[1]Dr. D. Y. Patil Institute of Management and Entrepreneur Development

Chapter 2: Process Management & Synchronization

PCB

1. Process ID:
When a new process is created by the user, the operating system assigns a unique ID i.e a process-ID to that process. This ID helps the process to be distinguished from other processes existing in the system.
The OS can handle at most N processes at a time.

So, process-ID will get the values from 0 to N-1.

2. Process State:
A process, from its creation to completion goes through different states. Generally, a process may be present in one of the 5 states during its execution:

New: This state contains the processes which are ready to be loaded by the operating system into the main memory.

Ready: This state contains the process which is both ready to be executed and is currently in the main memory of the system. The operating system brings the processes from secondary memory(hard disk) to main memory(RAM). As these processes are present in the main memory and are waiting to be assigned to the CPU, the state of these processes is known as Ready state.

Running: This state contains the processes which are currently executed by the CPU in our system.

[1] Prof.Harini Pawar

Block or wait: A process from its running state may transition to a block or wait for state depending on the scheduling algorithm or because of the internal behavior of the process (process explicitly wants to wait).

Termination: A process that completes its execution comes to its termination state.

3. Process Priority:
Process priority is a numeric value that represents the priority of each process. The lesser the value, the greater the priority of that process. This priority is assigned at the time of the creation of the PCB and may depend on many factors like the age of that process, the resources consumed, and so on. The user can also externally assign a priority to the process

4. Process Accounting Information:
This attribute gives the information of the resources used by that process in its lifetime.

5. Program Counter:
The program counter is a pointer that points to the next instruction in the program to be executed. This attribute of PCB contains the address of the next instruction to be executed in the process.

6. CPU registers:
A CPU register is a quickly accessible small-sized location available to the CPU. These registers are stored in virtual memory(RAM).

7. Context Switching:
A context switching is a process that involves switching the CPU from one process or task to another. It is the process of storing the state of a process so that it can be restored and resume execution at a later point. This allows multiple processes to share a single CPU and is an essential feature of a multitasking operating system.

8. PCB pointer:
This field contains the address of the next PCB, which is in ready state. This helps the operating system to hierarchically maintain an easy control flow between parent processes and child processes.

9. List of open files:
As the name suggests, It contains information on all the files that are used by that process. This field is important as it helps the operating system to close all the opened files at the termination state of the process.

10. Process I/O information**:**
In this field, the list of all the input/output devices which are required by that process during its execution is mentioned.

Concurrency

Advantages of Concurrency :

- **Running of multiple applications –**
  It enables you to run multiple applications at the same time.
- **Better resource utilization –**
  It enables the resources that are unused by one application can be used for other applications.
- **Better average response time –**
  Without concurrency, each application has to be run to completion before the next one can be run.
- **Better performance –**
  It enables better performance by the operating system. When one application uses only the processor and another application uses only the disk drive then the time to run both applications concurrently to completion will be shorter than the time to run each application consecutively.

DeadLock

Methods of Handling Deadlocks in Operating System

The first two methods are used to ensure the system never enters a deadlock.

Deadlock Prevention: This is done by restraining the ways a request can be made. Since deadlock occurs when all the above four conditions are met, we try to prevent any one of them, thus preventing a deadlock.

Deadlock Avoidance: When a process requests a resource, the deadlock avoidance algorithm examines the resource-allocation state. If allocating that resource sends the system into an unsafe state, the request is not granted.

Deadlock Detection and Recovery : We let the system fall into a deadlock and if it happens, we detect it using a detection algorithm and try to recover.

Some ways of recovery are as follows

- Aborting all the deadlocked processes.
- Abort one process at a time until the system recovers from the deadlock.

Deadlock Ignorance: In the method, the system assumes that deadlock never occurs. Since the problem of deadlock situation is not frequent, some systems simply ignore it