

Mermory Management(1)

<https://youtu.be/2rIXA8Qki9k>

메모리 관리의 개요

연속 메모리 할당

메모리 관리의 개념과 정책

메모리 관리의 개념

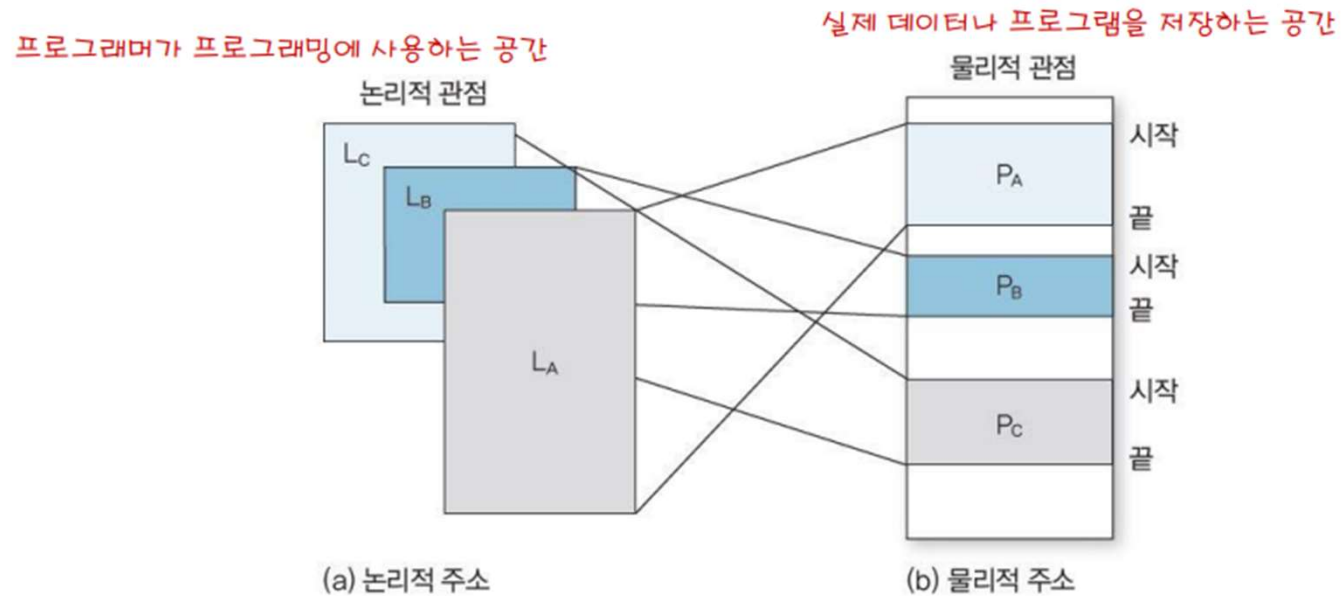
- 메모리 관리는 프로세스들을 위해 메모리 할당, 제거, 보호하는 활동.
- 디스크에 있는 프로그램을 실행하려면 먼저 메모리에 적재 후 메모리 관리자가 예약된 메모리에 할당해 줌.
- 다중 프로그래밍 시스템에서 여러 프로세스가 메모리에 상주할 수 있도록 운영체제가 동적으로 메모리 세분

메모리 관리 정책

- 적재(Fetch): 디스크에서 메모리로 프로세스 반입 시기 결정하는 것
 - 요구 적재: 운영체제나 시스템 프로그램, 사용자 프로그램 등 참조 요청에 따라 다음에 실행할 프로세스를 메모리에 적재하는 오래된 방법
 - 예상적재: 시스템의 요청을 미리 예측하여 메모리에 적재하는 방법
- 배치(Placement): 디스크에서 반입한 프로세스를 메모리 어느 위치에 저장할 것인지 결정
- 대치(Replacement): 메모리가 충분하지 않을 때 현재 메모리에 적재된 프로세스 중 제거할 프로세스를 결정하는 방법.

