운 영 체 제

3장 기억장치 관리(2)

3장 기억장치 관리

기억장치 관리의 발전

기억장치 관리의 발전

연속 적재 기법 (Contiguous Loading)		분산 적재 기법 (Scatter Loading)						
실기억 공간 (Real Memory)							가상 기억 공간 (Virtual Memory)	
단일 사용자 (Single User)		다중 프로그래밍 (Multi-Programming)						
오버레이 Overlay	교체 Swapping	고정 분할 Fixed Partition	동적 분할 Dynamic Partition	페이징 Paging	세그먼테이션 Segmentation	페이지/세그먼트의 혼합 형태	요구 페이징	요구 세그먼테이션

오버레이 (overlay)기법
=> 프로그램 크기가 클때
스와핑(Swapping)기법
⇒시스템 성능 향상
⇒다중 프로그래밍 지원
⇒페이징 기법으로 발전

기억장치 할당 기법

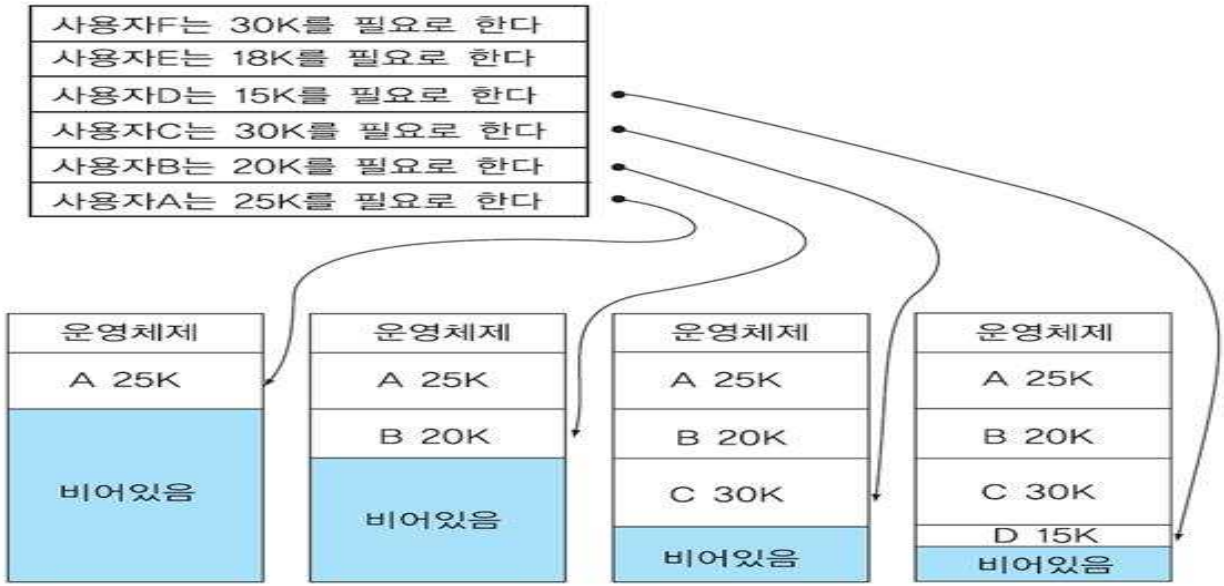
- 주기억장치 할당 기법
  - 단일 분할 할당 기법
    - 항상 시스템 내에 하나의 프로세스만 존재
    - 문제점 해결
    - 프로그램의 크기가 클 대: 오버레이 **(overlay)**기법
    - 프로그램 교체 :스와핑**(Swapping)**기법
    - 시스템 보호: 경계 레지스터 사용
    - 시스템 자원의 낭비, 시스템 성능 저하 : 다중프로그래밍 기법 사용
  - 고정 분할 할당 기법
    - 기억장치를 여러 개의 고정된 크기로 분할하는 기법
    - 정적 분할 다중 프로그래밍
    - 절대 로더와 재배치 로더에서 사용
    - 문제점
      - 시스템 보호 : 여러 개의 경계 레지스터를 사용하여 해결
      - 단편화 발생 : 내부 단편화, 외부 단편화 => 해결 방안 ?

