



9강. 가상 메모리 I

방송대 컴퓨터과학과
김진욱 교수

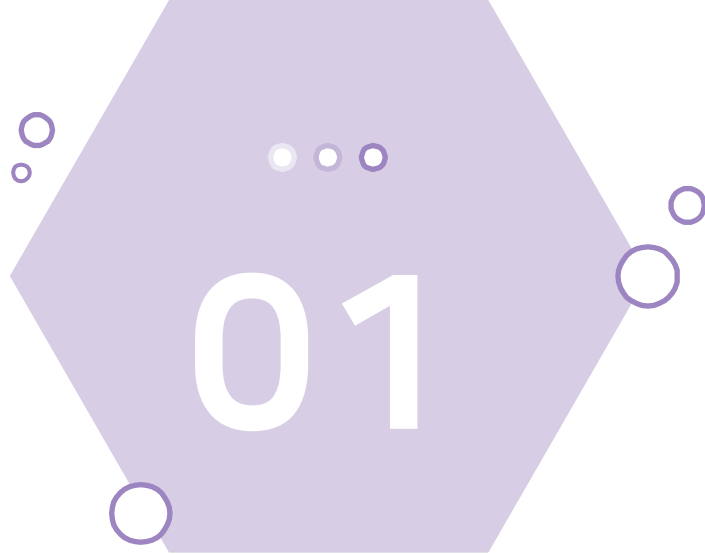


목차

01 가상 메모리의 개념

02 블록 단위 주소 변환

03 페이지 호출기법

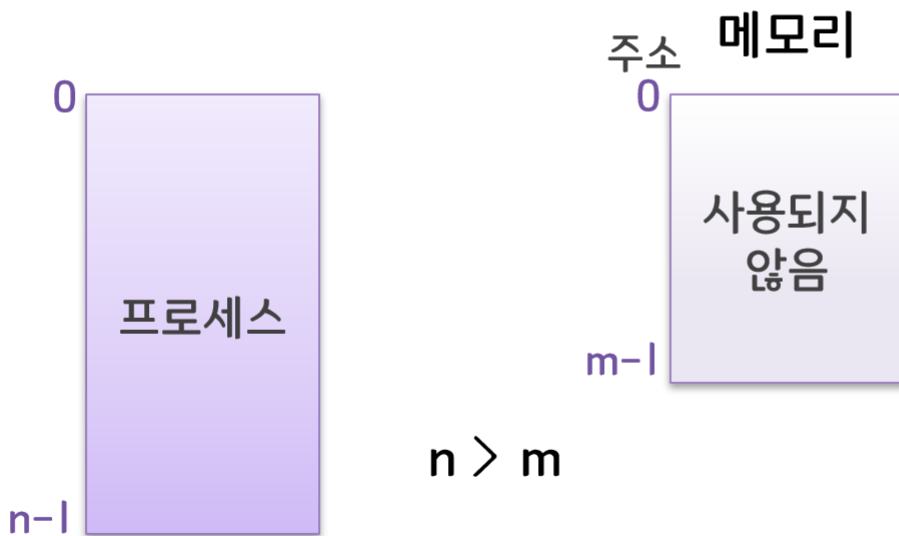


가상 메모리의 개념

가상 메모리의 개념

가상 메모리

- 메모리 크기보다 더 큰 기억공간이 필요한 프로세스를 실행할 수 있게 하는 방법

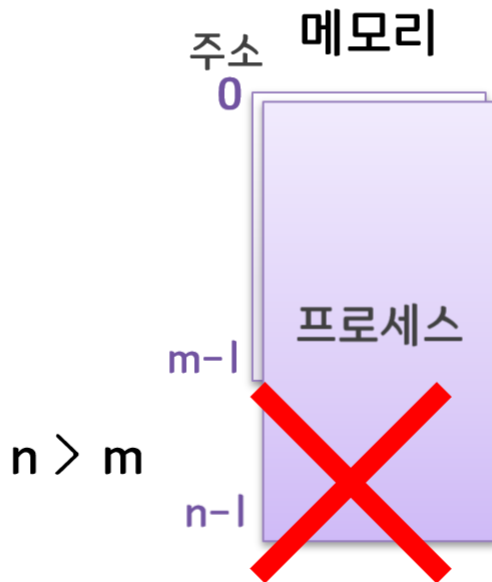


연속 메모리 할당

가상 메모리의 개념

가상 메모리

- 메모리 크기보다 더 큰 기억공간이 필요한 프로세스를 실행할 수 있게 하는 방법



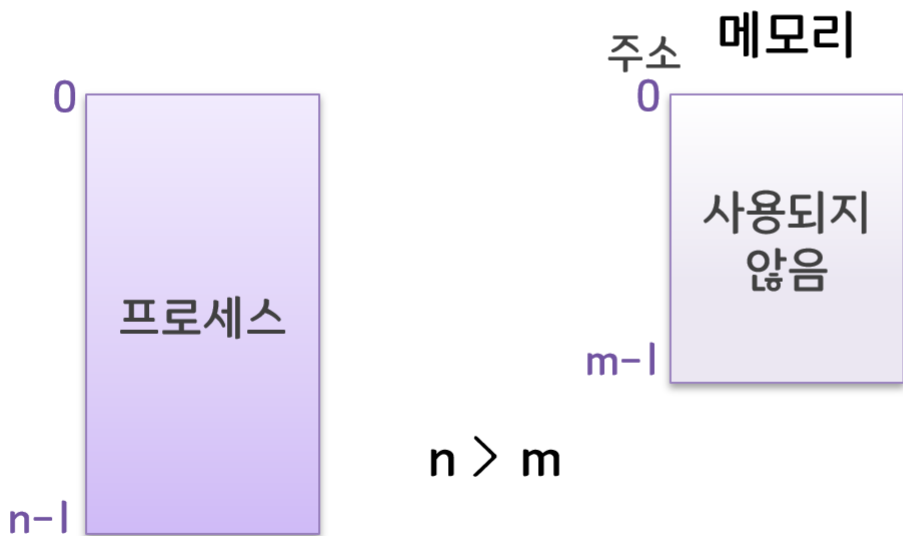
연속 메모리 할당

: 프로세스 실행 불가

가상 메모리의 개념

가상 메모리

- 메모리 크기보다 더 큰 기억공간이 필요한 프로세스를 실행할 수 있게 하는 방법