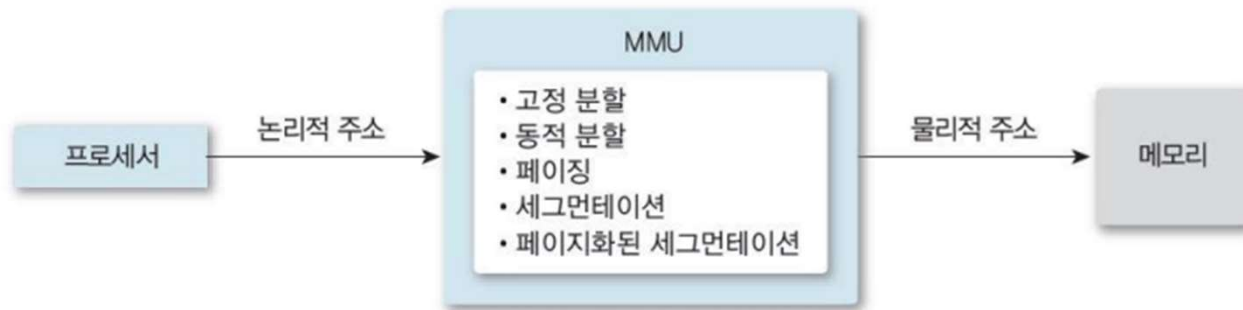
메모리의 구조와 매핑

메모리의 구조



메모리의 구조와 매핑

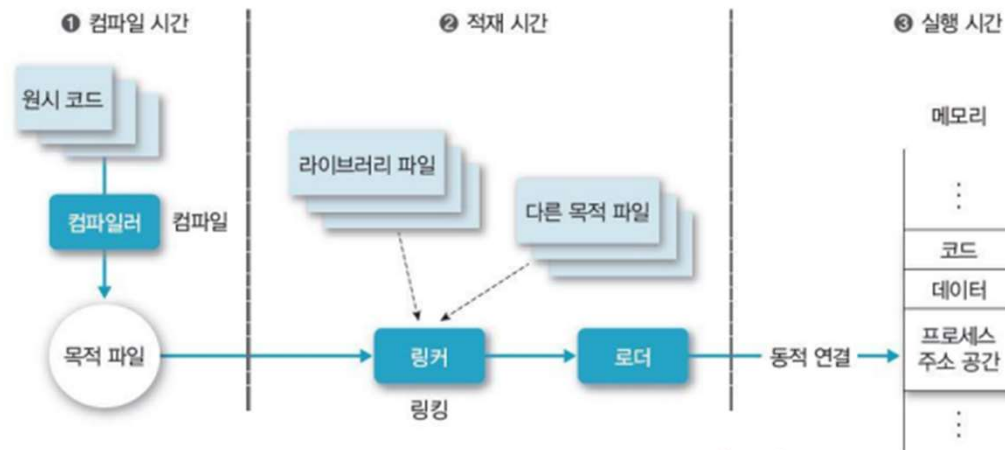
메모리의 장치의 주소 변환



메모리의 구조와 매핑

매핑(Mapping)

- 논리적 주소와 물리적 주소의 연결
- 바인딩: 매핑시켜 주는 작업



메모리 관련 용어

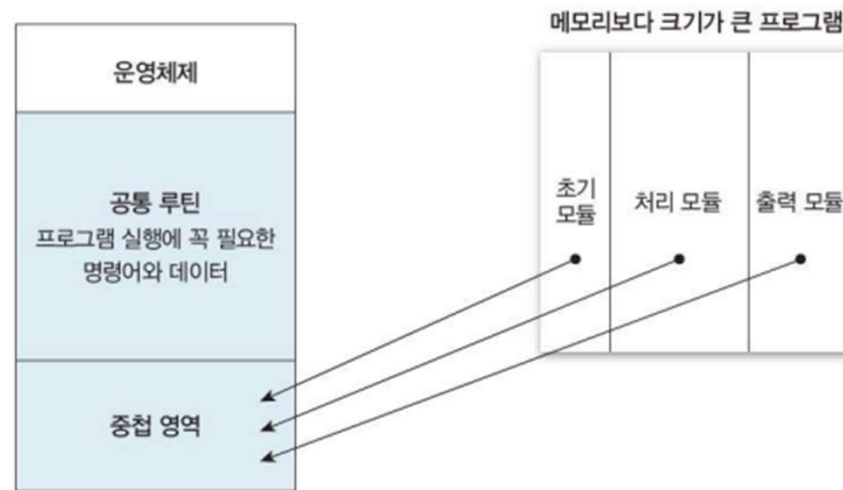
동적 적재(Dynamic loading)

- 바인딩을 최대한 늦춰 실행 직전에 주소를 확정하는 메모리 효율적으로 운영
- 모든 루틴에 메모리 적재 x 교체 가능한 형태로 디스크에 저장.
- 메인 프로그램만 먼저 메모리에 적재
- 메인 프로그램에 다른 루틴이 필요할 때 메모리에 적재되어 있는지 조사, 만약 없다면 주소 테이블 갱신
- 동적 적재는 사용하지 않을 루틴을 메모리에 적재하지 않으므로 메모리 효율적으로 사용

메모리 관련 용어

중첩(Overlay)

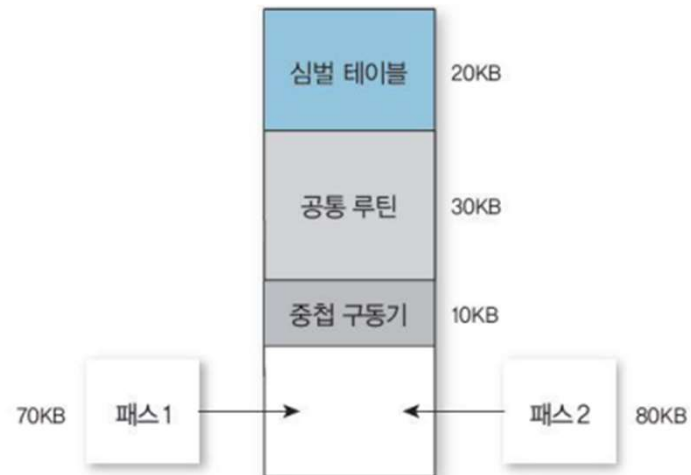
- 실행하려는 프로그램이 메모리보다 클 때는 당장 필요하지 않은 프로그램의 일부는 Overlay영역으로 설정
- 운영체제 영역과 메모리의 일부 영역에는 프로그램 실행에 꼭 필요한 명령어와 데이터만 저장



