

운영체제

2019.7.9



컴퓨터공학과 이병문



7/9 메모리관리2

10 가상메모리1(수행과제2)

11 가상메모리2

12 기말고사

분산메모리할당

- 페이징, 워드페이징
- 페이지 스케줄링
- 페이지테이블

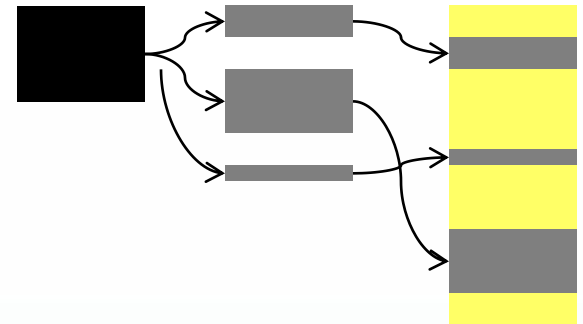
세그먼트 메모리할당

- 세그먼트, 세그먼트 매핑
- 페이지화된 세그먼트

분산 메모리 할당

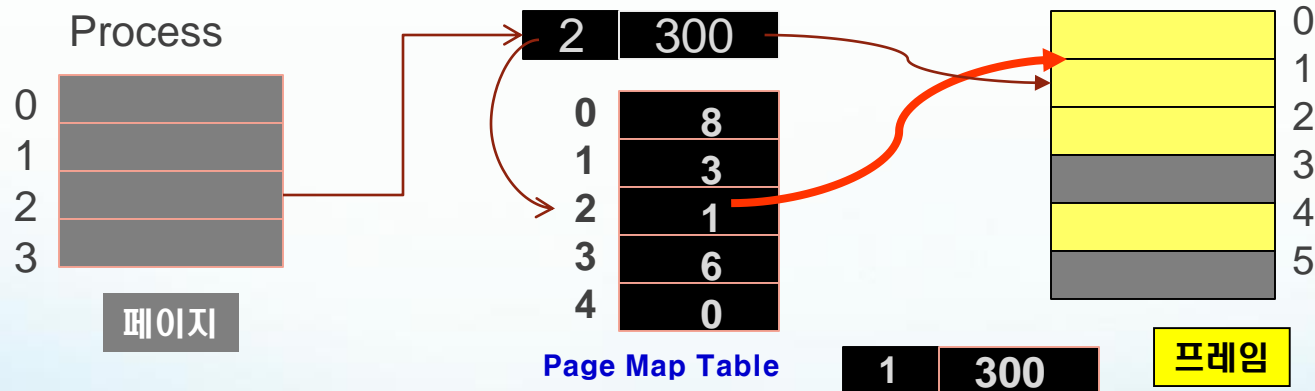
■ 분산메모리 할당

- ✓ 동적재배치를 통한 외부단편화, 내부단편화를 해결
- ✓ 한 프로세스(작업)의 메모리할당시 불연속 할당을 도입
- ✓ 분산 메모리 할당의 종류
 - 페이징 (Paging) 기법
 - 세그먼트(Segment) 기법
 - 페이지화된 세그먼트(Paged Segment) 기법



■ 페이징(Paging)

