

Chapter 5. Files and Directories

파일 시스템

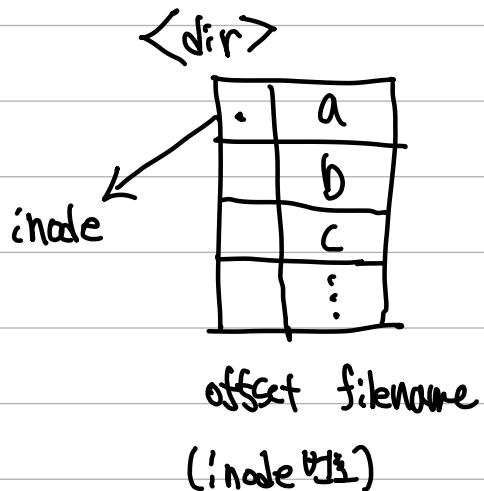
collection of files and attributes (이름, 이름 등)

physical 간 위치 대신 file name 과 offset 으로 특정.

디렉토리

파일의 한 종류.

directory entry (filename 과 offset 둘이 짚여온 것) 들을 포함하는 파일.



/ 로 시작하면 root 부터 시작하는 절대 경로.

/ 링크 ..이나 . 같은 걸로 시작하면 상대경로. → 상대경로의 가진수는 무한대!

/ root 디렉토리

.. 부모 디렉토리

. 현재 작업 디렉토리

~ home 디렉토리

int chdir (경로) 성공 0, 실패 -1 현재 프로세스의 working dir를 변경.

char* getcwd (buffer, size) 현재 작업 디렉토리 경로를 buffer에 반환.
성공 buffer 주소, 실패 NULL.

최대 size? → PATH_MAX 사용.

PATH_MAX 없으면? → #ifndef PATH_MAX
#define PATH_MAX 255
#endif

파일 명으로만 호출 했을 때 → PATH 설정 변수 검색 (.profile에서 설정 가능)
PATH = ... : ... : ...

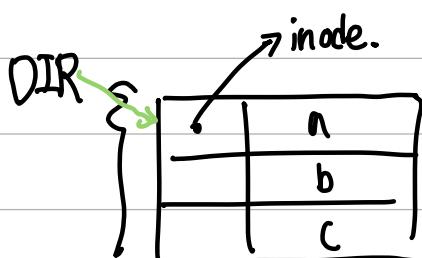
PATH 앞쪽에 . 등록은 위험 → 현재 디렉토리에 ls 명령이 쓰는 명령어 이후의
작성코드가 실행될 수 있음.

DIR* opendir (디렉토리 이름)

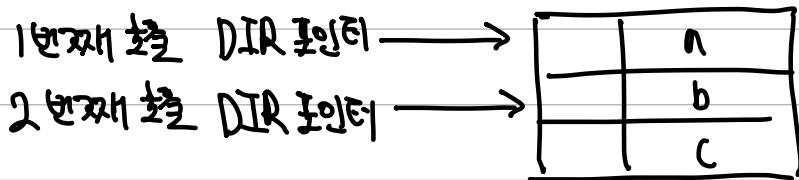
DIR 초기는 directory stream을 표현



directory entry들이 노마이로



Struct dirent * readdir (DIR * dirptr)



{ offset, filename } 를 구성된 dirent 만큼, DIR 포인터는 다음으로 이동시킴.
(inode 번호)

int closedir (DIR * dirptr) 성공시 0, 실패시 -1

void rewinddir (DIR * dirptr) 초기화 DIR 포인터 →

초기화 DIR 포인터 →

a
b
c

(inode 번에 있는 정보들)

stat (path, struct stat * buffer)

stat: 파일의 메타데이터 정보

lstat (path, struct stat * buffer)

- └ atime 마지막 access 시간
- └ mtime 마지막 수정 시간
- └ stime 마지막 stat(변경) 시간.

