

# 1

## 기억장치 계층 구조와 주기억장치 구성 정책

# 기억장치 계층 구조와 주기억장치 구성 정책

## 1 기억장치 계층 구조

- ◆ 기억장치 계층 구조 상에서 프로세서 쪽에 가까운 기억장치에 대한 접근 시간은 프로세서 쪽에서 먼 기억장치보다 빠르며, 프로세서로부터 먼 기억장치 일수록 기억 용량이 크다는 특징을 갖음
- ◆ 또한 기억 단위당 소요 비용은 프로세서로부터 먼 기억장치일수록 적게 드는 경향이 있음

# 기억장치 계층 구조와 주기억장치 구성 정책

## 2 주기억장치 구성 정책

- ◆ 주기억장치를 효과적으로 운영하기 위해서는 세부적인 관리 기법들을 설계하기 전에 우선 이에 대한 구성을 어떤 방법으로 할 것인가에 대해 결정해야 함
- ◆ 예를 들어 주기억장치에 동시에 여러 사용자 프로그램이 적재되도록 할 거인가 아니면 한 번에 한 사용자 프로그램만이 적재되도록 할 것인가를 결정해야 함

# 기억장치 계층 구조와 주기억장치 구성 정책

## 2 주기억장치 구성 정책

- ◆ 또한 동시에 여러 사용자 프로그램이 적재되도록 할 경우에는 각 사용자 프로그램들에게 주기억장치를 어떻게 분할하여 할당할 것인가 등에 대한 기본적인 정책이 세워져 있어야 함

# 기억장치 계층 구조와 주기억장치 구성 정책

## 2 주기억장치 구성 정책

- ◆ 세부적으로 결정되어야 할 정책들
  - 주기억장을 동시에 할당 받을 수 있는 프로세스 수
  - 각 프로세스에게 할당되는 주기억장치의 양
  - 주기억장치 분할 방법
  - 각 프로세스에게 할당된 분할영역의 교체 가능성
  - 프로세스에게 할당되는 주기억장치 영역의 연속성

# 2

## 주기억장치 관리 기법

## 1 반입정책

- ◆ 반입정책은 CPU에 의해 실행되거나 참조되기 위해서 주기억장치로 적재할 다음 프로그램이나 자료를 언제 가져올 것인가를 결정하는 문제
  - 요구반입정책 : 실행중인 프로그램에 의해 어떤 프로그램이나 자료가 참조될 때 그것을 주기억장치로 옮기는 요구 반입 기법
  - 예상반입정책 : 현 프로그램 수행 중에 앞으로 요구될 가능성이 큰 자료 또는 프로그램을 예상하여 주기억장치로 미리 옮기는 방법

# 주기억장치 관리 기법

## 2 배치정책

- ▶ 새로 반입된 자료나 프로그램을 주기억장치의 어디에 위치시킬 것인가를 결정하는 정책
- ▶ 최초 적합, 최적 적합, 최악 적합 등이 있음

# 주기억장치 관리 기법

## 3 교체정책

- ◆ 새로 들어온 프로그램이 들어갈 장소를 마련하기 위해서 어떤 프로그램이나 자료를 주기억장치로부터 제거할 것인가를 결정하는 정책
  - 최적화 원칙
  - 무작위 페이지 교체
  - FIFO, SCR, LRU, NUR, LFU

