

# 4

## Threads & Concurrency

Created time @April 30, 2024 4:03 PM

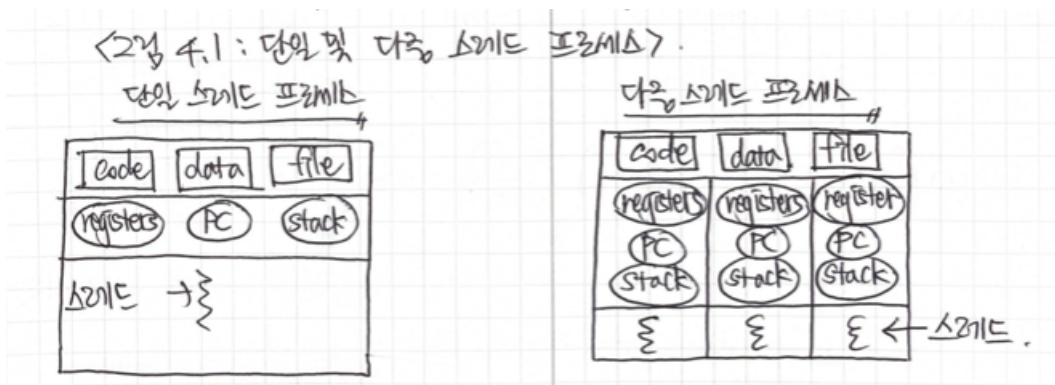
Subject 운영체제

### 4.1 개요

스레드는 CPU 이용의 기본 단위입니다.

스레드는 스레드 ID, 프로그램 카운터, 레지스터 집합, 그리고 스택으로 구성됩니다.

단일 스레드 프로세스는 하나의 제어 스레드를 가지고 있고, 다중 스레드 프로세스는 다수의 제어 스레드를 가지고 있어 프로세스가 동시에 하나 이상을 작업을 수행할 수 있습니다.

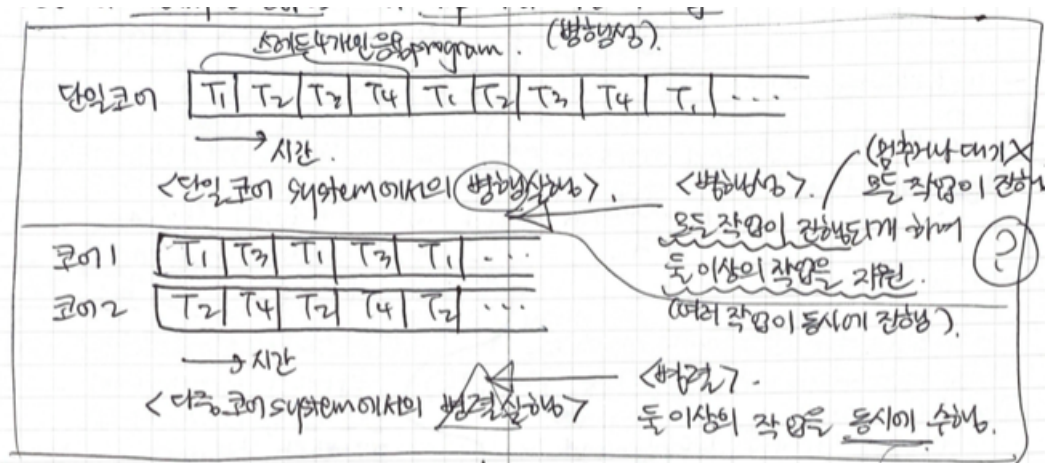


#### 4.1.2 장점

다중 스레드 프로그래밍의 장점은 4가지 큰 부류로 나눌 수 있습니다.

1. 응답성: 응용 프로그램의 일부분이 봉쇄되거나, 응용 프로그램이 긴 작업을 수행하더라도 프로그램이 계속되는 것을 허용할 수 있게 되는 것입니다.
2. 자원 공유: 한 응용 프로그램이 같은 주소 공간 내에 여러 개의 다른 작업을 하는 스레드를 가질 수 있게 되는 것입니다.
3. 경제성: 일반적으로 스레드 생성은 프로세스 생성보다 시간과 메모리를 덜 소비하기 때문에, 문맥 교환을 통해 여러 프로세스를 번갈아가며 실행하는 것보다 경제적인 것입니다.
4. 규모 적응성: 각각의 스레드가 다른 처리기에서 병렬로 수행될 수 있는 것이 장점입니다.

### 4.2 다중 코어 프로그래밍



단일 컴퓨팅 칩에 여러 컴퓨팅 코어를 배치하는 것을 다중 코어라고 하며, 다중 스레드 프로그래밍은 이러한 여러 컴퓨팅 코어를 보다 효율적으로 사용하고 병행성을 향상시키는 기법을 제공합니다.

