

야매로 서버 개발자 되는 법

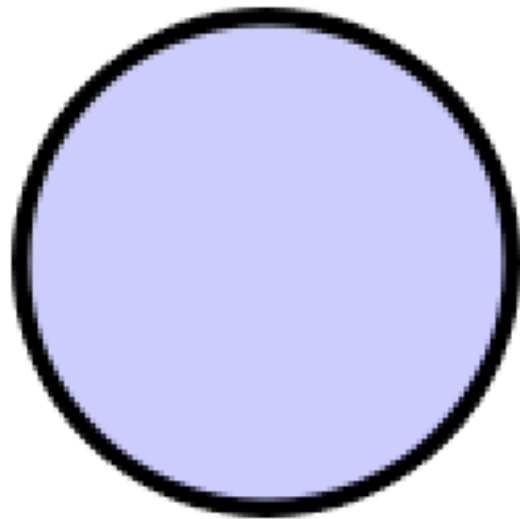
3강

멀티스레딩 이론

스레드

스레드의 정의

Process

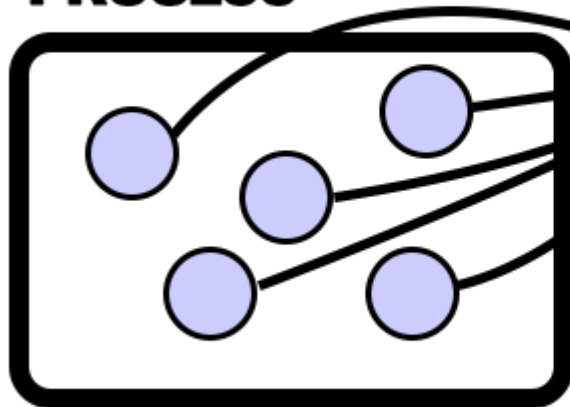


Thread



스레드의 정의

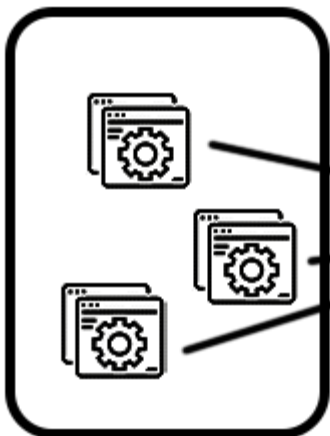
PROCESS



THREADS

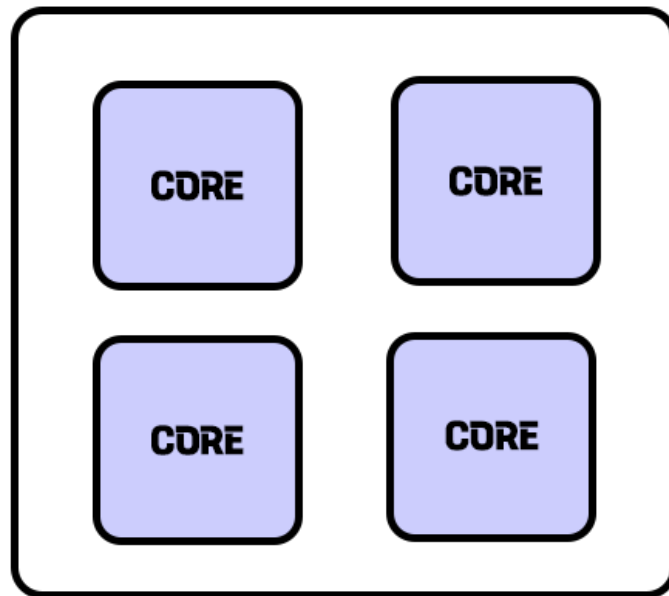
SCHEDULING

OS



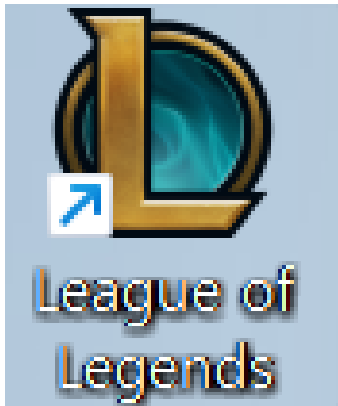
which one?

CPU

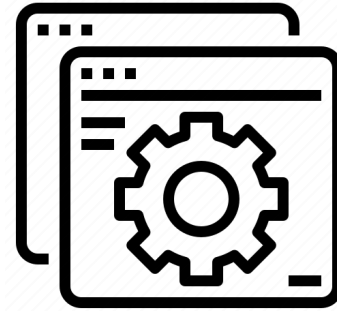


프로세스

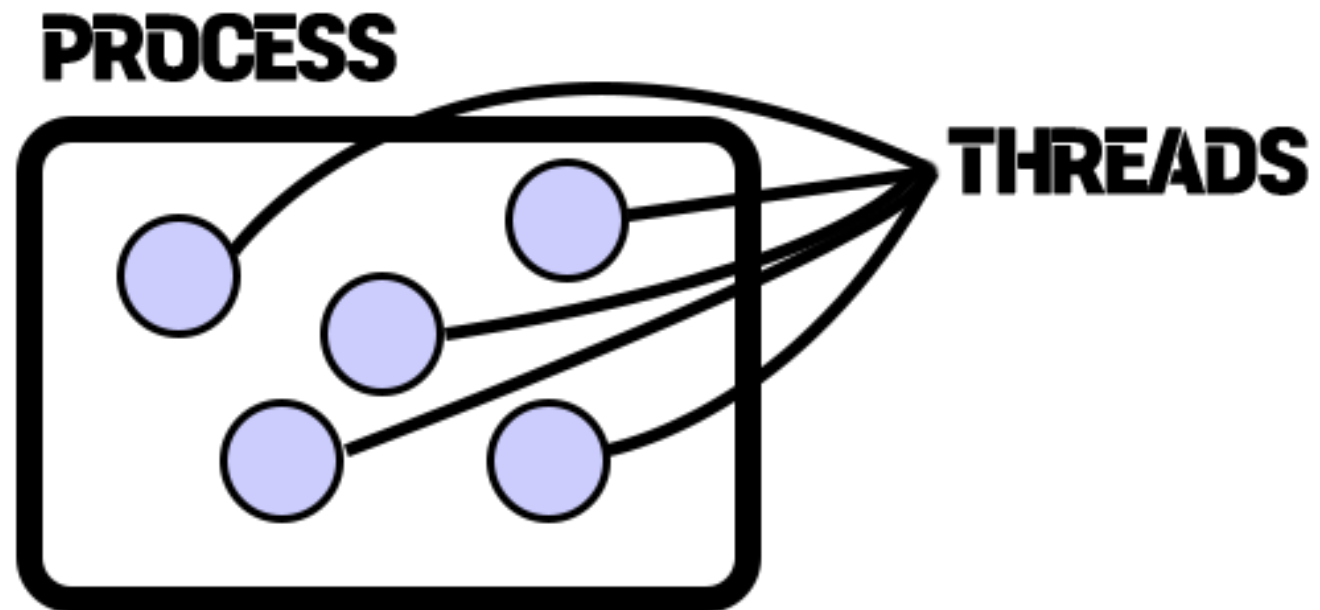
↓ 더블 클릭



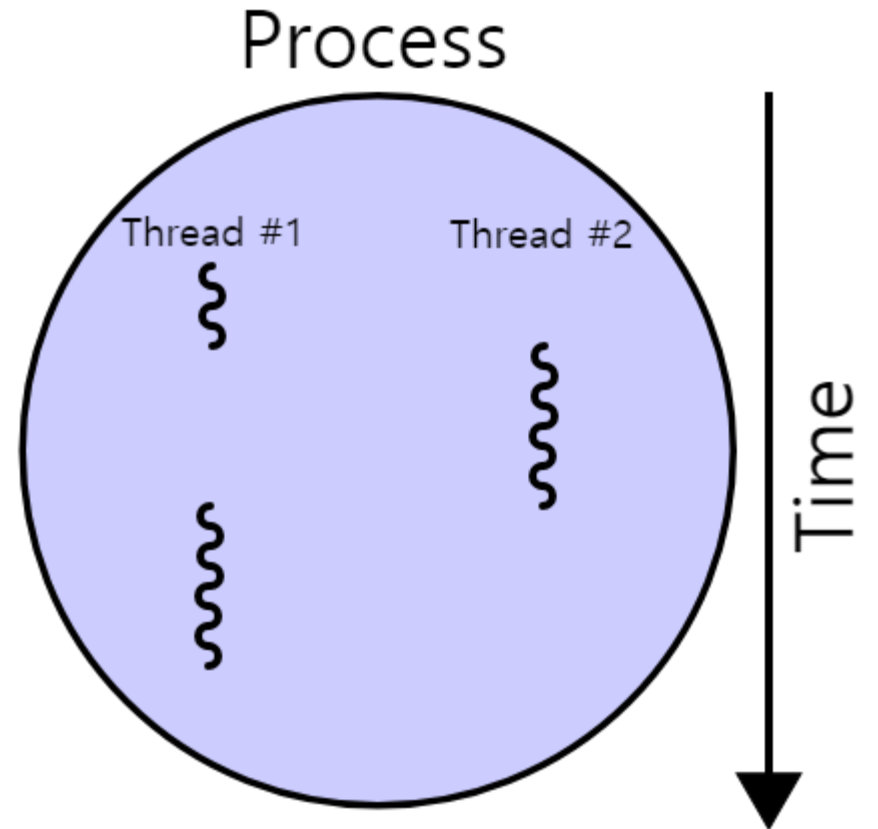
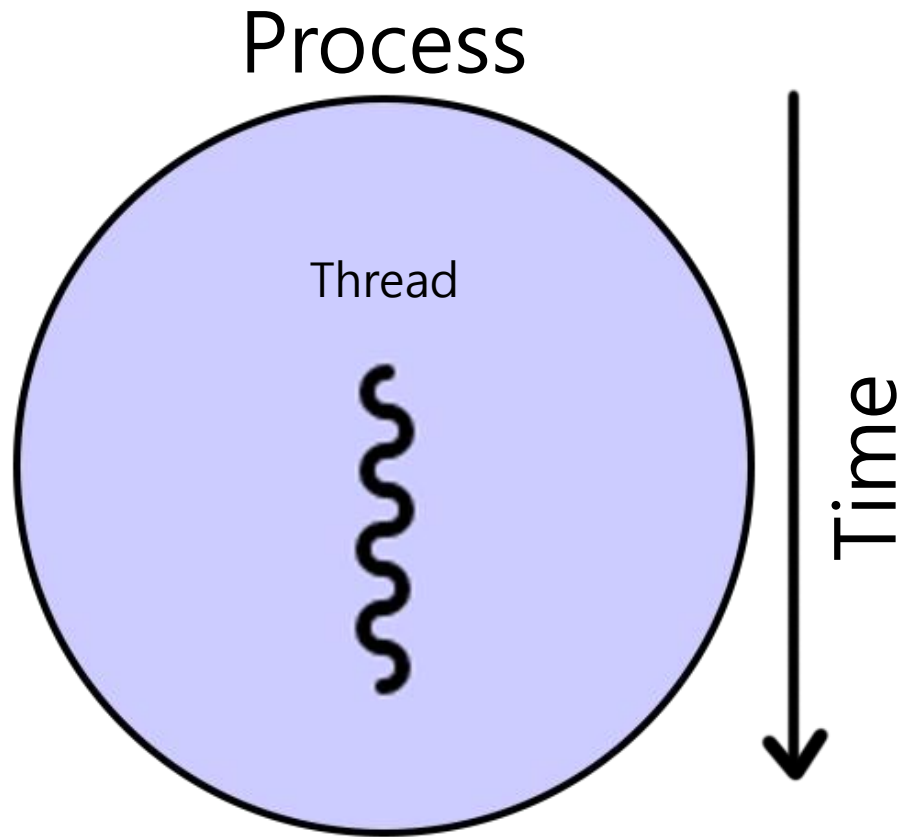
프로세스



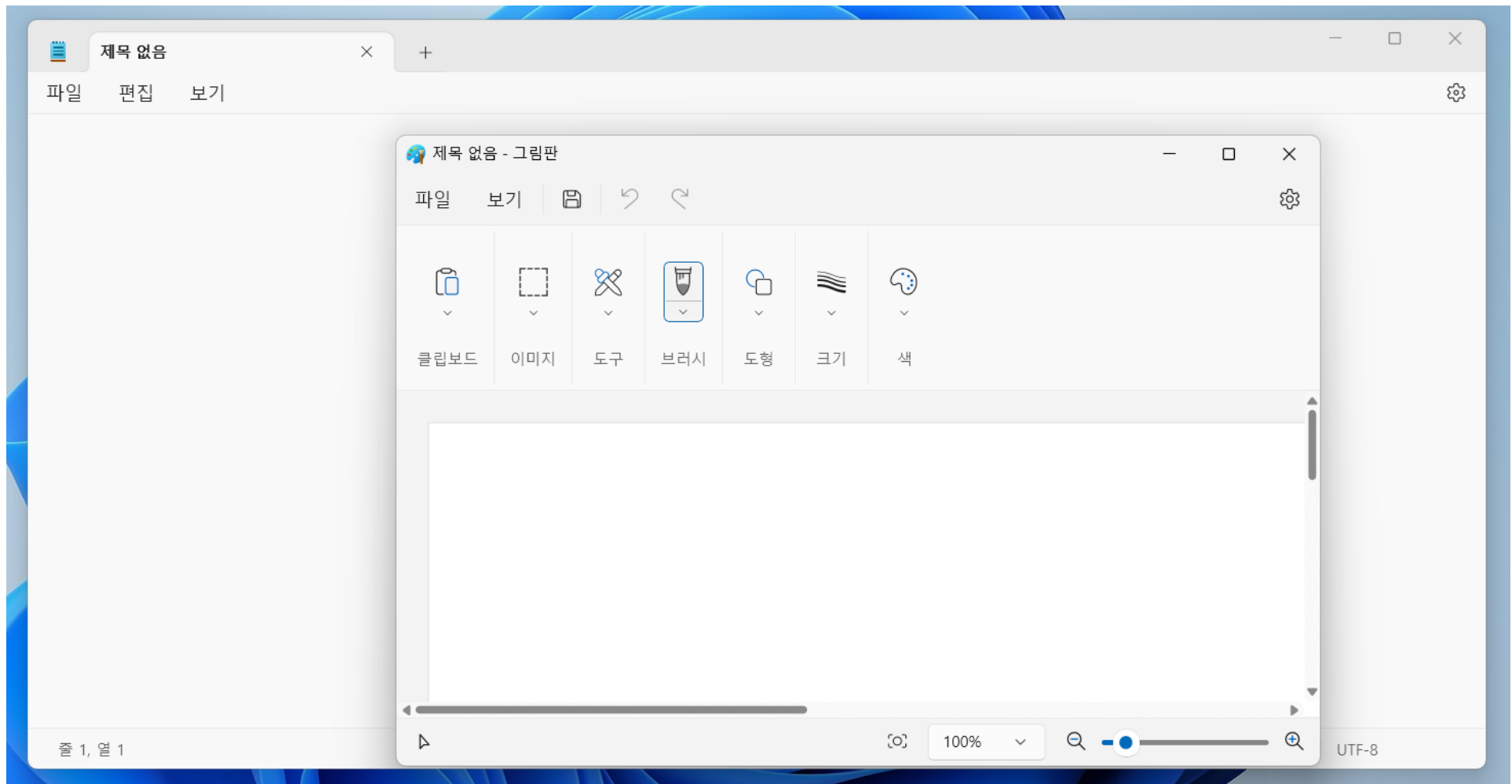
프로세스



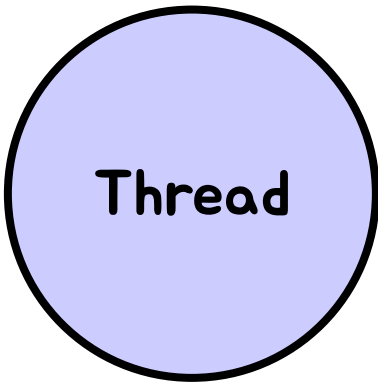
프로세스



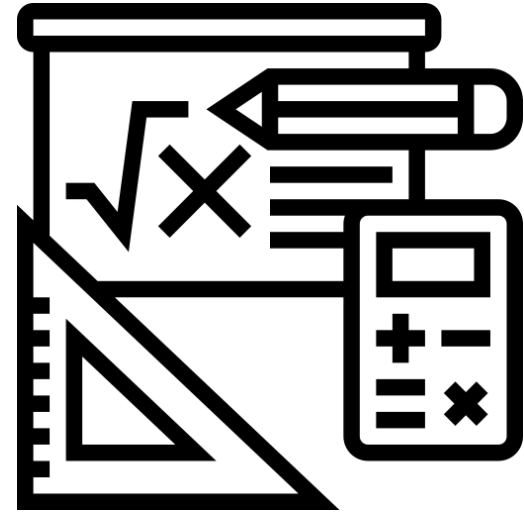
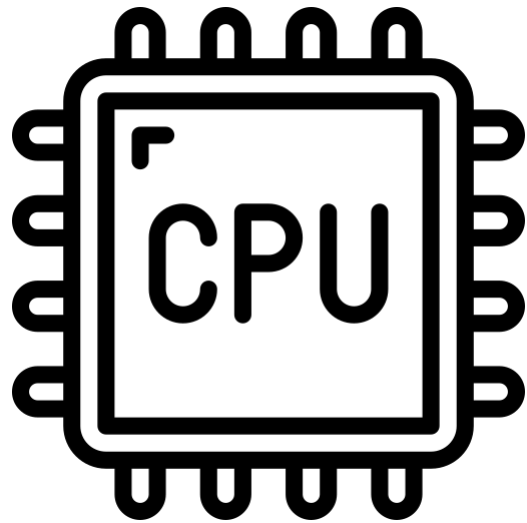
프로세스



