

A process is an Operating System concept and represents one execution of a program (roughly). That program might have many threads [of execution].

프로세스는 '작업'의 단위이다.

↑ 스레드는 '작업'에 대한 '실행'이다.

코어는 하나의 스레드를
실행시키는 장치이다.

A core is a hardware concept. It represents the ability to execute one [or more] thread[s].

Hyperthreading is an Intel term for the ability of a core to execute 2 threads. However, from the OS point of view each of those core threads is treated like a separate core. So, from the OS point of view, each core executes a single thread. If your "physical" core can execute 2 threads, to the OS it is 2 cores.

↑ 중요 전제!

Other than the fact that a core [from the OS point of view] can only execute one thread at a time, there isn't any natural relationship between cores and process threads. A process thread can run on any core [or core thread].

↑ process의 thread는 cpu의 thread (=core)에 의해
실행되는 것이다.

1코어 1스레드 CPU 쓰던 시절에도 스레드 수십

글쓴이: 익명 사용자 / 작성시간: 일, 2016/02/21 - 5:07오후

1코어 1스레드 CPU 쓰던 시절에도 스레드 수십 개씩 띄우는 거 가능했습니다.

엄밀함은 조금 떨어지더라도 쉽게 이해할 수 있게 나눠 설명해드리죠.

CPU가 n코어 m스레드라고 할 때...

n은 말 그대로 "물리적인" CPU 코어 유닛 갯수입니다. 명령어를 메모리에서 뽑아 해석하고 실행하는 반도체 유닛이 n개 있는겁니다.

m은 "논리적" 코어 갯수이며, 경우에 따라 "동시에 실행가능한 스레드 갯수"이기도 합니다. n=m인 경우가 가장 직관적인데 가끔 m이 n보다 크기도 합니다. 이 경우 물리적 코어 하나가 스레드 두 개 이상을 동시에 실행 가능하다는 의미가 됩니다.

구체적으로 어떻게 그렇게 하는가는 SMT(https://en.wikipedia.org/wiki/Simultaneous_multithreading) 같은 걸 찾아보세요. 간단한 기본 원리는 "스레드 하나 가지고는 (물리적) 코어 하나를 100% 활용할 수 없으니 여러 스레드를 동시에 쪼개놓아보자" 입니다. 또 덧붙이자면, SMT는 조금 학술적인 용어이고 인텔의 마케팅 용어로는 "하이퍼스레딩"이라고 합니다.

운영체제가 스케줄링을 할 때, "동시에 실행 가능한 스레드 수", 즉 할당 가능한 스레드의 슬롯은 m입니다. 하지만, 메모리(및 swap 영역)가 허용하는 한 "소프트웨어적" 스레드 수는 얼마든지 많을 수 있습니다. 이들 중 일부는 하드웨어(유저 입력, 네트워크 패킷, 타이머 인터럽트 등)를 기다리며 잠들어 있을 수 있고, 운영체제는 짧은 주기(보통 수 ms)로 "실행할 준비가 된 스레드"들 중에서 m개를 선택하여 동시에 실행하게 됩니다.

→ "cpu thread"를 의미.

