

Memory Management In OS (1)

IT융합공학부 사이버보안트랙 윤세영
유튜브 주소: <https://youtu.be/1WPt0IM1s4A>

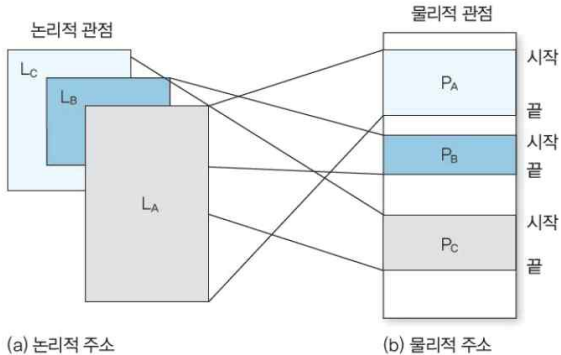
메모리 관리

- 메모리 관리란, 프로세스들을 위해 메모리를 할당하거나 제거하고, 보호하는 것과 같은 활동을 하는 것이다.
- 디스크에 있는 프로그램을 실행하려면, 먼저 메모리에 적재 후 메모리 관리자가 예약된 메모리를 할당해준다.
- 다중 프로그래밍 시스템에서 여러 프로세스가 메모리에 상주할 수 있도록 운영체제가 동적으로 메모리를 세분화 한다.

메모리 관리

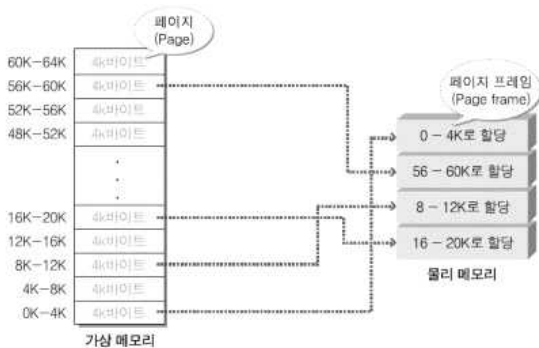
- 반입(Fetch) : 디스크에서 메모리로 프로세스 반입 시기 결정하는 것
 - 요구 반입 : 운영체제나 시스템 프로그램, 사용자 프로그램 등 참조 요청에 따라 다음에 실행 할 프로세스를 메모리에 적재하는 오래된 방법
 - 예상 반입(선반입) : 시스템의 요청을 미리 예측하여 메모리에 적재하는 방법
- 배치(Placement) : 디스크에서 반입한 프로세스를 메모리 어느 위치에 저장할 것인지 결정
- 대치(Replacement) : 메모리가 충분하지 않을 때 현재 메모리에 적재된 프로세스 중 제거할 프로세스를 결정하는 교체 방법

메모리의 구조



매핑이란?

- 논리적 주소(혹은 가상 주소)와 물리적 주소의 연결



메모리 장치의 주소 변환

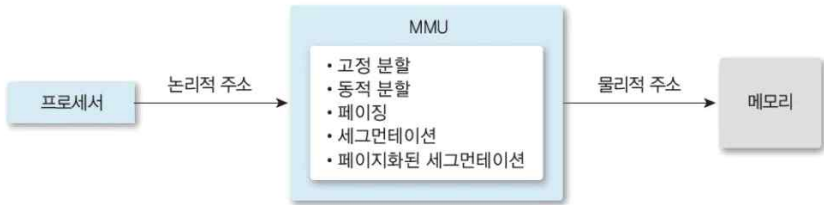
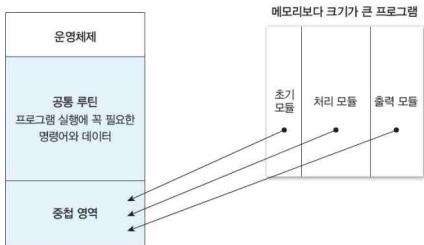


그림 7-2 메모리관리장치의 주소 변환(논리적 주소 → 물리적 주소)

중첩(Overlay)

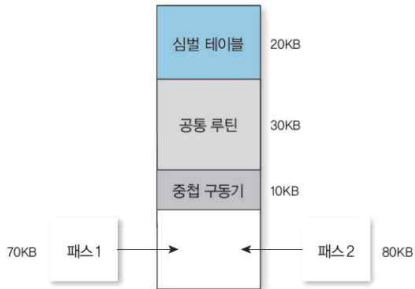
- 실행하려는 프로그램이 메모리보다 클 때는 당장 필요하지 않은 프로그램의 일부는 Overlay 영역으로 설정 가능.
- 운영체제 영역과 메모리의 일부 영역에는 프로그램 실행에 꼭 필요한 명령어와 데이터만 저장, 나머지 중첩 영역에는 필요할 때 호출하는 적재방법



