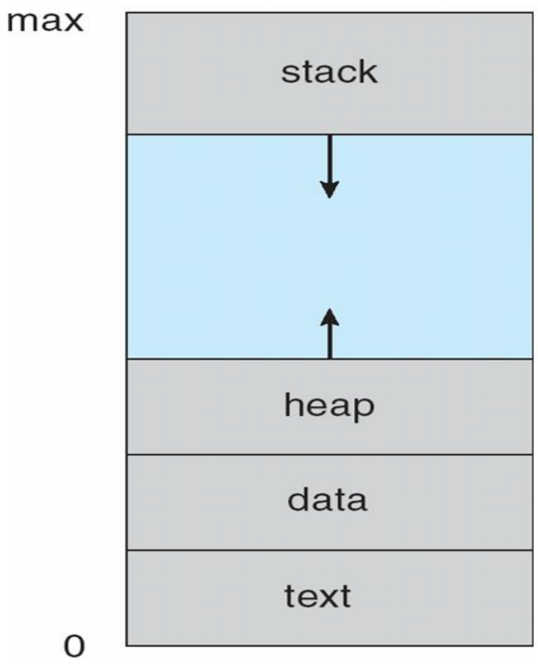
- A process is an instance of a program that is currently executing on a computer system.
    - Process – a program in execution
- Program is passive entity stored on disk (executable file), process is active
    - Program becomes process when executable file loaded into memory
- Execution of program started via GUI mouse clicks, command line entry of its name, etc
- One program can be several processes
    - Consider multiple users executing the same program

Parts of a Process

- The program code, also called **text** section
- Current activity including program counter, processor registers
- **Stack** containing temporary data
  - Function parameters, return addresses, local variables
- **Data** section containing global variables
- **Heap** containing memory dynamically allocated during run time

Process in memory

.

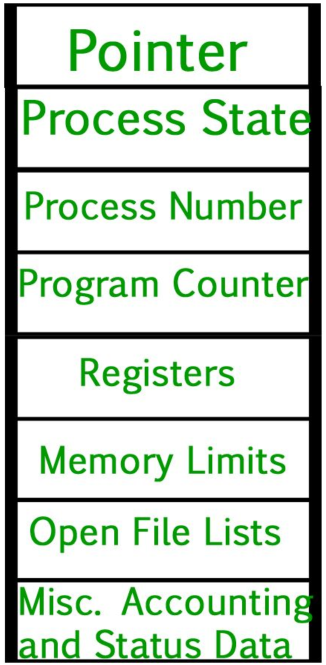


Preemptive and Non-Preemptive Process

- Preemptive and non-preemptive processes refer to the way processes are managed by an operating system.
- In a preemptive system, the operating system can interrupt a running process to allow another process to run
- In a non-preemptive system, a process runs until it voluntarily relinquishes control of the CPU.

Process Control Block

- The process control block (PCB) is a data structure used by an operating system to manage each process
- It contains all the information about a process that is necessary for the operating system to manage the process



Process Control Block

Process Control Block

- **Process Identification (ID) Section**: This section contains the unique identifier (PID) for the process.
- **Process State Section:** It stores the respective state of the process such as whether it is running, blocked, or ready to run
- **Program counter Section:** It stores the counter which contains the address of the next instruction that is to be executed for the process.
- **Register Section:** It contains information such as the values of CPU registers which includes: accumulator, base registers and general purpose registers.

Process Control Block

- **Memory Management Information Section**: This section contains information about the memory resources allocated to the process, such as the starting address, size.
- **Open files list Section**: This information includes the list of files opened for a process.
- **Pointer Section:** It contains pointers to various resources that the process may need.The purpose of the Pointer Section is to provide the operating system with quick access to the resources associated with a process, such as memory, files, or I/O devices.