- ✓ 페이지 (Page) ; 프로세스를 일정크기의 블록으로 나눈 단위
- ✓ 프레임(페이지 프레임) - 메모리를 고정 크기 블록으로 나눈 단위

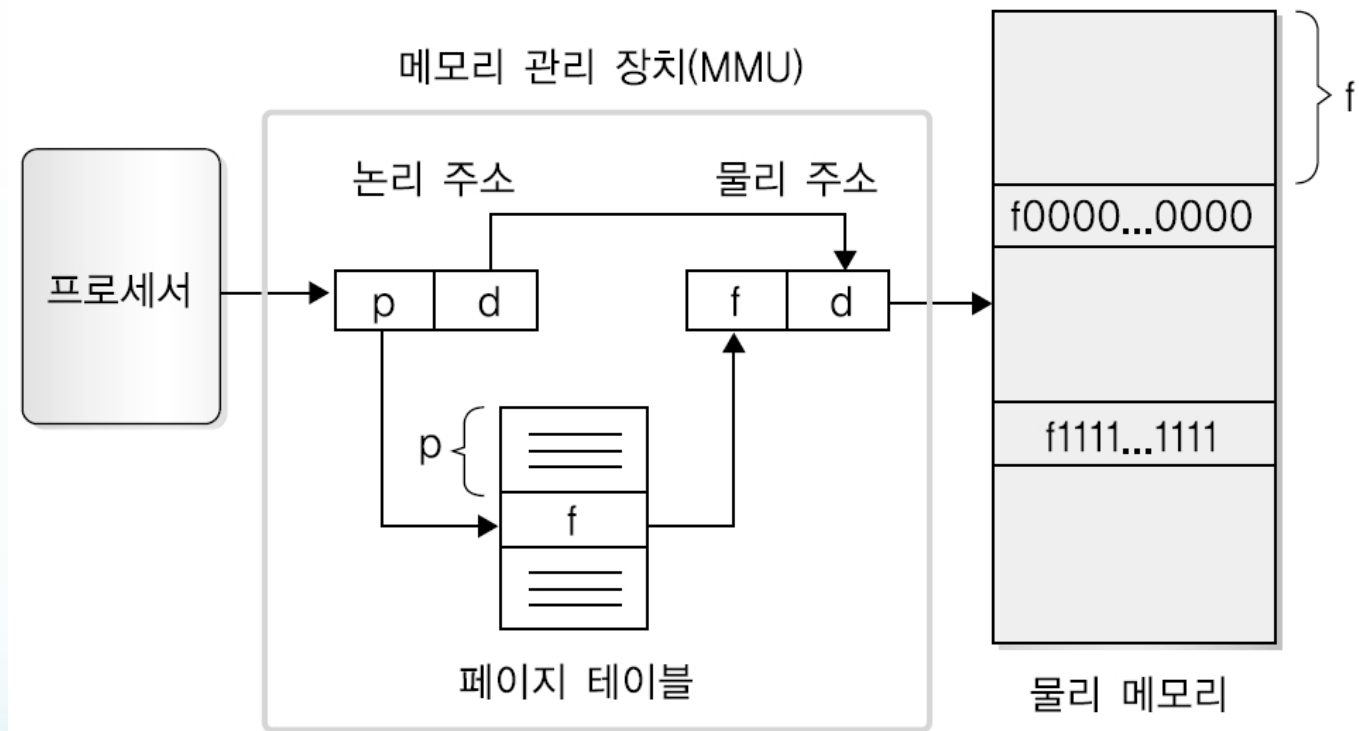


분산 메모리 할당 - 페이징

■ 페이징 시스템 하드웨어

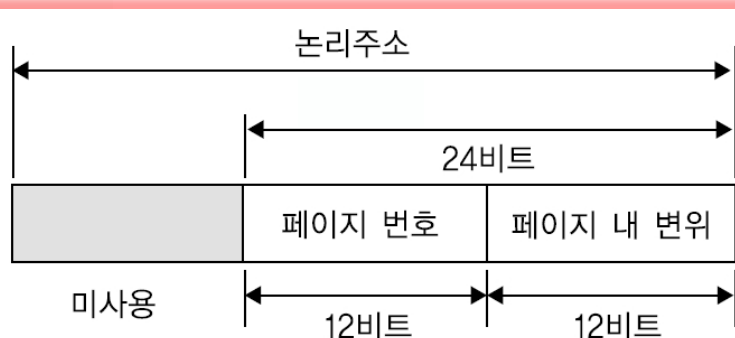
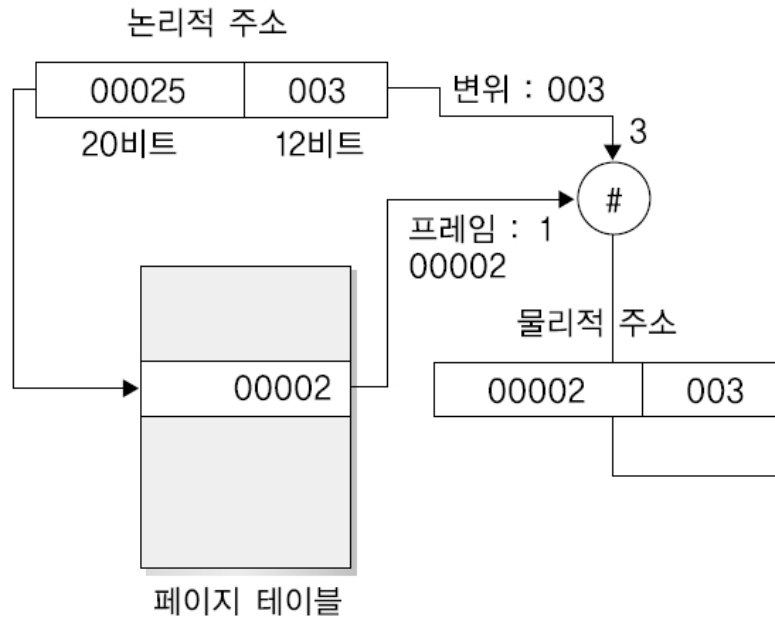
- ☑ 프로세서 생성 주소(논리 주소) : 페이지 번호(p)와 변위(d)
 - 페이지 번호 : 페이지 테이블 색인(메모리에서의 각 페이지의 기준주소(f) 유지)
 - 변위(Offset): 상대주소(페이지 시작위치로부터 프레임내 위치 표시)

실제 메모리 주소 = 기준 주소 + 페이지 변위



분산 메모리 할당 - 페이징

■ 페이징시스템의 물리주소변환



프레임 번호	페이지 변위	메모리	
00000	000		0
프레임 번호 0			
00000	FFF		4095
00001	000		4096
프레임 번호 1			
00001	FFF		6191
00002	000		8192
00002	001		8193
00002	002		8194
00002	003		8195
프레임 번호 2			
		...	
00002	FFF		12287
00003	000		12288
...			
FFFFFF	000		4G·4K
프레임 번호 1048575 (2 ¹⁰ -1)			
FFFFFF	FFF		4G·1

분산 메모리 할당 - 페이징

■ 페이지 번호

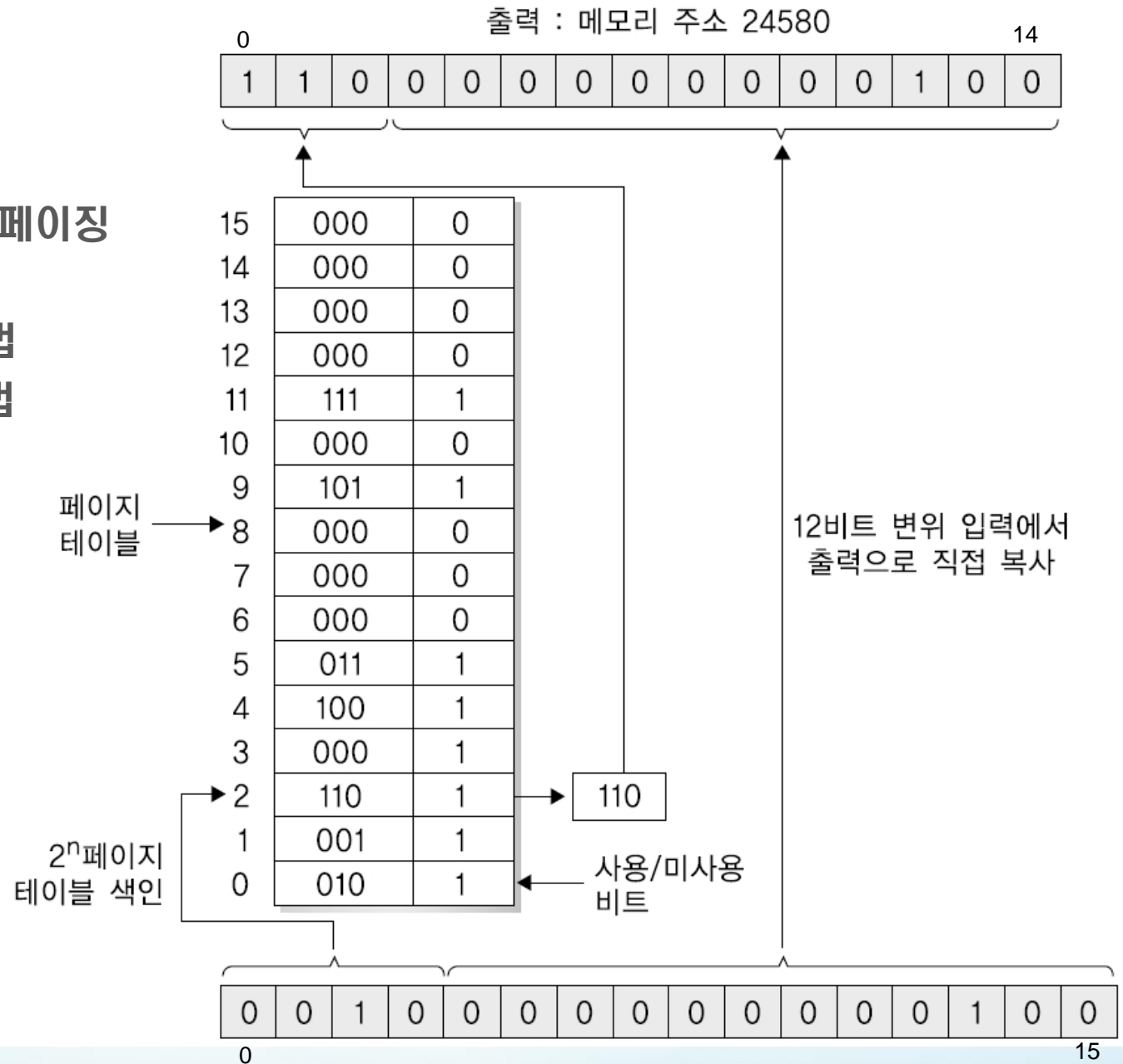
☑ 페이지 번호체계

- 주소변환체계 -> 페이징

- 메모리 페이징기법

워드페이징이용법

2차수준페이징



분산 메모리 할당 - 페이징

■ 워드페이징을 이용한 Paging

☑ 메모리의 사용자 관점

논리주소 - 물리주소의 사상

논리 주소 0 : 페이지 0, 변위 0
페이지 테이블 색인 :

페이지 0 → 프레임 5 확인

논리 주소 0

⇒ 메모리 주소 $20 (= 5 \times 4 + 0)$ 으로 사상

논리 주소 3(페이지 0, 변위 3)

