Process and CPU Scheduling

Prof. Harish D.G.
Dept. of Computer and IT
College of Engineering,Pune
www.harishgadade.com

Introduction

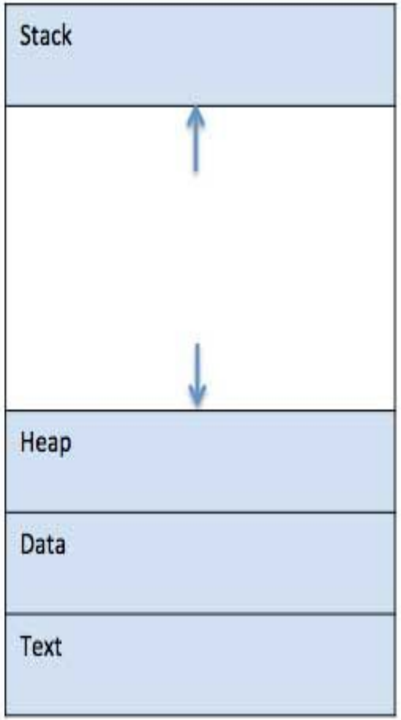
- An important aspect of multiprogramming is scheduling.
- The goal is to achieve
  - High processor utilization
  - High throughput
    - Number of processes completed per unit time
  - Low response time
    - Time elapse from the submission of a request to the beginning of the response

Program : A program is a passive entity, such as a file containing a list of instruction stored on disk

Process: Process ia an active entity, and is a program in execution

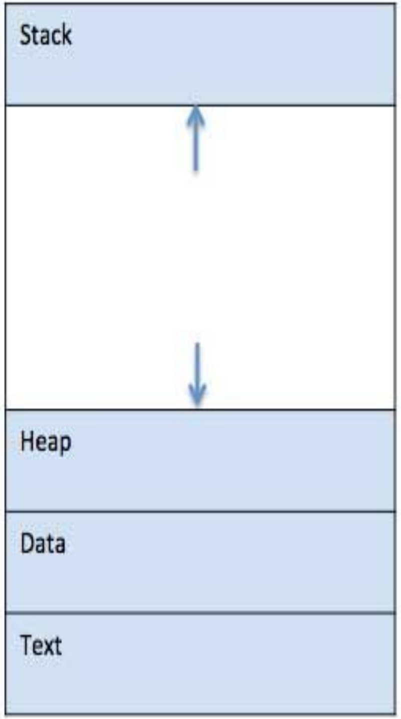
Process

- A process is basically a program in execution.
- 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.



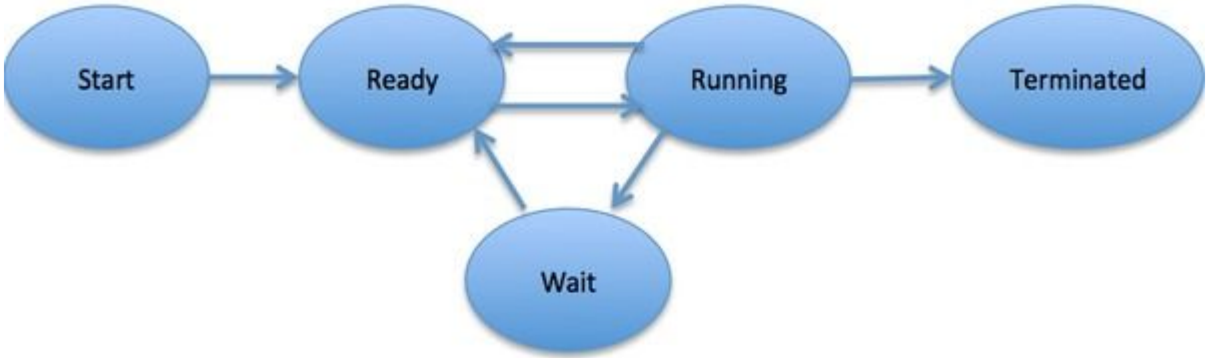
Process

- **Stack:** The process Stack contains the temporary data such as method/function parameters, return address and local variables
- **Heap** : This is dynamically allocated memory to a process during its run time.
- **Text** : This includes the current activity represented by the value of Program Counter and the contents of the processor's registers
- **Data:** This section contains the global and static variables.