프로세스



프로세스

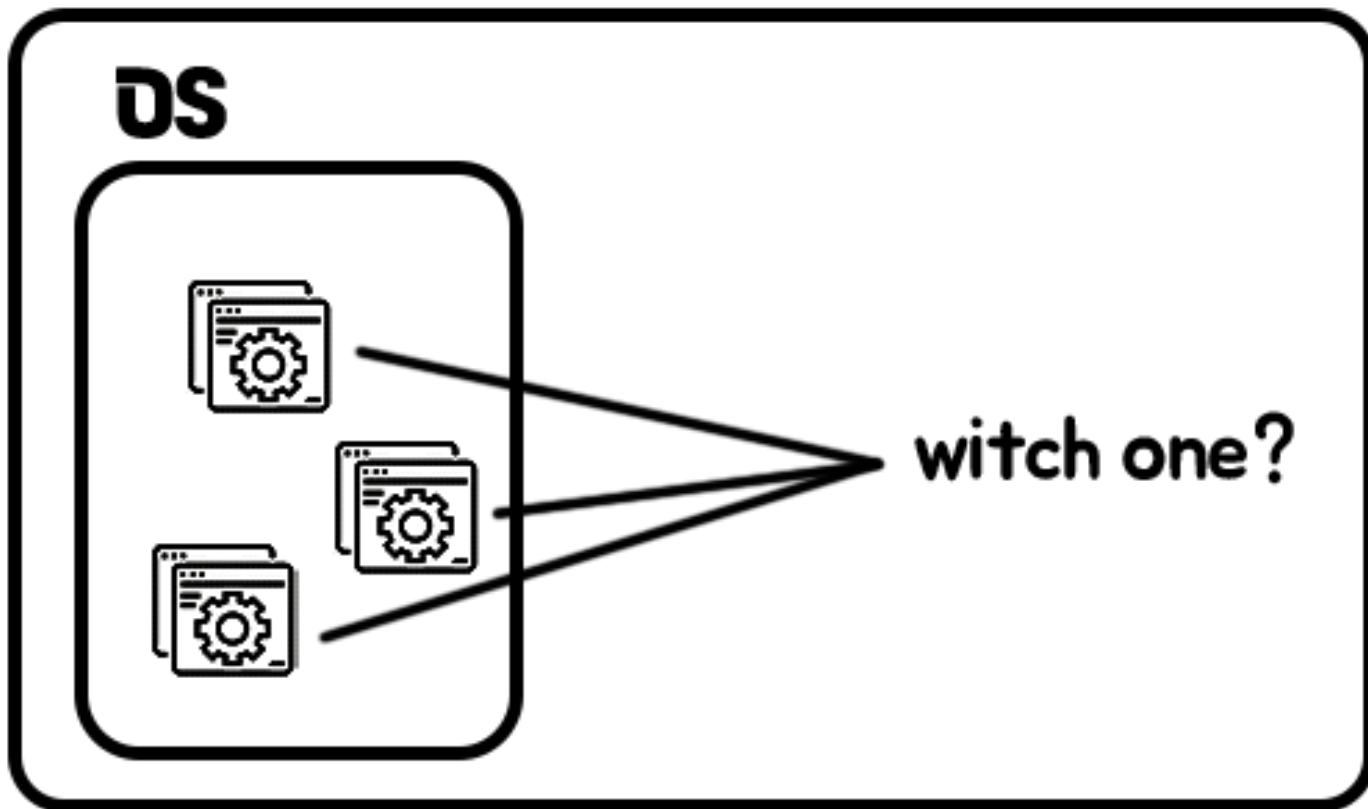


프로세스

Thread ?

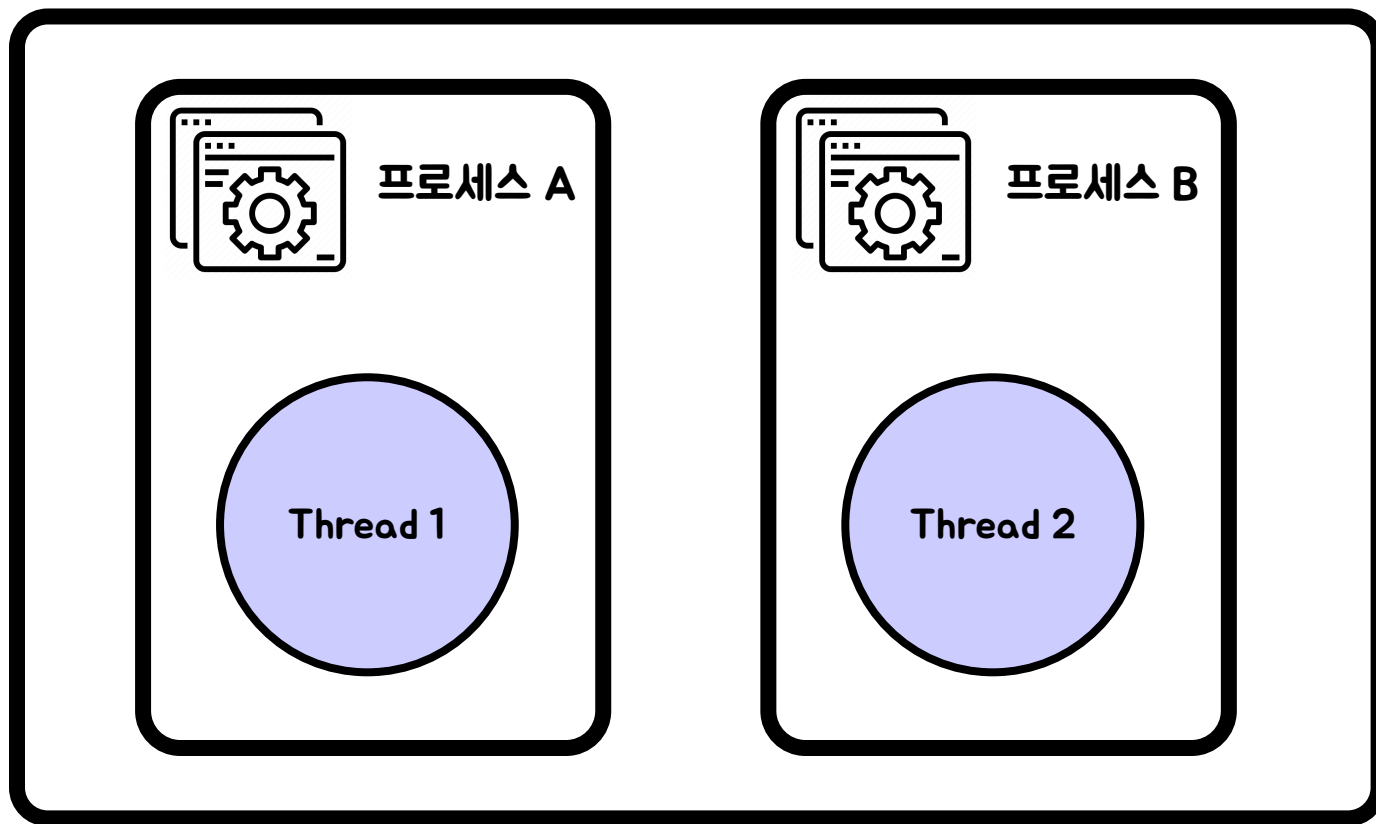
운영체제와 스케줄링

SCHEDULING



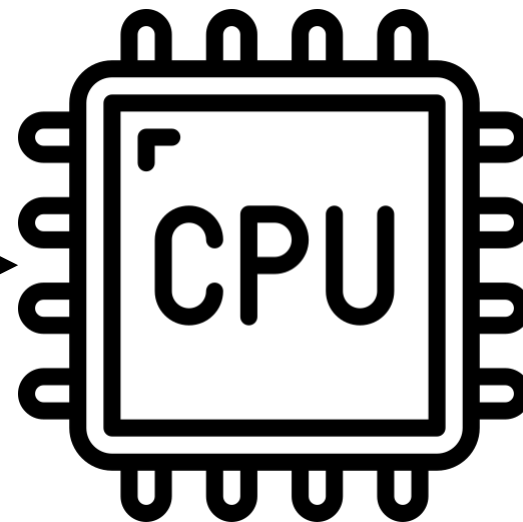
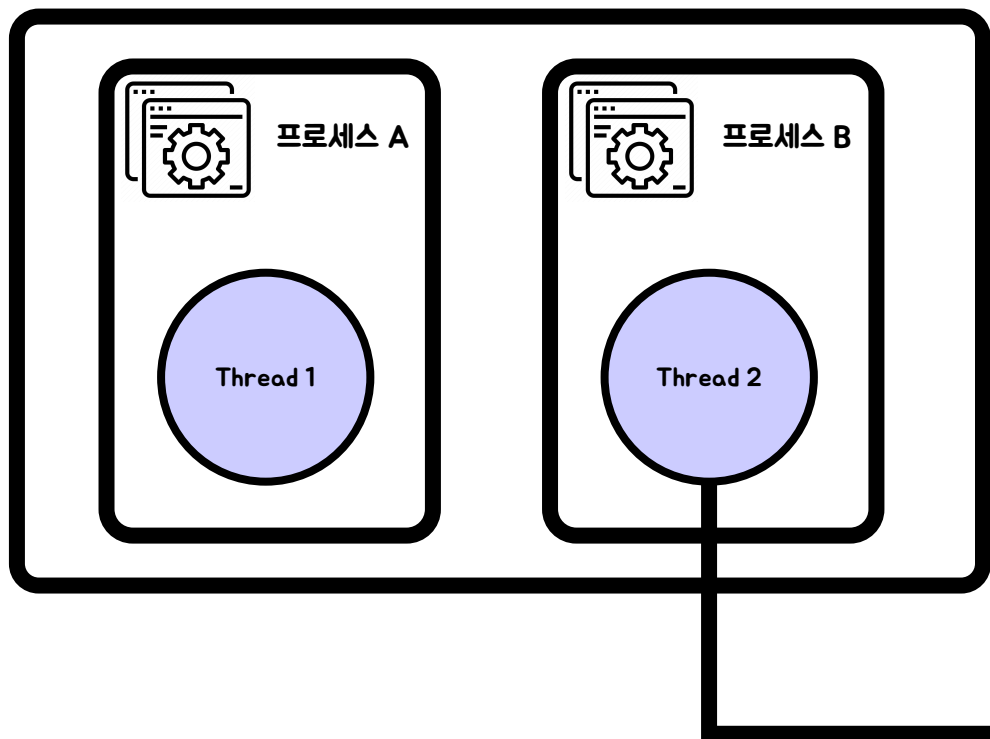
운영체제와 스케줄링

운영체제

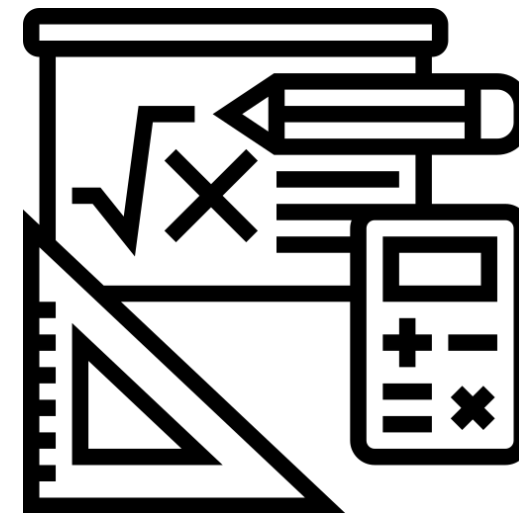
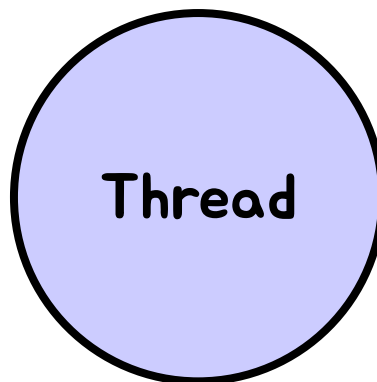
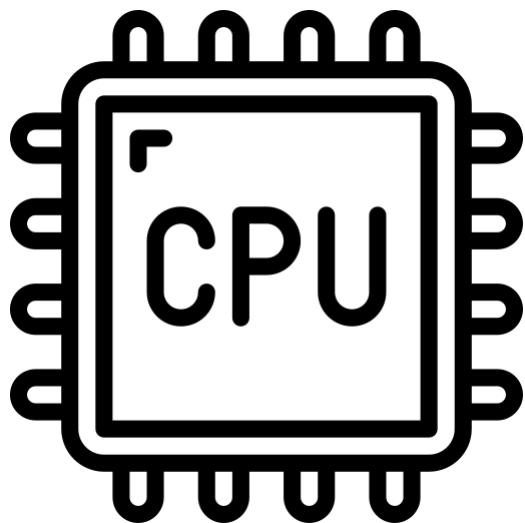


운영체제와 스케줄링

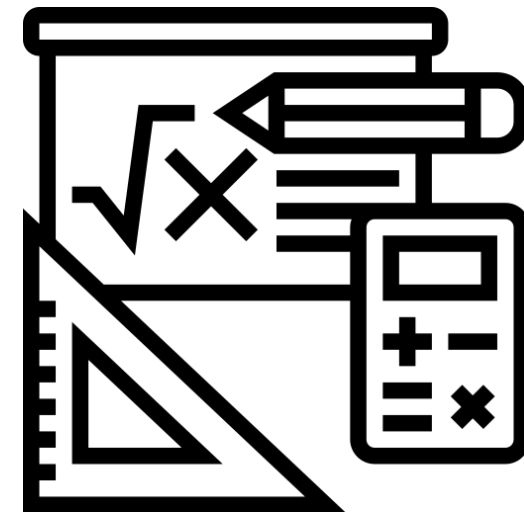
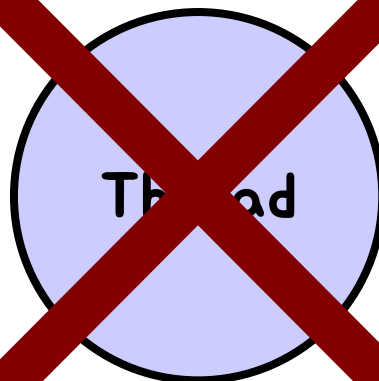
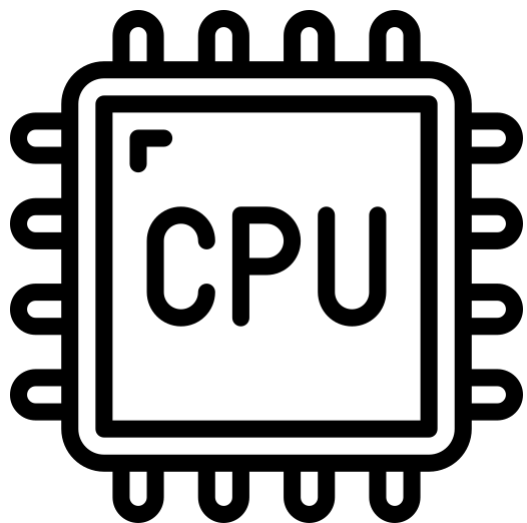
운영체제



