인덕대학교

메모리 관리
- 페이징과 세그먼테이션

컴퓨터소프트웨어학과

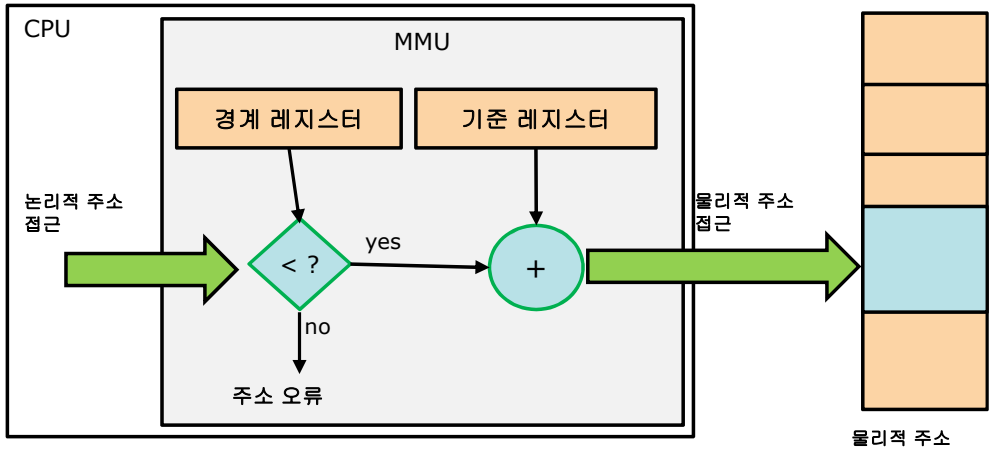
김병국 교수

학습목표

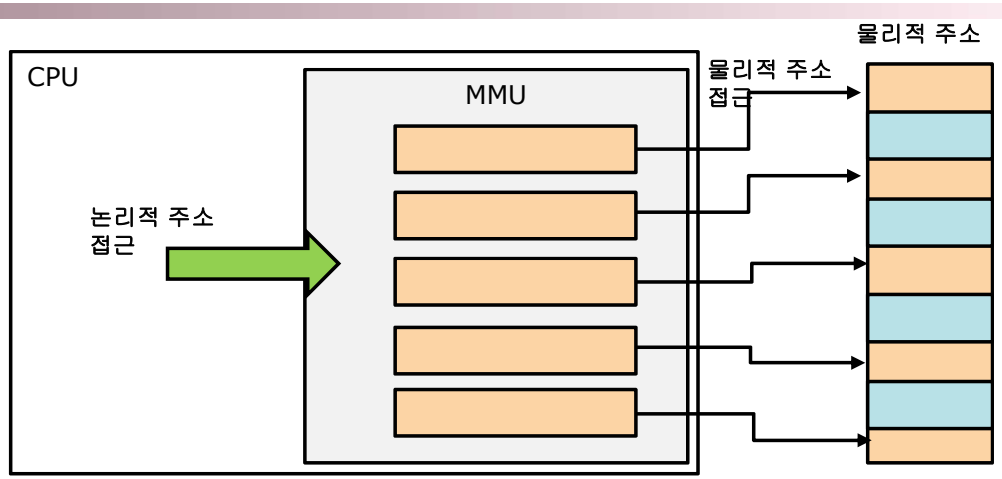
- 페이징 기법에 대하여 개념과 기본 동작 방식을 이해한다.
- 세그먼테이션 기법에 대하여 개념과 기본 동작 방식을 이해한다.

목차

1. 페이징 개념 (1/3)



[연속 메모리 할당 시스템]



[페이징 시스템을 이용한 비연속 메모리 할당]