⇒ 메모리 주소 $23 (= 5 \times 4 + 3)$ 으로 사상

논리 주소 4

⇒ 페이지1, 변위 0

페이지 1

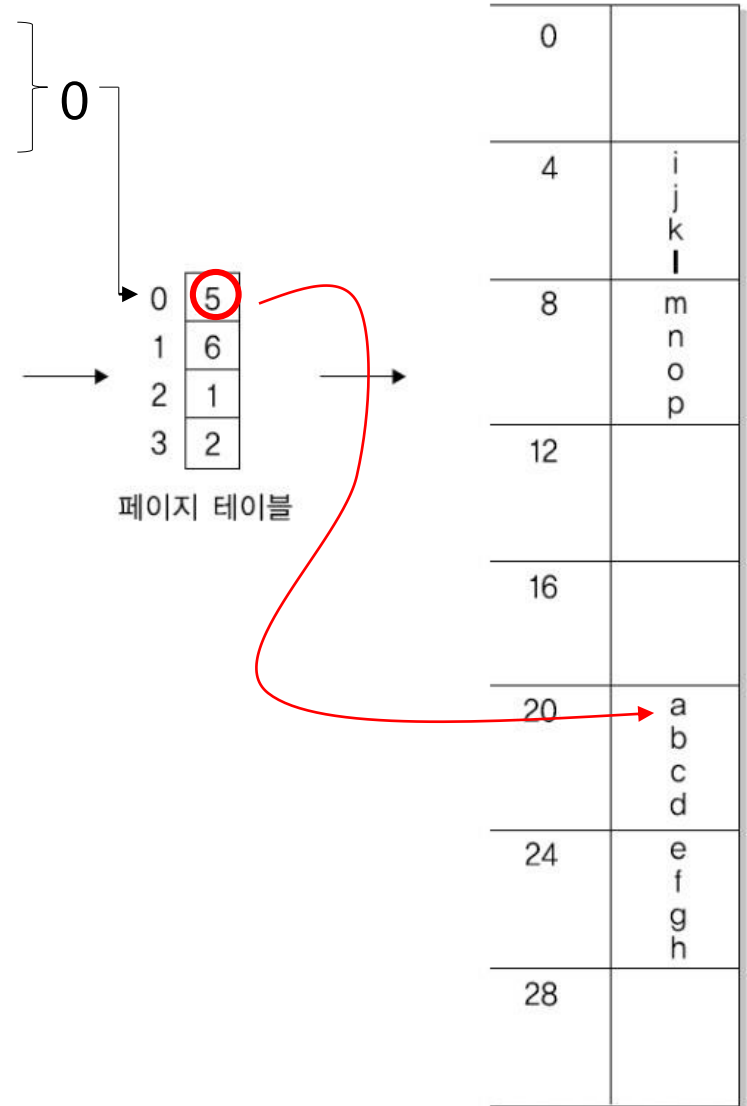
⇒ 프레임 6, 메모리 주소 $24 (= 6 \times 4 + 0)$

논리 주소 13

⇒ 메모리 주소 9로 사상

0	a
1	b
2	c
3	d
4	e
5	f
6	g
7	h
8	i
9	j
10	k
11	l
12	m
13	n
14	o
15	p

논리 메모리



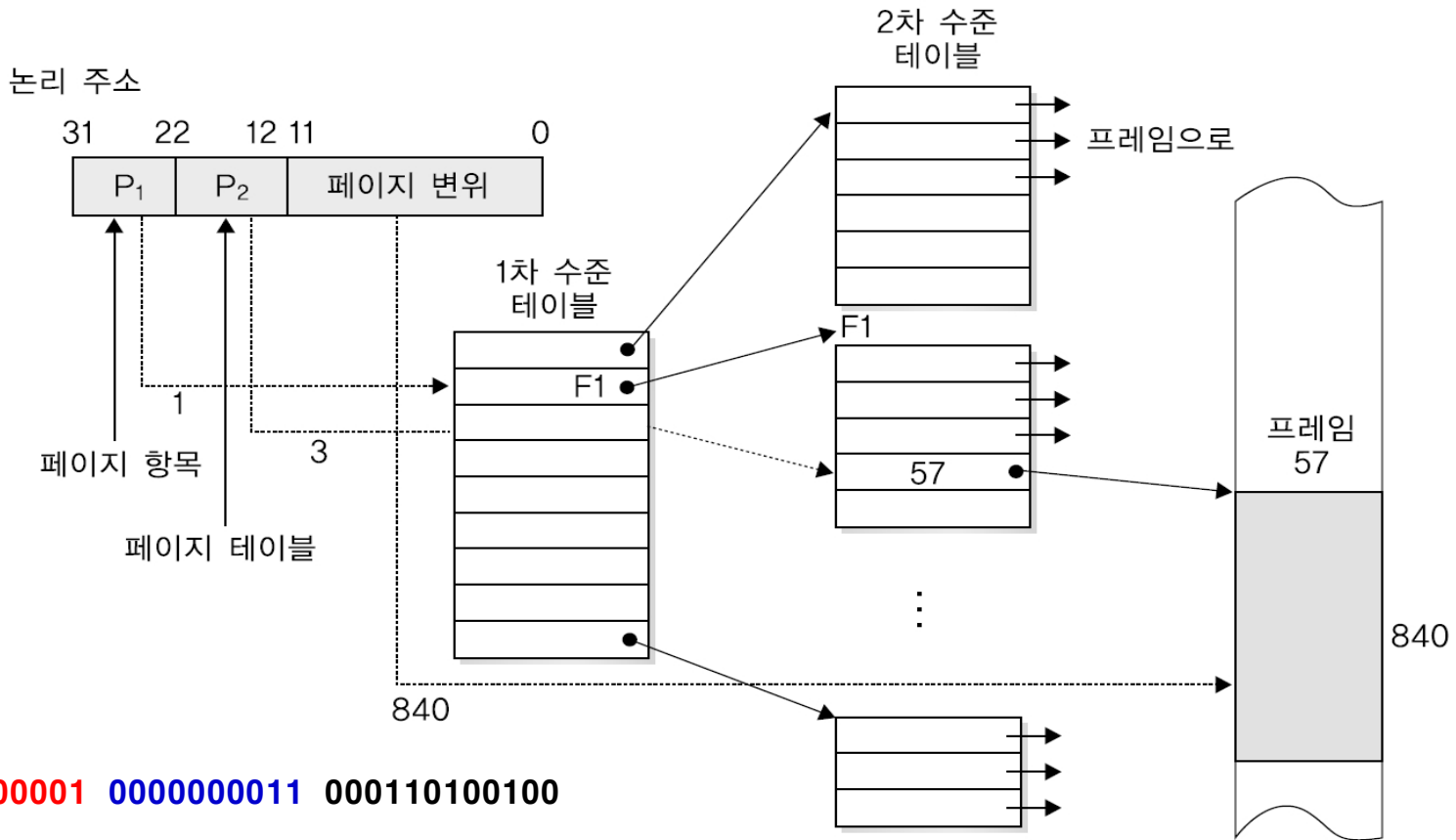
물리 메모리

분산 메모리 할당 - 페이징

■ 2차 레벨 페이지 테이블 구성

- ☑ 페이징 - 동적 재배치의 형태
논리 주소 : 실제 주소로 사상

메모리의 각 프레임에 대한 기준 레지스터 테이블을 사용하는 것과 유사



분산 메모리 할당 - 페이징

■ 페이지 스케줄링(Page Scheduling)