가변 분할 할당 기법(다중프로그래밍)

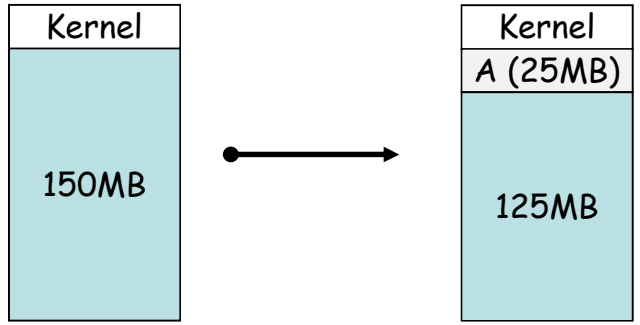
- 가변분할할당(**Multiple configuous Variable parTition allocation, MVT**)기법 = 동적할당 **(Dynamic allocation)**기법
  - 고정분할 기법의 단편화를 줄이기 위한 것으로 미리 주기억 장치를 분할해 놓는 것이 아니라 프로그램을 주기억장치에 적재하면서 필요한 만큼의 크기로 영역을 분할하는 기법이다.
  - 프로세스들의 활동에 따라 분할 형태를 동적으로 변화 시킴
  - 주기억장치를 효율적으로 사용할 수 있으며, 다중 프로그래밍의 정도를 높일 수 있다.
  - 고정분할할당 기법에 비해 실행될 프로세스 크기에 대한 제약이 적다.
  - 단편화를 상당 부분 해결할 수 있으나 영역과 영역 사이에 단편화가 발생될 수 있다. 즉, 분할 영역 내의 내부 단편화 현상 발생 않지만 외부 단편화는 발생
  - 관리 오버헤드 증가

가변 분할 할당 기법(다중프로그래밍)

