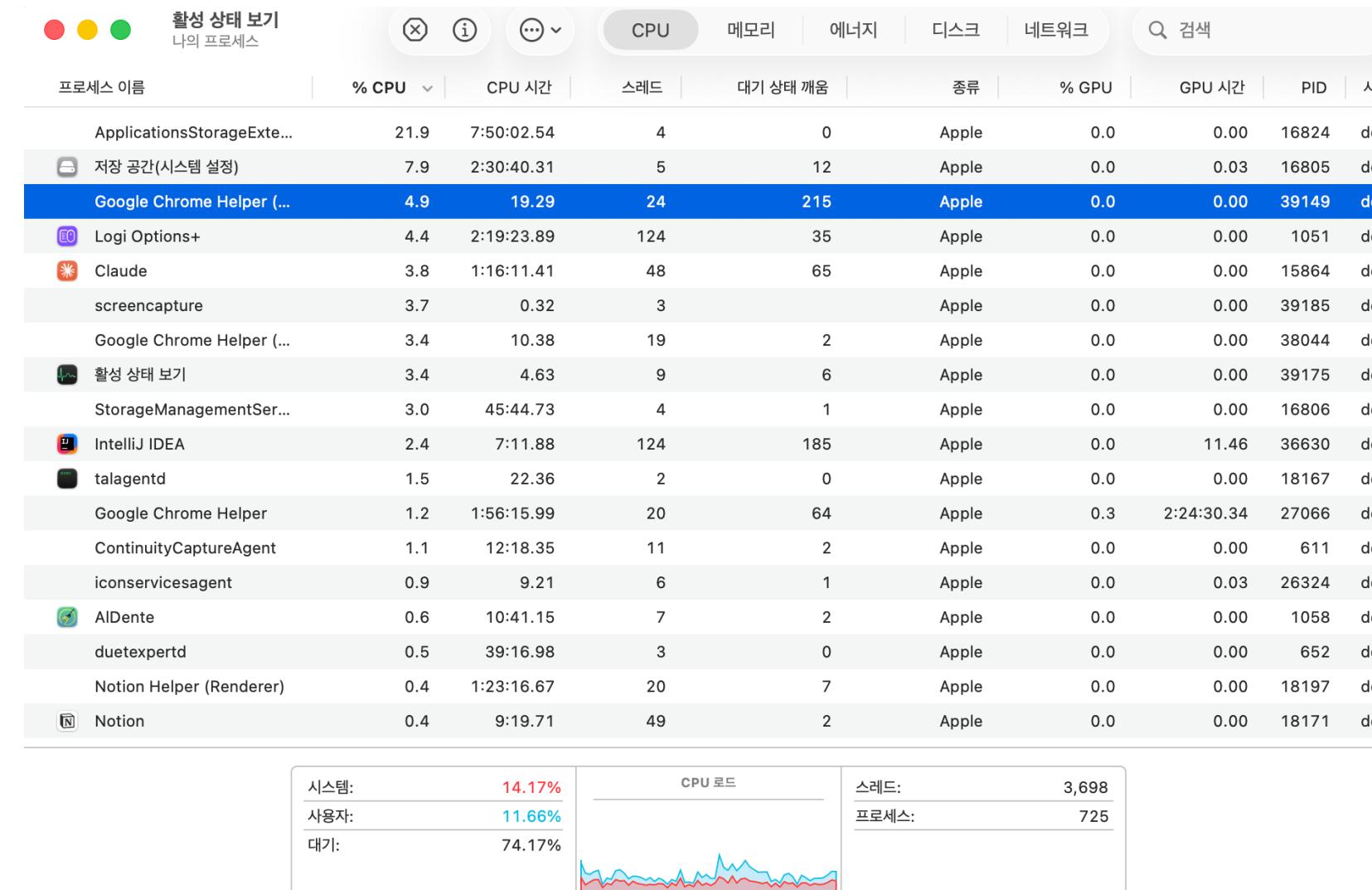


10장. 프로세스와 스레드

프로세스란?

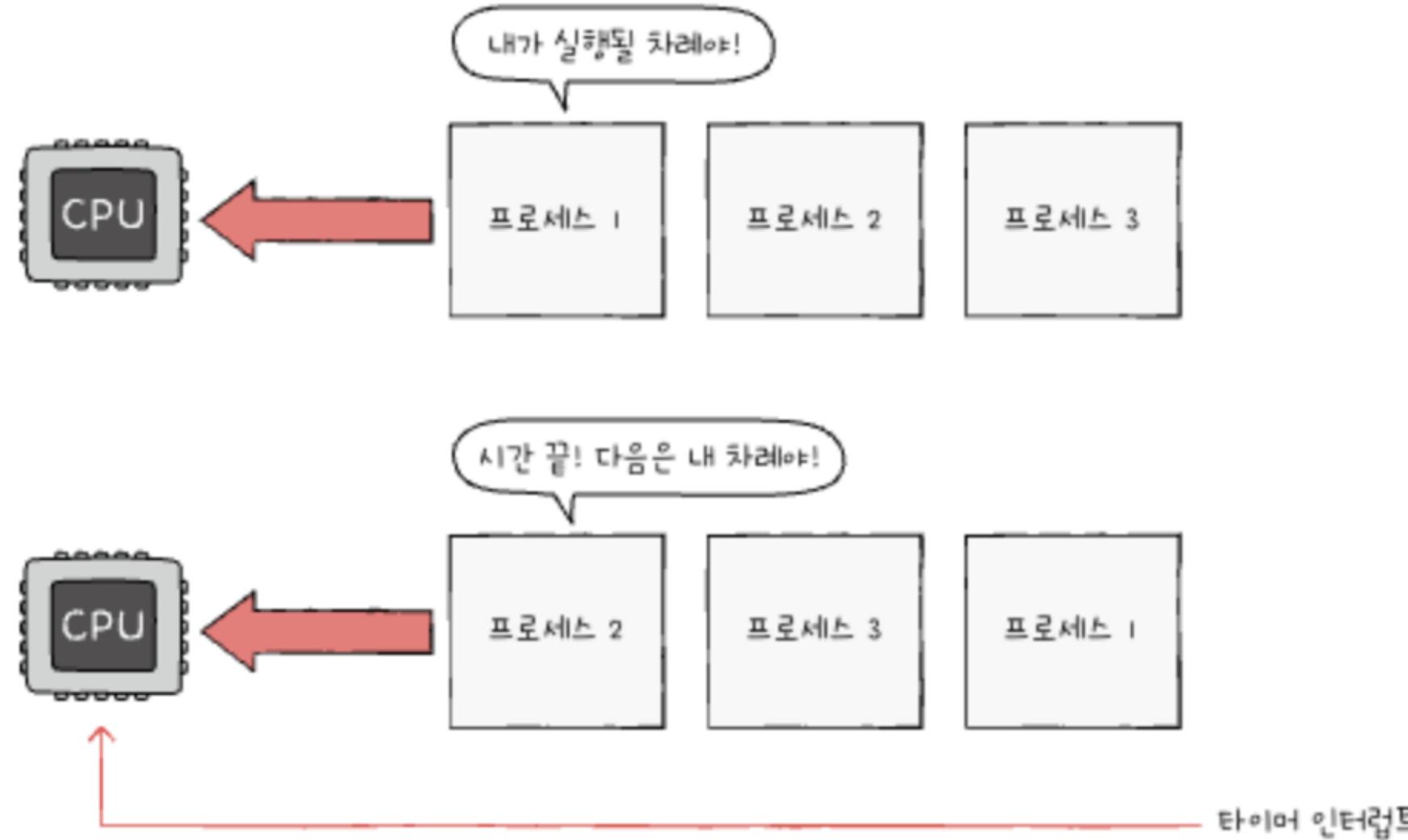


실행 중인 모든 프로그램을 프로세스라 표현

사용자가 보는 앞에서 실행되는 프로세스는 포그라운드 프로세스

사용자가 보지 못하는 뒤에서 실행되는 프로세스는 백그라운드 프로세스

프로세스 제어 블록(PCB)



프로세스들은 한정된 시간만큼 돌아가며 CPU를 사용한다.
운영체제는 PCB를 통해 프로세스 실행 순서를 관리하고, 자원을 배분한다.

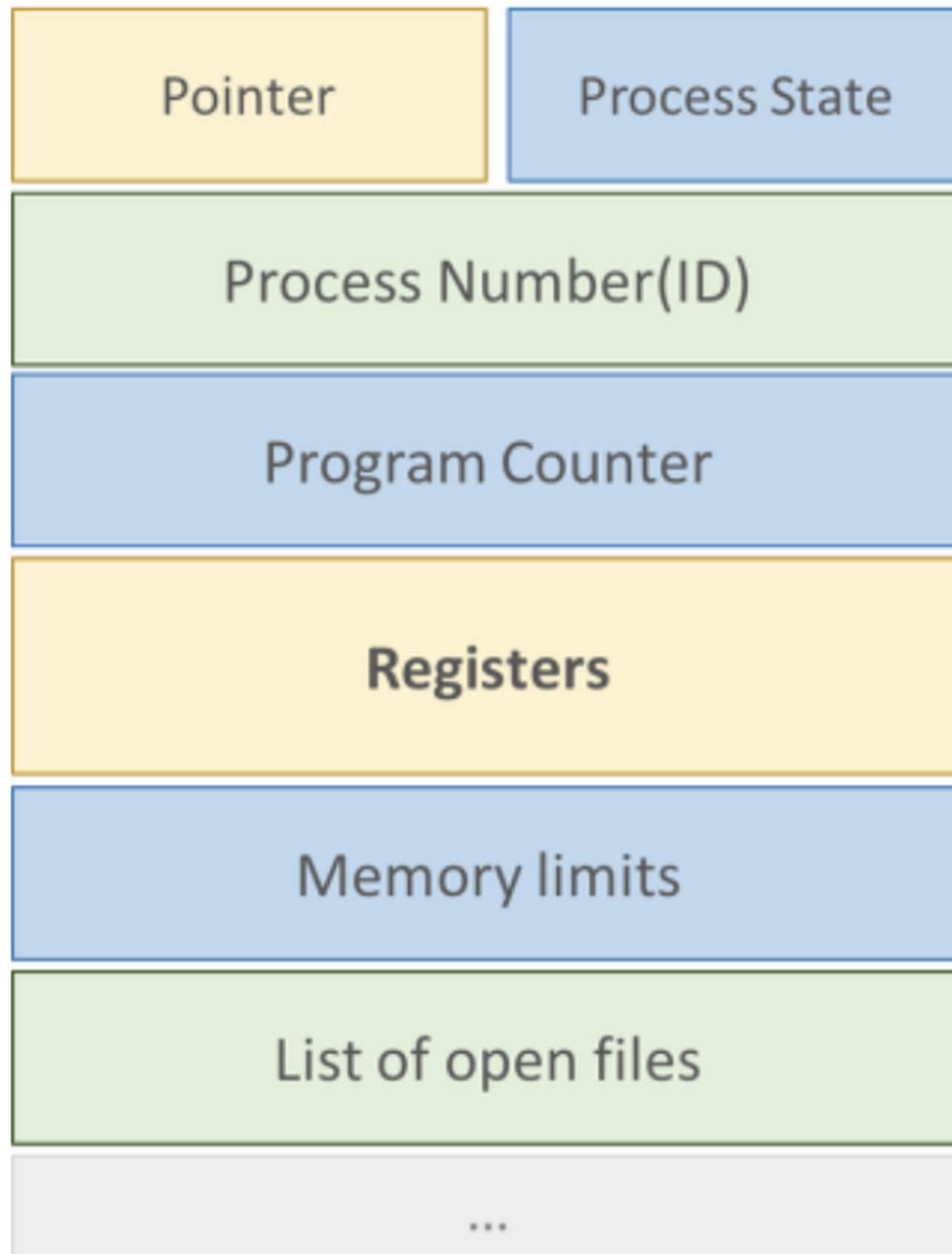
프로세스 제어 블록(PCB)