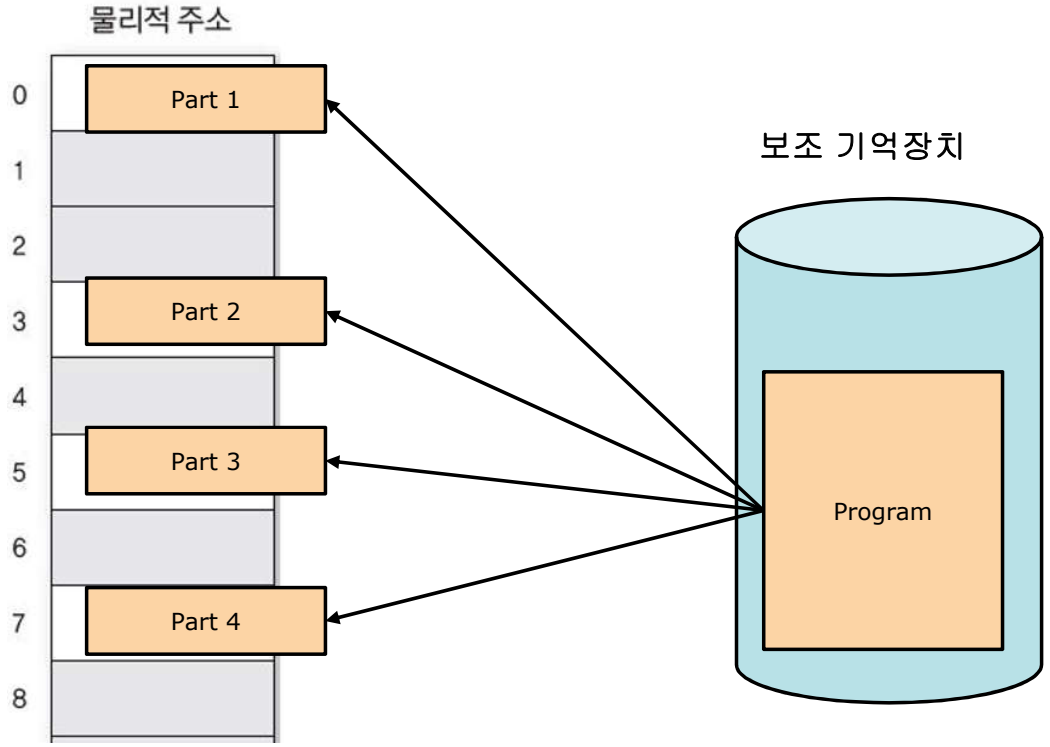
❑ 개념

- 비연속 메모리 할당 기법을 사용
- 프로세스 하나가 **여러 개로 나뉘어 메모리에 할당됨**
  - 페이지(page)의 크기는 모두 동일

1. 페이징 개념 (2/3)

❑ 프로그램 로딩

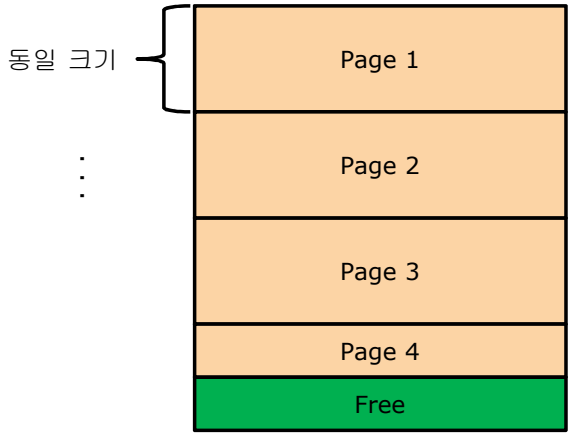
- 프로세스에 필요한 **페이지를 결정하여 페이지 번호 부여**
- 메모리의 빈 프레임을 조사하여 프로세스를 적재할 위치 파악
- 프로세스의 페이지를 빈 프레임에 적재



1. 페이징 개념 (3/3)

