

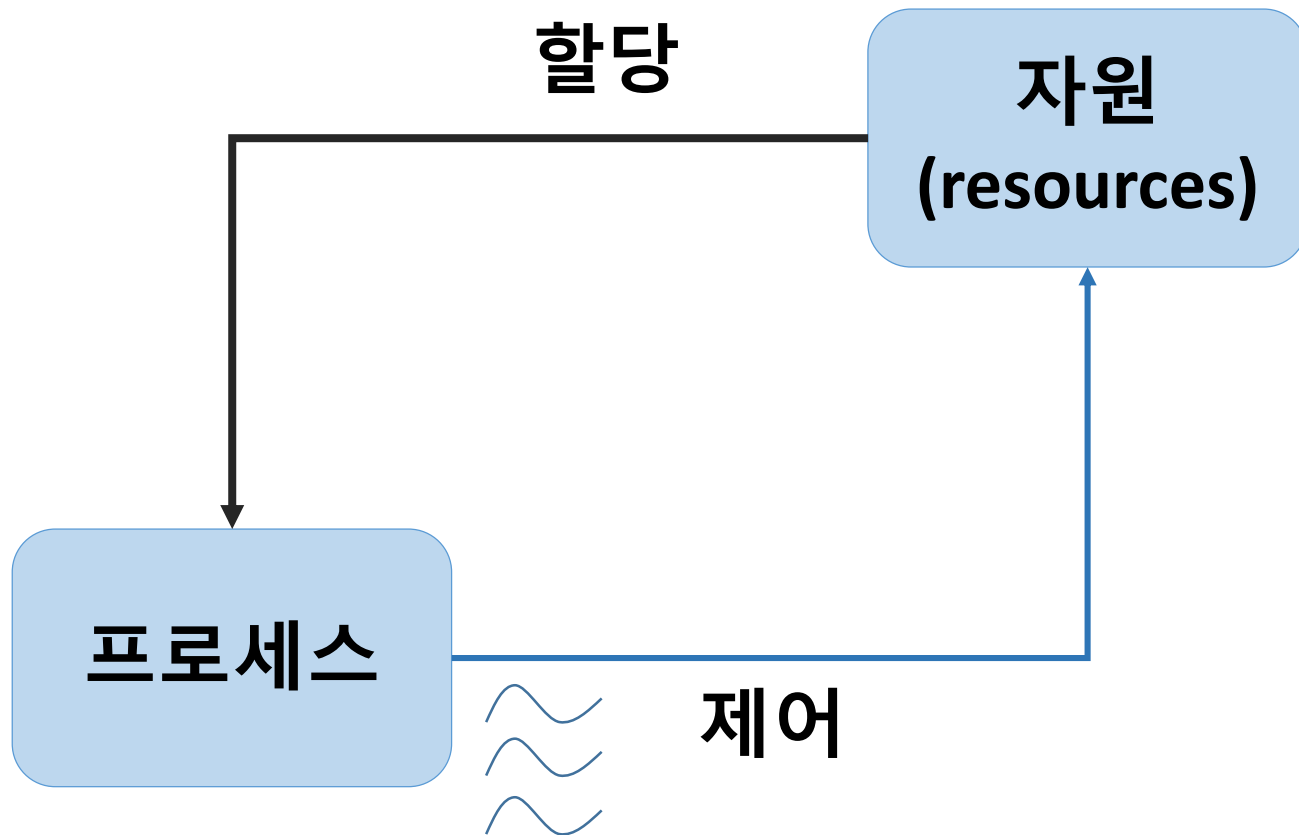
Chapter 3-2

스레드 관리

Thread Management



스레드(Thread)의 개념



스레드(Thread)의 개념



- SP^{Stack Pointer} : 스택 포인터
- SR^{Sequence Register} : 순서열 레지스터
- PC^{Program Counter} : 프로그램 카운터