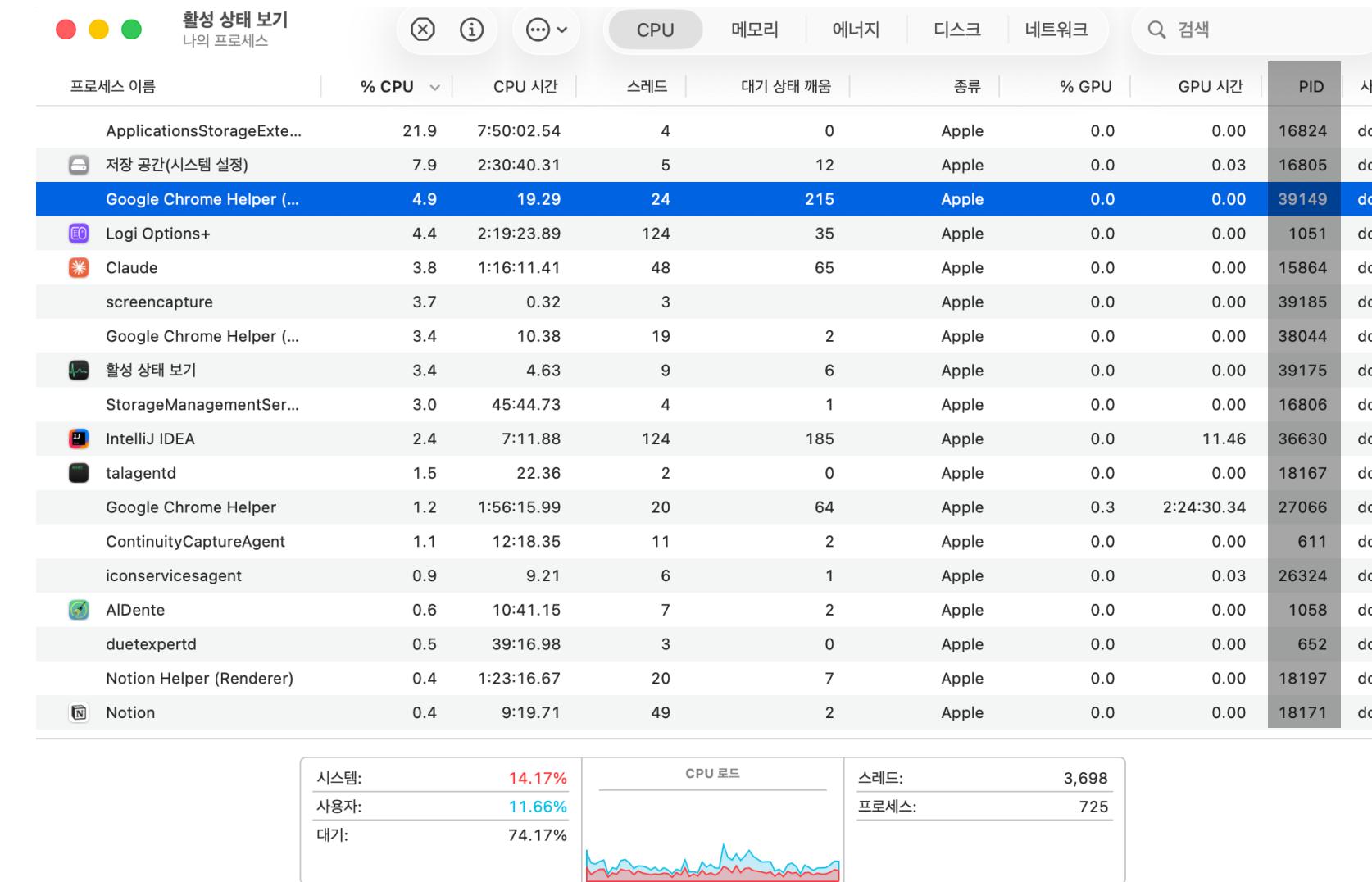
PCB는 프로세스와 관련된 정보를 저장하는 자료 구조
상품에 달린 태그가 간단한 예시
PCB는 커널 영역에 생성된다

프로세스 제어 블록(PCB)

Process Control Block

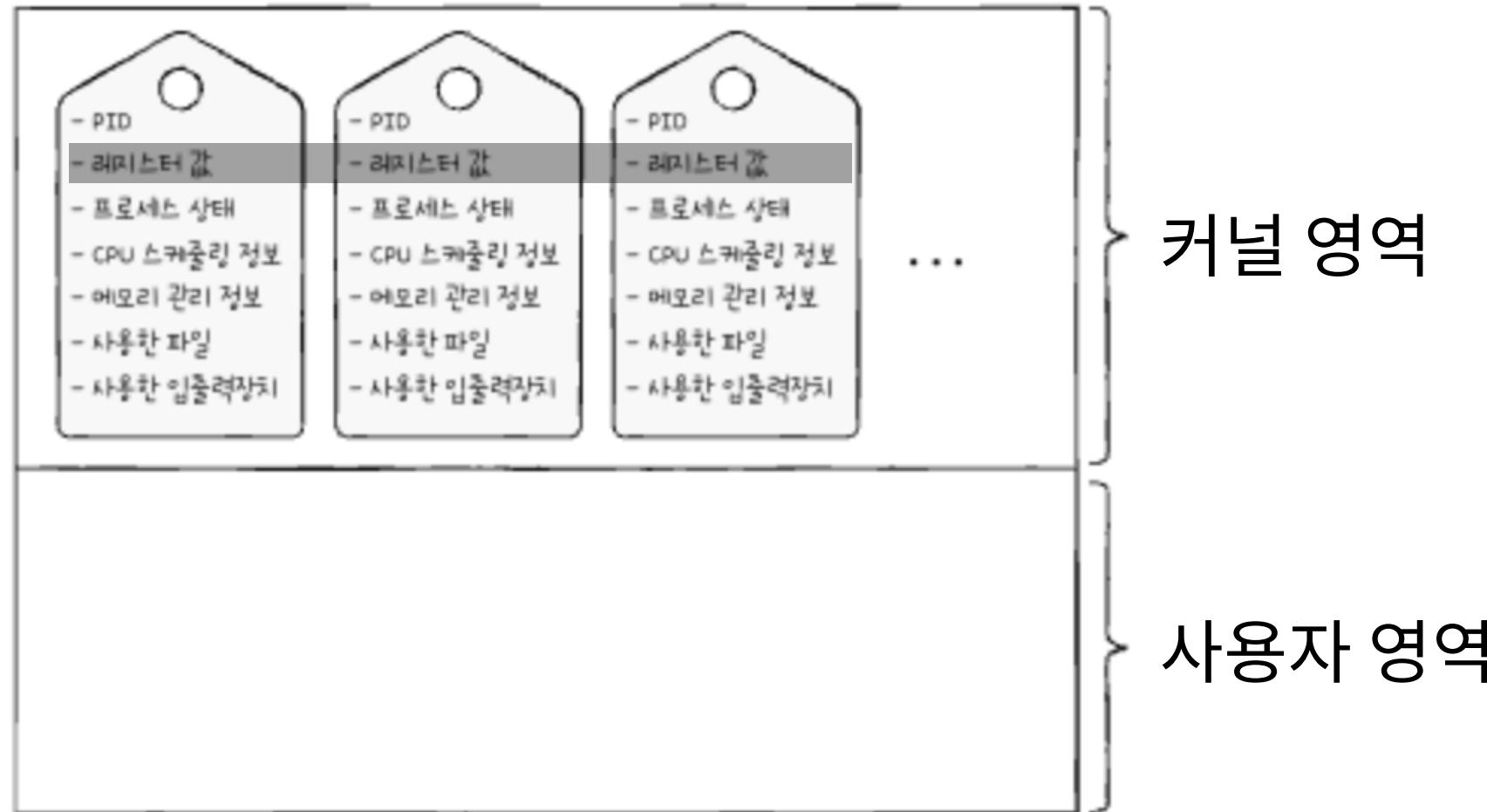


PID (Process ID)