❑ 특징

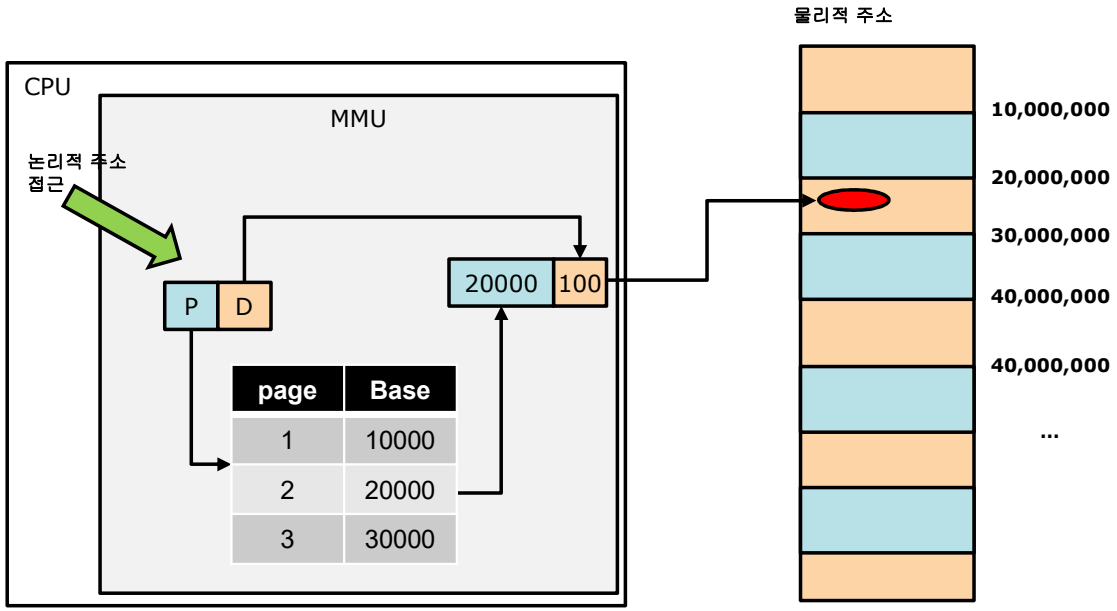
- 빈 프레임에 어떤 페이지이든 적재 → 효율적인 메모리 사용
- 프로세스가 분산된 위치에 적재 → 운영체제 관리 부담 큼
- 프레임 간 외부 단편화(External Fragmentation) 방지
  - 단, 마지막 페이지에 할당된 프레임이 완전히 차지 않을 경우 단편화 발생



2. 페이징 (1/4)

❑ 하드웨어적 접근 구조

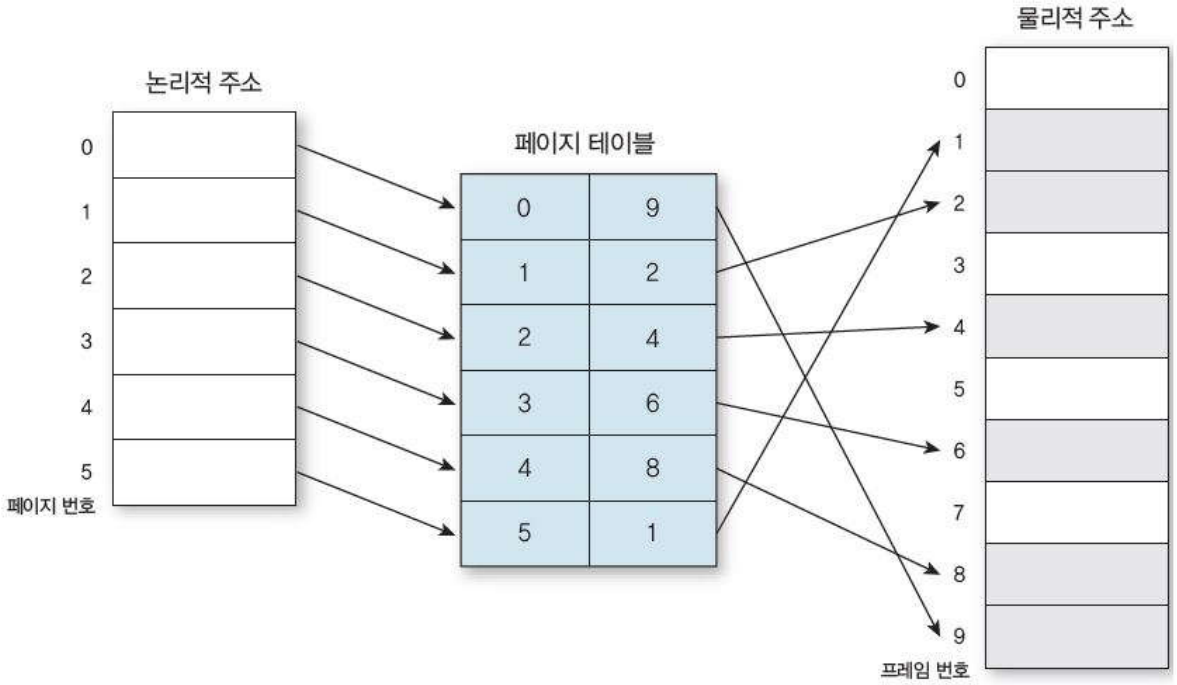
- CPU는 논리 주소로 접근
- MMU에 의해 물리적 주소로 전환되어 주 기억장치에 접근



2. 페이징 (2/4)

인덕대학교

❑ 접근 예



2. 페이징 (3/4)

