

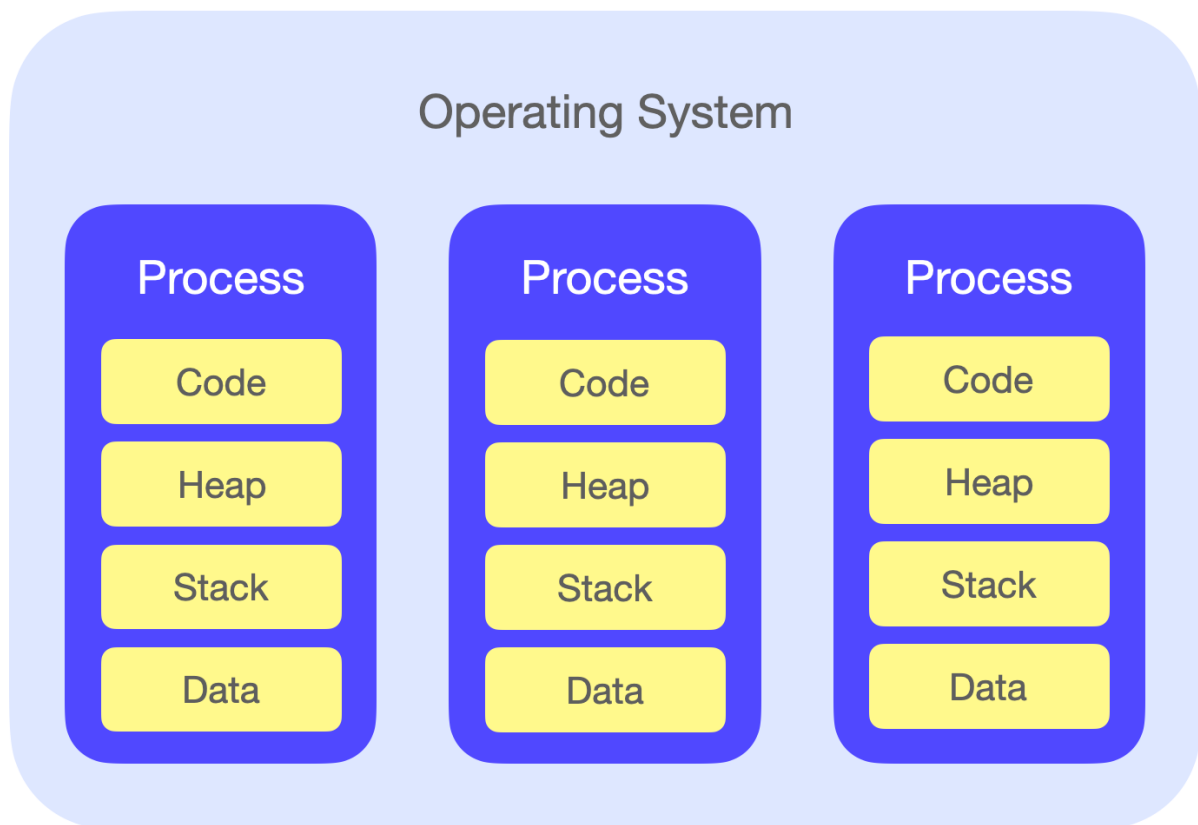
# 프로세스 & 스레드

1. 프로세스
2. 스레드
3. 멀티 프로세스 vs 멀티 스레드
4. 멀티 코어

## 1. 프로세스

- 컴퓨터가 일하고 있는 상태, 실행중에 있는 프로그램
- 프로세스 내부에는 최소 하나의 스레드를 가지고 있는데, 실제로는 스레드 단위로 스케줄링을 한다.
- 프로그램 실행 시, 실행을 위해 메모리 할당이 이루어 지고, 할당된 메모리 공간으로 바이트 코드가 올라가게 된다. 이 순간부터 **프로세스**라 부른다.