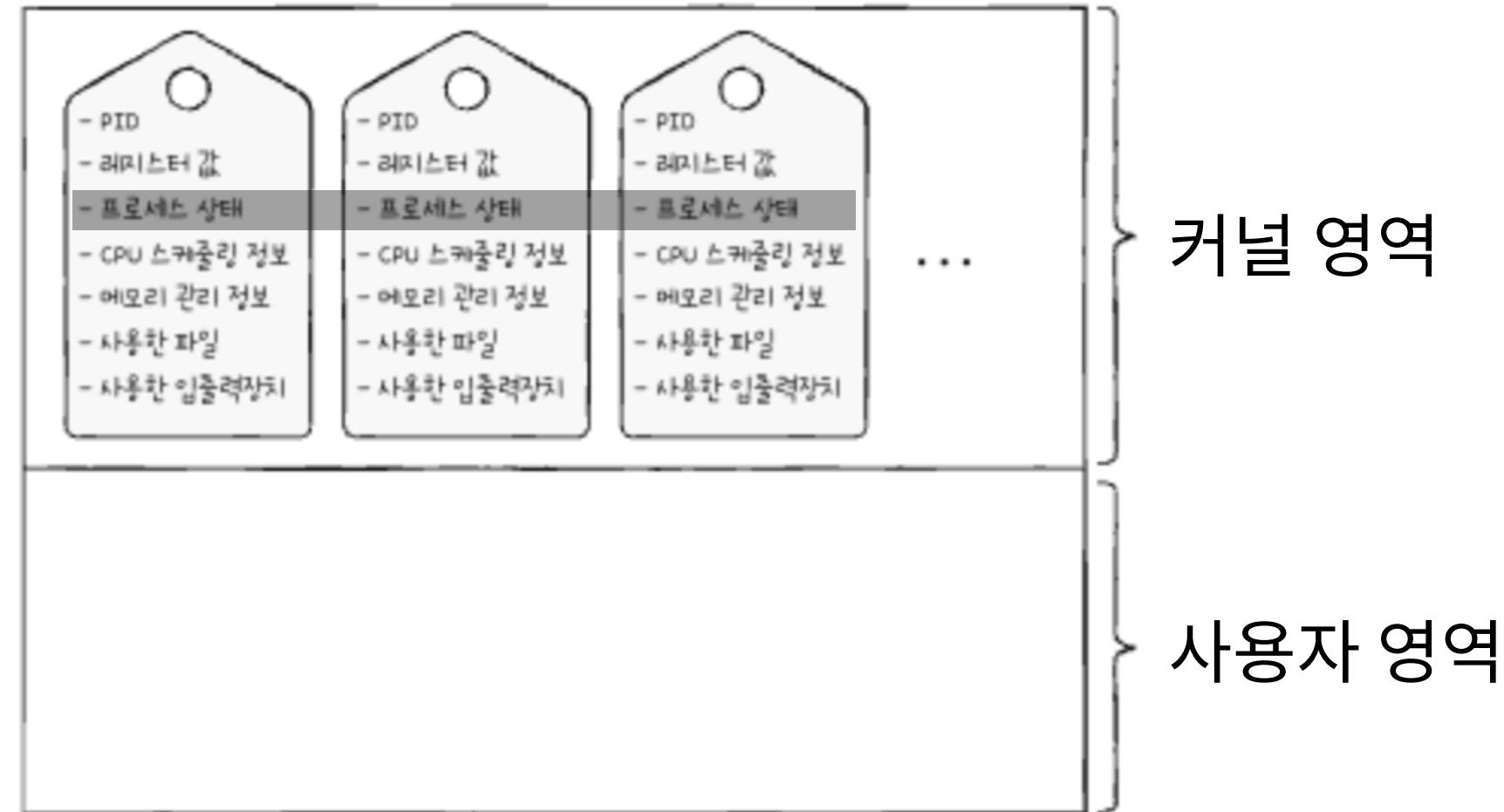
학번, 사번과 같이 프로세스를 식별하기 위한 고유 번호

레지스터 값



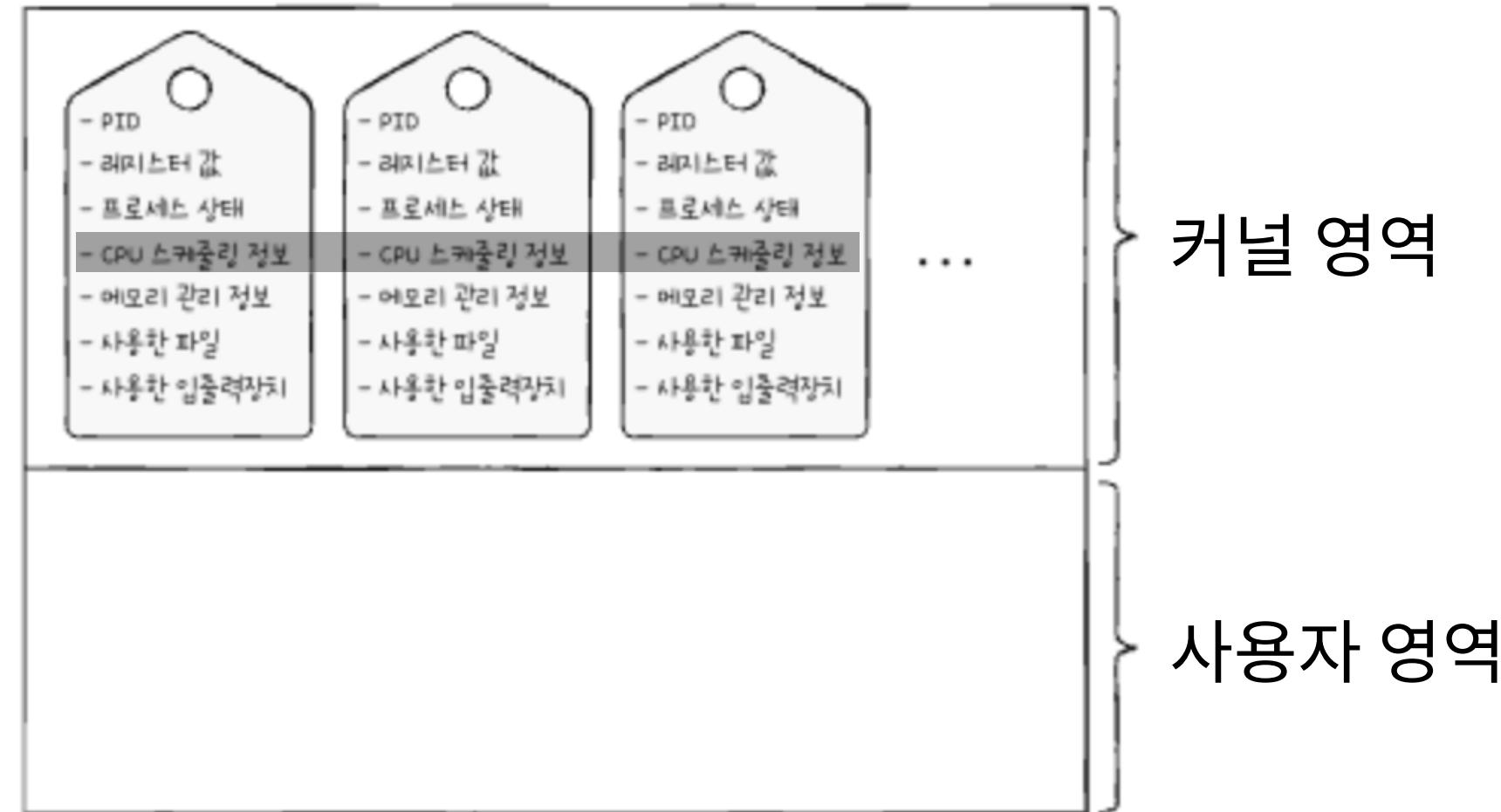
해당 프로세스가 실행하며 사용했던 프로그램 카운터를 비롯한 값을 저장
실행 차례 돌아올 때 레지스터 중간값들을 모두 복원

프로세스 상태



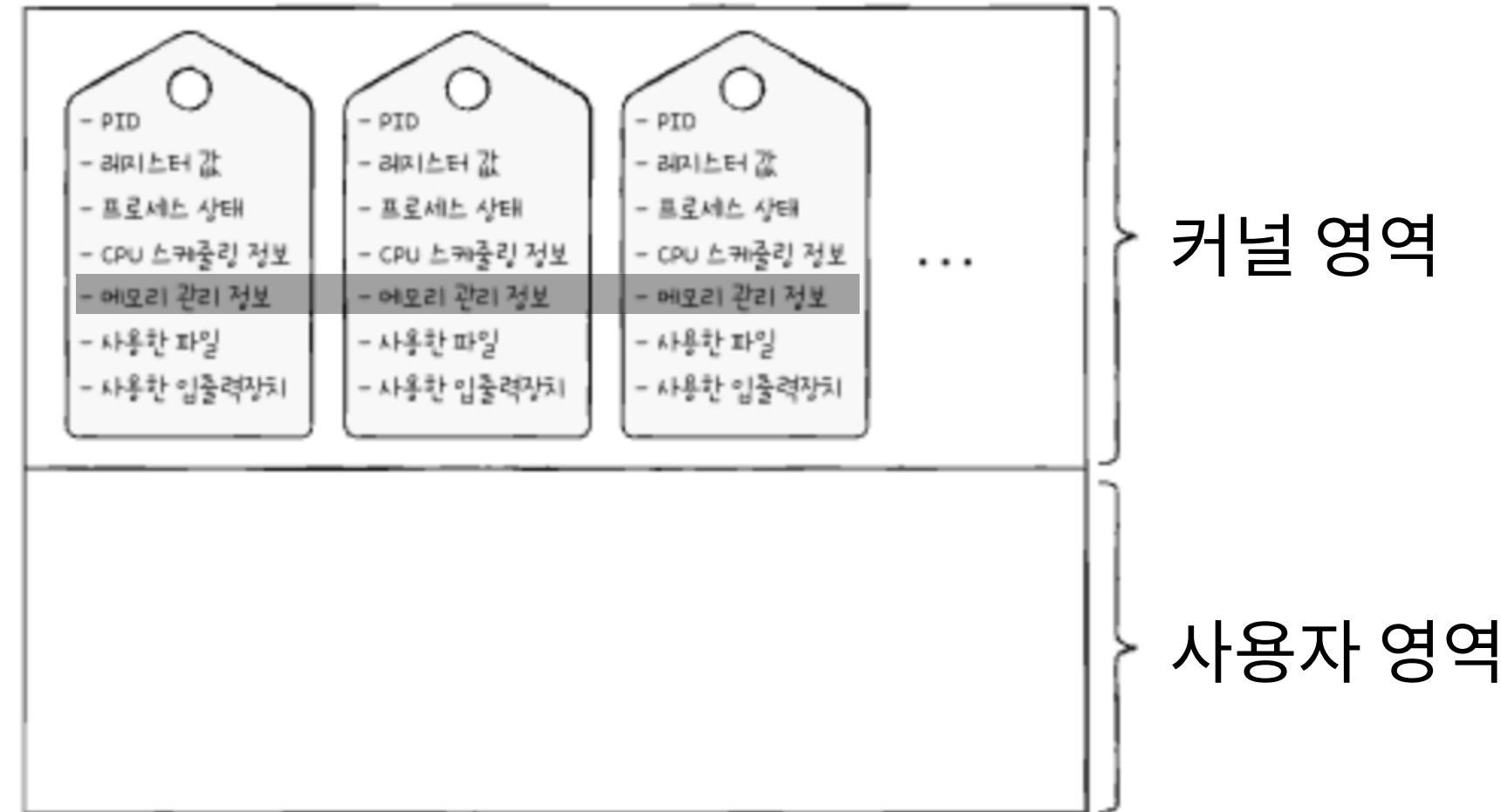
현재 프로세스가 어떤 상태인지 기록 (waiting, running 등)

