

## Process

A process is basically a program in execution. The execution of a process must progress in a sequential fashion.

A process is defined as an entity which represents the basic unit of work to be implemented in the system.

To put it in simple terms, we write our computer programs in a text file and when we execute this program, it becomes a process which performs all the tasks mentioned in the program.

When a program is loaded into the memory and it becomes a process, it can be divided into four sections – stack, heap, text and data. The following image shows a simplified layout of a process inside main memory –

## Process Components

S.N. Component & Description

1

Stack

The process Stack contains the temporary data such as method/function parameters, return address and local variables.

2

Heap

This is dynamically allocated memory to a process during its run time.

3

Text

This includes the current activity represented by the value of Program Counter and the contents of the processor's registers.

4

Data

This section contains the global and static variables.

## Program

A program is a piece of code which may be a single line or millions of lines. A computer program is usually written by a computer programmer in a programming language. For example, here is a simple program written in C programming language –

```
#include <stdio.h>
```

```
int main() {  
    printf("Hello, World! \n");  
    return 0;  
}
```

A computer program is a collection of instructions that performs a specific task when executed by a computer. When we compare a program with a process, we can conclude that a process is a dynamic instance of a computer program.

A part of a computer program that performs a well-defined task is known as an algorithm. A collection of computer programs, libraries and related data are referred to as a software.

#### Process Life Cycle

When a process executes, it passes through different states. These stages may differ in different operating systems, and the names of these states are also not standardized.

In general, a process can have one of the following five states at a time.

#### S.N. State & Description

1

Start

This is the initial state when a process is first started/created.

2

Ready

The process is waiting to be assigned to a processor. Ready processes are waiting to have the processor allocated to them by the operating system so that they can run. Process may come into this state after Start state or while running it by but interrupted by the scheduler to assign CPU to some other process.

3

Running

Once the process has been assigned to a processor by the OS scheduler, the process state is set to running and the processor executes its instructions.

4

Waiting

Process moves into the waiting state if it needs to wait for a resource, such as waiting for user input, or waiting for a file to become available.

5

Terminated or Exit

Once the process finishes its execution, or it is terminated by the operating system, it is moved to the terminated state where it waits to be removed from main memory.

#### File

A file is a named collection of related information that is recorded on secondary storage such as magnetic disks, magnetic tapes and optical disks. In general, a file is a sequence of bits, bytes, lines or records whose meaning is defined by the files creator and user.

### File Structure

A File Structure should be according to a required format that the operating system can understand.

A file has a certain defined structure according to its type.

A text file is a sequence of characters organized into lines.

A source file is a sequence of procedures and functions.

An object file is a sequence of bytes organized into blocks that are understandable by the machine.

When operating system defines different file structures, it also contains the code to support these file structure. Unix, MS-DOS support minimum number of file structure.

### File Type

File type refers to the ability of the operating system to distinguish different types of file such as text files source files and binary files etc. Many operating systems support many types of files. Operating system like MS-DOS and UNIX have the following types of files -

#### Ordinary files

These are the files that contain user information.

These may have text, databases or executable program.

The user can apply various operations on such files like add, modify, delete or even remove the entire file.

#### Directory files

These files contain list of file names and other information related to these files.

#### Special files

These files are also known as device files.

These files represent physical device like disks, terminals, printers, networks, tape drive etc.

These files are of two types -

Character special files - data is handled character by character as in case of terminals or printers.

Block special files - data is handled in blocks as in the case of disks and tapes.

### File Access Mechanisms

File access mechanism refers to the manner in which the records of a file may be accessed. There are several ways to access files -