

운영체제 보안

- 그룹 관련 APIs와 소유권

컴퓨터소프트웨어학과

김병국 교수



- 그룹에 대한 정보를 추출하는 프로그램을 만들 수 있다.
- 파일의 소유권에 대해 이해하고 이를 변경할 수 있다.
- 소유권을 변경할 수 있는 프로그램을 만들 수 있다.



목차

□ 그룹 정보 추출

□ 소유권

□ 소유권 변경

□ 소유권 변경 API

□ 관련응용 실습



1. 그룹 정보 추출 (1/4)

□ 그룹 정보 추출

- 함수 : getgrgid() & getgrnam()
 - 주어진 그룹 ID에 해당하는 그룹 계정 정보를 추출
 - 추출값은 group 구조체형 포인터를 가짐
 - re-entrance 문제 주의
 - 해결: getgrgid_r() 또는 getgrnam_r() 함수를 사용
- 인자:
 - getgrgid()의 gid : 그룹 ID
 - getgrnam()의 *name : 그룹명
- 결과 값:
 - 성공 : group 구조체형 포인터 변수
 - 실패: null

```
struct group
{
    char *gr_name;    /* group name */
    char *gr_passwd;  /* group password */
    gid_t gr_gid;     /* group ID */
    char **gr_mem;    /* NULL-terminated array of pointers
                       to names of group members */
};
```

[group 구조체]

```
#include <sys/types.h>
#include <grp.h>

struct group *getgrnam(const char *name);
struct group *getgrgid(gid_t gid);
```

[getgrgid() 함수의 프로토타입]

1. 그룹 정보 추출 (2/4)

□ 실습 1

■ 그룹에 대한 멤버 리스트 추출

```
1  #include <stdio.h>
2  #include <sys/types.h>
3  #include <unistd.h>
4  #include <grp.h>
5
6  int
7  main(int argc, char* argv[])
8  {
9      struct group* stGroup;
10
11     if (argc != 2) {
12         printf("Usage: %s [group name]\n", argv[0]);
13         return -1;
14     }
15
16     stGroup = getgrnam(argv[1]);
17     if (stGroup == NULL) {
18         printf("No Group!!\n");
19         return -1;
20     }
21
```

```
22     printf("Group Name : %s\n", stGroup->gr_name);
23     printf("Members are : \n");
24     for (int i = 0; ; i++)
25     {
26         if (stGroup->gr_mem[i] == NULL)
27             break;
28
29         printf("\t[%02d] : %s\n", i, stGroup->gr_mem[i]);
30     }
31
32     printf("Done.\n");
33
34     return 0;
35 }
```

파일명: get_grnam.c



1. 그룹 정보 추출 (3/4)

□ 사용자에게 대한 그룹정보 추출

■ 함수 : getgrouplist()

- 주어진 사용자와 GID를 통한 소속된 모든 그룹들을 추출
- 소속된 그룹의 최대 개수는 ngroups를 통해 전달하나, 실행과정에서 이 보다 더 많은 그룹에 속할 경우 -1을 반환
- 인자:
 - *user : 사용자명
 - group : 사용자의 GID
 - *groups : 소속된 모든 그룹 저장용
 - *ngroups : 소속된 그룹의 개수 알림 또는 저장용
- 결과 값:
 - 성공 : 그룹의 개수
 - 실패: -1

```
#include <grp.h>

int getgrouplist(const char *user, gid_t group,
                 gid_t *groups, int *ngroups);
```

[getgrouplist() 함수의 프로토타입]



1. 그룹 정보 추출 (4/4)

□ 실습 2

■ 사용자의 그룹리스트 추출

```
7 int
8 main(int argc, char* argv[])
9 {
10     gid_t nGIds[BUFSIZ];
11     int nCount = BUFSIZ;
12     int nRet = 0;
13
14     struct passwd* stPasswd;
15     struct group* stGroup;
16
17     if (argc < 2) {
18         stPasswd = getpwuid(getuid());
19     }
20     else {
21         stPasswd = getpwnam(argv[1]);
22     }
23
24     if (stPasswd == NULL) {
25         printf("No Accounts!!\n");
26         return -1;
27     }
```