Process States / 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



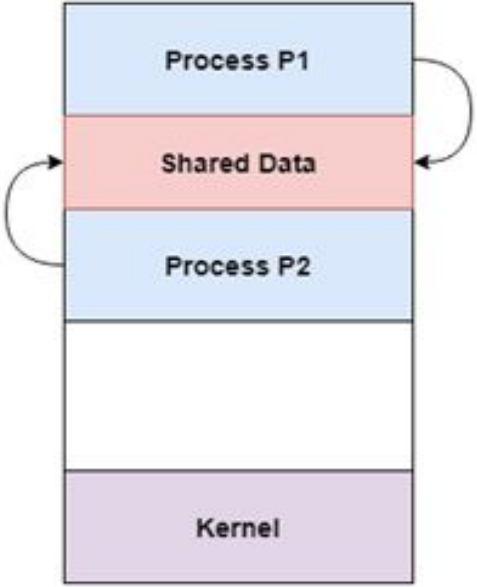
- CPU Burst
- IO Burst

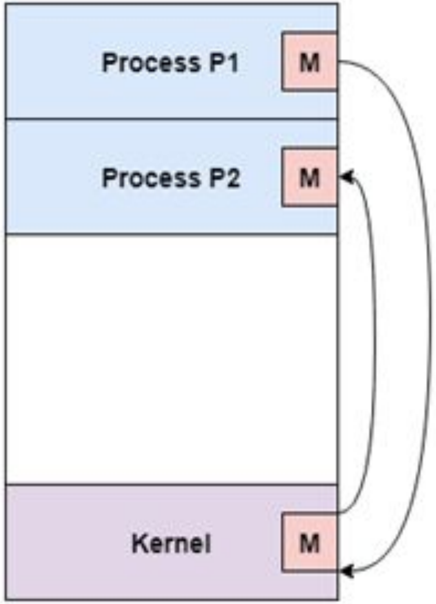
Cooperating Process

- The concurrent processes which are executing in OS may be either independent processes or cooperating processes.
- Cooperating processes are those that can affect or are affected by other processes running on the system. Cooperating processes may share data with each other.
- Reasons for needing cooperating processes
    - Modularity
    - Information Sharing
    - Convenience
    - Computation Speedup

Cooperating Process

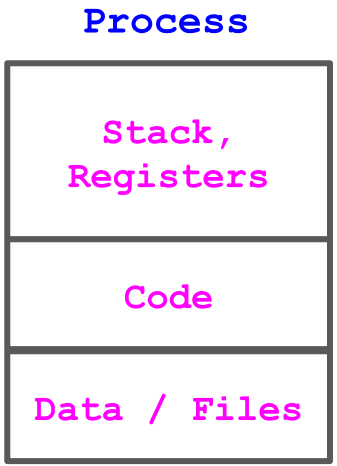
- **Methods of Corporations**
  - Cooperation by Sharing
  - Cooperation by Communication





Threads

- Thread is a light weight process, consisting of a program counter, a stack, and a set of registers.



Process

Stack, Registers

Code

Data / Files

Threads

- Thread is a light weight process, consisting of a program counter, a stack, and a set of registers.

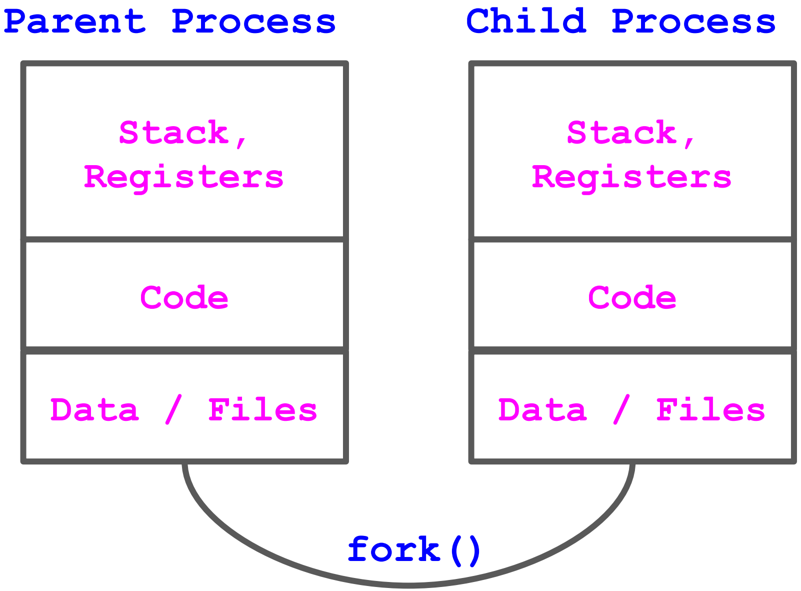


Fig. Process Concept

Threads

- Thread is a light weight process, consisting of a program counter, a stack, and a set of registers.