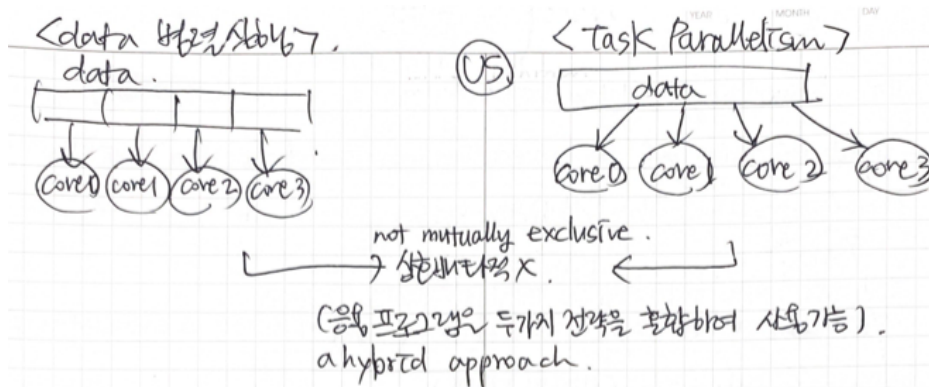
반면에, 여러 코어가 있는 시스템에서 병행성은 시스템이 각 코어에 별도의 스레드를 할당할 수 있기 때문에 일부 스레드가 병렬로 실행될 수 있음을 의미합니다.

## 4.2.2 병렬 실행의 유형

일반적으로 두 가지 병렬 실행 유형이 존재합니다.

첫 번째는 동일한 데이터 부분집합을 다수의 계산 코어에 분배한 뒤, 각 코어에서 연산을 실행하는 '데이터 병렬 실행'입니다.

두 번째는 '태스크 병렬 실행'으로, 데이터가 아니라 태스크를 다수의 코어에 분배하는 것입니다.



## 4.3 다중 스레드 모델

스레드를 위한 지원은 사용자 수준과 커널 수준에서 제공되는 것으로 나뉩니다.

사용자 스레드는 커널 위에서 지원되고 커널의 지원 없이 관리되는 반면에, 커널 스레드는 운영체제에 의해 직접 지원되고 관리되는 것이 특징입니다.

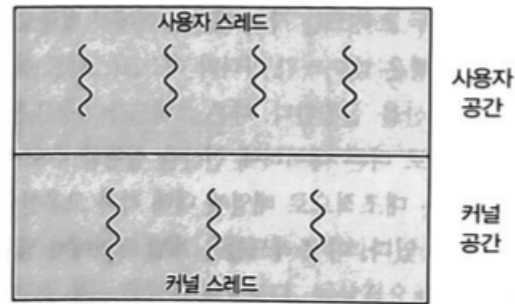
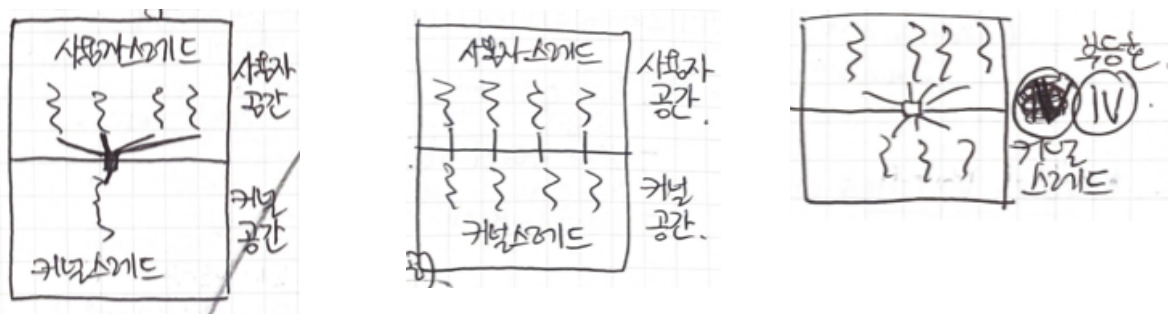


그림 4.6 사용자와 커널 스레드

사용자 스레드와 커널 스레드는 어떤 연관 관계가 존재해야하는데, 이 연관 관계를 확립하는 세 가지 일반적인 방법이 '다대일', '일대일', '다대다' 모델입니다.



## 4.4 스레드 라이브러리

### 4.5 암묵적 스레딩

앞서 설명해온 병행 및 병렬 응용의 설계를 도와주는 한 가지 방법은 스레딩의 생성과 관리 책임을 응용 개발자로부터 → 컴파일러와 실행시간 라이브러리에게 넘겨주는 것입니다. 이를 암묵적 스레딩이라고 합니다.

암묵적 스레딩을 이용하여 다중 코어 처리기를 활용할 수 있는 응용 프로그래밍을 설계하는 방법 중 2가지를 설명하겠습니다.

#### 4.5.1 스레드 풀

첫 번째는 스레드 풀입니다. 프로세스를 시작할 때 아예 일정한 수의 스레드들을 미리 풀로 만들어주어, 스레드가 무한정 만들어져서 CPU 시간과 메모리 공간 간은 시스템 자원이 고갈되는 것을 방지하는 방법입니다.

#### 4.5.2 Fork Join

메인 부모 스레드가 하나 이상의 자식 스레드를 생성한 다음 자식의 종료를 기다린 후 join하고 그 시점부터 자식의 결과를 확인하고 결합할 수 있는 방법입니다.