# 3

## 고정 분할 기법과 가변 분할 기법

# 고정 분할 기법과 가변 분할 기법

## 1 메모리 적재 방법

- ▶ 연속 메모리 적재 방법
- ▶ 비연속(분산) 메모리 적재 방법

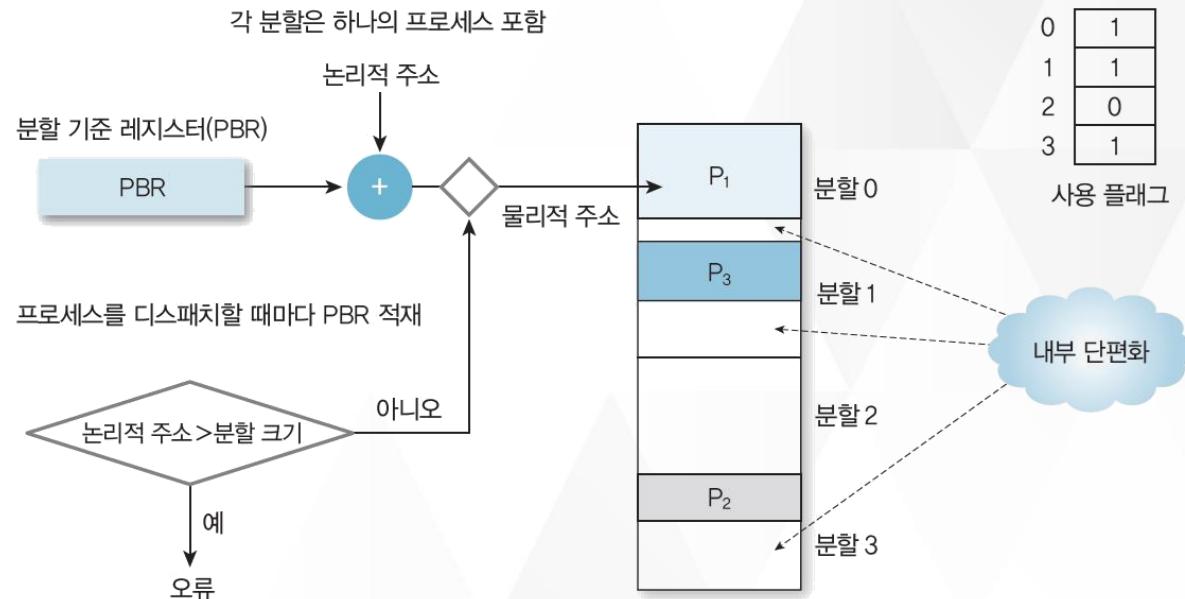


※출처: 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미, 2016

# 고정 분할 기법과 가변 분할 기법

## 2 고정 분할 기법

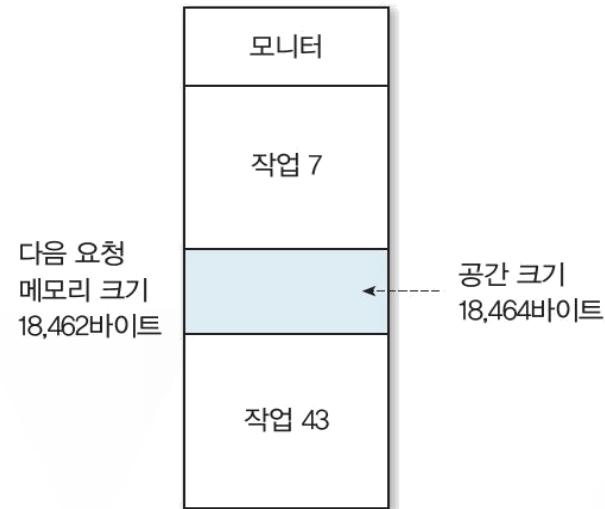
- 연속 메모리 할당에서는 메모리를 여러 개의 고정된 크기로 분할하고 분할된 각 메모리는 프로세스 하나 실행 가능



