



쓰레드

☰ 주차	20주차
📅 스터디 일자	@2024/03/15

프로세스와 쓰레드



프로세스 : 컴퓨터에서 실행되고 있는 컴퓨터 프로그램

쓰레드 : 프로그램. 즉, 프로세스 내에서 실행되는 흐름의 단위

- 컴퓨터의 하드디스크에 있는 프로그램을 실행한 것 → **프로세스**
 - 동작 중인 프로세스 내에서 여러 작업을 동시에 수행하기 위해서는 컴퓨터의 가장 비싼 자원 중 하나인 **CPU**를 나눠 사용해야 하는데, 그 때 쓰레드를 사용하면 된당! (**CPU를 최대한 사용하기 위함**) 비싸니까요,,
 - 물론, 프로세스를 여러 개 동시에 띄워서 사용해도 (Multi-Process) 동시 처리가 가능하지만, 각각의 프로세스별로 별도의 독립적인 주소공간(코드, 스택, Heap)을 갖기 때문에 프로세스 간에 정보를 공유하기 위해서는 복잡한 방식을 이용해야 한다.
 - 반면, 프로세스 내에서 여러 개의 쓰레드를 사용하는 **멀티 쓰레드(Multi-Thread)** 방식은 스택은 쓰레드 별로 별도로 관리하지만, 나머지 코드와 Heap은 쓰레드 간에 공유가 가능하다.
 - ⇒ 동시 작업을 처리할 때 더 간단하고 빠른 처리가 가능

멀티 프로세스와 멀티 쓰레드가 정말로 꼭 필요한 경우는 동시에 많은 사용자의 요청을 처리해야 하는 서버 측 애플리케이션!

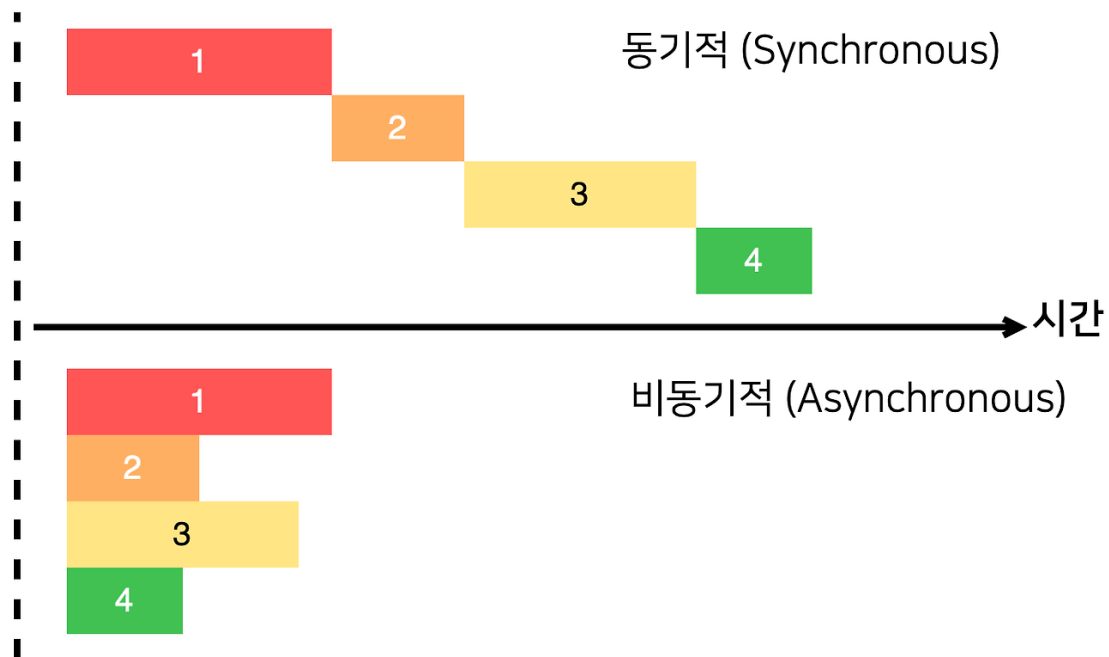
- 쓰레드를 사용하면 프로세스 내의 자원들과 메모리를 공유하기 때문에 메모리 사용량과 시스템 자원 사용량이 줄어든다.
- 쓰레드 간에 데이터를 주고받을 때도 복잡하게 통신할 필요 없이, **직접 사용할 수 있기 때문에 더욱 간단하고 빠르다.**

그러한 쓰레드에도 단점이,,?!