메모리 관련 용어

중첩(Overlay)

- 패스 1 : 70KB
- 패스 2 : 80KB
- 심벌 테이블 : 20KB
- 공통 루틴 : 30KB
- 중첩 구동기(오버레이 드라이버) : 10KB

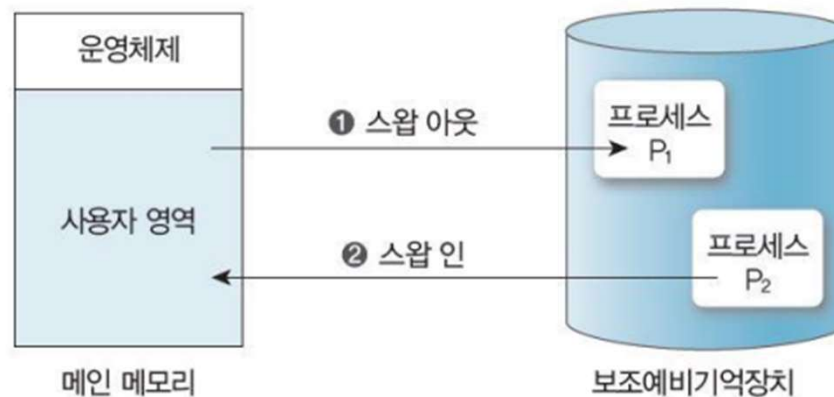


- 중첩 A : 심벌 테이블, 공통 루틴, 패스 1
- 중첩 B : 심벌 테이블, 공통 루틴, 패스 2

메모리 관련 용어

스와핑(Swapping)

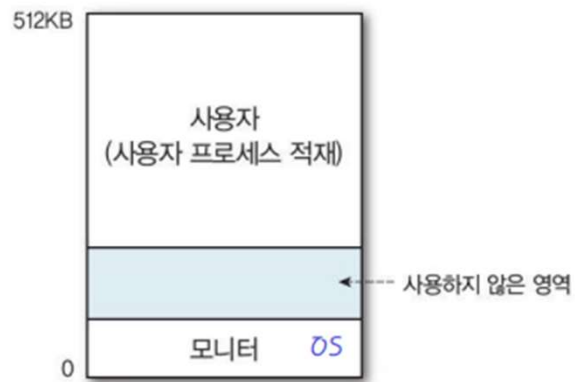
- 프로세서 할당이 끝나고 수행 완료된 프로세스는 보조기억장치로 보내고(스왑아웃), 새롭게 시작하는 프로세스는 메모리에 적재(스왑인). 프로세스는 메모리에 있어야 수행되므로 일시적으로 디스크로 이동했다가 메모리로 되돌아와 다시 수행 가능



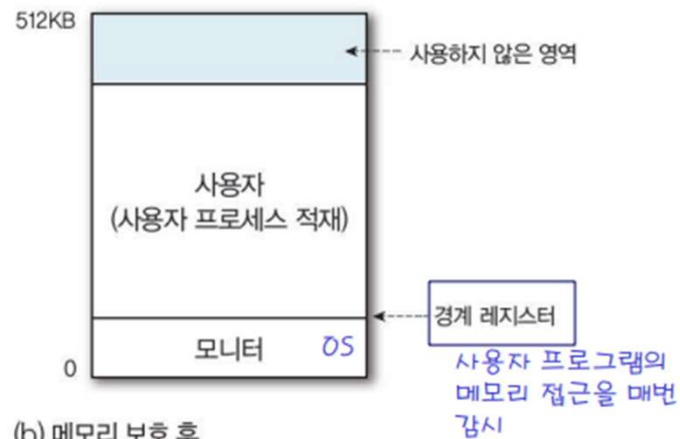
연속 메모리 할당

단일 프로그래밍 환경에서

- 초기 컴퓨터 시스템은 사용자 한 명만 컴퓨터를 사용.
- 프로그램은 메모리보다 클 수 없고 항상 같은 위치에 적재.
- 메모리를 제어하는 권한이 사용자에게 있어서 잘못된 주소 접근은 OS 손상.



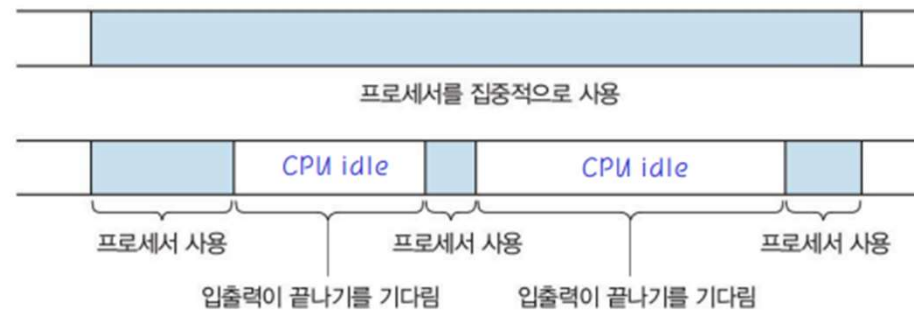
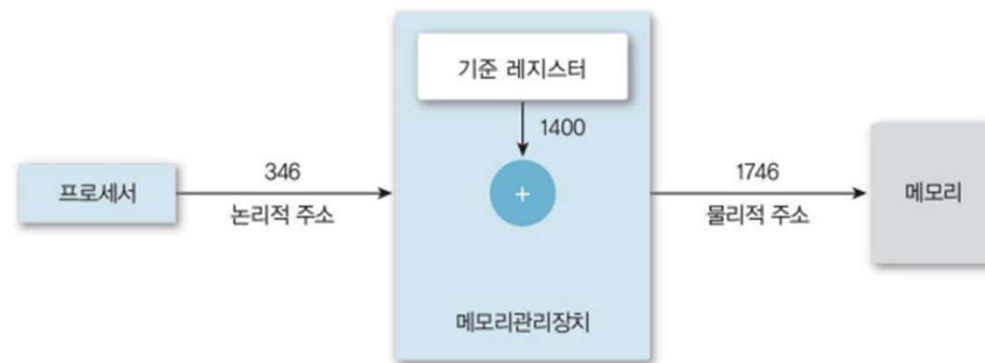
(a) 메모리 보호 전



