

- 파일 이름의 리스트가 주어졌을 때 리스트에 어떤 파일이 변경되었는지 조사한다. 이를 위해 파일의 변경시간 (st_mtime)을 모니터한다

Makefile test1.c test2.c

```
struct stat buf;  
stat(파일 이름, &buf);
```

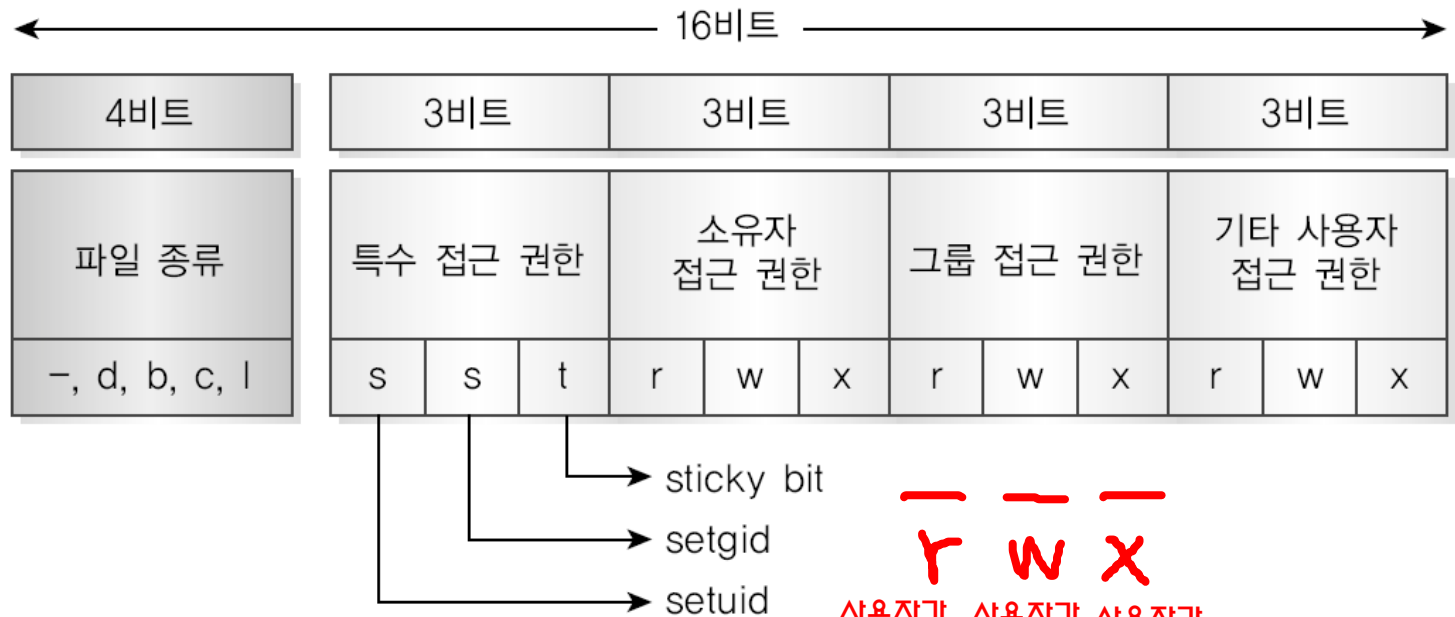
Block 의 개수는 기본 8개 이고,
Block size 는 4096 bytes 이다. 블록 하나 당 크기는 512 bytes
만약, 파일의 크기가 4096 bytes 가 넘어가게 되면 8개 였던 블록의 개수가 두 배인 16개로 늘어남.



파일 접근권한 제어

□ stat 구조체의 st_mode 항목에 파일의 종류와 접근권한 정보저장

□ st_mode 값의 구조



[그림 3-3] st_mode의 비트 구조

— — —
 r w x
 사용자가 읽을 권한 사용자가 쓸 권한 사용자가 실행할 권한

1 0 1
 1 1 0
 1 0 0

- 읽고 쓸수 있음.