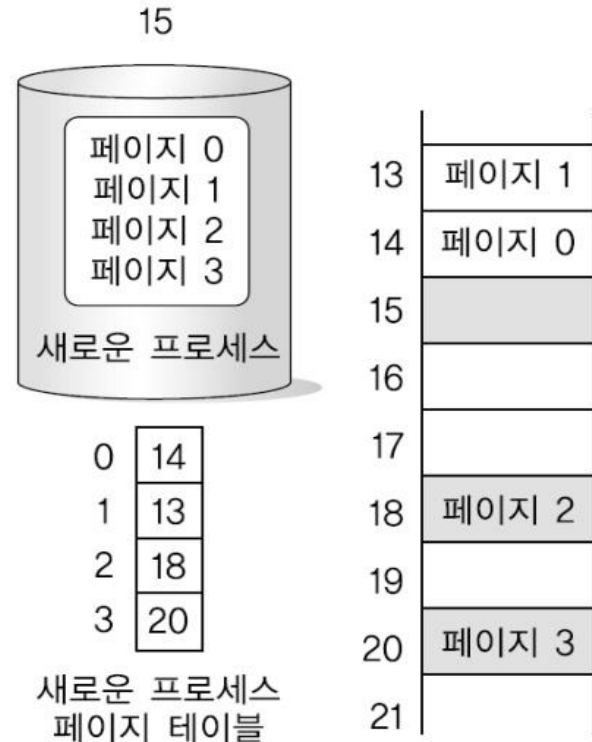
- ☑ 장기스케줄러(Longterm Scheduler)는 프로세스(n 개 페이지)를 메모리에 할당하기 위해 n 개 프레임이 준비되면 스케줄링한다

사용가능 프레임 리스트



(a) 사용가능 프레임의 할당 전

사용가능 프레임 리스트



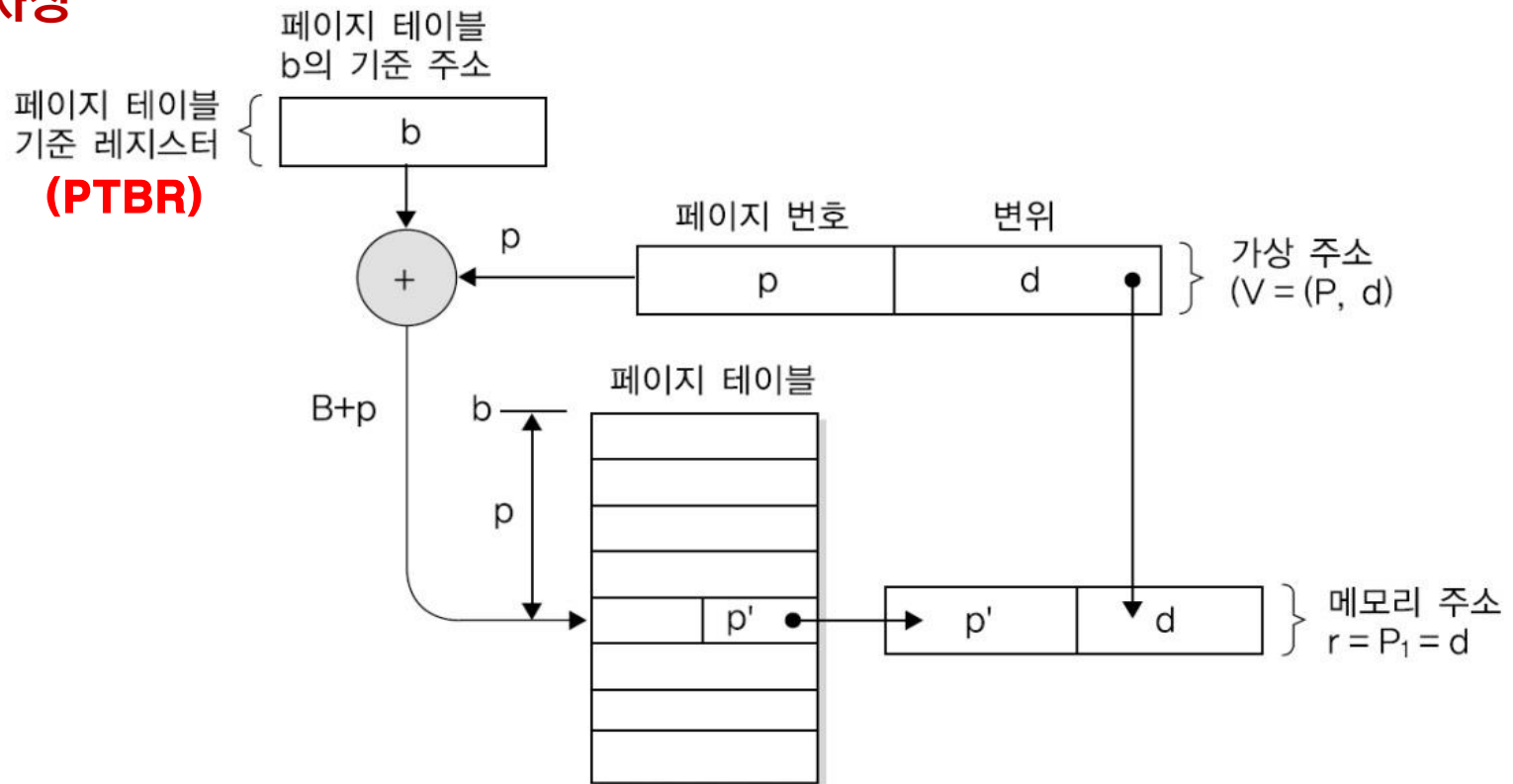
(b) 사용가능 프레임의 할당 후

분산 메모리 할당 - 페이징

■ 페이지 테이블의 구현

- ✓ 전용레지스터를 사용한 하드웨어적 구현방법
- ✓ 메모리와 PTBR를 이용한 구현방법
 - 3가지 주소변환(Mapping) 방식 : 직접사상, 연관사상, 연관/직접사상

✓ 직접사상



분산 메모리 할당 - 페이징

■ 페이지 테이블의 구현

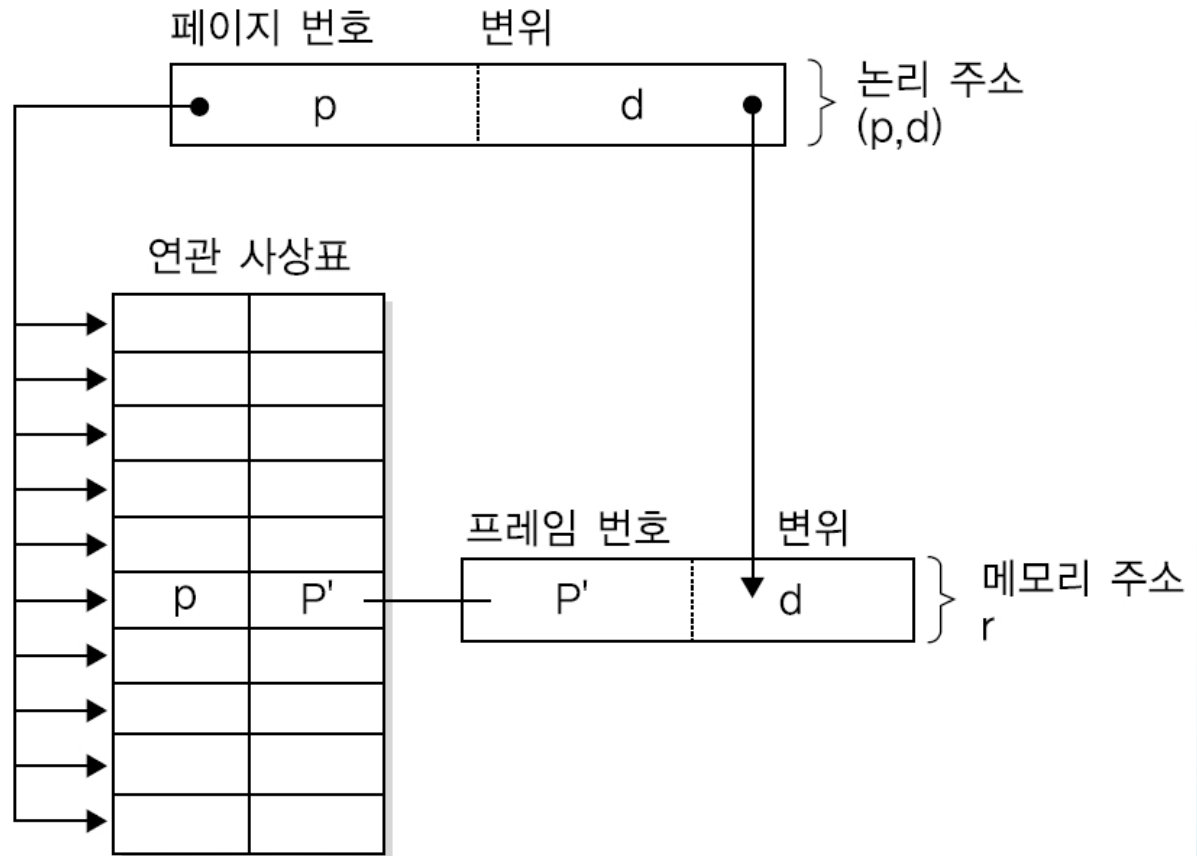
☑ **연관사상** (Associative Mapping)