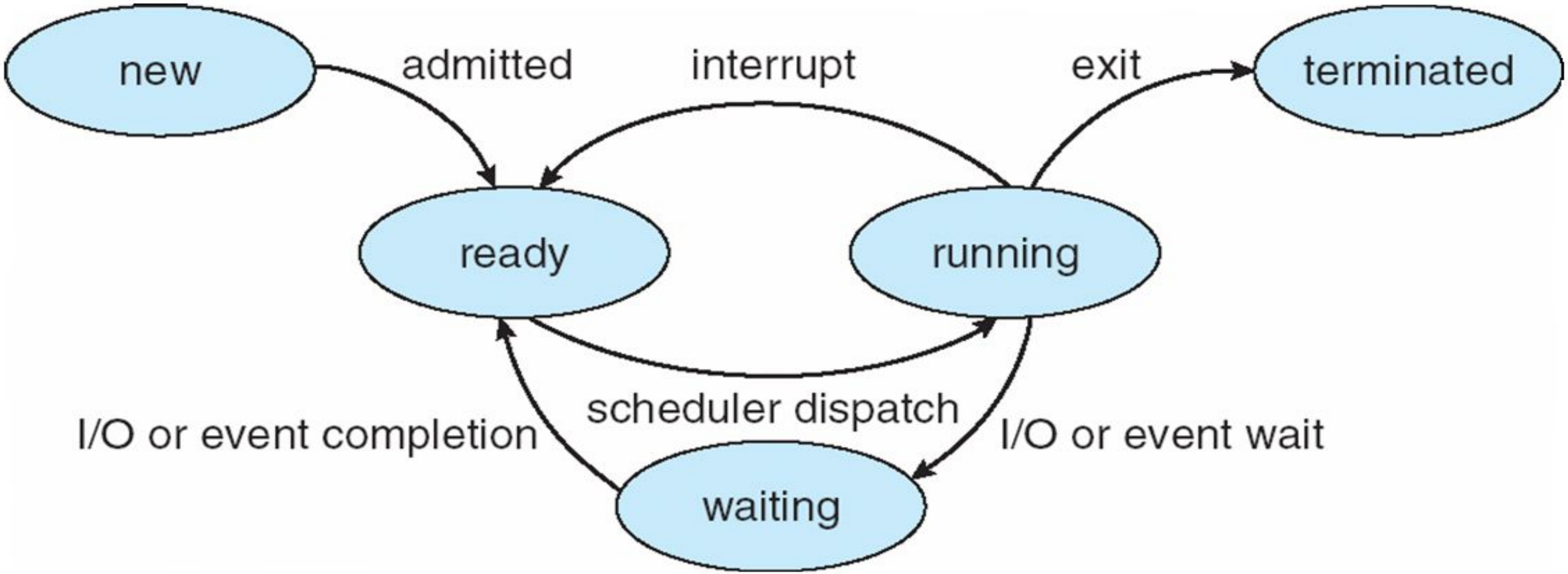
Process Control Block

- **Accounting Information Section**: This section contains information about the resources used by the process, such as CPU time used, memory usage, and I/O operations performed.
- **Input/Output (I/O) Status Information Section**: This section contains information about the I/O devices used by the process, such as the files and devices the process has opened, the status of I/O operations, and the I/O requests made by the process

Process Life Cycle

- The process life cycle consists of several stages: new, ready, running, blocked, and terminated.
- As a process executes, it changes state
  - **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

Process Life Cycle



Introduction to Schedulers

- Schedulers are an essential part of process management. They determine which process should run next and for how long.
- They are responsible for managing and coordinating the execution of multiple processes in a way that optimizes system resources and ensures fair and efficient execution of tasks.
- There are three types of schedulers - short-term, medium-term, and long-term

Two Types of Processes

- Processes can be described as either:
    - **I/O-bound process** – spends more time doing I/O than computations, many short CPU bursts
    - **CPU-bound process** – spends more time doing computations; few very long CPU bursts