인덕대학교

❑ 16비트 논리적 주소 사용 예

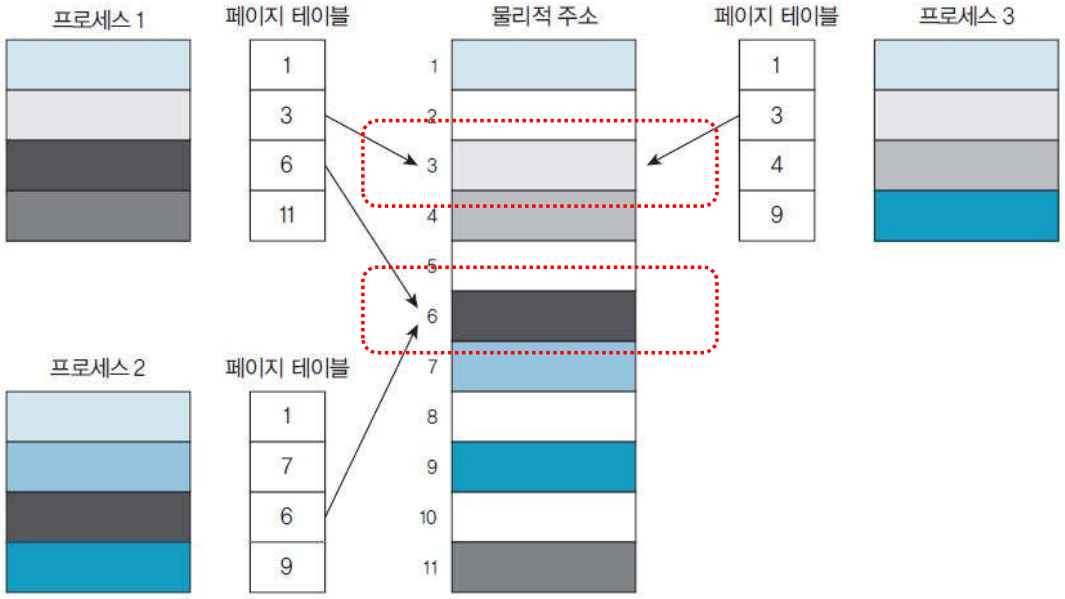
- $2^6 = 64$ ← 최대 64개의 페이지를 구성 가능
- $2^{10} = 1024$ ← 페이지의 크기는 최대 1024B가 될 수 있음

2. 페이징 (4/4)

인덕대학교

❑ 공유 페이지

- 다중 태스크환경에서 공유되는 명령코드들에 대한 공간을 공유
- 공유 코드는 재 진입을 허용하므로 재진입 코드(re-entrant code) 또는 순수 코드(pure code)라고 함
- 오직 읽을 수만 있으며 스스로 수정하지는 못함.



3. 페이징 관련 명령어

인덕대학교

❑ 명령어

- 명령: vmstat
  - 가상 메모리에 대한 통계자료를 출력하는 프로그램
- 명령: getconf PAGESIZE
  - 시스템 설정값을 추출하는 프로그램

```
[kali@kali:~]$
[kali@kali:~]$getconf PAGESIZE
4096
[kali@kali:~]$getconf PAGE_SIZE
4096
[kali@kali:~]$
```