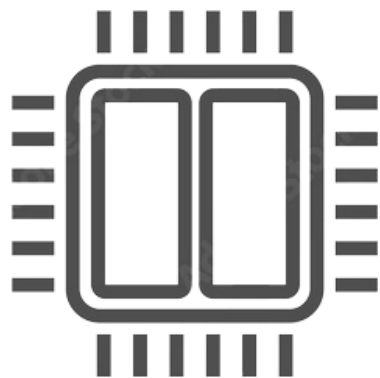
운영체제와 스케줄링



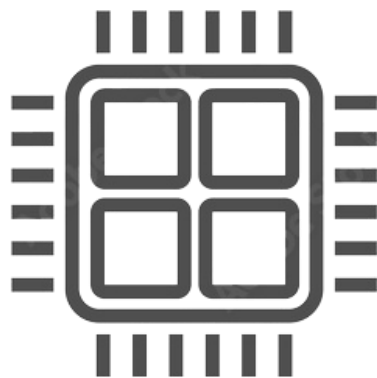
운영체제와 스케줄링



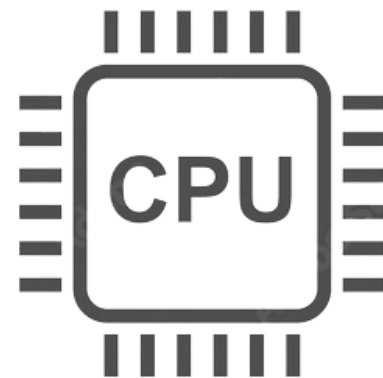
CPU와 CORE



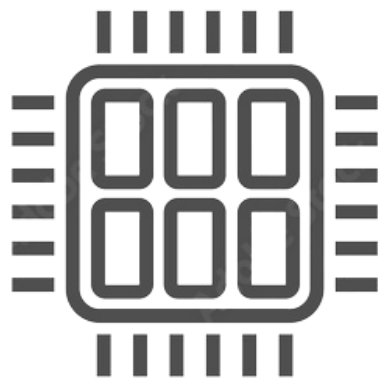
Dual-core CPU



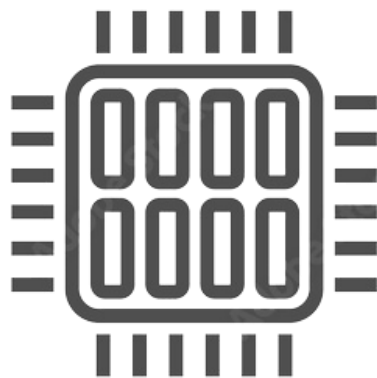
Quad-core CPU



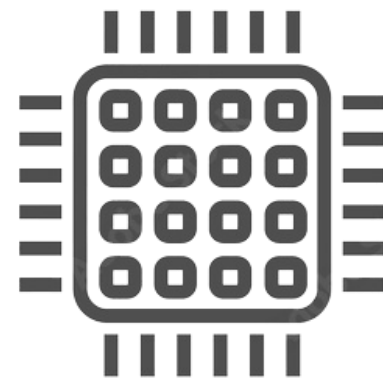
CPU chip



Six-core CPU

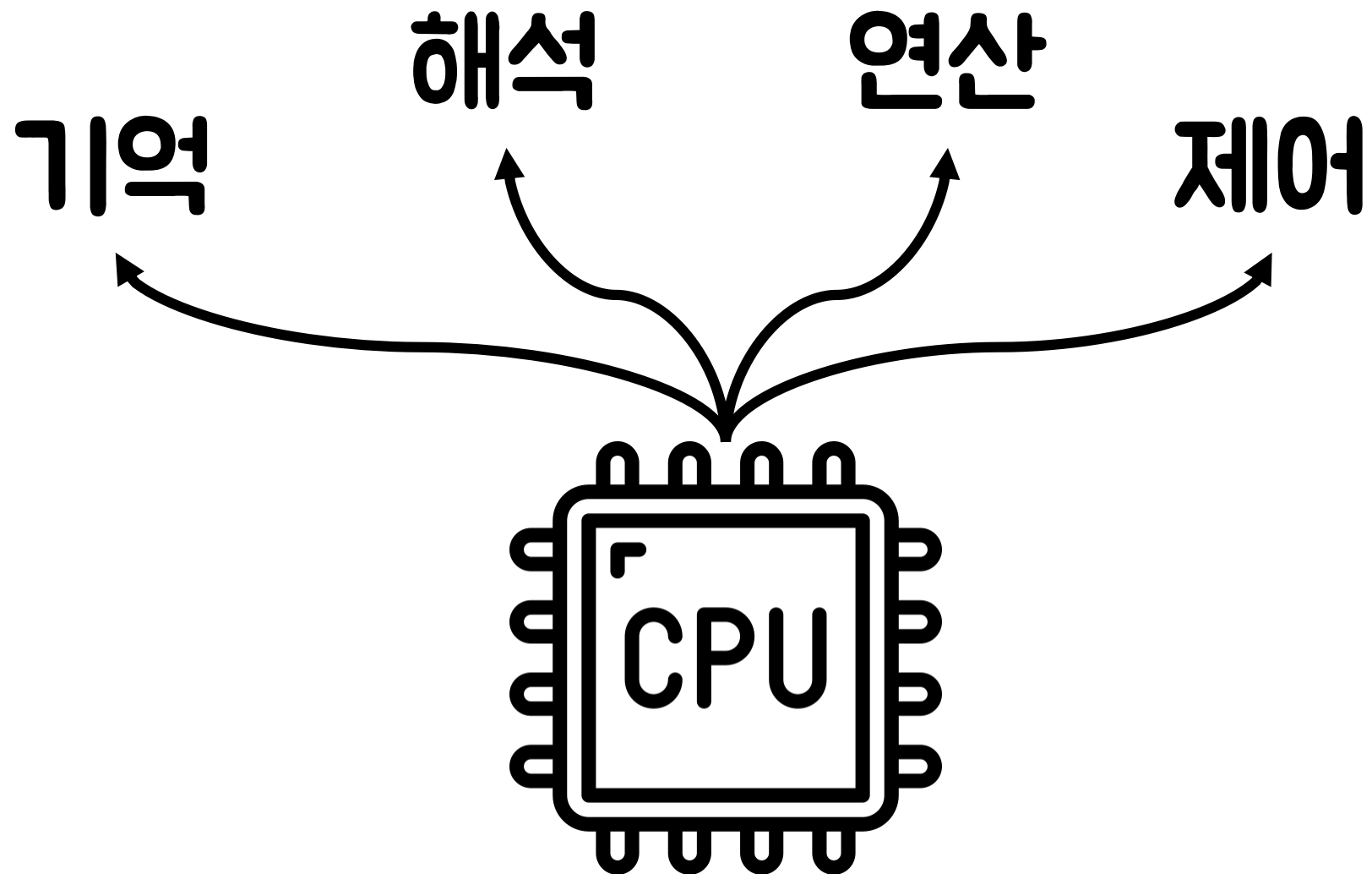


Octa-core CPU

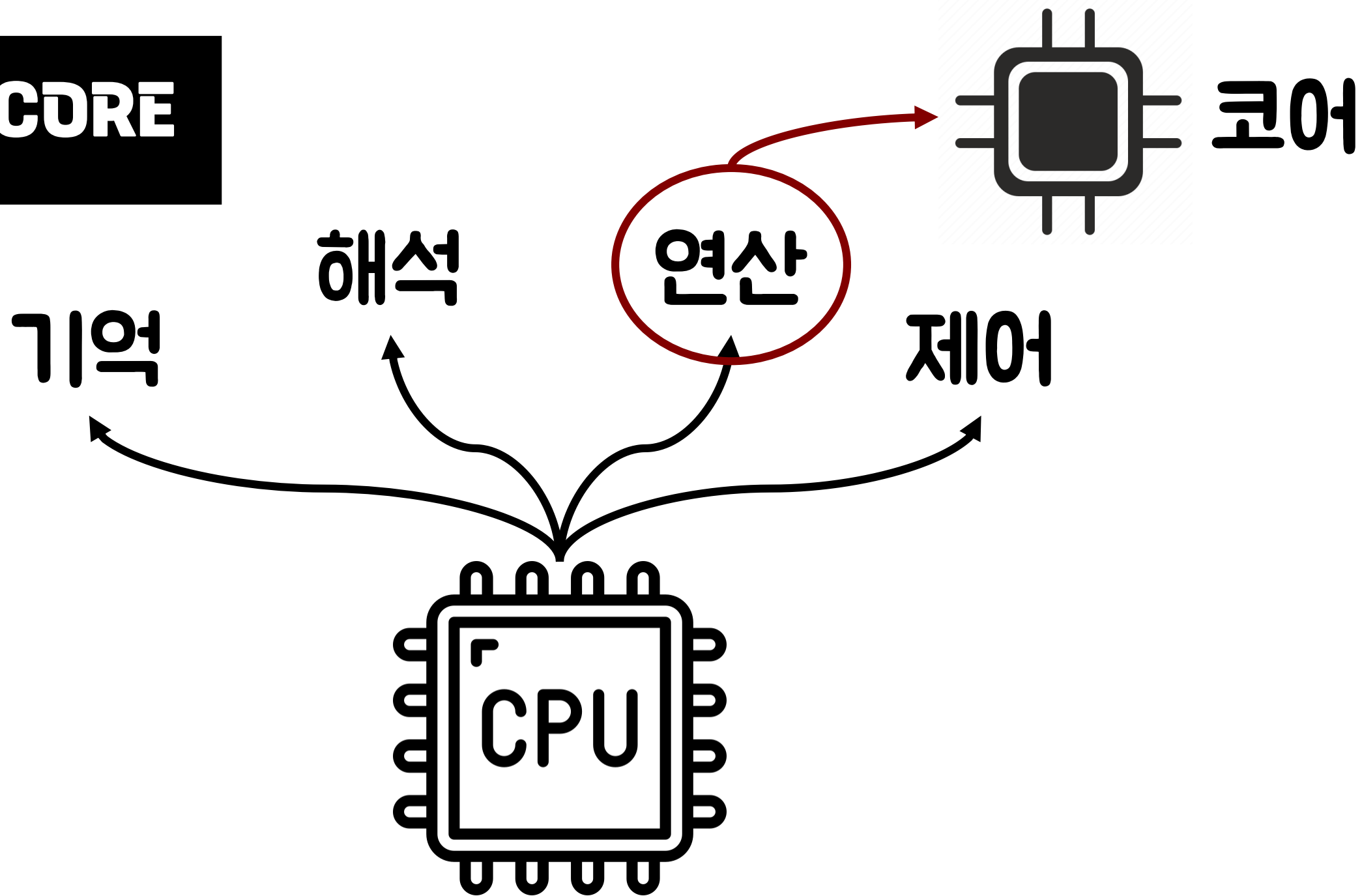


Multiple-core CPU

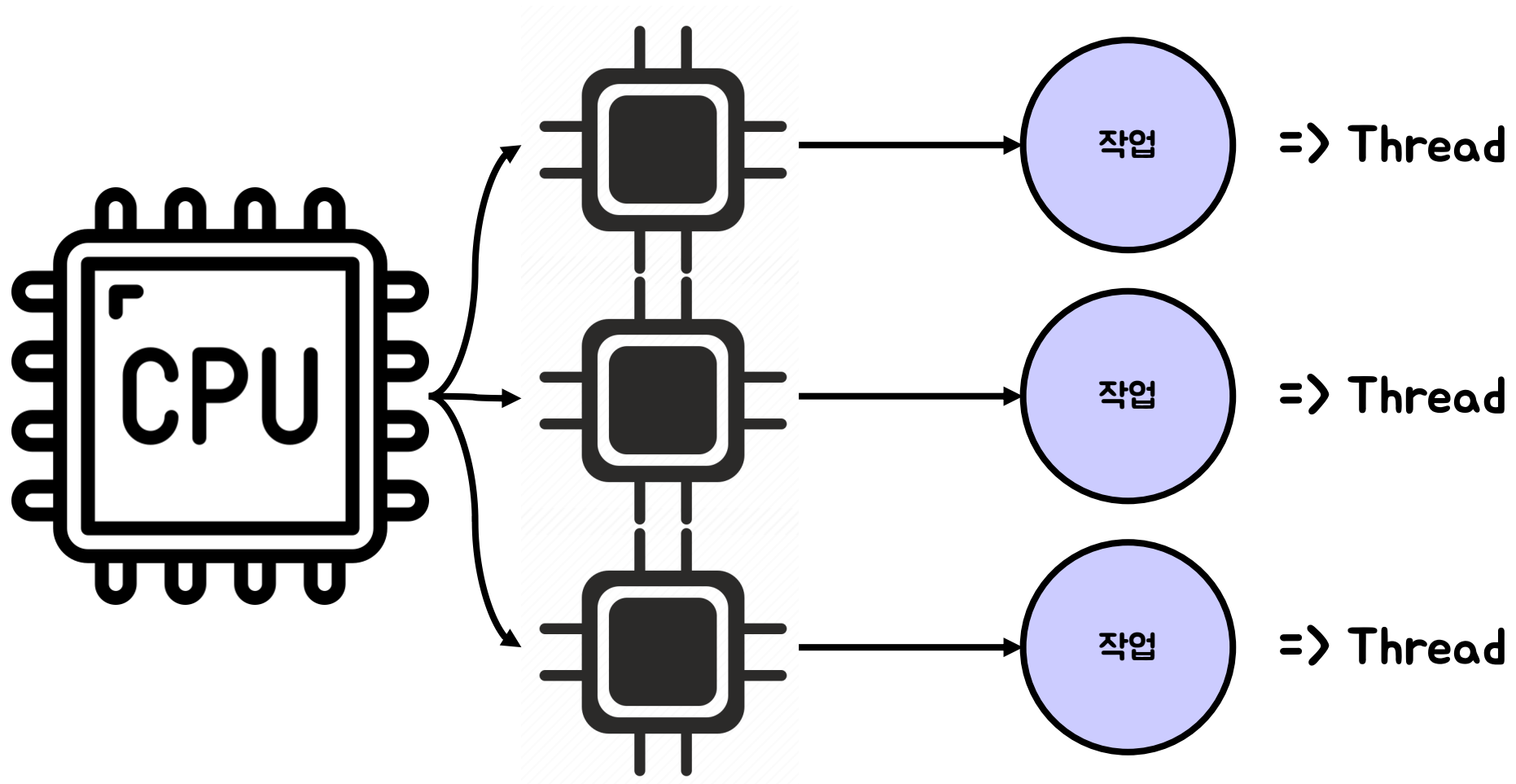
CPU와 CORE



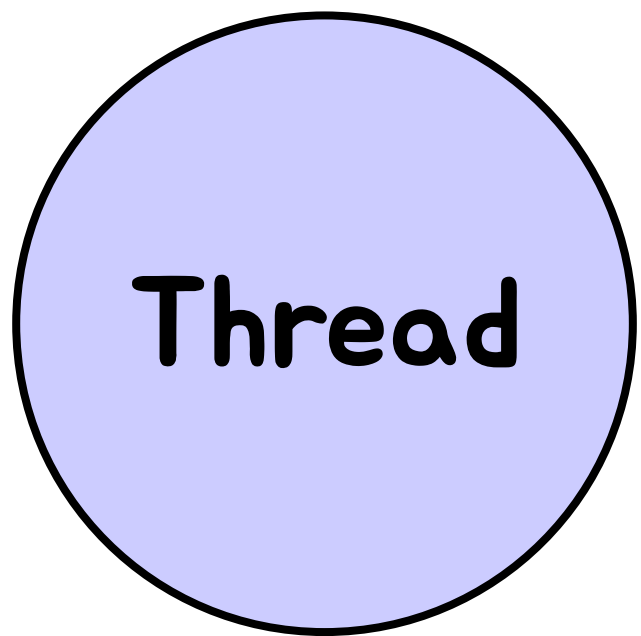
CPU와 CORE



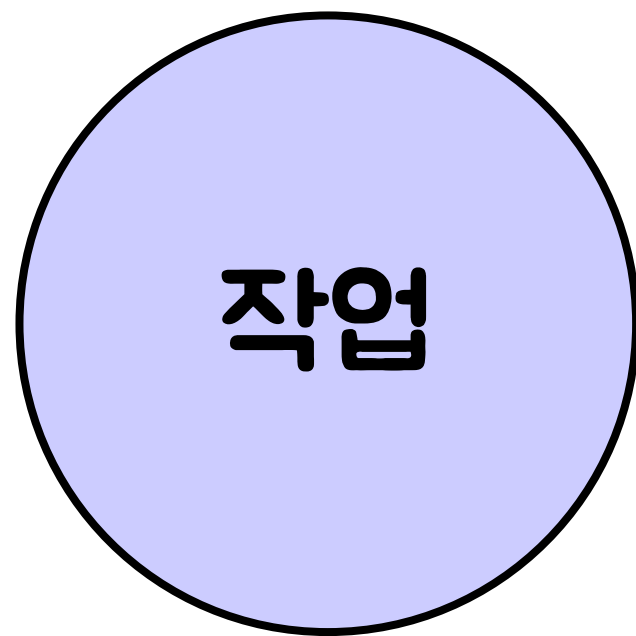
CPU와 CORE



요약



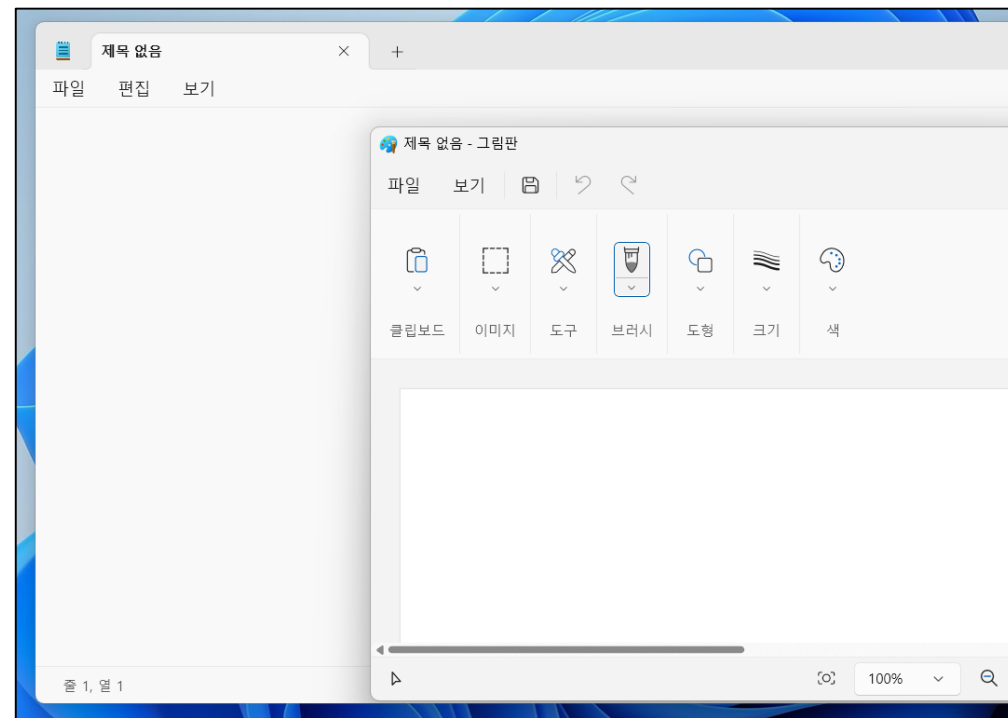
==



요약

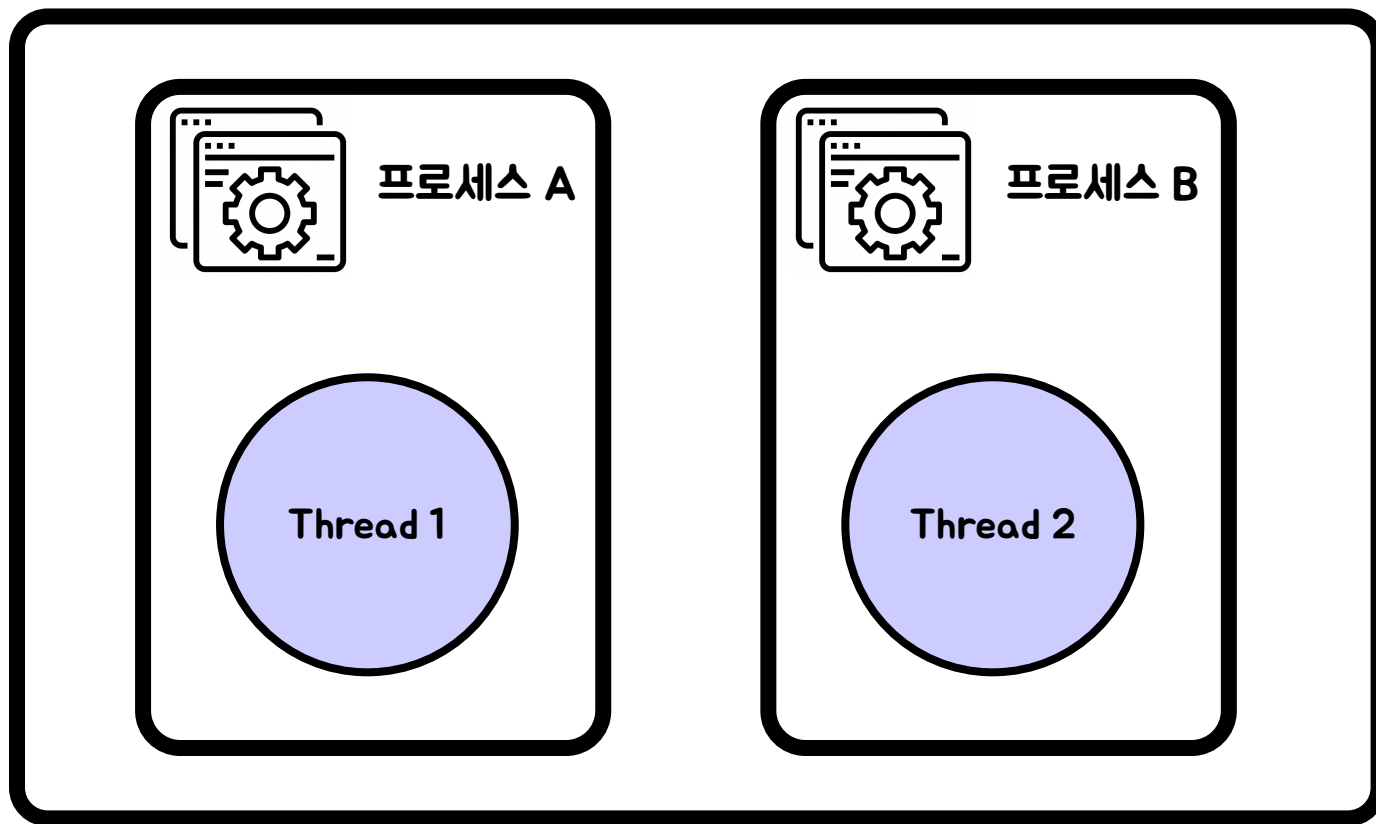
작업

끝!