Short Term Scheduler

- The short-term scheduler, also known as the CPU scheduler, determines which process should run next in the ready queue.
- It runs frequently, typically every few milliseconds, to ensure that the CPU is used efficiently

Long Term Scheduler

- The long-term scheduler, also known as the job scheduler, determines which processes should be admitted into the system
- Long-term scheduler is responsible for selecting processes from the pool of incoming processes and adding them to the pool of processes that are ready to be executed
- Long-term scheduler is invoked infrequently (seconds, minutes) (may be slow)
- Long-term scheduler strives for good process mix(I/O bound and CPU bound)

Medium Term Scheduler

- Medium-term scheduler (also known as swapping scheduler) is responsible for managing the process's memory allocation. It determines which processes should be swapped out of main memory and onto secondary storage.
- It is responsible for managing memory and disk space.
- When the number of processes in the system exceeds the available physical memory, the medium-term scheduler selects processes that have been idle for a long time and moves them to the disk. This helps to free up memory space and improve system performance.

Process Scheduling

- Process scheduling is an essential operating system function that determines the order and timing of executing different processes or threads
- The primary purpose of process scheduling is to allocate the system's resources, such as CPU time, memory, and input/output devices, to the processes in a fair and efficient manner.

Benefits of Process Scheduling

Here are some of the uses and benefits of process scheduling:

- **Fair resource allocation**: Process scheduling ensures that each process gets a fair share of the available resources, such as CPU time and memory. This ensures that no process monopolizes the system resources, which can lead to poor system performance and stability.
- **Optimal system performance**: By scheduling processes effectively, the operating system can maximize the utilization of the CPU and other resources, which can result in improved system performance.

Benefits of Process Scheduling

- **Prioritization of processes**: Process scheduling allows the operating system to prioritize critical processes, such as those involved in system maintenance or security, over other less important tasks.
- **Multitasking**: Process scheduling enables the operating system to perform multiple tasks simultaneously, even on a single-core processor. This is achieved by rapidly switching between different processes, giving the illusion of parallelism.
- **Time-sharing**: Process scheduling allows the operating system to time-share the CPU among multiple processes. This means that each process gets a small time slice of the CPU's attention, which is typically a few milliseconds long. This provides the illusion of each process running continuously, even though they are sharing the CPU with other processes.

Process Scheduling Algorithms

- Process scheduling algorithms are the methods and techniques used by the operating system to determine which process should be executed next.
- The scheduling algorithm is responsible for selecting the next process from the pool of processes waiting to be executed and allocating system resources, such as the CPU, to that process.
- Switching the CPU to another process requires performing a state save of the current process and a state restore of a different process. This task is known as a **context switch**

Process Scheduling Algorithms

- First-Come, First-Serve (FCFS)
- Shortest Job First (SJF)
- Priority Scheduling
- Round Robin (RR)
- Multilevel Queue (MLQ)

Some Important Terms w.r.t Process Scheduling

In process scheduling, different types of time are used to measure and manage the execution of processes:

- **Arrival time**: This is the time when a process arrives in the ready queue and requests the CPU for execution.
- **Burst time**: This is the time required by a process to complete its execution once it gets the CPU
- **Completion time**: This is the time when a process completes its execution, i.e., when it finishes its last instruction

Some Important Terms w.r.t Process Scheduling

Different types of time are used to measure and manage the execution of processes:

- **Waiting time**: This is the total amount of time a process spends in the ready queue waiting for the CPU to become available.
- **Turnaround time**: This is the total amount of time a process takes from when it arrives in the ready queue to when it completes its execution.

Process Scheduling Algorithms

- First Come First Serve (FCFS)
  - First come first serve scheduling algorithm states that the process that requests the CPU first is allocated the CPU first.
  - It is implemented by using the FIFO queue
  - Tasks are always executed on a First-come, First-serve concept.
  - FCFS is easy to implement and use.
  - This algorithm is not much efficient in performance, and the wait time is quite high

An example (FCFS)

Example-1: Consider the following table of arrival time and burst time for five processes P1, P2, P3, P4 and P5.

