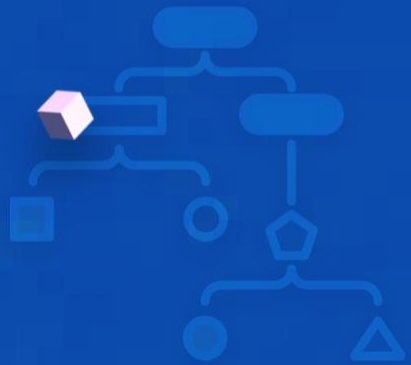
강원혁신플랫폼

리눅스프로그래밍

파일 정보와 권한



chmod 명령을 사용한 파일의 접근 권한에서,
접근 권한 설정 구분이 아닌 것은 무엇인가요?

1 user　　2 group　　3 other　　4 everyone

4번

학습목표

학습 내용

1. 파일 소유자와 그룹 변경
2. chmod()
3. 시간 처리

학습 목표

- 파일 소유자와 그룹 변경에 대해 설명할 수 있다.
- chmod() 함수에 대해 파악할 수 있다.
- 시간 처리 함수에 대해 파악할 수 있다.

파일 소유자와 그룹 변경

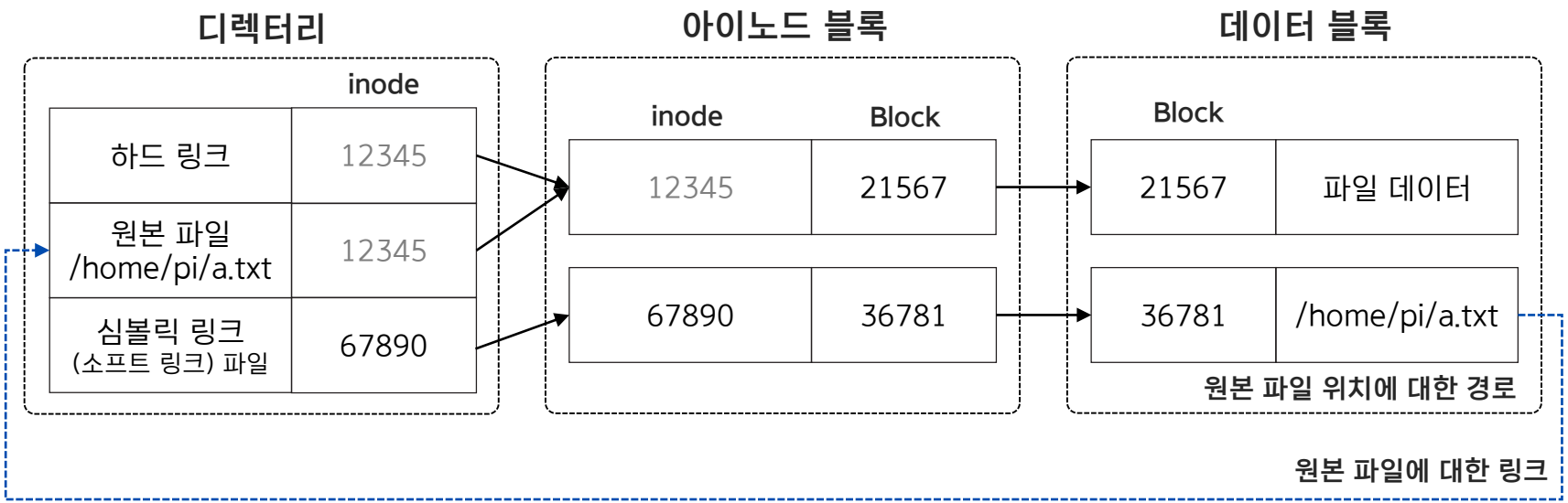
하드 링크와 심볼릭 링크


```
$ touch 1.txt
$ ln 1.txt 2.txt
$ ln -s 1.txt 3.txt
$ ls -il [1-3].txt
29836 -rwxr-xr-x 2 freetime freetime 4 Jul  8 14:53 1.txt
29836 -rwxr-xr-x 2 freetime freetime 4 Jul  8 14:53 2.txt
29837 lrwxrwxrwx 1 freetime freetime 5 Jul  8 14:53 3.txt -> 1.txt
$ echo "123" >> 2.txt
$ more 1.txt
123
123
```