- 읽을 수만 있음.

r w _ r _ _ r _ _
 1 1 0 1 0 0 1 0 0
 소유자가 그룹이 기타 사용자
 read, write read read



파일 접근권한 제어

□ 파일의 소유자는 사용자 식별번호로 구별

user identification

- uid, gid 유저가 속한 group identification
- /etc/passwd

□ 유효 사용자 식별번호 (effective user-id, euid)

파일을 만든 사용자

- 파일에 대해 실제 소유권을 갖는 사용자의 사용자 식별번호

□ 진짜 사용자 식별번호 (real user-id, ruid)

프로세스를 실행시킨 사용자

- 실제로 프로세스를 갖는 사용자의 사용자 식별번호

□ 유효 그룹 식별번호, 진짜 그룹 식별번호

-rws-r-xr-x

에서 s 는 사용자 식별번호가 부여되어 있음을 말함.

□ 대부분의 경우, 유효 사용자 식별번호와 진짜 사용자 식별번호는 동일

□ set user-id(04000)

-rw-r-xr-x + 04000 = -rwSr-xr-x

set user-id

원래 x 에 대한 권한이 없으므로, 대문자 S 로 나타남.

- 생성된 프로세스에게 그 프로세스를 시작시킨 사용자의 uid대신 파일 소유자의 유효 사용자 식별번호를 부여 `ls -al /usr/bin/passwd`

□ set group-id(02000)

-rw-r--r-- + 02000 = -rw-r-Sr--

set group-id

`chmod u+s,g-s unix.txt`

□ sticky bit(01000)

drwxrwxrwt

에서 t는 소유자, 슈퍼 권한자가 아니면 지울 수 없음.

- 공유디렉토리(/tmp)에 대한 접근 권한 OR 텍스트-이미지를 swap영역에 남겨둠

모든 사람이 tmp 에 대한 접근권한을 갖는데, 삭제는 불가.

실행 파일,
실행 권한이 없으면
대문자 T 로 표현됨.

옛날에는
그랬음.

파일 종류 검색[1]

□ 상수를 이용한 파일 종류 검색

■ 파일의 종류 검색 관련 상수

chmod : 권한을 바꾸는 명령어

chmod 744 unix.txt

chmod u-x,g+x unix.txt

user 에 x 권한을 뺌.

group 에 x 권한을 더함.

시험에서 제시할 것임.

문제에서 적절하게 사용해야 함.

상수명	상수값(16진수)	기능
<u>S_IFMT</u>	0xF000	<u>st_mode</u> 값에서 파일의 종류를 정의한 부분을 가져옴
<u>S_IFIFO</u>	0x1000	<u>FIFO</u> 파일
<u>S_IFCHR</u>	0x2000	<u>문자 장치 특수 파일</u>
<u>S_IFDIR</u>	0x4000	<u>디렉토리</u>
<u>S_IFBLK</u>	0x6000	<u>블록 장치 특수 파일</u> 물리장치인 경우, 필요함.
S_IFREG	0x8000	일반 파일
S_IFLNK	0xA000	심볼릭 링크 파일
S_IFSOCK	0xC000	소켓 파일

■ st_mode 값과 상수값을 AND(&) 연산하면 파일의 종류 부분만 남게 된다.

Mode = 40755 (디렉토리)

chmod u+s,g+s,o+t testdir

+ set user-id = 04000

+ set group-id = 02000

+ set sticky bit = 01000

=> Mode = 47755 (디렉토리)



[예제 3-3] 상수를 이용해 파일 종류 검색하기 (test3.c)

```
01 #include <sys/types.h>
02 #include <sys/stat.h>
03 #include <stdio.h>
04
05 int main(void) {
06     struct stat buf;
07     int kind;
08
09     stat("unix.txt", &buf);
10     printf("Mode = %o (16진수: %x)\n", (unsigned int)buf.st_mode,
11           (unsigned int)buf.st_mode);
12
13     kind = buf.st_mode & S_IFMT;
14     printf("Kind = %x\n", kind);
15     switch (kind) {
16         case S_IFIFO:
17             printf("unix.txt : FIFO\n");
18             break;
19         case S_IFDIR:
20             printf("unix.txt : Directory\n");
21             break;
22     }
```



[예제 3-3] 상수를 이용해 파일 종류 검색하기

```
23         case S_IFREG:
24             printf("unix.txt : Regular File\n");
25             break;
26     }
27
28     return 0;
29 }
```

```
# ex3_3.out
Mode = 100644 (16진수: 81a4)
Kind = 8000
unix.txt : Regular File
```

