동시성과 병렬성

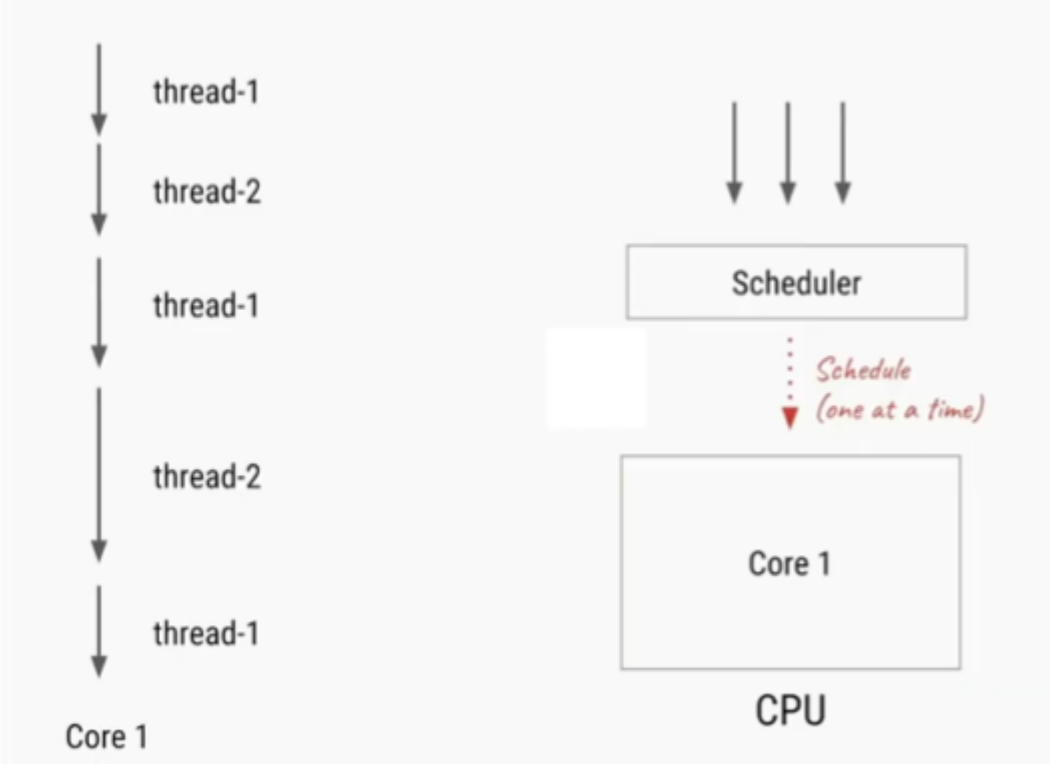
Date	@2021/11/25

Concurrency(동시성)
Parallelism(병렬성)
7가지 동시성 모델
하이젠버그(Hisenbug)
Java concurrency
Visibility
Memory Model - Reordering
Java Parallelism를 사용하려면
Java stream example
Python Concurrency - gevent
Python Parallelism - Pool
암달의 법칙(Amdahl's law)

