Chapter 03
운영체제의 역할

한양대학교
HANYANG UNIVERSITY

학습목표

- 운영체제의 다양한 기능에 대해 자세히 이해한다.
- 프로세스 관리, 메모리 관리, 저장장치 관리의 개념 및 방법에 대해 이해한다.

학습내용

- 프로세스 관리
  - 프로세스의 개념
  - PCB와 프로세스 상태
- 메모리 관리
  - 메모리 구조와 MMS
  - 가상 메모리
- 저장장치 관리
  - 파일과 확장자
  - 파일 시스템
  - 빈 공간 관리 기법

프로그램과 프로세스의 차이

프로세스의 개념

- 프로세스(process)는 하나의 작업 단위
- 사용자가 마우스를 더블클릭하여 프로그램(program)을 실행하면 그 프로그램은 프로세스가 됨

(a) 프로그램 실행 (b) 프로세스 종료

그림 6-12 프로그램 실행과 프로세스 종료

프로그램과 프로세스 차이

프로그램과 프로세스의 비유

- 레시피 → 조리 → 요리
- 프로그램 → 생성 → 프로세스

그림 6-13 레시피와 요리에 비유한 프로그램과 프로세스

프로세스 관리

프로그램과 프로세스 차이

프로그램

- 어떤 데이터를 사용하여 어떤 작업을 할지 그 절차를 적어 놓은 것
- 하드디스크 같은 저장 장치에 보관하고 있다가 마우스로 더블클릭하면 실행

프로세스

- 프로그램으로 작성된 작업 절차를 실제로 실행에 옮긴다는 의미 (‘실행한다’고 표현)
- 해당 코드가 메모리에 올라와서 작업이 진행된다는 의미

프로세스 관리

프로그램과 프로세스 차이

팟플레이어는 하드디스크에 저장된 프로그램 중 하나

- 저장 장치에 보관된 정적인 상태

팟플레이어를 실행시키면 프로세스가 됨

- 메모리에 올라와서 작업을 수행하는 동적인 상태

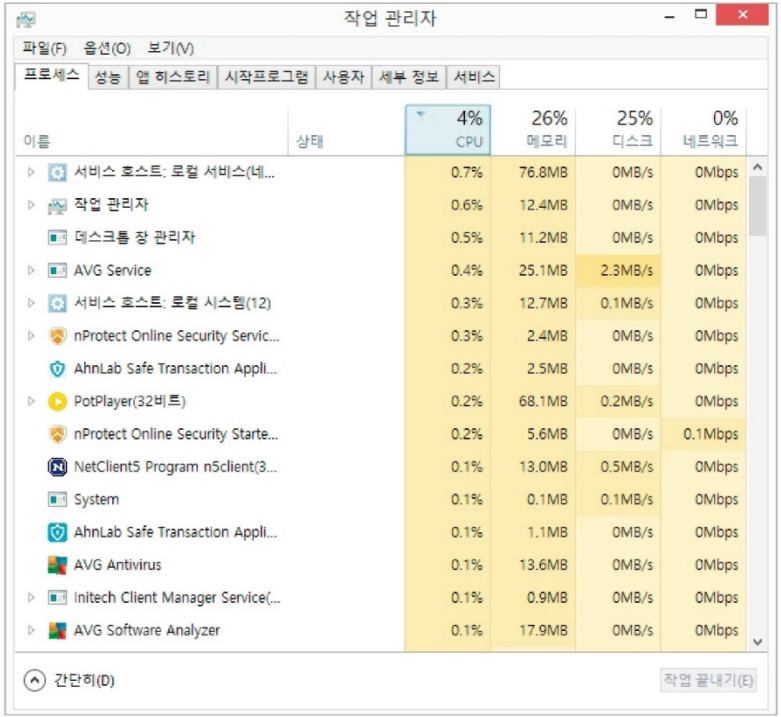


그림 6-14 윈도우 작업 관리자 화면

프로세스 관리

프로그램의 상태

일괄 처리(batch) 방식

- 한 번에 작업을 1개만 처리하는 시스템
- 컴퓨터 초창기의 운영체제나 MS-DOS가 사용하던 방식

시분할(timesharing) 방식

- 프로세스 여러 개가 아주 짧은 시간 동안 CPU를 사용하는 방식
- 프로세스 여러 개가 동시에 실행되는 것처럼 보이는 작업

프로그램의 상태

시분할 작업 예시

그림 6-15 시분할 작업의 주문서 처리

프로세스 관리

프로그램의 상태

프로세스 제어 블록(Process Control Brock, PCB)

- 운영체제에서 주문서에 해당하는 것

프로세스와 프로그램 관계

프로세스 = 프로그램 + 프로세스 제어 블록

프로그램 = 프로세스 − 프로세스 제어 블록

프로세스 관리

프로세스 상태

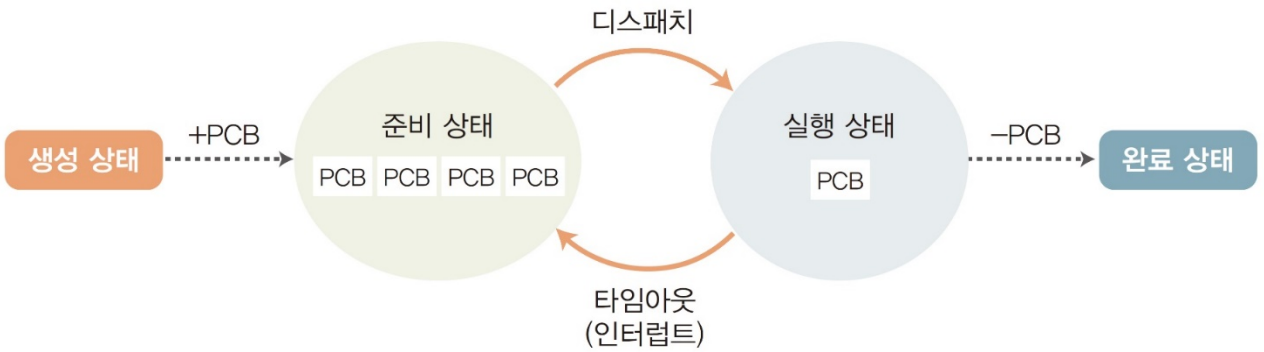


그림 6-16 프로세스 상태

생성 상태(create status)

