**Process-:** A process is an program under execution, it's a instance of a program that

is being executed.

A program can be segregated into four pieces when put into memory to become

a process -: stack, heap, text, and data.

Component of a process -: Stack <-> Heap <-> Data <-> Text

Process Different States -:

1. New : The creation of a process.

2. Ready : waiting for the process that is assigned to any processor.

3. Running : Execution of the process.

4. Waiting : The waiting of the process for some event that is about to

occur (like an I/O completion, a signal reception, etc.).

5. Terminated : A process has completed execution.

Process Control Block (PCB)- 1. Process ID

1. Process ID
2. Process State
3. Priority
4. Program Counter
5. CPU Register
6. I/O Information

Threads-:A thread is a single sequential flow of execution of tasks of a process

so it is also known as thread of execution or thread of control.

Also known as the light weight process.

All thread has this three Component

1. Program Counter ,

2. Register set

3. Stack Space

Types of Thread -:

1. User Level thread
2. Kernel level thread

Benefits of threads -:

1. Enhanced throughput of the system
2. Effective utilisation of multiprocessor system
3. Faster context switching
4. Responsiveness
5. Communication
6. Resource sharing

Difference between Thread and Process-:

| **Thread** | **Process** |
| --- | --- |
| Thread is a segment of a process or a lightweight process | process is an instance of a program that is being executed or processed. |
| Thread are interdependent and share memory spaces | Process are independent and does not share memory space |
| Context switching is very fast | Context switching is slow as compare to thread |
| Threads can be terminated in very little time and also take less time for creation. | The operating system takes more time to terminate and create a process. |

Process scheduling in os-:

1. *Long term scheduler-:* Long term scheduler is also known as job scheduler. It chooses the processes from the pool (secondary memory) and keeps them in the ready queue maintained in the primary memory.
2. *Short term scheduler* -: Short term scheduler is also known as CPU scheduler. It selects one of the Jobs from the ready queue and dispatch to the CPU for the execution.
3. *Mid term scheduler-*: Medium term scheduler takes care of the swapped out processes.If the running state processes needs some IO time for the completion then there is a need to change its state from running to waiting.

OS scheduling algorithms-:

1. Non-preemptive -: Non- preemptive algorithm works on when any process gets cpu it will not prompt until it executes completly.

i First-Come, First-Served (FCFS) Scheduling

ii. Shortest job First

iii. Priority based scheduling

1. Preemptive -: In preemptive mainly focus on priority and less executing time if higher priority jobs come then we can prompt our running process and give cpu to higher priority process.

i Round robin scheduling

ii. shortest remaining time

**A program of threads -:**

Program 1-:

#include<bits/stdc++.h>

using namespace std;

void check\_thread(int id)

{

for(int i=0;i<5;i++)

{

cout<<"thread " <<id<<"is running"<<endl;

}

}

int main() {

cout<<"main thread is executing"<<endl;

std::thread t1(check\_thread ,1);

std::thread t2(check\_thread ,2);

std::thread t3(check\_thread ,3);

t1.join();

t2.join();

t3.join();

cout<<"main thread is finished"<<endl;

return 0;

}

**Output-:**

**main thread is executing**

**thread 1is running**

**thread 1is running**

**thread 1is running**

**thread 1is running**

**thread 1is running**

**thread 3is running**

**thread 3is running**

**thread 3is running**

**thread 3is running**

**thread 3is running**

**thread 2is running**

**thread 2is running**

**thread 2is running**

**thread 2is running**

**thread 2is running**

**main thread is finished**