- 연관메모리(사상표)의
키값과 동시비교

- 빠른검색 **우수**

- 메모리비용 **▲**

페이지 p를
찾기 위해
연관 메모리의
모든 항목을
동시에 조사.



분산 메모리 할당 - 페이징

■ 페이지 테이블의 구현

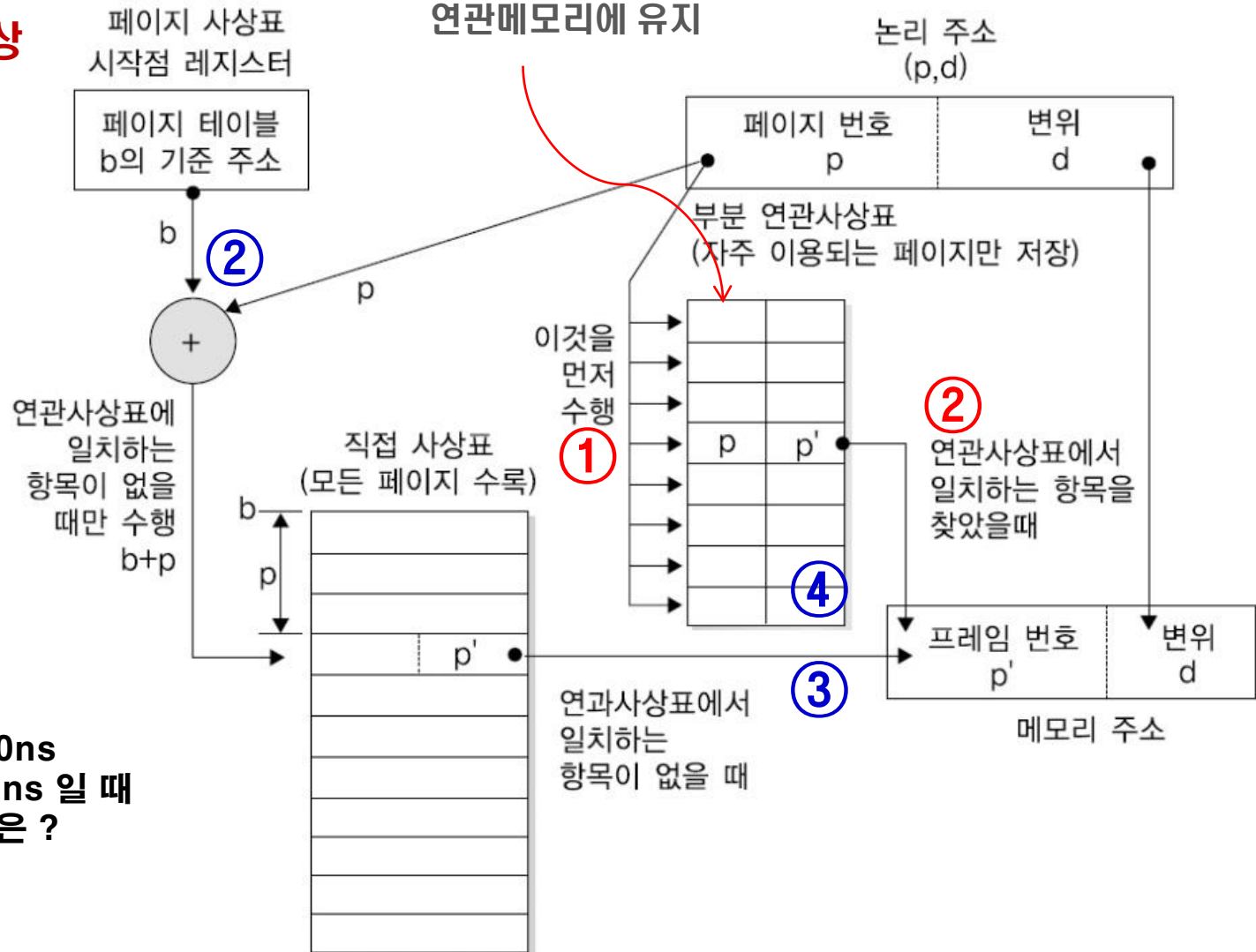
☑ 연관/직접 사상

찾았을때

① ②

못 찾았을때

② ③ ④



적중률(80%)

레지스터접근시간: 50ns

메모리접근시간: 750ns 일 때

메모리 유효접근시간은 ?

세그먼트 메모리 할당

페이지 매핑 테이블의 하드웨어비용

페이지 매핑 과정에서의 속도저하

내부단편화 문제는 여전히 존재

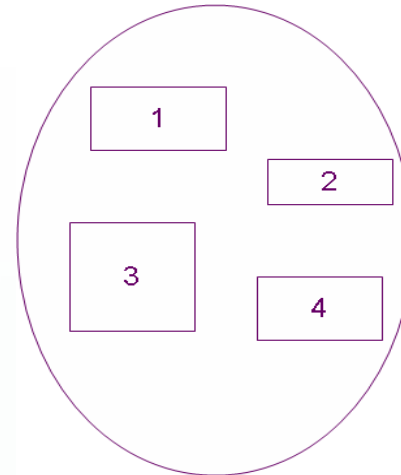
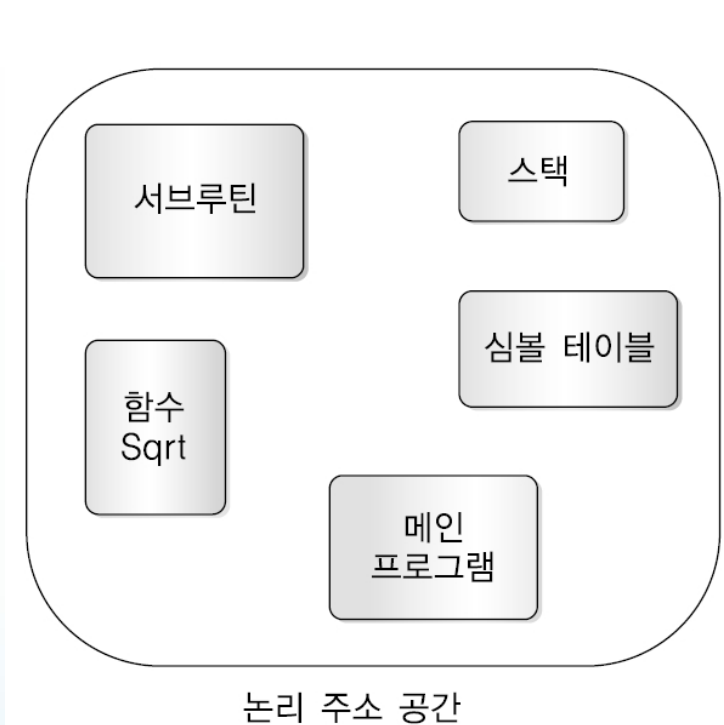
분산 메모리 할당 - 세그먼트기법