Processes	Arrival Time	Burst Time
P1	0	4
P2	1	3
P3	2	1
P4	3	2
P5	4	5

An example (FCFS)

- **Waiting Time**
  - P1 = 0 – 0 = 0
  - P2 = 4 – 1 = 3
  - P3 = 7 – 2 = 5
  - P4 = 8 – 3 = 5
  - P5 = 10 – 4 = 6

- **Average Waiting time**

= (0 + 3 + 5 + 5+ 6)/ 5 = 19 / 5 = 3.8

An example (FCFS)

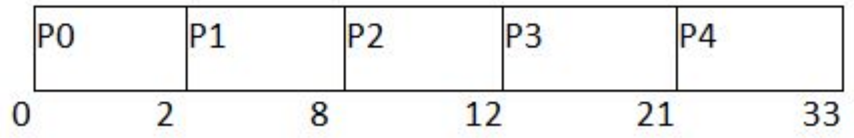
- **Turnaround Time**
  - P1 = 4 – 0 = 4
  - P2 = 7 – 1 = 6
  - P3 = 8 – 2 = 6
  - P4 = 10 – 3 = 7
  - P5 = 15 – 4 = 11

Question (FCFS)

Consider the following table of arrival time and burst time for five processes P0, P1, P2, P3 and P4.

Processes	Arrival Time	Burst Time
P0	0	2
P1	1	6
P2	2	4
P3	3	9
P4	6	12

Solution



Completion Time	Turnaround Time	Waiting Time
2	2	0
8	7	1
12	10	6
21	18	9
33	29	17

Process Scheduling Algorithms

- **Shortest Job First(SJF)**
  - SJF selects the waiting process with the smallest execution time to execute next.
  - Shortest Job first has the advantage of having a minimum average waiting time among all scheduling algorithms.
  - SJF can be used in specialized environments where accurate estimates of running time are available

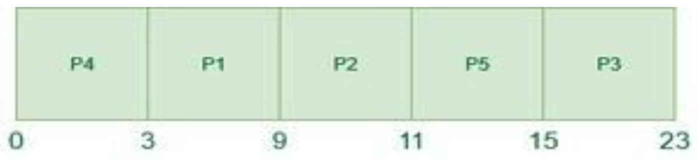
An example (SJF)

Example-1: Consider the following table of arrival time and burst time for five processes P1, P2, P3, P4 and P5.

Processes	Arrival Time	Burst Time
P1	2	6
P2	5	2
P3	1	8
P4	0	3
P5	4	4

An example (SJF)

- **Waiting Time**
  - P4 = 0 – 0 = 0
  - P1 = 3 – 2 = 1
  - P2 = 9 – 5 = 4
  - P5 = 11 – 4 = 7
  - P3 = 15 – 1 = 14



- Average Waiting Time

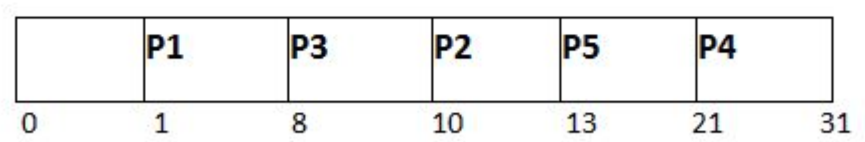
  = 0 + 1 + 4 + 7 + 14/5 = 26/5 = 5.2

Question (SJF)

Consider the following table of arrival time and burst time for five processes P1, P2, P3, P4 and P5.

Processes	Arrival Time	Burst Time
P1	1	7
P2	3	3
P3	6	2
P4	7	10
P5	9	8

Solution



Completion Time	Turnaround Time	Waiting Time
8	7	0
13	10	7
10	4	2
31	24	14
21	12	4