- 하나의 프로그램(프로세스)에서 동시에 처리해야 하는 만큼 **코딩이 복잡해진다.**
특히, 자원을 공유할 수 있기 때문에 여러 개의 쓰레드가 동시에 한 데이터를 업데이트하면 다른 쓰레드가 엉뚱한 값을 읽어올 수도 있다.
 - 다른 쓰레드가 건드리지 못하도록 Lock을 거는 작업이 꼭 필요
- 쓰레드를 많이 사용하면 오히려 더 느려질 수도 있다. → **컨텍스트 스위칭** 때문
 - CPU에 연산을 하기 위해서는 Register에 데이터가 세팅되어 있어야 해서, 동시에 여러 작업을 하려면 현재 실행 중인 Register값을 어딘가 보관해 놓고, 다른 작업을 위한 값을 Register에 다시 읽어들여 작업하고, 다시 또 다른 작업 Register를 올려 작업하는 것이 반복됨.

이러한 작업이 너무 많으면 실제 작업에 소요되는 시간보다 작업을 전환하기 위해 필요한 작업에 오히려 더 많은 시간을 사용해서 시스템이 극도로 느려질 수가 있다.

동기와 비동기



동기

어떤 작업을 실행할 때 그 작업이 끝나기를 기다리는 방식

즉, 작업이 완료될 때까지 다음 코드의 **실행을 멈추고 기다리는 것**이다.

이러한 방식은

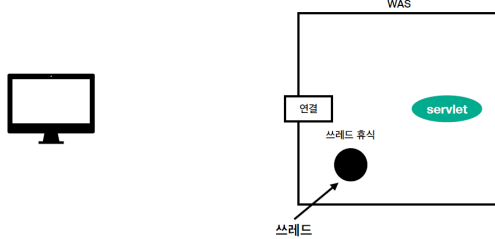
작업의 순서를 보장하고, 작업이 끝날 때까지 결과를 기다리는 것이 가능하다.

비동기

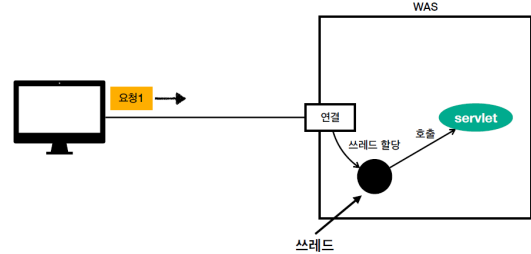
어떤 작업을 실행할 때 그 작업이 완료되지 않더라도 다음 코드를 실행하는 방식

즉, 작업이 완료되지 않았더라도 **결과를 기다리지 않고 다음 코드를 실행하는 것**이다.