### 1) 프로세스의 메모리 구조



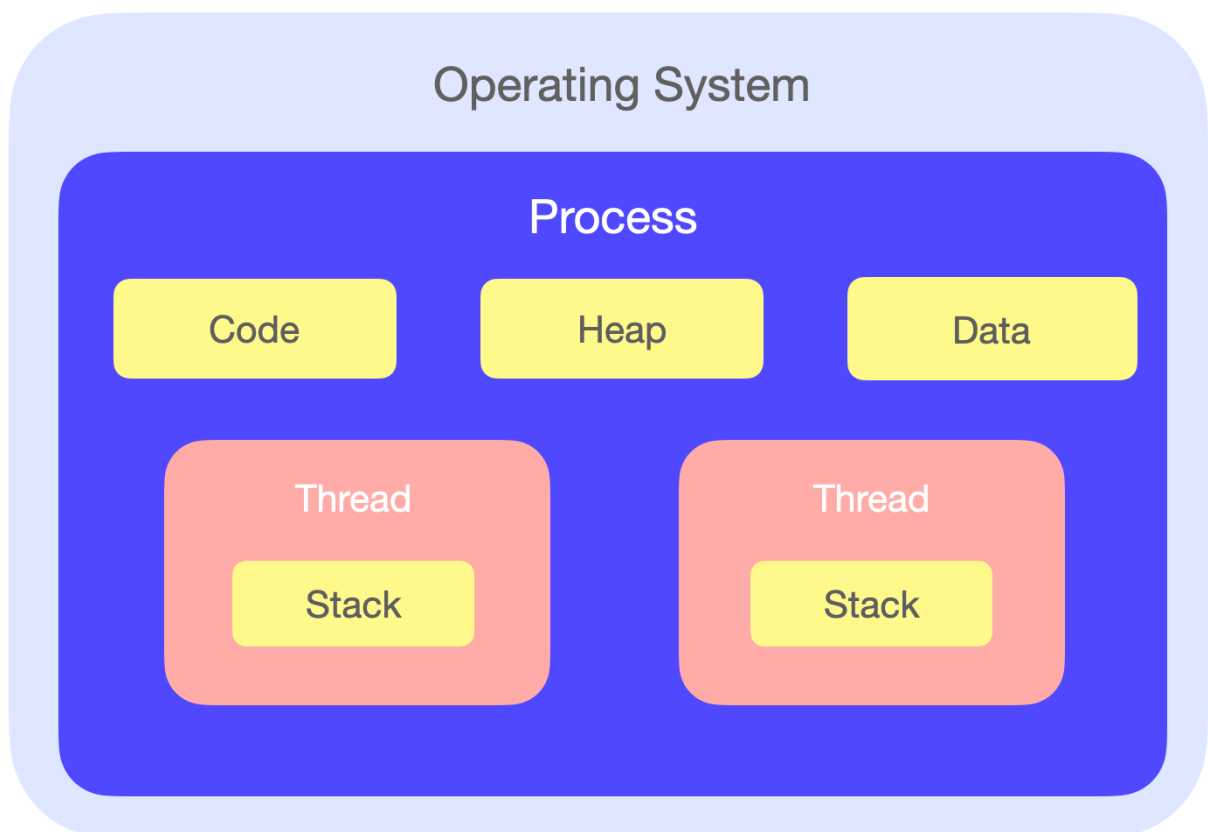
- Code 영역 : 소스코드
- Data 영역 : 전역변수, static 변수의 할당
- Heap 영역 : 동적 할당을 위한 메모리 영역
- Stack 영역 : 지역변수, 함수 호출 시 전달되는 인자를 위한 메모리 영역

## 2) 특징

- 각 프로세스는 별도의 주소 공간에서 실행되며, 한 프로세스는 다른 프로세스의 변수나 자료구조에 접근할 수 없다.
- 한 프로세스가 다른 프로세스의 자원에 접근하려면 프로세스 간의 통신(IPC, inter-process communication)을 사용해야 한다.
  - 파이프, 파일, 소켓 등을 이용한 통신 방법 이용