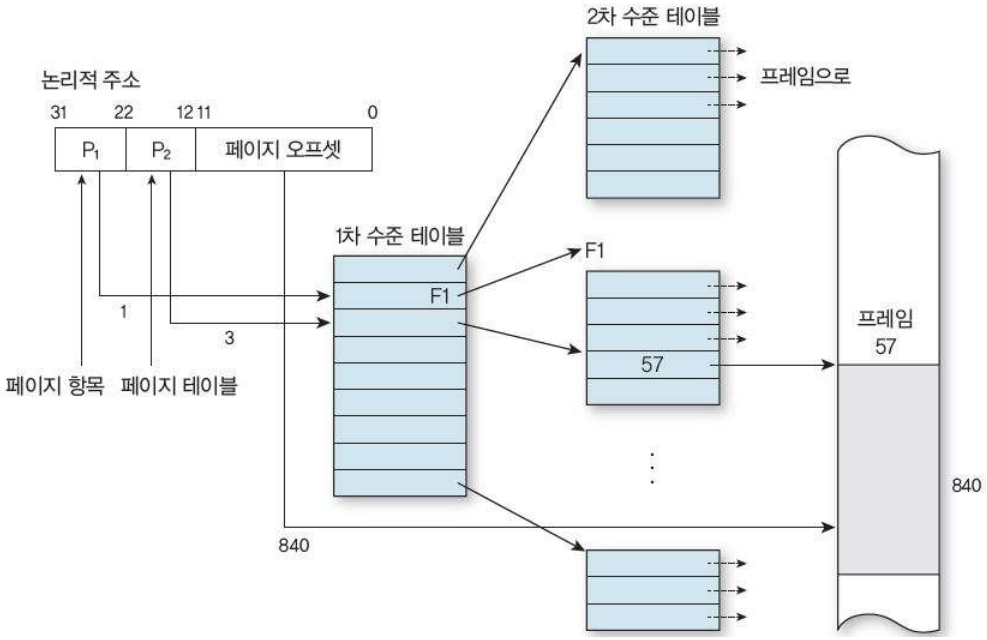
```
[kali@kali:06_1]$vmstat -s
      4027360 K total memory
       466748 K used memory
       270032 K active memory
       687344 K inactive memory
      2905112 K free memory
       119152 K buffer memory
       536348 K swap cache
       998396 K total swap
            0 K used swap
       998396 K free swap
         2678 non-nice user cpu ticks
           14 nice user cpu ticks
         3134 system cpu ticks
     12069281 idle cpu ticks
         1840 IO-wait cpu ticks
            0 IRQ cpu ticks
         7120 softirq cpu ticks
            0 stolen cpu ticks
       562047 pages paged in
       114172 pages paged out
            0 pages swapped in
            0 pages swapped out
      6507221 interrupts
      3937792 CPU context switches
   1617073727 boot time
         3266 forks
[kali@kali:06_1]$
```

4. 다단계 페이징

인덕대학교

❑ 원리

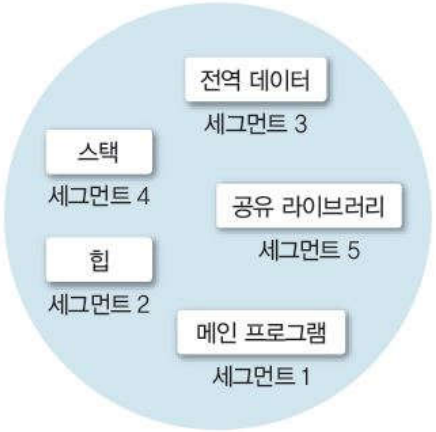
- 논리적 주소가 클수록 페이지 테이블 크기도 증가
- 메모리에 더 큰 적재 공간 필요
- 더 넓은 영역을 접근하기 위해 테이블을 계층화 함



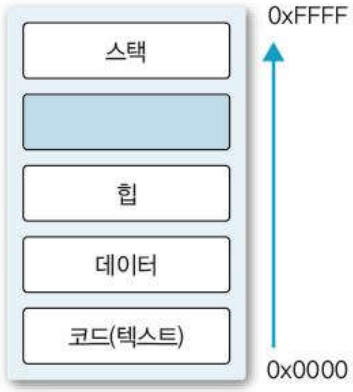
5. 세그먼테이션 개념 (1/2)

❑ 개념 (1/2)

- 비연속 메모리 할당 방법
- 프로세스를 동적의 크기를 갖는 여러 개의 세그먼트(Segmentation)로 나눔
- 세그먼트를 나눌 때, 프로그램을 구성하는 서브루틴(subroutine), 프로시저(procedure), 함수(function) 단위로 함
  - 각 세그먼트는 기능을 수행하는 하나의 작은 프로그램으로 간주