- 프로세스가 메모리에 올라와 실행 준비를 완료한 상태, PCB 생성

준비 상태(ready status)

- 생성된 프로세스가 CPU를 얻을 때까지 기다리는 상태

프로세스 관리

프로세스 상태

실행 상태(running status)

- 준비 상태에 있는 프로세스 중 하나가 CPU를 얻어 실제 작업을 수행하는 상태
- 자신의 작업이 끝날 때까지 준비 상태와 실행 상태를 반복

완료 상태(terminate status)

- 실행 상태의 프로세스가 주어진 시간 동안 작업을 마치면 완료 상태로 진입
- PCB 반납

프로세스와 메모리 관리

메모리 관리

- 폰노이만 구조의 컴퓨터에서 모든 프로그램은 메모리에 올라와야 실행 가능
- 과거 일괄 처리 시스템에서는 한 번에 작업 하나만 처리했기 때문에 메모리 관리가 어렵지 않음
- 오늘날 시분할 시스템에서는 운영체제를 포함한 여러 프로세스가 한꺼번에 메모리에 올라오기 때문에 메모리 관리가 복잡

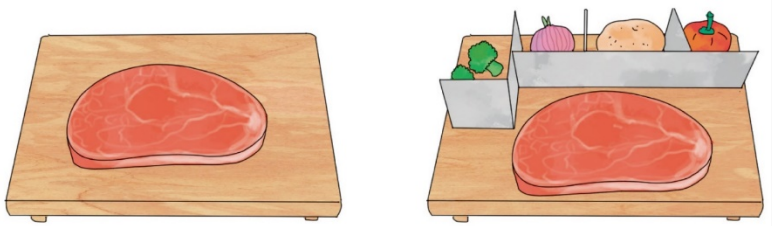


그림 6-17 일괄 처리 시스템과 시분할 시스템의 메모리 관리 예

메인메모리 관리

프로세스와 메모리 관리

메모리 구조

- 운영체제도 프로세스이기 때문에 메모리로 올라와야 실행 가능
- 메모리 관리 시스템(Memory Management System, MMS)
  - 운영체제 영역과 다른 작업 영역 침범 막기
  - 자리 부족 시 빈 공간 확보

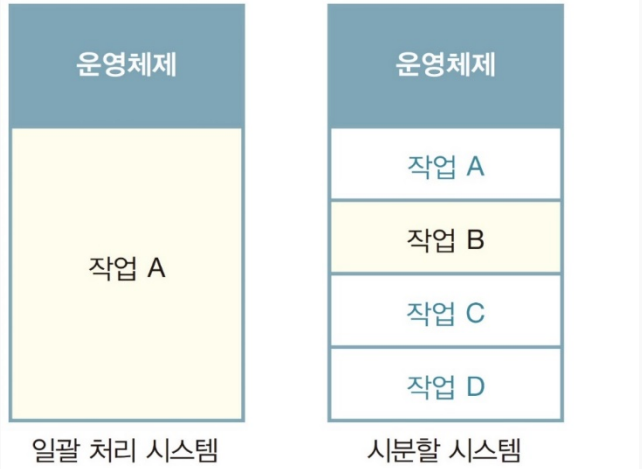
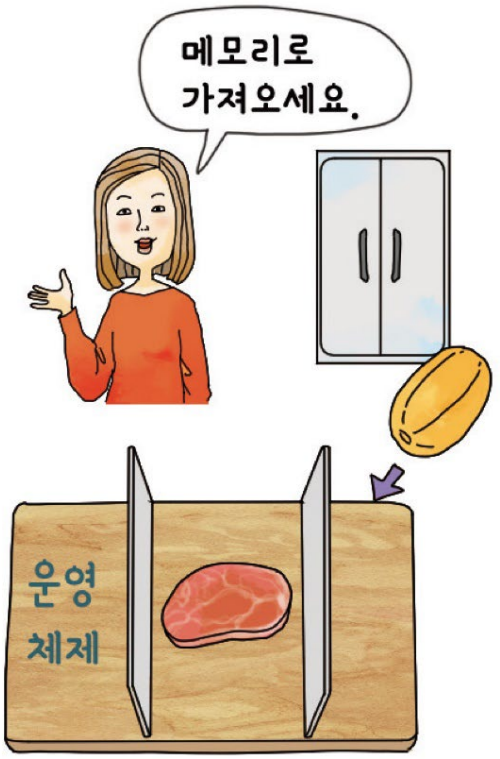


그림 6-18 일괄 처리 시스템과 시분할 시스템의 메모리 구조

프로세스와 메모리 관리

메모리 관리자가 하는 일

- 가져오기 작업
  - 프로세스와 데이터를 메모리로 가져오는 것



(a) 가져오기

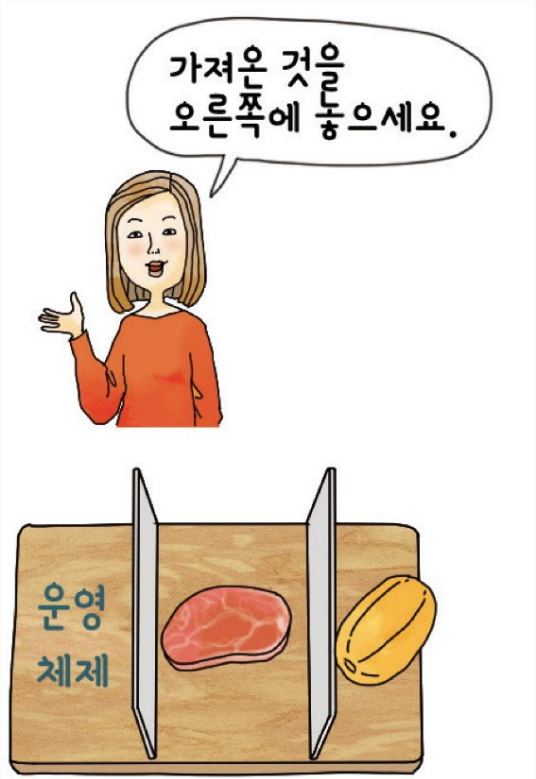
그림 6-19 메모리 관리자가 하는 작업

메인메모리 관리

프로세스와 메모리 관리

메모리 관리자가 하는 일

- 배치 작업
  - 가져온 프로세스와 데이터를 메모리의 어떤 부분에 올려놓을지 결정하는 것
  - 배치 작업 전에 같은 크기로 자르느냐, 프로세스 크기에 맞게 자르느냐에 따라 복잡성이 달라짐