(b) 메모리 보호 후

연속 메모리 할당

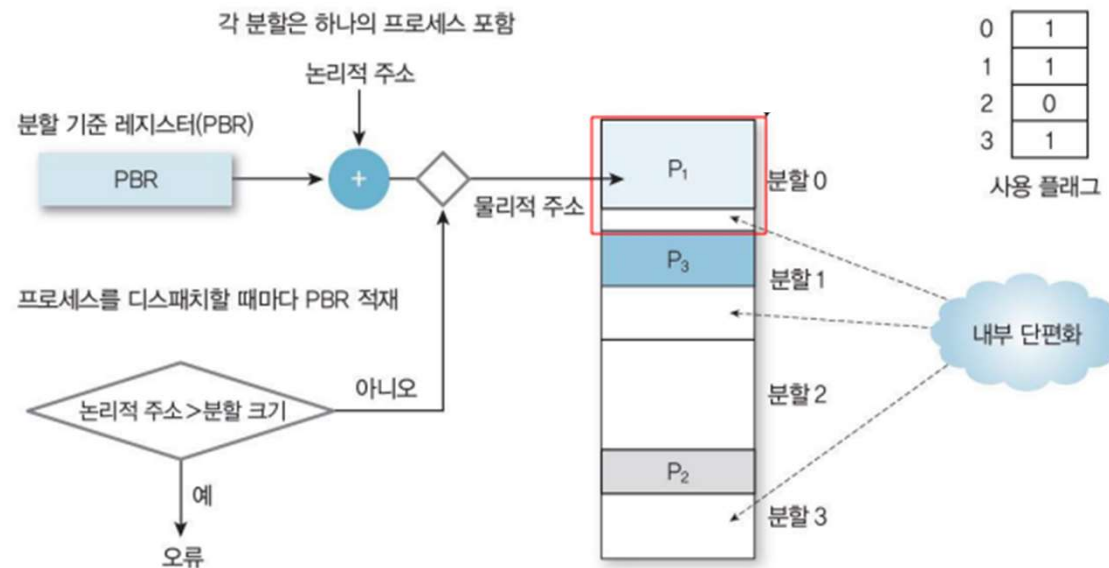
단일 프로그래밍 환경에서



연속메모리 할당

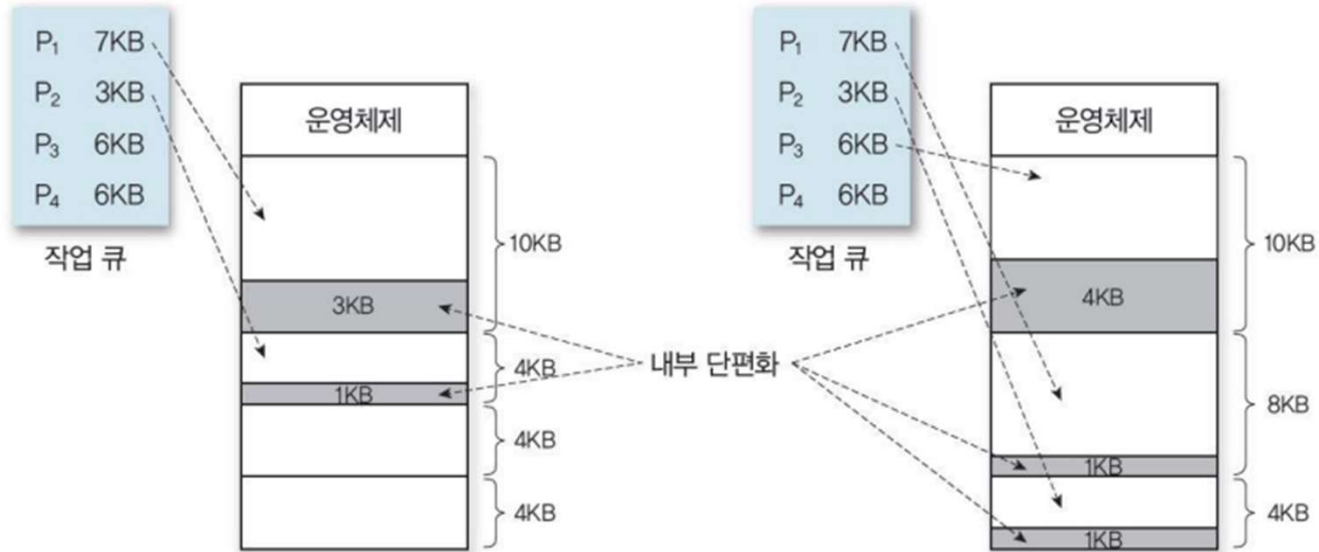
다중 프로그래밍 환경에서 고정 분할 방법

- 연속 메모리 할당에서는 메모리를 여러 개의 고정된 크기로 분할하고 분할된 각 메모리는 프로세스 하나 실행 가능.