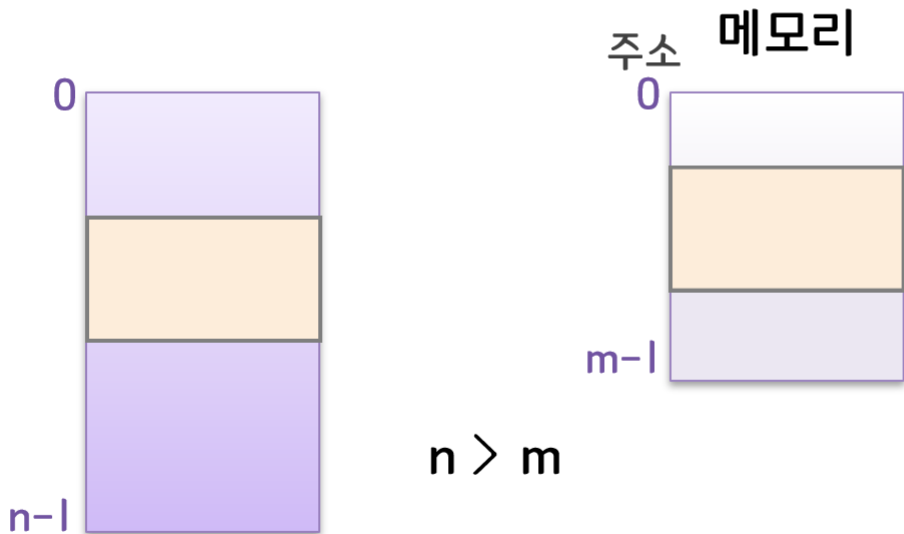


가상 메모리

가상 메모리의 개념

가상 메모리

- 메모리 크기보다 더 큰 기억공간이 필요한 프로세스를 실행할 수 있게 하는 방법



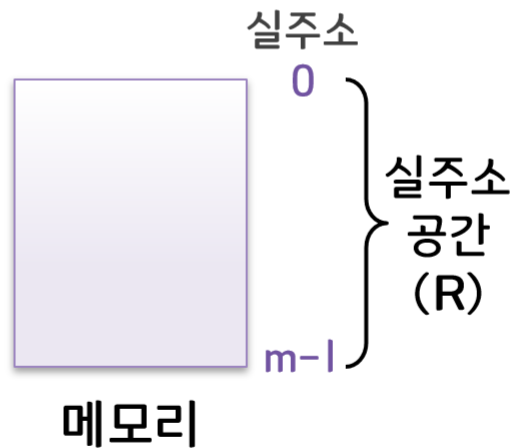
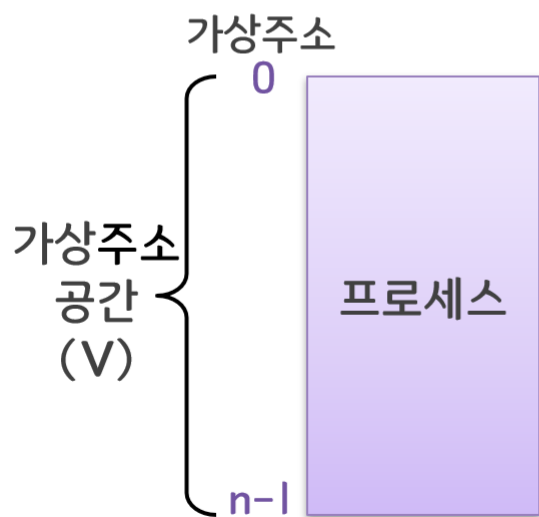
가상 메모리

: 프로세스 실행 가능

가상 메모리의 개념

가상 메모리

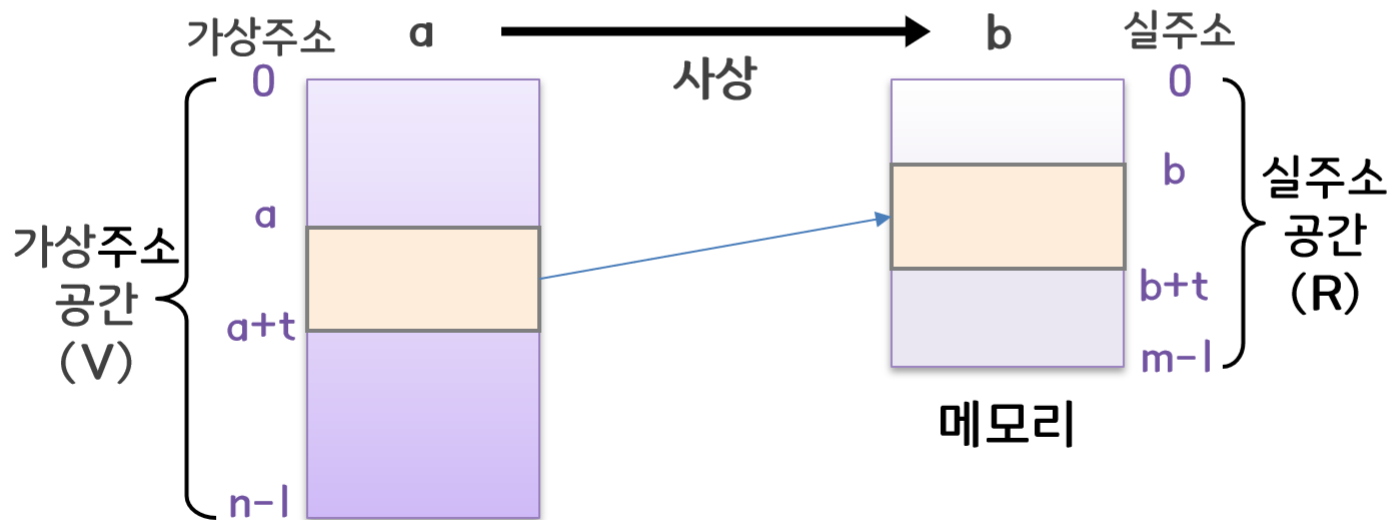
- 프로세스에 의해 참조되는 주소를 메모리에서 사용하는 주소와 분리
- 현재 필요한 일부만 메모리에 적재



가상 메모리의 개념

가상 메모리

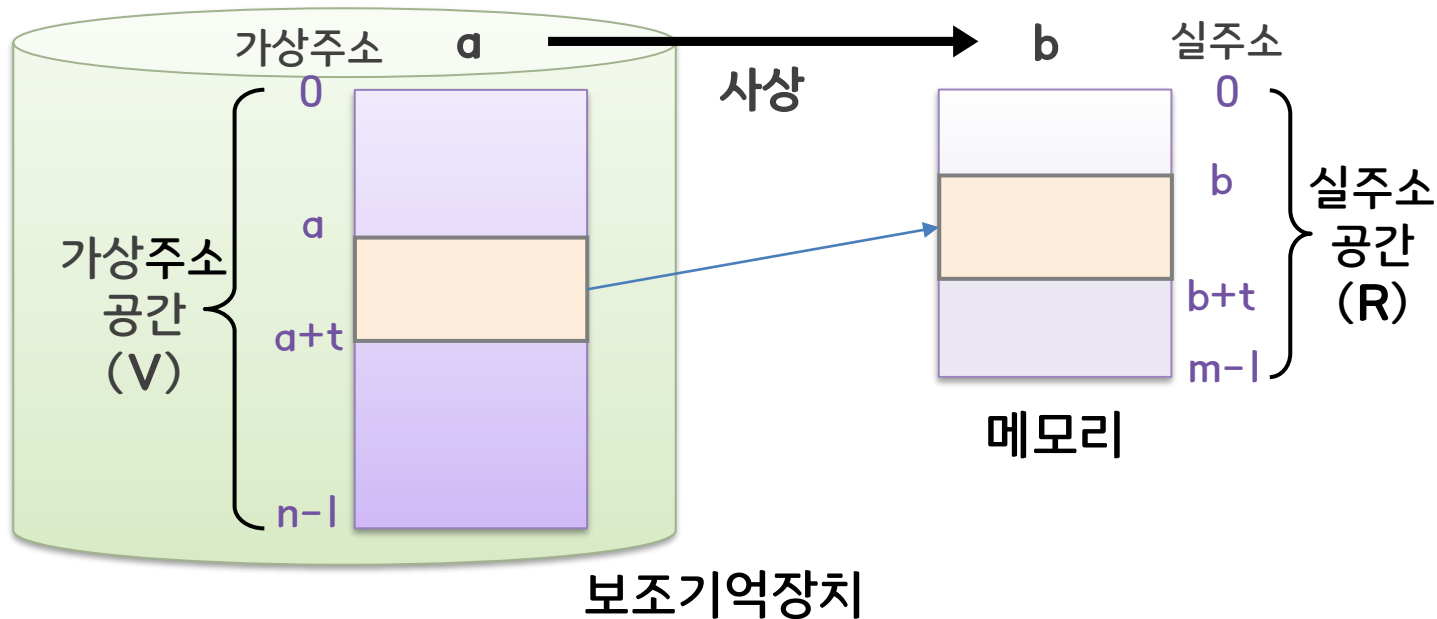
- 프로세스에 의해 참조되는 주소를 메모리에서 사용하는 주소와 분리
- 현재 필요한 일부만 메모리에 적재



