Mermory Management(2)

https://youtu.be/ExrLhGaFVvI

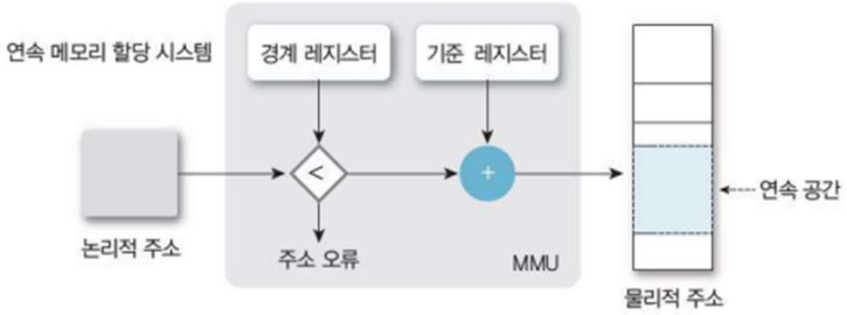
분산 메모리 할당 : 페이징

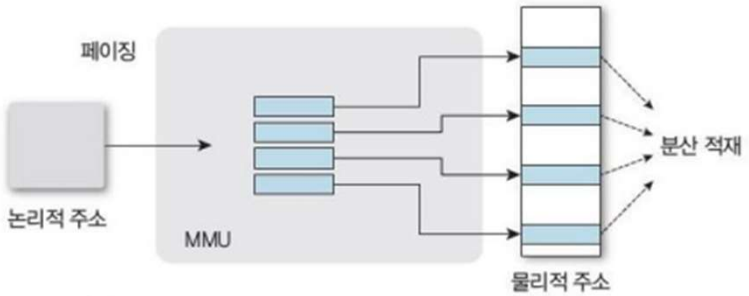
분산 메모리 할당 : 세그먼트

분산 메모리 할당: 페이징

페이징의 개념

- 작업을 크기가 동일한 페이지로 나눠 처리하는 방법
- 연속 메모리 할당과 비연속 메모리 할당 있음.





분산 메모리 할당: 페이징

페이징 시스템에서 프로그램 실행 준비

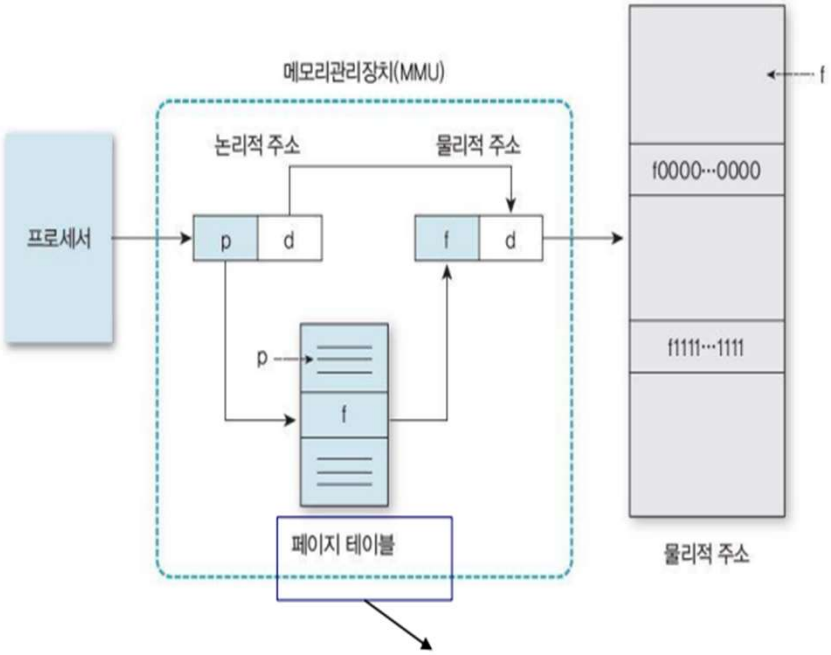
- 프로세스에 필요한 페이지를 결정하여 페이지 번호 부여
- 메모리의 빈 프레임을 조사하여 프로세를 적재할 위치 파악
- 프로세스의 페이지를 빈 프레임에 적재하도록 준비