※출처: 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미, 2016

## 2 고정 분할 기법

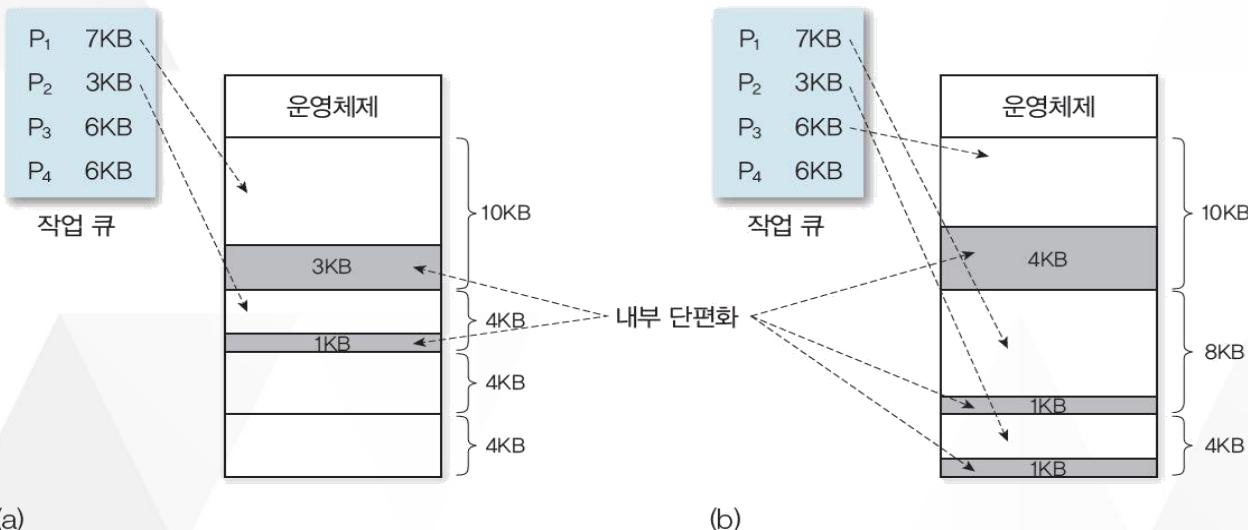
## ▶ 내부 단편화의 개념



※출처: 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미, 2016

### 3 다중 프로그래밍 환경에서 연속 메모리 할당

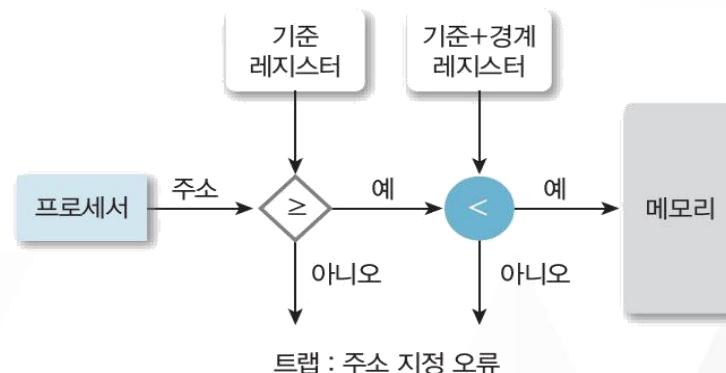
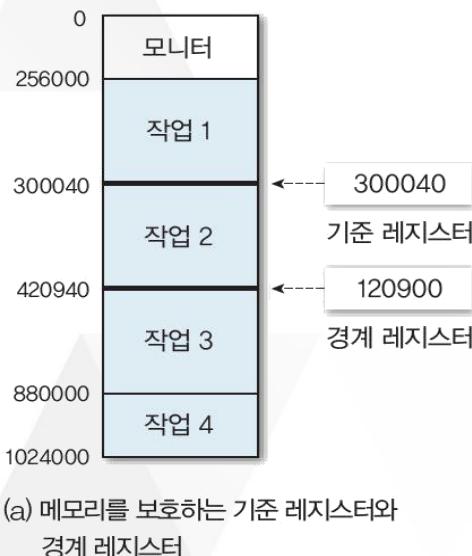
◆ 스케줄링과 분할 크기에 따른 내부 단편화의 변화