연속메모리 할당

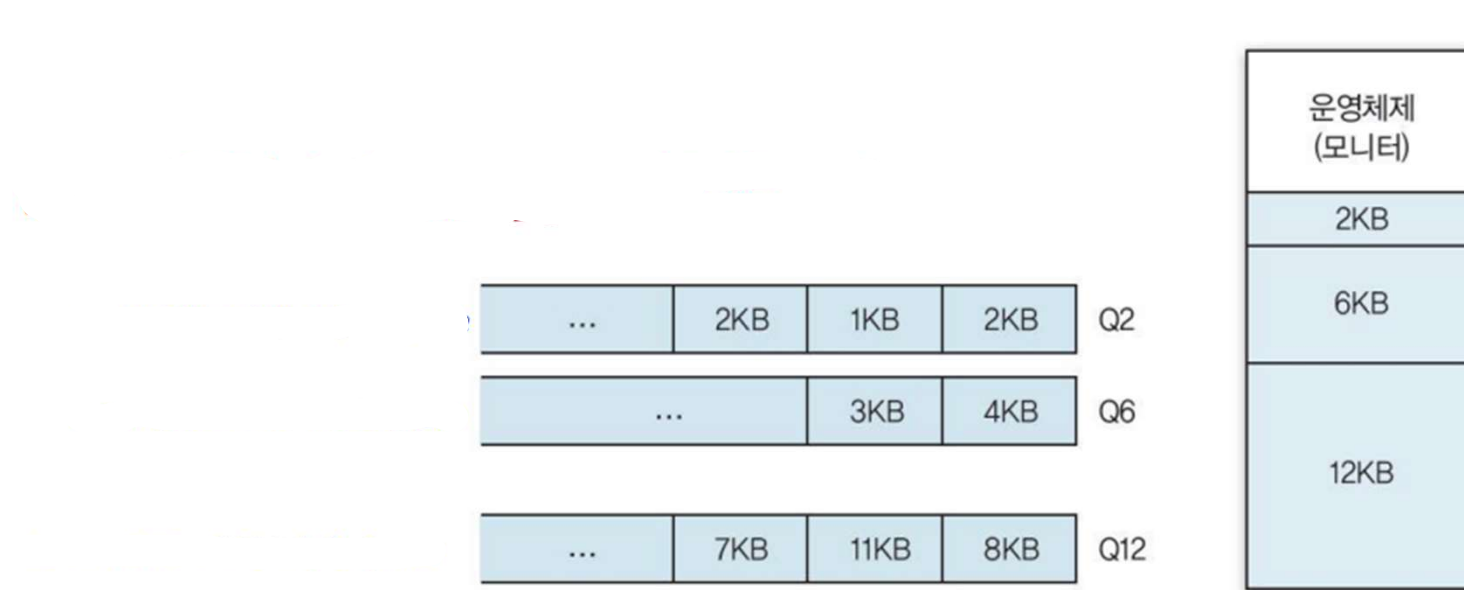
다중 프로그래밍 환경에서 내부 단편화의 변화



연속메모리 할당

다중 프로그래밍 환경에서 변화

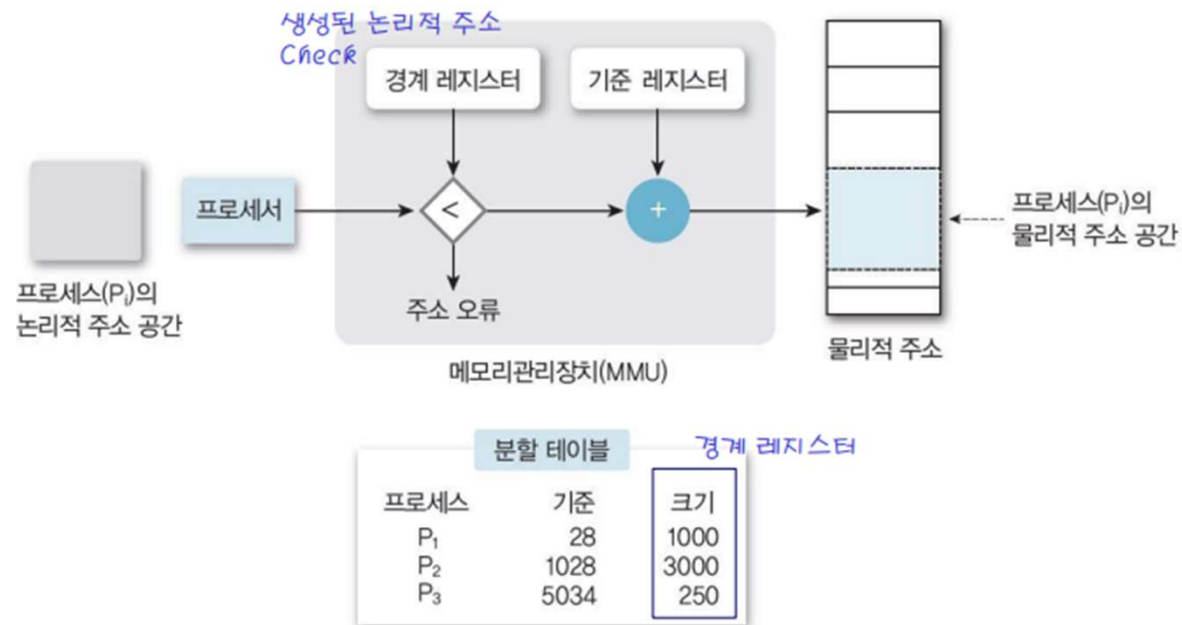
- 각 영역별로 독립된 큐가 있는 고정 분할 시스템



연속 메모리 할당

다중 프로그래밍 환경에서 가변 분할 방법