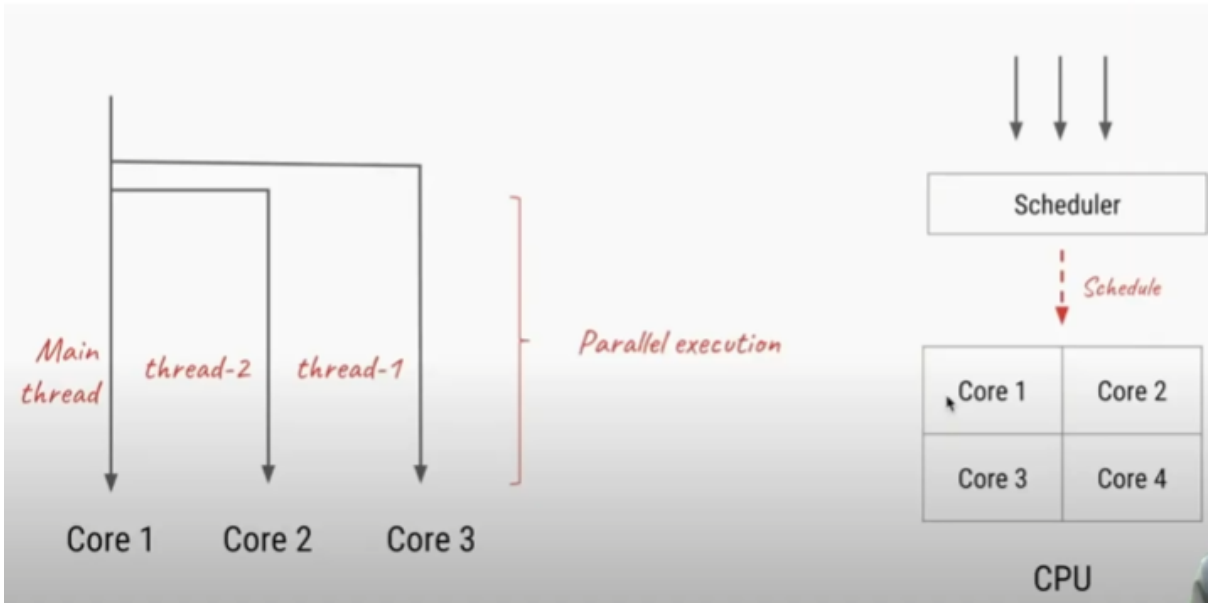
Concurrency is about **dealing with** lot of things at once.

Parallelism is about **doing** lot of things at once.

Concurrency(동시성)



Parallelism(병렬성)



7가지 동시성 모델

1. 스레드와 잠금장치
2. 함수형 프로그래밍
3. 클로저 방식
4. 액터
5. 순차 프로세스 통신(CSP)
6. GPGPU
7. 람다 아키텍쳐

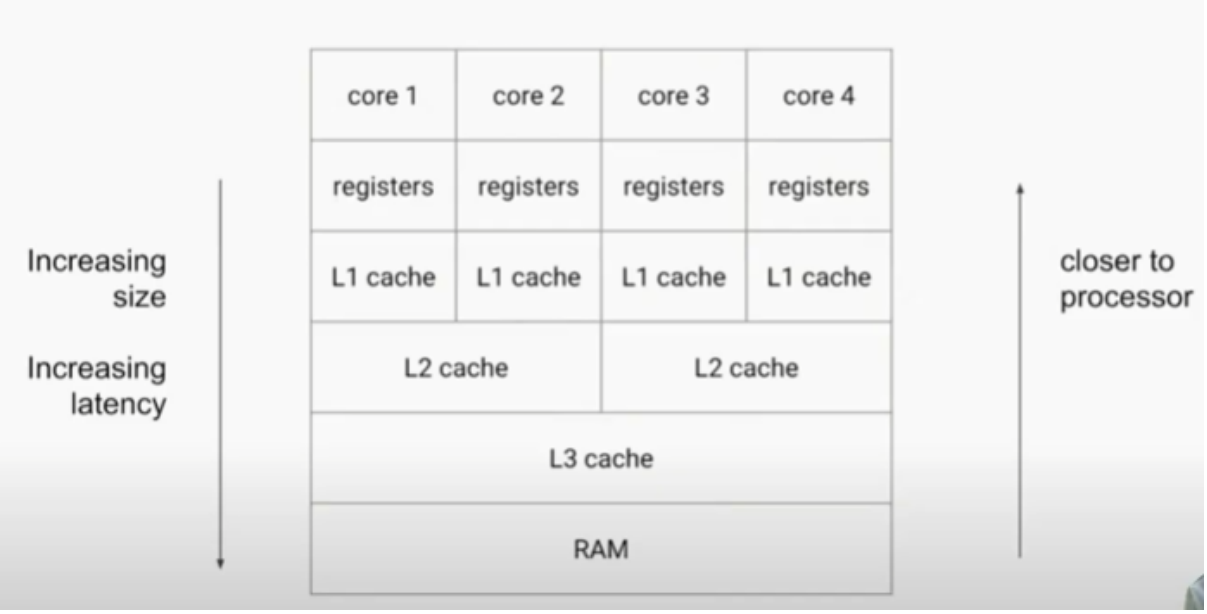
하이젠버그(Hisenbug)