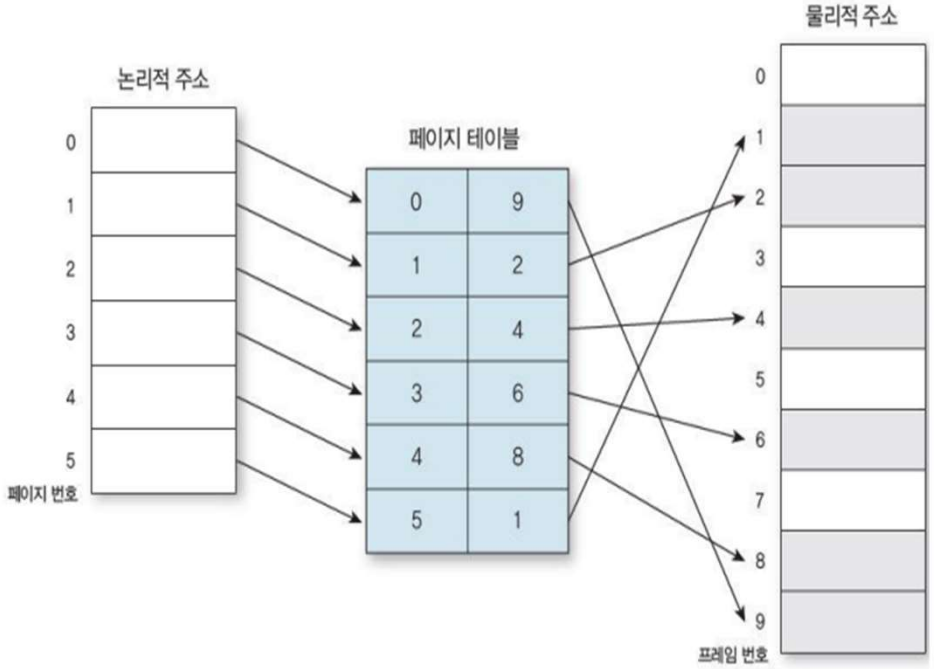
페이징의 특징

- 빈 프레임에 어떤 페이지든 적재할 수 있어 메모리 효율적 관리 가능
- 프레임 간에 외부 단편화 x
- 한 프로세스의 페이지를 메인 메모리의 여러 위치에 분산 적재하여 운영체제의 페이지 관리 부담 큼
- 내부 단편화 발생 가능성은 여전히 있음.