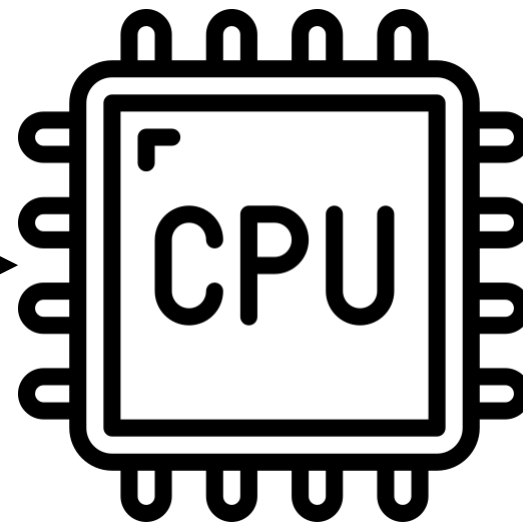
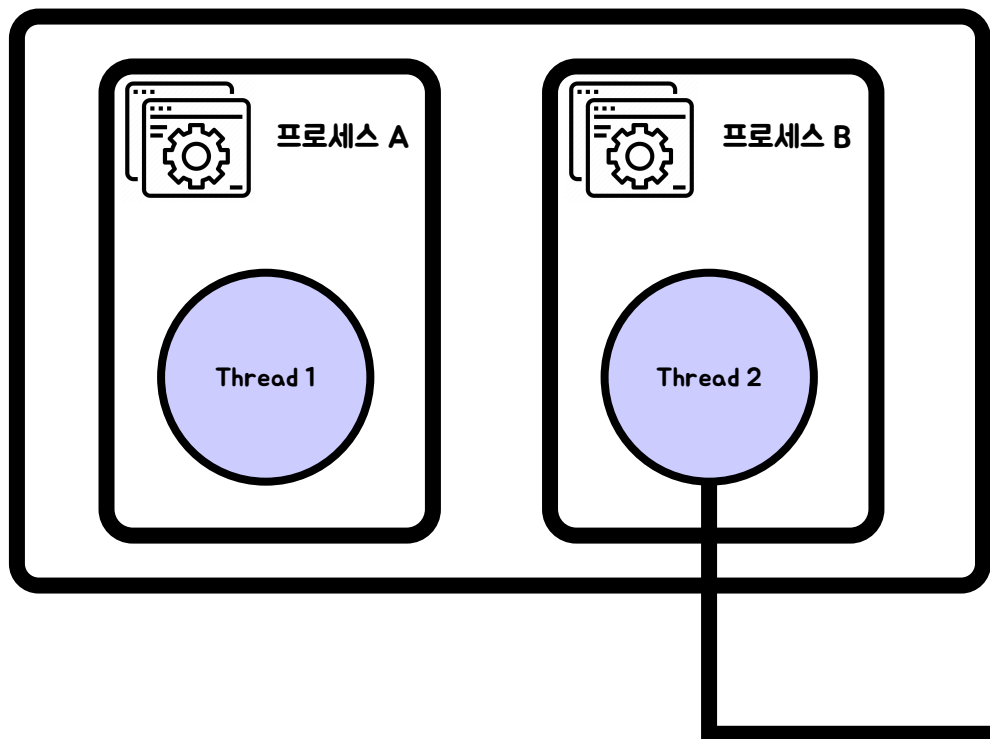
요약

운영체제

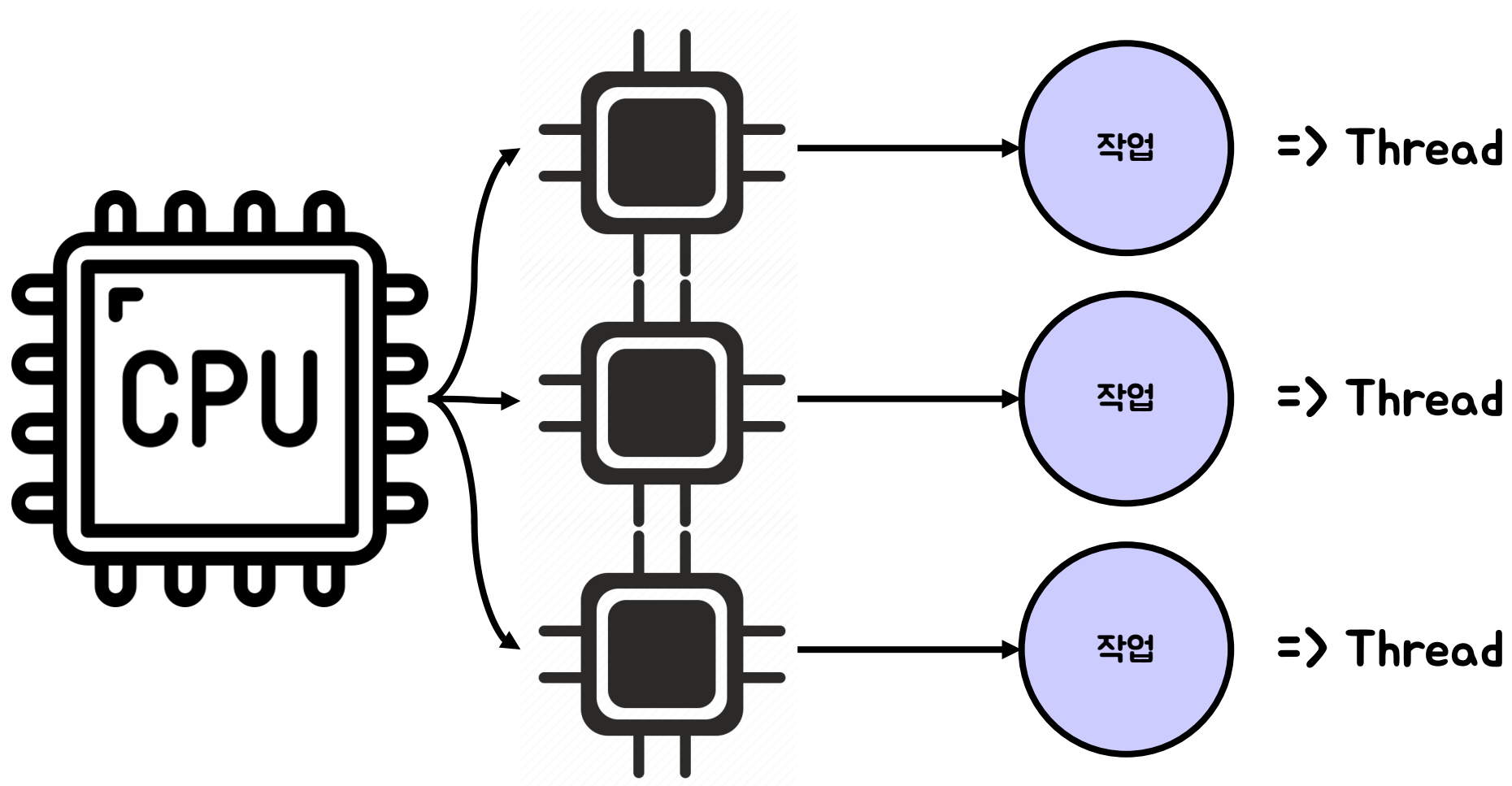


요약

운영체제



요약

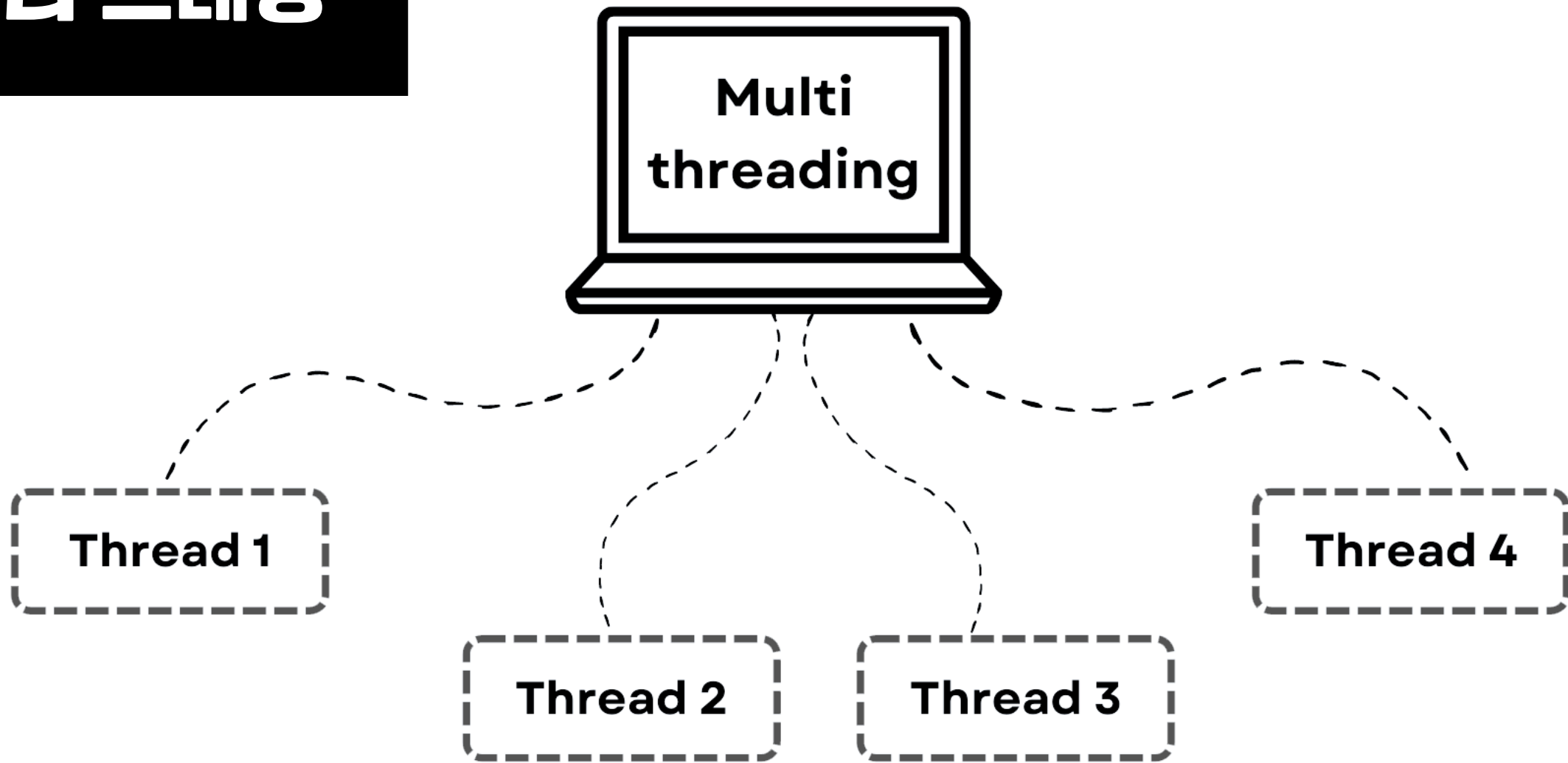


멀티 스테킹

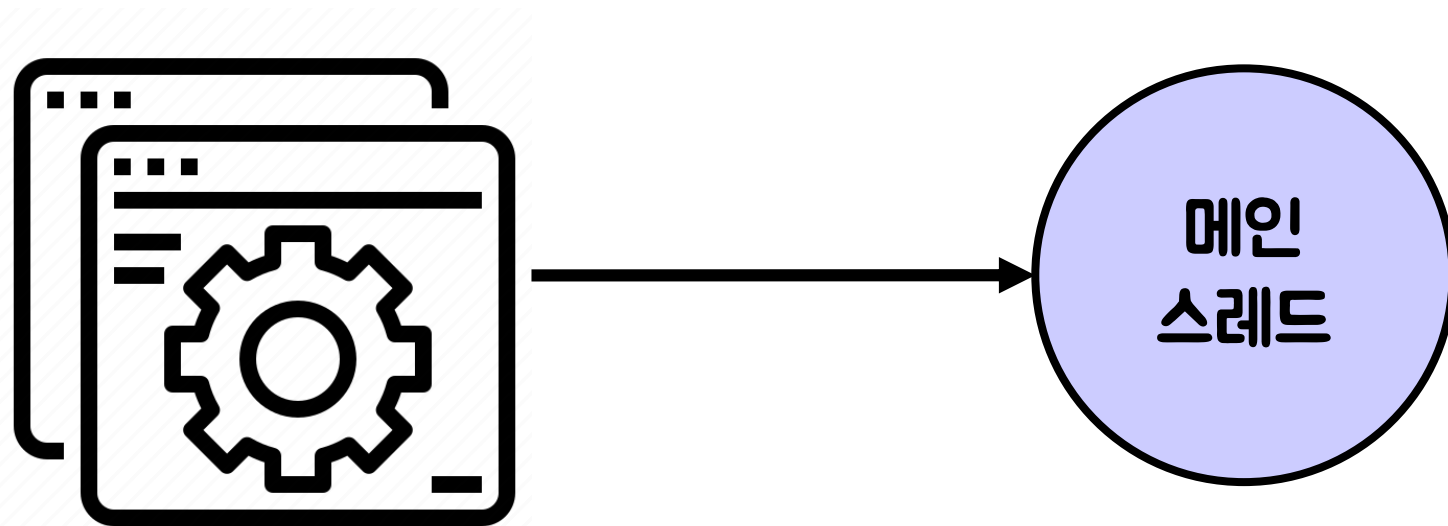
멀티 스테징

멀티 스테징?