- 작업들이 필요로 하는 만큼의 공간을 동적으로 할당
- 기억 공간의 효율화



가변 분할 다중 프로그래밍에서의 초기의 분할 할당

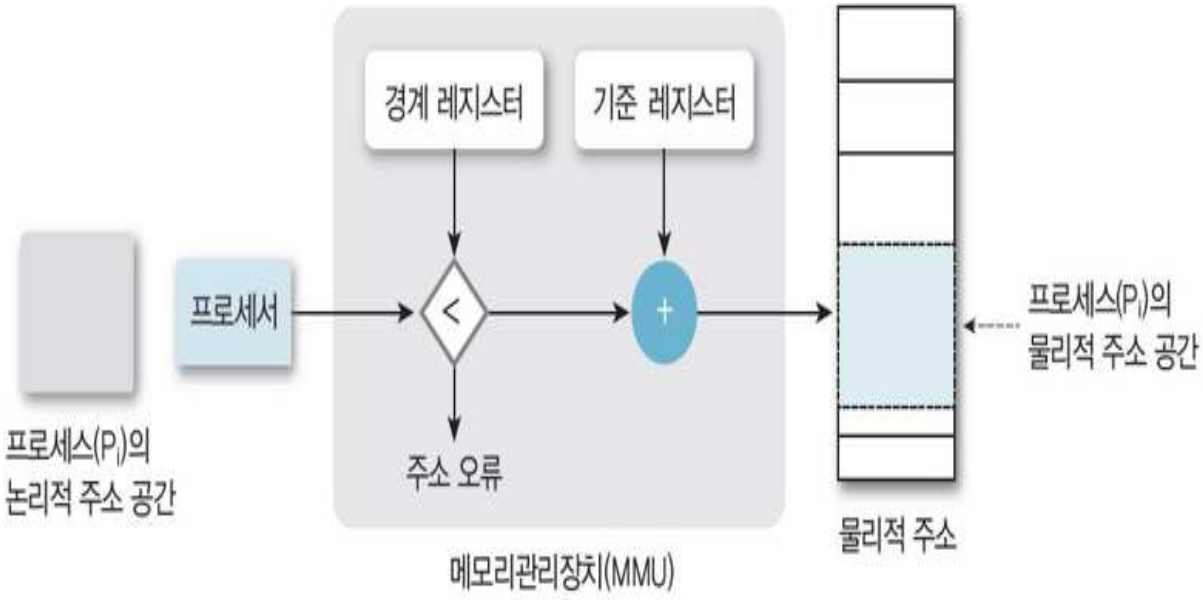


partition	start address	size	current process ID	other field
1	u	25	A	· · ·
2	u+25	20	B	· · ·
3	u+45	30	C	· · ·
4	u+75	15	D	· · ·
5	u+90	60	none	· · ·

기억장소 상태 테이블

가변 분할 할당 기법(다중프로그래밍)

- 고정된 경계를 없애고 각 프로세스가 필요한 만큼 메모리 할당
- 기억 공간의 효율화



다중 프로그래밍에서의 가변 분할 할당 기법

partition	start address	size	current process ID	other field
1	u	25	A	· · ·
2	u+25	20	B	· · ·
3	u+45	30	C	· · ·
4	u+75	15	D	· · ·
5	u+90	60	none	· · ·

기억장소 상태 테이블

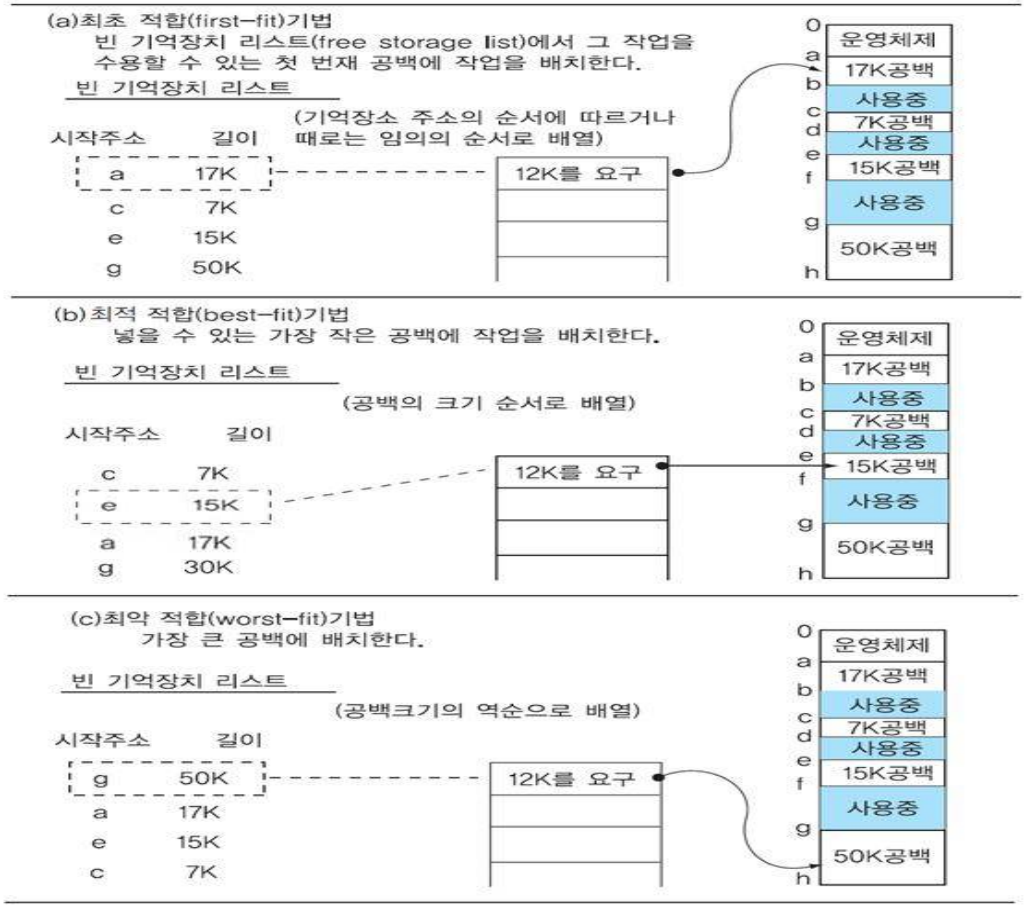
가변 분할 할당 기법(다중프로그래밍)