2

```
29 nRet = getgrouplist(stPasswd->pw_name, stPasswd->pw_gid, nGIds, &nCount);
30 if (nRet < 0)
31 {
32     printf("Getting group list was failed.\n");
33     return -1;
34 }
35
36 printf("Groups of %s is : \n", stPasswd->pw_name);
37 for (int i = 0; i < nCount; i++)
38 {
39     stGroup = getgrgid(nGIds[i]);
40     printf("\t[%02d] : %s(%d)\n", i, stGroup->gr_name, nGIds[i]);
41 }
42
43 printf("Done.\n");
44
45 return 0;
46 }
```

3

```
1 #include <stdio.h>
2 #include <sys/types.h>
3 #include <unistd.h>
4 #include <pwd.h>
5 #include <grp.h>
6
```

1

파일명: get_grouplist.c



2. 소유권 (1/2)

□ 파일 및 디렉토리 보기

■ 파일 속성

```
kali@kali:~$ ls -l
total 72
drwxr-xr-x  2 kali kali  4096 Feb 23 19:37 Desktop
drwxr-xr-x  2 kali kali  4096 Feb 23 19:37 Documents
drwxr-xr-x  2 kali kali  4096 Feb 23 19:37 Downloads
-rwxr-xr-x  1 kali kali 16608 Apr  6 15:13 hello
-rw-r--r--  1 kali kali   75 Apr 29 17:08 hello.c
```

1

2

3

4

번호	값	동작
1	d	파일 종류 (- : 일반파일, d: 디렉토리)
2	rwx-r--r-x	파일을 읽고, 쓰고, 실행할 수 있는 접근 권한 표시
3	kali	파일 소유자의 사용자 명
4	kali	파일 소유자의 그룹명

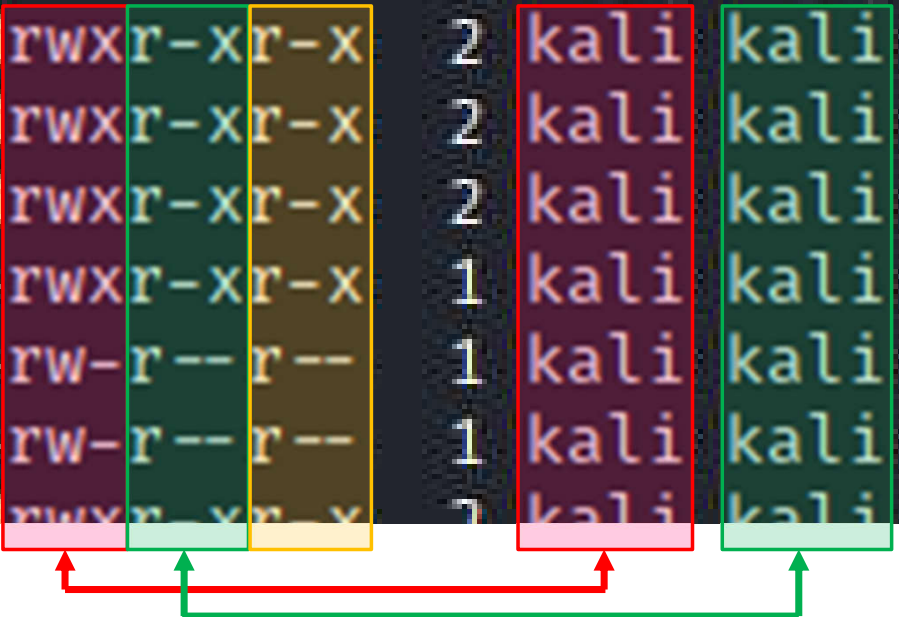


2. 소유권 (2/2)

□ 파일 및 디렉토리 보기

- 접근 권한과 소유권

```
kali@kali:~$ ls -l
total 72
drwxr-xr-x  2 kali kali 4096 Feb 10 10:10 kali
drwxr-xr-x  2 kali kali 4096 Feb 10 10:10 kali
drwxr-xr-x  2 kali kali 4096 Feb 10 10:10 kali
-rwxr-xr-x  1 kali kali 16608 Feb 10 10:10 kali
-rw-r--r--  1 kali kali  75 Feb 10 10:10 kali
-rw-r--r--  1 kali kali 2824 Feb 10 10:10 kali
drwxr-xr-x  2 kali kali 4096 Feb 10 10:10 kali
```



3. 소유권 변경 (1/2)