하이젠베르크의 불확정성원리에서 이름을 따온 하이젠버그는디버거나 혹은 다른 수단으로 이를 찾아내려 하는 순간 사라지거나 모습을 바꾸는 버그를 뜻한다.

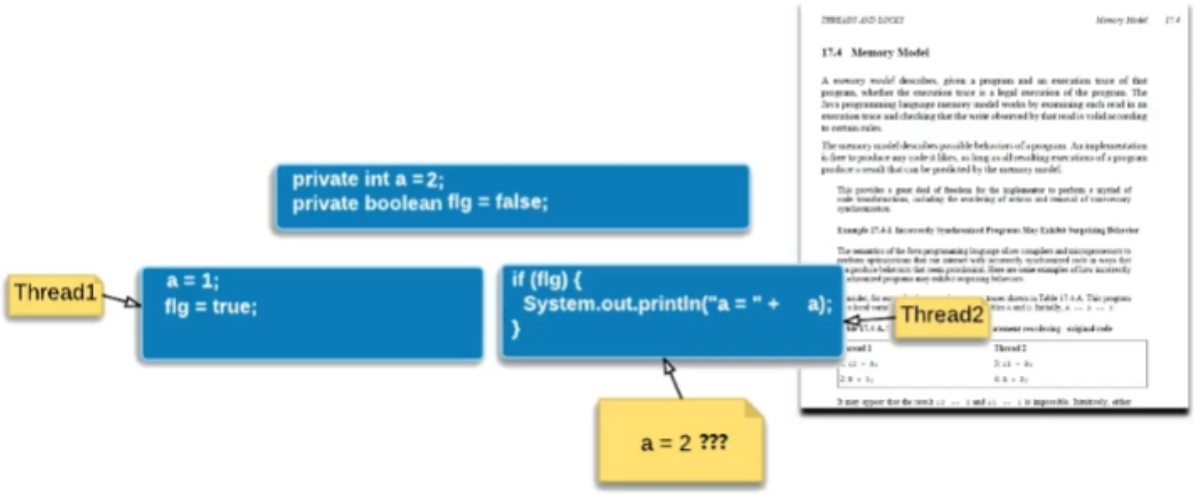
Java concurrency

- Locks / synchronized
- Atomic classes
- Concurrent data structures (ex: ConcurrentHashMap, BlockingQueue)
- CompletableFuture
- CountdownLatch / Phaser / CyclicBarrier / Semaphore etc.

Visibility



Memory Model - Reordering



Java Parallelism를 사용하려면

- Threads
- ThreadPoll
  - ExecutorService
  - ForkJoinPool

    - Custom ThreadsPools
- ParallelStream
- Requires > 1 CPU core

Java stream example

```
final long startTime = System.currentTimeMillis();
Arrays.asList(1, 2, 3, 4)
      .stream()
      .peek(i -> {
          try {
              TimeUnit.SECONDS.sleep(1);
          } catch (InterruptedException ignored) {
          }
      })
      .forEach(System.out::println);
System.out.println(System.currentTimeMillis() - startTime + " ms");
```

```
Run:  StreamExamples2
"C:\Program Files\Java\jdk-11.0.11+9\bin\java.exe" -javaagent:C:\Users\multicampus\AppData\L
1
2
3
4
4041 ms
```