- 단편화를 상당 부분 해결할 수 있으나 영역과 영역 사이에 단편화가 발생될 수 있다.
- 내부 단편화 현상은 해결되어 발생하지 않는다.
- 외부 단편화는 발생한다.

가변 분할 다중 프로그래밍에서의 외부 단편화 현상

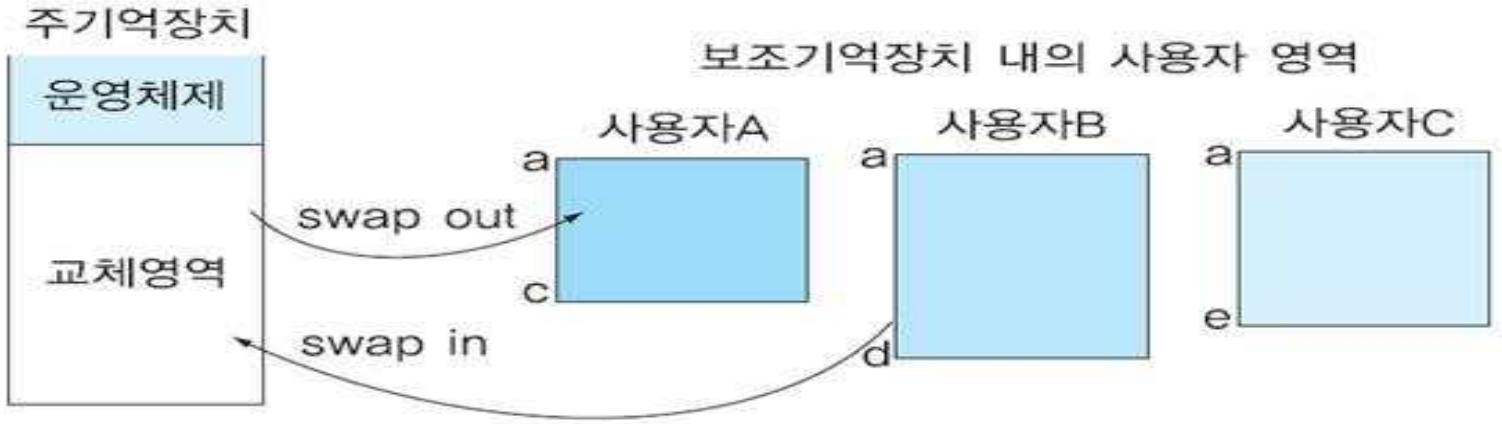
가변 분할 할당 기법(다중프로그래밍)

- 기억장치 배치 전략(memory placement strategy)
  - 최초 적합 기법(first-fit strategy)
    - 주기억장치의 첫 번째 유용한 공백을 우선적으로 선택
    - 널리 사용 됨
  - 최적 적합 기법(best-fit strategy)
    - 가장 적합한 공간을 선택
    - 기억장치의 단편화를 최소화 하는 방법
  - 최악 적합 기법(worst-fit strategy)
    - 가장 큰 공백에 배치



가변 분할 할당 기법(다중프로그래밍)