가상 메모리의 개념

가상 메모리

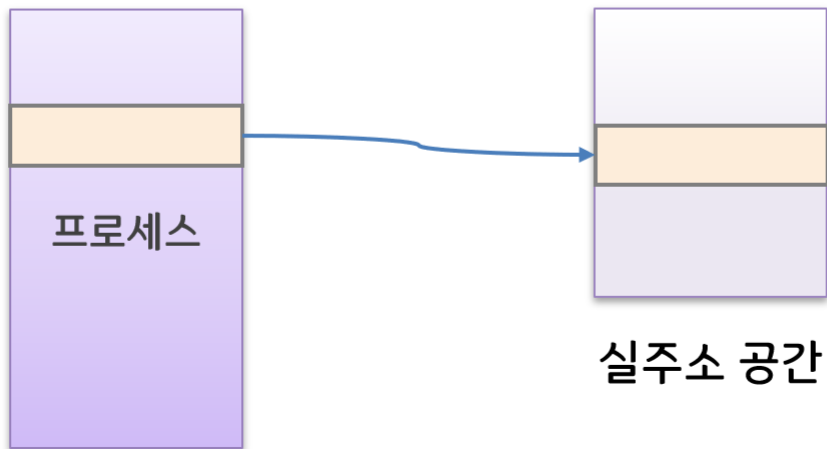
- 프로세스에 의해 참조되는 주소를 메모리에서 사용하는 주소와 분리
- 현재 필요한 일부만 메모리에 적재



가상 메모리의 개념

■ 사상(mapping)

- 프로세스 실행을 위해 가상주소를 실주소로 변환

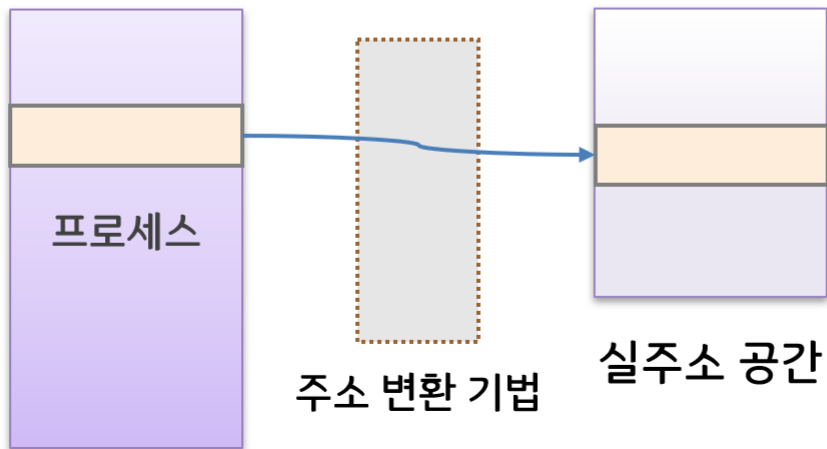


가상 메모리의 개념

■ 사상(mapping)

- 프로세스 실행을 위해 가상주소를 실주소로 변환
- 동적 주소 변환(DAT): 프로세스가 실행되는 동안 사상
- 인위적 연속성

» 가상주소 공간에서는 연속이지만 실주소 공간에서도 연속일 필요는 없음

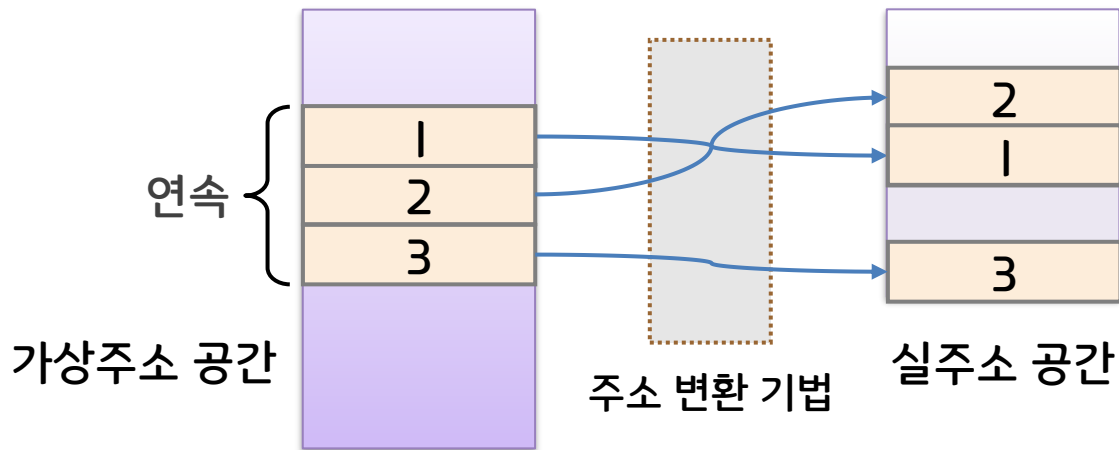


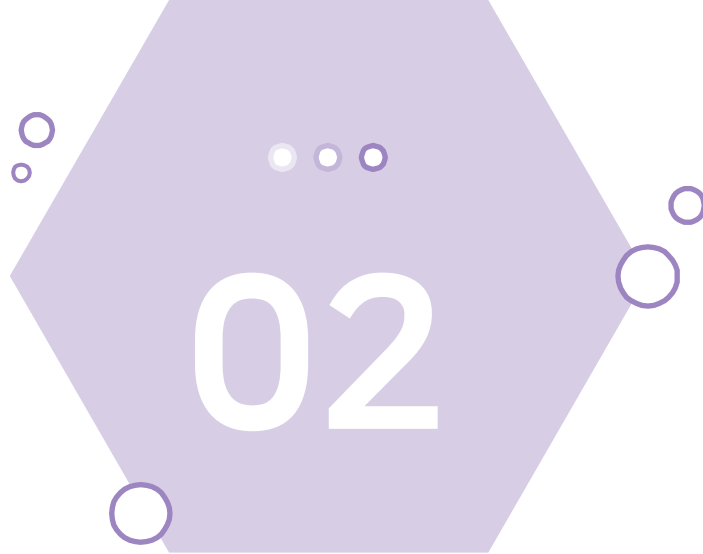
가상 메모리의 개념

■ 사상(mapping)