분산 메모리 할당: 페이징

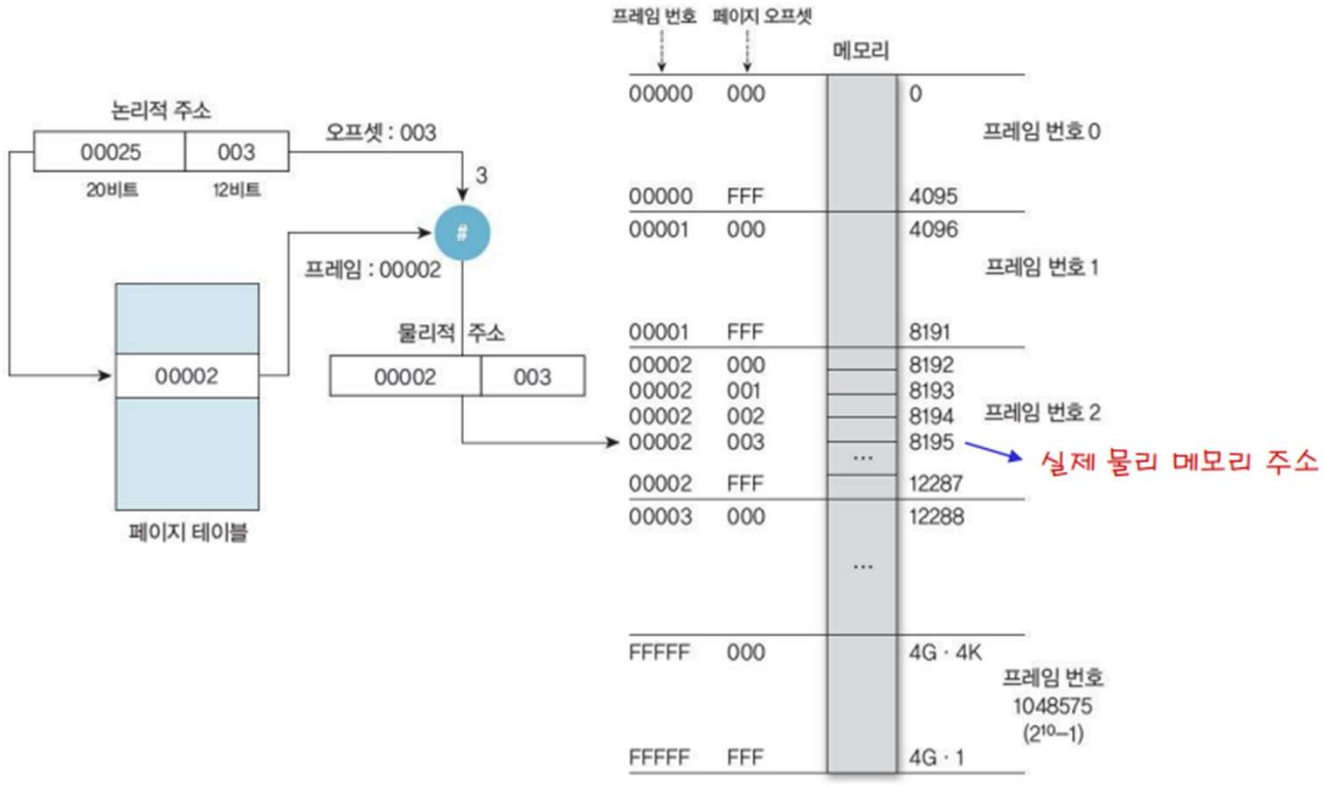
페이징 시스템의 하드웨어 구조





분산 메모리 할당: 페이징

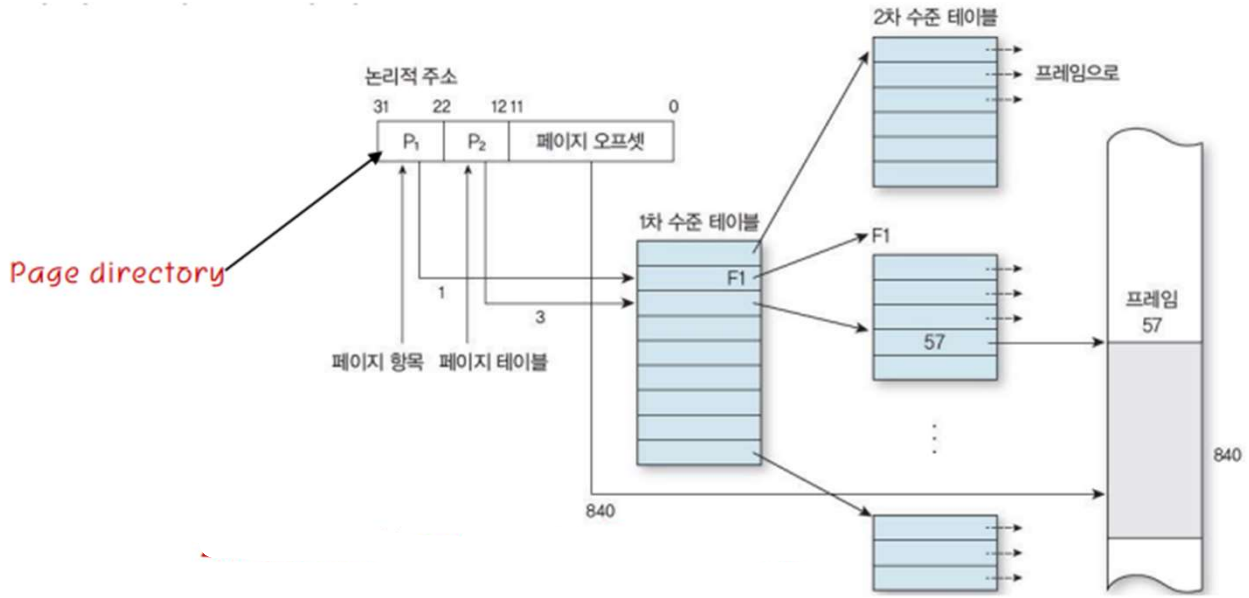
페이지 테이블을 이용한 물리적 주소 변환(32bit system)