하드 링크와 심볼릭 링크


```
$ ls -il [1-3].txt
29836 -rwxr-xr-x 2 freetime freetime 8 Jul  8 14:54 1.txt
29836 -rwxr-xr-x 2 freetime freetime 8 Jul  8 14:54 2.txt
29837 lrwxrwxrwx 1 freetime freetime 5 Jul  8 14:53 3.txt -> 1.txt
$
```

하드 링크와 심볼릭 링크

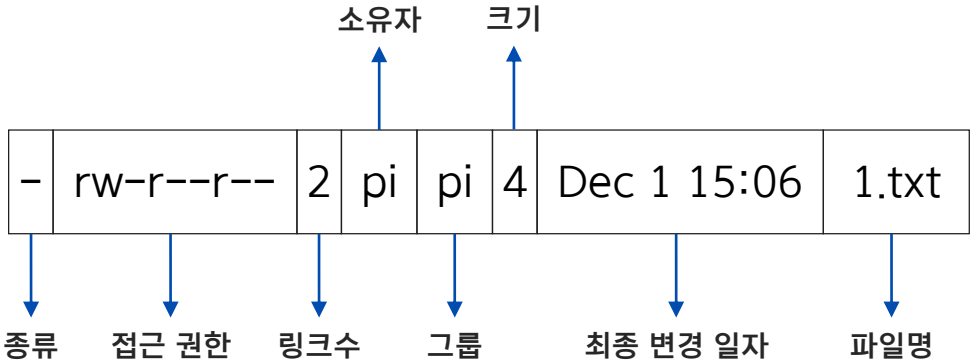


출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

파일 정보와 권한


```
$ ls -l ?.txt
-rwxr-xr-x 2 freetime freetime 8 Jul  8 14:54 1.txt
-rwxr-xr-x 2 freetime freetime 8 Jul  8 14:54 2.txt
lrwxrwxrwx 1 freetime freetime 5 Jul  8 14:53 3.txt -> 1.txt
```

파일 정보와 권한



출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

ls -l 옵션의 출력 구분

구분	내용	비고
1	파일, 디렉터리, 블록 디바이스 등과 같은 파일의 종류를 나타냄	1글자
2	문자는 파일의 접근 권한을 의미함 접근 권한은 이 파일을 어떤 식으로 보호할지에 대해 나타내는 것 앞에서부터 3개씩 rwx(읽기, 쓰기, 실행하기)로 구분해 볼 수 있음 소유자, 그룹, 다른 사람의 순으로 각각 나타내 볼 수 있음	9글자
3	링크의 수로 이 파일을 몇 번 링크되어 있는지 명시하는 것	
4	파일의 소유자로 이 파일을 접근할 수 있는 파일의 소유자를 의미함	
5	파일의 그룹으로 이 파일을 접근할 수 있는 파일의 그룹을 의미함	

출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

ls -l 옵션의 출력 구분

구분	내용	비고
6	파일의 크기로 바이트 값임	
7	파일의 최종 변경 일자를 의미함	
8	파일의 이름을 의미함	

출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

파일 소유자와 그룹 변경

chgrp 변경한_그룹명 파일명

```
$ users
freetime
$ sudo groups
[sudo] password for freetime:
root
$ groups
freetime adm dialout cdrom floppy sudo audio dip video plugdev netdev
$
$ sudo chown root 2.txt
$ sudo chgrp users 2.txt
```

파일 소유자와 그룹 변경


```
$ groups
freetime adm dialout cdrom floppy sudo audio dip video plugdev netdev
$ ls -al 2.txt
-rwxr-xr-x 2 root users 8 Jul  8 14:54 2.txt
```

문자열을 이용한 권한 설정

순서	의미
1	u(user/owner) : 소유자, g(group) : 그룹, o(others) : 다른 사람, a(all) : 모든 사람
2	+ : 권한 추가, - : 권한 삭제
3	r(read) : 읽기, w(write) : 쓰기, x(execute) : 실행, s(set-user-ID/set-group-ID)

