Zakho Technical Institute / IT

Operation System -

Theory

3. Process Management

Lecturer:

Sipan M. Hameed

www.sipan.dev

2023-2024

Contents

3. Process Management 1

Process Management in OS 3

The Process 4

Explanation of Process 4

Process Control Block 5

Process State 6

Operations on the Process 7

Process term Scheduling in OS 8

Process Queues 8

Various Times related to the Process 8

CPU Scheduling idea 8

CPU Scheduling 9

Scheduling Algorithms in OS (Operating System) 10

Scheduling Algorithms 10

CPU Scheduling Criteria 12

Inter Process Communication (IPC) 12

A process can be of two types: 12

Processes can communicate with each other through both: 12

Context Switching in Operating System 13

Context Switching Triggers 13

Process Management in OS

A Program does nothing unless its instructions are executed by a CPU. A program in execution is called a process. In order to accomplish its task, process needs the computer resources.

There may exist more than one process in the system which may require the same resource at the same time. Therefore, the operating system has to manage all the processes and the resources in a convenient and efficient way.

Some resources may need to be executed by one process at one time to maintain the consistency otherwise the system can become inconsistent and deadlock may occur.

The operating system is responsible for the following activities in connection with Process Management

1. Scheduling processes and threads on the CPUs.
2. Creating and deleting both user and system processes.
3. Suspending and resuming processes.
4. Providing mechanisms for process synchronization.
5. Providing mechanisms for process communication.

The Process

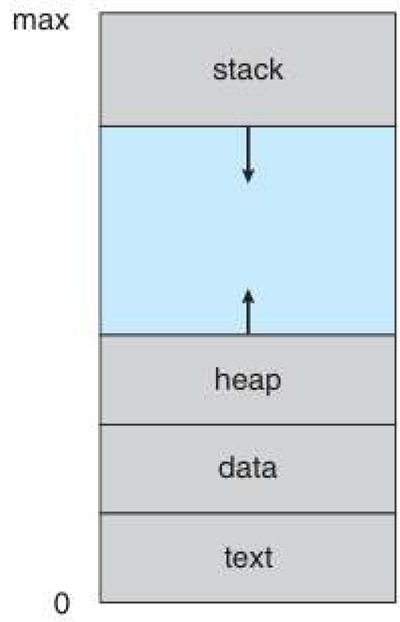


Figure 3.1 Layout of a process in memory.

Explanation of Process

1. **Text Section**: A Process, sometimes known as the Text Section, also includes the current activity represented by the value of the Program Counter.
2. **Stack**: The stack contains temporary data, such as function parameters, returns addresses, and local variables.
3. **Data Section**: Contains the global variable.
4. **Heap Section**: Dynamically memory allocated to process during its run time.

Process Control Block

Each process is represented in the operating system by a process control block (PCB)—also called a task control block. A PCB is shown in Figure 3.3. It contains many pieces of information associated with a specific process, including these:

1. **Process state.** The state may be new, ready, running, waiting, halted, and so on.
2. **Program counter**. The counter indicates the address of the next instruction to be executed for this process.
3. **CPU registers**. The registers vary in number and type, depending on the computer architecture. They include accumulators, index registers, stack pointers, and general-purpose registers, plus any condition-code information . Along with the program counter, this state information must be saved when an interrupt occurs, to allow the process to be continued correctly afterward when it is rescheduled to run.
4. **CPU-scheduling information**. This information includes a process priority , pointers to scheduling queues, and any other scheduling parameters.(Chapter 5 describes process scheduling.)
5. **Memory-management information**. This information may include such items as the value of the base and limit registers and the page tables, or the segment tables, depending on the memory system used by the operating system (Chapter 9).
6. **Accounting information**. This information includes the amount of CPU and real time used, time limits, account numbers, job or process numbers, and so on.
7. **I/O status information**. This information includes the list of I/O devices allocated to the process, a list of open files, and so on.

Figure 3.3 Process control block (PCB).