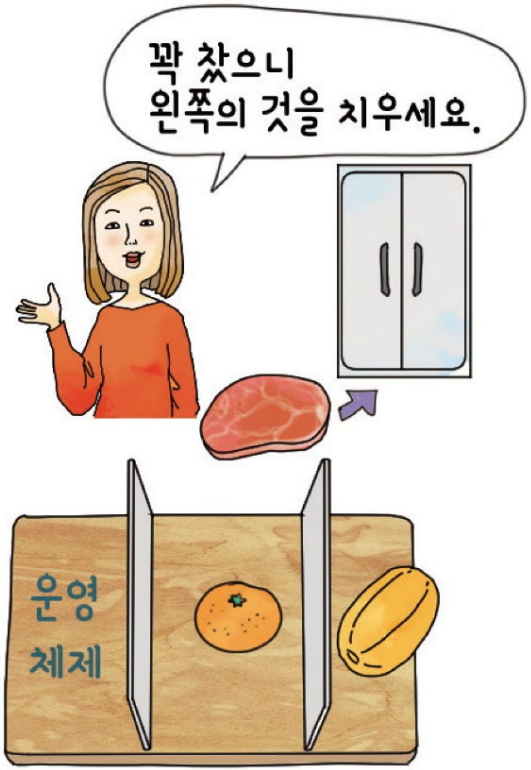
(b) 배치

메인메모리 관리

프로세스와 메모리 관리

메모리 관리자가 하는 일

- 재배치 작업
  - 꽉 차 있는 메모리에 새로운 프로세스를 가져오려고 오래된 프로세스를 내보내는 작업



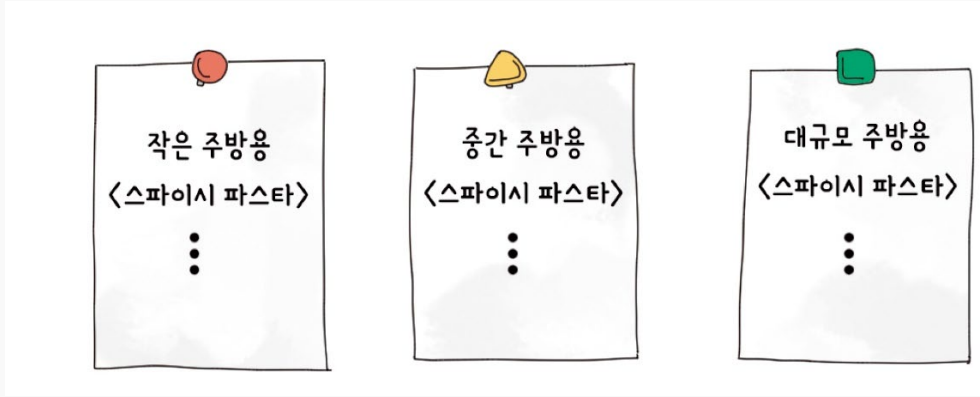
(c) 재배치

메인메모리 관리

가상 메모리

메모리 크기를 고려한 개발

- 사용자 메모리는 1GB부터 16GB까지 다양
- 프로그램에 따라 작은 메모리에서는 동작하지 않을 수 있음
- 메모리 크기를 고려하여 프로그래밍하기는 매우 어려움



메인메모리 관리

가상 메모리

가상 메모리(virtual memory)란?

- 사용자가 가지고 있는 실제 메모리 크기와 프로세스가 올라갈 메모리 위치를 신경 쓰지 않고 프로그래밍을 하도록 지원하는 메모리 시스템
- 실제 메모리 크기와 상관없이 프로세스에 커다란 메모리 공간을 제공하는 기술

가상 메모리

가상 메모리 구성

- 가상 메모리는 <프로세스가 바라보는 메모리 영역>과 <메모리 관리자가 바라보는 메모리 영역>으로 나뉨
- 이론적으로 가상 메모리 크기는 무한대

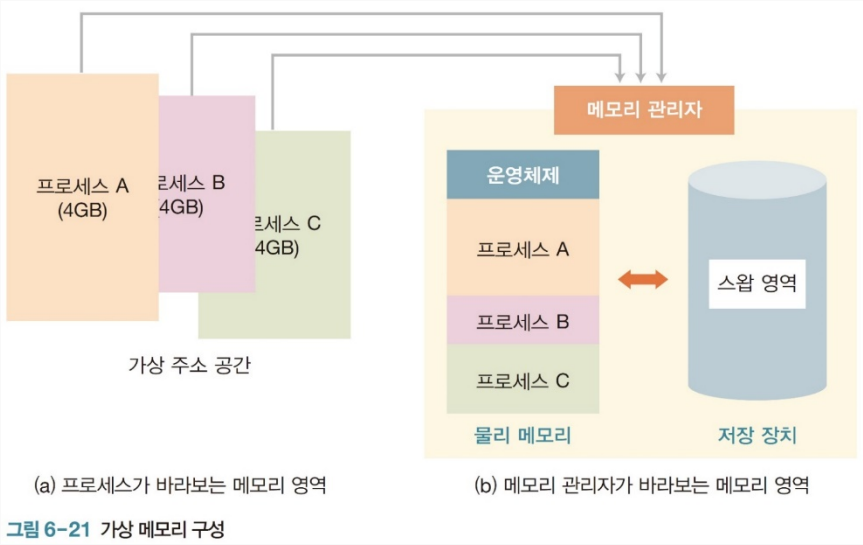


그림 6-21 가상 메모리 구성

메인메모리 관리

가상 메모리

스왑 영역(swap area)

- 메모리가 모자라서 쫓겨난 프로세스를 저장 장치의 특별한 공간에 모아두는 영역
- 하드디스크 같은 저장 장치는 장소만 빌려주고 메모리 관리자가 담당