□ 소유권자 변경

- 명령어: `chown {옵션} [소유권자] [대상 ...]`
 - 대상(파일, 디렉토리)에 대한 소유권자(ownership)를 변경
 - 소유권자에 대한 값은 이름 또는 숫자로 표현이 가능
 - 소유권자 지정 패턴
 - `[사용자]` : 사용자를 변경
 - `[사용자:그룹]` 또는 `[사용자.그룹]` : 사용자와 그룹을 변경
 - `[:그룹]` 또는 `[.그룹]` : 그룹을 변경
 - 대표적 옵션:
 - `-R, --recursive` : 디렉토리일경우 하단의 모든 파일이나 서브디렉토리들에 대하여 일괄 변경



3. 소유권 변경 (2/2)

□ 소유권 그룹 변경

- 명령어: `chgrp {옵션} [소유권 그룹] [대상 ...]`
 - 대상(파일, 디렉토리)에 대한 소유권 그룹을 변경
 - 소유권자에 대한 값은 이름 또는 숫자로 표현이 가능
- 대표적 옵션:
 - `-R, --recursive` : 디렉토리일경우 하단의 모든 파일이나 서브디렉토리들에 대하여 일괄 변경



4. 접근 권한과 소유권 변경 실습

□ 실습 내용

- 파일의 소유권 변경
- 접근 권한 변경
- 프로그램 실행



5. 소유권 변경 API (1/2)

□ 소유권 변경

■ 함수 : chown() & fchown() & lchown()

- 지정한 파일에 대하여 소유권을 변경하는 함수
- fchown() 함수는 파일기술자를 통해 처리
- lchown() 함수는 링크파일(심볼릭)의 경우 역참조(de-reference)를 수행하지 않음

• 인자: (생략)

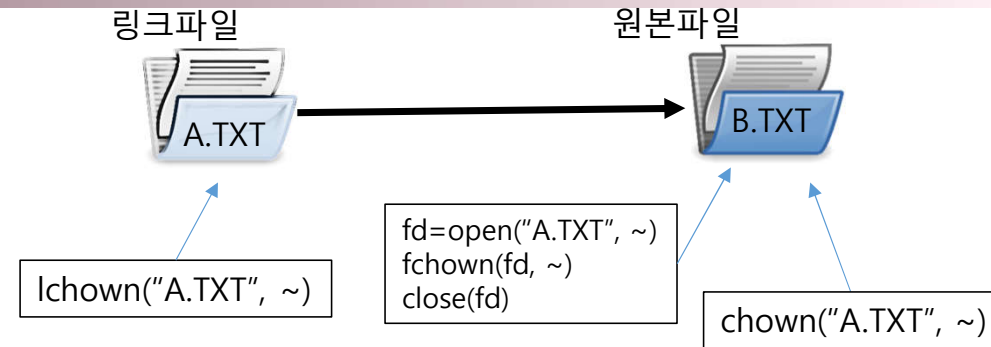
• 결과값:

- 성공 : 0
- 실패 : -1

```
#include <unistd.h>

int chown(const char *pathname, uid_t owner, gid_t group);
int fchown(int fd, uid_t owner, gid_t group);
int lchown(const char *pathname, uid_t owner, gid_t group);
```

【소유권 변경 함수들의 프로토타입】



5. 소유권 변경 API (2/2)

□ 실습

- 파일 소유권 변경
- 실행 1st: 관리자권한 획득
- 실행 2nd: ./chown <사용자명> <그룹명> <대상파일>

```
1 #include <stdio.h>
2 #include <sys/types.h>
3 #include <unistd.h>
4 #include <pwd.h>
5 #include <grp.h>
6
7 int main(int argc, char *argv[])
8 {
9     struct passwd *stPasswd;
10    struct group *stGroup;
11
12    if (argc != 4)
13    {
14        printf("Usage: %s <owner> <group> <filename>\n", argv[0]);
15        return -1;
16    }
17
18    stPasswd = getpwnam(argv[1]);
19    stGroup = getgrnam(argv[2]);
```

```
21    if (stPasswd == NULL || stGroup == NULL)
22    {
23        printf("Neither User nor Group exists in this system!!\n");
24        return -1;
25    }
26
27    printf("UID: %d, GID: %d\n", stPasswd->pw_uid, stGroup->gr_gid);
28
29    lchown(argv[3], stPasswd->pw_uid, stGroup->gr_gid);
30
31    return 0;
32 }
```

파일명: chown.c



수고하셨습니다.