그림 3-12 스레드의 구조



스레드(Thread)의 개념

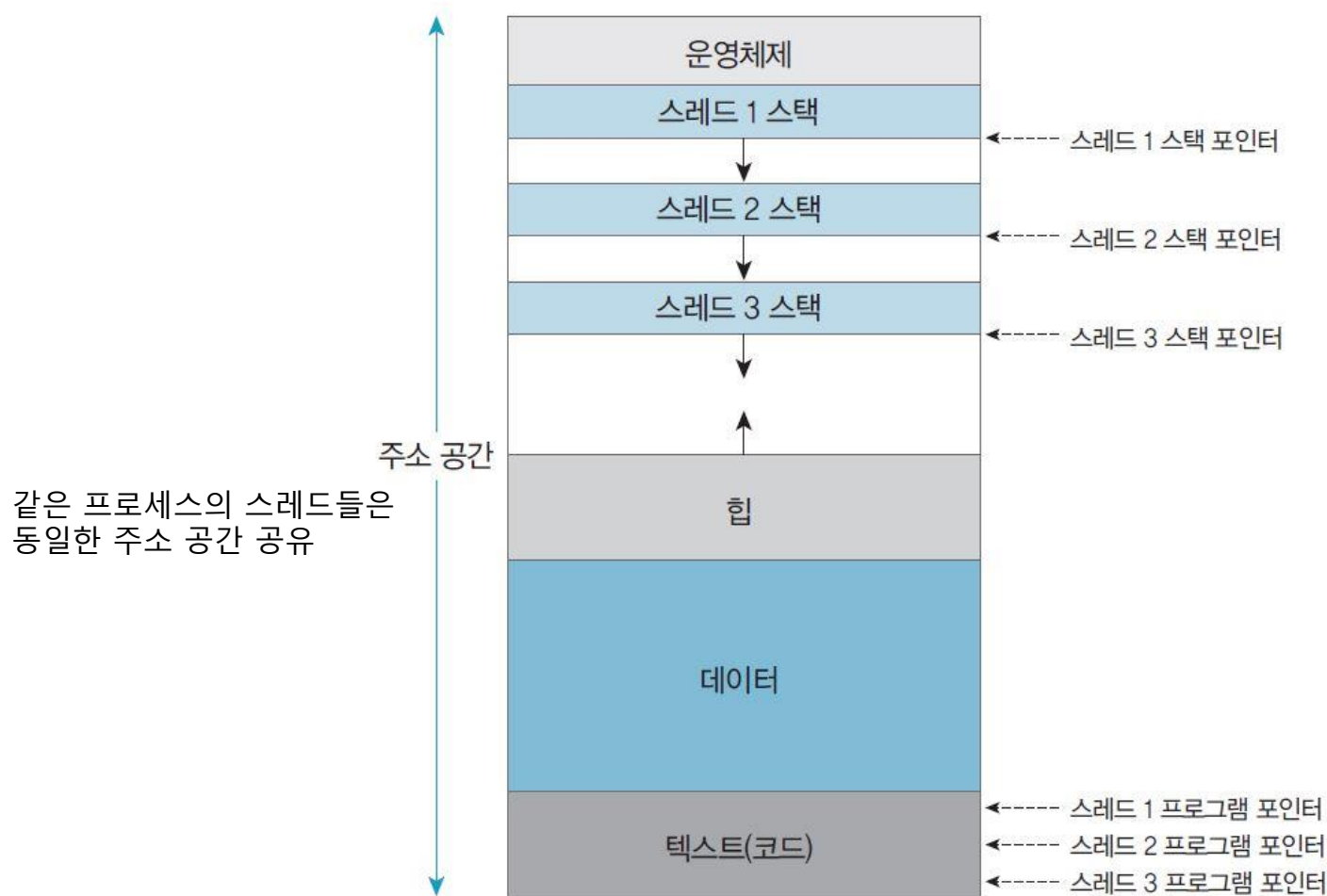


그림 3-13 스레드의 주소 공간



스레드(Thread)의 개념

- Light Weight Process (LWP)
- 프로세서(e.g, CPU) 활용의 기본 단위
- 구성요소
 - Thread ID
 - Register set (PC 등)
 - Stack (i.e. local data)
- 제어 요소 외 코드, 데이터 및 자원들은 프로세스의 다른 스레드 들과 공유
- 전통적 프로세스 = 단일 스레드 프로세스



Single-thread vs Multi-threads



(a) 단일 스레드의 프로세스

(b) 다중 스레드의 프로세스

다중 처리(다중 프로세스) vs 다중 스레드?



