

메모리 관리 3

페이지 테이블 낭비 문제의 해결책

1. 역페이지 테이블
2. 멀티 레벨 페이지 테이블
3. 2-level 페이지 테이블

Inverted Page Table (역 페이지 테이블)

- 시스템에 하나의 역 페이지 테이블만 둔다.
- 역 페이지 테이블 항목
 - 프로세스 번호 pid
 - 페이지 번호 p
- 역 페이지 테이블 인덱스
 - 프레임 번호
- 역 페이지 테이블 사용 시 논리 주소 형식
 - [프로세스 번호, 페이지 번호, 오프셋] 형식
- 역 페이지 테이블을 사용한 주소 변환
 - 논리 주소에서 (프로세스 번호, 페이지 번호)로 역 페이지 테이블 검색
 - 일치하는 항목 발견 시 항목 번호가 바로 프레임번호가 된다.
 - 프레임 번호와 오프셋을 연결하면 물리주소가 됨
- 역 페이지 테이블은 시스템에 1개 존재한다.
- 역 페이지 테이블의 크기
 - 항목 크기 → [프로세스 번호, 페이지 번호]로 구성 → 4byte + 4byte → 8byte
 - 항목의 수 = 물리 메모리 크기 / 프레임 크기

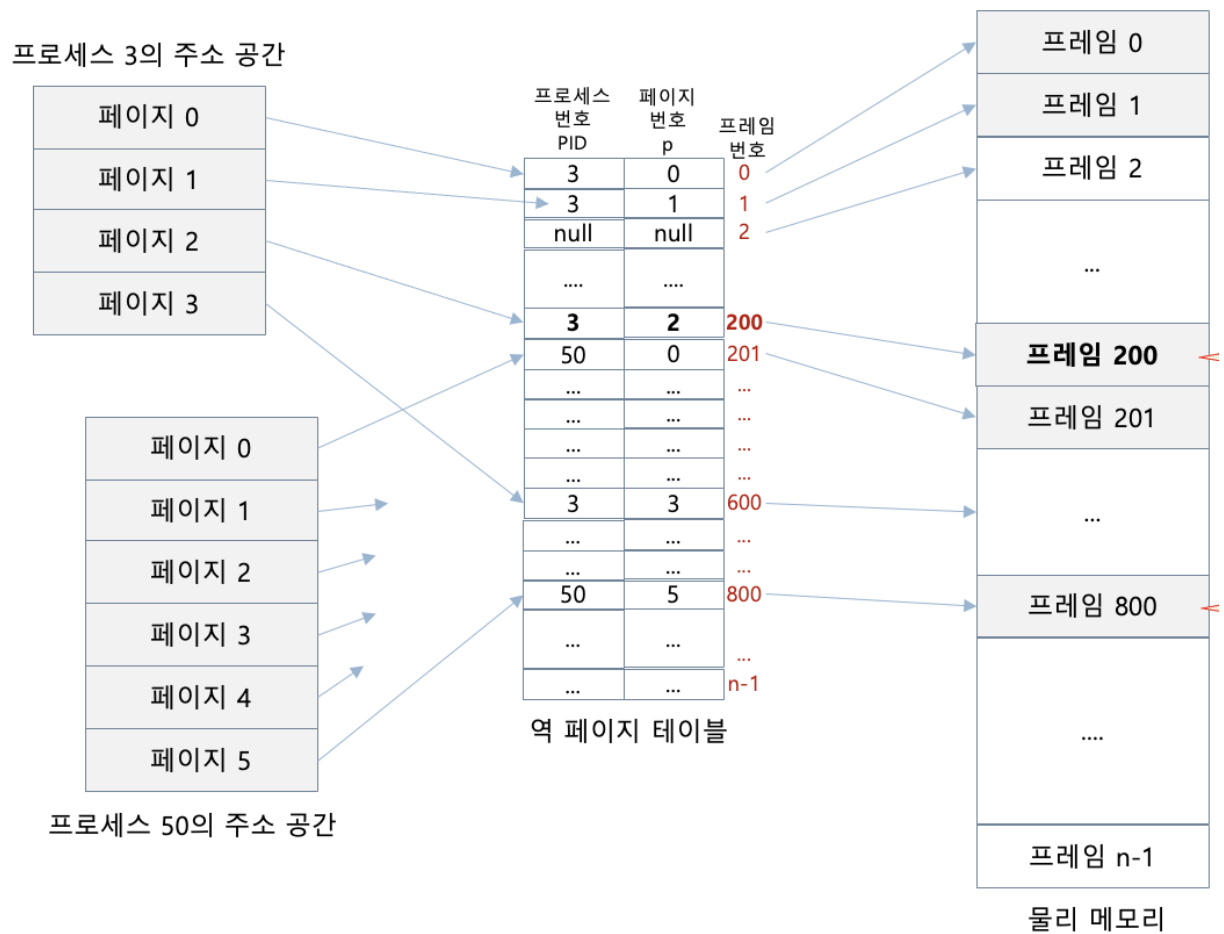


예시

물리메모리가 4GB, 프레임 크기가 4KB 상황

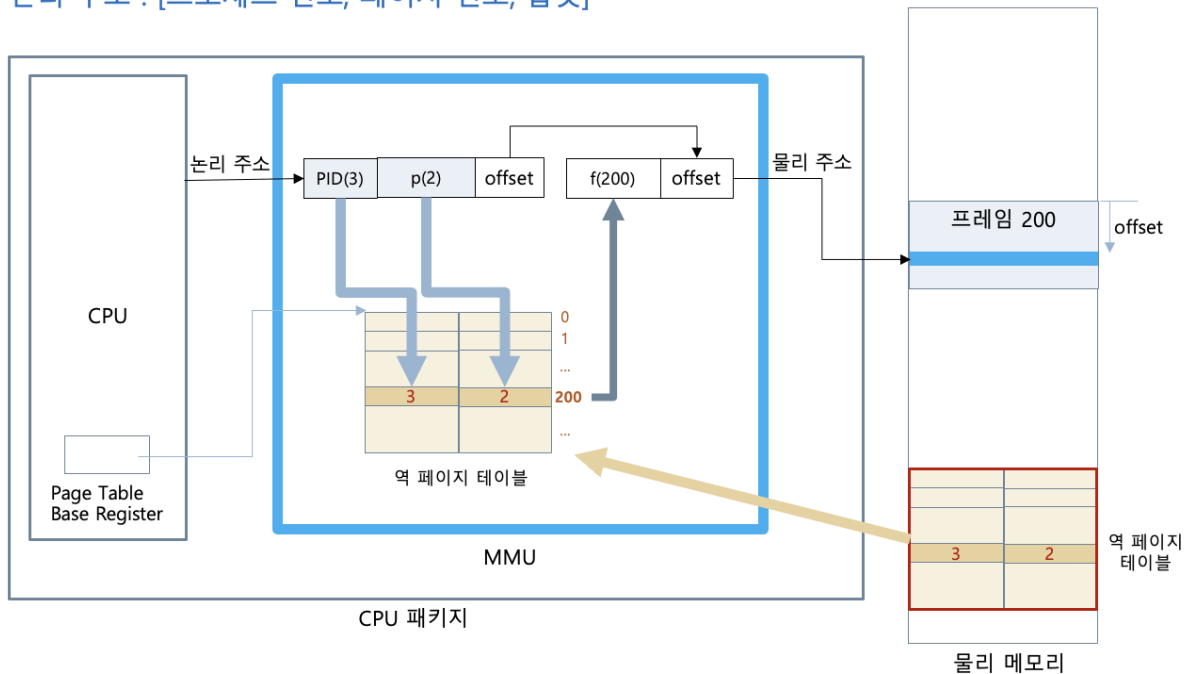
역 페이지 테이블 항목 수 = 4GB / 4KB = 2^{20} 개 = 약 100만개

- 역 페이지 테이블의 크기는 설치된 물리메모리 크기에 따라 달라진다.