- 기억장치 교체(swapping)
  - 하나의 작업이 전체 기억장치를 사용한 후, 필요에 따라 그 작업은 제거(swap out)되고 다시 다음 작업이 적재(swap in)
  - 오늘날 일반적으로 사용되는 페이징 시스템(paging system)의 기초



다중 프로그래밍 시스템에서의 기억장치 교체

주기억장치 관리 기법의 문제점과 해결 방법

단편화(Fragmentation)

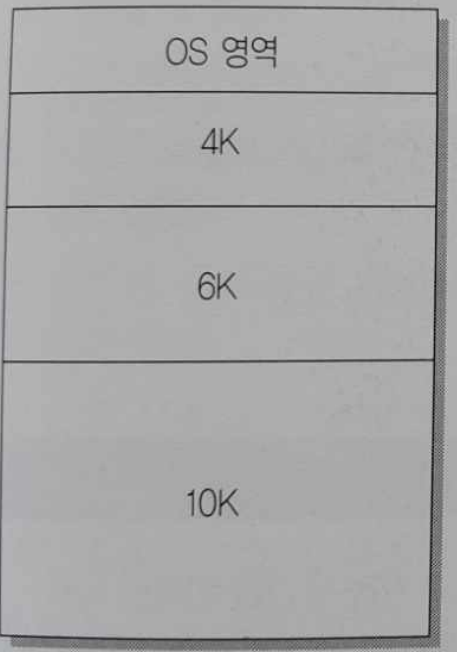
- 단편화는 분할된 주기억장치에 프로그램을 할당하고 반납하는 과정을 반복하면서 사용되지 않고 남는 기억장치의 빈 공간 조각을 의미하며, 내부 단편화와 외부 단편화가 있다.

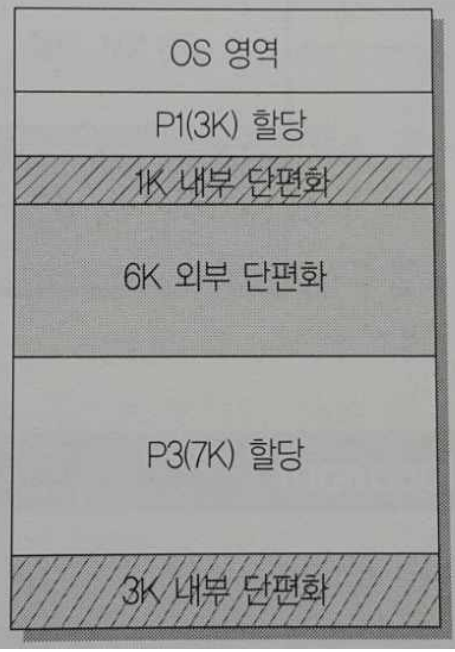
- 내부 단편화(Internal Fragmentation)
  - 분할된 영역이 할당될 프로그램의 크기보다 크기 때문에 프로그램이 할당된 후 사용되지 않고 남아있는 빈 공간

- 외부 단편화(External Fragmentation)
  - 분할된 영역이 할당될 프로그램의 크기보다 작기 때문에 프로그램이 할당될 수 없어 사용되지 않고 빈 공간으로 남아있는 분할된 전체 영역

단편화(Fragmentation)

예제 4K, 6K, 10K 크기의 분할된 영역으로 구성된 주기억장치에 P1(3K), P2(12K), P3(7K) 크기의 프로그램을 차례로 할당할 때 생기는 내부 단편화와 외부 단편화의 크기는?