- 프로세스 실행을 위해 가상주소를 실주소로 변환
- 동적 주소 변환(DAT): 프로세스가 실행되는 동안 사상
- 인위적 연속성

» 가상주소 공간에서는 연속이지만 실주소 공간에서도 연속일 필요는 없음



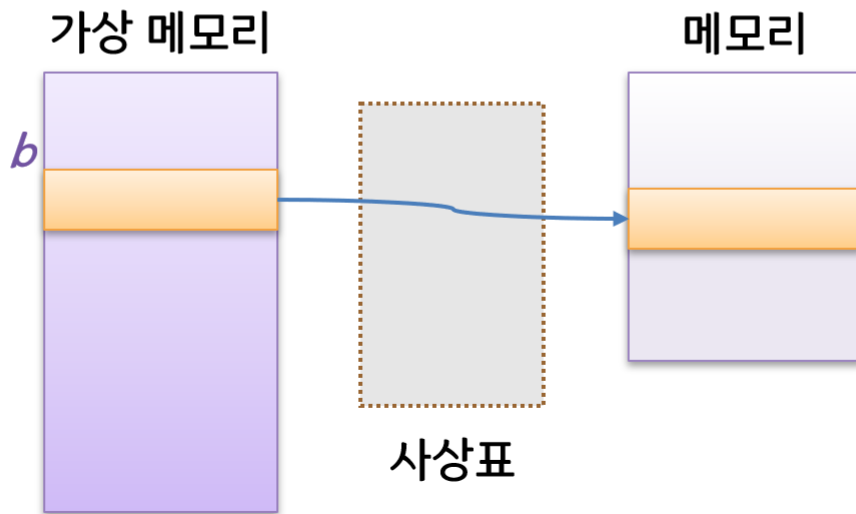


블록 단위 주소 변환

주소 변환

■ 블록 단위 주소 변환

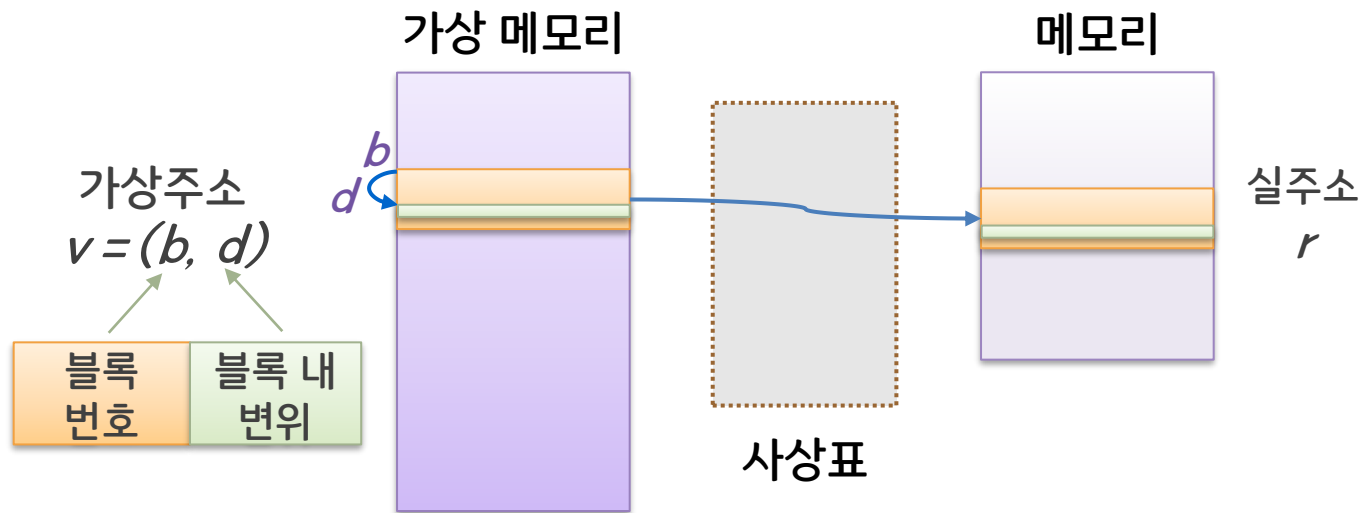
- 블록 단위로 분류하여 각 블록이 메모리의 어디에 위치하는지를 관리
- 블록의 크기
 - » 작으면 사상정보 ↑, 크면 블록 전송시간 ↑ 및 적재할 프로세스 수 ↓



주소 변환

■ 블록 단위 주소 변환

- 블록 단위로 분류하여 각 블록이 메모리의 어디에 위치하는지를 관리
- 블록의 크기
 - » 작으면 사상정보 ↑, 크면 블록 전송시간 ↑ 및 적재할 프로세스 수 ↓