스레드 장점

- **사용자 응답성 (Responsiveness)**

- 일부 스레드의 처리가 지연되어도, 다른 스레드는 작업을 계속 처리 가능

- **자원 공유 (Resource sharing)**

- 자원을 공유해서 효율성 증가 (커널의 개입을 피할 수 있음)
 - 예) 동일 address space에서 스레드 여러 개

- **경제성 (Economy)**

- 프로세스의 생성, context switch에 비해 효율적

- **멀티 프로세서(multi-processor) 활용**

- 병렬처리를 통해 성능 향상



스레드 사용의 예

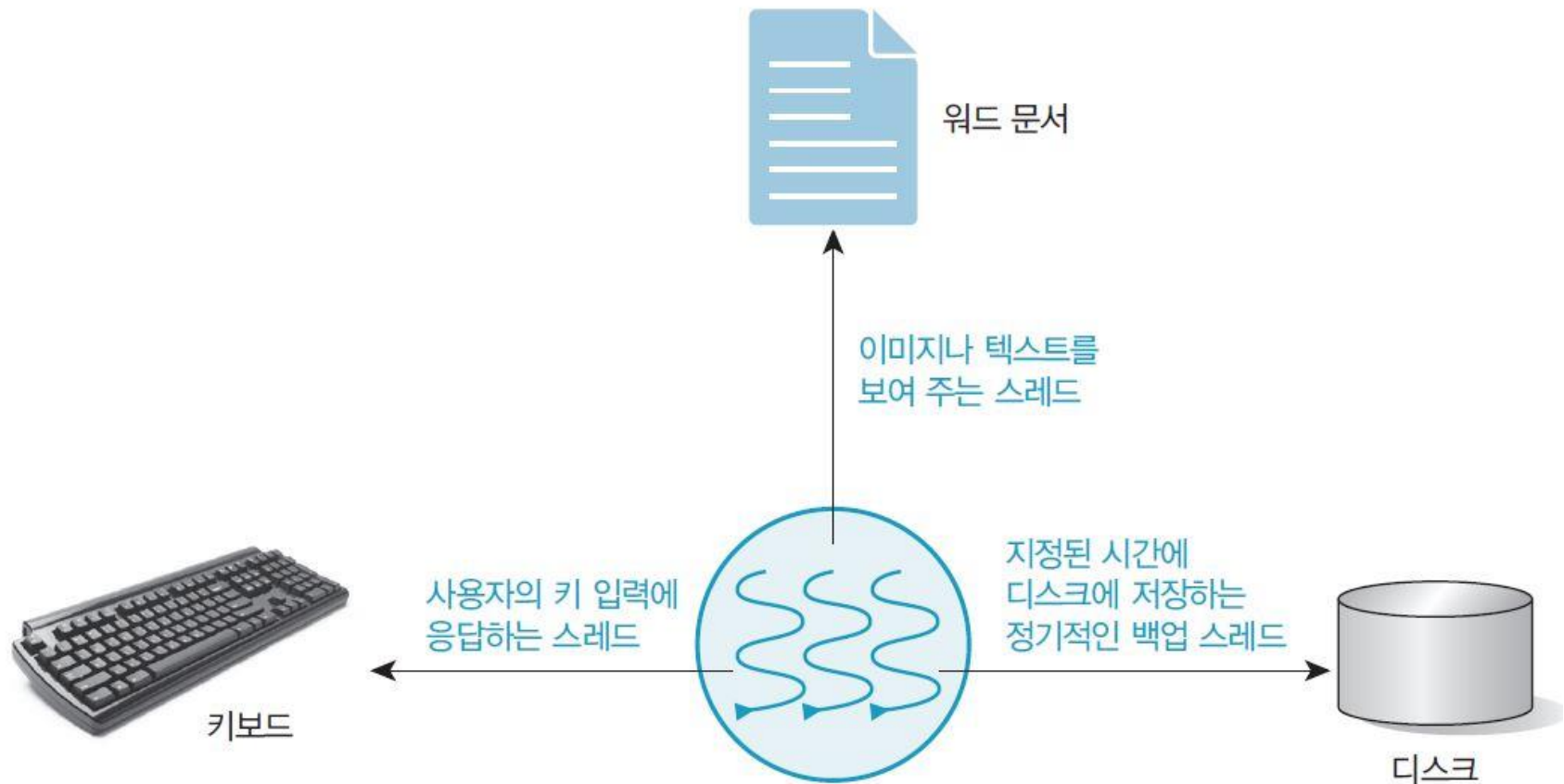


그림 3-16 다중 스레드를 이용한 워드 편집기 프로세스

스레드 사용의 예

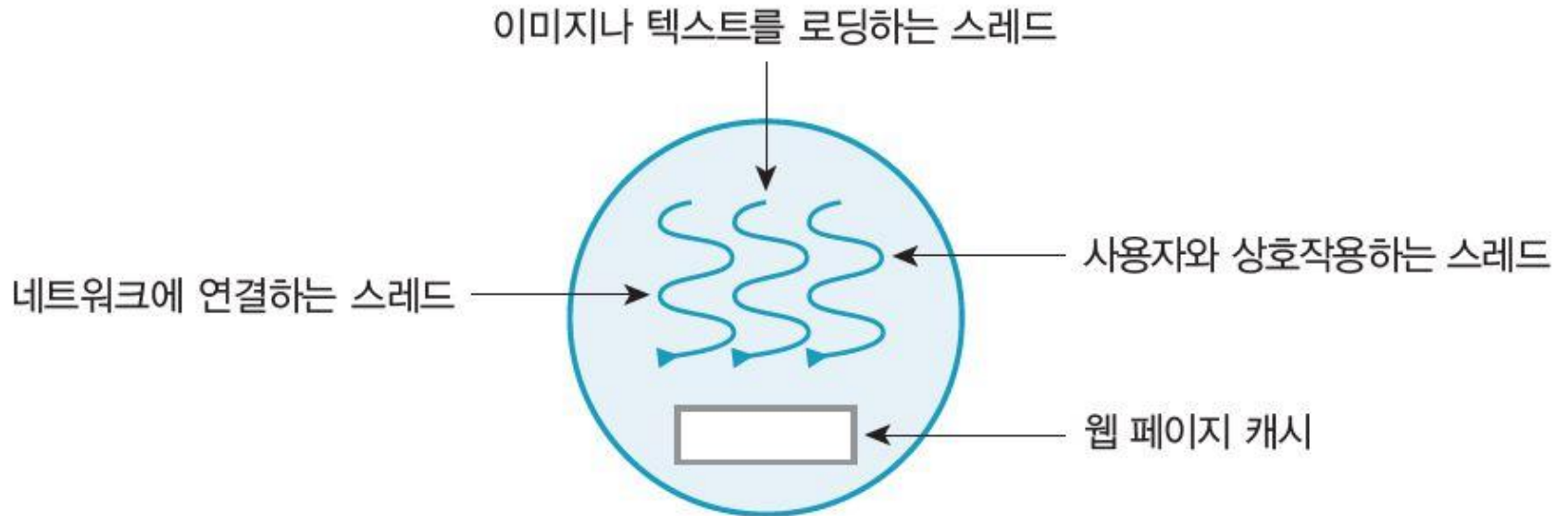


그림 3-17 다중 스레드를 이용한 웹 브라우저 프로세스

스레드(Thread)의 구현

- 사용자 수준 스레드 (User threads)
- 커널 수준 스레드 (Kernel threads)