- 내부 단편화 = 1K + 3K = 4K
- 외부 단편화 = 6K

단편화(Fragmentation) 해결 방법

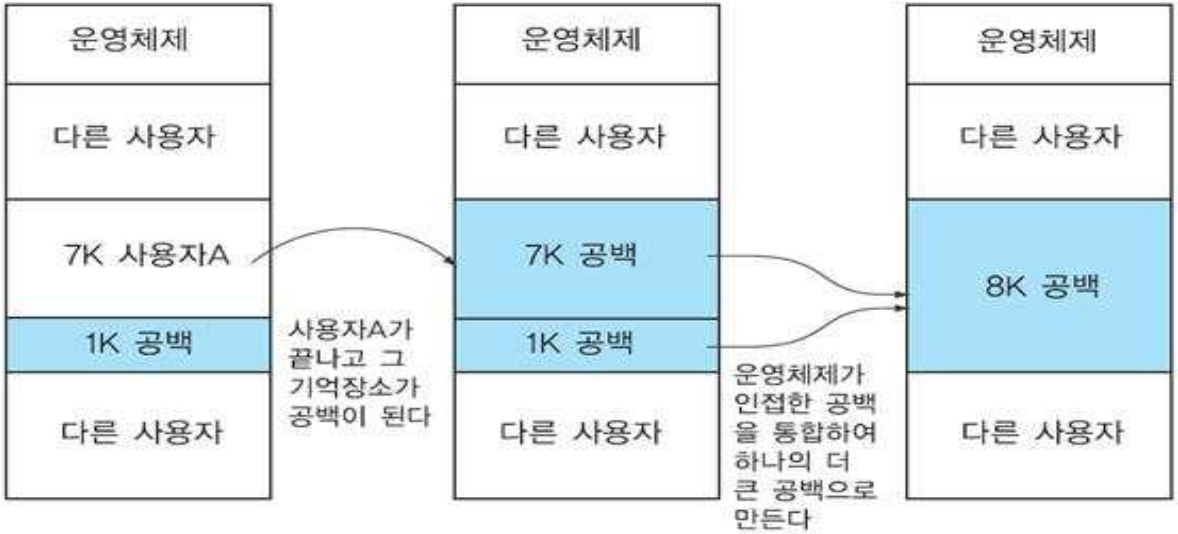
- 주기억장치를 재사용할 수 있도록 단편화된 작은 공간을 모아서 하나의 큰 공간으로 만드는 것으로 자원과 프로그램을 효율적으로 사용할 수 있다.
- 통합기법과 압축기법이 있다.

- 통합(Coalescing)기법
  - 주기억장치 내에 인접해 있는 단편화된 공간을 하나의 공간으로 통합하는 작업을 의미한다.

- 압축(Compaction)기법
  - 주기억장치 내에 분산되어 있는 단편화된 빈 공간을 결합하여 하나의 큰 가용공간을 만드는 작업을 의미한다.
  - 쓰레기 수집(garbage collection)이라고도 한다.

