

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

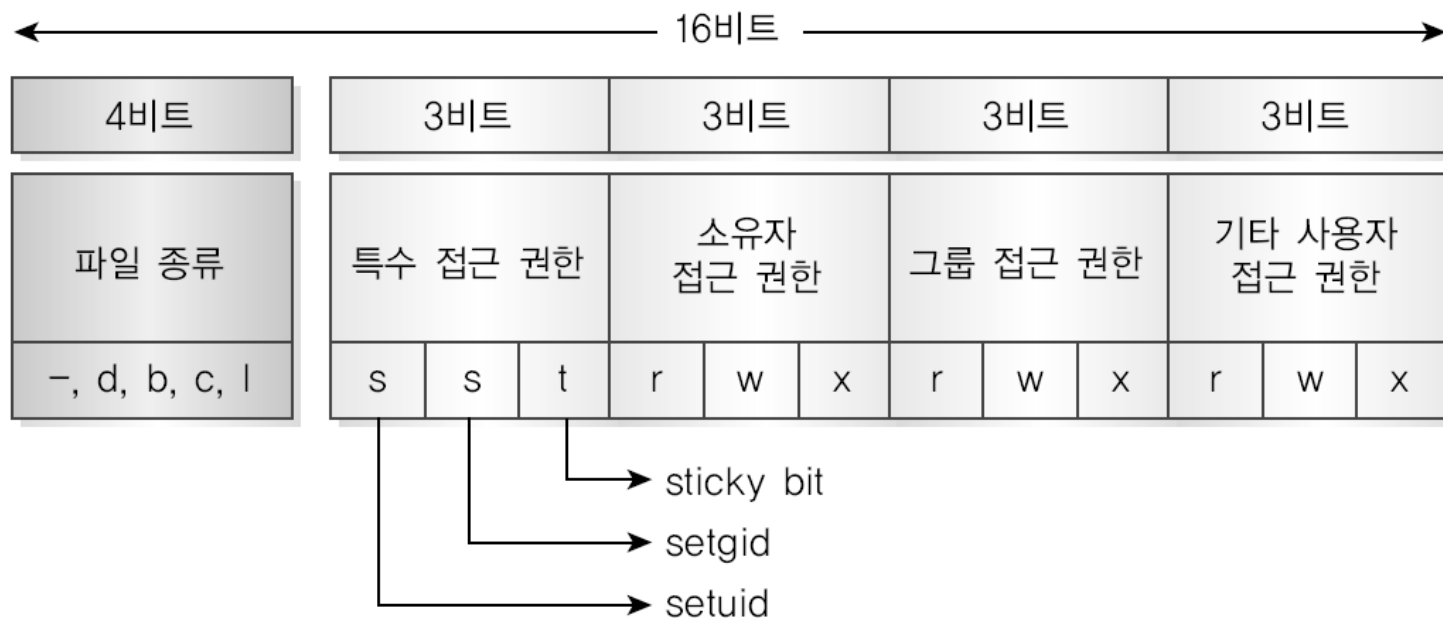
Makefile test1.c test2.c



파일 접근권한 제어

□ stat 구조체의 st_mode 항목에 파일의 종류와 접근권한 정보저장

□ st_mode 값의 구조



[그림 3-3] st_mode의 비트 구조



파일 접근권한 제어

□ 파일의 소유자는 사용자 식별번호로 구별

- uid, gid
- /etc/passwd

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

- 파일에 대해 실제 소유권을 갖는 사용자의 사용자 식별번호

□ 진짜 사용자 식별번호 (real user-id, ruid)

- 실제로 프로세스를 갖는 사용자의 사용자 식별번호

□ 유효 그룹 식별번호, 진짜 그룹 식별번호

□ 대부분의 경우, 유효 사용자 식별번호와 진짜 사용자 식별번호는 동일

□ set user-id(04000)

- 생성된 프로세스에게 그 프로세스를 시작시킨 사용자의 uid대신 파일 소유자의 유효 사용자 식별번호를 부여

□ set group-id(02000)

□ sticky bit(01000)

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



파일 종류 검색[1]

□ 상수를 이용한 파일 종류 검색

■ 파일의 종류 검색 관련 상수

상수명	상수값(16진수)	기능
S_IFMT	0xF000	st_mode 값에서 파일의 종류를 정의한 부분을 가져옴
S_IFIFO	0x1000	FIFO 파일
S_IFCHR	0x2000	문자 장치 특수 파일
S_IFDIR	0x4000	디렉토리
S_IFBLK	0x6000	블록 장치 특수 파일
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.



[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
11     printf("Mode = %o (16진수: %x)\n", (unsigned int)buf.st_mode,
(unsigned int)buf.st_mode);
12
13     kind = buf.st_mode & S_IFMT;
14     printf("Kind = %x\n", kind);
15
16     switch (kind) {
17         case S_IFIFO:
18             printf("unix.txt : FIFO\n");
19             break;
20         case S_IFDIR:
21             printf("unix.txt : Directory\n");
22             break;
```



[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
```

