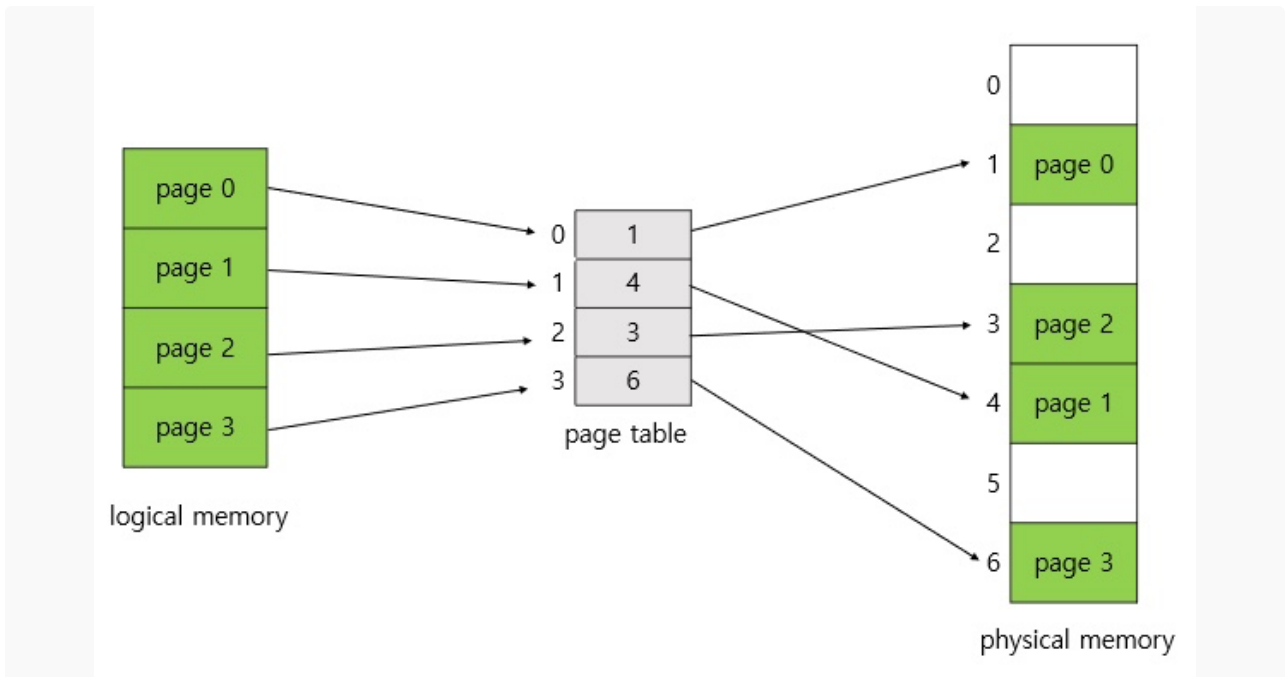


페이징 & 세그멘테이션

페이징(Paging)과 세그멘테이션(Segmentation)은 메모리 관리 기법 중 하나로, 가상 메모리를 물리적 메모리로 매핑하는 방식입니다.

페이징



프로세스를 고정 크기의 페이지(Page)로 분할하여, 물리적 메모리에 적재합니다. 이 때, 페이지 테이블(Page Table)을 사용하여 가상 주소와 물리 주소의 매핑 정보를 저장합니다. 프로세스가 필요한 페이지를 요청할 때마다, 페이지 테이블을 참조하여 해당 페이지의 물리 주소를 찾아 접근합니다.