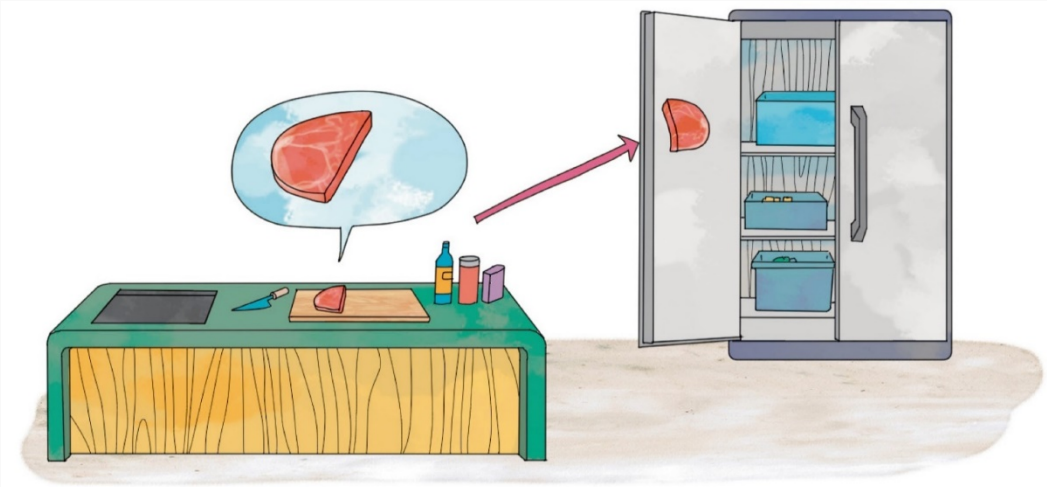


그림 6-22 도마에서 손질 중인 고깃덩어리 일부를 다시 보관 창고에 가져다 놓는 모습

가상 메모리

스왑 영역(swap area)

- [최대 절전 모드] 사용 시 CPU와 메모리의 전력 공급을 끊기 때문에 현재 메모리에 있는 데이터를 옮겨 가는 곳이 스왑 영역

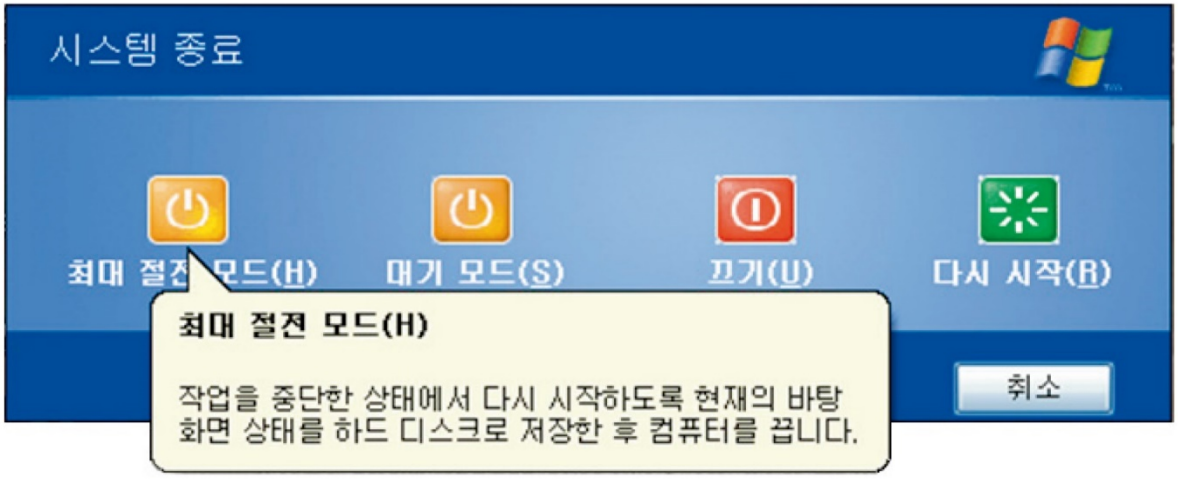


그림 6-23 최대 절전 모드와 스왑

메인메모리 관리

가상 메모리

가상 메모리 크기

- 가상 메모리 시스템은 실제 메모리와 스왑 영역을 활용하여
  메모리 크기에 상관없이 모든 프로그램을 실행할 수 있는 시스템

가상 메모리의 크기

가상 메모리 크기 = 실제 메모리 크기 + 스왑 영역 크기

가상 메모리

가상 메모리 크기

- 스왑 영역 크기는 사용자가 조절 가능하나
  너무 작게 하면 프로그램 실행이 불가능할 수 있음

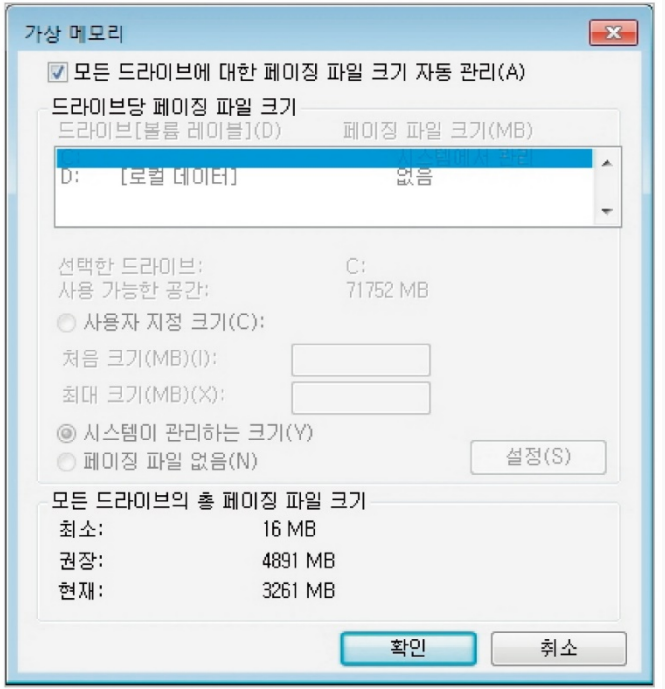


그림 6-24 윈도우 가상 메모리 설정 화면

저장장치 관리

파일

확장자

- 파일은 논리적인 데이터 집합으로 하드디스크나 CD 같은 제 2 저장장치에 저장
- 파일 구분은 확장자를 사용하고, 확장자에 따라 파일 성격 구분
  ** 표기 : 파일이름.확장자
- 초창기 운영체제에서는 파일 이름은 여덟 글자, 확장자는 세 글자로 제한
- 파일 이름
  - 마지막 마침표 다음 글자를 확장자로 인식
  - 현재 경로 이름을 포함하여 최대 255자
  - ₩, /, :, *, ?, ", <, >, | 등은 사용 불가

