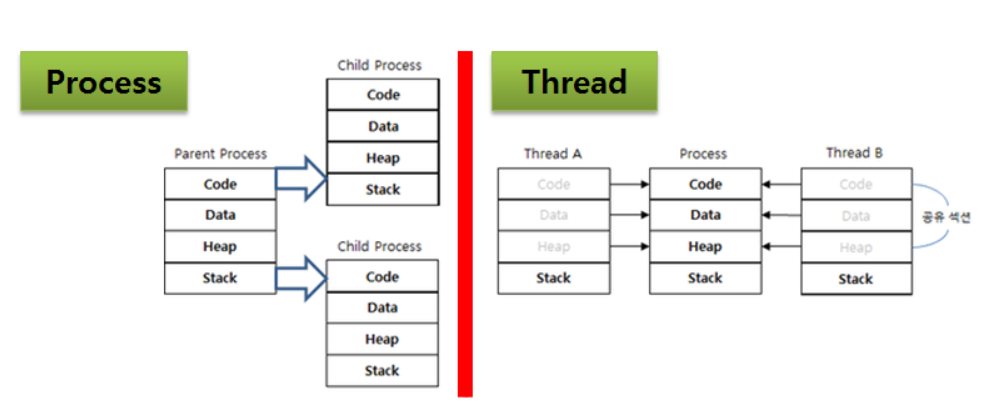
**<프로세스와 스레드에 대해서 조사 >**

•프로세스와 스레드의 정의 및 차이점

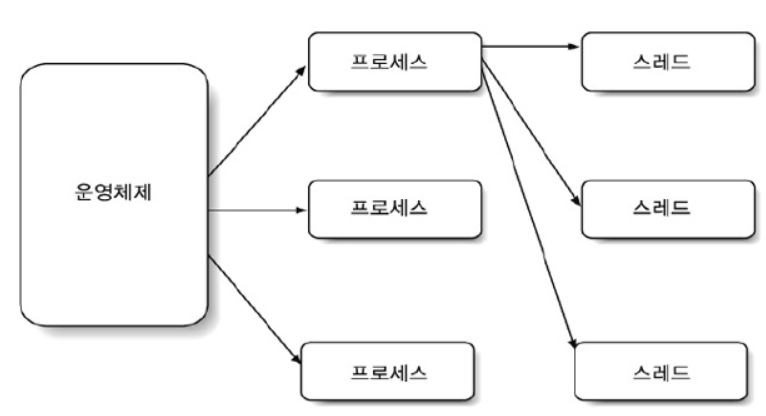
-프로세스: 프로세스(process)는 컴퓨터에서 연속적으로 실행되고 있는 [컴퓨터 프로그램](https://ko.wikipedia.org/wiki/%EC%BB%B4%ED%93%A8%ED%84%B0_%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%A8)을 말한다. 종종 스케줄링의 대상이 되는 작업(task)이라는 용어와 거의 같은 의미로 쓰인다. 여러 개의 프로세서를 사용하는 것을 [멀티프로세싱](https://ko.wikipedia.org/wiki/%EB%A9%80%ED%8B%B0%ED%94%84%EB%A1%9C%EC%84%B8%EC%8B%B1)이라고 하며 같은 시간에 여러 개의 프로그램을 띄우는 시분할 방식을 [멀티태스킹](https://ko.wikipedia.org/wiki/%EB%A9%80%ED%8B%B0%ED%83%9C%EC%8A%A4%ED%82%B9)이라고 한다. [프로세스 관리](https://ko.wikipedia.org/wiki/%ED%94%84%EB%A1%9C%EC%84%B8%EC%8A%A4_%EA%B4%80%EB%A6%AC)는 [운영 체제](https://ko.wikipedia.org/wiki/%EC%9A%B4%EC%98%81_%EC%B2%B4%EC%A0%9C)의 중요한 부분이 되었다.

-스레드: 스레드(thread)는 어떠한 프로그램 내에서, 특히 [프로세스](https://ko.wikipedia.org/wiki/%ED%94%84%EB%A1%9C%EC%84%B8%EC%8A%A4) 내에서 실행되는 흐름의 단위를 말한다. 일반적으로 한 프로그램은 하나의 스레드를 가지고 있지만, 프로그램 환경에 따라 둘 이상의 스레드를 동시에 실행할 수 있다. 이러한 실행 방식을 [멀티스레드](https://ko.wikipedia.org/wiki/%EB%A9%80%ED%8B%B0%EC%8A%A4%EB%A0%88%EB%94%A9)(multithread)라고 한다.