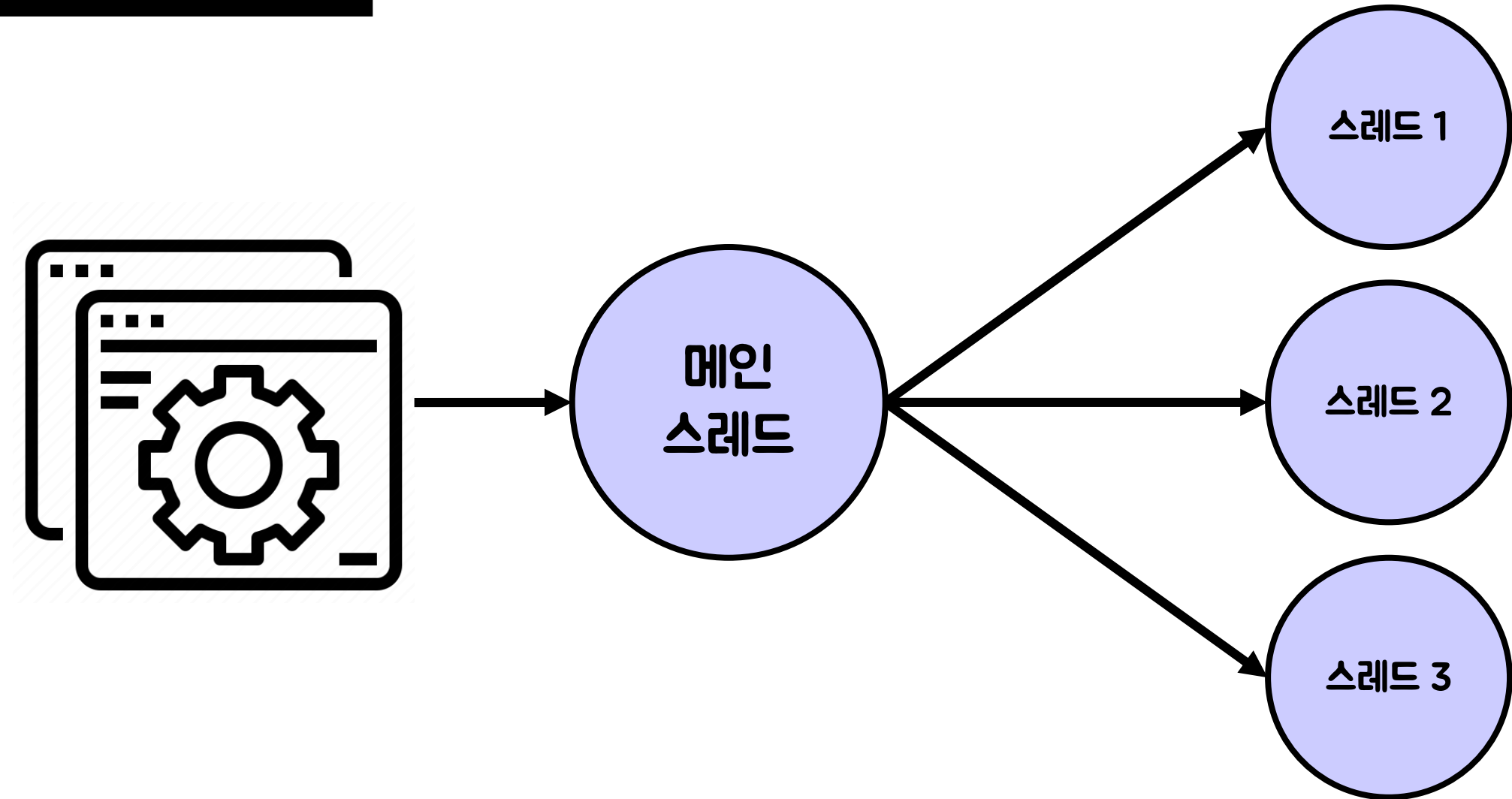
멀티 스레딩



멀티 스레딩



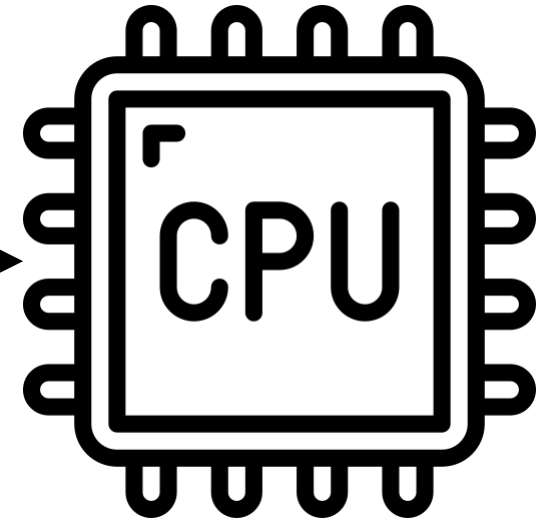
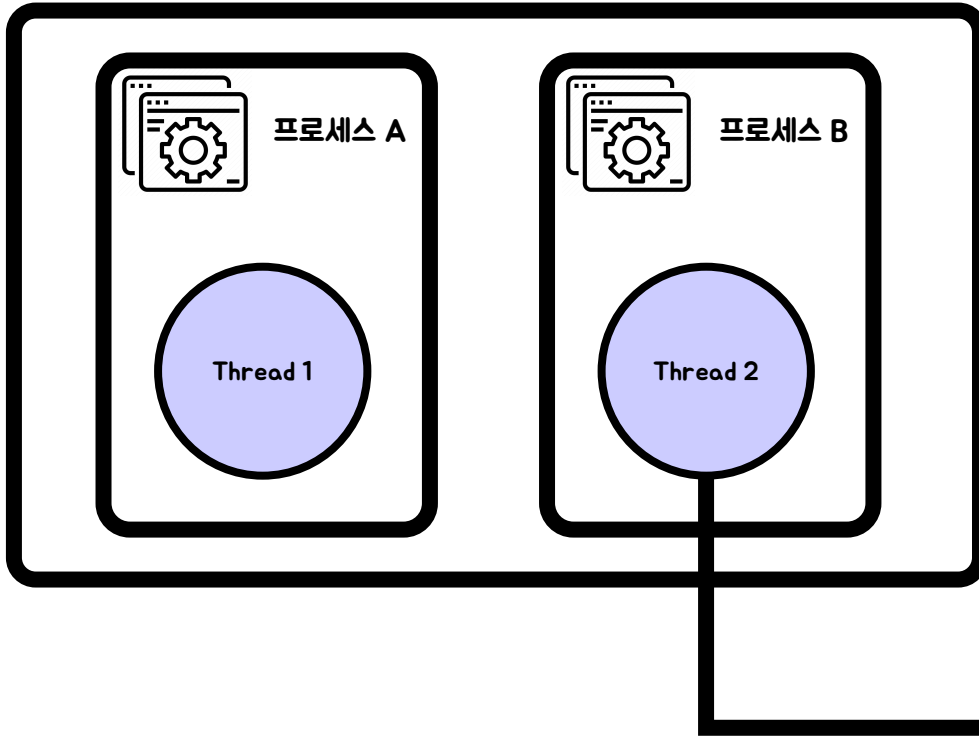
멀티 스레딩



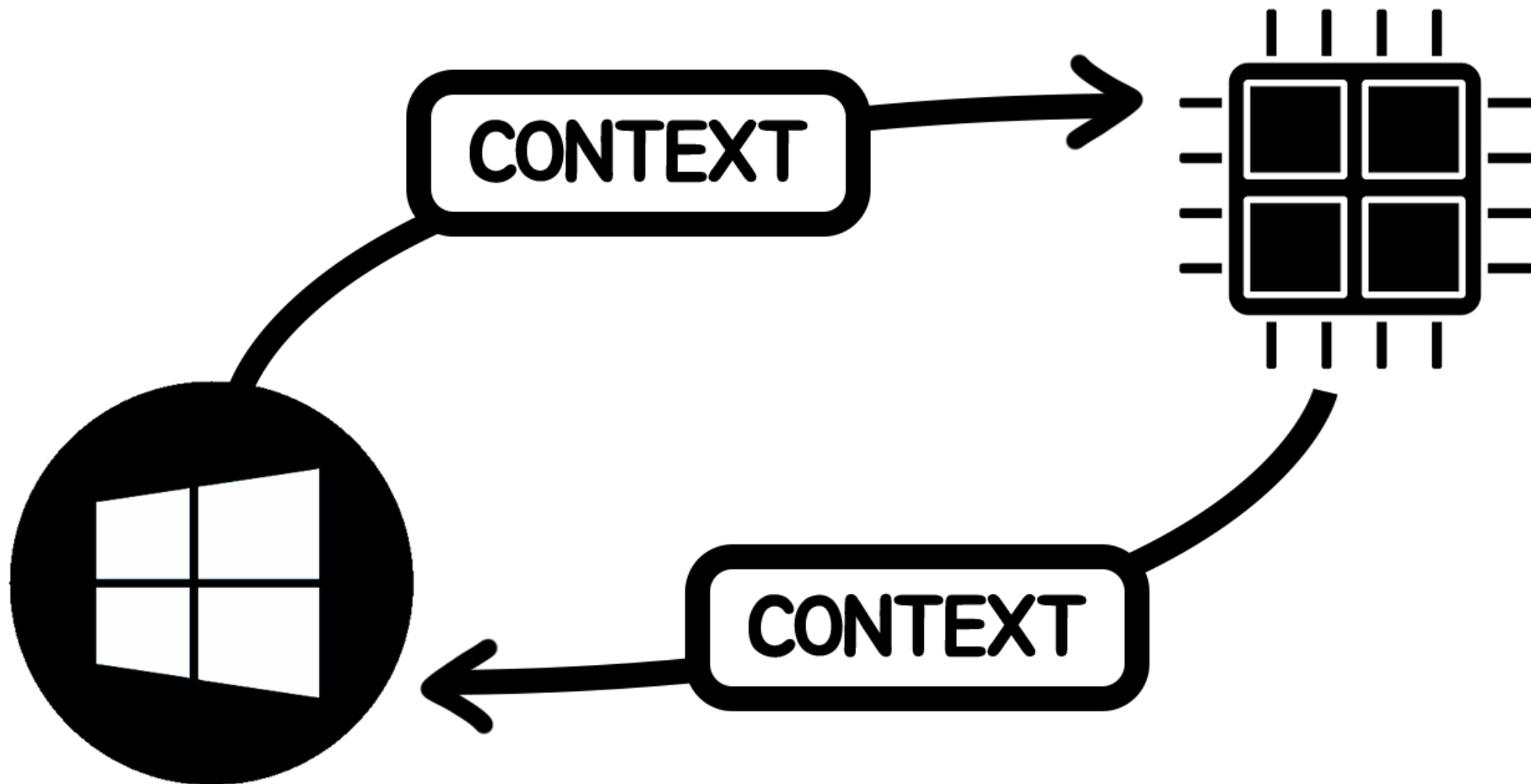
컨텍스트 스위칭 (Context Switching)

컨텍스트 스위칭 (Context Switching)

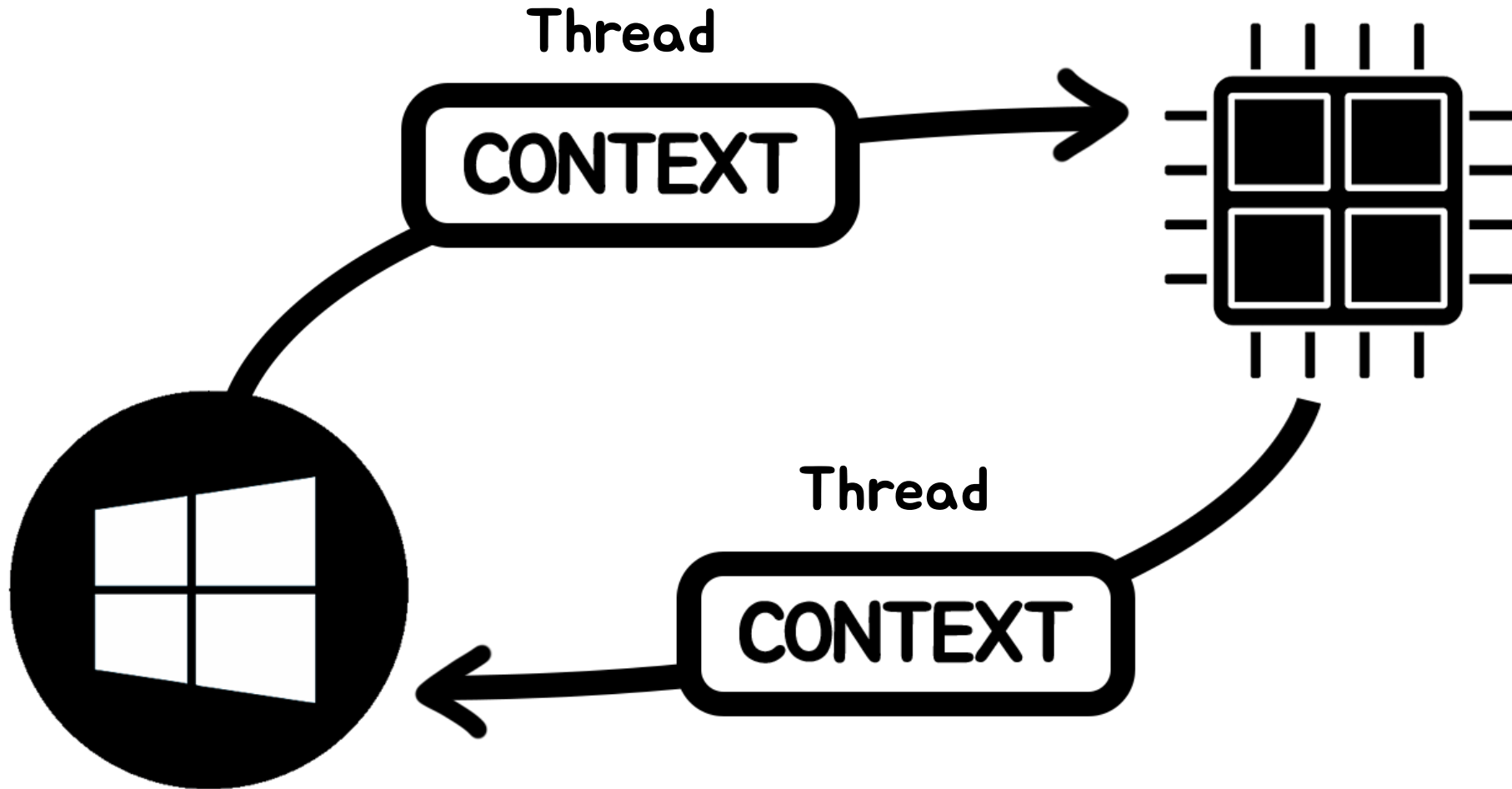
운영체제



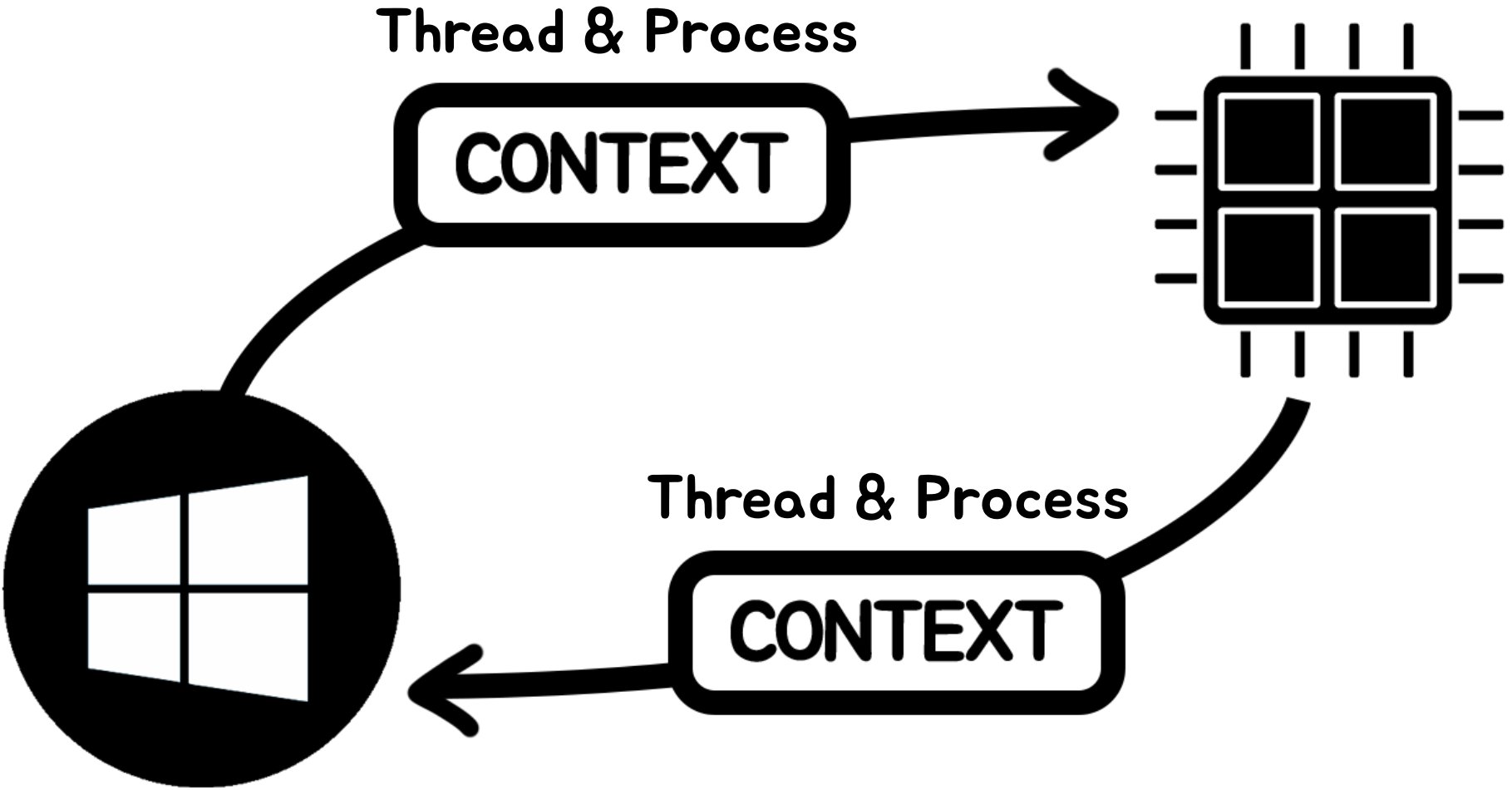
컨텍스트 스위칭 (Context Switching)