출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

문자열을 이용한 권한 설정


```
$ ls -l 2.txt
-rw-r--r-- 2 root users 8 Jul  8 14:54 2.txt
$ sudo chmod ug+x 2.txt
$ ls -l 2.txt
-rwxr-xr-- 2 root users 8 Jul  8 14:54 2.txt
```

8진수를 이용한 권한 설정

구분	읽기	쓰기	실행
2진수	100	010	001
8진수	4	2	1

출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

8진수를 이용한 권한 설정


```
$ ls -l 2.txt
-rwxr-xr-- 2 root users 8 Jul  8 14:54 2.txt
$ sudo chmod 777 2.txt
$ ls -l 2.txt
-rwxrwxrwx 2 root users 8 Jul  8 14:54 2.txt
$ sudo chmod 660 2.txt
$ ls -l 2.txt
-rw-rw---- 2 root users 8 Jul  8 14:54 2.txt
$
```

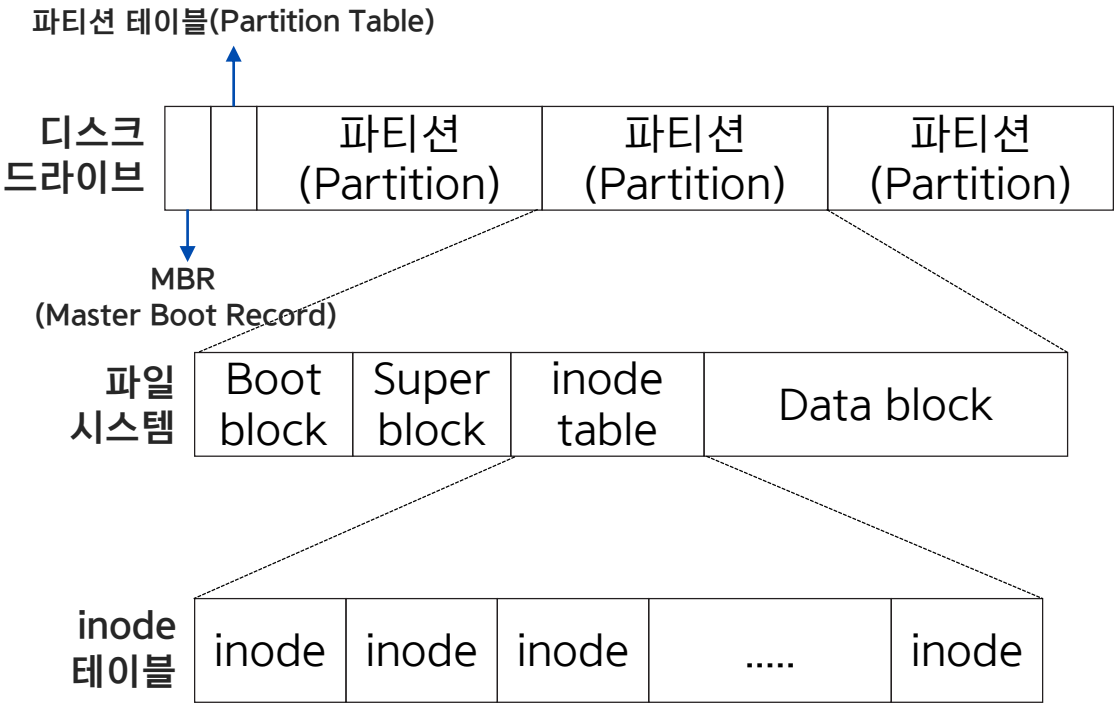
파일 생성시의 접근 권한


```
$ umask
0022
$ touch 4.txt
$ ls -l 4.txt
-rw-r—r- 1 freetime freetime 0 Jul  8 15:10 4.txt
$ umask 420
$ touch 5.txt
$ ls -l 5.txt
--w-r--rw- 1 freetime freetime 0 Jul  8 15:13 5.txt
$
```