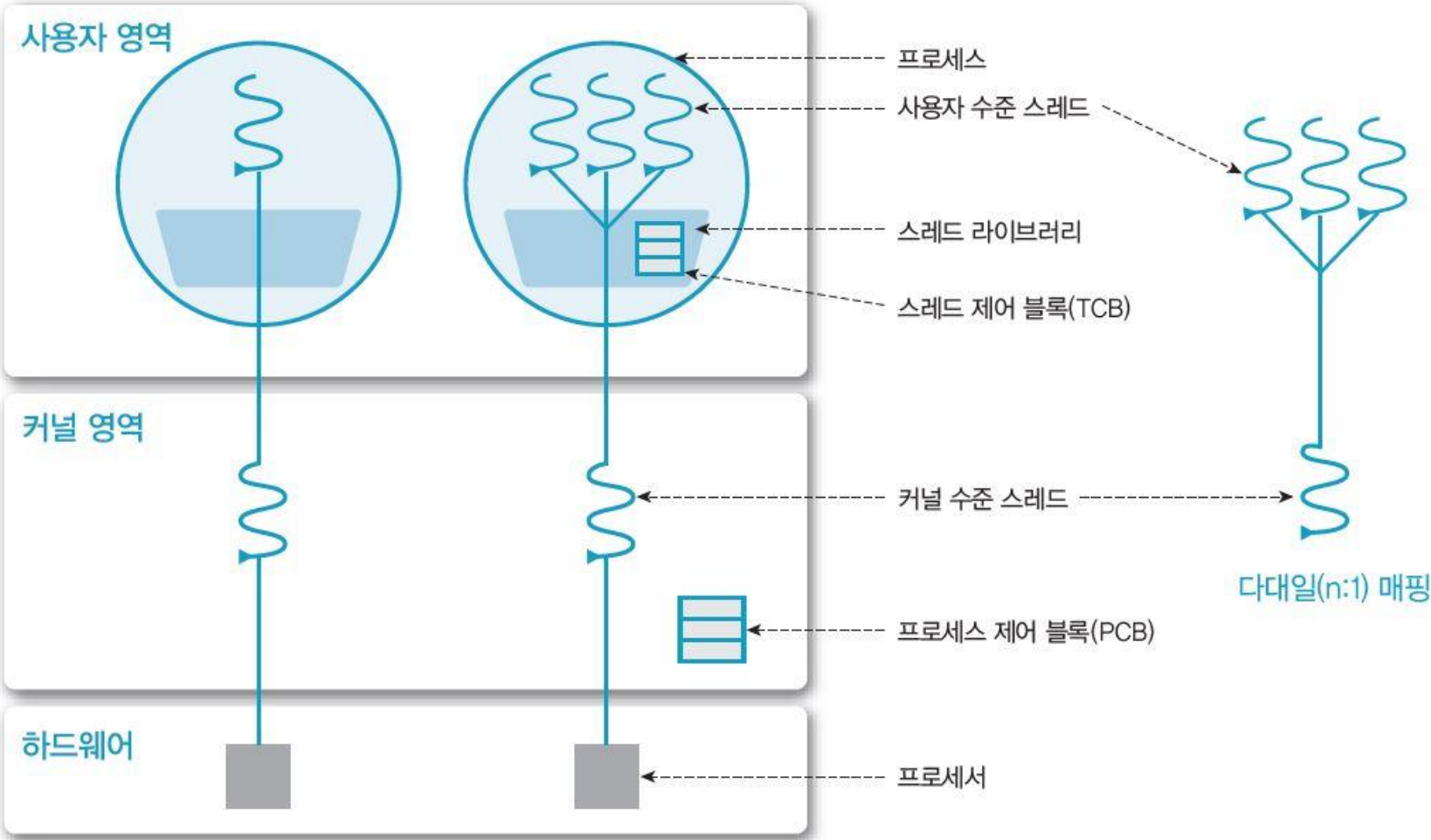


사용자 수준 스레드 (User Threads)