중첩(Overlay) 예

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

(a) 2단계 어셈블러 예

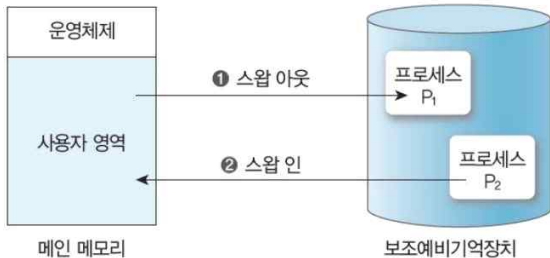


- 중첩 A : 심벌 테이블, 공통 루틴, 패스 1
- 중첩 B : 심벌 테이블, 공통 루틴, 패스 2

(b) 2단계 어셈블러 중첩

Swapping

- 프로세서 할당이 끝나고 수행 완료된 프로세스는 보조기억장치로 보내고(스왑 아웃), 새롭게 시작하는 프로세스는 메모리에 적재(스왑 인). 프로세스는 메모리에 있어야 수행되므로 일시적으로 디스크로 이동했다가 메모리로 돌아와 다시 수행 가능



연속 메모리 할당 - 다중 프로그래밍 환경에서 할당

• 고정 분할 방법

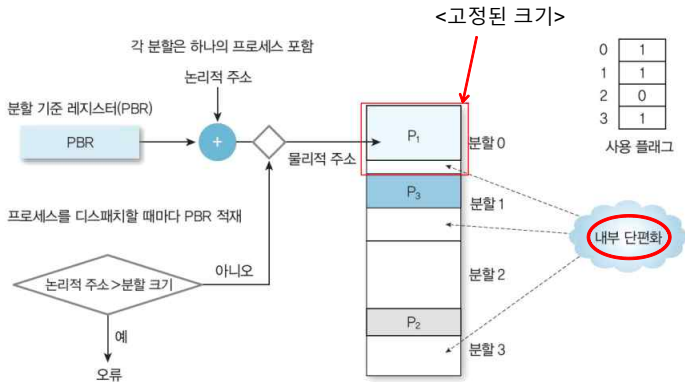


그림 7-12 다중 프로그래밍 환경에서 고정 분할 방법 예(IBM OS/360)

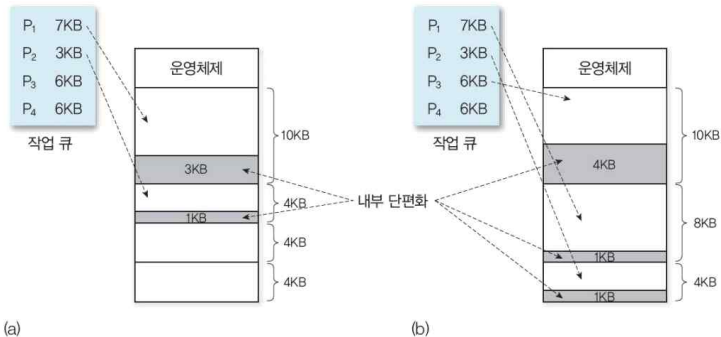
연속 메모리 할당 - 다중 프로그래밍 환경에서 할당

- 고정 분할 방법
- 내부 단편화란?