분산 메모리 할당: 페이징

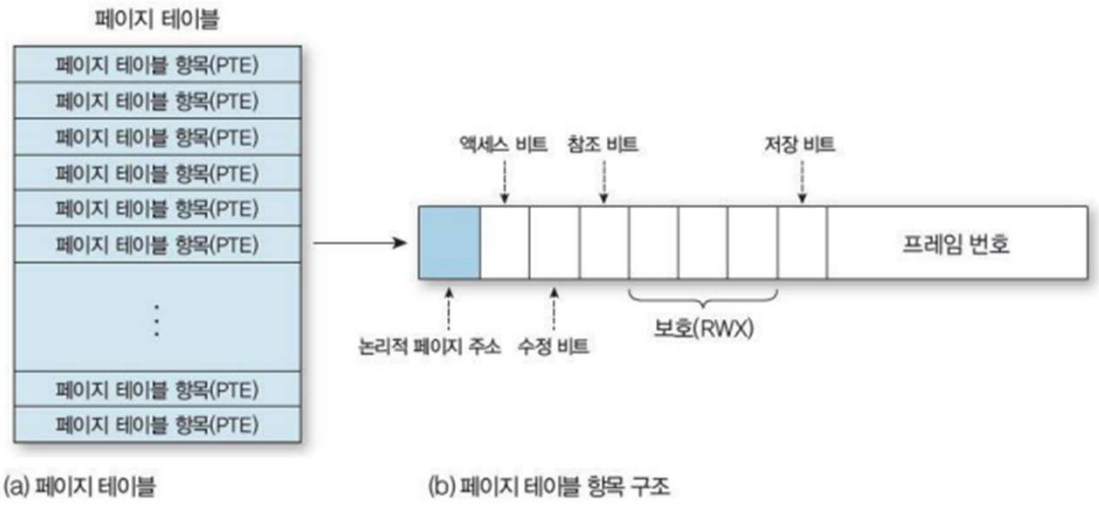
다중 단계 페이징 시스템의 구조와 원리

- 논리적 주소가 클수록 물리적 주소로 변환하는 과정에서 필요한 페이지 테이블 크기도 증가하므로 메모리에 더 큰 적재 공간 필요



분산 메모리 할당: 페이징

페이지 테이블



(a) 페이지 테이블 (b) 페이지 테이블 항목 구조

- 액세스 비트 : 메모리 액세스 여부 표시
- 수정 비트 : 메모리에 적재한 이후 수정했는지 여부
- 참조 비트 : 액세스 여부
- 보호(RWX) : 읽기 · 쓰기 · 실행 권한
- 저장 비트 : 참조 페이지의 메모리 저장 여부
- 프레임 번호 : 메인 메모리에서 페이지 프레임 번호