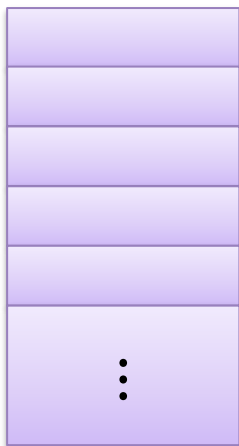


블록 단위 주소 변환

■ 블록 구성 방식에 따른 분류

- 페이징 기법: 블록의 크기가 동일한 페이지로 구성
- 세그먼테이션 기법: 블록의 크기가 서로 다른 세그먼트로 구성

가상 메모리



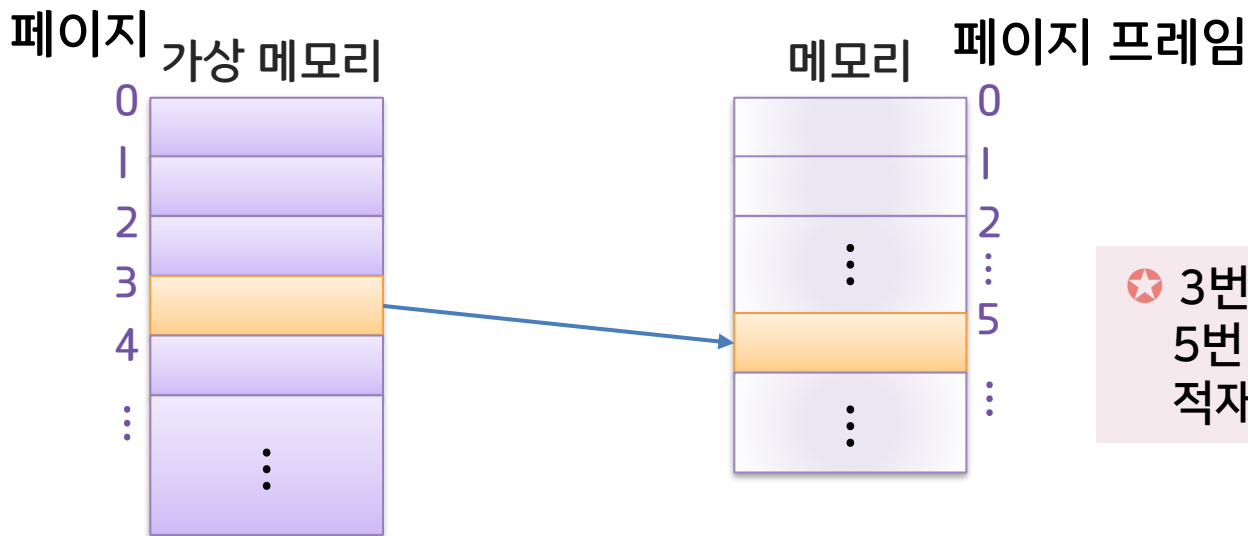
가상 메모리



○페이징 기법

■ 페이지와 페이지 프레임

- 가상 메모리를 고정된 크기의 블록인 페이지 단위로 나누어 관리
- 메모리 영역도 페이지와 동일한 크기의 블록인 페이지 프레임으로 나눔

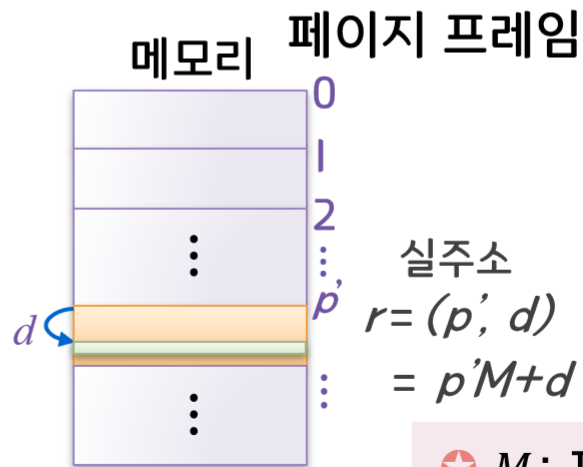
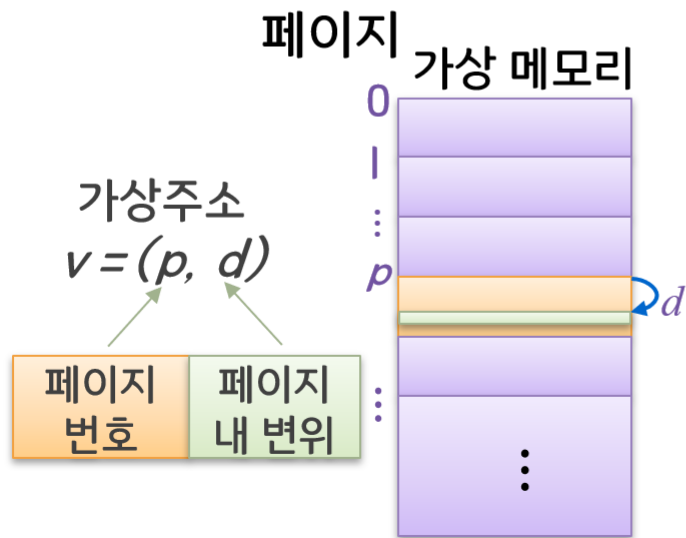


★ 3번 페이지를
5번 페이지 프레임에
적재

페이징 기법

■ 페이지 사상표

- 가상주소를 실주소로 동적 변환하기 위해 필요
- 가상주소의 페이지 번호에 대한 실주소의 페이지 프레임 번호를 저장

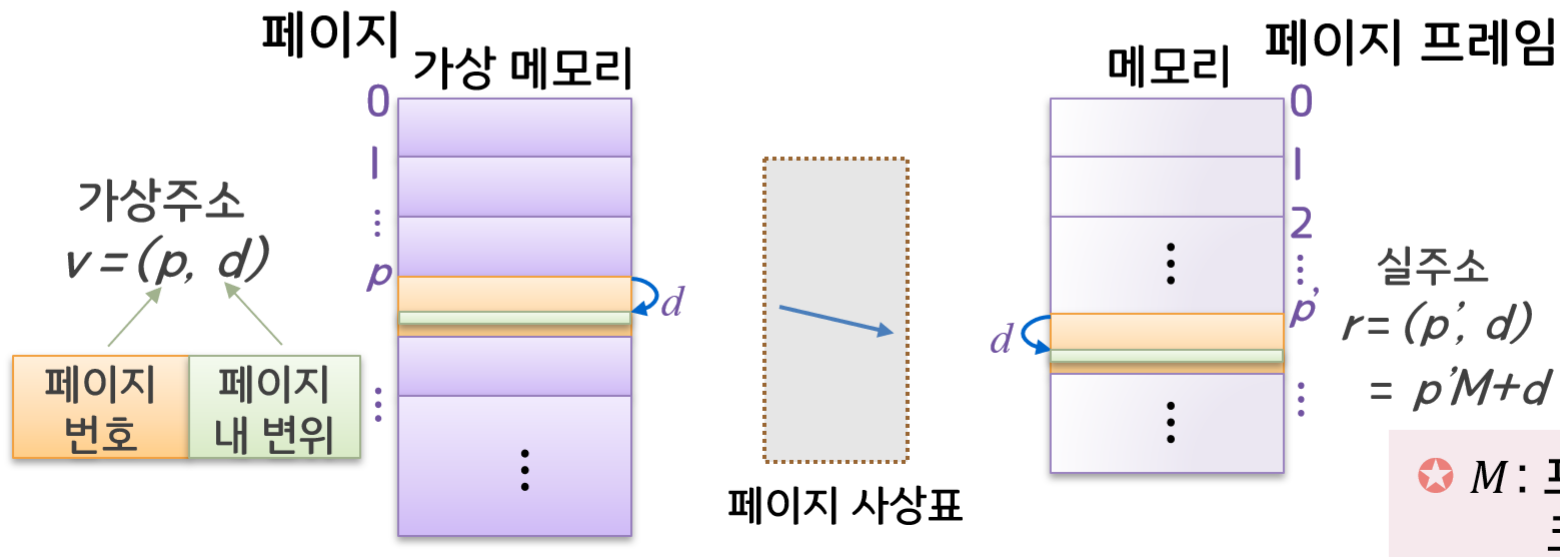


★ M : 프레임의 크기

페이징 기법

■ 페이지 사상표

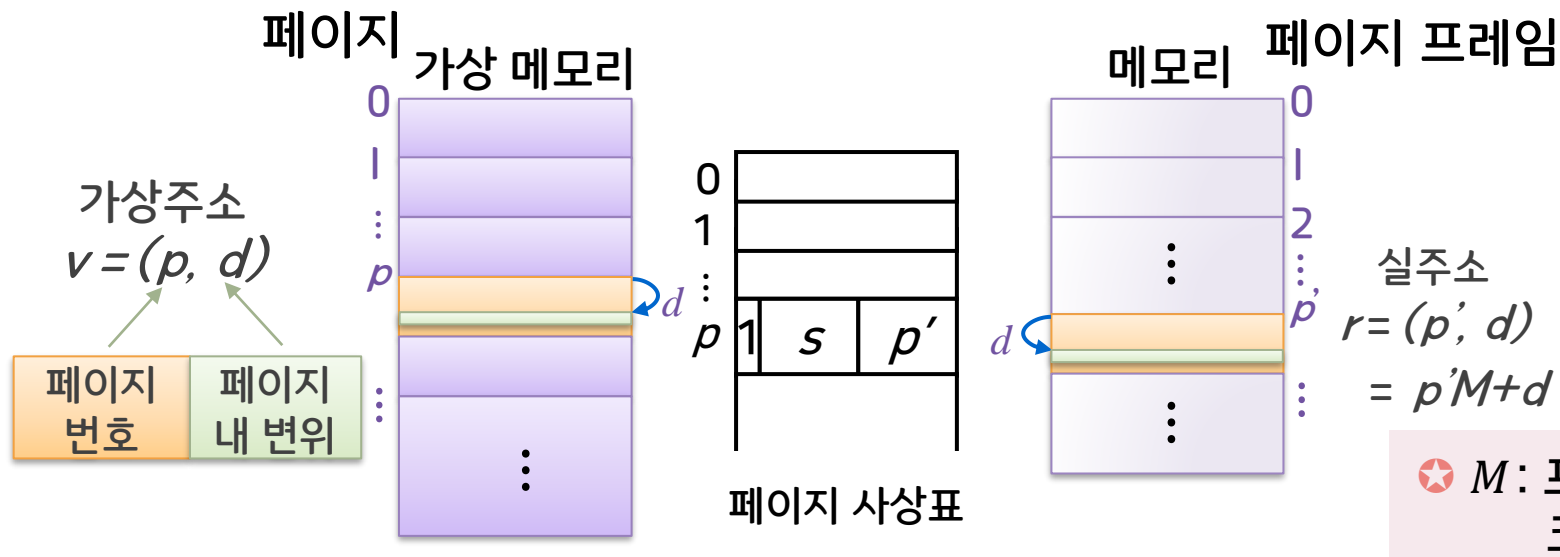
- 가상주소를 실주소로 동적 변환하기 위해 필요
- 가상주소의 페이지 번호에 대한 실주소의 페이지 프레임 번호를 저장