CPU 스케줄링 정보



프로세스가 언제 어떤 순서로 CPU를 할당받을지에 대한 정보

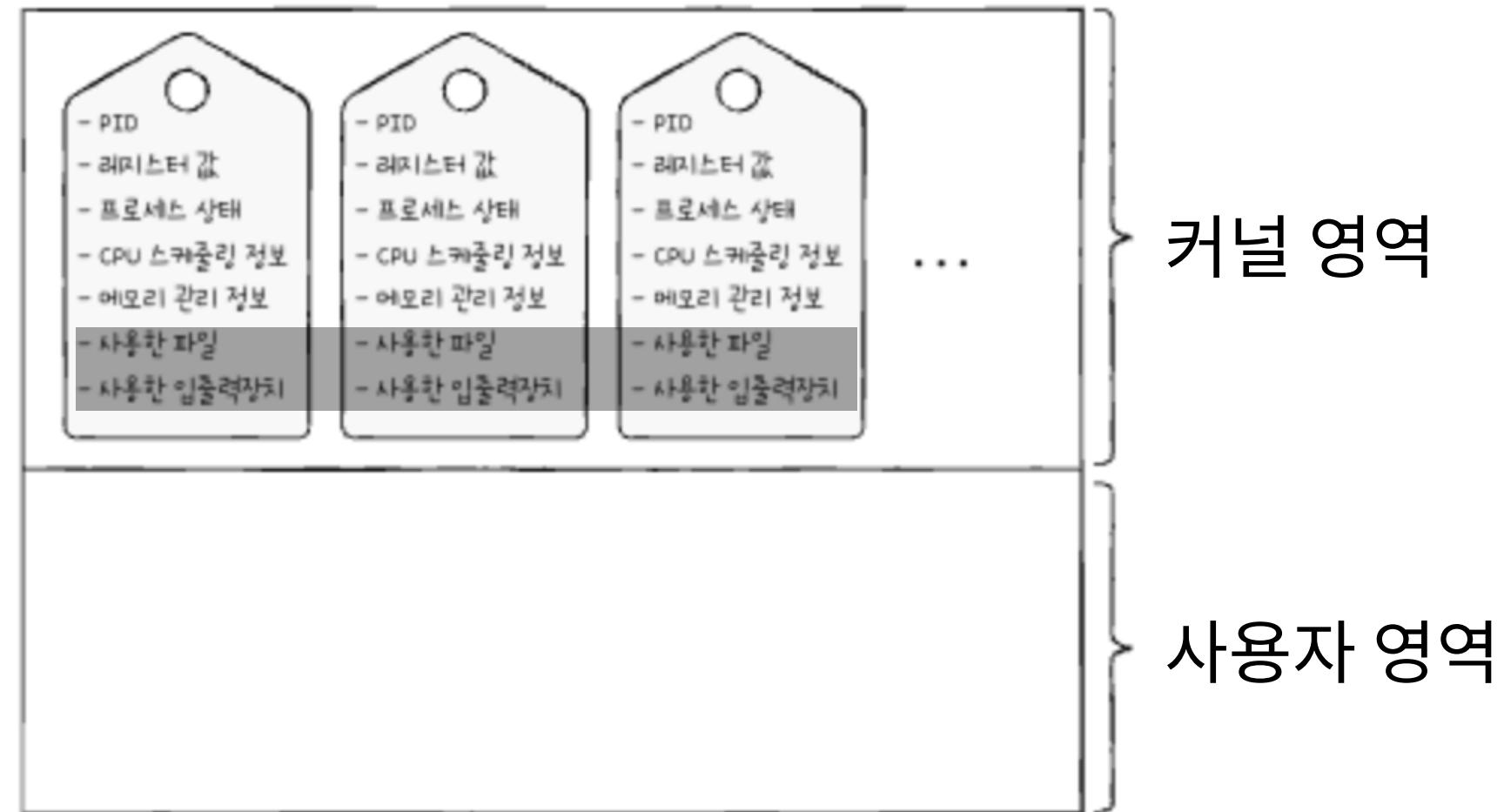
메모리 관리 정보



프로세스마다 메모리에 저장된 위치가 다르다.

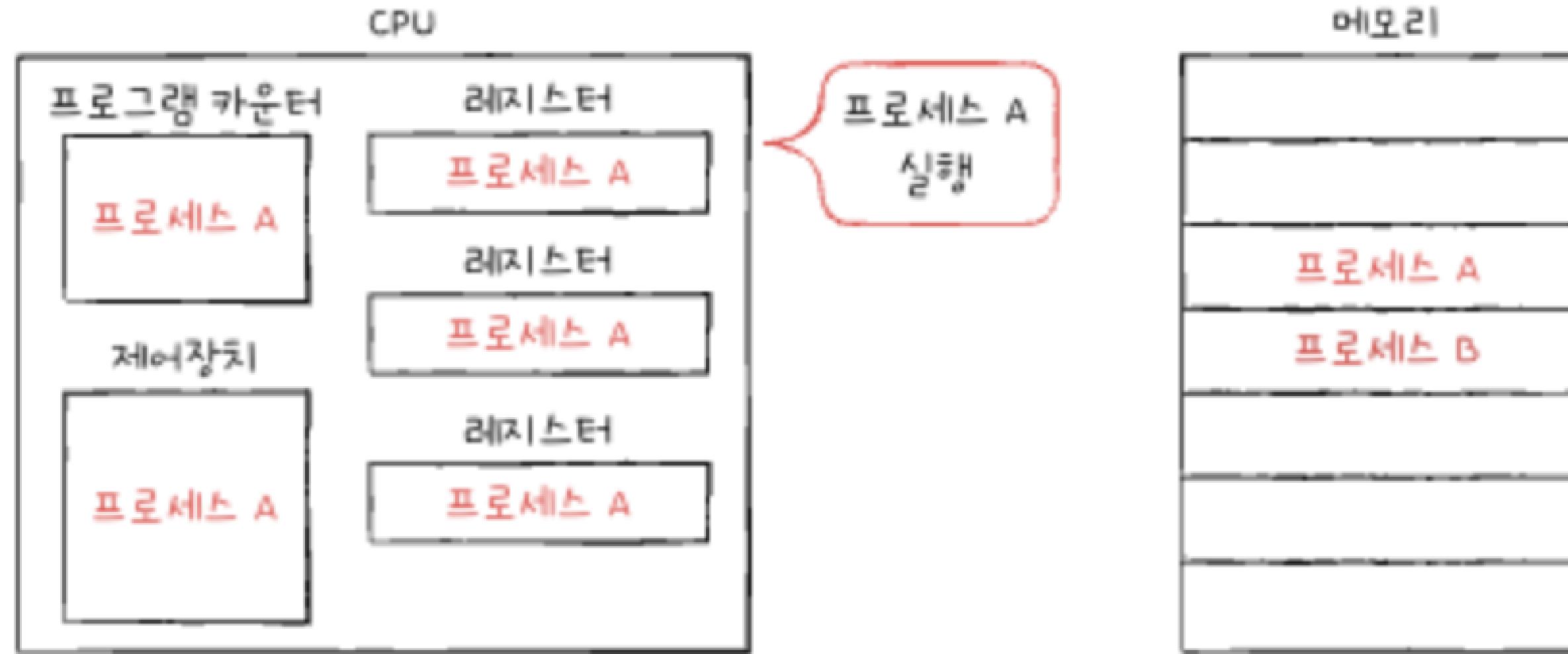
프로세스가 어느 주소에 저장되어 있는지에 대한 정보

사용한 파일과 입출력장치 목록



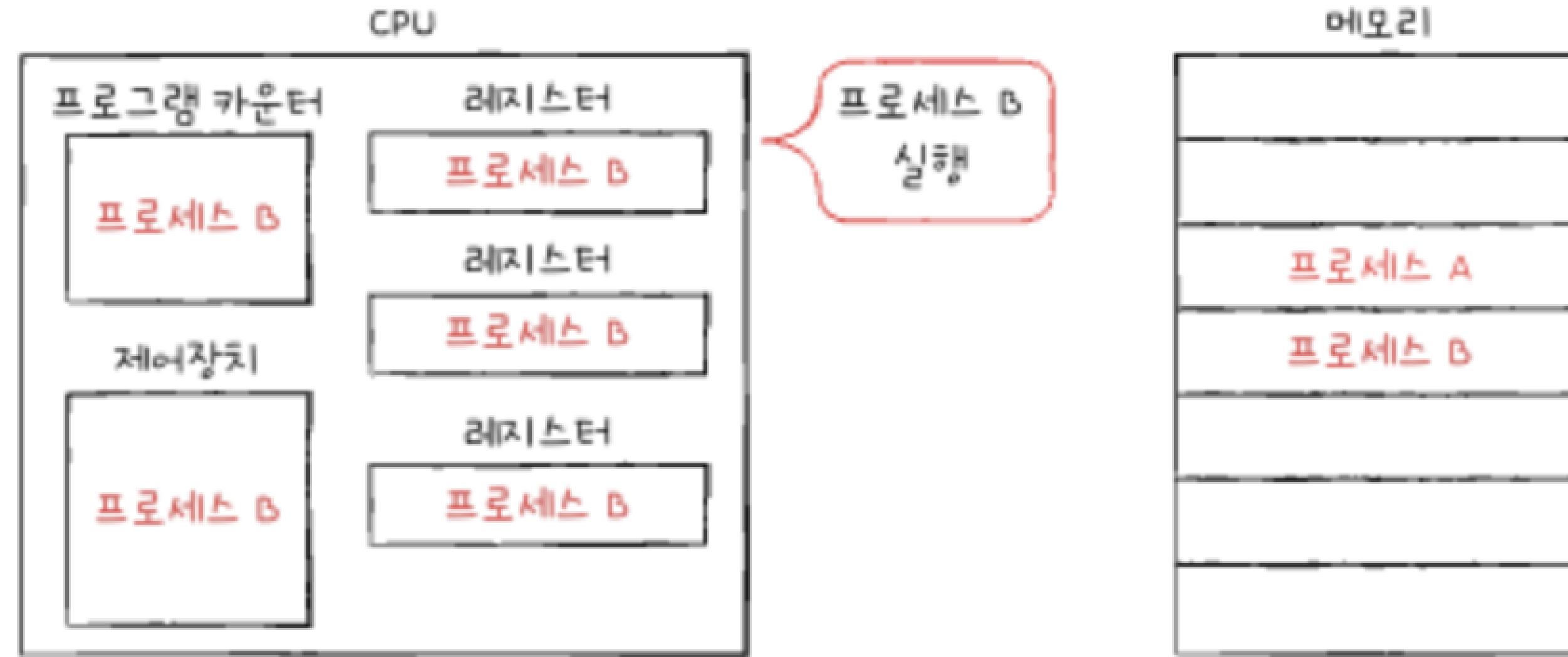
어떤 입출력장치가 프로세스에 할당되었는지,
어떤 파일들을 열었는지에 대한 정보들