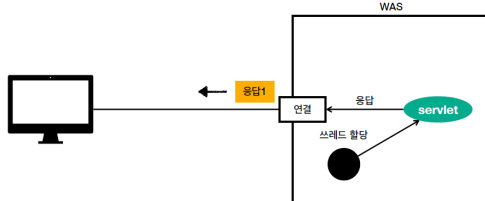
단일 요청 - 쓰레드 하나 사용



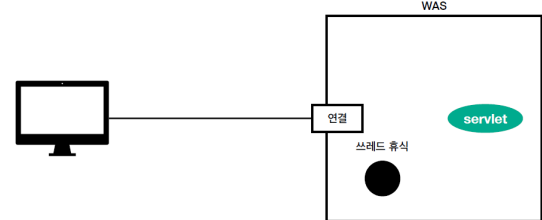
단일 요청 - 쓰레드 하나 사용



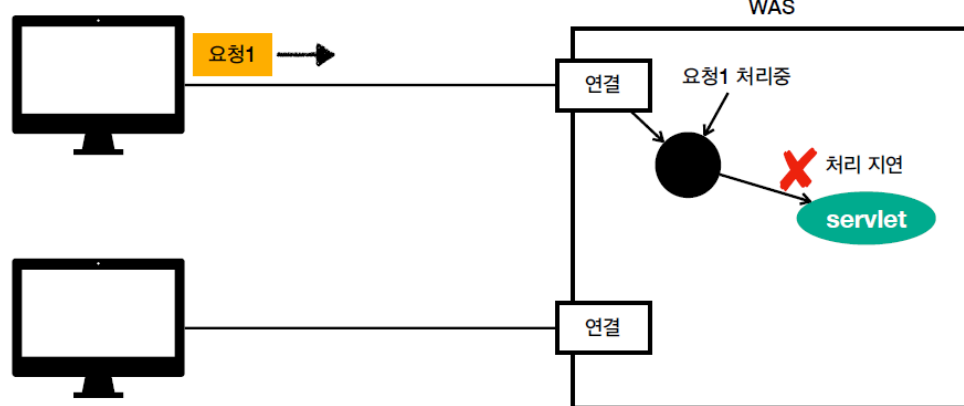
단일 요청 - 스레드 하나 사용



단일 요청 - 쓰레드 하나 사용

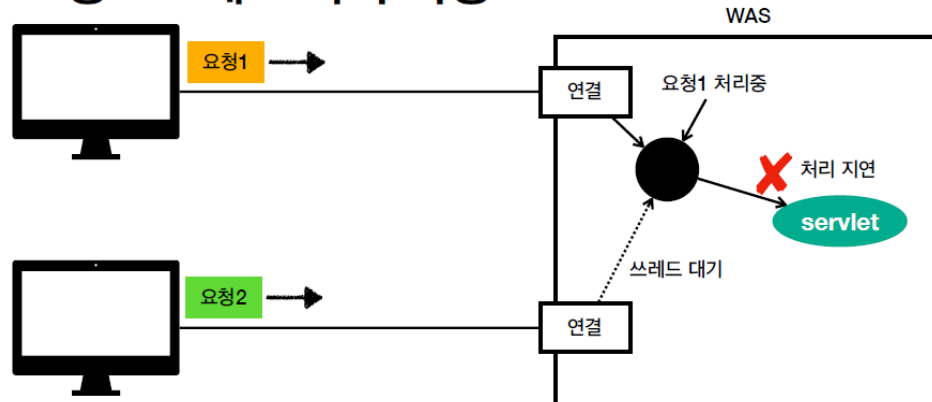


다중 요청 - 쓰레드 하나 사용



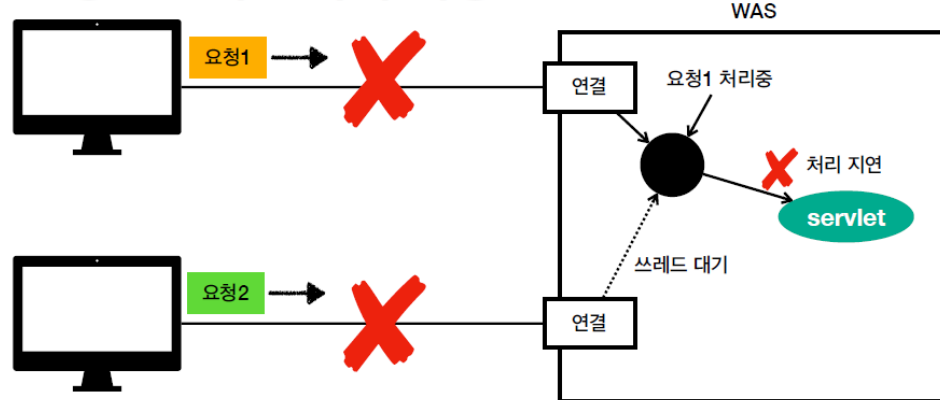
요청 1번이 서블릿을 사용하고 있는데, 잘되면 좋지만 서블릿 안에서 코드가 느려지거나 여러가지 요인에 의해 처리가 지연된다면

다중 요청 - 쓰레드 하나 사용



요청 2가 들어오면 사용할 수 있는 쓰레드가 없어서 대기하게 된다.