- 사용자 영역의 스레드 라이브러리로 구현 됨
 - 스레드의 생성, 스케줄링 등
 - POSIX threads, Win32 threads, Java thread API 등
- 커널은 스레드의 존재를 모름
 - 커널의 관리(개입)를 받지 않음
 - 생성 및 관리의 부하가 적음, 유연한 관리 가능
 - 이식성(portability)이 높음
 - 커널은 프로세스 단위로 자원 할당
 - 하나의 스레드가 block 상태가 되면, 모든 스레드가 대기 (single-threaded kernel의 경우)



사용자 수준 스레드 (User Threads)

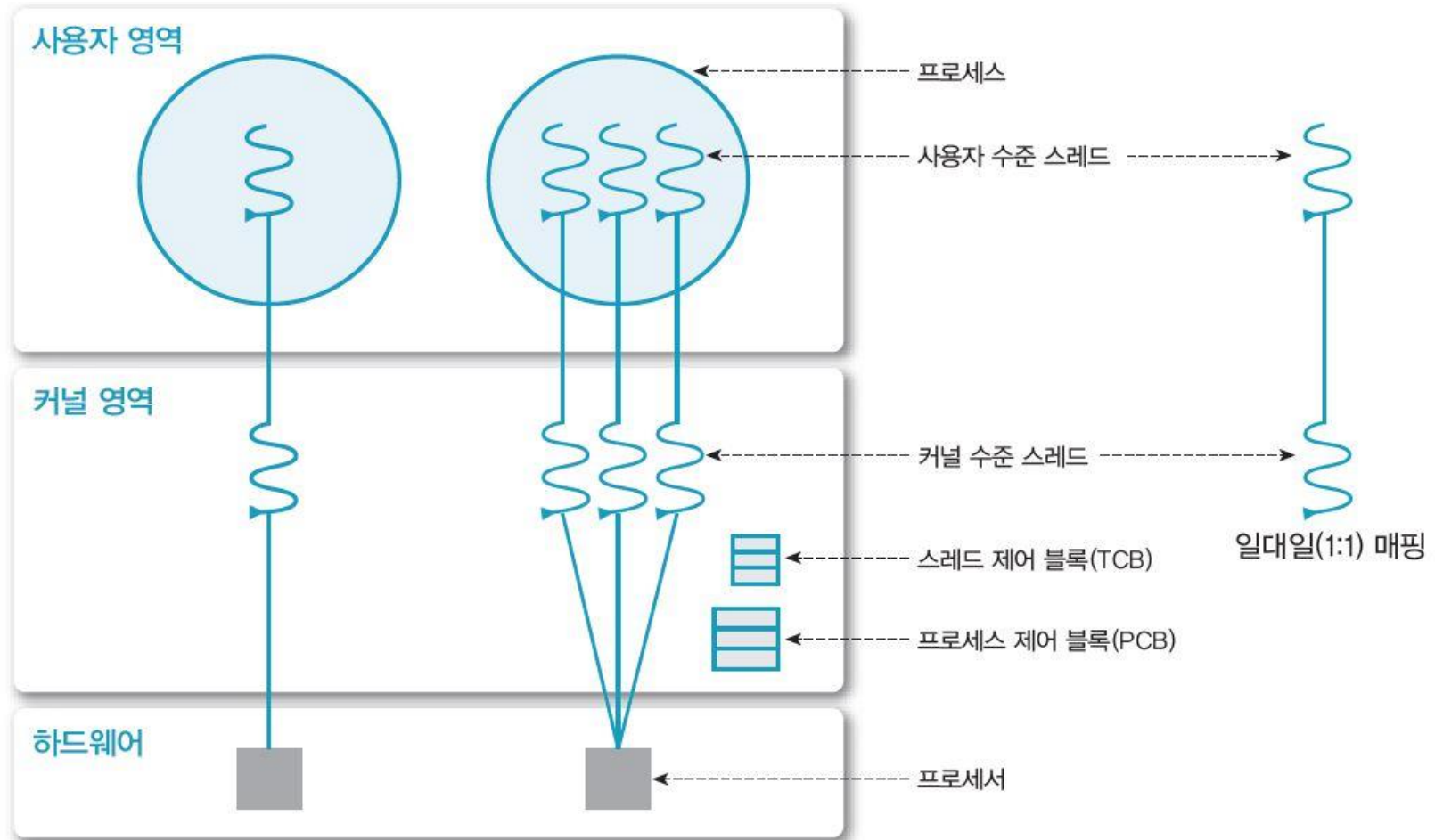


커널 수준 스레드 (Kernel Threads)