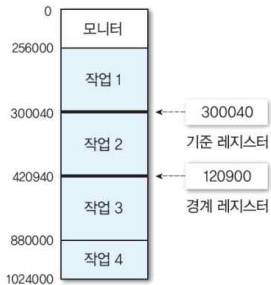
연속 메모리 할당 - 다중 프로그래밍 환경에서 할당

- 고정 분할 방법에서 분할 크기에 따른 내부 단편화의 변화

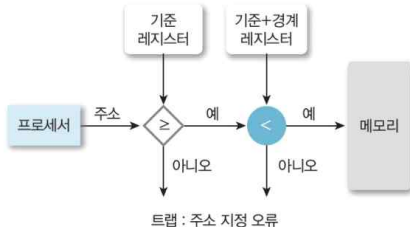


연속 메모리 할당 - 다중 프로그래밍 환경에서 할당

• 고정 분할 방법에서 메모리 보호 예



(a) 메모리를 보호하는 기준 레지스터와
경계 레지스터

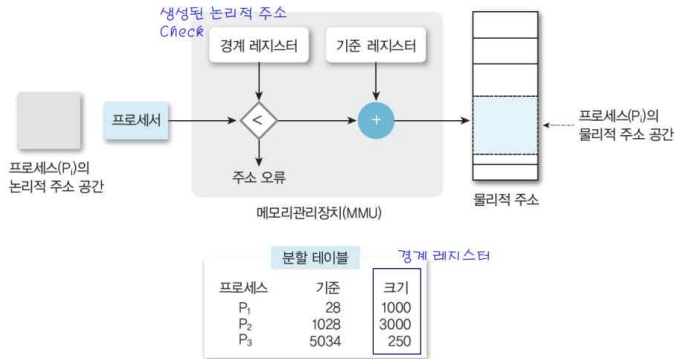


(b) 메모리 보호 과정

그림 7-15 고정 분할 방법에서 메모리 보호 예

연속 메모리 할당 - 다중 프로그래밍 환경에서 할당

- 가변 분할 방법
- 고정된 경계를 없애고 각 프로세스가 필요한 만큼 메모리 할당

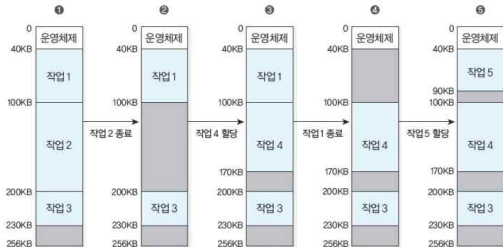


연속 메모리 할당 - 다중 프로그래밍 환경에서 할당

• 가변 분할 방법의 예



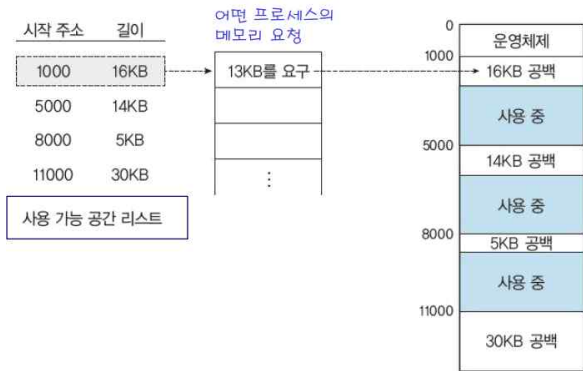
(a) 가변 분할 할당을 설명하는 메모리와 작업 큐 예



(b) 메모리 할당과 스케줄링 과정

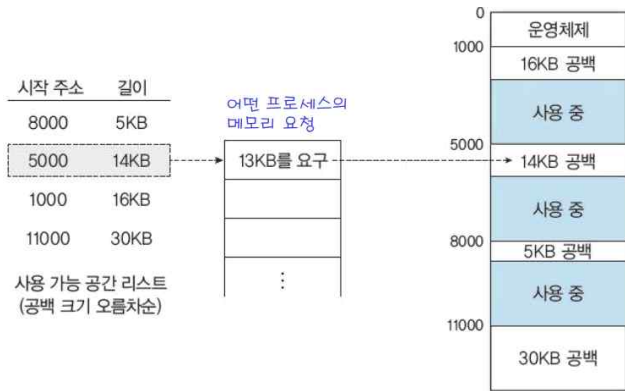
연속 메모리 할당 - 다중 프로그래밍 환경에서 할당

- 가변 분할 방법
- 최초 적합 방법



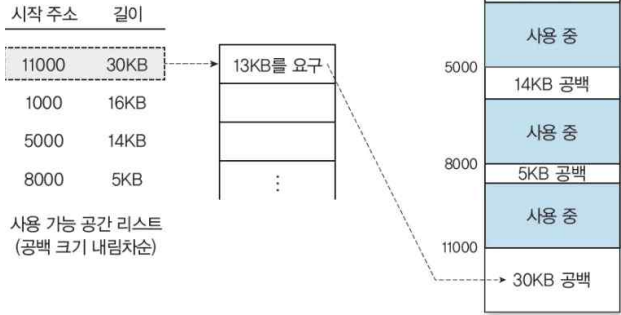
연속 메모리 할당 - 다중 프로그래밍 환경에서 할당

- 가변 분할 방법
- 최적 적합 방법



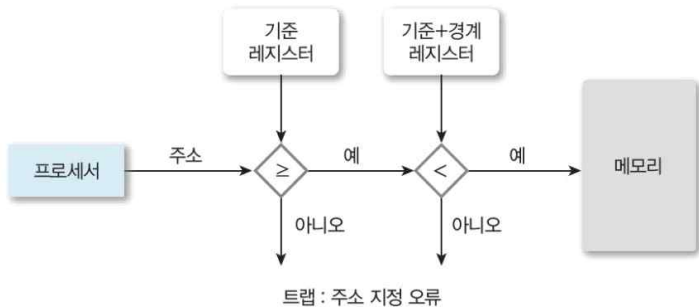
연속 메모리 할당 - 다중 프로그래밍 환경에서 할당

- 가변 분할 방법
- 최악 적합 방법



연속 메모리 할당 - 다중 프로그래밍 환경에서 할당

- 가변 분할 방법에서 메모리를 보호하는 과정



Q & A