다중 요청 - 쓰레드 하나 사용

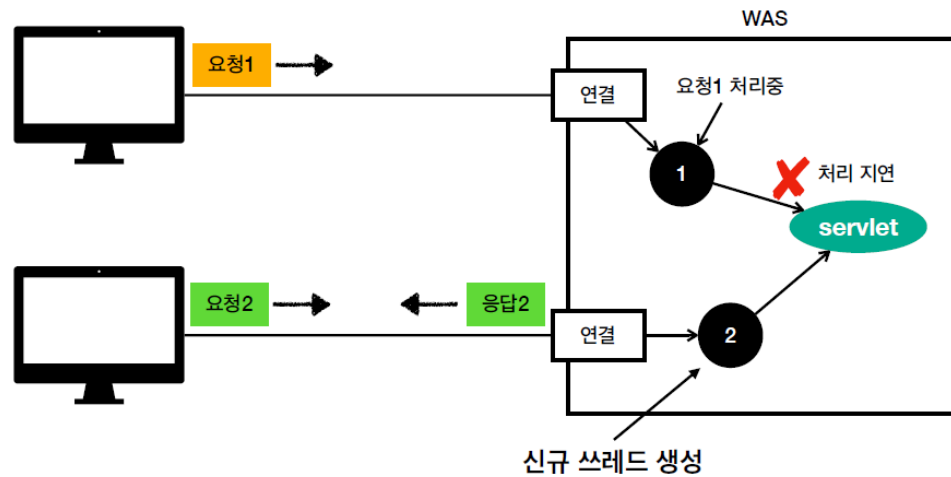


그러면 결과적으로 요청 1과 2가 모두 실패하게 된다.

둘다 타임아웃 나거나 등의 이유로 오류가 발생,,

- 이를 해결하는 방법?

요청 마다 쓰레드 생성



요청 1이 지연되고 있어도, 이런걸 고민할 필요 없이 요청 2가 오면 새로운 쓰레드를 만드는 방법이 있다. **(요청마다 쓰레드 생성)**

실행이 끝나고 응답메세지를 보낸 후 쓰레드를 없애는 식이다.

요청마다 쓰레드 생성의 장단점?

장점

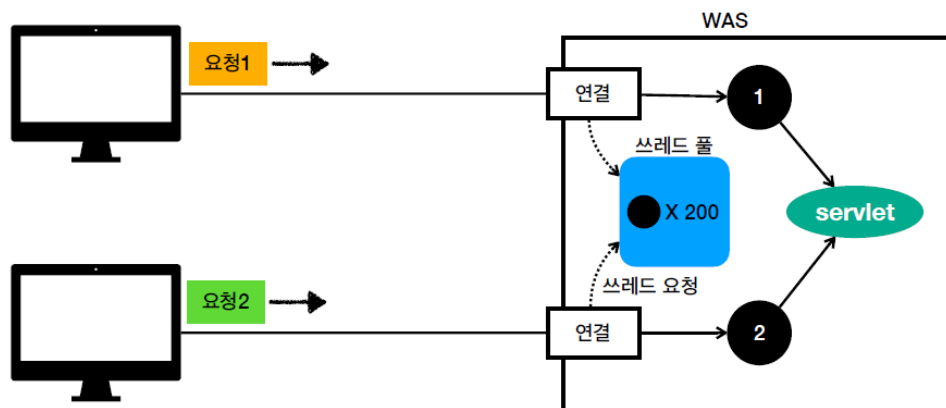
- 동시 요청 처리 가능
- 하나의 쓰레드가 지연되어도, 나머지 쓰레드는 정상 동작함.

단점

- 쓰레드는 생성 비용이 매우 비쌈
 - 고객의 요청이 올 때마다 쓰레드를 생성하면, 응답속도가 현저히 늦어진다.
- 쓰레드는 컨텍스트 스위칭 비용이 발생
- 쓰레드 생성에 제한이 없음
 - 요청이 너무 많이 오면, CPU, 메모리 임계점을 넘어서 서버가 죽을 수도 있음

→ 이를 해결하는 방법?

쓰레드 풀

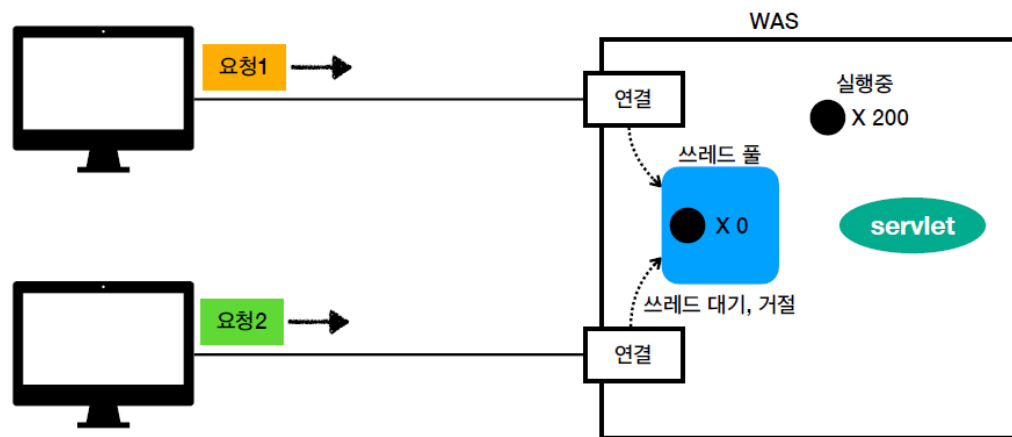


위 문제를 해결하기 위해 보통 WAS들은 내부에 쓰레드 풀이라는 것이 있다.