페이징 기법

■ 페이지 사상표

- 가상주소를 실주소로 동적 변환하기 위해 필요
- 가상주소의 페이지 번호에 대한 실주소의 페이지 프레임 번호를 저장



페이지 기법

■ 페이지 사상표

| 페이지번호 | 페이지 존재 비트 | 보조기억장치 주소 | 페이지 프레임 번호 |
|-------|-----------|-----------|------------|
| 0 | ⋮ | ⋮ | ⋮ |
| 1 | ⋮ | ⋮ | ⋮ |
| 2 | ⋮ | ⋮ | ⋮ |
| 3 | 1 | s | 5 |
| ⋮ | ⋮ | ⋮ | ⋮ |

- 직접 사상 : 페이지 사상표를 직접 이용
- 연관 사상 : 연관기억장치에 저장한 연관 사상표를 이용

★ 연관기억장치

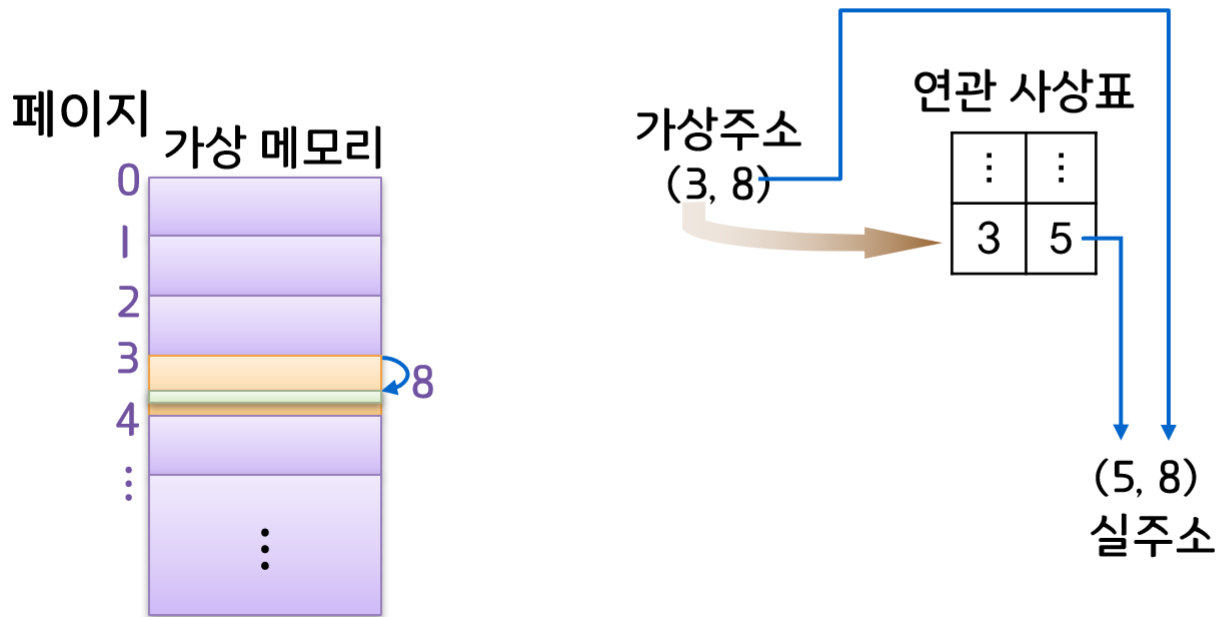
저장된 값으로 데이터를 액세스하는 고속 메모리 장치

| 페이지 번호 | 페이지 프레임 번호 |
|--------|------------|
| ⋮ | ⋮ |
| 3 | 5 |

페이징 기법

■ 연관/직접 사상

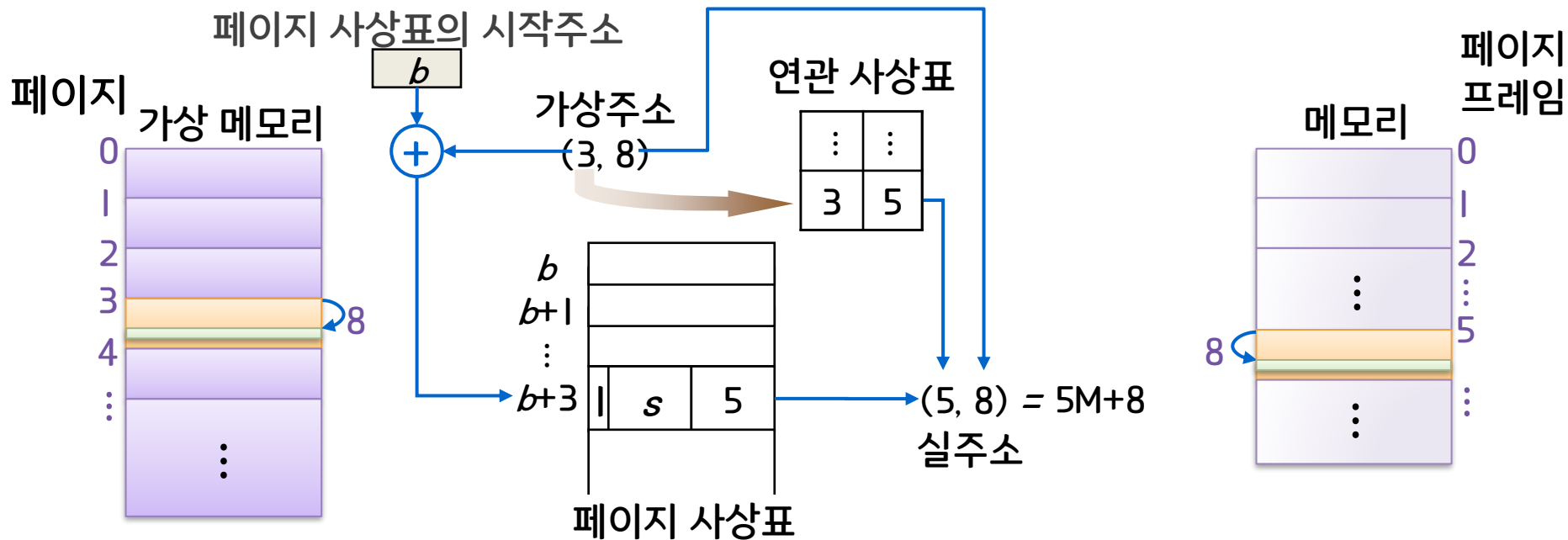
- 연관 사상표에는 가장 최근에 참조된 페이지들만 보관, 나머지는 페이지 사상표에 보관



페이징 기법

■ 연관/직접 사상

- 연관 사상표에는 가장 최근에 참조된 페이지들만 보관,
나머지는 페이지 사상표에 보관



○페이징 기법

■ 페이징 기법의 특징

- 논리적 의미와 무관하게 동일 크기의 페이지로 가상 메모리를 나눔
- 프로세스 사이의 메모리 보호는 페이지 단위로 이루어짐
- 외부 단편화는 발생하지 않으나 내부 단편화는 발생할 수 있음