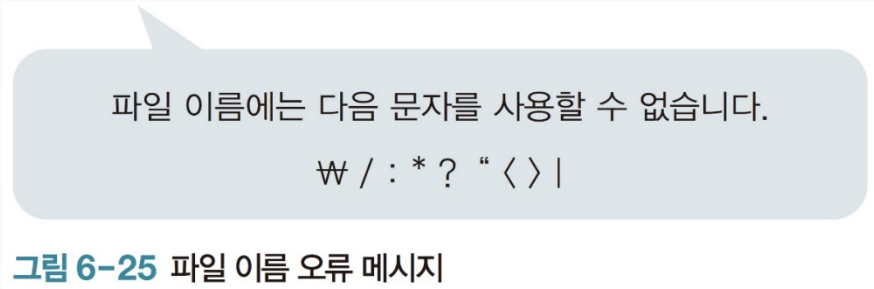


그림 6-25 파일 이름 오류 메시지

저장장치 관리

파일

파일 포맷의 종류

- 이미지 파일 포맷 : BMP, JPG, GIF, PNG 등
- 음악 파일 포맷 : MP3, WAV, OGG, AAC, FLAC
- 응용 프로그램 제작자는 파일 포맷에 따라 확장자를 새로 만들어 사용 가능
- mp3, bmp, jpg, zip 등 전 세계적 표준도 있음

파일 헤더

- 파일 이름, 버전, 크기, 만든 날짜 등 정보가 저장

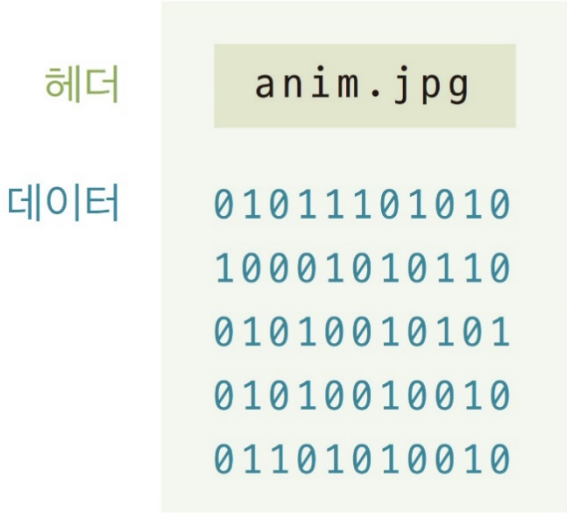


그림 6-26 파일 구조

파일

표 6-2 파일 종류와 확장자

파일	확장자	설명
실행 파일	exe, com	CPU가 작업하는 프로세스 확장자
소스코드 파일	c, cpp, java, py	다양한 소스코드의 확장자
문서 파일	txt, doc, hwp, pdf, ps 등	문서 데이터 파일 확장자
동영상 파일	avi, mp4, mkv, mov	동영상 데이터 파일 확장자
음악 파일	wav, mp3, ogg, flac, aac	음악 데이터 파일 확장자
이미지 파일	bmp, gif, jpg, png, tiff	이미지 데이터 파일 확장자
압축 파일	rar, zip, arc, al	원래 크기보다 줄여서 저장한 파일 확장자

저장장치 관리

파일

실행 파일과 데이터 파일

- 실행 파일 : 운영체제가 메모리로 가져와 CPU를 사용하여 작업하는 파일
  (사용자 요청으로 프로세스가 되는 파일)
- 데이터 파일 : 프로세스나 응용 프로그램이 사용하는 데이터를 모아 놓은 파일

파일

연결 프로그램

- 데이터 파일을 더블클릭하면 실행되는 응용 프로그램

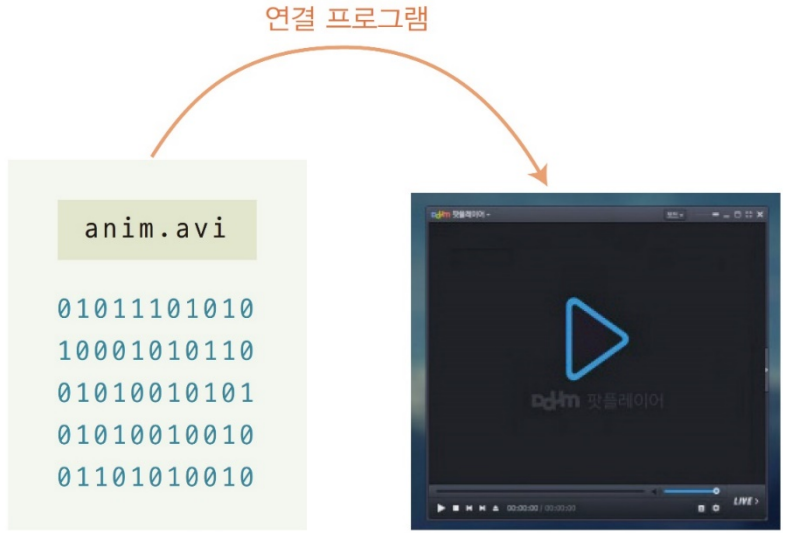


그림 6-27 동영상 파일 실행

저장장치 관리

파일

연결 프로그램 변경 (Windows)

