

Slide 0

**3부 저장장치 관리
(11장. 파일시스템 인터페이스)****파일개념**

- 파일이란?
 - 정보를 저장하는 논리적 저장단위
 - 보조기억장치에 기록된 관련 정보의 집합
 - 논리적 보조기억장치의 가장 작은 할당 요소

Slide 1

- 파일 속성
 - 이름: 사용자에게 인식되는 기호화된 이름
 - 식별자: 파일시스템 내에서 각 파일을 구분하기 위한 식별자
 - 유형: 여러가지 유형을 지원
 - 위치: 저장 매체에 파일이 저장된 위치
 - 크기: 파일의 크기, 최대 허용가능한 크기등
 - 보호: 파일의 접근제어 정보
 - 시간, 날짜, 사용자 식별: 파일의 생성, 마지막 변경, 마지막 사용등에 관한 정보

파일개념 (계속)

- 파일 조작
 - 기본적인 6개의 파일 조작 명령
 - * 파일 생성
 - 파일을 저장할 수 있도록 파일 시스템 내에서 빈 공간을 찾음
 - 새로 생성된 파일에 대한 항목을 디렉토리에 만들
 - * 파일 기록
 - 파일내에 기록할 위치를 가리키는 기록 포인터를 유지함
 - 기록 포인터는 파일에 기록이 일어날 때마다 갱신됨
 - * 파일 읽기
 - 파일 안의 읽을 위치에 대한 읽기 포인터를 유지함
 - 읽기가 발생할 때마다 읽기 포인터가 갱신됨
 - * 파일 안에서의 재설정
 - 실제적인 파일 내용의 입출력이 발생하는 것이 아님
 - 단지, 파일에서의 위치만 재설정됨
 - 파일 탐색 (file seek) 라고 불림

Slide 2

파일개념 (계속)

- * 파일 삭제
 - 디렉토리에서 해당되는 파일을 찾아서
 - 해당 파일이 점유하고 있던 모든 공간을 해제하고
 - 디렉토리에서 해당 항목을 삭제함
- * 파일 절단 (truncating a file)
 - 파일의 속성은 유지하고 단지 파일의 크기만 0으로 함
 - 파일에게 할당되었던 공간은 모두 해제됨
- 기타 명령
 - * 파일 첨가 (appending): 기존 파일의 끝에 새로운 내용을 추가함
 - * 파일 이름 변경 (renaming): 파일의 이름을 변경함
 - * 파일 속성을 설정 및 취소
 - * 파일 열기 (open): 디스크를 검색해 해당 파일의 정보를 이용 연결을 생성함
 - * 파일 닫기 (close): 파일의 정보를 디스크로 저장하고 연결을 해제함

Slide 3

파일개념 (계속)

Slide 4

- 파일 조작을 위해 운영체제는 보통 두개의 테이블을 사용
 - * 프로세스별 테이블
 - 프로세스의 PCB에 유지됨
 - 해당 프로세스가 연 파일 관리
 - 파일 포인터 나 파일 접근 모드등이 유지됨
 - * 시스템 전체 테이블
 - 프로세스와 독립적으로 존재함
 - 디스크 내의 파일의 위치, 접근 날짜, 파일 크기등이 유지됨
 - 열린 계수 (open count)로 파일을 연 프로세스의 수를 유지함

파일개념 (계속)

Slide 5

- 파일 유형
 - 파일의 확장자를 이용하여 파일 유형을 식별
 - 운영체제는 일부 확장자 (보통 실행, 텍스트, 이진)만 식별

file type	usual extension	function
executable	exe, com, bin or none	read to run machine- language program
object	obj, o	compiled, machine language, not linked
source code	c, cc, java, pas, asm, a	source code in various languages
batch	bat, sh	commands to the command interpreter
text	txt, doc	textual data, documents
word processor	wp, tex, rrf, doc	various word-processor formats
library	lib, a, so, dll, mpeg, mov, rm	libraries of routines for programmers
print or view	arc, zip, tar	ASCII or binary file in a format for printing or viewing
archive	arc, zip, tar	related files grouped into one file, sometimes com- pressed, for archiving or storage
multimedia	mpeg, mov, rm	binary file containing audio or A/V information