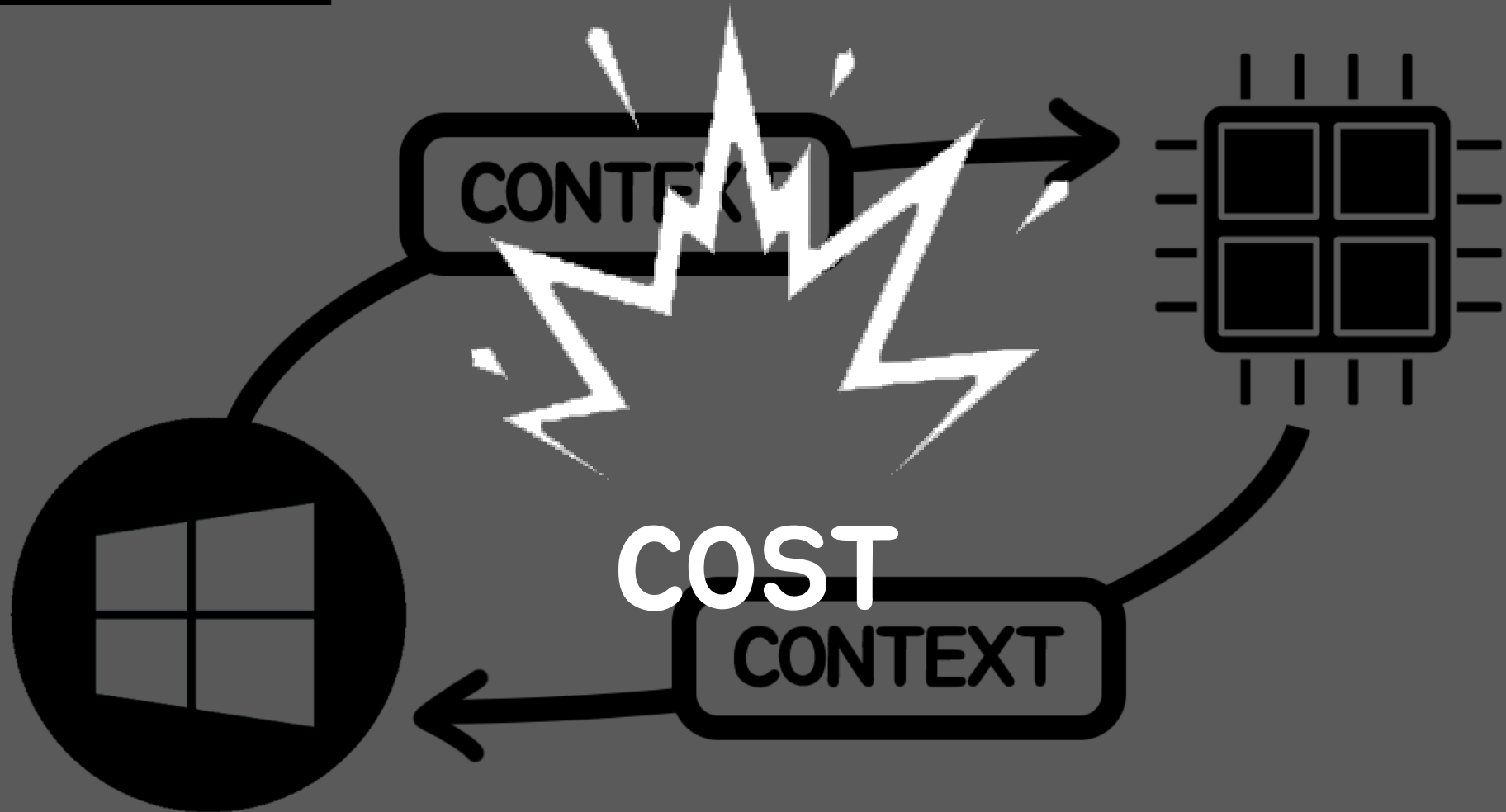
**스레드 컨텍스트 스위칭
(Thread Context Switching)**



**프로세스 컨텍스트 스위칭
(Process Context Switching)**

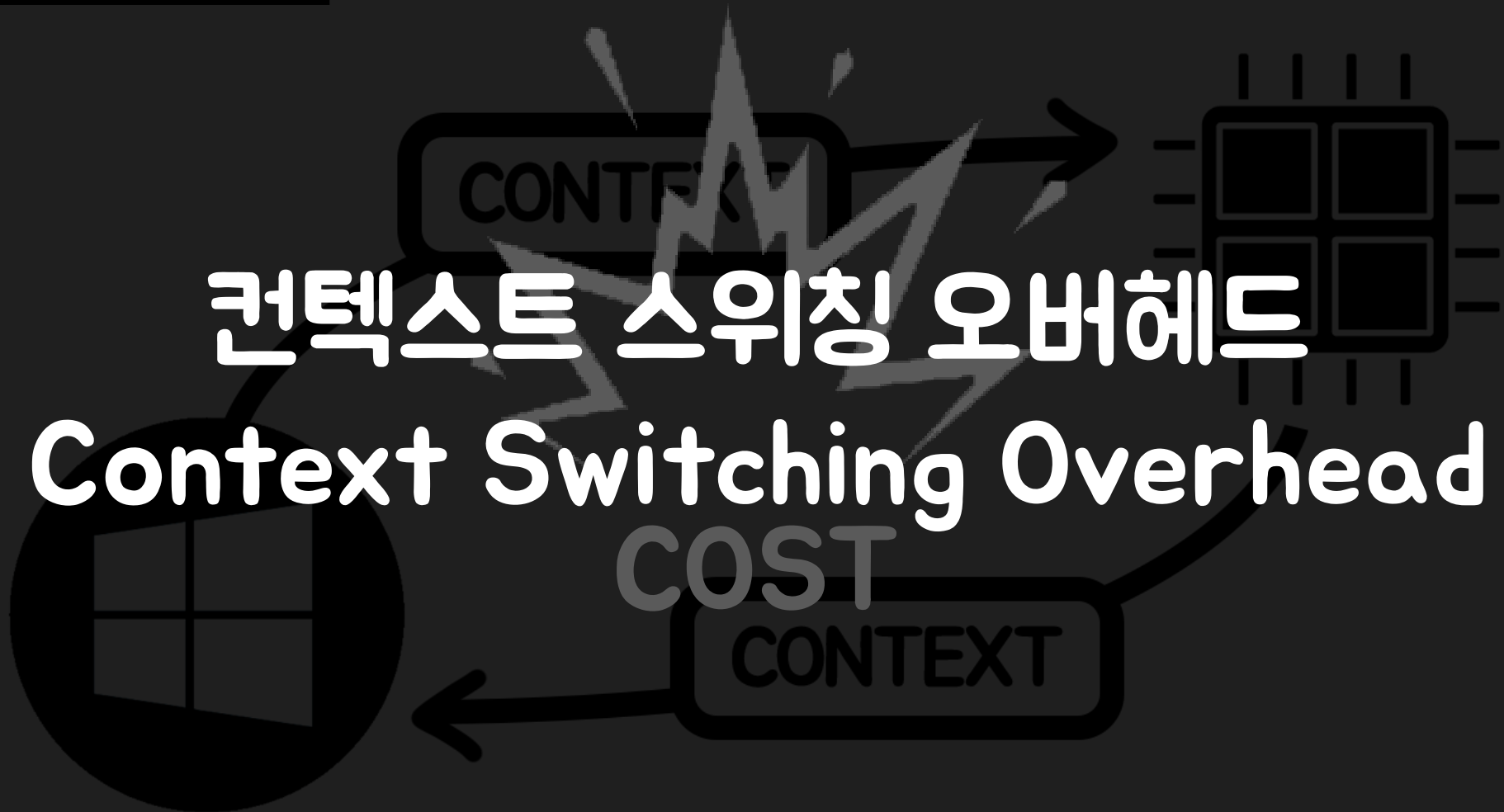


**컨텍스트 스위칭 오버헤드
(Context Switching Overhead)**



컨텍스트 스위칭 오버헤드
(Context Switching Overhead)

컨텍스트 스위칭 오버헤드 (Context Switching Overhead)



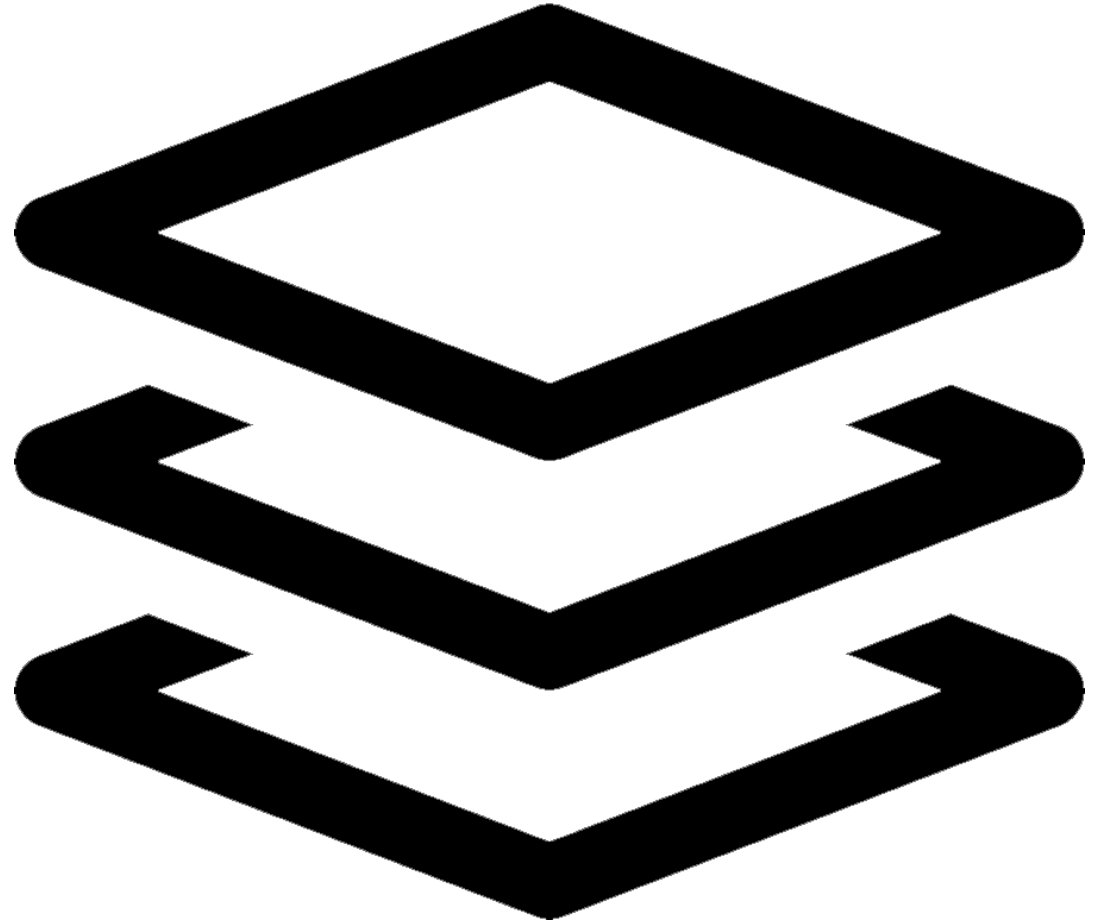
PCB

(Process Control Block)

**프로세스 제어 블록
(Process Control Block)**

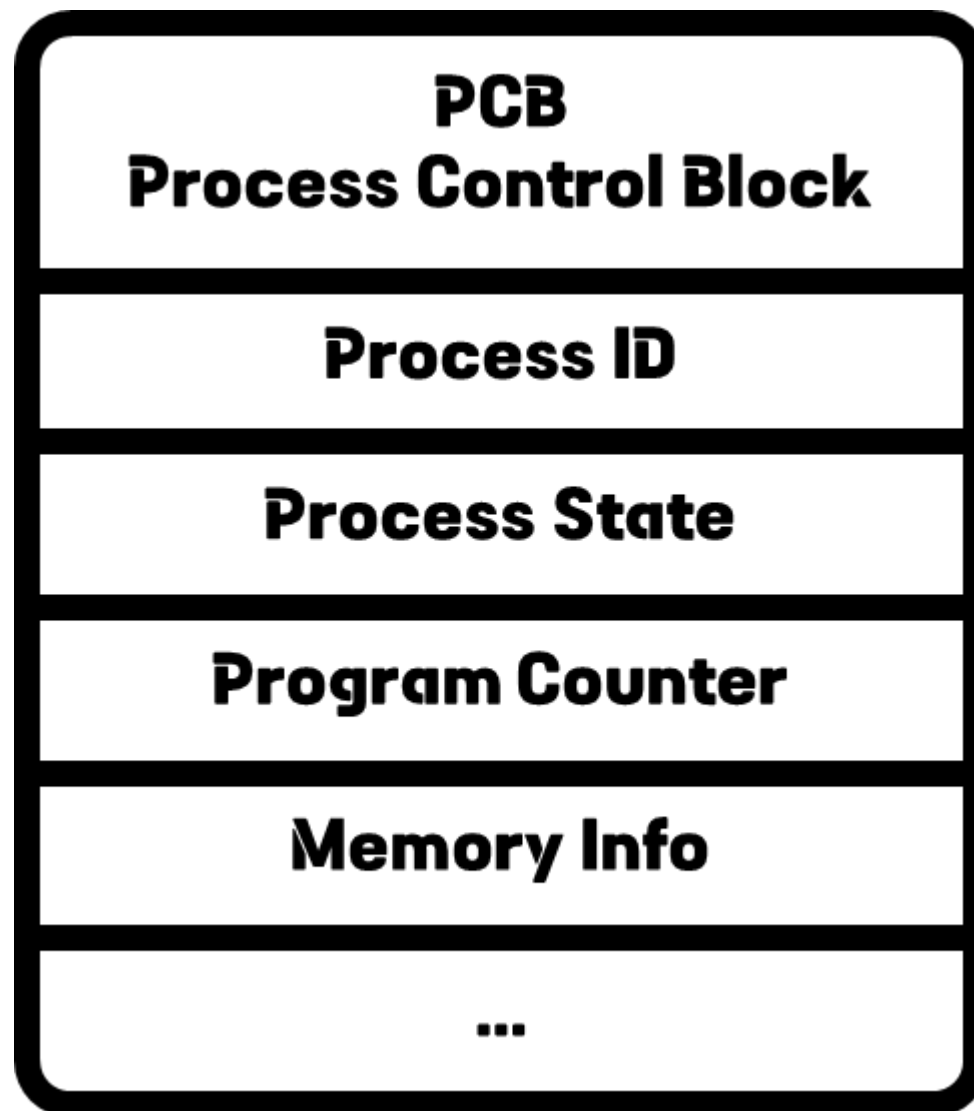
데이터 구조체

Process Data



프로세스 제어 블록
(Process Control Block)

총 12 가지



프로세스 제어 블록
(Process Control Block)

**이정도만
알고있자**

- 1. Process ID (PID)**
 - 프로세스 ID
 - 프로세스를 식별하기 위한 고유 번호
- 2. Process State**
 - 프로세스 상태
 - 신규, 준비, 실행중, 대기중, 종료 와 같은 상태 정보
- 3. Program Counter**
 - 프로그램 계수기
 - 프로세스에 대해 실행될 다음 명령의 주소
- 4. CPU Register**
 - 실행 상태, 실행 정도 등을 저장하는 레지스터
- 5. Memory Management Information**
 - 메모리 관리 정보
 - 할당된 자원에 대한 정보

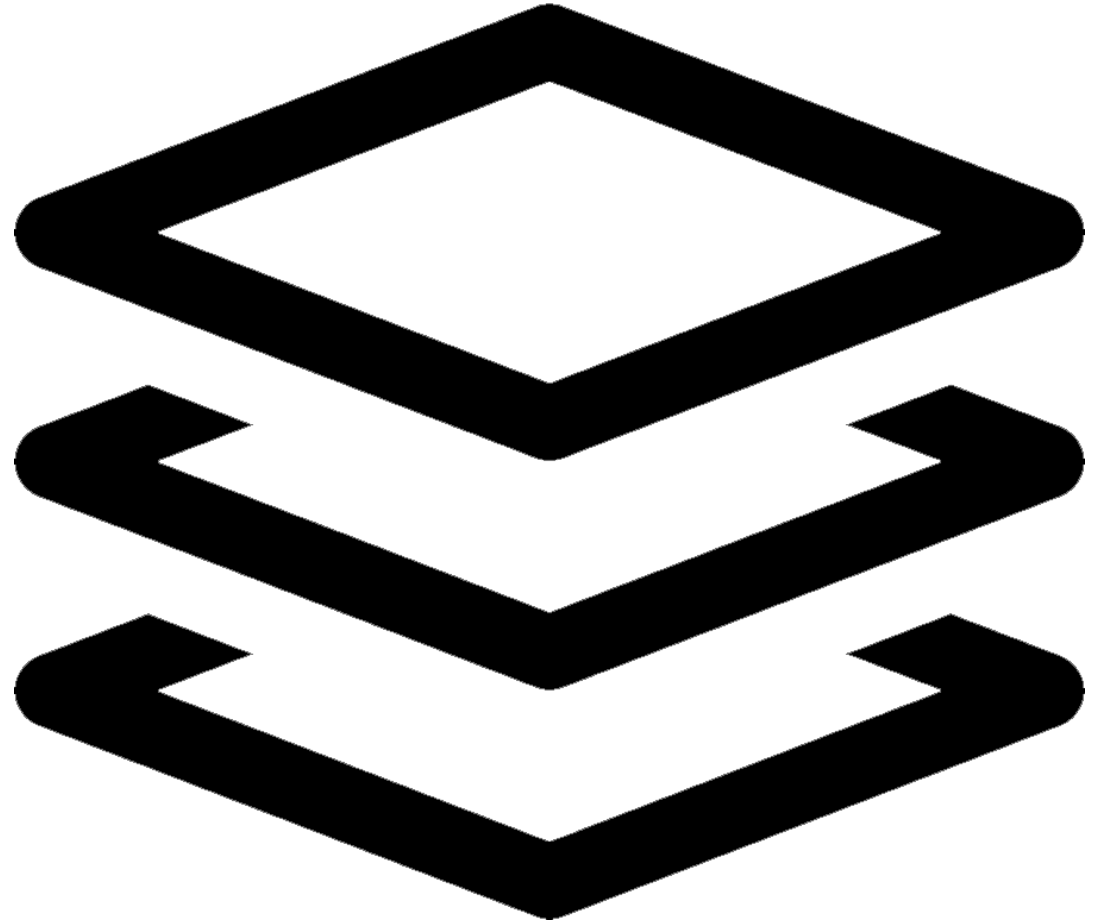
TCB

(Thread Control Block)

스레드 제어 블록
(Thread Control Block)