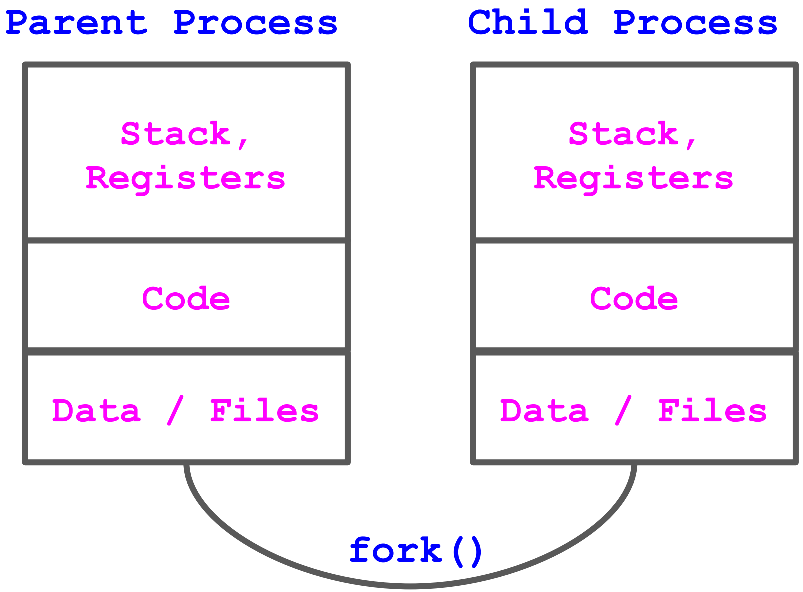


Fig. Process Concept

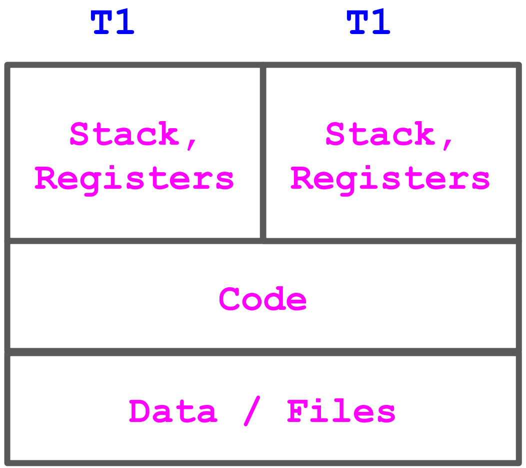


Fig. Thread Concept

Threads

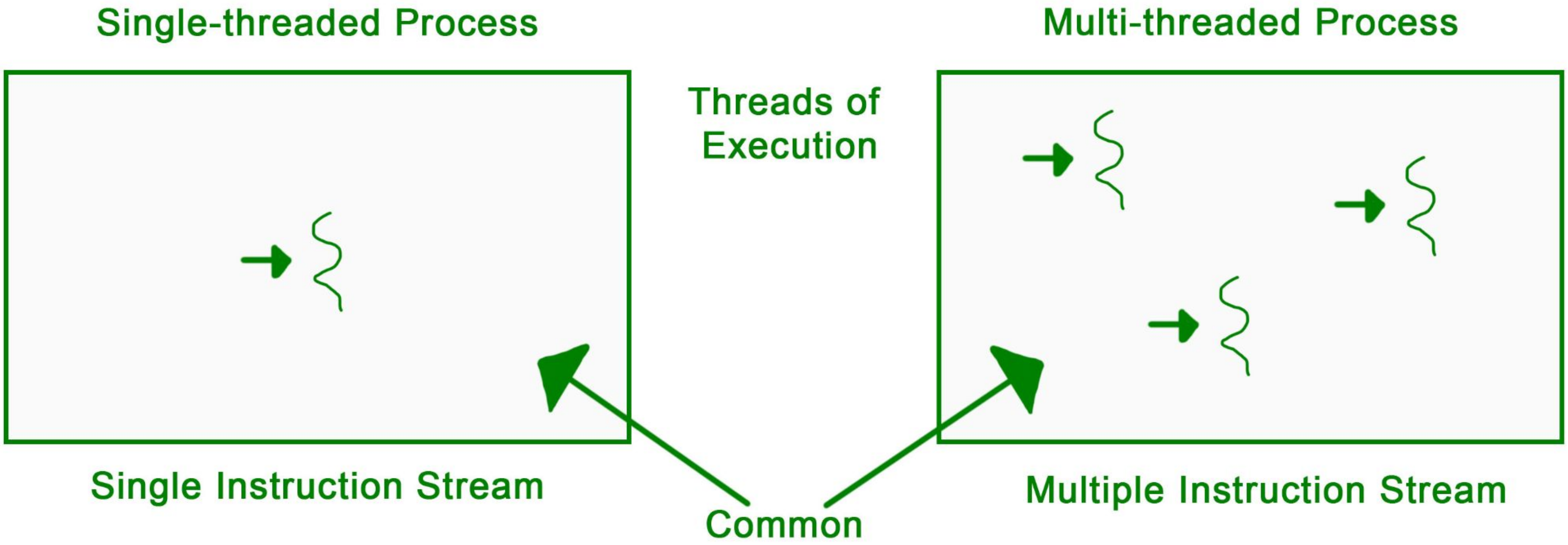
- **Process:**
  - System calls involved in process
  - OS treats different processes differently
  - Different processes have different copies of Data,Files and Codes
  - Context switching is slower
  - Blocking of processes will not block another process
  - independent

