

동시성과 병렬성

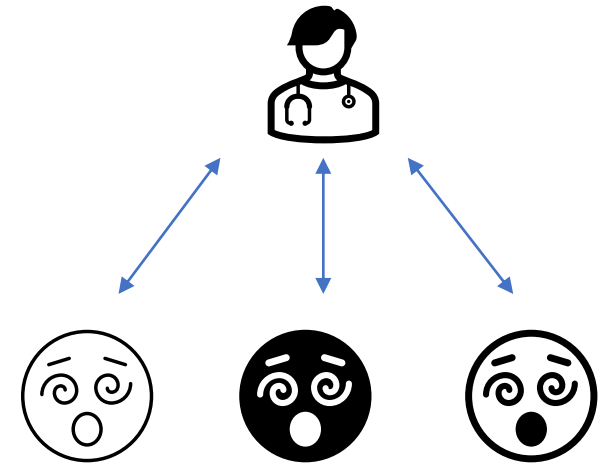
PyQT를 활용한 GUI 프로그래밍

동시성과 병렬성

동시성 (Concurrency)	병렬성 (Parallelism)
동시에 실행되는 것처럼 보이는 것	실제로 동시에 여러 작업이 처리되는 것
한 사람이 여러 일을 조금씩 아주 빨리 처리하는 방식	여러 사람이 여러 일을 나눠서 처리하는 방식
싱글 코어에서 멀티 스레드	멀티 코어에서 멀티 스레드
Concurrent Programming	Parallel Programming

동시성(Concurrency)의 예 (1/2)

- 동시성 (=멀티태스킹)
 - 여러분이 회사에서 업무를 보는데 처리해야 할 일이 A, B, C, D가 있을 때 각 일을 잘게 분할한 후 이를 조금씩 번갈아 가면서 처리하는 방식을 의미
 - 멀티태스킹에서 여러분은 **한 순간에 하나의 일을 하고 있지만** 이를 아주 빨리 번갈아 할 수 있다면 마치 동시에 처리하는 것처럼 보인다는 의미임



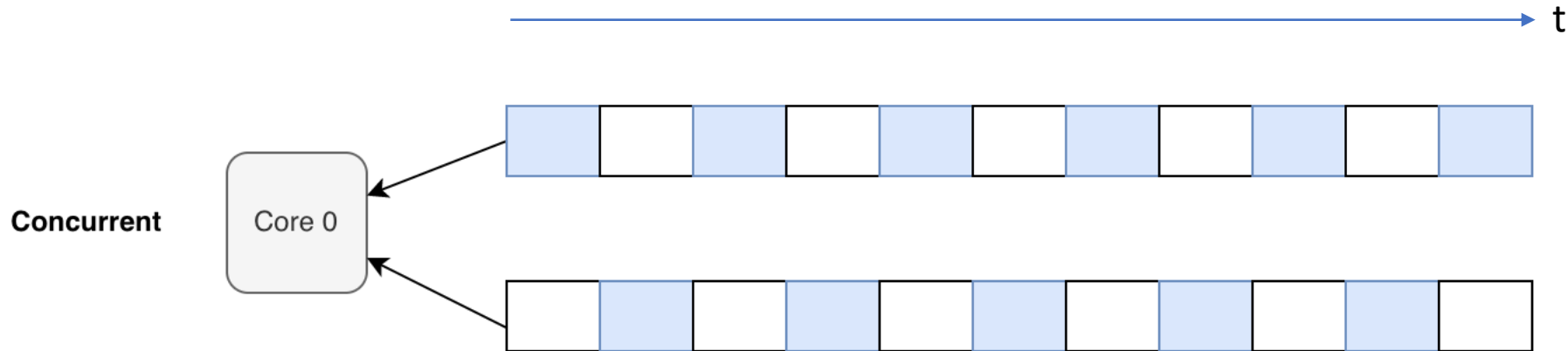
의사가 여러 명의 간호사와 일함

동시성(Concurrency)의 예 (2/2)