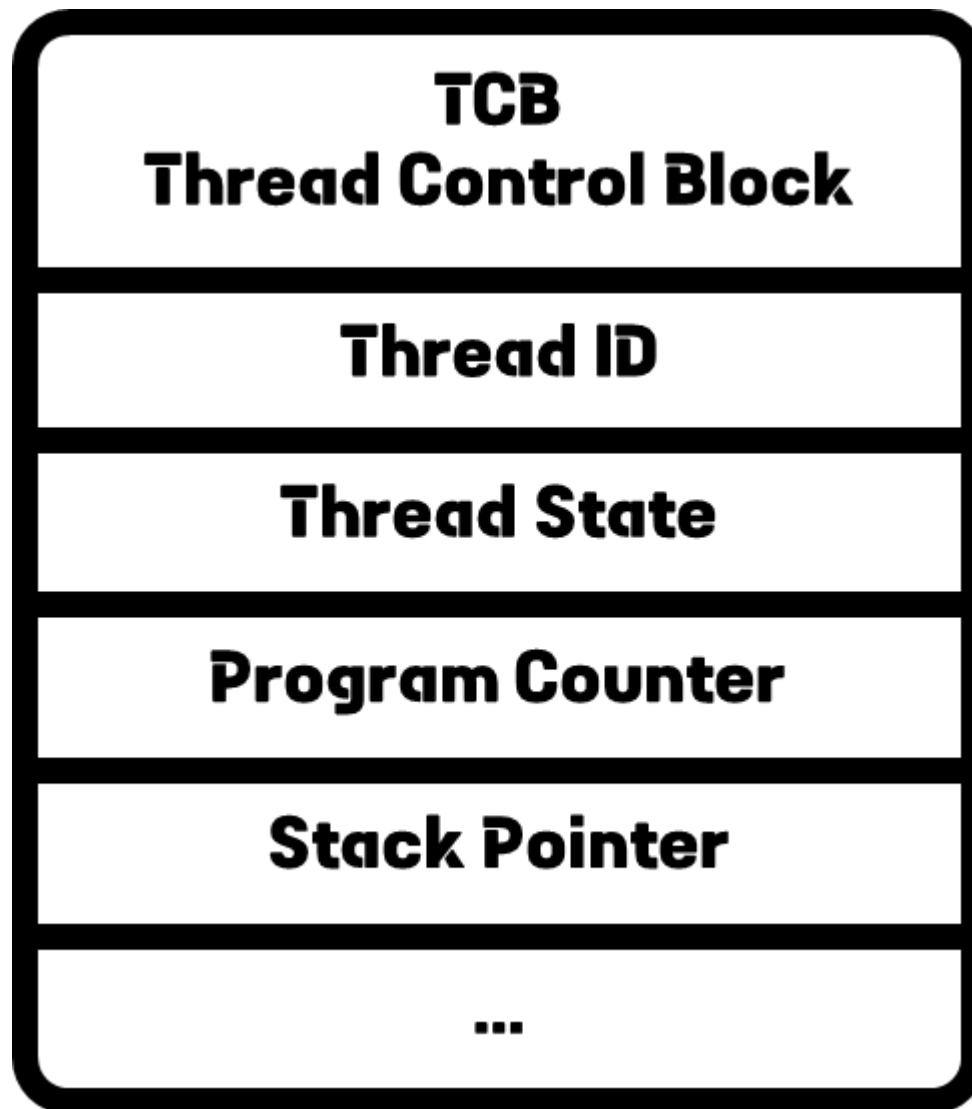
데이터 구조체

Thread Data



스레드 제어 블록
(Thread Control Block)

총 6 가지



스레드 제어 블록 (Thread Control Block)

1. Thread ID (TID)

- 스레드 ID
- 스레드를 식별하기 위한 고유 번호

2. Thread State

- 스레드 상태
- 실행중, 준비, 대기, 시작, 종료 와 같은 상태 정보

3. Program Counter

- 프로그램 계수기
- 현재 실행될 명령의 주소

4. Register Information

- 실행 상태, 실행 정도 등을 저장하는 레지스터 정보

5. Stack Pointer

- 고유 스택 메모리 주소
- 스레드는 같은 프로세스의 Code, Data, Heap 메모리 영역을 공유하지만 Stack 메모리 영역은 고유한 영역을 보유하고 있다

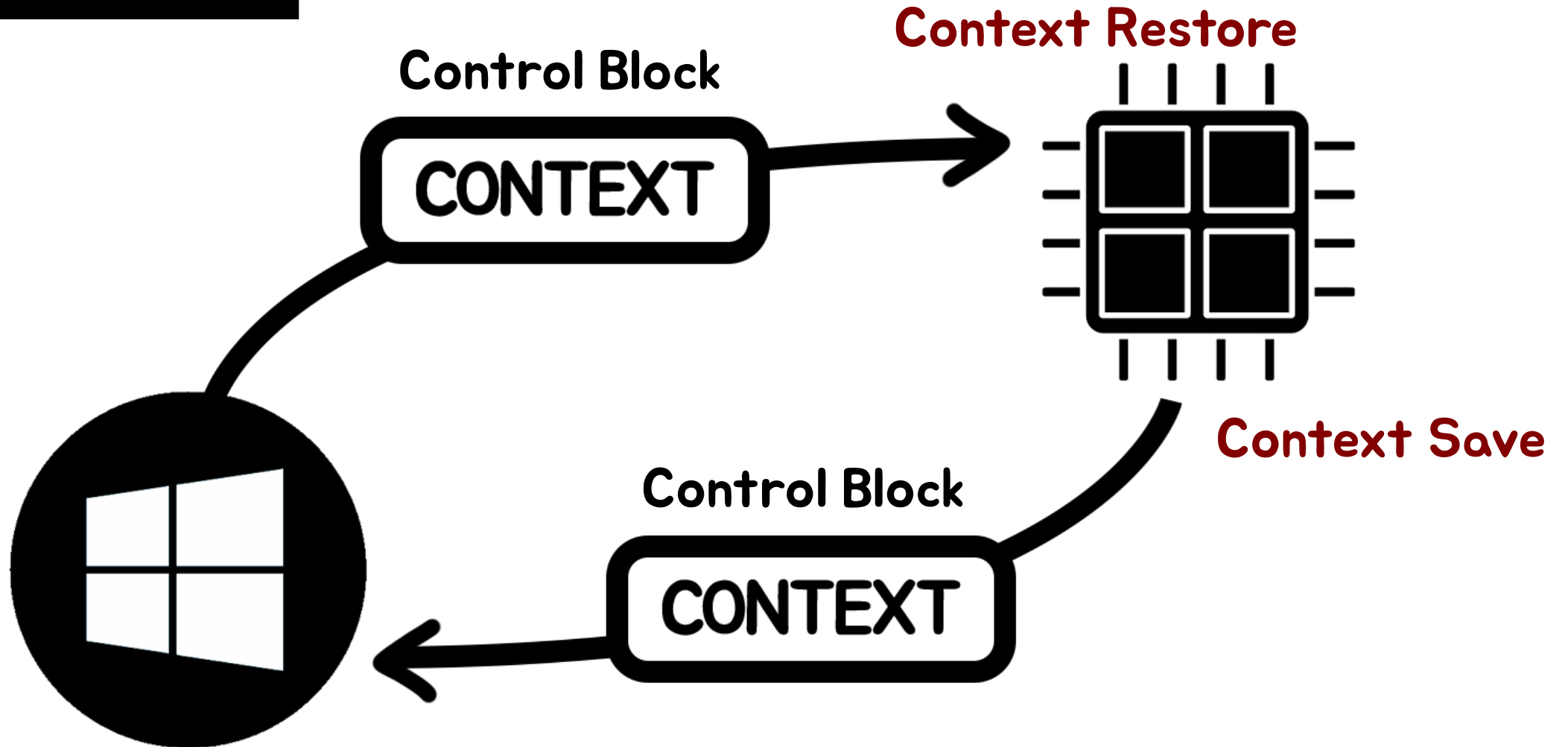
6. PCB Pointer

- 현재 스레드가 작동중인 프로세스 정보 (PCB) 주소

컨텍스트 스위칭 오버헤드

(Context Switching Overhead)

**컨텍스트 스위칭 오버헤드
(Context Switching Overhead)**



**컨텍스트 스위칭 오버헤드
(Context Switching Overhead)**

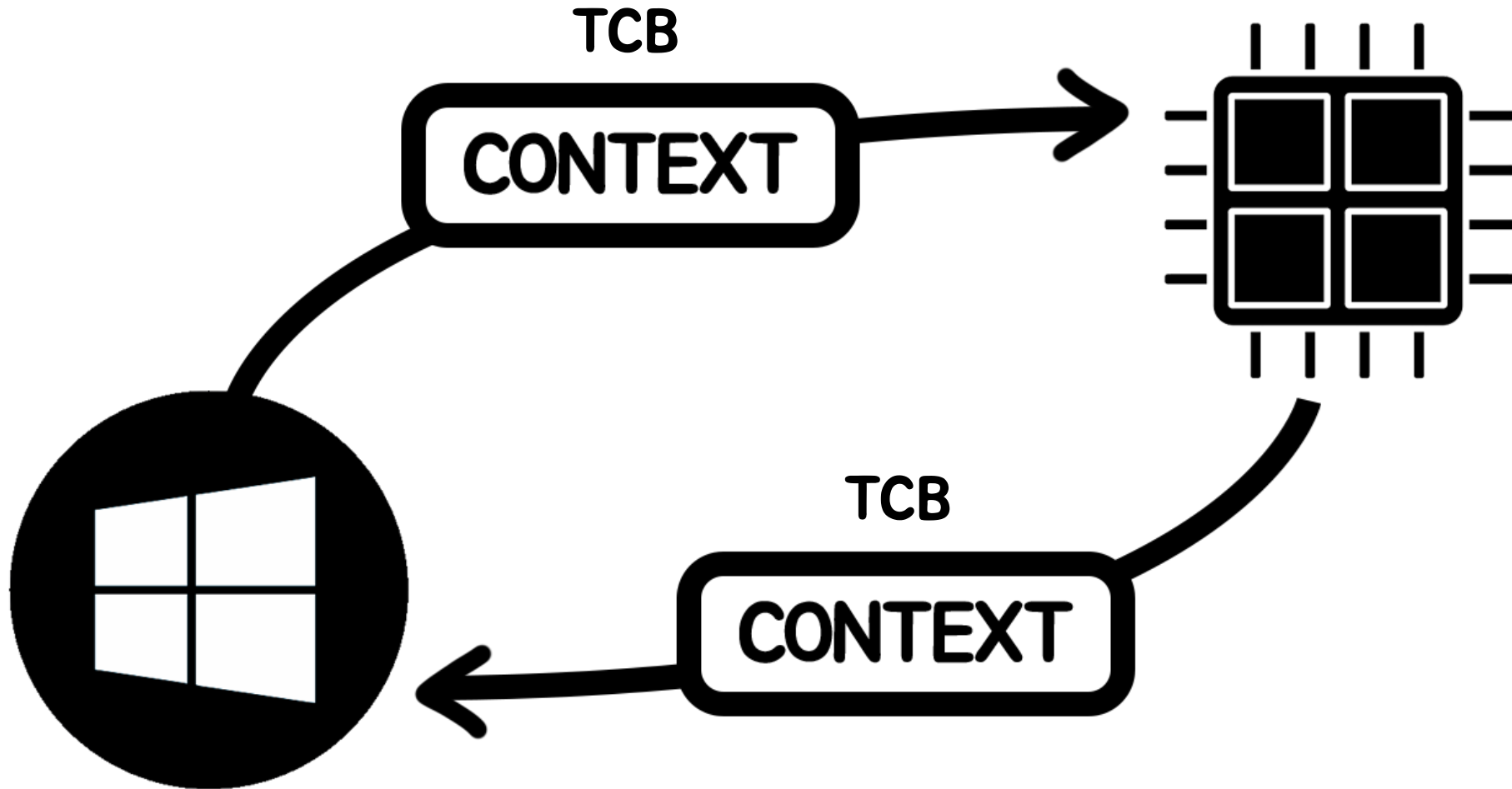
이때 드는 비용

**Context Restore
Context Save**

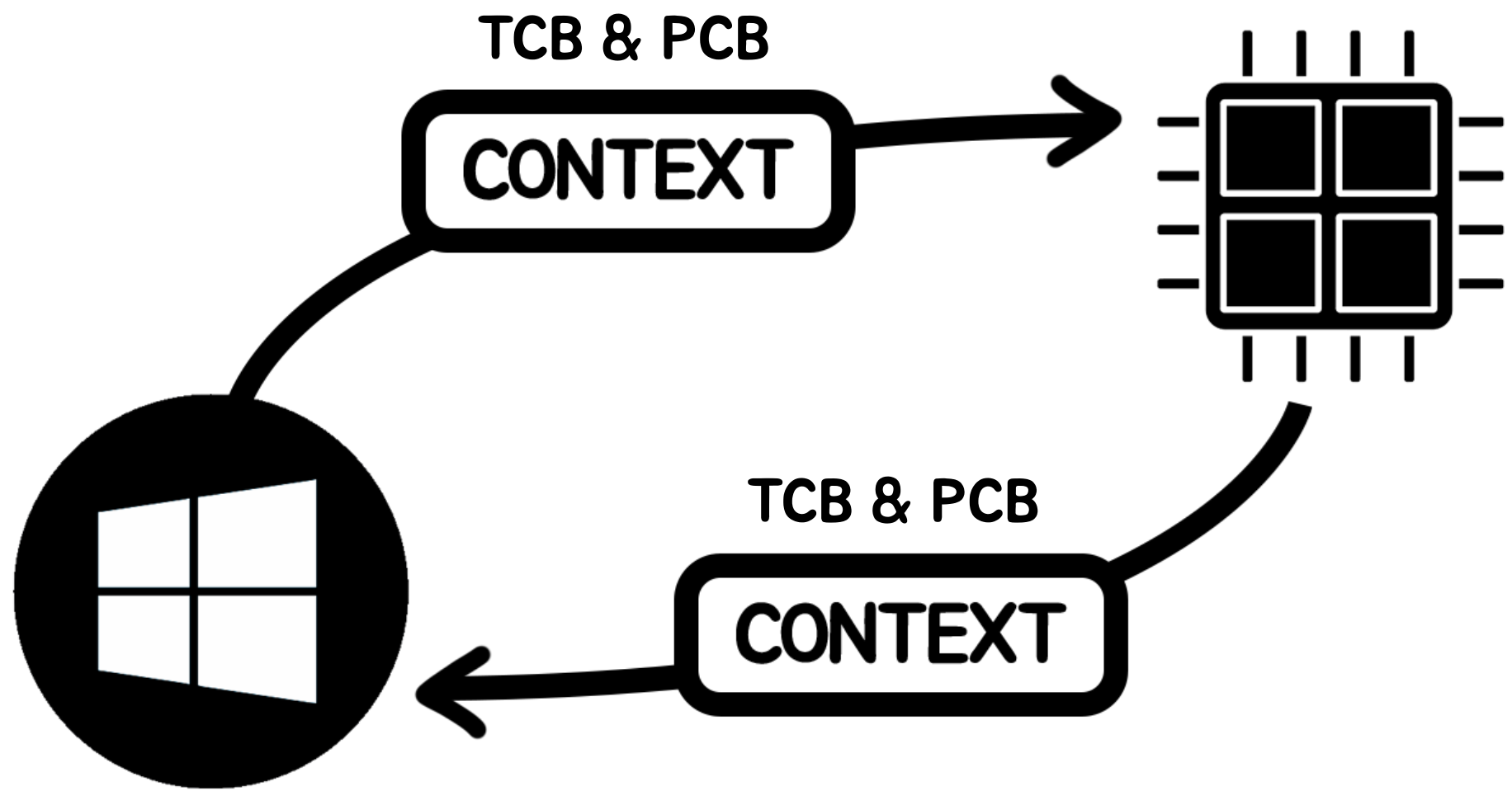


**Context Switching
Overhead**

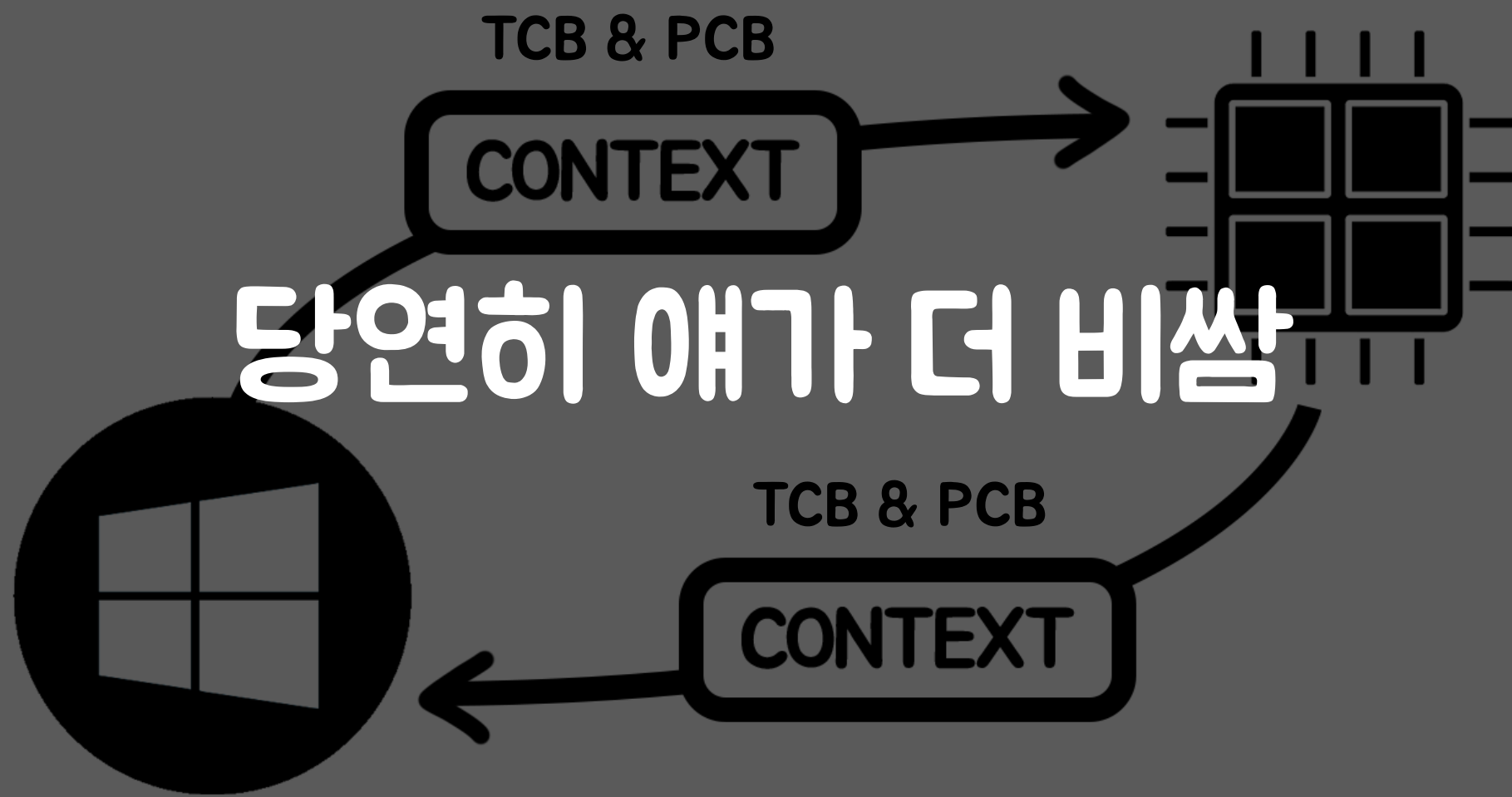
**스레드 컨텍스트 스위칭
(Thread Context Switching)**