페이지 테이블 관리 방법

전용 레지스터 사용

- 대부분의 컴퓨터는 페이지 테이블이 매우 커서 레지스터 구현하기에 부적합

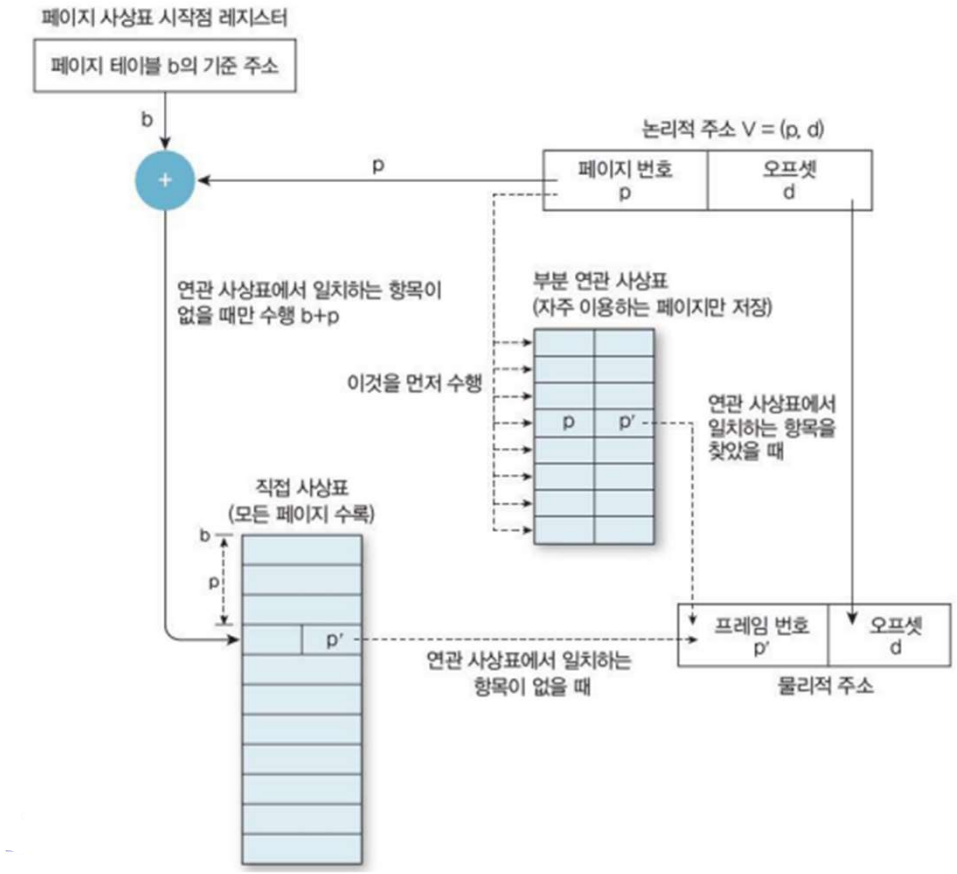
- 대게 페이지 테이블 메모리에 두고 페이지 테이블 기준 레지스터로 페이지 테이블 지시