- 동시성 (=멀티태스킹)
 - 컴퓨터를 생각해 봅시다. 여러분은 인터넷을 하다가 이메일도 확인하고 가끔은 메신저도 확인합니다. 그런데 컴퓨터가 하나의 일을 모두 끝내야 다른 일을 처리해준다면 컴퓨터를 사용하는 입장에서 매우 불편할 겁니다.
 - 그래서 컴퓨터의 운영체제들은 멀티태스킹을 지원하며 이를 위해 하드웨어인 CPU도 태스크(task)와 태스크 사이의 전환을 빨리할 수 있도록 다양한 하드웨어적인 지원을 하고 있습니다.
 - Context switching

동시성(Concurrent)

- Concurrent의 핵심은 하나의 CPU는 **한 순간에 하나의 일을 처리**



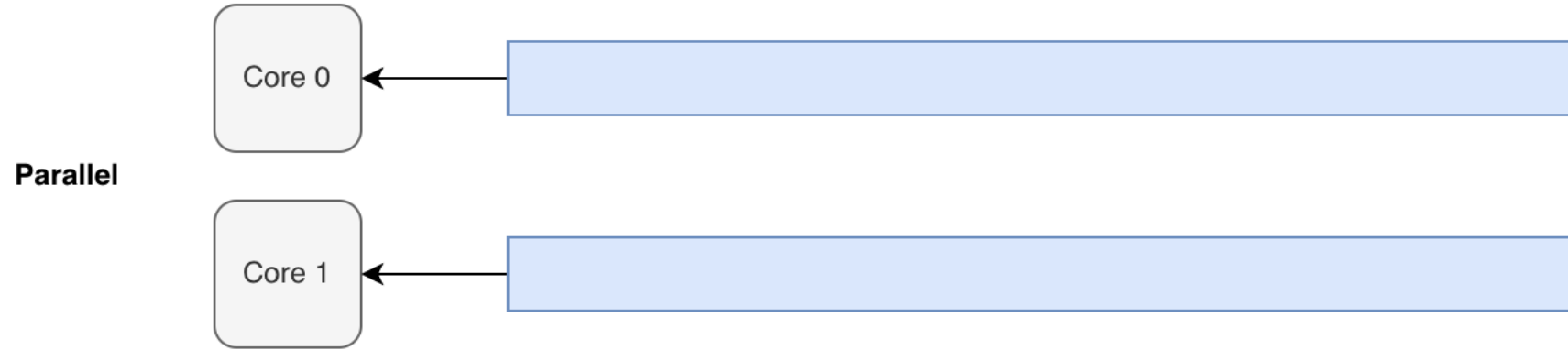
병렬성(parallelism)

- 병렬성은 진짜로 동시에 여러 개의 일을 처리하는 것
 - 2000년 이후 멀티코어 프로세서(processor)가 빠르게 보급됨
 - PC 뿐만 아니라 스마트폰 CPU도 이미 멀티코어 프로세서임
 - 이처럼 일을 처리할 수 있는 '일꾼'이 여러 개라면 당연히 한 순간에 여러 개의 일(태스크)을 처리할 수 있을 텐데 이를 병렬성이라고 함

시간	코어0	코어1	코어2	코어3
T0	A0	B0	C0	D0
T1	A1	B1	C1	D1
T2	A2	B2	C2	D2
T3	A3	B3	C3	D3

한 순간에 4개의 태스크가 동시에 처리됨

Parallelism



병렬성이 항상 좋을까?

- 병렬성은 물리적인 코어의 개수에 제한을 받음
 - 쿼드코어(4개 코어) CPU라면 최대 8개 정도 병렬처리가 가능함(하이퍼스레딩)
 - 실제 작업이 병렬적으로 처리할 수 없는 경우가 많음
- 처리하는 작업이 주로 I/O 위주의 작업인 경우
 - CPU로 데이터를 처리하는 시간보다 대부분의 데이터를 가져오는데 시간을 소비함
 - CPU가 놀고 있을 때 다른 작업을 처리할 수 있음
 - 동시성이 효과적인 케이스