- 고정된 경계를 없애고 각 프로세스가 필요한 만큼 메모리 할당

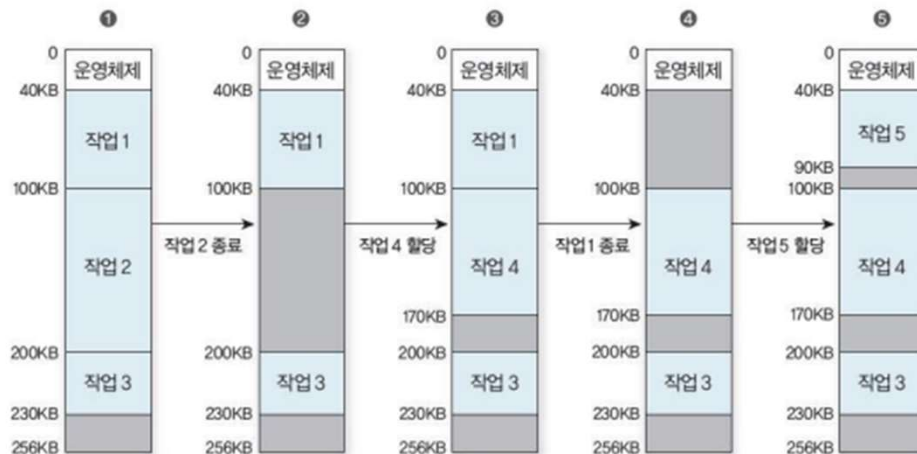


연속 메모리 할당

다중 프로그래밍 환경에서 가변 분할 방법



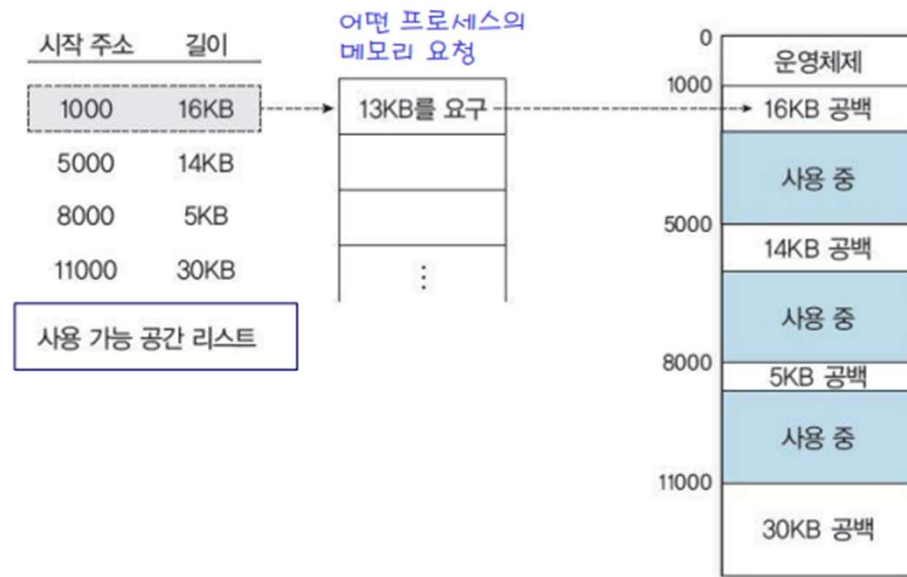
(a) 가변 분할 할당을 설명하는 메모리와 작업 큐 예



연속 메모리 할당

다중 프로그래밍 환경에서 메모리 배치 방법

