**프로세스(Process)** : 실행 중인 프로그램

- 디스크로부터 메모리에 적재되어 CPU, 주소 공간, 파일, 힙[[1]](#footnote-1) 등을 할당 받음

- 기본적으로 최소 1개의 스레드(=메인 스레드)를 소유

*\* 다른 프로세스의 변수나 자료구조에 접근할 수 없으며, 접근을 위해서 IPC 통신이 필요*

**◼ 프로세스 제어 블록(PCB, Process Control Block)**

- 프로세스 생성 시 함께 생성된다

**◼ 멀티 프로세스(Multi Process)**

- 여러 프로세스가 하나의 작업을 수행

- 프로세스가 각자의 메모리 영역을 가지므로 안정성이 보장됨

- 프로세스마다 독립된 메모리 영역을 갖고 있어 작업량이 많을수록 오버헤드 발생, Context Switching[[2]](#footnote-2)으로 인한 성능 저하 발생

**스레드(Thread)** : 프로세스의 실행 단위

- 한 프로세스 내에서 실행되는 여러 흐름 단위

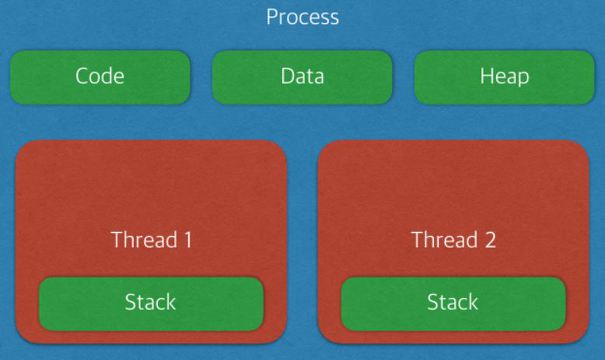
- 스레드는 자신의 스택을 가지며 프로세스 내의 주소 공간이나 자원을 공유한다.

**◼ 멀티 스레드(Multi Thread)**

- 하나의 프로그램 내에서 여러 스레드가 공유 메모리를 통해 다수의 작업을 동시에 처리함

- 공유 메모리를 사용하여 멀티 프로세스 대비 시간/자원 사용량 감소.

- 공유 메모리의 공간이 망가지면 모든 스레드의 동작에 영향을 끼치므로 안정성이 보장되지 않음[[3]](#footnote-3)



1. 힙(Heap), 동적으로 할당되는 메모리 [↑](#footnote-ref-1)
2. Context Switching이란 프로세스의 상태 정보를 저장/복원하는 일련의 과정. 동작 중인 프로세스가 대기하면서 해당 프로세스의 상태를 보관하고, 대기하고 있던 프로세스의 상태를 복구한다. [↑](#footnote-ref-2)
3. Critical Section 기법으로 대비한다. 하나의 스레드가 공유 데이터 값을 변경하는 시점에 다른 스레드가 그 값을 읽으려고 할 때 발생하는 문제를 해결하기 위한 동기화 과정. 상호 배제/진행/한정된 대기의 조건을 충족해야 한다. [↑](#footnote-ref-3)