-차이점: 프로세스는 운영체제 관점에서 별도의 실행 흐름을 구성하는 단위이고, 스레드는 운영체제 관점이 아닌 프로세스 관점이다. 멀티프로세스와 멀티스레드는 양쪽 모두 여러 흐름이 동시에 진행된다는 공통점을 가지고 있다. 하지만 멀티프로세스에서 각 프로세스는 독립적으로 실행되며 각각 별개의 메모리를 차지하고 있는 것과 달리 멀티스레드는 프로세스 내의 메모리를 공유해 사용할 수 있다. 또한 프로세스 간의 전환 속도보다 스레드 간의 전환 속도가 빠르다.

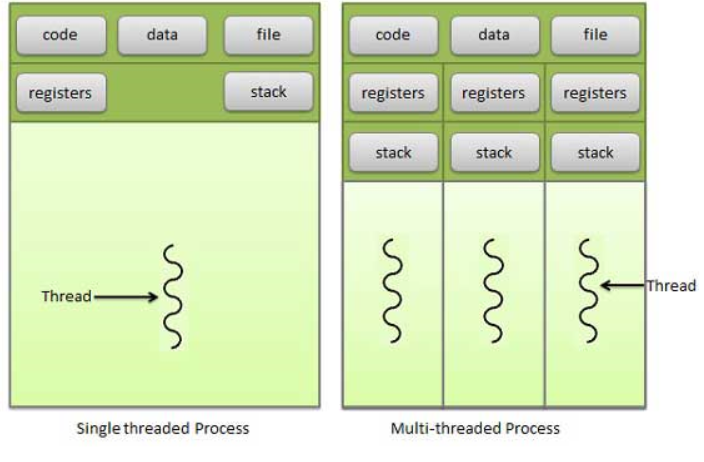


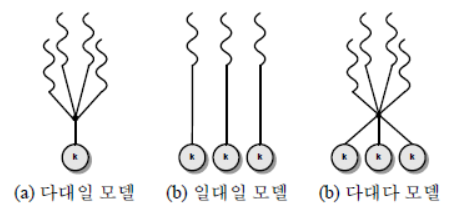
Ex) 프로세스: 각각의 은행 지점  
 스레드: 은행 지점 하나에 속한 고객 창구 여러 개



•스레드 사용 이유 및 스레드 모델

-사용 이유: 스레드는 스택은 따로따로이지만, 코드 영역과 데이터 영역은 하나를 공유하기 때문에 데이터 영역에 속하는 변수를 통해서 쉽고 빠르고 편하게 통신할 수 있다.

-스레드 모델:

프로세스에 하나의 제어 모델이 있으면 단일 스레드 이며, 프로세스가 다수의 제어 스레드를 가진다면 다중 스레드 모델이다.

•Python에서 프로세스 및 스레드를 사용하는 방법

파이썬 프로그램은 기본적으로 하나의 쓰레드(Single Thread)에서 실행된다. 즉, 하나의 메인 쓰레드가 파이썬 코드를 순차적으로 실행한다. 코드를 병렬로 실행하기 위해서는 별도의 쓰레드(Subthread)를 생성해야 하는데, 파이썬에서 쓰레드를 생성하기 위해서는 threading 모듈 (High 레벨) 혹은 thread 모듈 (Low 레벨)을 사용할 수 있다. 일반적으로 쓰레드 처리를 위해서는 thread 모듈 위에서 구현된 threading 모듈을 사용하고 있으며, thread 모듈은 거의 사용하고 있지 않다. 파이썬은 전역 인터프리터 락킹(Global Interpreter Lock) 때문에 특정 시점에 하나의 파이썬 코드만을 실행하게 되는데, 이 때문에 파이썬은 실제 다중 CPU 환경에서 동시에 여러 파이썬 코드를 병렬로 실행할 수 없으며 인터리빙(Interleaving) 방식으로 코드를 분할하여 실행한다. 다중 CPU 에서 병렬 실행을 위해서는 다중 프로세스를 이용하는 multiprocessing 모듈을 사용한다.

스레드를 사용하기 위해서는 일반적으로 threading.Thread를 상속받은 클래스객체를 생성하여 사용하며, 아래 나열된 메서드들을 주로 이용한다. 생성자를 재정의하는 경우 반드시 Thread.\_\_init\_\_()을 수행해야 한다.

>Thread.start()

스레드를 시작할때 사용함

>Thread.run()

스레드의 주요 동작을 정의함

>Thread.join([timeout])

스레드가 종료되기를 기다린다. timeout이 정의된경우, 최대 정의된 시간(초)만큼 기다린다.