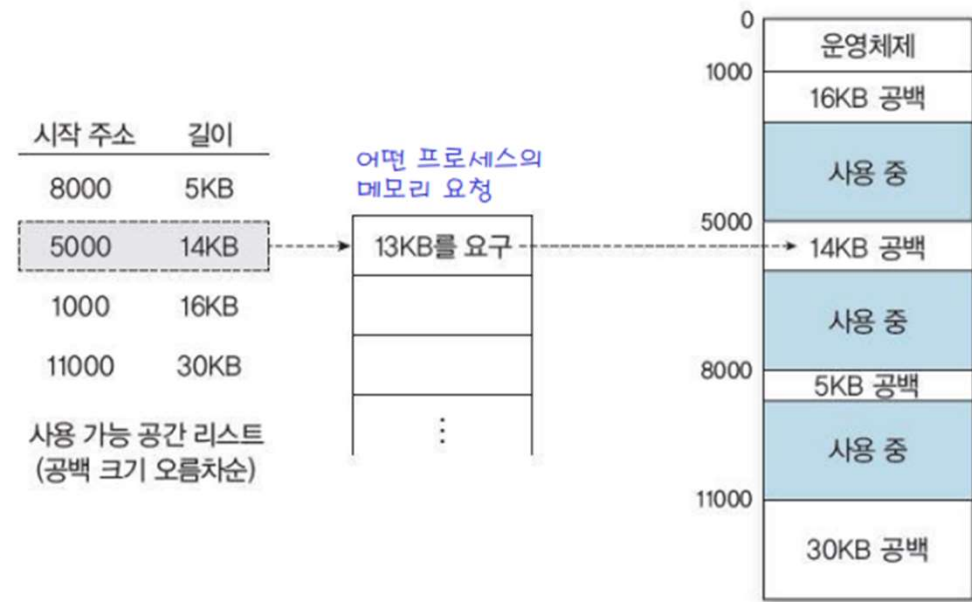
- 최초 적합 방법: 프로세스를 사용 가능 공간 중 충분히 큰 첫 번째 공간에 할당



연속 메모리 할당

다중 프로그래밍 환경에서 메모리 배치 방법

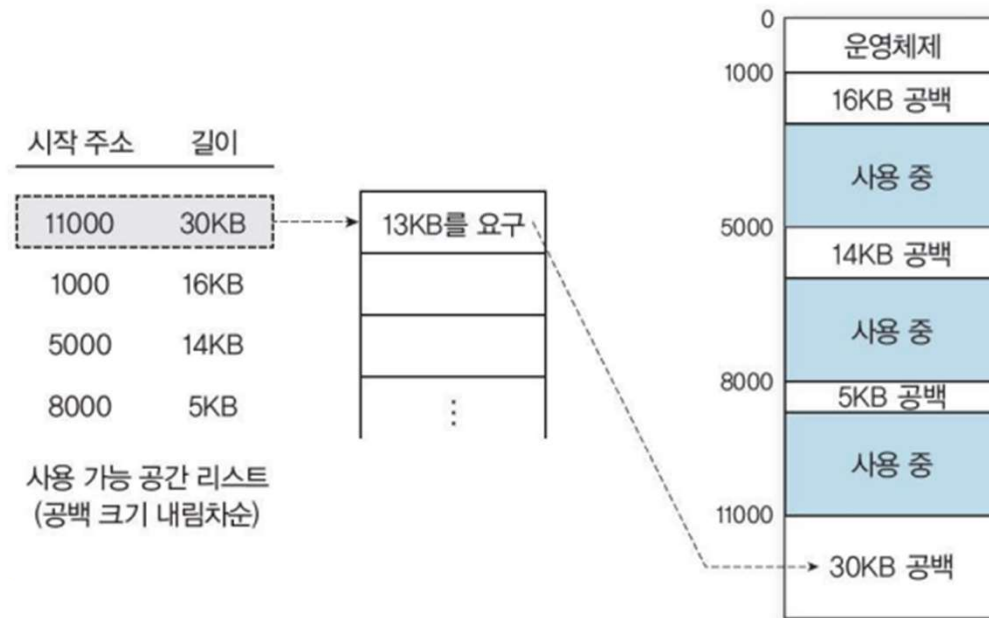
- 최적 적합 방법: 프로세스를 사용 가능 공간 중 들어갈 수 있는 가장 작은 공간에 할당