link (링크) 가 같은 symbolic link) 가 주어지면

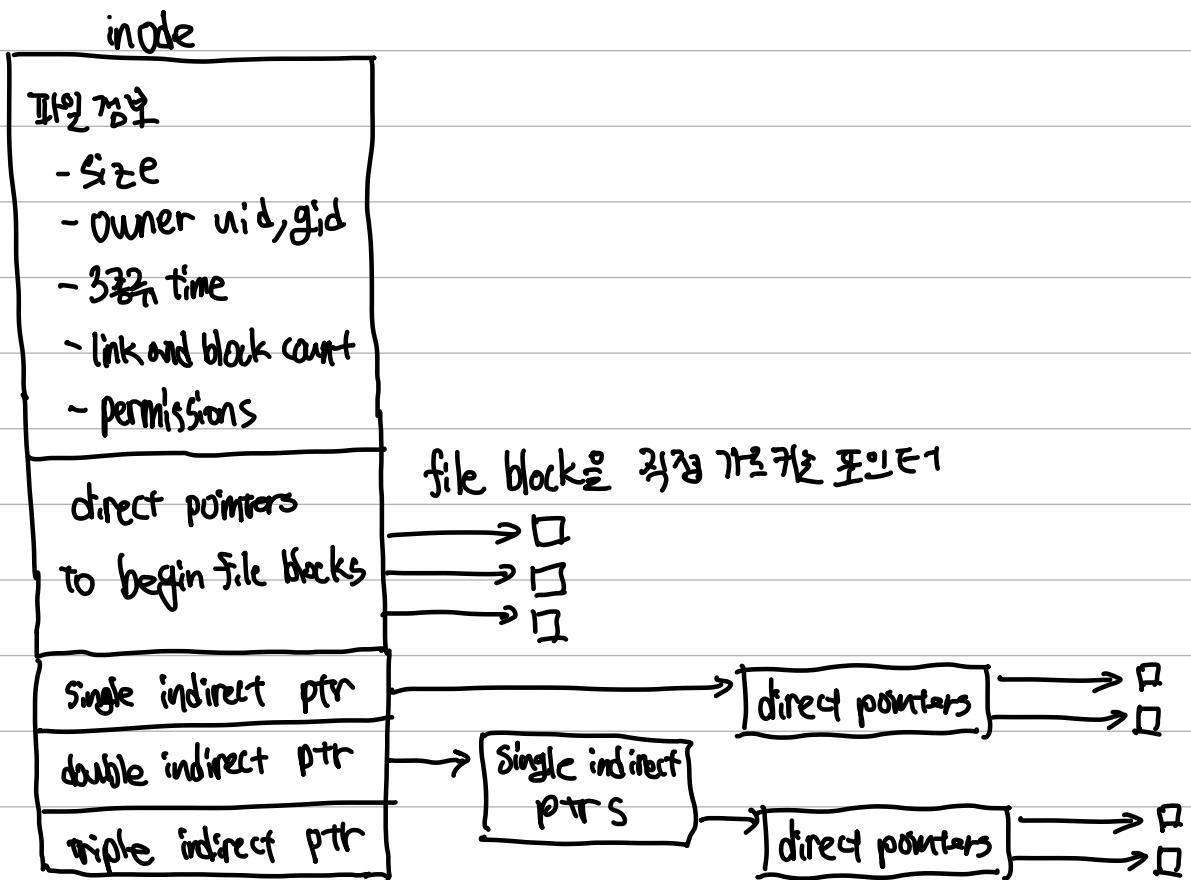
stat은 link가 가르키는 원본 파일의 정보 반환.

lstat은 link 파일 자체의 정보 반환.

inode

파일마다 하나씩 생성. 최대 inode 개수는 시스템에 정해져 있음.

inode의 크기는 모두 같음.