※출처: 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미, 2016

### 3 다중 프로그래밍 환경에서 연속 메모리 할당

#### ◆ 고정 분할 기법에서 메모리 보호 예시

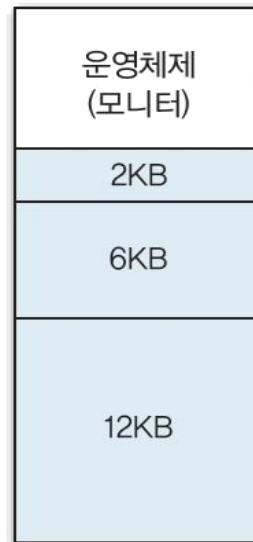
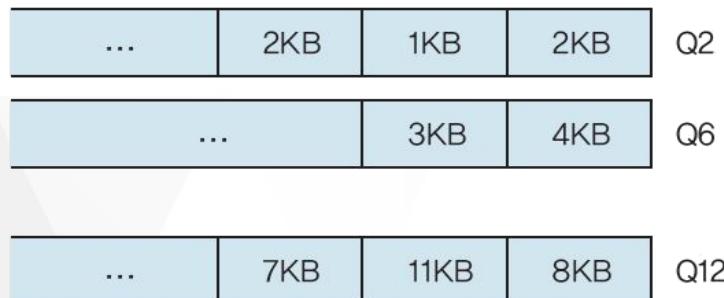