연속 메모리 할당

다중 프로그래밍 환경에서 메모리 배치 방법

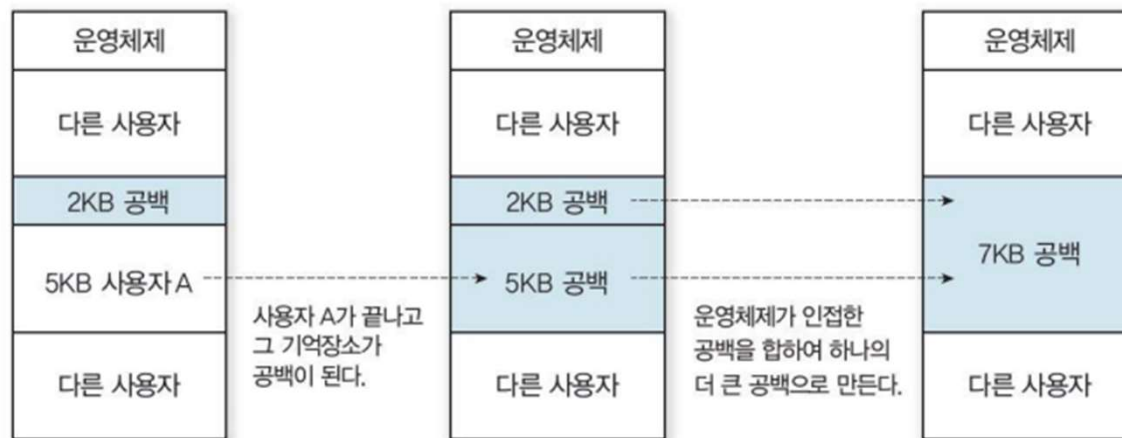
- 최악 적합 방법: 프로세스를 가장 큰 사용 가능 공간에 할당



연속 메모리 할당

다중 프로그래밍 환경에서 메모리 배치 방법

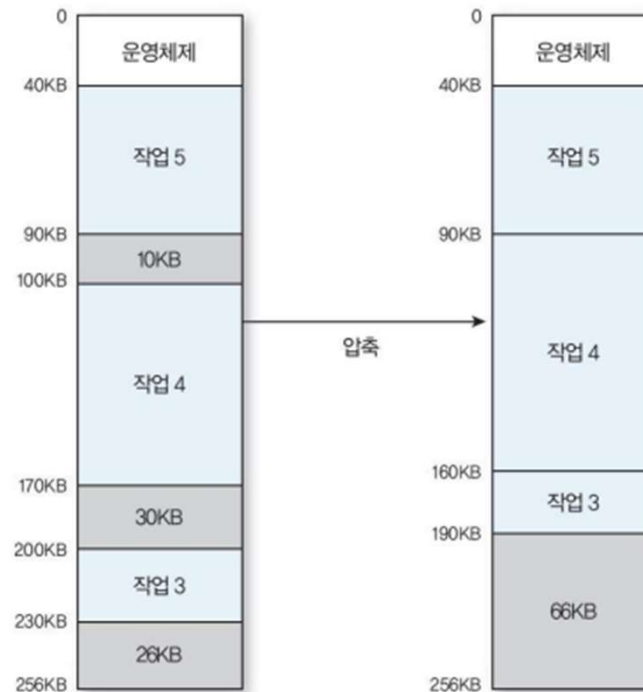
- 메모리 통합 방법: 하나의 작업이 끝났을 때 다른 빈 공간과 인접해 있는지 점검하여 하나로 합침.



연속 메모리 할당

다중 프로그래밍 환경에서 메모리 배치 방법

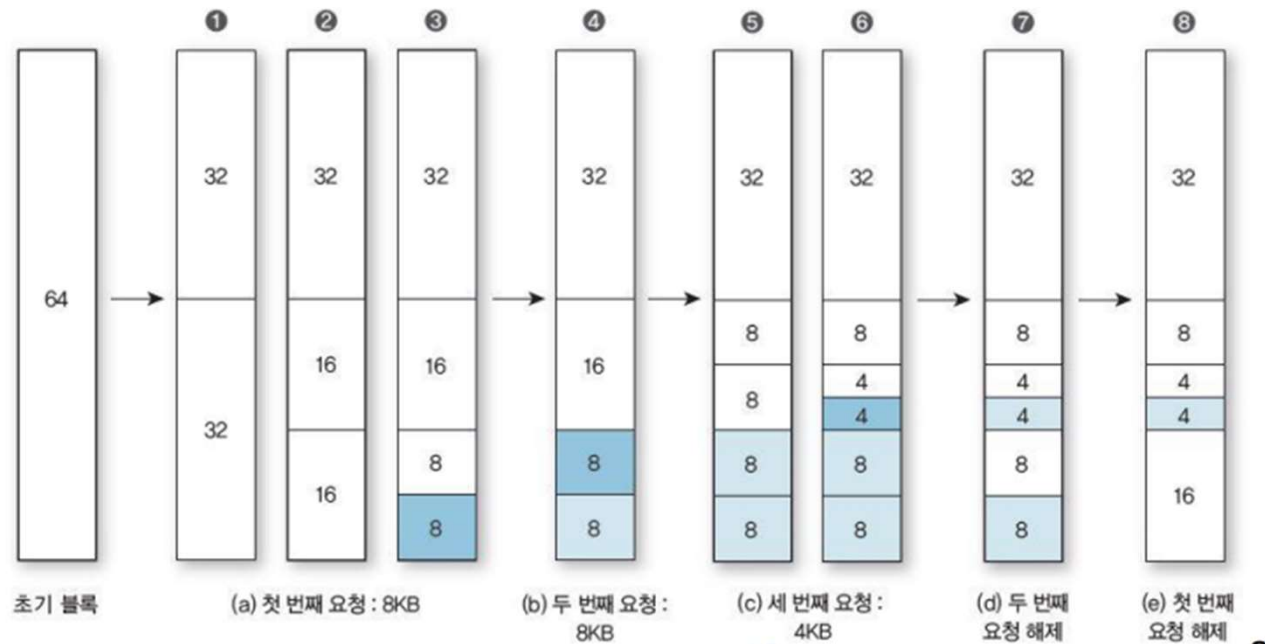
- 메모리 압축 방법: 메모리의 내용을 적절히 움직여 사용 가능 공간을 큰 블록 하나로 만드는 것



연속 메모리 할당

다중 프로그래밍 환경에서 버디 시스템

- 단편화 현상을 해결하는 방법
- 큰 버퍼들을 반복적으로 이등분하여 작은 버퍼들을 만들며, 가능할 때마다 인접한 빈 버퍼들을 합치는 과정.



Q & A