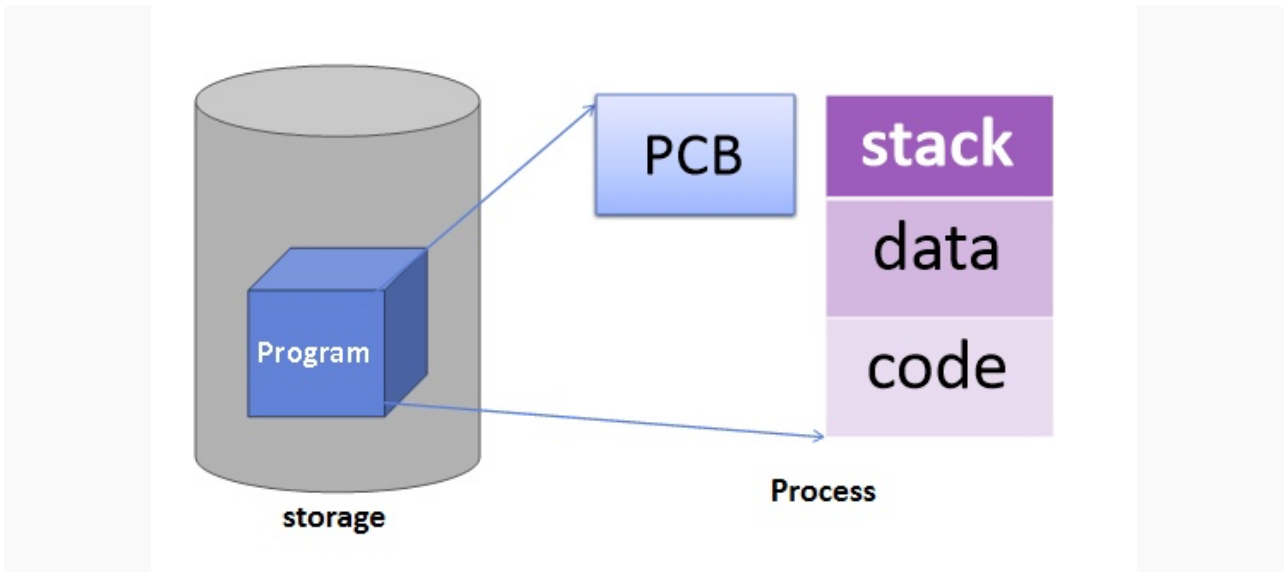


## PCB와 Context Switching



### PCB(Process Control Block)

프로세스 메타데이터들을 저장해 놓은 곳입니다. 하나의 PCB 안에는 하나의 프로세스 정보가 담겨있습니다.

프로그램 실행 ➡ 프로세스 생성 ➡ 프로세스 주소 공간( 코드, 데이터, 스택 )에 생성 ➡ 프로세스의 메타데이터들이 PCB에 저장

### PCB가 필요한 이유는?

CPU에서는 프로세스의 상태에 따라 교체 작업이 이루어집니다. (인터럽트가 발생해서 할당받은 프로세스가 Block 상태가 되고 다른 프로세스를 running으로 바꿀 때)

이때, 앞으로 다시 수행할 Block 상태의 프로세스의 상태값을 PCB에 저장 해두는 것입니다.

### PCB의 관리 방식

Linked List 방식으로 관리가 되어 PCB List Head에 PCB들이 생성될 때마다 붙게 됩니다. 주솟값으로 연결이 이루어져 있는 연결 리스트 형태로, 삽입 삭제가 용이합니다.

즉, 프로세스가 생성되면 해당 PCB가 생성되고 프로세스 완료 시 제거가 됩니다.

이렇게 수행 중인 프로세스를 변경할 때, CPU의 레지스터 정보가 변경되는 것을 Context Switching이라고 합니다.

## Context Switching

CPU가 현재 실행하고 있는 Task(Process, Thread)의 상태를 저장하고, 다음 진행할 Task의 상태 및 Register 값들에 대한 정보(Context)를 읽어 새로운 Task의 Context 정보로 교체하는 과정

### Context Switching 수행 과정

1. Task의 대부분 정보는 Register에 저장되고 PCB(Process Control Block)로 관리가 되고 있습니다.
2. 현재 실행하고 있는 Task의 PCB 정보를 저장하게 됩니다.(Process Stack, Ready Queue)
3. 다음 실행할 Task의 PCB 정보를 읽어 Register에 적재하고 CPU가 이전에 진행했던 과정을 연속적으로 수행할 수 있게 됩니다.