문맥 교환



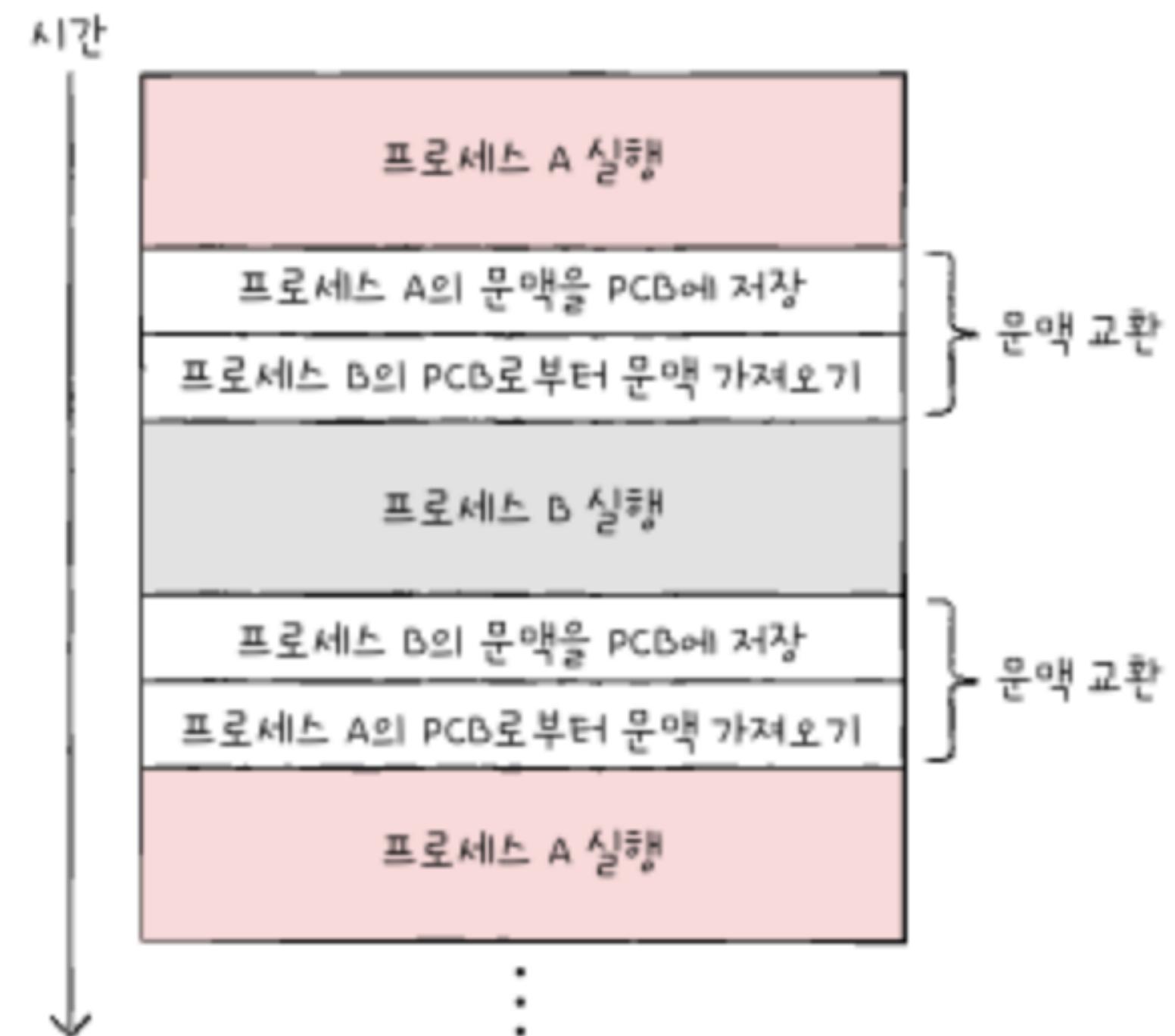
프로세스 수행을 재개하기 위해 기억해야 하는 정보
PCB에 기록되는 정보들이 문맥