일반적인 stream을 사용하는 경우

```
final long startTime = System.currentTimeMillis();
Arrays.asList(1, 2, 3, 4)
      .parallelStream()
      .peek(i -> {
          try {
              TimeUnit.SECONDS.sleep(1);
          } catch (InterruptedException ignored) {
          }
      })
      .forEach(System.out::println);
System.out.println(System.currentTimeMillis() - startTime + " ms");
```

```
System.setProperty("java.util.concurrent.ForkJoinPool.common.parallelism", "1");
```

```
Run:  StreamExamples2
"C:\Program Files\Java\jdk-11.0.11+9\bin\java.exe" -javaagent:C:\Users\multi
4
2
1
3
1018 ms

Process finished with exit code 0
```

parallel stream을 사용하는 예시

- 연산 등 많은 내용을 하는 곳에서 사용하면 효율적이다.
- 하지만, 모든 코어가 사용하면 다른 업무를 사용할 수 있는 코어가 없어 전체 진행이 멈추게 된다.

⇒ 그래서 코어 개수를 제한해줘야 한다. 갯수 역시 zero-basic이기에 1이라고 쓰면 2개로 제한할 수 있다.

Python Concurrency - gevent

Python Parallelism - Pool

```python
import multiprocessing
import time

def work_func(x):
    print("processing ", x)
    time.sleep(1)

if __name__ == '__main__':
    print('Cores = ', multiprocessing.cpu_count())

    start_time = time.time()

    list(map(work_func, range(0, 12)))

    elapsed_time = time.time() - start_time
    print('Pool elapsed time ', elapsed_time)
```

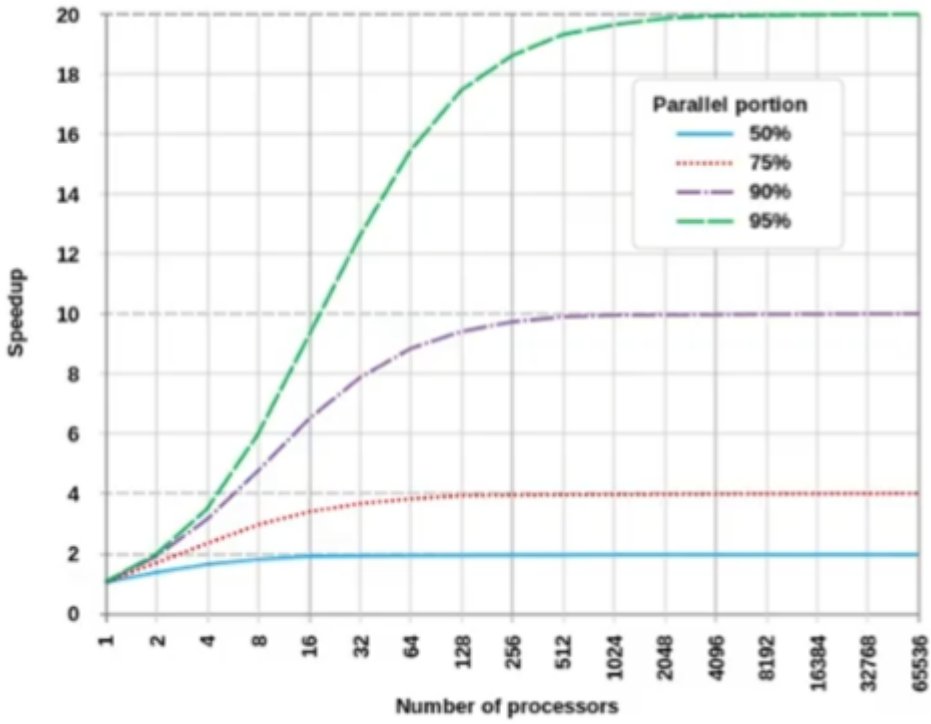
```python
from multiprocessing import Pool

    ...

    p = Pool()
    p.map(work_func, range(0, 12))
    p.close()

    ...
```

암달의 법칙(Amdahl's law)



멀티 코어를 사용하는 프로그램의 속도는 프로그램 내부에 존재하는 순차적(sequential) 부분이 사용하는 시간에 의해서 제한된다.