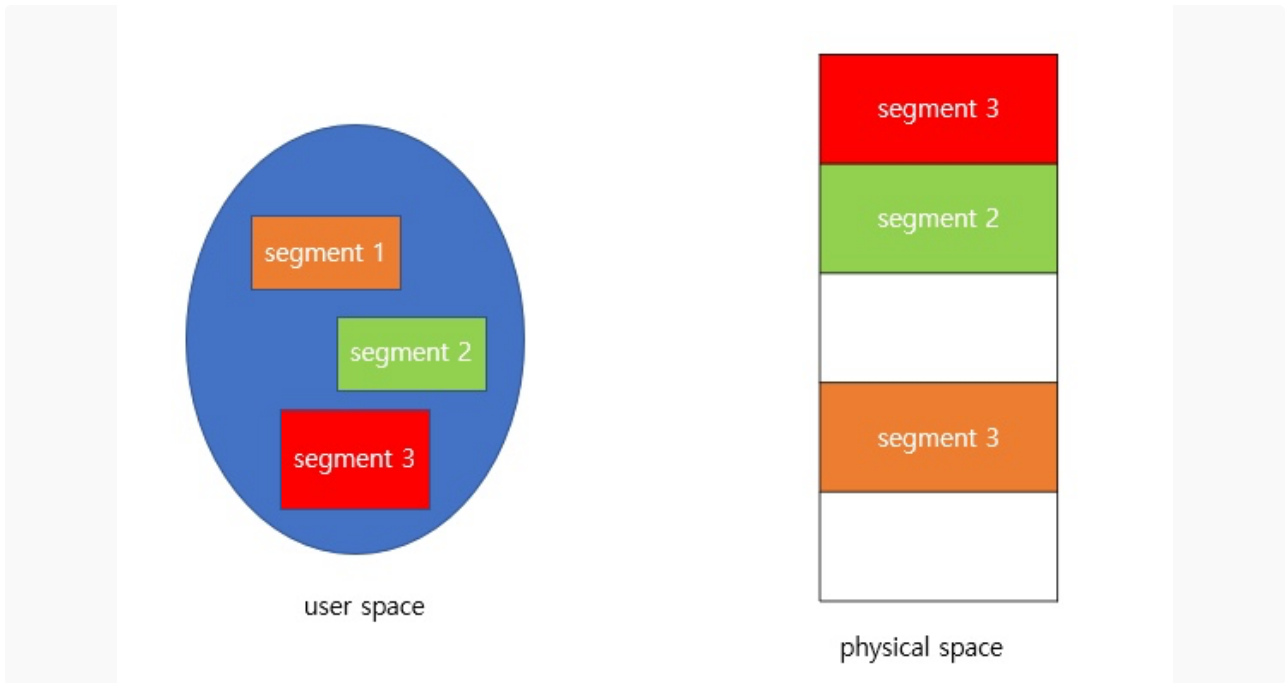
🚀 장점

- 논리적 주소 공간을 동일한 크기의 페이지로 분할하기 때문에 메모리 공간을 효율적으로 사용할 수 있습니다.
- 내부 단편화를 해결할 수 있습니다.

💔 단점

- 페이지 크기를 정해야 하기 때문에, 페이지의 크기가 작으면 페이지 테이블의 크기가 커지게 되고, 큰 페이지를 사용하면 내부 단편화 문제가 발생할 수 있습니다.
- 페이지 테이블이 메모리 상에 존재해야 하기 때문에, 페이지 테이블의 접근 시간이 추가됩니다.

세이멘테이션



프로세스를 논리적 단위인 세그먼트(Segment)로 분할하여, 물리적 메모리에 적재합니다. 이 때, 세그먼트 테이블(Segment Table)을 사용하여 가상 주소와 물리 주소의 매핑 정보를 저장합니다. 각 세그먼트는 크기가 가변적이며, 논리적 단위로 분할된 코드, 데이터, 스택 등이 있을 수 있습니다.

🚀 장점

- 논리적 주소 공간을 세그먼트로 분할하기 때문에, 프로세스의 논리적 구조를 자연스럽게 반영할 수 있습니다.
- 내부 단편화를 해결할 수 있습니다.

💔 단점

- 세그먼트의 크기가 다양하면, 외부 단편화 문제가 발생할 수 있습니다.
- 세그먼트 테이블을 유지하기 위해 추가적인 메모리 공간이 필요합니다.