■ 세그먼트 메모리 관리

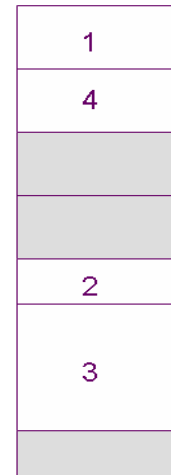
☑ 세그먼트' 단위 모임(크기 변함) vs 페이징(고정크기)

☑ 세그먼트 : 연관된 기능을 수행하는 하나의 모듈 프로그램

예) 서브루틴, 프로시저(procedure), 함수(function), 모듈(module) 등



사용자 공간



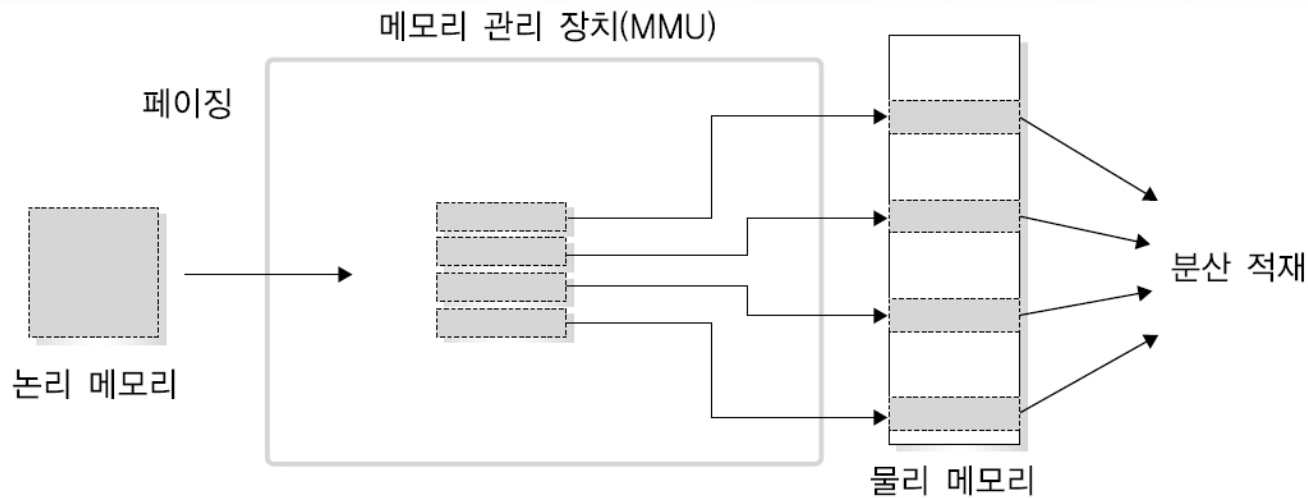
실제 메모리 공간

세그먼트 크기가 다르기 때문에
메모리가 페이지 프레임으로
나누어지지 않고 동적분할(가변 분할)
기법으로 메모리 할당

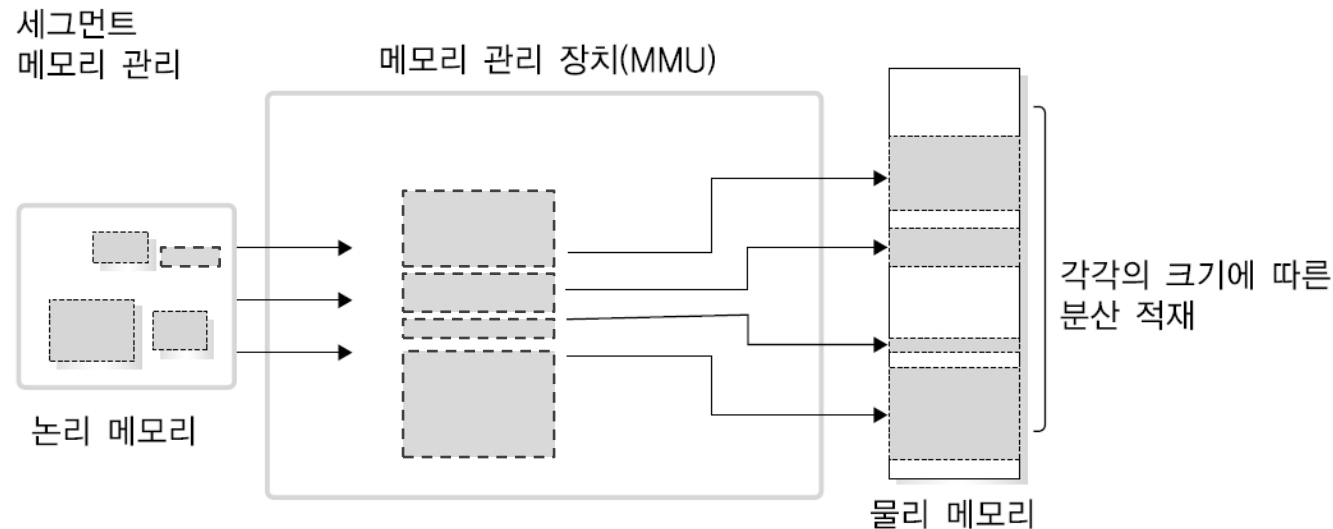
분산 메모리 할당 - 세그먼트기법

■ 세그먼트와 페이지의 비교

고정크기
(페이지)



가변크기
(세그먼트)