거기 안에서 쓰레드들이 있다가 요청이 오면 쓰레드 풀에 요청해서 쓰레드를 받아서 쓰고, 쓰레드를 다 쓰면 쓰레드를 죽이는게 아니라 쓰레드 풀에 반납한다.

그럼 다시 쓰레드 풀의 쓰레드가 200개가 된다.

쓰레드 풀



그러다가 200개 넘는 요청부터는 쓰레드풀의 쓰레드가 없으면 **대기하거나 또는 요청 거절을 할 수 있다.** (둘 중 선택 가능)

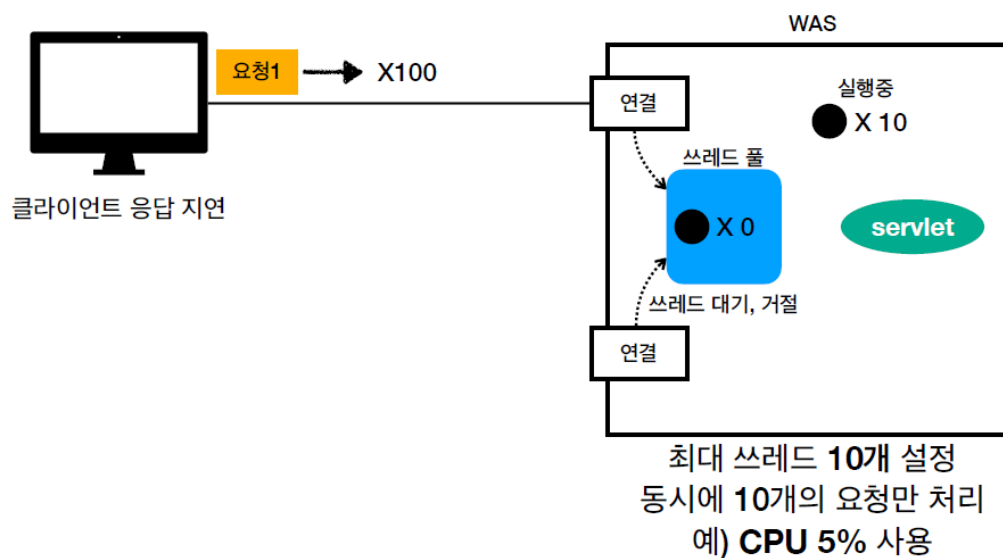
쓰레드 풀 특징

- 필요한 쓰레드를 쓰레드 풀에 보관하고 관리한다.
 - 쓰레드 풀에 생성 가능한 쓰레드의 최대치를 정해놓.
 - 톰캣은 최대 200개 기본 설정 (변경 가능)
- 쓰레드가 미리 생성되어 있으므로, 쓰레드를 생성하고 종료하는 비용이 절약되고, 응답 시간이 빠르다.
- 생성 가능한 쓰레드의 최대치가 있으므로 너무 많은 요청이 들어와도 기존 요청은 안전하게 처리할 수 있다.

쓰레드 풀을 잘 사용하기

이러한 유용한 쓰레드 풀의 주요 포인트는 **최대 쓰레드 수를 어떻게 정하나** 이다.

쓰레드 풀 - 너무 낮게 설정



- 요청은 100개가 오는데 최대 쓰레드 수를 10개로 세팅했다?
 - 나머지 90개는 모두 대기 중 → 응답 지연 → **서버 장애**

반대로 너무 많이 설정한다면?

- 쓰레드 풀을 100000개로 설정해버림
 - 요청이 너무 많아서 CPU, 메모리 임계점을 넘어서 서버가 죽는다.



적정 수를 찾는 것이 중요하다!

<https://icandoitprogramming.tistory.com/entry/3-동시-요청-멀티-쓰레드-쓰레드-풀>

<https://choi-geonu.medium.com/백엔드-개발자들이-알아야할-동시성-4-스케줄링-e684331afc77>