(a) 프로그램(사용자) 관점의 메모리

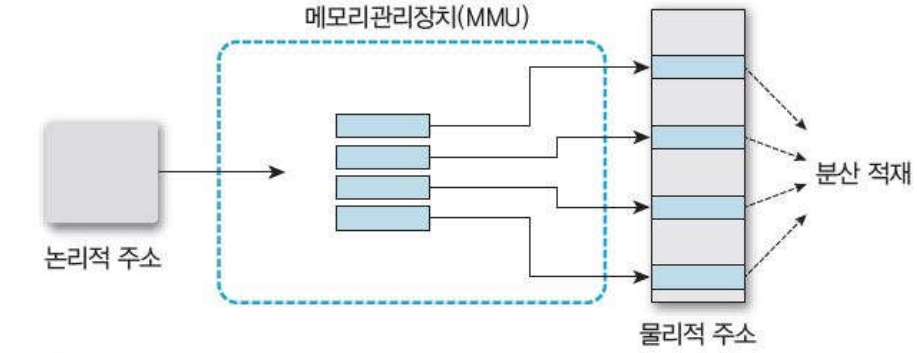


(b) 프로세스 관점의 메모리

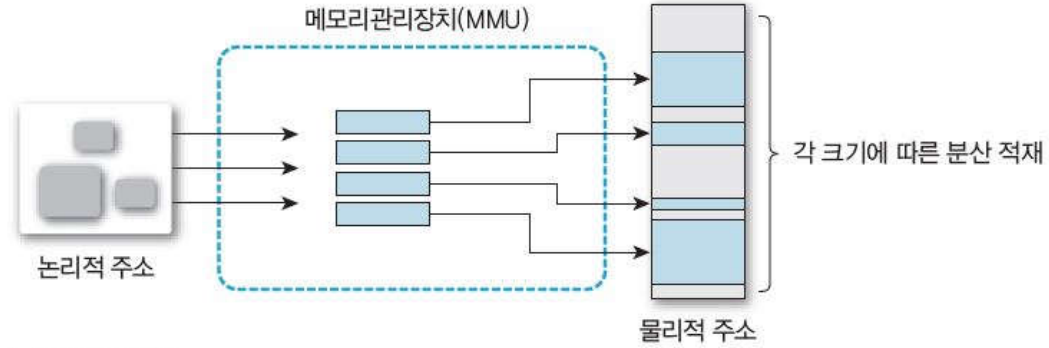
5. 세그먼테이션 개념 (2/2)

❑ 개념 (2/2)

- 관리에 필요한 사항은 페이징과 비슷하거나 동일
  - 세그먼트 크기가 달라 메모리를 동적 분할(가변 분할) 방법으로 할당이 차이



(a) 페이징

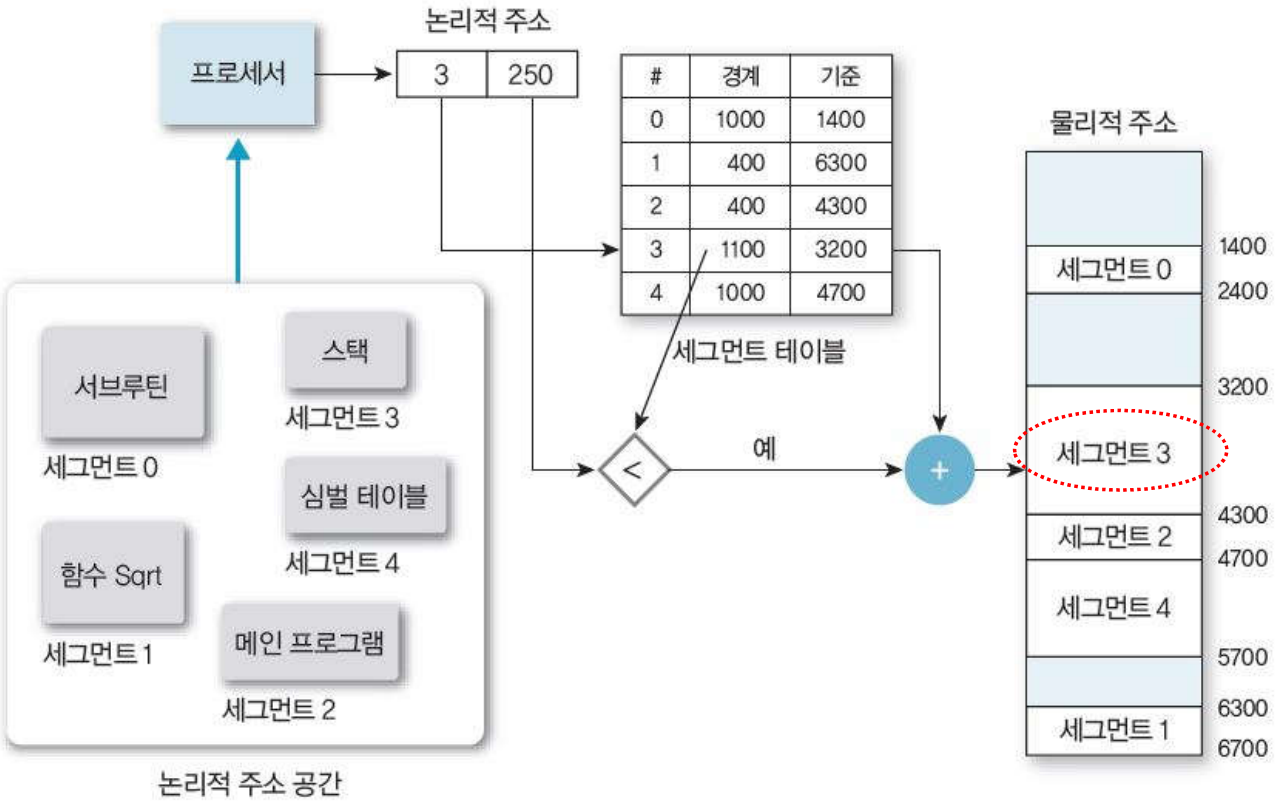


(b) 세그먼테이션

6. 세그먼테이션 (1/2)