문맥 교환

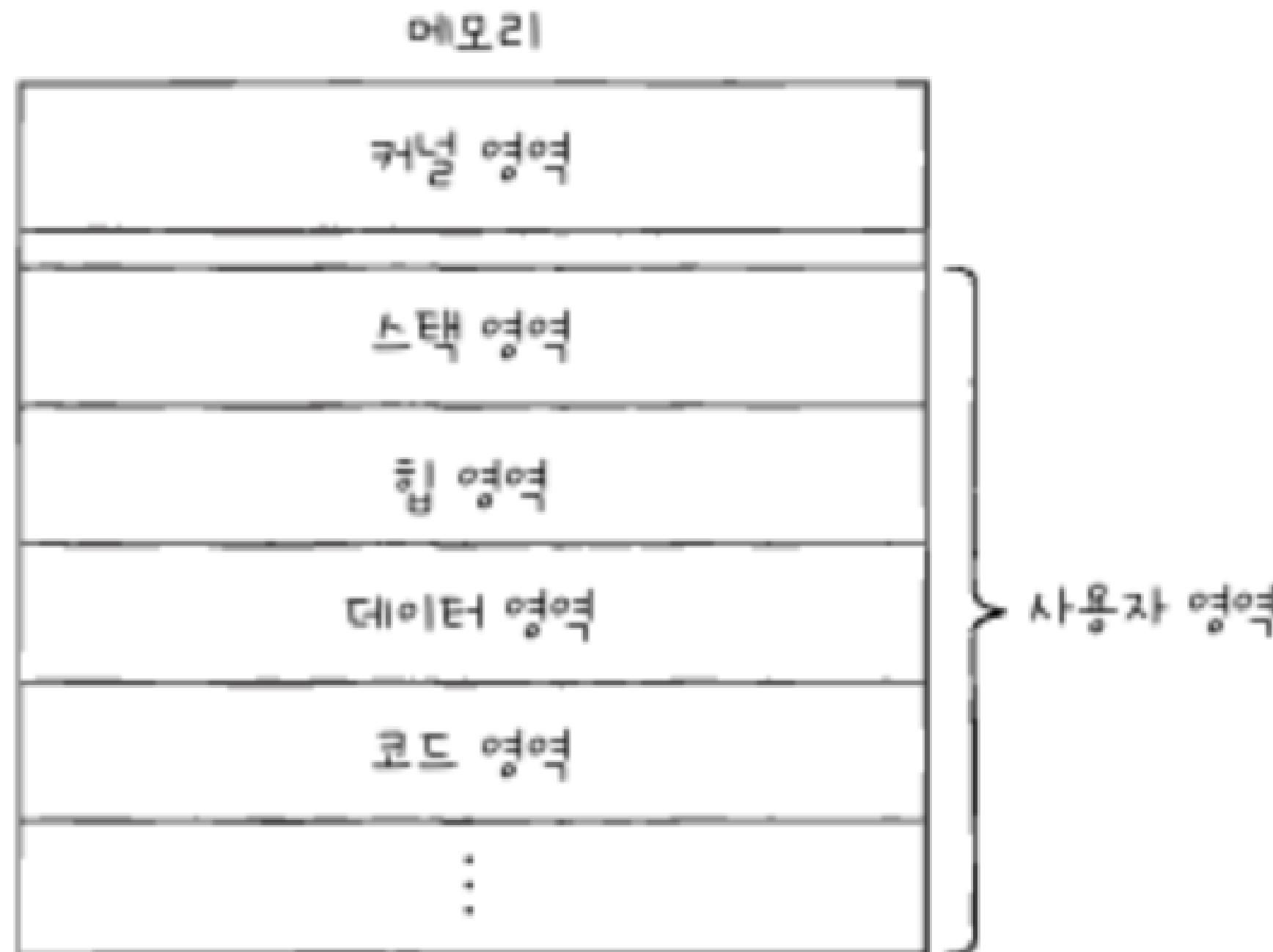


자원 수행 시간이 다 되거나, 인터럽트가 되면
기존 프로세스 문맥을 PCB에 저장

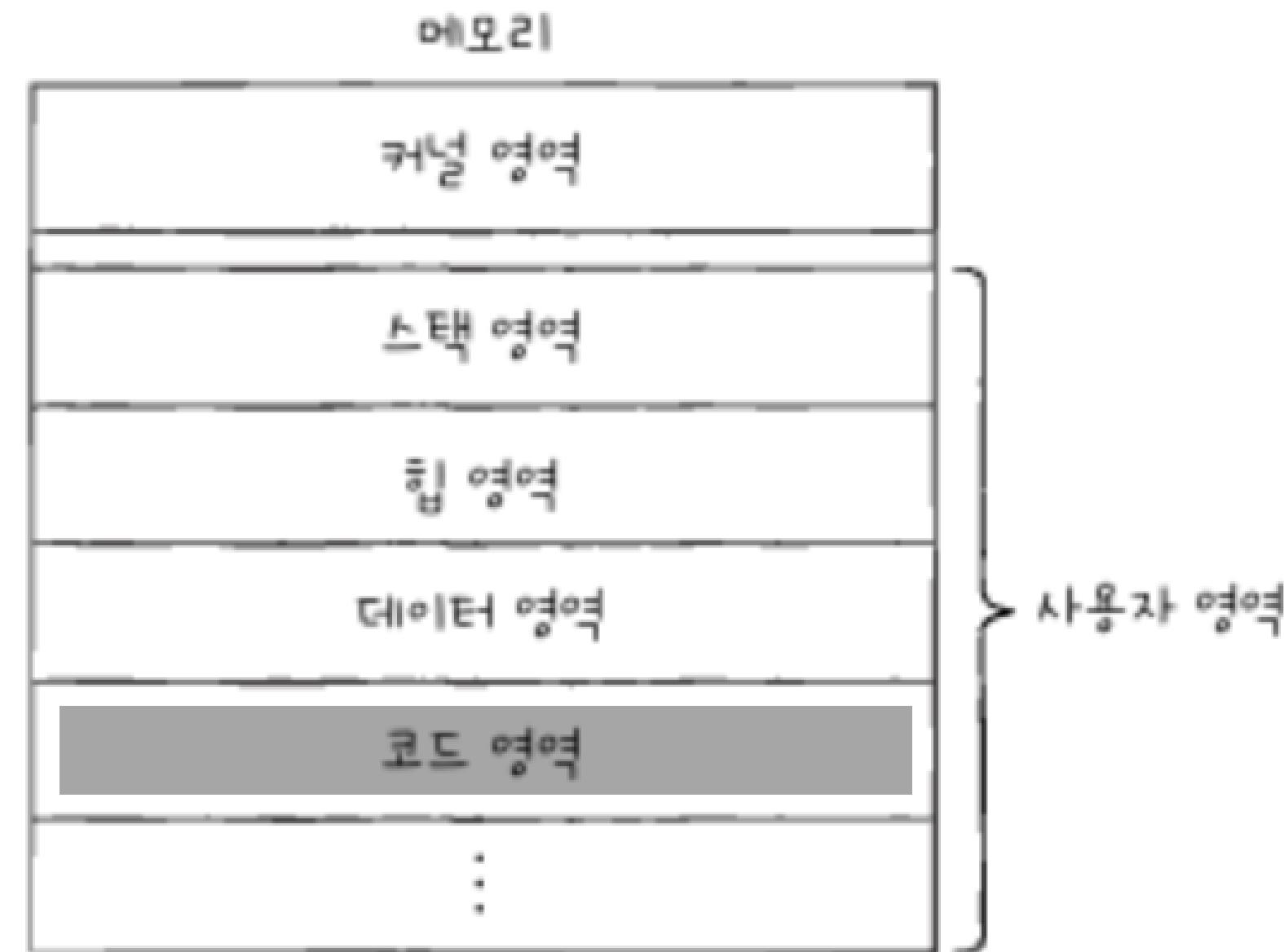
문맥 교환



프로세스의 메모리 영역

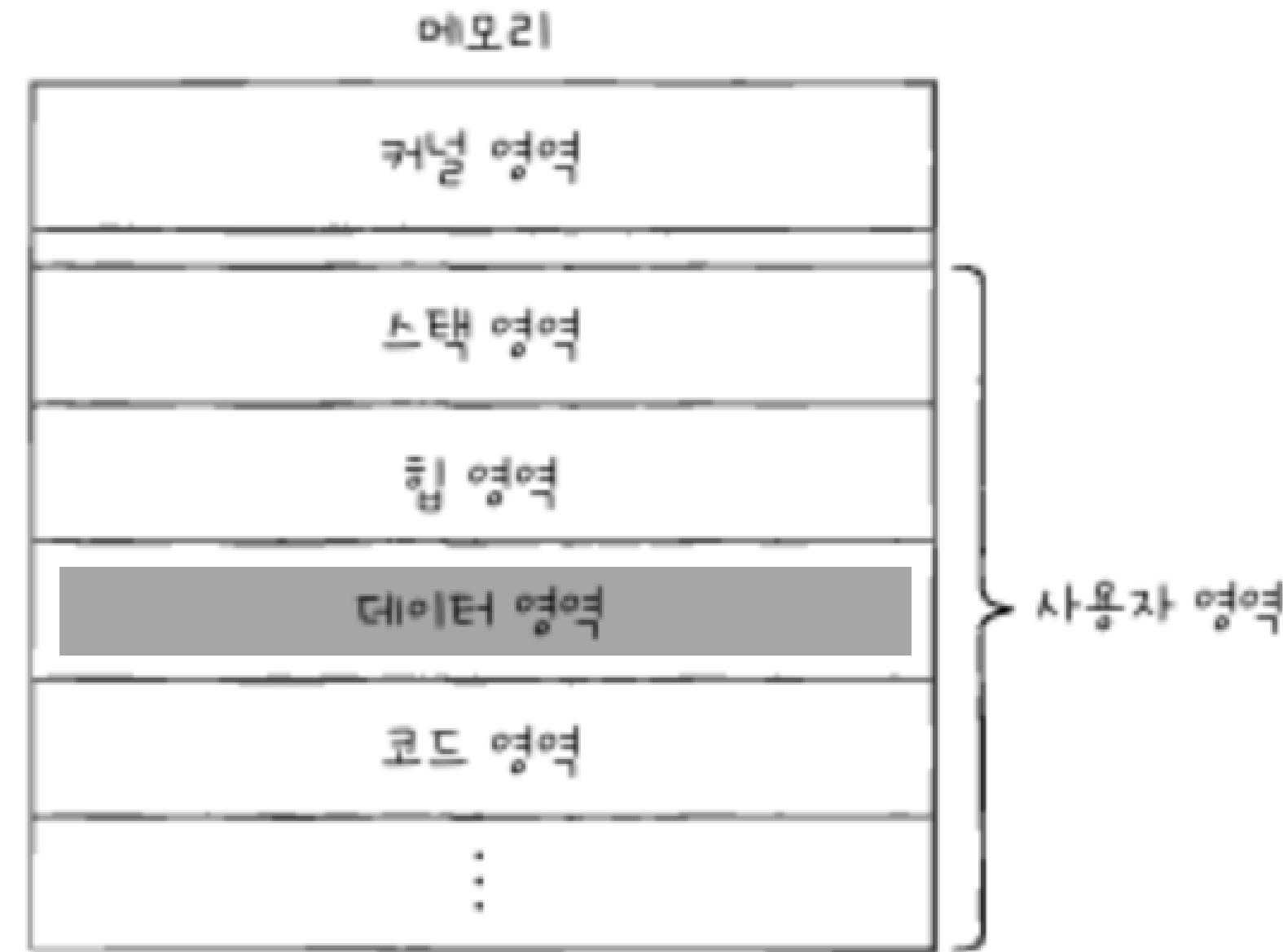


코드 영역



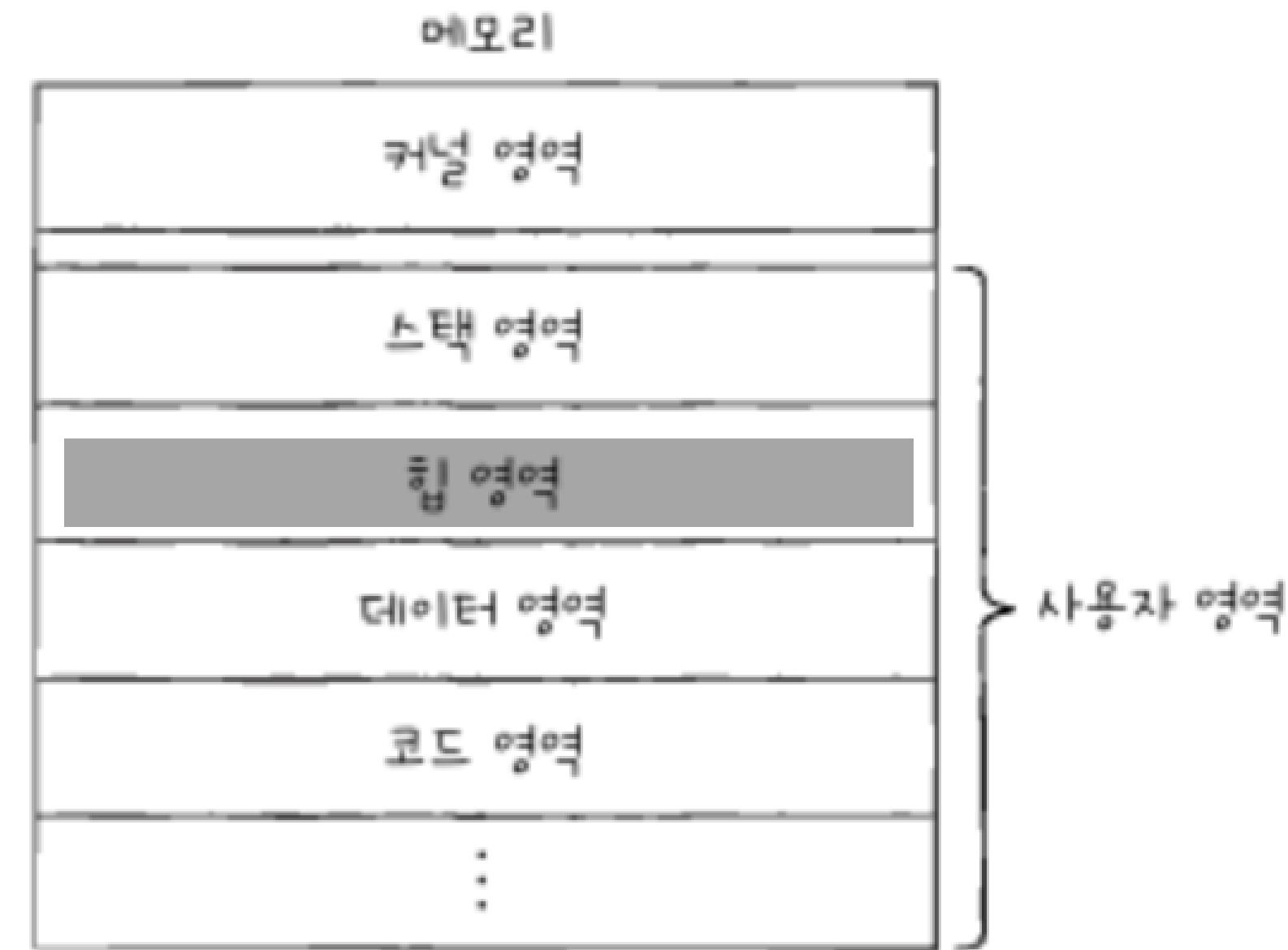
기계어로 이루어진 명령어가 저장
데이터가 아니라 CPU가 실행할 명령어가 담겨 있음
read-only