디렉토리 파일, directory entry를 저장.

directory entry = file location (inode number) + file name

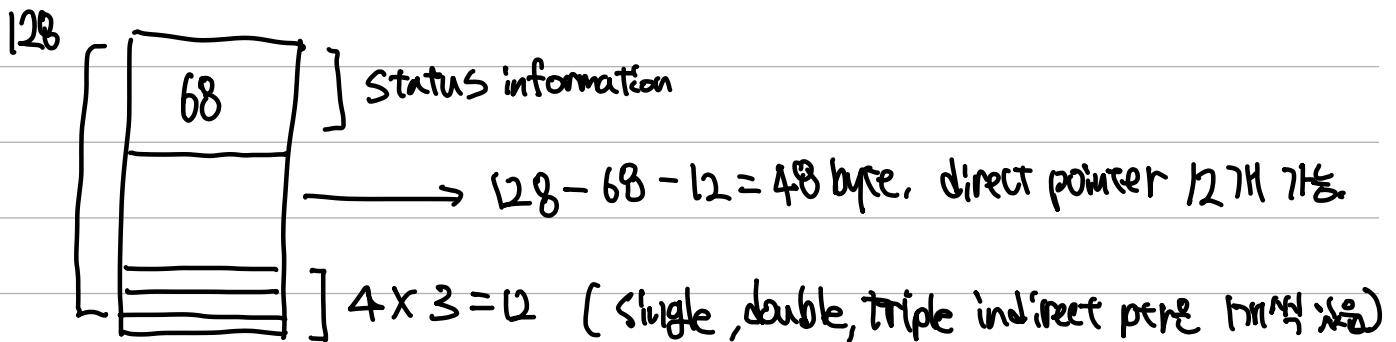
장점: 파일명 수정하거나, 이동할 때 directory entry만 수정하면 됨.

단점: 실제 파일이 하나여도 가르치는 링크를 여러개로 할 수도 있다. (link)

파일 최대 크기 계산해보기

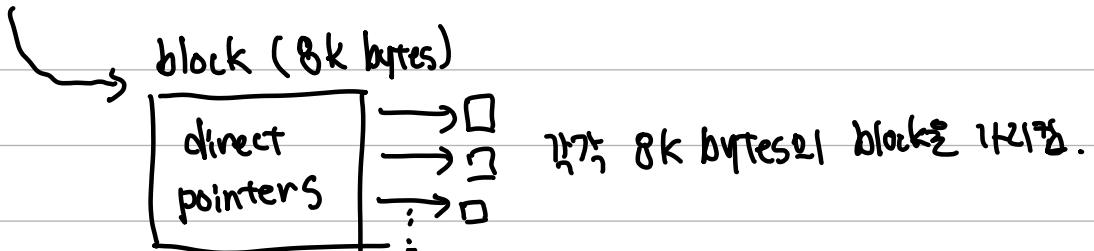
inode: 128 byte, pointer: 4 byte, status info: 68 byte

block size: 8k byte, block pointer: 32 bit



12 direct pointer 2 레벨의 디렉터리 파일 구조: $12 \times 8\text{k byte} = 96\text{ kB}$ ($8\text{k} = 8 \times 2^{10}$)
 $12 \times 8192 = 98304$ byte.

single indirect pointer 0 레벨의 파일 구조?



$$8192 / 4 = 2048 \text{ 개의 direct pointers.}$$

$$2048 \times 8192 = 16777216 \text{ bytes}$$

$$2 \times 2^{10} \times 8 \times 2^{10} = 16 \times 2^{20} = 16 \text{ MB}$$

Link

link는 filename과 inode 외의 관계

[Hard Link : directory entry는 hard link. filename을 inode로 직접 연결]

[Symbolic Link (soft) : 파일안에 원본 파일은 없거나 있어서 대안가면 원본의 inode 사용.]

각 inode에는 2개의硬link를 가리키는 hard link의 개수 counter가 있다.

hard link는 같은 파일 시스템 내의 파일을 참조할 수 있다.

link (path1, path2) : 원본 path1의 파일을 link해서 path2 디렉토리에 생성.

unlink (path) : path의 디렉토리에 삭제