❑ 하드웨어적 접근 구조

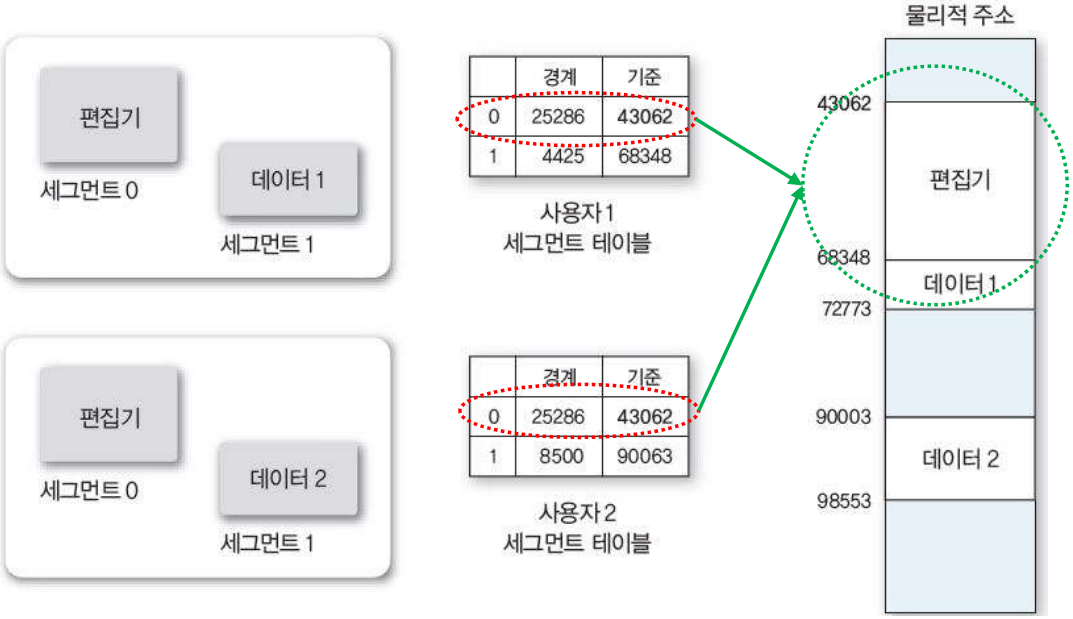
- CPU는 논리 주소로 접근
- MMU에 의해 물리적 주소로 전환되어 주 기억장치에 접근



6. 세그먼테이션 (2/2)

❑ 공유 세그먼테이션

- 페이징 시스템 대비 단순한 구조
  - 페이징 시스템은 다수의 페이지 블록을 공유
  - 세그먼트는 소수의 블록을 공유
- 공유 기능을 사용하는 시스템은 적절한 보호 체계를 제공하여 승인한 사용자만 세그먼트에 액세스할 수 있음
- 세그먼테이션에서 세그먼트 공유



7. 페이징 vs. 세그먼테이션

❑ 비교

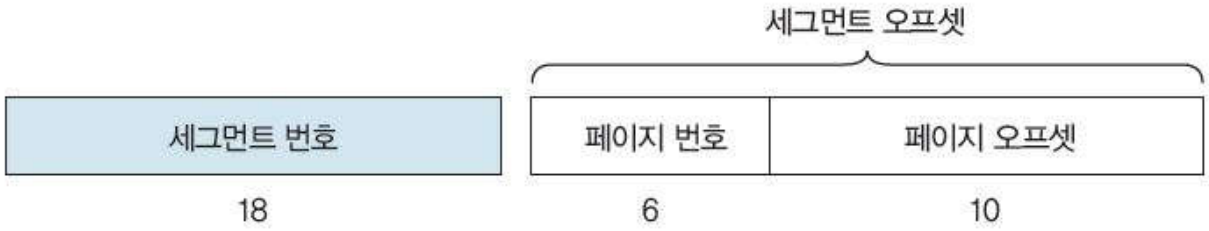
- 메모리의 빈 공간을 찾아 할당하는 방식은 서로 비슷함
- 페이징과 달리 프로그램을 나누는 크기가 변함
- 가변 크기 분할 방법처럼 세그먼테이션도 후속 처리가 필요
  - 보통 최적 적합(Best Fit) 또는 최초 적합(First Fit) 알고리즘을 사용