## 2. 스레드

- 프로세스 내에서 실행되는 여러 흐름의 단위

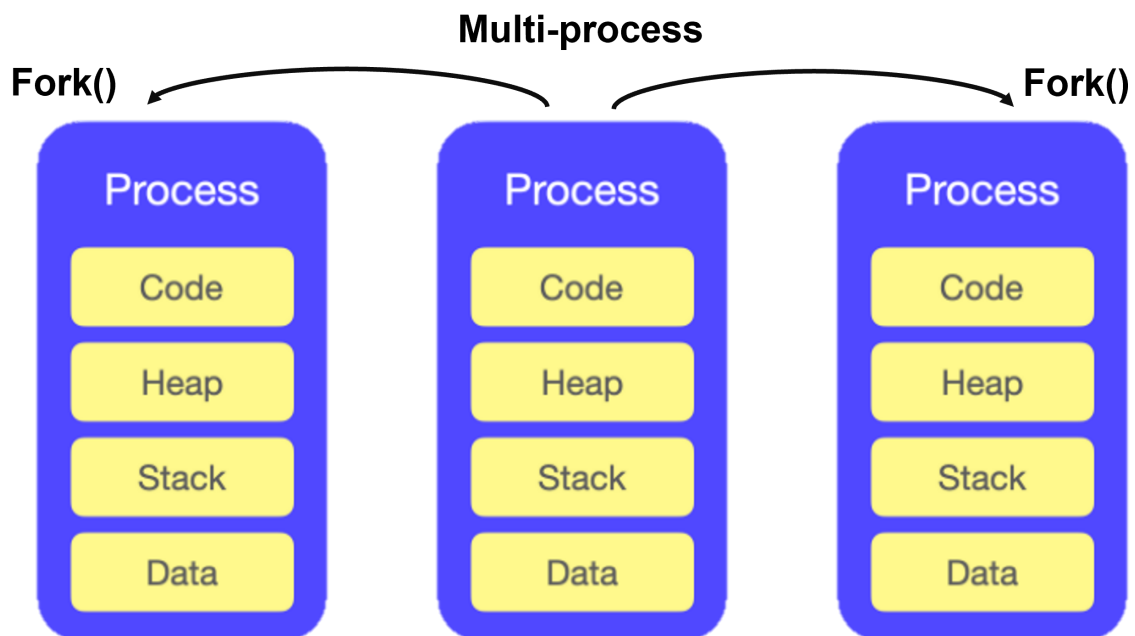


- 스레드는 프로세스 내에서 각각 stack 만 따로 할당 받고 다른 영역은 공유한다.
- 같은 프로세스 안에 있는 여러 스레드들은 같은 힙 공간을 공유한다.
- 한 스레드가 프로세스 자원을 변경하면, 다른 이웃 스레드도 그 변경 결과를 즉시 볼 수 있다.

### 3. 멀티 프로세스 vs 멀티 스레드

- 멀티 프로세스, 멀티 스레드는 한 응용프로그램에 대한 처리방식

#### 멀티 프로세스



- 하나의 응용프로그램을 여러 개의 프로세스로 구성하여 각 프로세스가 하나의 작업을 처리하도록 하는 것
- 장점
  - 여러개의 자식 프로세스 중 하나에 문제가 발생하면 그 자식 프로세스만 죽는 것 이 상으로 다른영향이 확산되지 않는다.
- 단점
  - 자원 소모적, 개별 메모리 차지

- (\*)Context Switching 비용이 큼
  - 캐쉬 메모리 초기화 등 무거운 작업이 진행되고 많은 시간이 소모되는 등의 오버헤드가 발생
  - 프로세스는 각각의 독립된 메모리 영역을 할당받았기 때문에 프로세스 사이에서 공유하는 메모리가 없어, Context Switching가 발생하면 캐쉬에 있는 모든 데이터를 모두 리셋하고 다시 캐쉬 정보를 불러와야함
- 동기화 작업이 필요하지 않다.

## 멀티 스레드

- 하나의 응용프로그램을 여러 개의 스레드로 구성하고 각 스레드로 하여금 하나의 작업을 처리하도록 하는 것
- 장점
  - 공유된 자원으로 통신 비용 절감
  - 공유된 자원으로 메모리가 효율적임
  - context switching 비용이 적음
- 단점
  - 주의 깊은 설계가 필요하다.
  - 디버깅이 까다롭다.
  - 멀티 스레드의 경우 자원 공유의 문제가 발생한다. (동기화 문제)
  - 하나의 스레드에 문제가 발생하면 전체 프로세스가 영향을 받는다.

**이렇게 보면 멀티 스레드가 좋아보이는데 왜 멀티 프로세스를 사용할까?**

예 ) 구글 크롬 / 인터넷 익스플로러

## 4. 멀티 코어

- 동시성 : 프로세서하나가 이거조금 하면서 여러작업 돌아가면서 일부분씩 진행하는 것 . 진행중인 작업을 바꾸는것을 context switching → 다양한 알고리즘과 방식이 쓰인다.

- 병렬적 : 프로세서하나에 코어 여러개가 달려서 각각 동시에 작업들을 수행. 듀얼코어, 쿼드코어, 옥타코어같은 명칭이 붙은 멀티코어 프로세서가 달린 컴퓨터에서 가능 → CPU 속도가 발열등 물리적 제약때문에 그 대안으로 코어를 여러개 달아서 작업 분담할 수 있도록 한 것.

#### (\*) Context Switching

- CPU에서 여러 프로세스를 돌아가면서 작업을 처리하는 과정

#### [참고]

<https://velog.io/@nnnyeong/OS-멀티프로세스-멀티스레드-멀티-프로그래밍-멀티프로세스-멀티스레드에서의-데이터-통신>

<https://www.youtube.com/watch?v=1grtWKqTn50>