데이터 영역



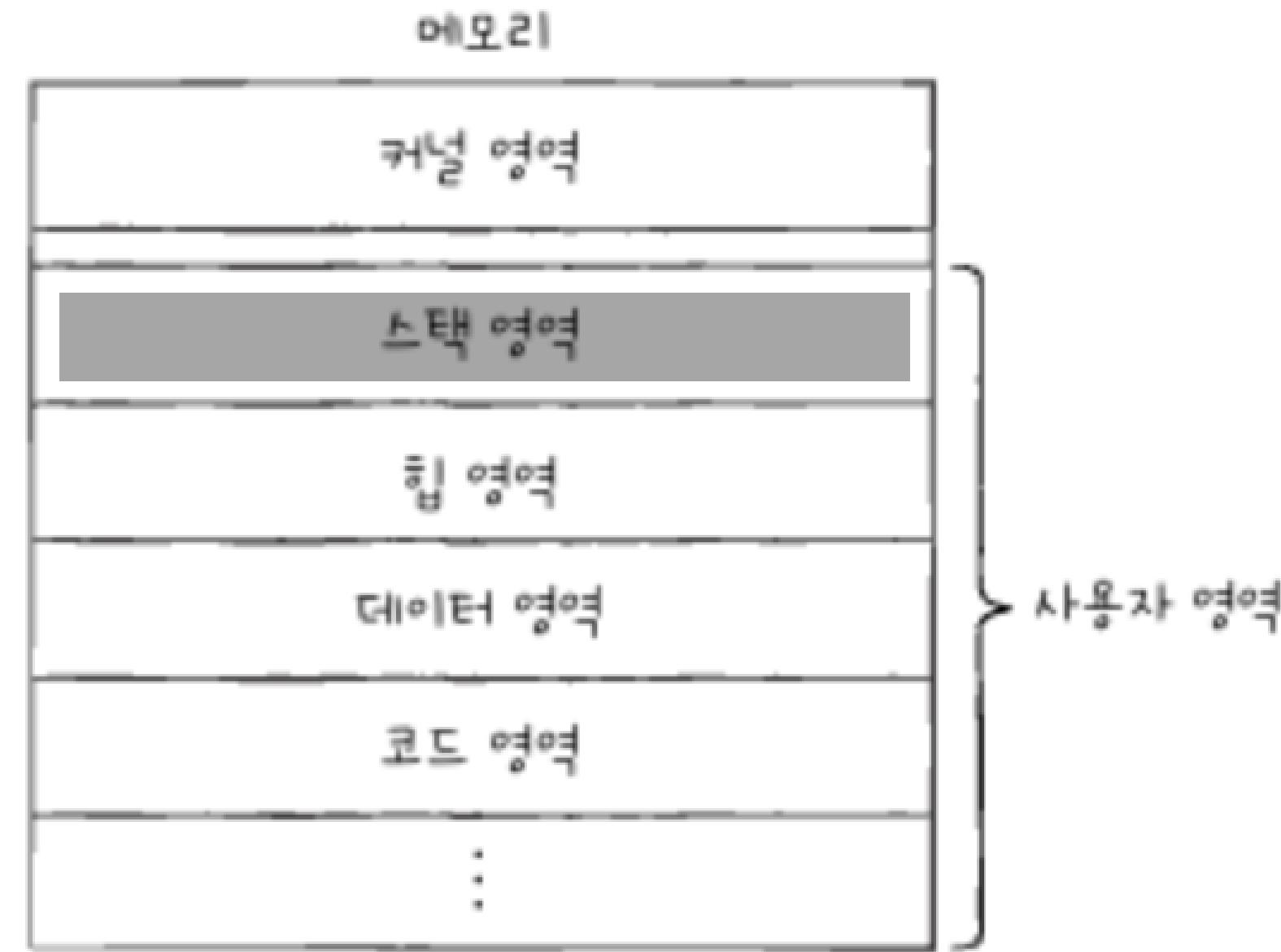
프로그램이 실행되는 동안 유지되는 데이터
전연 벡수가 대표적
코드, 데이터 영역은 크기가 변하지 않아 정적 할당 영역

힙 영역



프로그래머가 직접 할당할 수 있는 저장 공간
할당하고 꼭 반환 필요(메모리 누수 방지)