- 설계 목적의 차이
  - 페이징 :
    - 물리적 주소 없이도 큰 가상 주소 공간 사용 가능
  - 세그먼테이션 :
    - 프로그램과 데이터를 논리적으로 독립된 주소 공간으로 나눔

8. 페이지화된 세그먼테이션 (1/2)

❑ 개념

- 페이징과 세그먼테이션의 장점을 취합
  - 페이징은 내부 단편화가 발생할 수 있으나, 메모리 효율적 사용, 작업 크기가 동일
  - 세그먼테이션은 외부 단편화가 발생할 수 있으나, 가변적인 데이터 구조와 모듈 처리, 공유와 보호의 지원 편리
- 외부 단편화 문제를 제거하면서 할당 과정 쉽게 해결
- 인텔 386이후 적용
- 윈도우10의 경우 이 방식을 사용

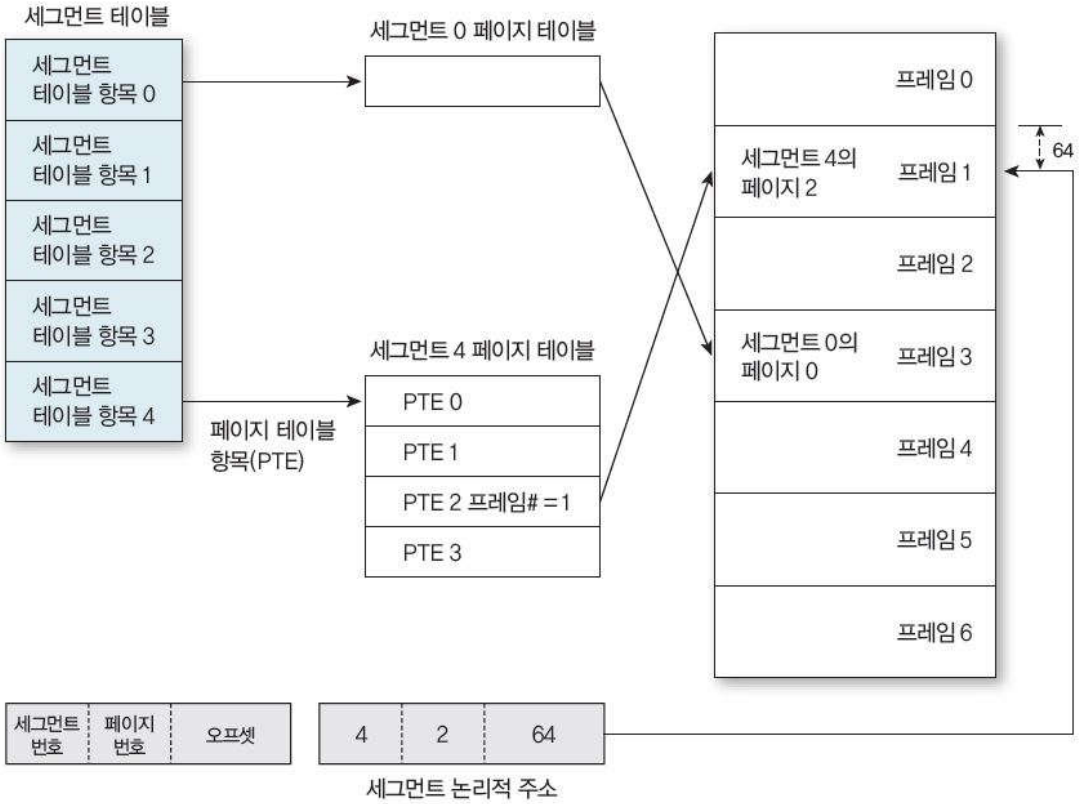


8. 페이지화된 세그먼테이션 (2/2)

인덕대학교

❑ 하드웨어적 접근 구조

- CPU는 논리 주소로 접근
- MMU에 의해 물리적 주소로 전환되어 주 기억장치에 접근



수고하셨습니다.