- 변경하는 파일에서 마우스 오른쪽 버튼
  → [속성] 메뉴 선택
  → '변경' 버튼
  → 프로그램 선택

그림 6-28 연결 프로그램 변경

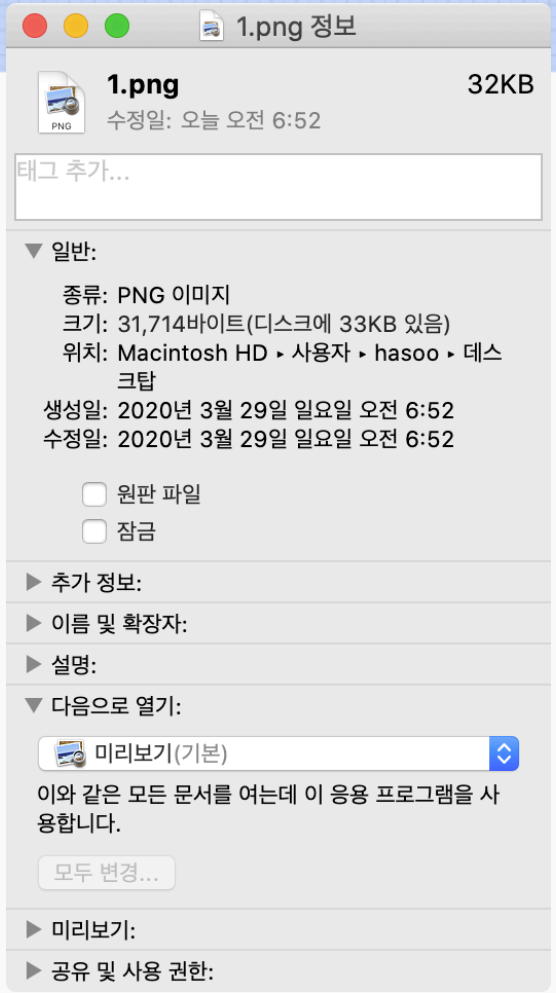
저장장치 관리

파일

연결 프로그램 변경 (OSX)

- 변경하는 파일에서 마우스 오른쪽 버튼
  → [정보 가져오기] 메뉴 선택
  → [다음으로 열기] 에서
  원하는 프로그램 선택

- 해당 종류 파일에 대해 일괄 적용하려면
  프로그램 선택 후 [모두 변경] 버튼 클릭



디렉터리

파일과 디렉터리

- 관련 있는 파일을 하나로 모아 놓은 곳
- 여러 층으로 구성 가능(디렉터리 밑에는 또 다른 디렉터리를 만들 수 있음)
- 최상위 디렉터리를 루트 디렉터리(root directory)라고 함

그림 6-29 파일과 디렉터리

저장장치 관리

디렉터리

계층 구조

- 역슬래시(\ )는 루트 디렉터리를 의미
- 한글 자판에서 역슬래시는 ₩(원화 표시)로 대체
- 디렉터리 헤더에는 디렉터리 이름, 만든 시간, 접근 권한 등 정보가 기록되어 있음

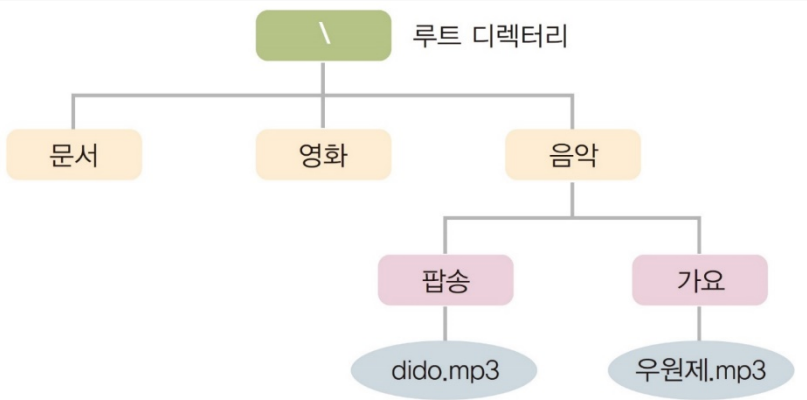


그림 6-30 디렉터리 계층 구조

저장장치 관리

파일 시스템

파일 테이블

- 파일이 저장된 파일 이름, 위치 정보 등이 저장
- 모든 운영체제는 고유의 파일 테이블을 가짐
- 윈도우는 FAT(File Allocation Table)나 NTFS,
  유닉스는 i-node 같은 파일 시스템 운영

파일 시스템

파일 테이블

파일 A	1, 3, 9
파일 B	4, 2
파일 C	13
파일 D	15, 12
파일 E	23, 7

파일 테이블

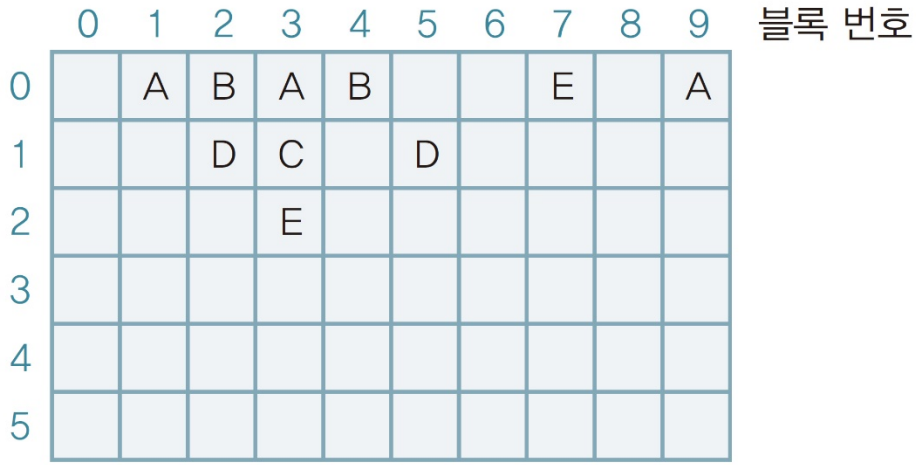


그림 6-31 파일 테이블

저장장치 관리

파일 시스템

포맷팅(formating)

- 디스크에 파일 시스템을 탑재하고 디스크 표면을 초기화하여 사용할 수 있는 형태로 만드는 작업
  - 빠른 포맷팅 : 데이터는 그대로 둔 채 파일 테이블을 초기화하는 방식
  - 느린 포맷팅 : 파일 시스템을 초기화할 뿐 아니라 저장 장치의 모든 데이터를 0으로 만들어 버림