- OS(Kernel)이 직접 관리
- 커널 영역에서 스레드의 생성, 관리 수행
 - Context switching 등 부하(Overhead)가 큼
- 커널이 각 스레드를 개별적으로 관리
 - 프로세스 내 스레드들이 병행 수행 가능
 - 하나의 스레드가 block 상태가 되어도, 다른 스레드는 계속 작업 수행 가능



커널 수준 스레드 (Kernel Threads)



Language-level Threads

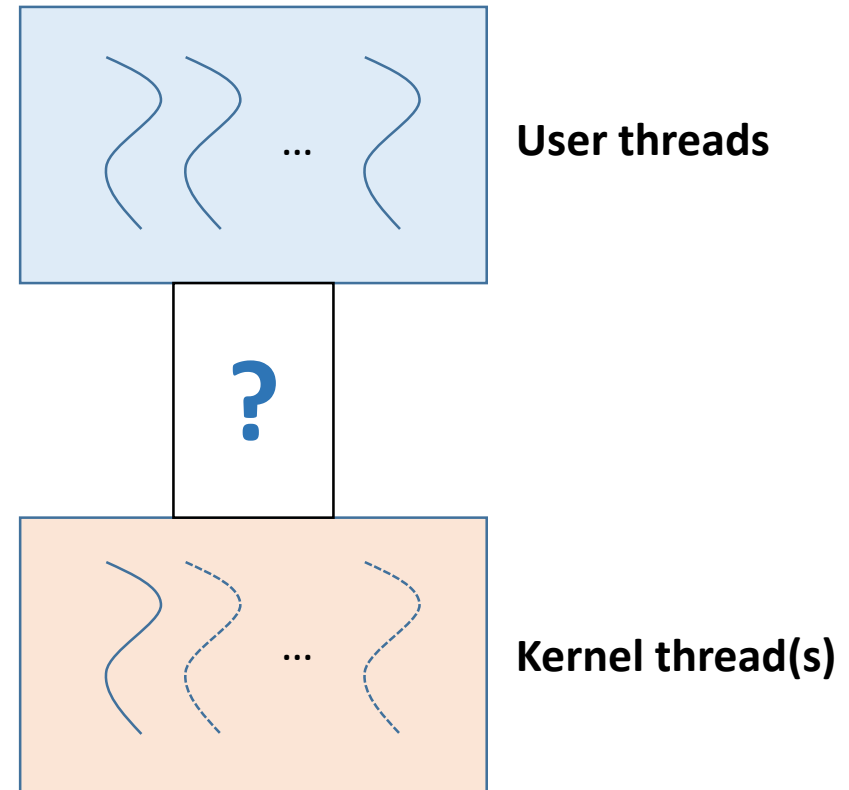
- **Java threads**

- Java Virtual Machine (JVM)에 의해 생성 관리 됨
 - Middleware-level threads



Multi-Threading Model

- 다대일(n:1) 모델
 - 사용자 수준 스레드
- 일대일(1:1) 모델
 - 커널 수준 스레드
- 다대다(n:m) 모델
 - $n > m$
 - 혼합형 스레드