※출처: 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미, 2016

# 고정 분할 기법과 가변 분할 기법

## 3 다중 프로그래밍 환경에서 연속 메모리 할당

- 각 영역별로 독립된 큐가 있는 고정 분할 시스템

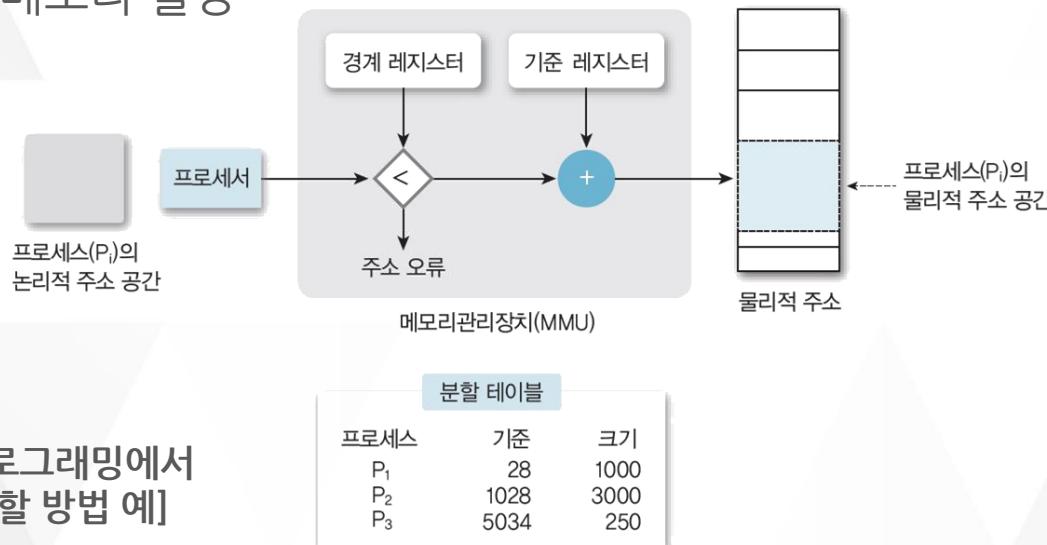


※출처: 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미, 2016

# 고정 분할 기법과 가변 분할 기법

## 4 가변 분할 기법

- ▶ 고정된 경계를 없애고 각 프로세스가 필요한 만큼 메모리 할당



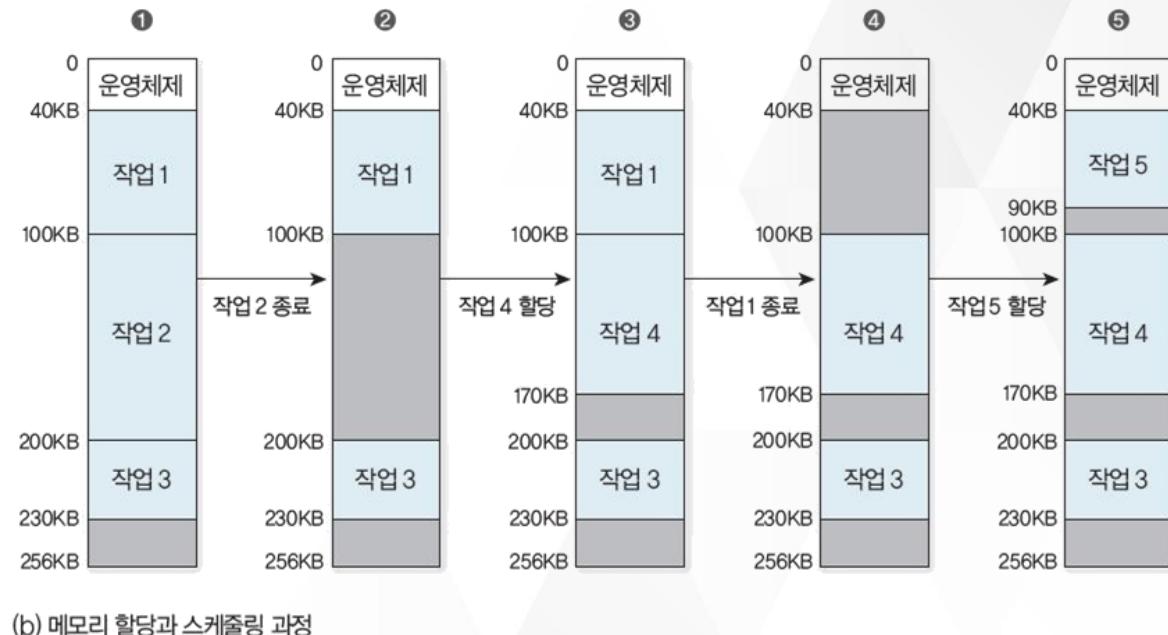
[다중 프로그래밍에서  
가변 분할 방법 예]

※출처: 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미, 2016

# 고정 분할 기법과 가변 분할 기법

## 4 가변 분할 기법

예



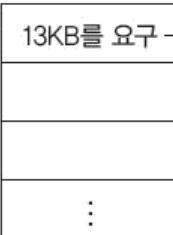
※출처: 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미, 2016

## 5 최초 적합 방법

▶ 예

시작 주소      길이

|       |      |
|-------|------|
| 1000  | 16KB |
| 5000  | 14KB |
| 8000  | 5KB  |
| 11000 | 30KB |



사용 가능 공간 리스트

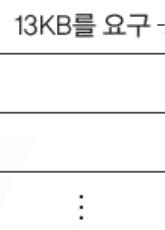


※출처: 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미, 2016

## 6 최적 적합 방법

▶ 예

| 시작 주소                        | 길이   |
|------------------------------|------|
| 8000                         | 5KB  |
| 5000                         | 14KB |
| 1000                         | 16KB |
| 11000                        | 30KB |
| 사용 가능 공간 리스트<br>(공백 크기 오름차순) |      |
| :                            |      |



※출처: 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미, 2016

## 7 최악 적합 방법

▶ 예

시작 주소      길이

|       |      |
|-------|------|
| 11000 | 30KB |
| 1000  | 16KB |
| 5000  | 14KB |
| 8000  | 5KB  |

사용 가능 공간 리스트  
(공백 크기 내림차순)

|          |
|----------|
| 13KB를 요구 |
|          |
|          |
|          |
| ⋮        |



※출처: 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미, 2016