- 페이지 테이블 항목과 워드를 위한 메모리 엑세스 필요, 이 엑세스 때문에 속도가 느려짐

→연관 레지스터나 TLB(변환 우선참조 버퍼)를 이용하여 해결 가능

분산 메모리 할당: 페이징

페이지 테이블: 연관 레지스터나 TLB(변환 우선참조 버퍼)를 활용해 주소 변환 과정

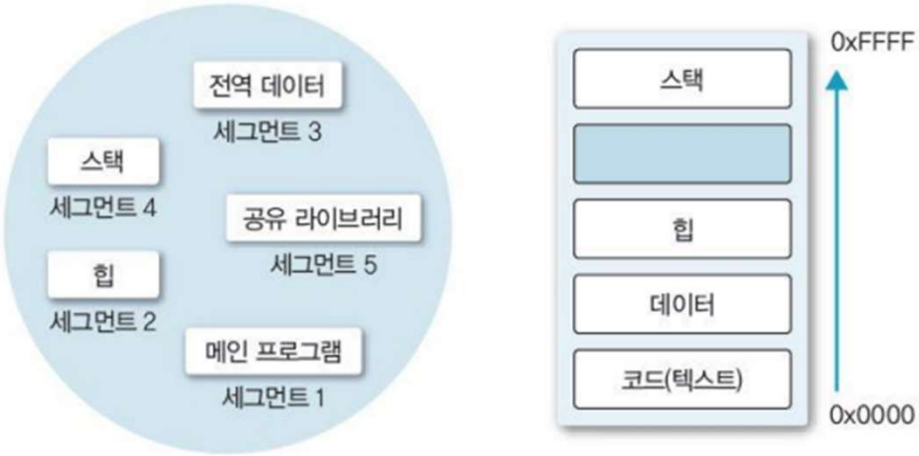


분산 메모리 할당: 세그먼트

세그먼테이션의 개념

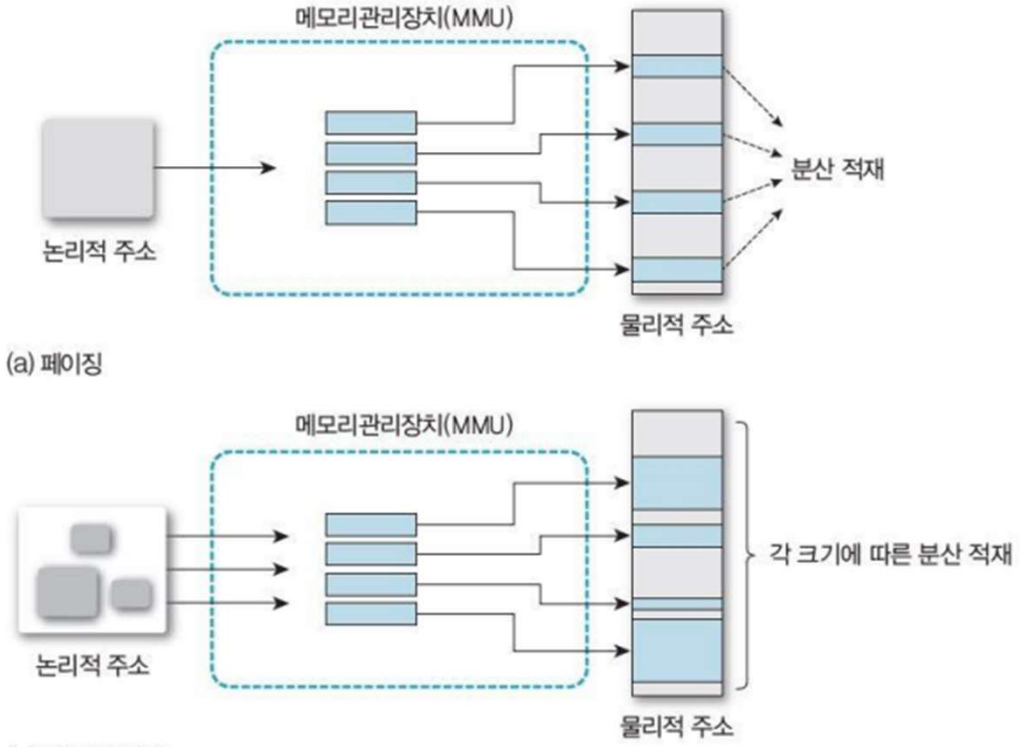
- 프로세스 관점을 지원하여 메모리를 크기가 변할 수 있는 세그먼트로 나누는 것
- 프로그램을 구성하는 서브루틴, 프로시저, 함수나 모듈 등으로 세그먼트 구성
- 각 세그먼트는 연관된 기능을 수행하는 하나의 모듈 프로그램으로 생각, 메모리의 연속된 위치에 구성하되 인접할 필요는 없음