○ 세그멘테이션 기법

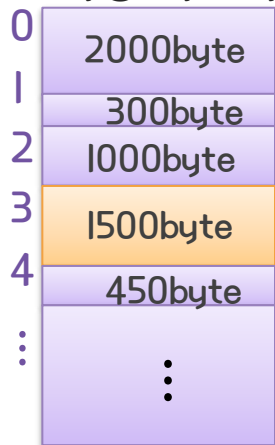
■ 세그먼트

- 가상 메모리를 논리적 의미에 맞는 다양한 크기의 세그먼트 단위로 나누어 관리
- 가상주소 $v = (s, d)$

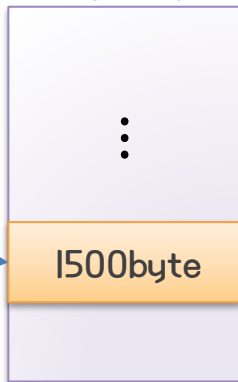
세그먼트 번호

세그먼트 내 변위

세그먼트 가상 메모리



메모리



세그먼트 사상표

★ 3번 세그먼트를
메모리의 사용 가능한
영역에 적재

→ 메모리 배치기법 이용

○ 세그멘테이션 기법

■ 세그먼트 사상표

| 세그먼트 번호 | 세그먼트 존재 비트 | 보조기억장치 주소 | 세그먼트 길이 | 세그먼트 시작주소 |
|------------|---------------|--------------|------------|--------------|
| 0 | ⋮ | ⋮ | 300 | ⋮ |
| 1 | ⋮ | ⋮ | 1000 | ⋮ |
| 2 | ⋮ | ⋮ | 500 | ⋮ |
| 3 | 1 | s | 1500 | 1200 |
| ⋮ | ⋮ | ⋮ | 2000 | ⋮ |

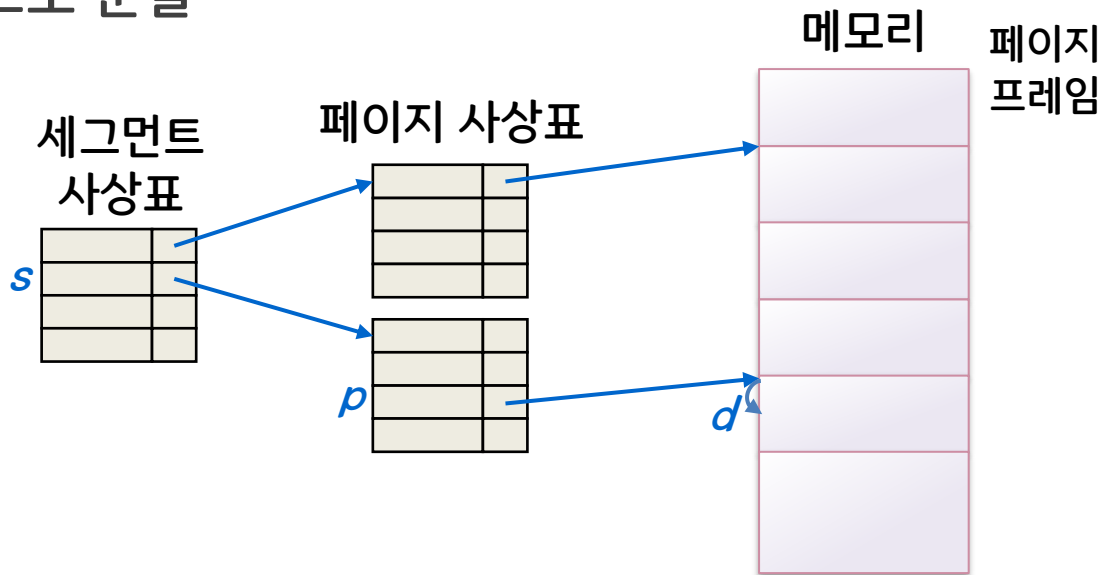
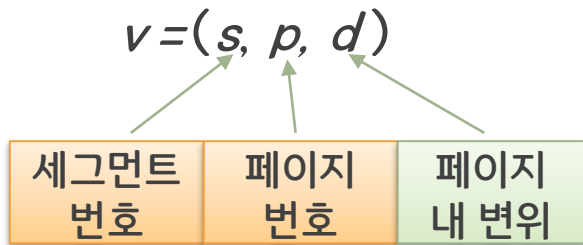
- 세그먼트 시작주소: 메모리에서의 시작위치
- 세그먼트 길이: 오버플로 확인용

○페이징/세그먼테이션 혼용기법

■ 두 기법 혼합

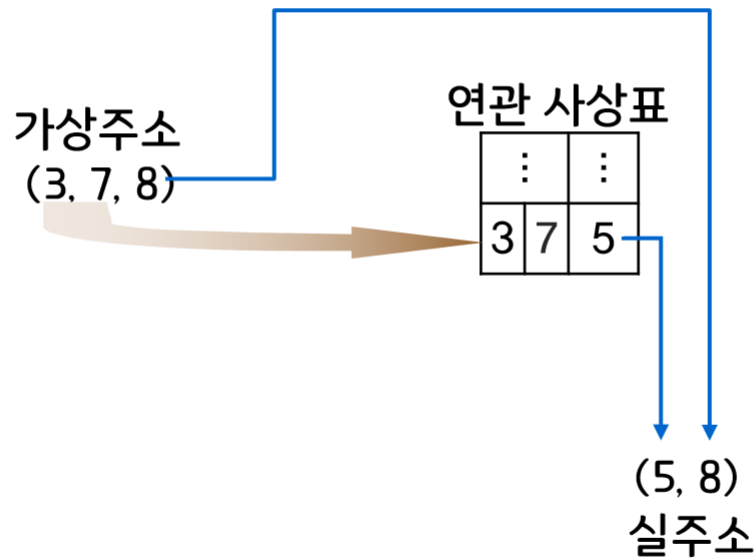
- 세그먼테이션 기법의 논리적 장점 + 페이징 기법의 메모리 관리 장점
- 가상 메모리를 세그먼트 단위로, 각 세그먼트를 다시 페이지 단위로 분할
- 메모리는 페이지 프레임으로 분할