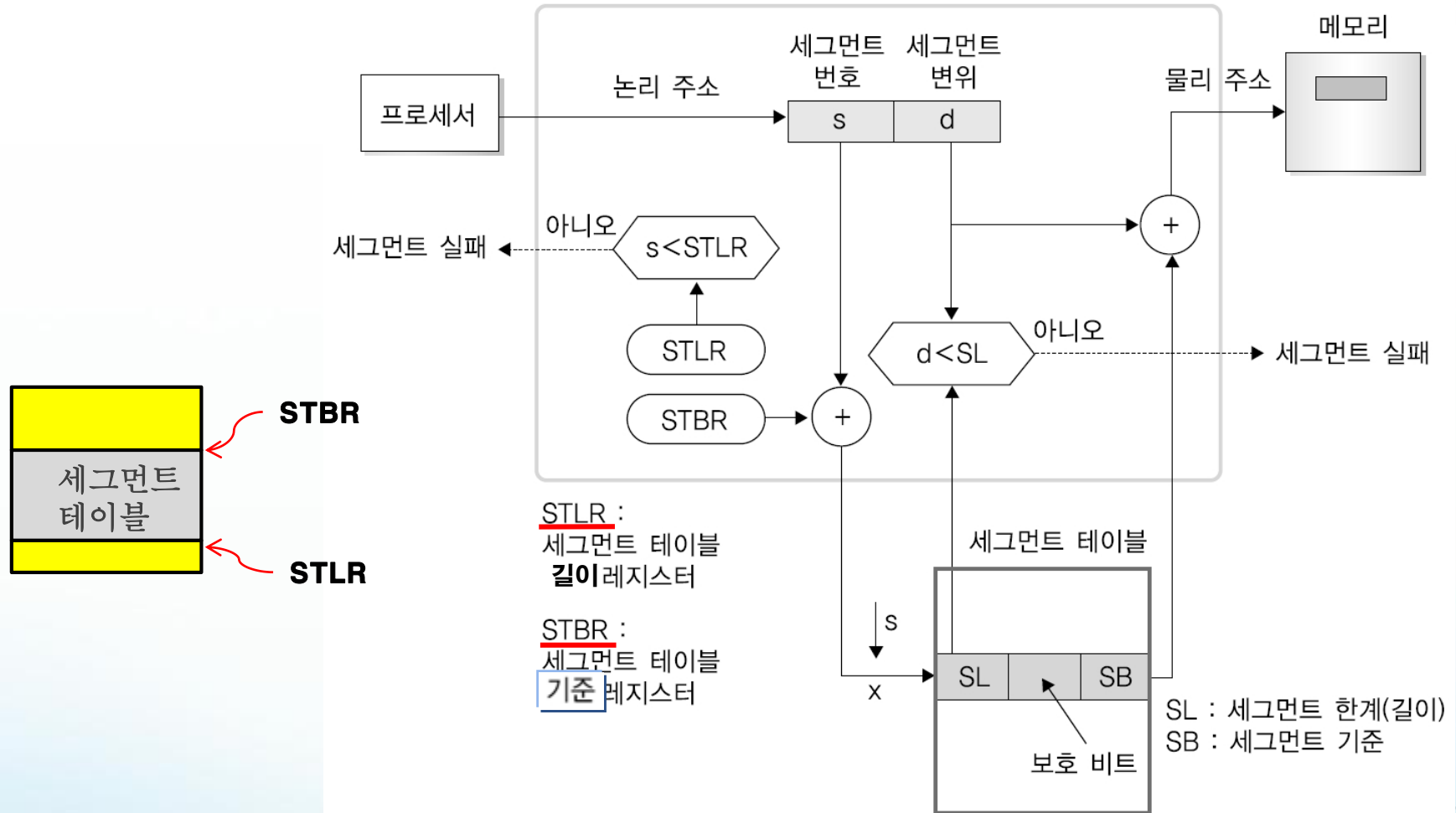


분산 메모리 할당 - 세그먼트기법

■ 세그먼트 주소사상(Mapping)

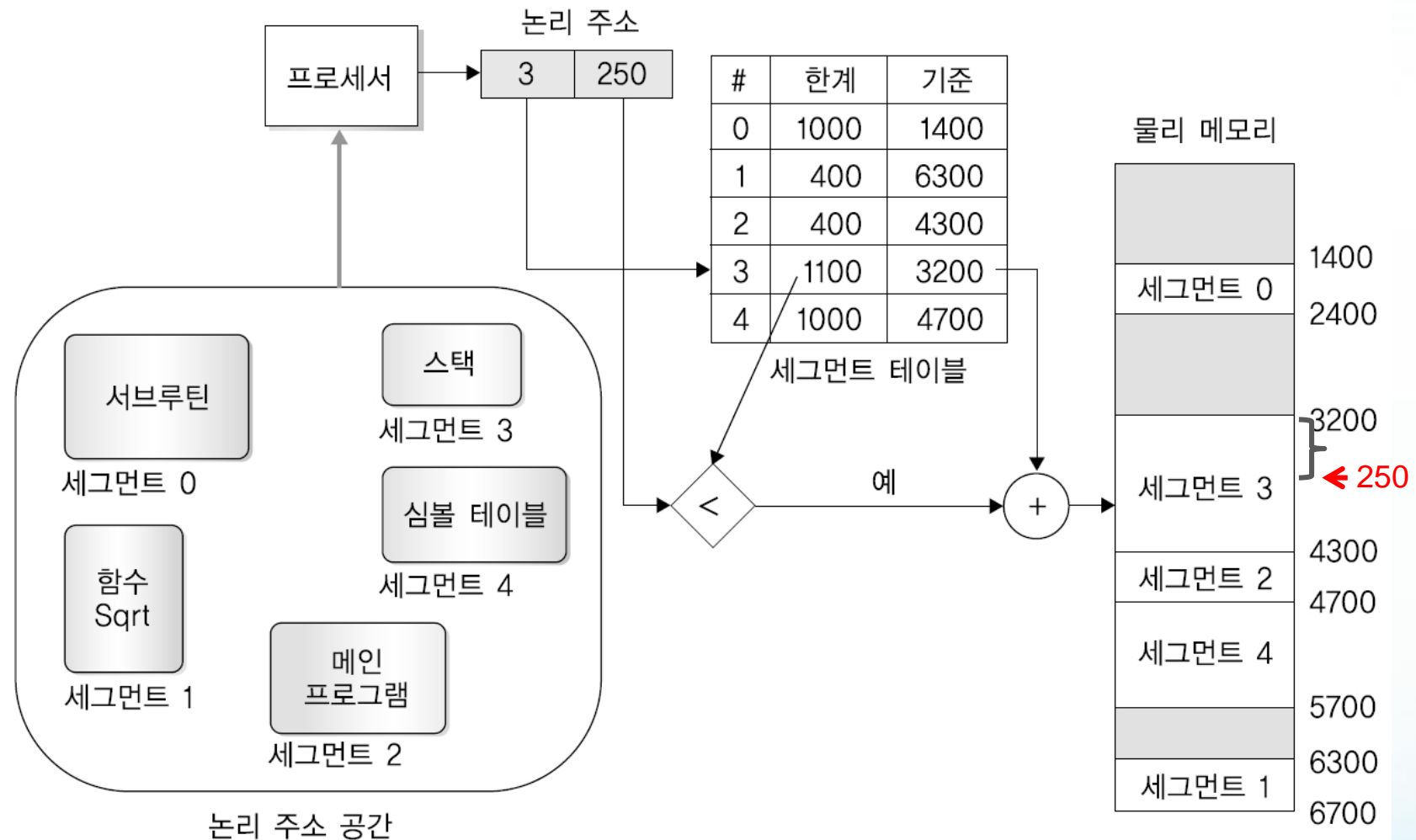
☑ 세그먼트의 논리주소(세그먼트번호 S, 변위(offset) d)

메모리 관리 장치(MMU)