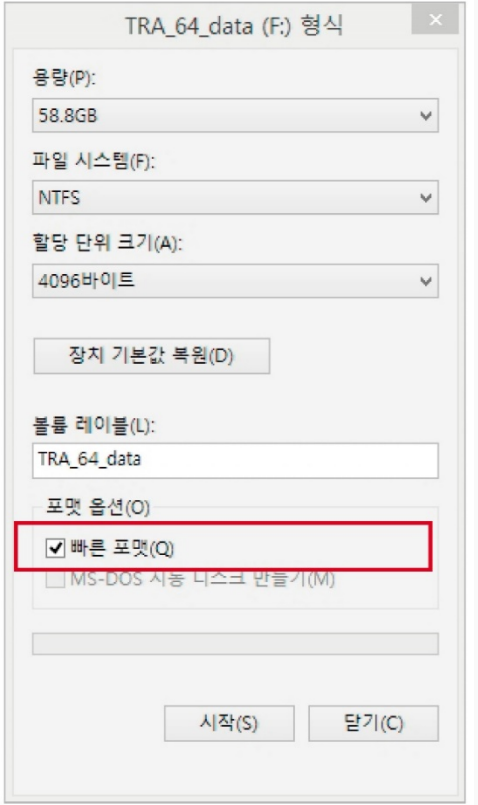


그림 6-32 저장 장치 포맷 화면

저장장치 관리

파일 시스템

섹터

- 하드디스크의 물리적인 구조상 가장 작은 저장 단위
- 섹터마다 주소를 부여하면 너무 많은 양의 주소가 필요하기 때문에 파일 관리자는 여러 섹터를 묶어 하나의 블록으로 만들고 블록 하나에 주소 하나를 배정

블록

- 저장 장치에서 사용하는 가장 작은 단위
- 한 블록에 주소 하나를 할당
- 데이터는 운영체제와 저장 장치 간에 블록 단위로 전송

저장장치 관리

파일 시스템

블록 크기

- 블록 크기는 시스템마다 다름
- 포맷할 때 시스템이 정한 기본 블록 크기를 사용 또는 4,096B~64KB의 다양한 블록 크기를 직접 지정
- 블록 크기를 작게 설정하면 저장 장치를 효율적으로 쓸 수 있지만, 파일이 여러 블록으로 나뉘어 파일 입출력 속도가 느려짐

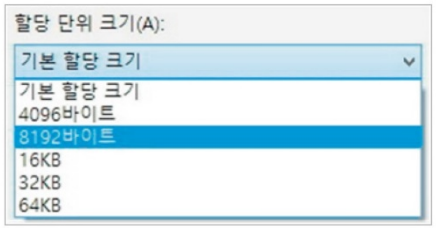


그림 6-33 윈도우의 블록 크기 설정

파일 시스템

조각화(단편화)

- 하드디스크를 처음 사용할 때는 데이터가 앞부터 차곡차곡 쌓이지만, 사용하다 보면 파일이 삭제되면서 중간중간 빈 공간이 생김
- 하드디스크에 조각이 많이 생기면 큰 파일을 여러 조각으로 나누어 저장 → 성능 저하

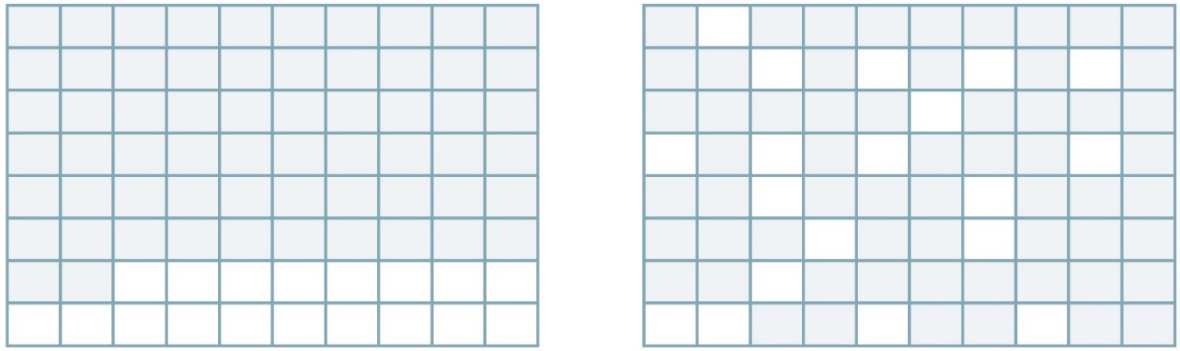


그림 6-34 초기 상태와 조각 난 상태

저장장치 관리

파일 시스템

조각모음(defragmentation)

- 주기적으로 조각모음 실행
- 시간이 오래 걸리는 작업이므로 작업이 없는 특정 시간에 조각모음을 실행
- USB 메모리, SSD 등 반도체를 사용하는 저장 장치는 조각모음을 하지 않음
- 조각모음을 통해 특정 위치의 메모리만 계속 사용하면 수명 단축

저장장치 관리

파일 시스템

FAT 파일 시스템 구조

- 왼쪽 테이블은 파일 정보와 함께 파일의 시작 블록 정보를 가짐
- 파일 B : 2 → 4 → 12 → 8(null)번 블록
- 파일 C : 0 → 3 → 10 → 6 → 9 → 7 → 11 → 14(null)번 블록