Process State

As a process executes, it changes state. The state of a process is defined in part by the current activity of that process. A process may be in one of the following states:

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

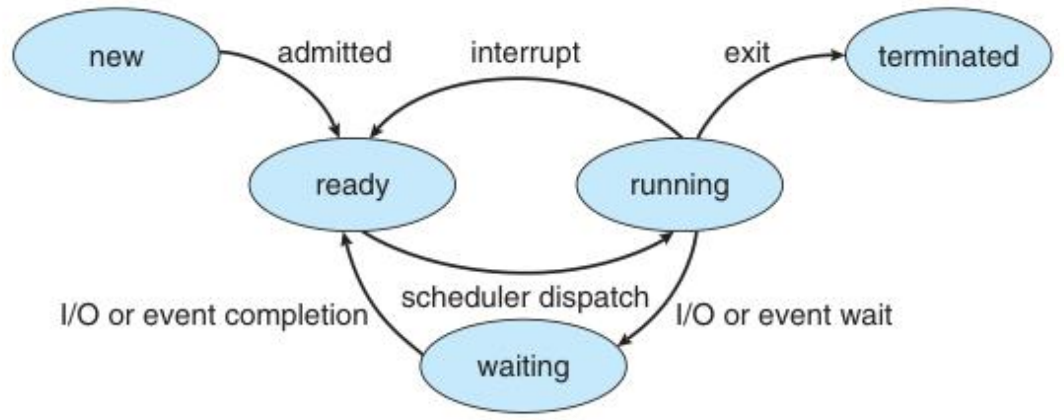


Figure 3.2 Diagram of process state.

Operations on the Process

1. Creation

Once the process is created, it will be ready and come into the ready queue (main memory) and will be ready for the execution.

2. Scheduling

Out of the many processes present in the ready queue, the Operating system chooses one process and start executing it. Selecting the process which is to be executed next, is known as scheduling.

3. Execution

Once the process is scheduled for the execution, the processor starts executing it. Process may come to the blocked or wait state during the execution then in that case the processor starts executing the other processes.

4. Deletion/killing

Once the purpose of the process gets over then the OS will kill the process. The Context of the process (PCB) will be deleted and the process gets terminated by the Operating system.

Process term Scheduling in OS

Operating system uses various schedulers for the process scheduling described below.

1. Long term scheduler
2. Short term scheduler
3. Medium term scheduler

Process Queues

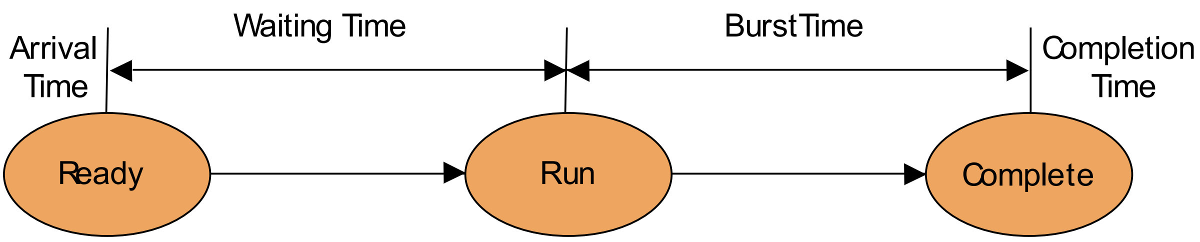
1. Job Queue
2. Ready Queue
3. Waiting Queue

Various Times related to the Process

1. Arrival Time
2. Burst Time
3. Completion Time
4. Turnaround time
5. Waiting Time
6. Response Time

CPU Scheduling idea

A process migrates among the ready queue and various wait queues through-out its lifetime. The role of the CPU scheduler is to select from among the processes that are in the ready queue and allocate a CPU core to one of them.