• 가상주소



페이징/세그먼테이션 혼용기법

■ 동적 주소 변환 예



페이징/세그먼테이션 혼용기법

동적 주소 변환 예

세그먼트 사상표의
시작주소

b

+

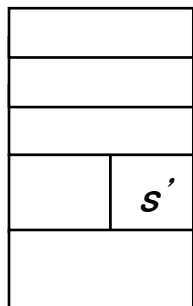
가상주소

$(3, 7, 8)$

연관 사상표

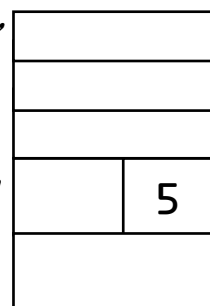
| | |
|---|-----|
| ⋮ | ⋮ |
| 3 | 7 5 |

b
 $b+1$
⋮
 $b+3$



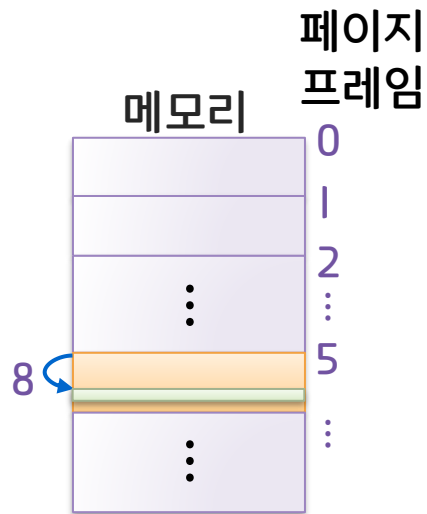
세그먼트 사상표

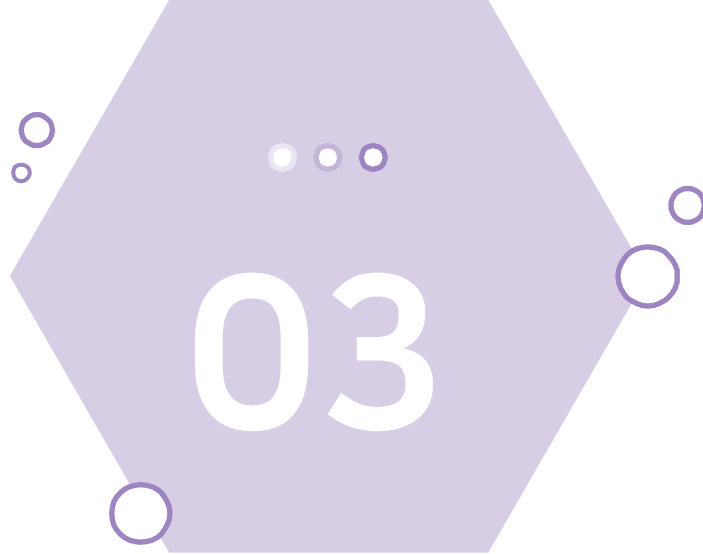
s'
 $s'+1$
⋮
 $s'+7$



페이지 사상표

$(5, 8) = 5M+8$
실주소





페이지 호출기법

◉ 메모리 호출기법

■ 메모리 호출기법

- 페이지를 어느 시점에 메모리에 적재할 것인가를 결정
- 종류: 요구 페이지 호출기법, 예상 페이지 호출기법

메모리 호출기법

■ 요구 페이지 호출기법

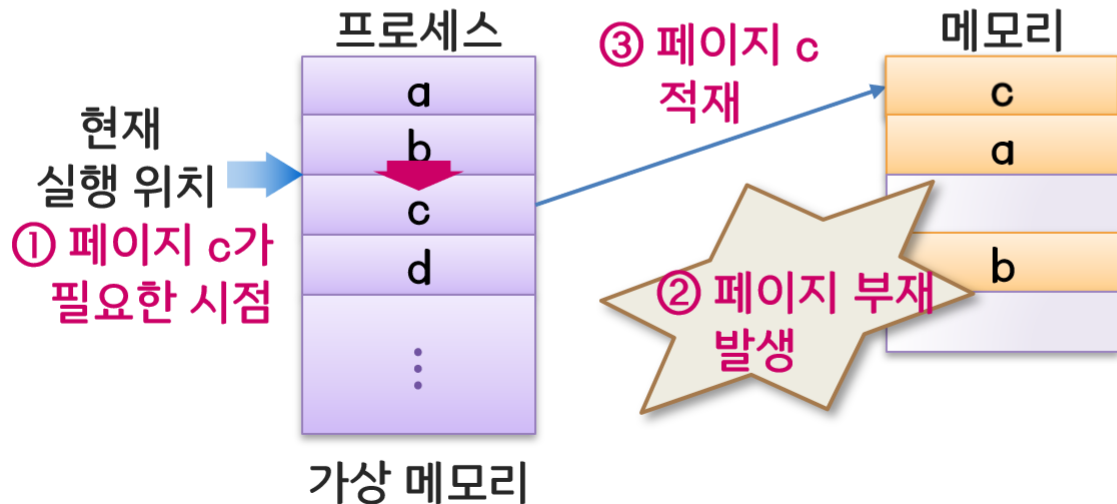
- 한 프로세스의 페이지 요구가 있을 때 요구된 페이지를 메모리로 이동
- 즉, 명령어나 데이터가 실제로 참조되면 해당 페이지를 메모리에 적재



메모리 호출기법

■ 요구 페이지 호출기법

- 한 프로세스의 페이지 요구가 있을 때 요구된 페이지를 메모리로 이동
- 즉, 명령어나 데이터가 실제로 참조되면 해당 페이지를 메모리에 적재



메모리 호출기법

■ 예상 페이지 호출기법

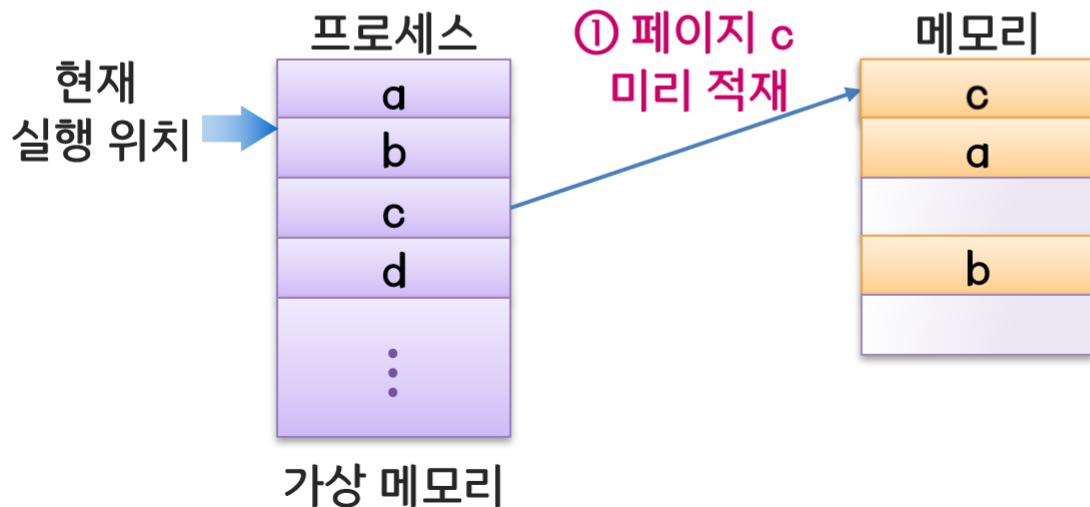
- 현재 요구되지는 않지만 곧 사용될 것으로 예상되는 페이지를 미리 메모리로 이동
- 실제 필요한 시점이 되었을 때 프로세스 실행이 단절되지 않음



메모리 호출기법

■ 예상 페이지 호출기법

- 현재 요구되지는 않지만 곧 사용될 것으로 예상되는 페이지를 미리 메모리로 이동
- 실제 필요한 시점이 되었을 때 프로세스 실행이 단절되지 않음



메모리 호출기법

■ 예상 페이지 호출기법

- 현재 요구되지는 않지만 곧 사용될 것으로 예상되는 페이지를 미리 메모리로 이동
- 실제 필요한 시점이 되었을 때 프로세스 실행이 단절되지 않음



메모리 호출기법

■ 요구 페이지 호출기법

- 옮길 페이지를 결정하는데 오버헤드를 최소화
- 메모리에 옮겨진 페이지는 모두 프로세스에 의해 실제로 참조된 것임
- 프로세스 시작 시점에는 프로세스 진행에 따라 연속적으로 페이지 부재 발생 (성능 저하)

■ 예상 페이지 호출기법

- 예상이 잘못된 경우 메모리 공간 낭비
- 프로세스 시작 시점에 적용하면 성능이 개선됨



강의를 마쳤습니다.

다음시간에는
10강. 가상 메모리 II