분산 메모리 할당 - 세그먼트기법

■ 세그먼트 주소사상(Mapping) 예

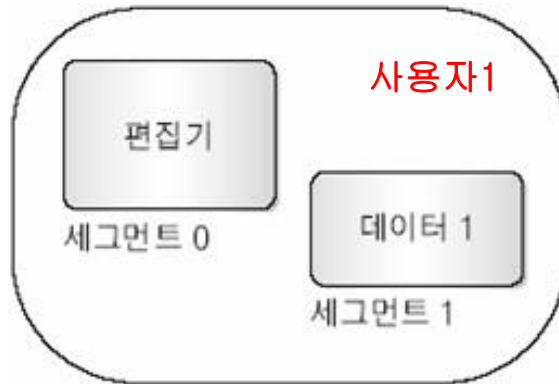


분산 메모리 할당 - 세그먼트기법

■ 세그먼트의 공유

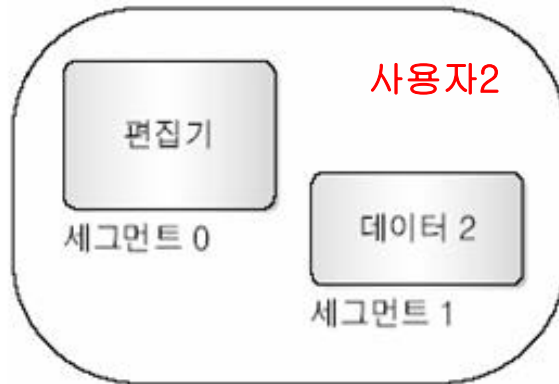
예) 편집기 세그먼트의 공유

세그먼트(43062)지정
해당 세그먼트를 공유



	한계	기준
0	25286	<u>43062</u>
1	4425	68348

사용자1
세그먼트 테이블



	한계	기준
0	25286	<u>43062</u>
1	8500	90063

사용자2
세그먼트 테이블



- 내부단편화 없음(가변분할), 외부단편화 있음 -> 압축으로 해결

분산 메모리 할당 - 페이지화된 세그먼트기법

■ 페이지화된 세그먼트(Paged Segment) 기법

페이징 - 내부 단편화 현상 + 세그멘테이션 - 외부 단편화 현상

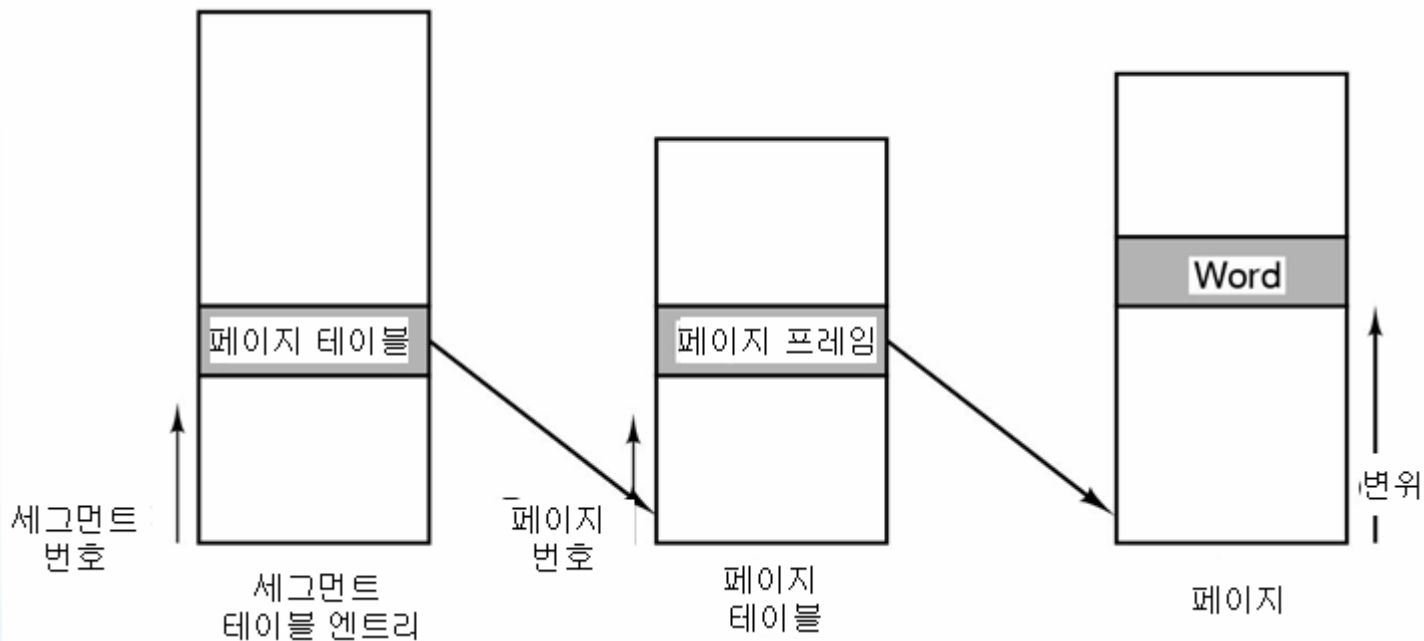
메모리 효율적 사용

가변적인 자료 구조와 모듈 처리

세그먼트를 페이징하는 방법 - 장단점을 해결

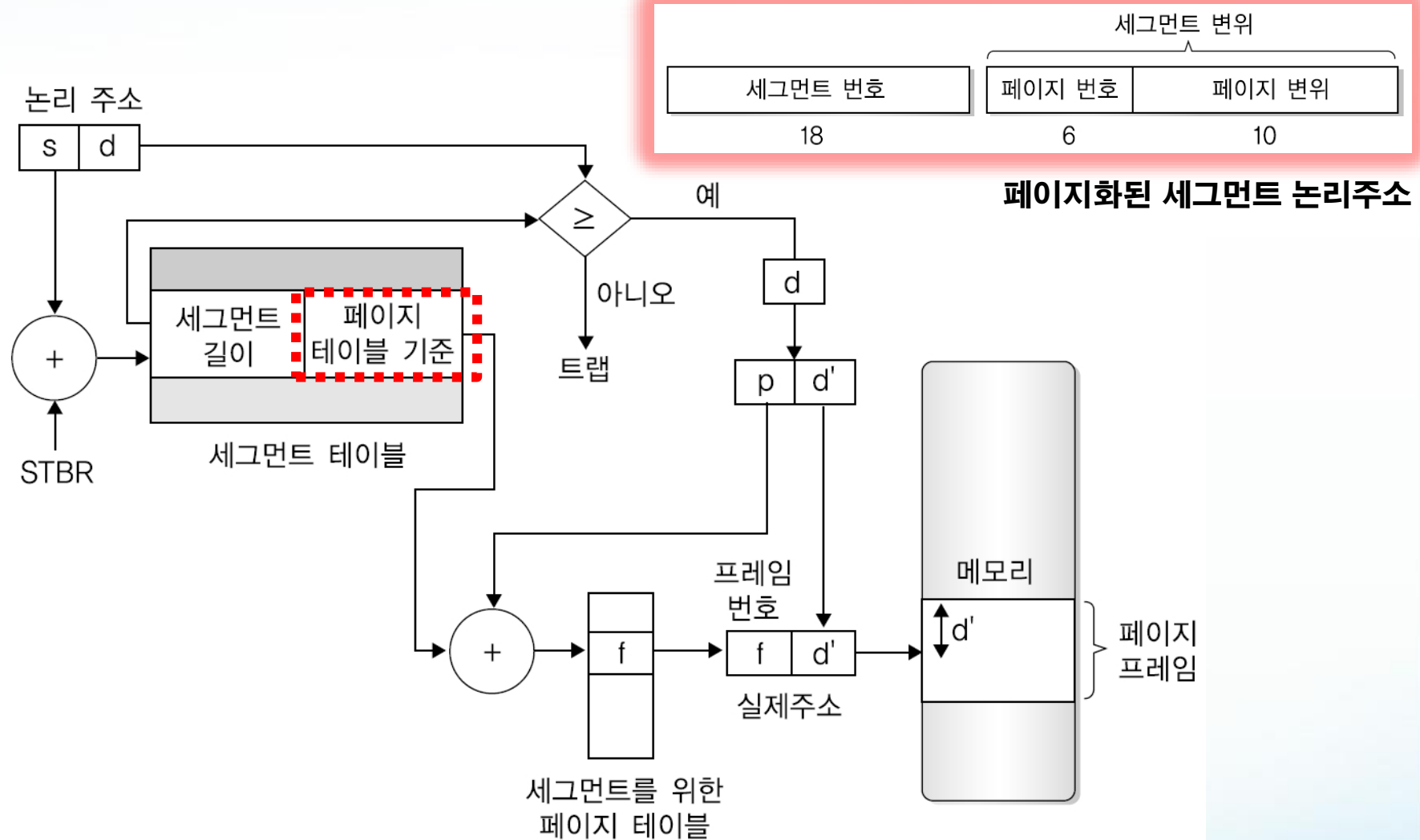
페이징 기법은 외부 단편화 문제를 제거하면서 할당 과정을 쉽게 해결

결합된 구조는 Multics 시스템에서 사용, Intel 386 계열에서 사용



분산 메모리 할당 - 페이지화된 세그먼트기법

■ 페이지화된 세그먼트의 주소 매핑



Q&A