CT - AT = WT + BT

TAT = CT - AT

Waiting Time = TAT - BT

TAT → Turn around time

BT → Burst time

AT → Arrival time

Figure 1.Times related to the Process

CPU Scheduling

1. In the **uniprogrammming systems** like MS DOS, when a process waits for any I/O operation to be done, the CPU remains idol. This is an overhead since it wastes the time and causes the problem of starvation. However, In Multiprogramming systems, the CPU doesn't remain idle during the waiting time of the Process and it starts executing other processes. Operating System has to define which process the CPU will be given.
2. In **Multiprogramming systems**, the Operating system schedules the processes on the CPU to have the maximum utilization of it and this procedure is called CPU scheduling. The Operating System uses various scheduling algorithm to schedule the processes.

This is a task of the short term scheduler to schedule the CPU for the number of processes present in the Job Pool. Whenever the running process requests some IO operation then the short term scheduler saves the current context of the process (also called PCB) and changes its state from running to waiting. During the time, process is in waiting state; the Short term scheduler picks another process from the ready queue and assigns the CPU to this process. This procedure is called context switching.

Scheduling Algorithms in OS (Operating System)

There are various algorithms which are used by the Operating System to schedule the processes on the processor in an efficient way.

The Purpose of a Scheduling algorithm

1. Maximum CPU utilization
2. Fare allocation of CPU
3. Maximum throughput
4. Minimum turnaround time
5. Minimum waiting time
6. Minimum response time

Scheduling Algorithms

There are the following algorithms which can be used to schedule the jobs.

1. First Come First Serve
2. Round Robin
3. Shortest Job First
4. Multilevel Feedback Queue Scheduling
5. Priority based scheduling
6. Multilevel Queue Scheduling
7. Longest Job First (LJF)
8. Shortest Remaining Time First (SRTF)

9. Longest Remaining Time First (LRTF)
10. Highest Response Ratio Next (HRRN)

CPU Scheduling Criteria

1. CPU utilization.
2. Throughput
3. Turnaround time.
4. Waiting time.
5. Response time.

Inter Process Communication (IPC)

A process can be of two types:

1. Independent process.
2. Co-operating process.

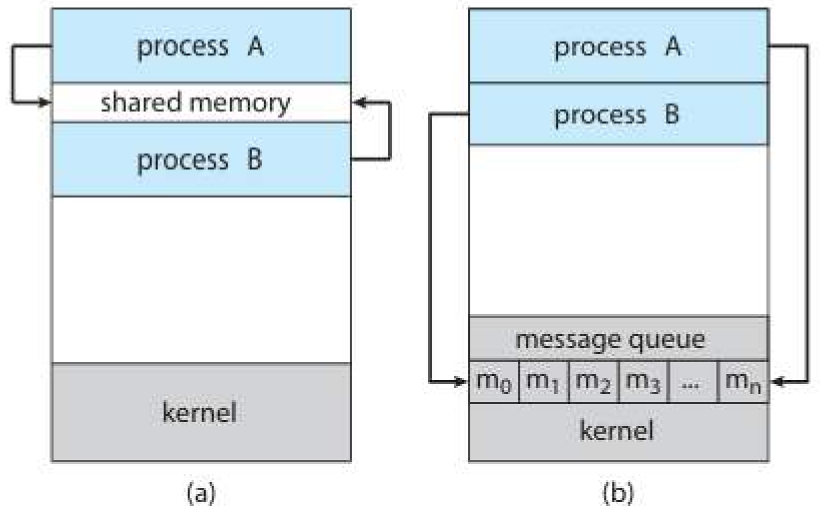


Figure 3.11 Communications models. (a) Shared memory. (b) Message passing.

Processes can communicate with each other through both:

1. Shared Memory
2. Message passing

Context Switching in Operating System

Context switching in an operating system involves saving the context or state of a running process so that it can be restored later, and then loading the context or state of another. process and run it.

Context Switching refers to the process/method used by the system to change the process from one state to another using the CPUs present in the system to perform its job.

Context Switching Triggers

The three different categories of context-switching triggers are as follows.

1. Interrupts
2. Multitasking
3. User/Kernel switch