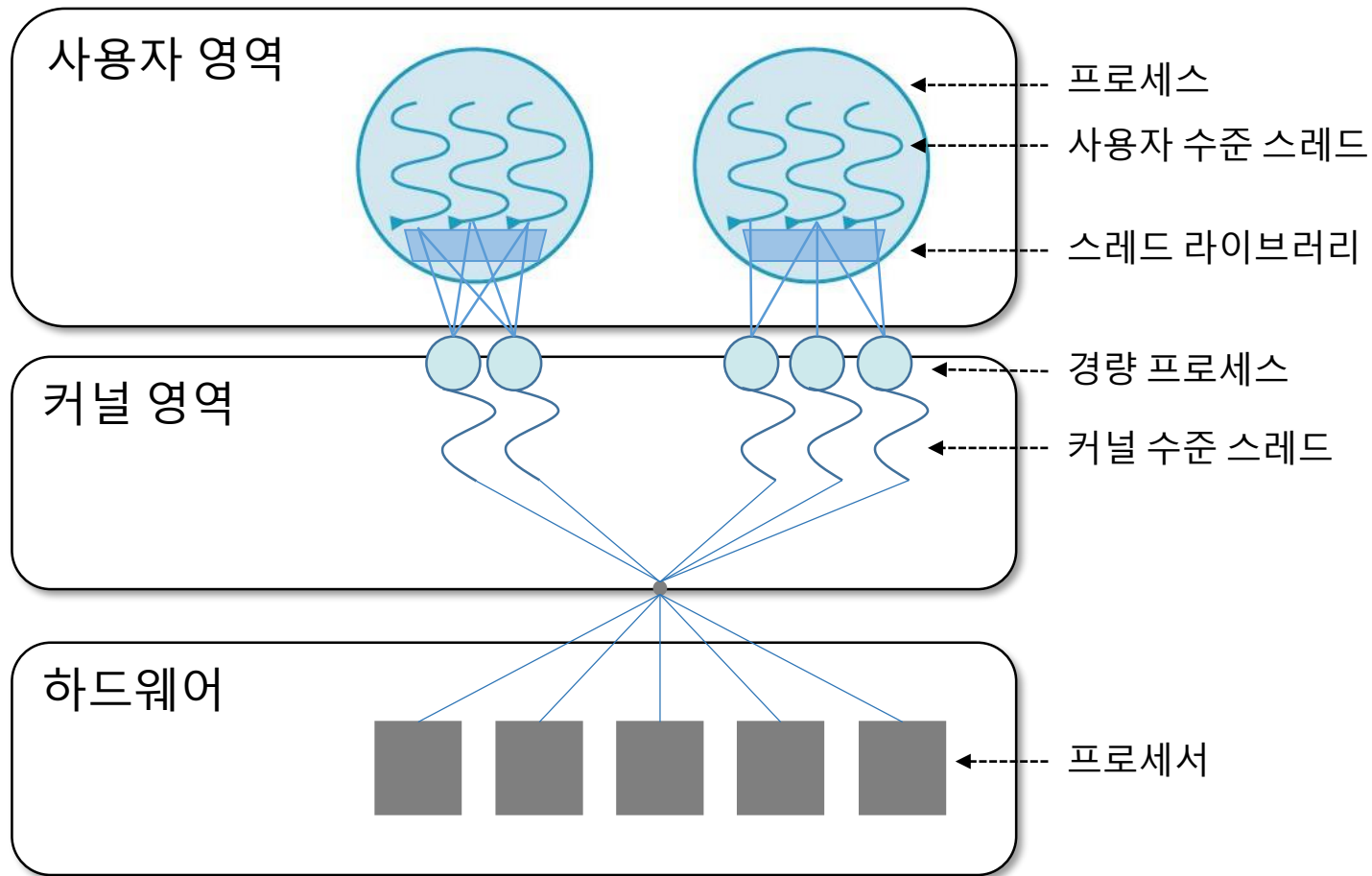


혼합형 (n:m) 스레드

- n개 사용자 수준 스레드 – m개의 커널 스레드
($n > m$)
 - 사용자는 원하는 수만 큼 스레드 사용
 - 커널 스레드는 자신에게 할당된 하나의 사용자 스레드가 block 상태가 되어도, 다른 스레드 수행 가능
 - 병행 처리 가능
- 효율적이면서도 유연함



혼합형 (n:m) 스레드



요약

- 스레드(Thread)의 개념
- 스레드의 구현
 - 사용자 수준 스레드
 - n:1 모델
 - 커널 수준 스레드
 - 1:1 모델
 - 혼합형 스레드
 - n:m 모델