단편화(Fragmentation) 해결 방법

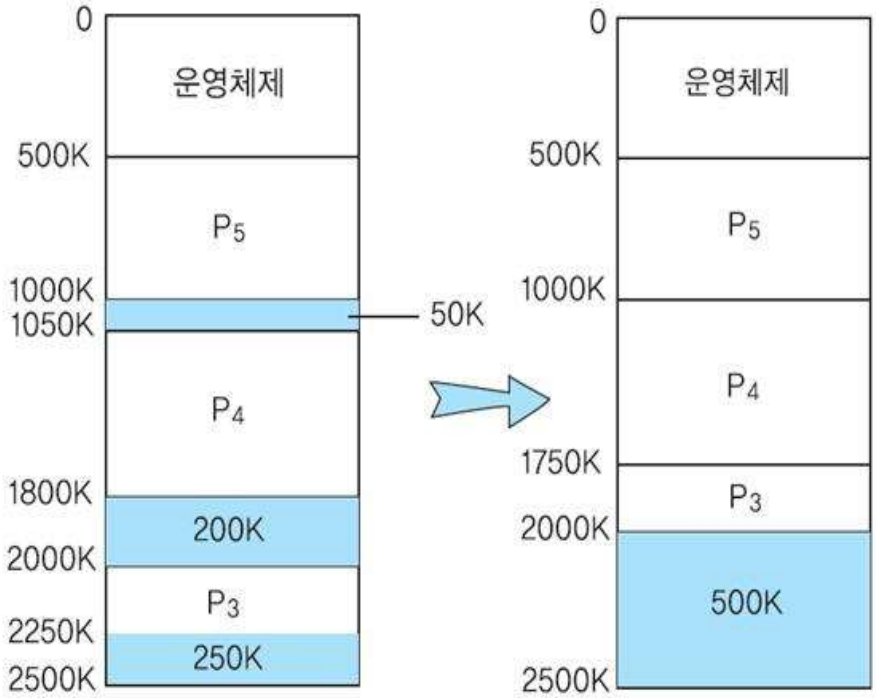
- 통합(Coalescing)기법 = 합병 기법=인접 공간 통합
  - 주기억장치 내에 인접해 있는 단편화된 공간을 하나의 공간으로 통합하는 작업을 의미한다.
  - 주기억장치에 빈 공간이 발생할 경우 이 빈 공간이 다른 빈 공간과 인접되어 있는지 점검한 후 결합하여 사용한다.



가변 분할 다중 프로그래밍에서의 공백의 합병

단편화(Fragmentation) 해결 방법

- 압축(Compaction)기법 =집약 기법
  =기억장소의 집약
  - 주기억장치 내에 분산되어 있는 단편화된 빈 공간을 결합하여 하나의 큰 가용공간을 만드는 작업을 의미한다.
  - 쓰레기 수집(garbage collection)이라고도 한다.
  - 여러 위치에 분산된 단편화된 공간을 주기억 장치의 한쪽 끝으로 옮겨서 큰 가용 공간을 만든다.
  - 압축이 실행되는 동안 시스템은 모든 일을 일시 중단한다.



기억장소의 집약

단편화 문제

[문제]기억장치 관리에서 60k의 사용자 공간이 아래와 같이 분할 되어 있다고 가정할 때 24K, 14K, 12K, 6K의 작업을 최적적합(best fit)전략으로 각각 기억공간에 들어온 순서대로 할당할 경우 생기는 총 내부 단편화의 크기와 외부 단편화의 크기는 얼마인가?

가. 내부단편화 4K ,외부단편화 6K

나. 내부단편화 6K ,외부단편화 8K

다. 내부단편화 6K ,외부단편화 10K

라. 내부단편화 4K ,외부단편화 12K

운영체제
25K
15K
10K
10K

학습 내용 정리

- 주기억장치 할당 기법
  - 단일 분할 할당 기법
    - 항상 시스템 내에 하나의 프로세스만 존재
    - 문제점 해결 기법
      - 오버레이 (overlay)기법,스와핑(Swapping)기법, 시스템 보호로 경계 레지스터 사용
      - 시스템 자원의 낭비, 시스템 성능 저하 : 다중프로그래밍 기법 도입사용
  - 고정 분할 할당 기법
    - 기억장치를 여러 개의 고정된 크키로 분할하는 기법(정적 분할 다중 프로그래밍)
    - 절대 로더와 재배치 로더에서 사용
    - 문제점 => 단편화 발생 : 내부 단편화, 외부 단편화 => 해결 방안 (통합 기법,압축 기법)
  - 가변 분할 할당 기법
    - 작업들이 필요로 하는 만큼의 공간을 동적으로 할당하는 기법으로 기억 공간의 효율화 제공
    - 기억장소의 배치 전략 => 최초 적합 기법, 최적 적합 기법, 최악 적합 기법
    - 내부 단편화 현상은 해결 되지만 외부 단편화는 발생한다.
- 주기억장치 관리 기법의 문제점과 해결 방법
  - 단편화 해결 => 인접 공간 통합 기법, 기억 장소 압축 기법