파일개념 (계속)

Slide 6

- 파일 구조
 - 파일의 유형에 따라서 특별한 파일 구조가 필요
(예로, 실행파일이 적재될 기억장소의 위치, 첫번째 시작명령의 위치등의 정보)
 - 모든 운영체제는 최소한 실행파일을 위한 파일구조를 지원해야함
(그래야 시스템 프로그램을 적재하고 실행가능함)
 - 자주 사용되는 구조 → 운영체제가 지원하게하는 것이 유용
 - 유연성 제공을 위해 자주 사용되지 않는 것은 운영체제가 지원하지 않게 함

파일개념 (계속)

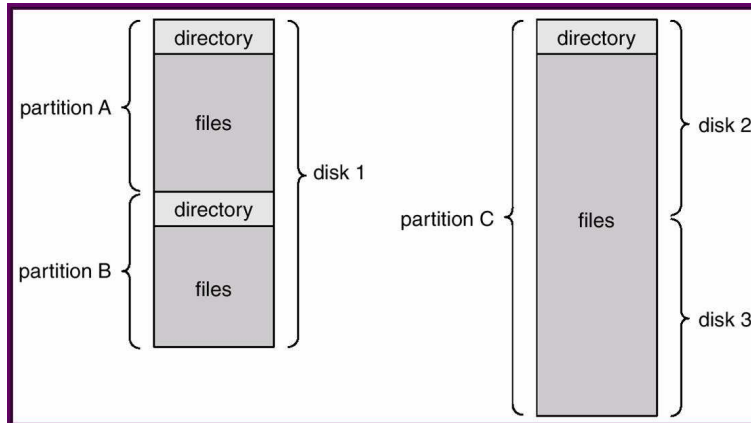
Slide 7

- 내부 파일 구조
 - UNIX
 - * 모든 파일을 바이트의 열로 간단히 정의
 - * 모든 바이트는 파일의 시작(혹은 끝)에서 변위로 위치 지정 가능
(논리적 레코드는 1바이트)
 - * 실제로 파일이 디스크에 저장될 때는 섹터크기로 결정된 블록 크기를 가짐
 - * 조합(packing): 논리적 레코드 → 물리적 레코드 (블록)
 - * 분해(unpacking): 물리적 레코드 → 논리적 레코드 (바이트)
 - * 파일의 두 크기 (윈도우에서 크기와 디스크 할당 크기)
 - 실제크기 (바이트 단위)
 - 할당크기 (블록 단위)
 - 예)
 - 블록 크기: 512 바이트
 - 파일 크기: 1949 바이트
 - 할당 크기: 2048 바이트 (99바이트는 낭비: 내부 단편화)

디렉토리 구조

- 설명
 - 디스크는 파티션(partition)으로 분할됨
 - 디스크는 최소 하나의 파티션을 가짐
 - 여러 디스크를 하나의 파티션으로 사용할 수도 있음

Slide 8



디렉토리 구조 (계속)

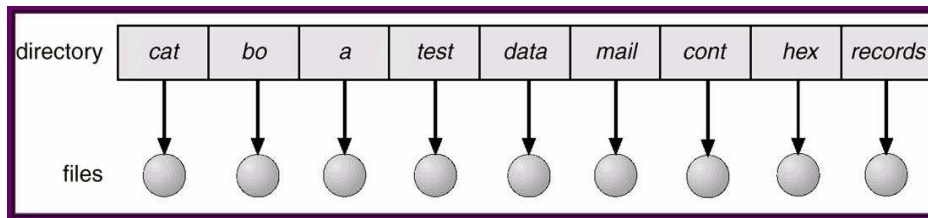
Slide 9

- 파티션의 구성: 파일 + 파일관련 정보 (디렉토리 혹은 볼륨 테이블에 유지됨)
- 디렉토리와 관련된 동작
 - * 파일 탐색: 특정 파일을 찾기 위해 디렉토리 구조를 탐색
 - * 파일 생성
 - * 파일 삭제
 - * 디렉토리 나열
 - * 파일 재명명
 - * 파일 시스템 순회(traverse): 파일 시스템 내 모든 디렉토리를 방문하는 것