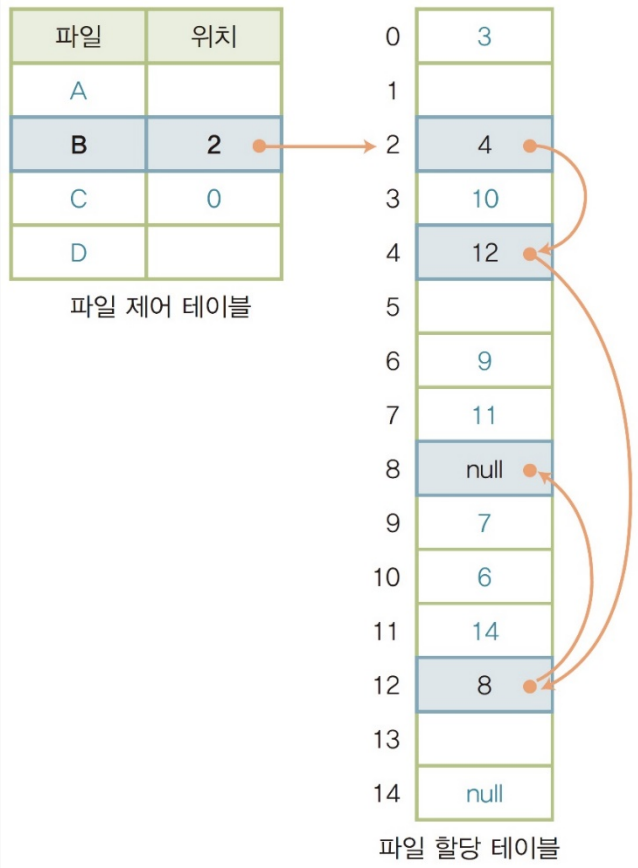


그림 6-35 FAT 파일 시스템 구조

저장장치 관리

파일 시스템

FAT32와 NTFS

- 윈도우는 파일 시스템으로 FAT32 또는 NTFS를 사용
- FAT32는 32GB까지 지원하고 파일 하나의 크기가 4GB로 한정
- USB 메모리는 대부분 FAT32를 사용
  - 4GB보다 큰 파일을 저장하려고 하면 빈 공간이 있어도 '빈 공간 없음'이라는 메시지를 표시
  - FAT32가 4GB 이상의 파일을 지원하지 않아 발생하는 문제
  - NTFS 파일 시스템으로 바꾸면 해결
  - 대신 NTFS는 자동차나 오디오와 같은 기기에서 인식이 안될 수 있음.

저장장치 관리

빈 공간 관리

빈 공간 리스트(free block list)

- 디스크에 파일을 저장할 때 모든 테이블을 뒤져 빈 공간을 찾는 것은 비효율적
- 빈 공간을 효율적으로 관리하려고 파일 시스템은 빈 블록 정보만 모아 놓은 빈 공간 리스트(free block list)를 유지

저장장치 관리

빈 공간 관리

파일 B를 지우는 경우 빈 공간 리스트의 변화

- B가 사용한 2, 4, 12, 8번 블록의 내용이 지워지지 않고 빈 공간 리스트에 삽입
- 파일을 삭제하면
  실제 파일이 지워지는 것이 아니라
  '파일 테이블의 헤더를 삭제하고
  사용한 블록을 빈 공간 리스트에 등록함

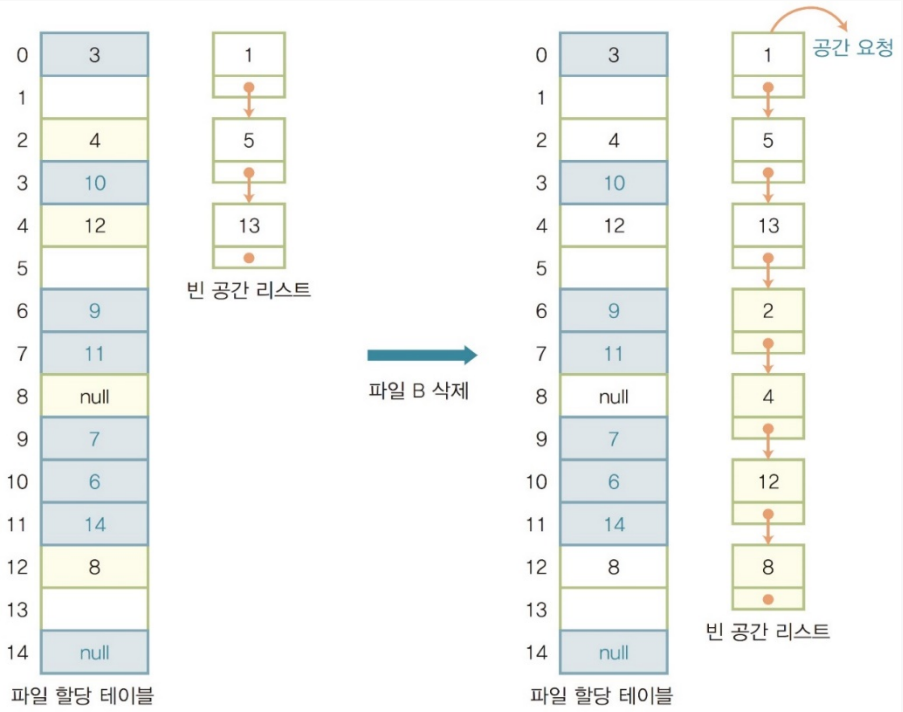


그림 6-36 파일 B를 삭제하기 전과 삭제한 후 빈 공간 리스트의 변화

저장장치 관리

빈 공간 관리

빈 공간 리스트의 변화

- 빠른 포맷은 데이터를 그대로 남겨 둔 채 파일 테이블만 초기화하는 방식
- 블록 내용을 지우지 않고 빈 공간 리스트에 삽입하는 것은 해당 블록에 새로운 데이터를 덮어 쓰지 않는 한 원래 데이터를 복구할 수 있는 여지 있음
- 빈 공간 리스트를 보고 새로운 블록을 할당할 때는 리스트에 먼저 들어온 블록부터 할당
- 리스트에 먼저 들어온 블록부터 할당하기 때문에 휴지통에서 삭제한 파일이나 빠른 포맷을 한 이후에 데이터를 되살릴 수 있음

빈 공간 관리

파일 삭제 또는 포맷 후 저장 장치 상태

그림 6-37 파일을 지우거나 포맷한 이후의 저장 장치 상태

저장장치 관리

빈 공간 관리

데이터 관리

- 파일을 지우면 내용이 사라지는 것이 아니라 파일 테이블에서 파일 정보만 삭제
- 새로운 파일을 저장하면 방금 지운 파일 공간에 덮어 쓰는 것이 아닌
  앞의 빈 공간부터 덮어 씀
- 스마트폰에서 사용하는 클라우드 저장 장치는
  자신의 스마트폰에 있는 전화번호부, 사진, 파일을 클라우드로 전송
  ** 따라서 데이터를 지워도 클라우드로 전송한 데이터를 지우기 전까지는 시간 차이가 있을 수 있음

Q & A

Thank You