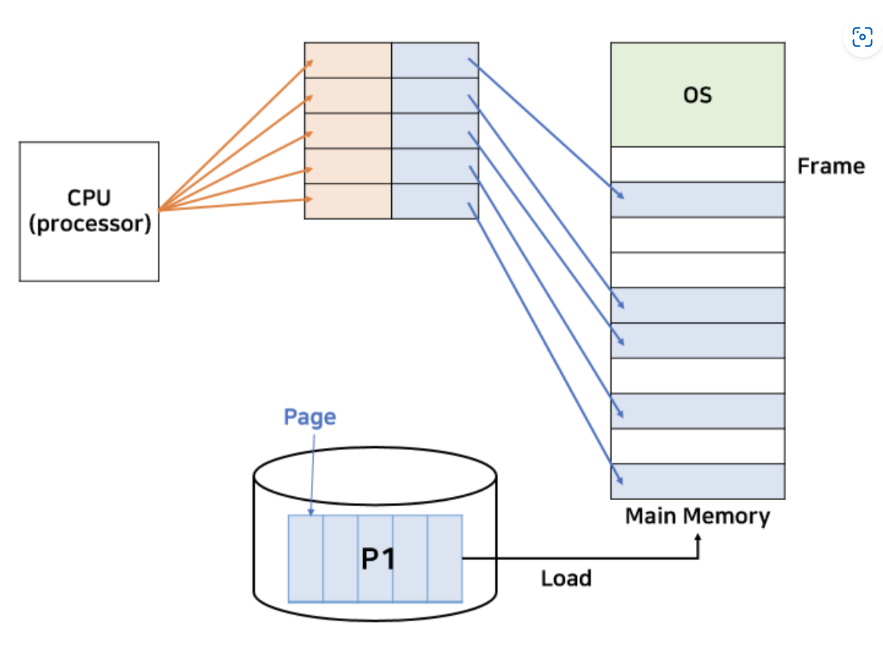
메모리 단편화

컴퓨터에서 어떤 프로그램을 실행할 때, 메모리의 공간을 연속적인 형태로 할당하여 사용하게 된다. 이렇메 프로그램이 메모리에 할당되고, 해제되고, 다시 새로운 프로그램이 할당되고, 해제되고를 반복하다보면 메모리 공간이 조각조각 나뉘게 되어 실제로는 사용가능한 메모리가 충분히 존재하지만 할당이 불가능한 상태가 발생하게 된다. 이를 메모리 단편화라고 한다.

이에 대한 해결 방법이 페이징과 세그먼테이션이다.

페이징



프로세스를 일정한 크기의 페이지로 분할해서 메모리에 적재하는 방식이다.

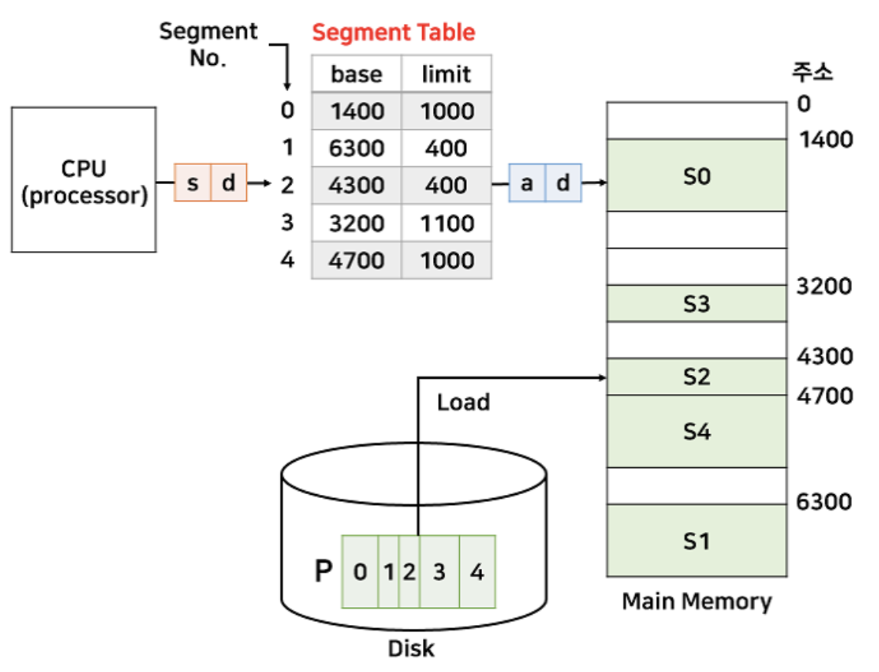
물리 메모리는 프레임으로 나누어서 각 프레임에 페이지를 할당한다.

페이지 테이블에는 각 페이지 번호 + 할당된 프레임의 시작 물리주소를 저장.

연속되어 저장될 필요가 없고, 남는 프레임에 적절하게 배치되기 때문에 외부 단편화가 생기지 않으나, 내부 단편화가 생길 수 있다. ex)페이지 단위가 100인데, 프로세스가 100단위로 떨어지지 않으면 내부 단편화가 생긴다.

페이지 단위를 작게 하면 해결이 가능하지만, 매핑 과정이 복잡해져서 비효율적이다.

세그먼테이션



세그먼테이션은 페이징과는 다르게, 같은 크기로 자르는 것이 아닌, 각각 맞는 크기로 분할해서 메모리에 적재하는 방식이다. -> 페이징과 같이 그냥 나누는 것이 아닌, 논리적 내용을 기반으로 나눠서 배치할 수 있다.

그래서 페이징과 반대로 내부 단편화의 문제가 해소되지만, 외부 단편화 문제가 생길 수 있다. ex)100크기의 세그먼트가 있던 곳에 90이 들어오면 10의 부분이 남을 것이고, 이것이 반복되면 외부 단편화 문제가 생길 것이다.

세그먼트 테이블은 세그먼트 번호와 시작 주소를 갖는다는 점은 페이징과 같지만, 세그먼트 크기라는 엔트리를 추가로 가진다.

CPU에서 해당 세그먼트 크기를 넘어서는 주소가 들어오면 인터럽트가 발생해서 해당 프로세스를 강제로 종료한다.