디렉토리 구조 (계속)

- 단일 단계 디렉토리
 - 모든 파일이 하나의 디렉토리에 포함되는 구조
 - 가장 쉬운 방법이나 단점이 존재
 - 단점
 - * 다수의 사용자가 다수의 파일을 유지할 경우
 - * 모두 다른 이름을 가져야함
 - * 파일의 유지 및 관리가 어려움

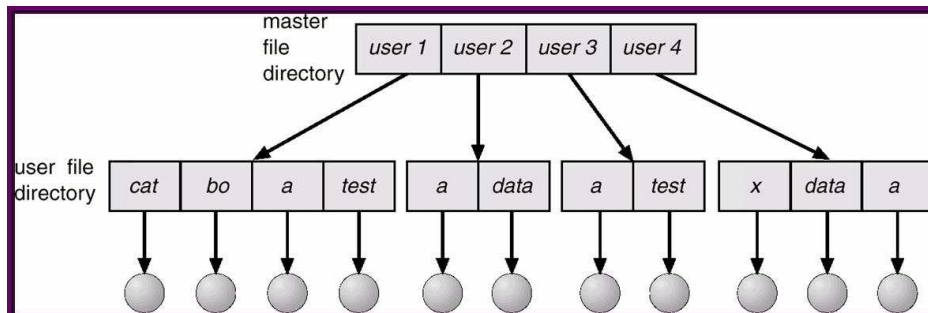
Slide 10



디렉토리 구조 (계속)

- 2단계 디렉토리
 - 사용자별 다른 디렉토리를 가능하게 함
 - 디렉토리가 다른 경우 같은 파일이름이 가능
 - 파일은 경로(path)로 찾아감

Slide 11



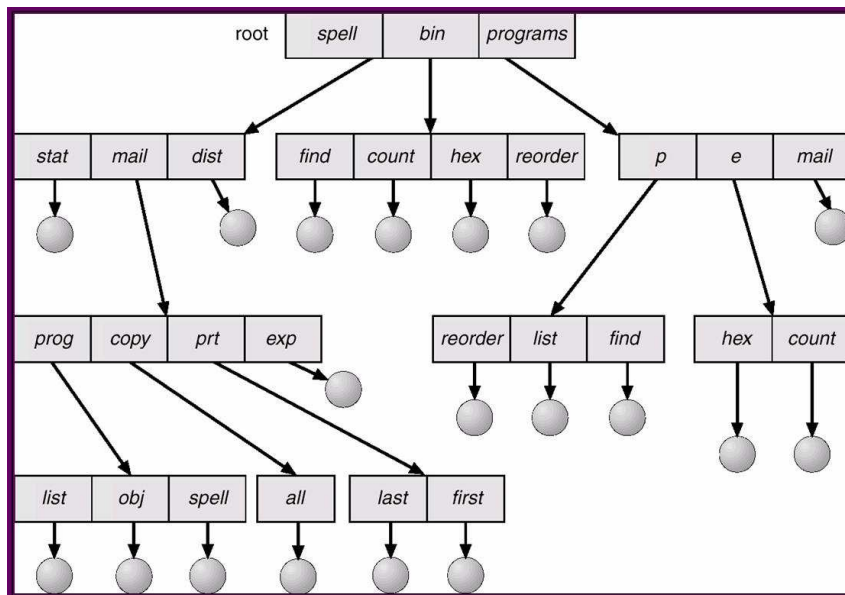
디렉토리 구조 (계속)