**프로세스 컨텍스트 스위칭
(Process Context Switching)**

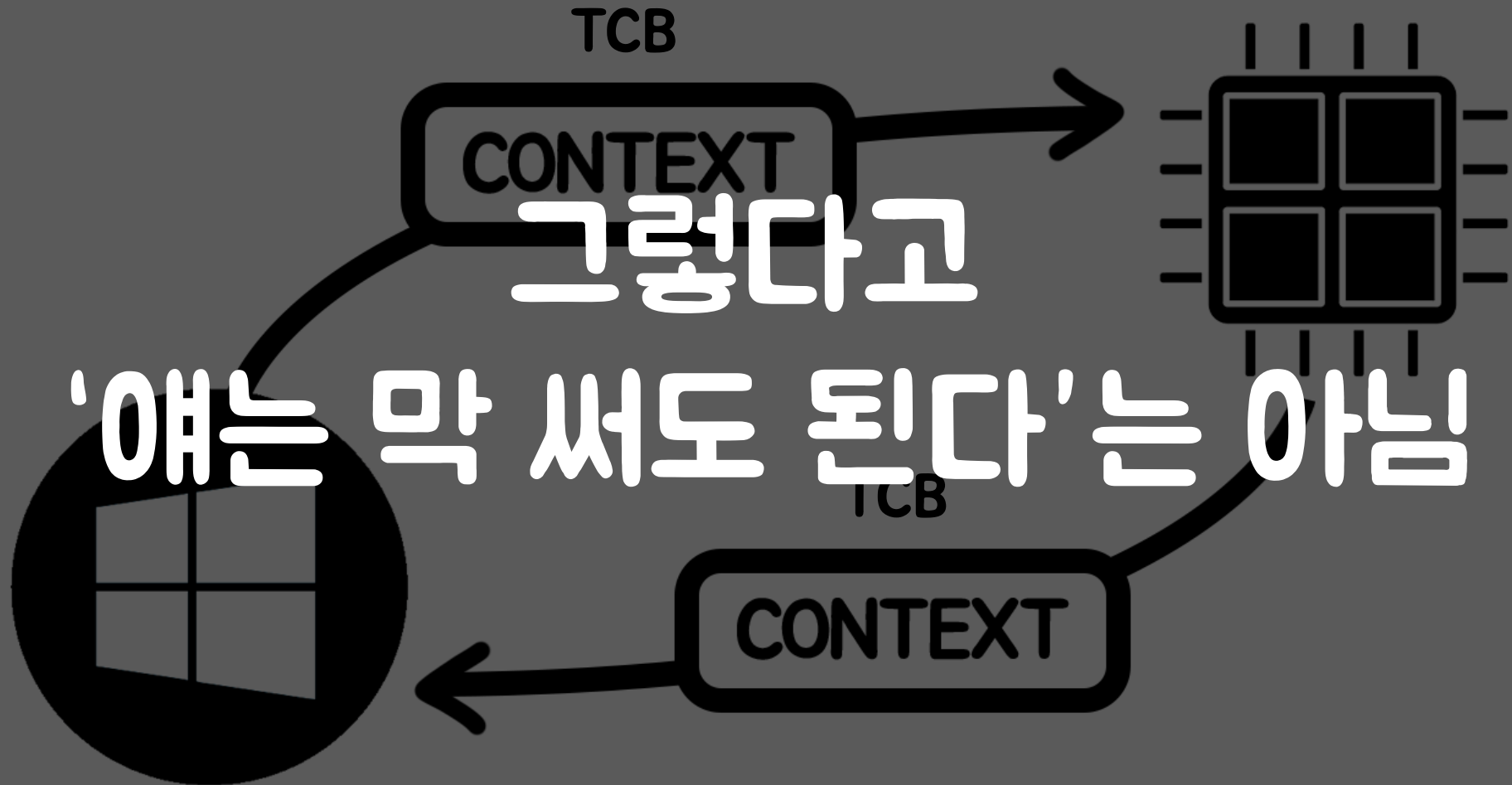


**프로세스 컨텍스트 스위칭
(Process Context Switching)**



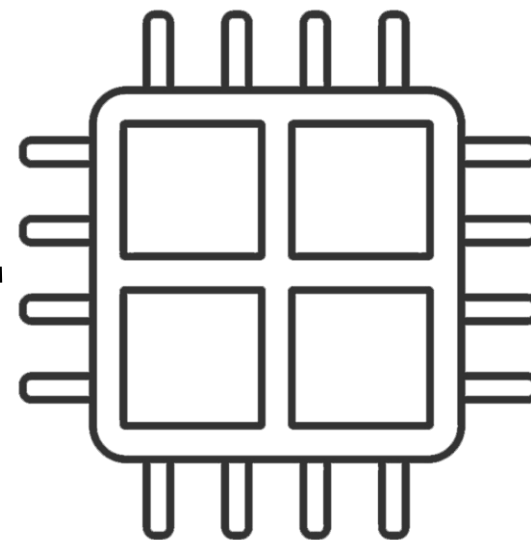
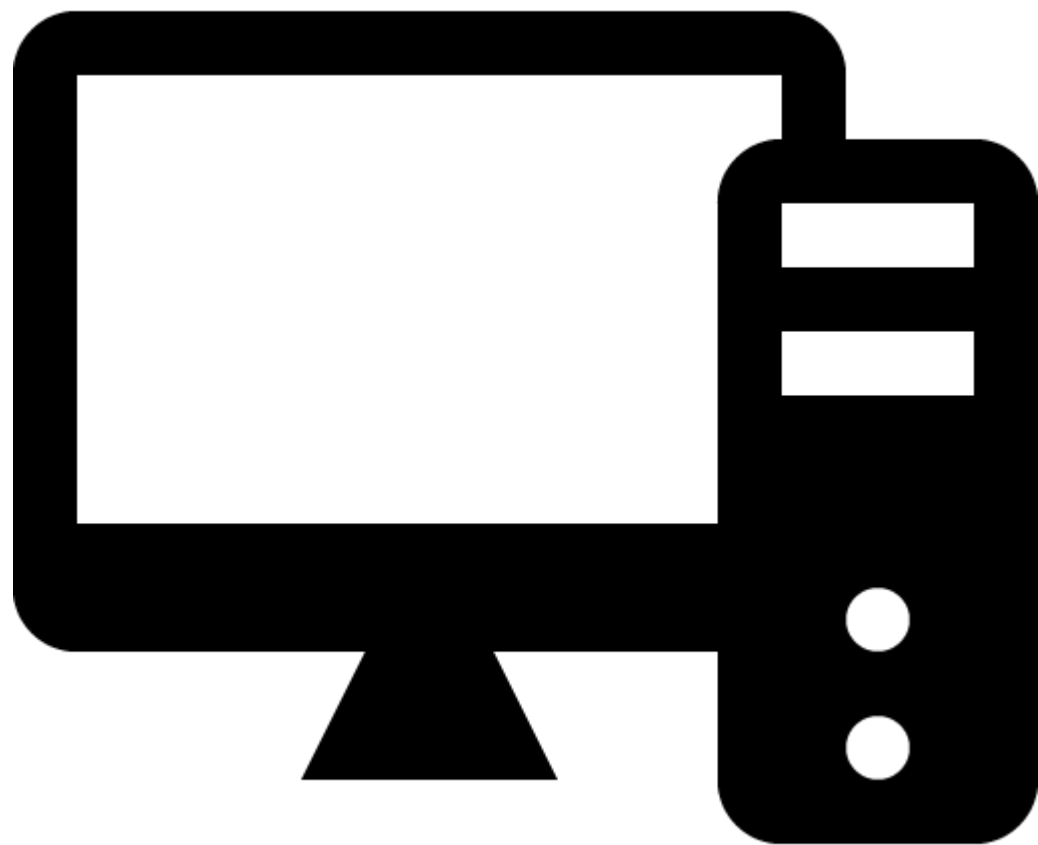
당연히 애가 더 비쌌

스레드 컨텍스트 스위칭
(Thread Context Switching)



멀티스레딩 주의점

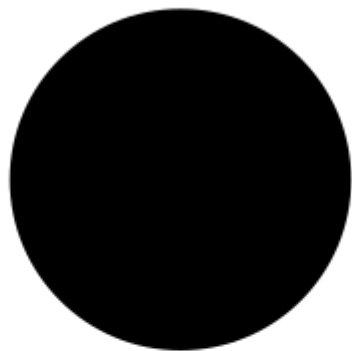
멀티스레딩 주의점



Quad-Core

멀티스레딩 주의점

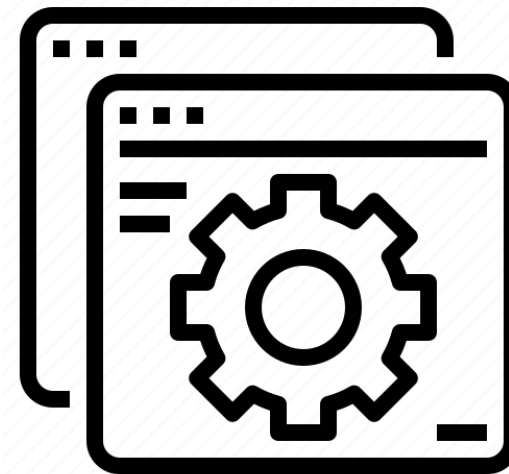
스레드
최고



미친 개발자



스레드
400개



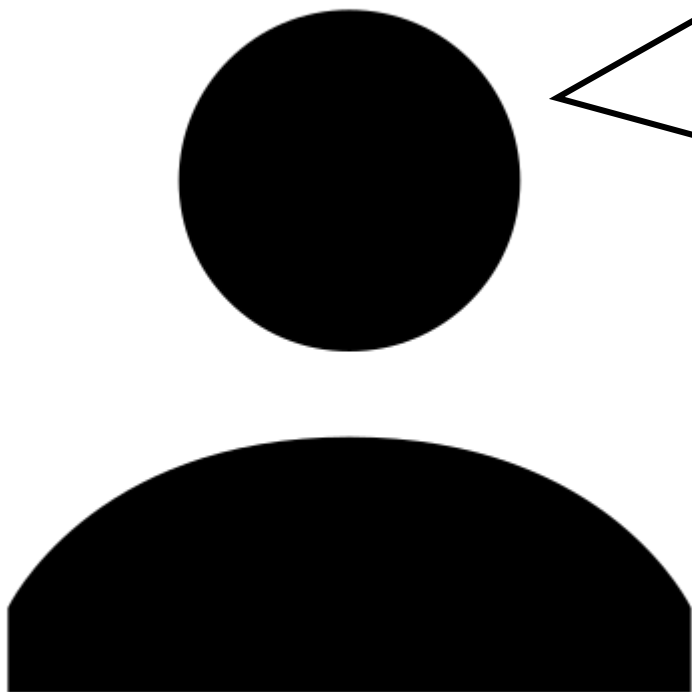
미친 프로그램

멀티스레딩 주의점



미친 개발자

멀티스레딩 주의점



미친 개발자

하나의 프로세스에 여러 개의 스레드를 할
당하면 속도가 늘어난대!

스레드를 400개 만들어버리자!!

하나의 스레드가 1초가 걸리는 작업씩 담당
하면 총 400초가 걸리는 작업을 1초안에
끝낼 수 있겠지?

이론적으로 싹가능 〇〇

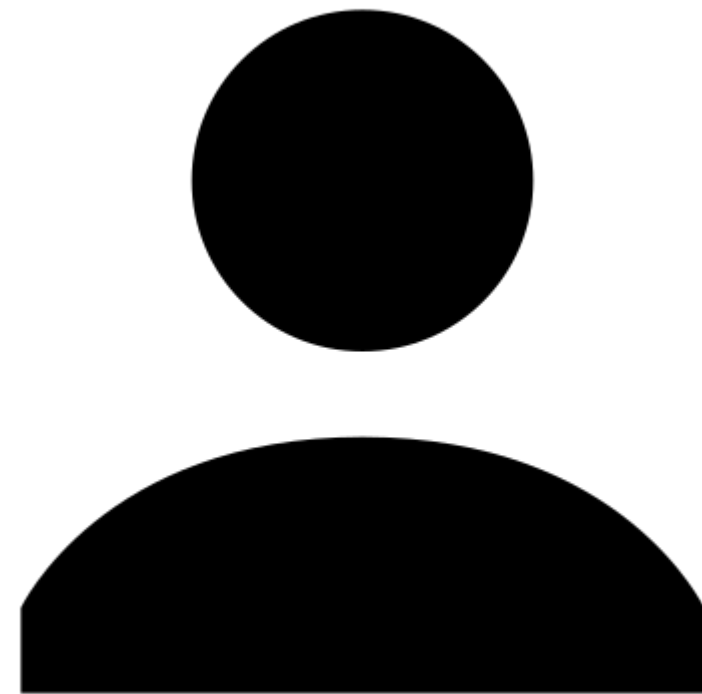
멀티스레딩 주의점

그게 되겠냐 임마.

쿼드 코어를 쓰게 되면 한번에 4개의
스레드만 실행되겠지.

따라서 400개의 스레드를 한번에 4개씩
100번에 걸쳐 실행하게 될 거임.

그렇게 되면 100초라는 시간이 걸리겠지.



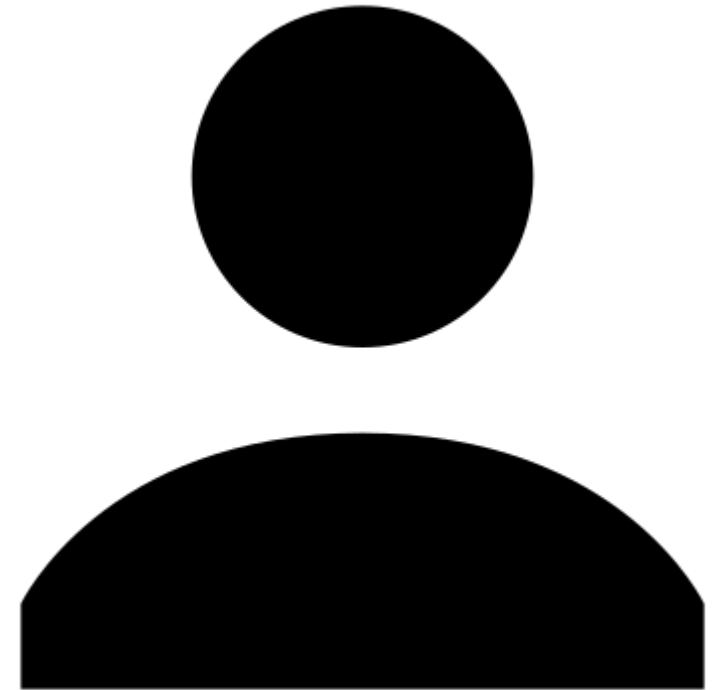
선한샘

멀티스레딩 주의점

그리고 각 코어가 100번씩 걸쳐서 스레드를 실행한다는 소리는 컨텍스트 스위칭이 각각 100번 발생한다는 소리랑 같음.

즉 100번의 스레드 컨텍스트 스위칭 오버헤드가 발생하는 거다.

이렇게 되면 100초라는 시간조차 기대하기 힘들 수 있어.

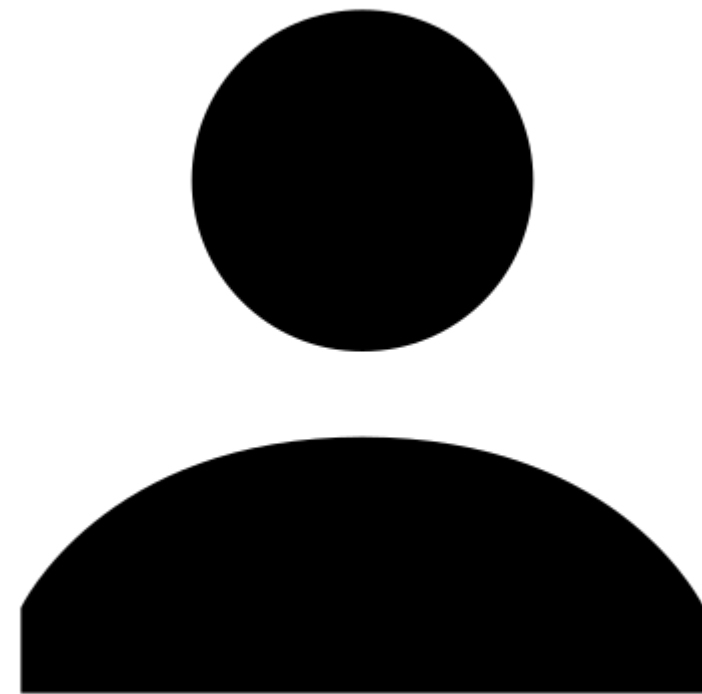


선한샘

멀티스레딩 주의점

혼자 신나서 멀티스레딩 하겠다고 아무리 설치대도 컨텍스트 스위칭 오버헤드를 고려하지 않으면 싱글스레드 보다는 못한 퍼포먼스를 내는 상황이 생긴다.

멀티스레딩 환경을 구축할 땐 꼭 알아두도록 해.



선한샘