- **Threads:**
  - There is no system call involved
  - All user level threads treated as single task for OS
  - Threads share same copy of data and code
  - Context switching is faster
  - Blocking of a thread will block entire process.
  - Interdependent

Threads Libraries

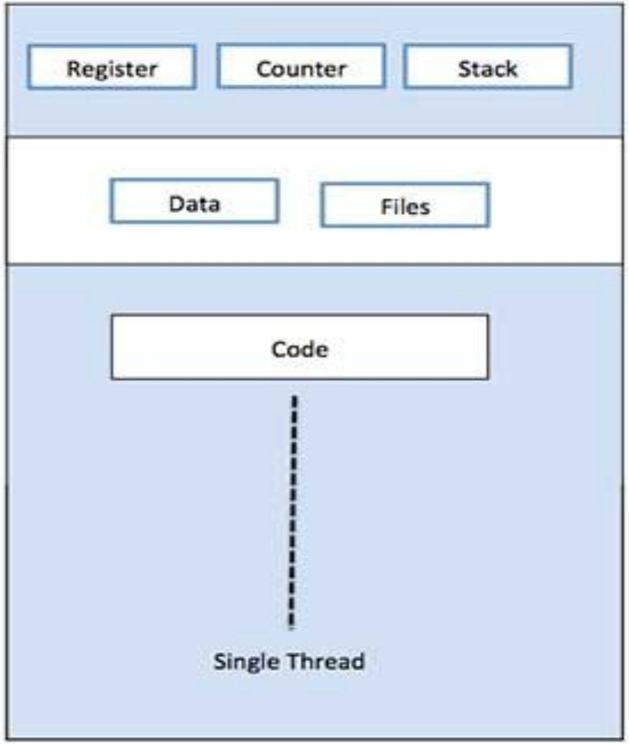
- Thread libraries provide programmers with an API for creating and managing threads.
- Thread libraries may be implemented either in user space or in kernel space.
- There are three main thread libraries in use today:
    - POSIX Pthreads - may be provided as either a user or kernel library, as an extension to the POSIX standard.
    - Win32 threads - provided as a kernel-level library on Windows systems.
    - Java threads - Since Java generally runs on a Java Virtual Machine, the implementation of threads is based upon whatever OS and hardware the JVM is running on, i.e. either Pthreads or Win32 threads depending on the system.

Multi Threading

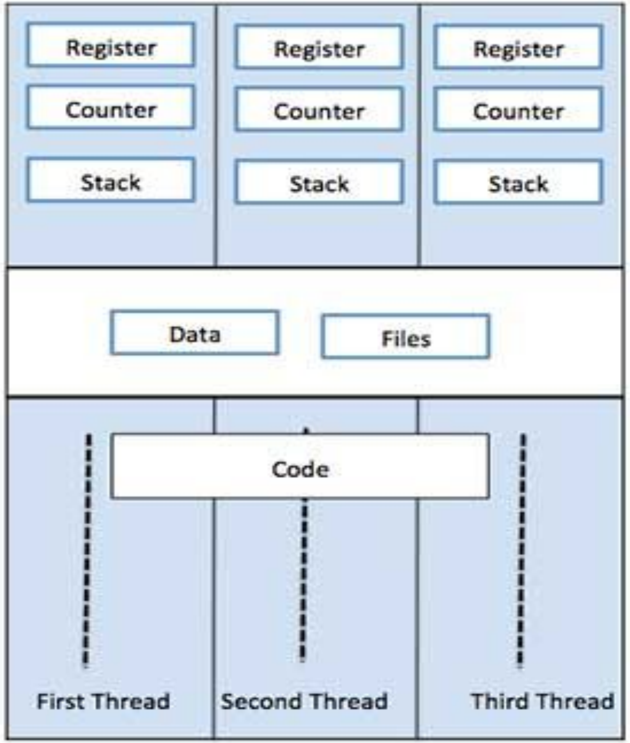


Single Thread and Multi Thread Process

Multi Threading



Single Process P with single thread



Single Process P with three threads