프로그램(사용자) 관점의 메모리와 프로세스 관점의 메모리



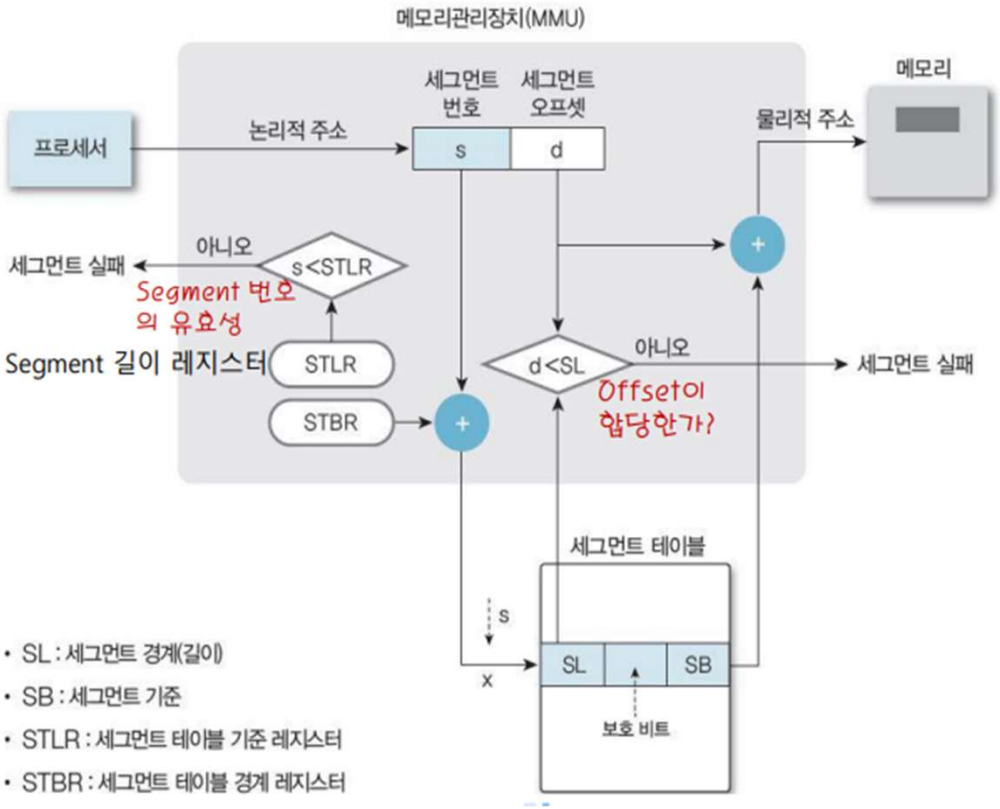
분산 메모리 할당: 세그먼트

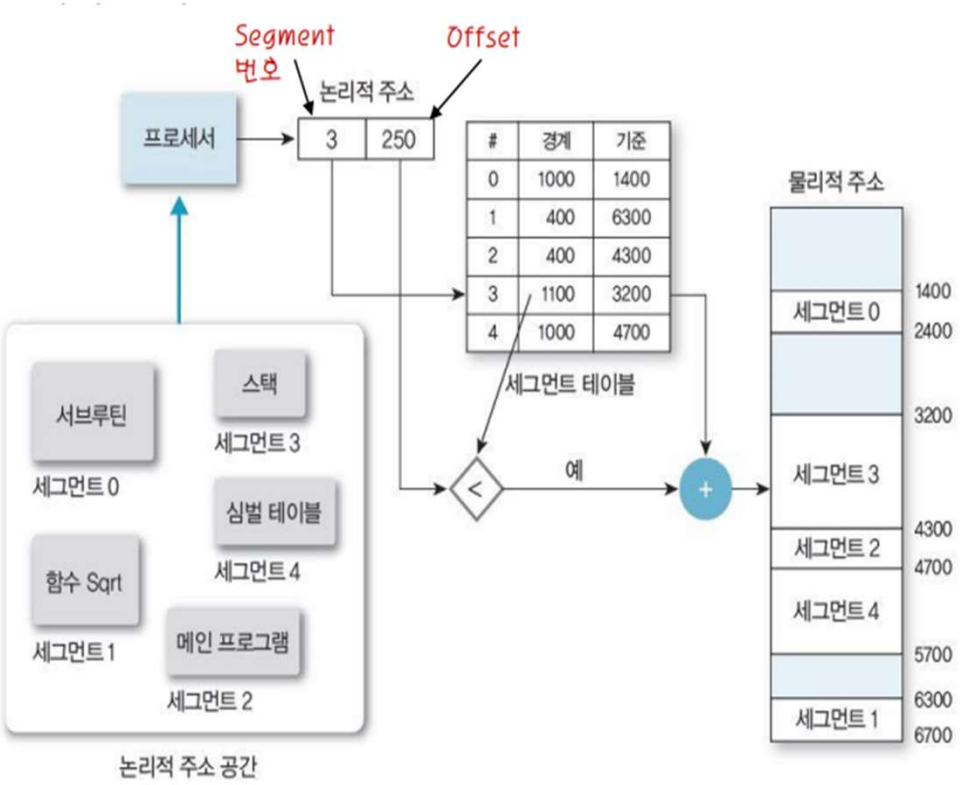
페이징과 세그먼테이션 메모리 할당 비교



분산 메모리 할당: 세그먼트

세그먼트 하드웨어 시스템 구조





분산 메모리 할당: 페이징 vs 분산 메모리 할당: 세그먼트

페이징과 세그먼테이션 비교

- 프로그램을 나눈 모든 세그먼트에서 메모리의 빈 공간을 찾아 할당하는 것 페이징과 비슷
- 페이징과 달리 프로그램을 나누는 크기가 변함
- 세그먼테이션도 최적 적합 알고리즘, 최초 적합 알고리즘으로 동적 메모리 할당 방법 이용
- 따라서 외부 단편화 가능, 사용 가능한 메모리의 모든 블록이 너무 작아서 세그먼트를 수용할 수 없을 때 발생
- 압축 가능.
- 페이징은 물리적 주소 없이도 큰 가상 주소 공간이 가능하게 하려고 등장, 세그먼테이션은 프로그램과 데이터를 논리적으로 독립된 주소 공간으로 나누고 쉽게 보호할 수 있게 하려고 등장.

Q & A