역 페이지 테이블을 이용한 페이지와 프레임 사이의 관계

논리 주소 : [프로세스 번호, 페이지 번호, 오프셋]

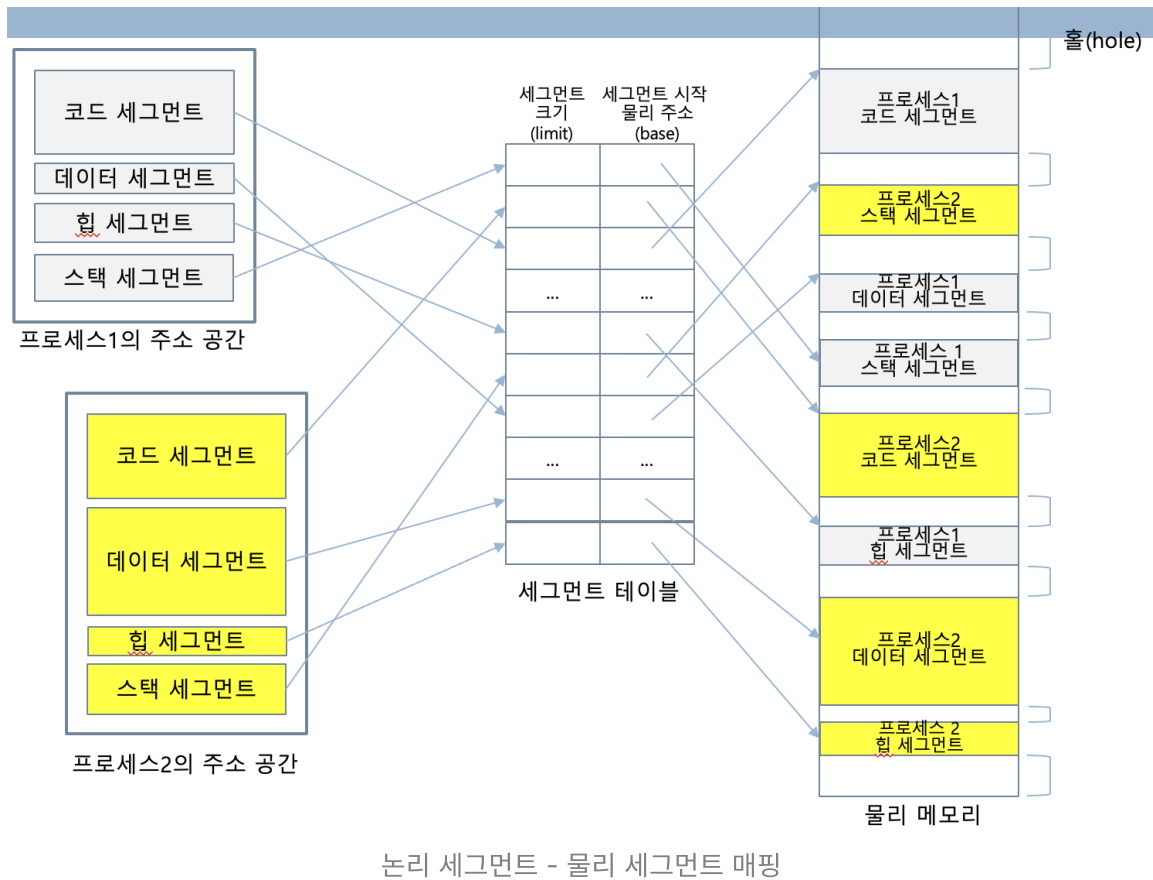


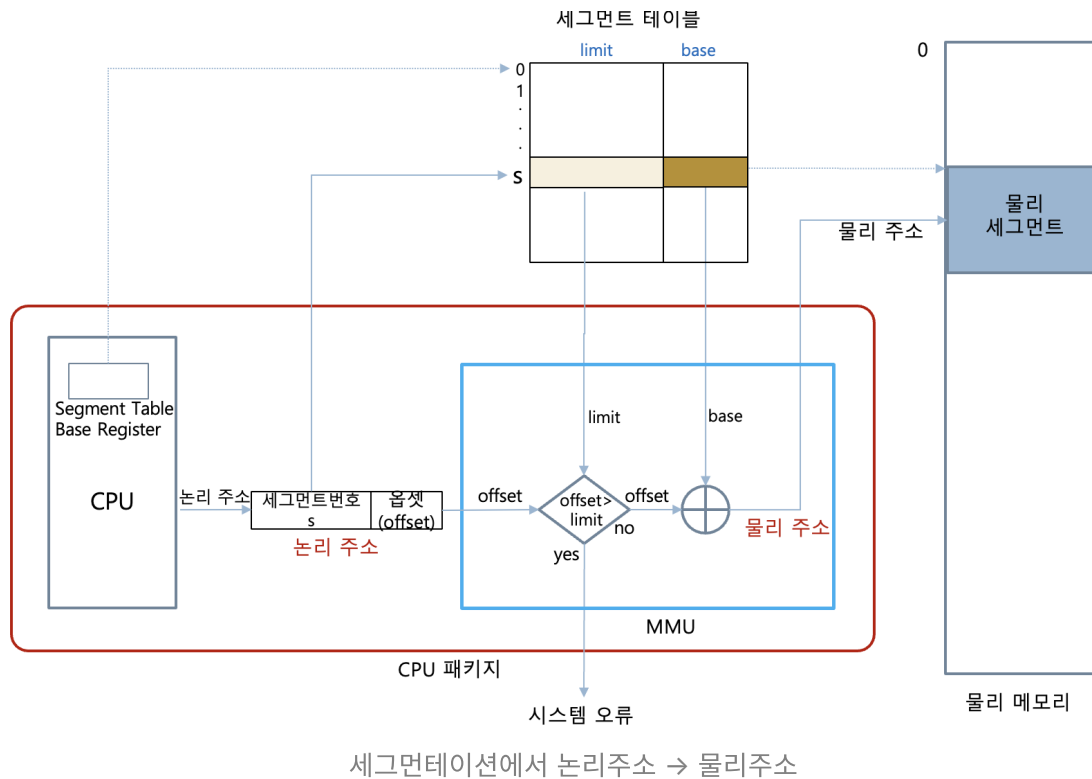
역 페이지 테이블을 사용한 논리 주소의 물리 주소 변환

Segmentation

- segment (세그먼트)
 - 세그먼트는 논리적 단위임 → 개발자의 관점에서 보는 프로그램 논리적 구성 단위
 - 세그먼트 마다 크기는 다르다.
- 프로그램 구성하는 세그먼트 종류
 - 코드 세그먼트
 - 데이터 세그먼트
 - 스택 세그먼트
 - 힙 세그먼트
- 세그멘테이션 기법
 - 프로세스를 세그먼트 크기로 나누고 각 세그먼트를 물리 메모리에 할당하고 관리하는 메모리 관리 기법이다.
 - 프로세스 주소 공간
 - 프로세스의 주소 공간을 여러 개의 논리 세그먼트들로 나눈다.

- 각 논리 세그먼트를 물리 세그먼트에 매핑한다.
- 프로세스를 세그먼트로 나누는 과정은 컴파일러, 링커, 로더, 운영체제에 의해 이루어짐
- 논리 세그먼트, 물리 세그먼트의 매핑
 - 시스템 전체 세그먼트 매핑 테이블을 두고 논리 주소를 물리 주소로 변환한다.
- 외부 단편화가 발생한다.





Segmentaion with Paging

- 세그먼테이션 + 페이징의 장점 결합하여 세그먼테이션 단점 보완
- 논리 메모리를 세그먼트로 나누고, 각 세그먼트를 페이지로 분할
- 프로세스의 주소 공간이 세그먼트로 나누어져 있음
- 페이지화된 세그먼트는 페이지 단위로 메모리에 할당되기에 내부 단편화가 줄고, 세그먼테이션의 외부 단편화 문제도 해결된다.