스택 영역

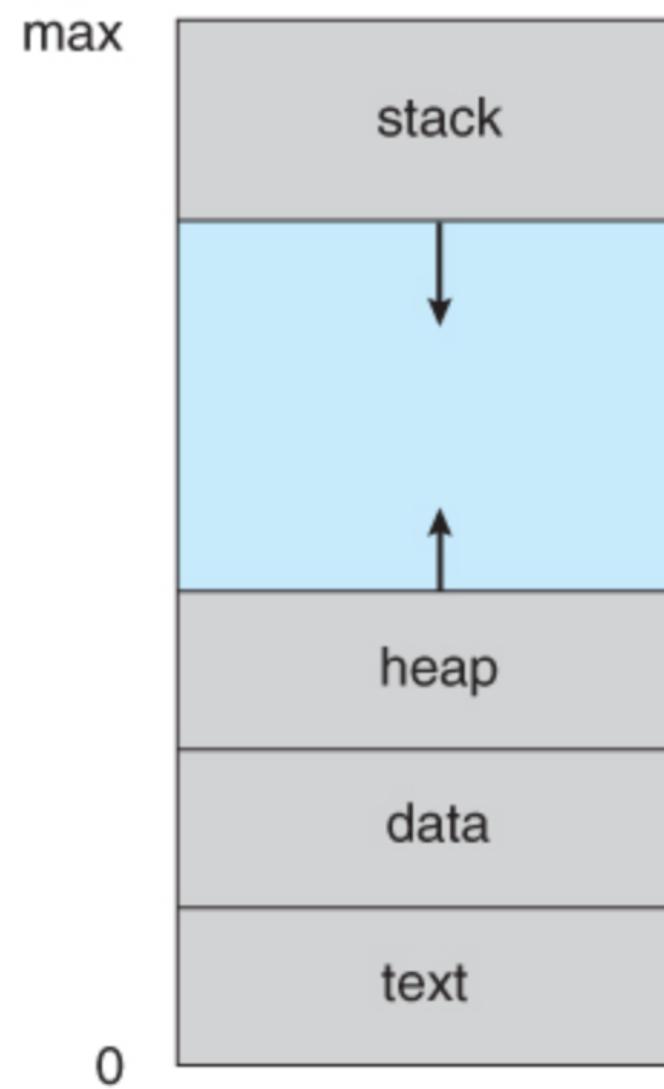


데이터를 일시적으로 저장하는 공간

매개 변수나 지역 변수가 대표적

힙, 스택 영역은 크기가 변할 수 있어 동적 할당 영역

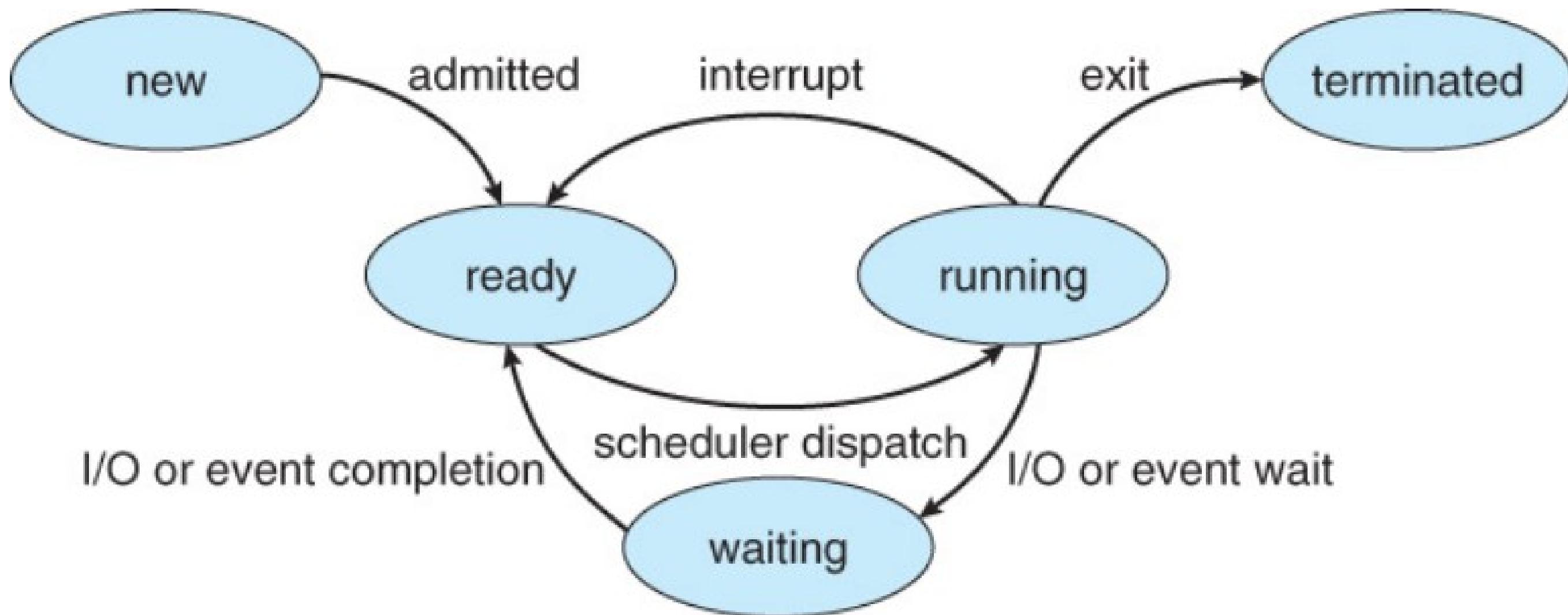
구조 정리



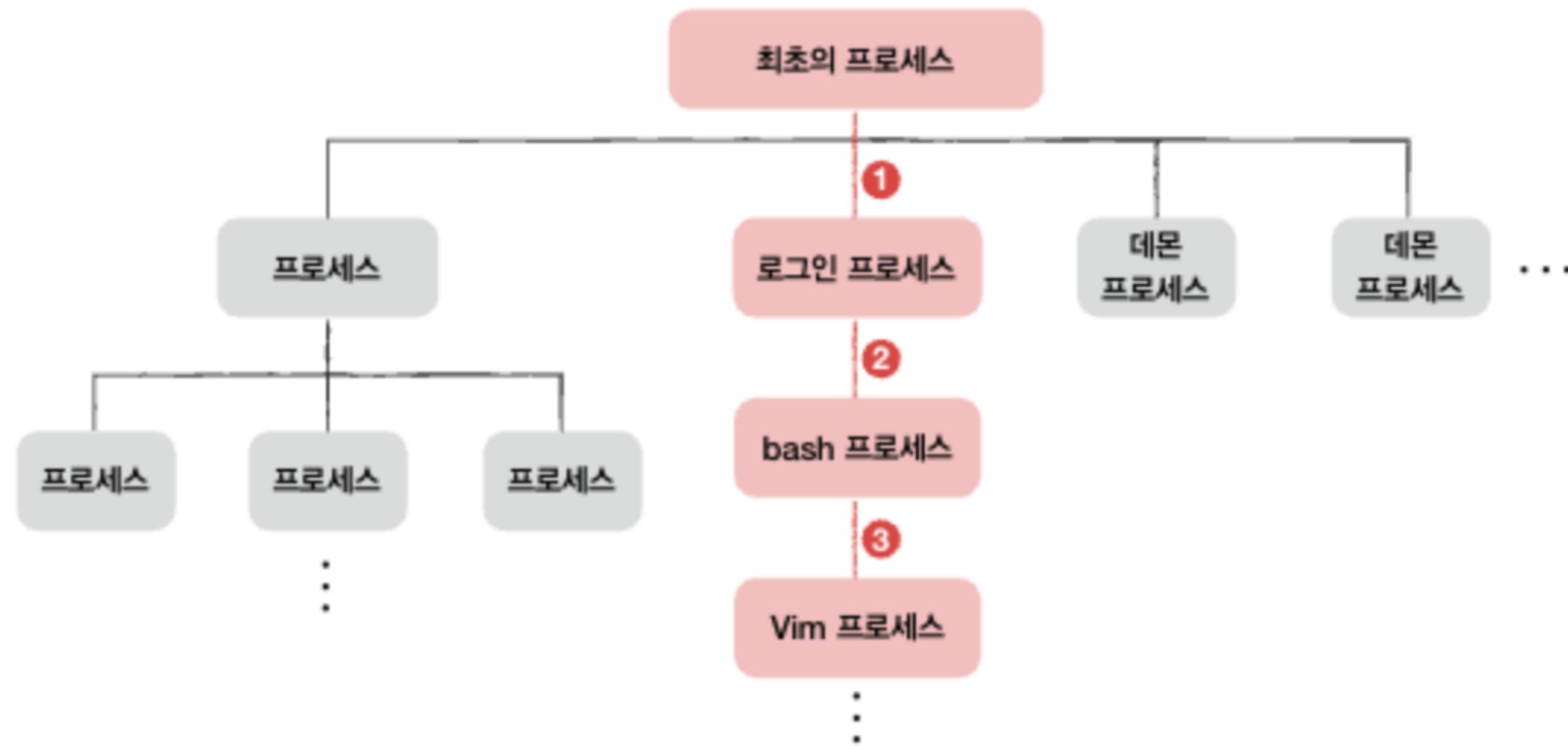
스택, 힙 영역은 충돌 방지를 위해 서로 반대부터 시작

10-2. 프로세스 상태와 계층 구조

프로세스 상태

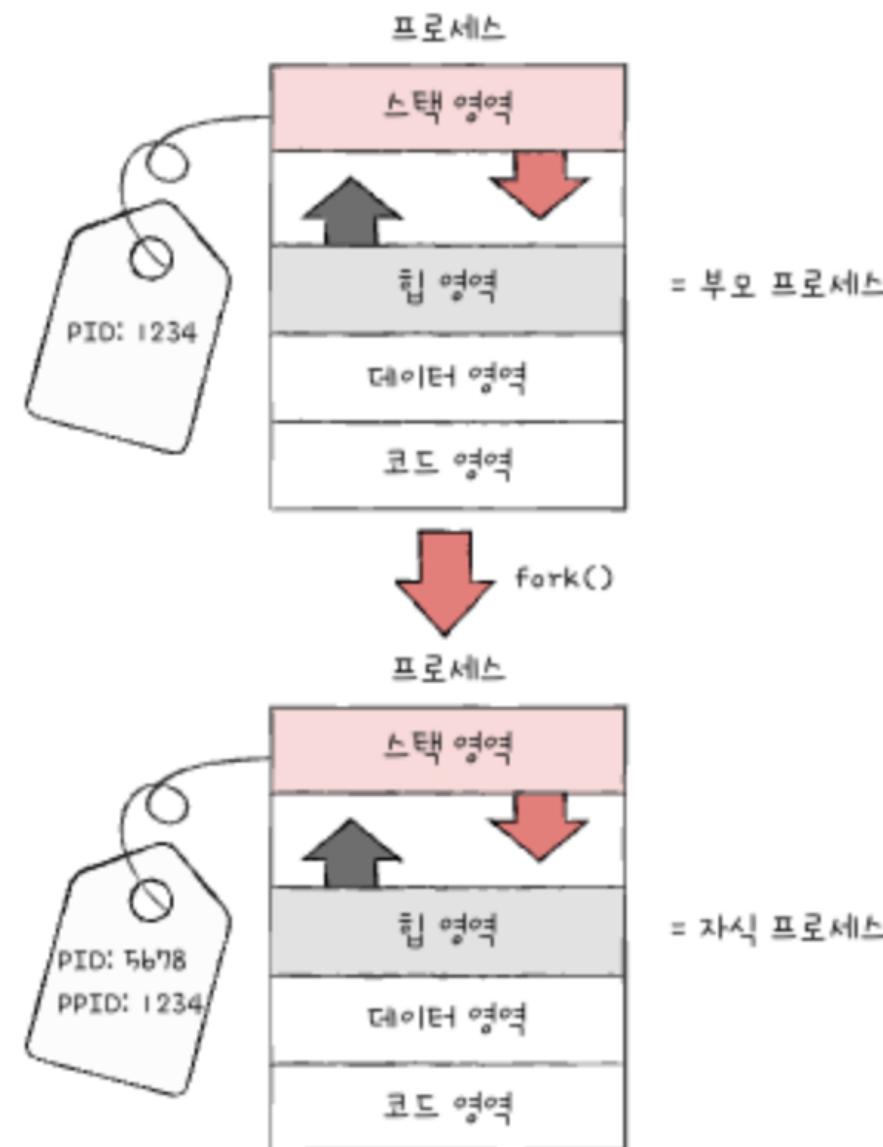


프로세스 계층 구조



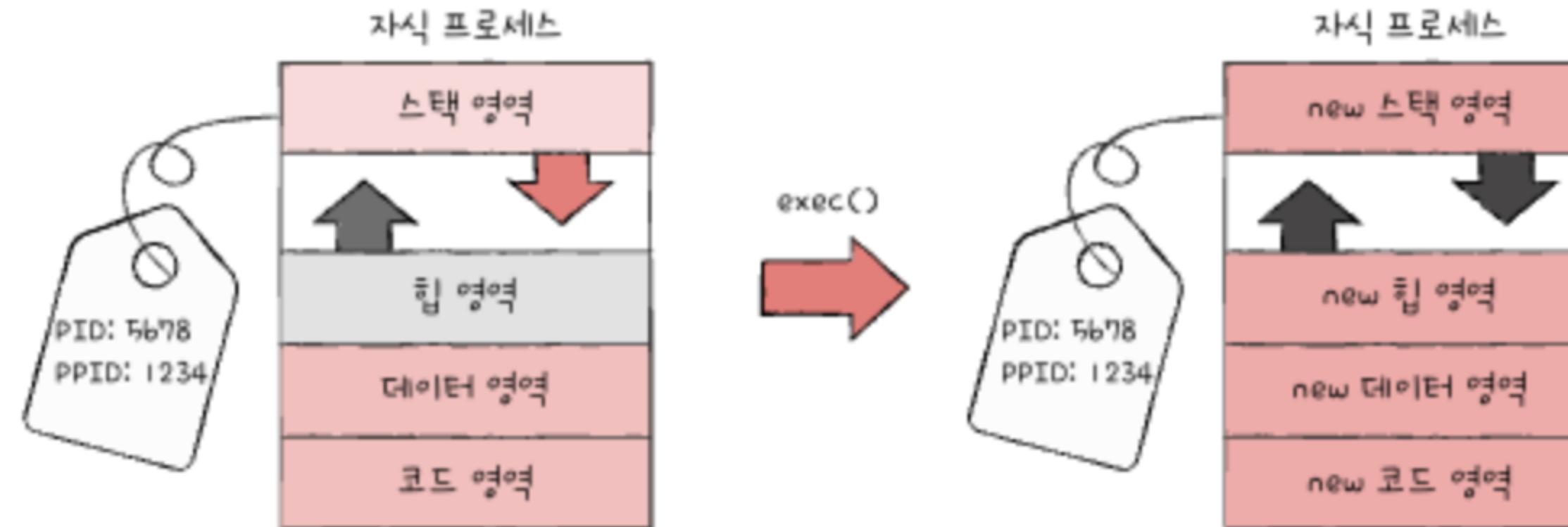
프로세스들이 자식 프로세스를 생성하며
트리와 같은 계층 구조를 띠게 되는 것

프로세스 생성 기법



부모 프로세스가 `fork()`를 하면
복사본을 자식 프로세스로 생성한다

프로세스 생성 기법



`exec` 호출을 통해 새로운 프로그램으로 전환

코드, 데이터 영역이 실행할 프로그램 내용으로 변경