Slide 12

- 트리 구조의 디렉토리
 - 트리 형태로 디렉토리를 확장 가능
 - 최상위 디렉토리 → 루트 디렉토리 (root directory)
 - 현재 작업 디렉토리 → 현재 디렉토리 (current directory)
 - 절대경로: 루트 디렉토리로 부터의 경로
 - 상대경로: 현재 디렉토리로 부터의 경로

디렉토리 구조 (계속)

Slide 13



디렉토리 구조 (계속)

Slide 14

- 디렉토리나 파일의 공유
 - 파일의 공유를 위하여 보통 두개의 방법이 사용됨
 - 하드 링크 (hard link)
 - * 새로운 파일형태로 생성됨 (그러므로 디렉토리는 안되고 파일만 가능)
 - * 단 파일의 내용은 같은 내용을 가리킴
 - * 파일에 참조계수 (reference count)가 증가됨
 - * 파일 하나를 삭제시 참조계수만 감소함 (참조 계수가 0이면 실제 삭제됨)
 - * 유닉스 계열에서 ln 명령이 이에 해당함
 - 기호 링크 (symbolic link)
 - * 파일 또는 디렉토리에 대한 새로운 이름의 포인터
 - * 단지 기존의 파일이나 디렉토리에 대한 포인터임
 - * 기호 링크 삭제시 포인터만 삭제됨 (원본은 그대로 유지됨)
 - * 기호 링크가 있는데 원본이 삭제되는 경우 (파일 없음 오류 출력)
 - * 유닉스 계열에서 ln -s 명령이 이에 해당함

디렉토리 구조 (계속)

Slide 15

- 다른 사용자간 파일의 공유
 - 파일이나 디렉토리에 소유및 공유에 대한 속성을 두어서 해결
 - 유닉스의 경우
 - * 파일이나 디렉토리에 소유자, 그룹, 일반의 속성을 둠
 - * 그룹은 사용자들의 부분집합으로 설정 가능
 - * 파일이나 디렉토리의 속성을 설정함으로 공유가능
 - * 공유시 해당 파일의 동작별(유닉스의 경우 읽기, 쓰기, 실행)로 설정 가능
- 다른 시스템 간의 파일의 공유
 - 간단한 방법 →ftp 를 통한 파일의 공유
 - 편리한 방법 →분산 파일 시스템 (DNS: Distributed File System) 사용
 - 새로운 방법 →WWW 를 사용한 방법

보호

- 보호
 - 물리적 손상으로부터의 보호 (신뢰성): 백업으로 보호
 - 부적합한 접근으로부터의 보호: 접근 제어(access control)로 보호
 - 접근 제어
 - 접근을 유형별로 제어
- Slide 16**
- 접근 유형
 - * 읽기
 - * 쓰기
 - * 실행
 - * 추가
 - * 삭제
 - * 리스트: 파일의 속성이나 이름 보기
 - 기타 변경, 복사, 편집등은 위의 접근 유형에 따라 제어됨
(예로, 파일의 복사는 읽기 권한이 있으면 허락됨)

보호 (계속)

- Unix에서의 보호방법
 - 사용자를 근거로 접근허용 여부를 결정
 - 파일별로 접근제어 목록이 있어서 이를 이용하여 접근을 제어함
 - 사용자 분류
 - * 소유자(owner): 파일을 생성한 사용자
 - * 그룹(group): 소유자가 속한 그룹
 - * 일반(universe): 시스템에 등록된 모든 사용자
 - 각 사용자별로 세개의 접근 유형(읽기,쓰기,실행)을 유지
 - * 사용자(3) × 접근유형 (3) = 9비트로 접근 제어
- | 소유자 | 그룹 | 일반 |
|-----|-----|-----|
| rwX | rwX | rwX |
- 예) test라는 파일의 접근을
 - * 소유자는 읽기,쓰기,실행 가능
 - * 그룹은 읽기,실행가능
 - * 일반은 읽기만 가능하게 하려면
 - * `chmod 754 test`
- Slide 17**