파일 정보 출력


```
$ stat 1.txt
  File: 1.txt
  Size: 8               Blocks: 8          IO Block: 4096   regular file
Device: 820h/2080d      Inode: 29836       Links: 2
Access: (0660/-rw-rw----)  Uid: (    0/    root)   Gid: (  100/   users)
Access: 2023-07-08 14:54:11.222301081 +0900
Modify: 2023-07-08 14:54:07.040635187 +0900
Change: 2023-07-08 15:09:43.961300468 +0900
 Birth: 2023-07-08 11:26:04.993146705 +0900
```

디스크 파티션과 파일 시스템의 구조



출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

디스크 파티션과 파일 시스템의 구조

01 02 03 04

부트 블록(Boot Block)

- 파일 시스템의 첫 번째 블록으로 부팅을 위한 부트로더(Boot Loader) 등의 파일 시스템으로부터 유닉스 커널을 로딩하기 위한 부트스트랩 코드(시스템 부팅을 위한 프로그램)를 저장함

출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

디스크 파티션과 파일 시스템의 구조

슈퍼 블록(Super Block)

- 파일 시스템의 크기, 저장공간의 상태, 자유공간 정보 등의 파일 시스템의 상태를 기술함
- 전체 파일 시스템에 대한 정보를 저장함
- 사용 가능한 아이노드의 수와 디스크(disk) 블록의 수를 저장함

출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

디스크 파티션과 파일 시스템의 구조

슈퍼 블록(Super Block)

- 파일 시스템 크기
- 파일 시스템 내의 자유 블록의 수
- 파일 시스템 내에서 사용 가능한 자유 블록의 리스트
- 아이노드 리스트의 크기
- 파일 시스템 내의 사용 가능한 아이노드의 수
- 파일 시스템 내의 사용 가능한 아이노드의 리스트
- 잠금(lock) 정보

출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

디스크 파티션과 파일 시스템의 구조

01 02 03 04

아이노드 테이블(아이노드 List)

- 각 파일이나 디렉터리에 대한 정보를 가지고 있는 블록으로 아이노드의 자료 구조 정보가 저장됨

출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

디스크 파티션과 파일 시스템의 구조

01 02 03 04

데이터 블록(Data Block)

- 파일이나 디렉터리에 대한 실제의 데이터가 저장되어 있는 블록

출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

따라하기 chmod.c


```
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>

int main(int argc, char **argv)
{
    struct stat statbuf;

    if (argc < 3) {
        fprintf(stderr, "Usage: %s file1 file2\n", argv[0]);
        return -1;
    }
```

따라하기 chmod.c


```
/* 파일에 존재 여부에 대한 정보를 가져온다. */
    if (stat(argv[1], &statbuf) < 0) {
        perror("stat");
        return -1;
    }

 /* set-group-ID를 설정하고 그룹의 실행 권한을 해제한다. */
    if (chmod(argv[1], (statbuf.st_mode & ~S_IXGRP) | S_ISGID) < 0) {
        perror("chmod");
        return -1;
    }
```

따라하기 chmod.c


```
/* 파일의 권한을 644("rw-r--r--") 로 설정한다. */
    if (chmod(argv[2], S_IRUSR | S_IWUSR | S_IRGRP | S_IROTH) < 0) {
        perror("chmod");
        return -1;
    }

    return 0;
}
```

따라하기 chmod.c

실행 확인

기본 설정

```
$ umask 0022
$ gcc -o chmod chmod.c

$ chmod g+x 4.txt
$ ls -l 4.txt 5.txt
-rw-r-xr- 1 freetime freetime 0 Jul  8 15:10 4.txt
--w-r—rw- 1 freetime freetime 0 Jul  8 15:13 5.txt
```

실행 확인

```
$ ./chmod 4.txt 5.txt
/* set-group-ID를 설정하고 그룹의 실행 권한을 해제한다. */
$ ls -l 4.txt 5.txt
-rw-r-Sr-- 1 freetime freetime 0 Jul  8 15:10 4.txt
-rw-r--r-- 1 freetime freetime 0 Jul  8 15:13 5.txt
$
```

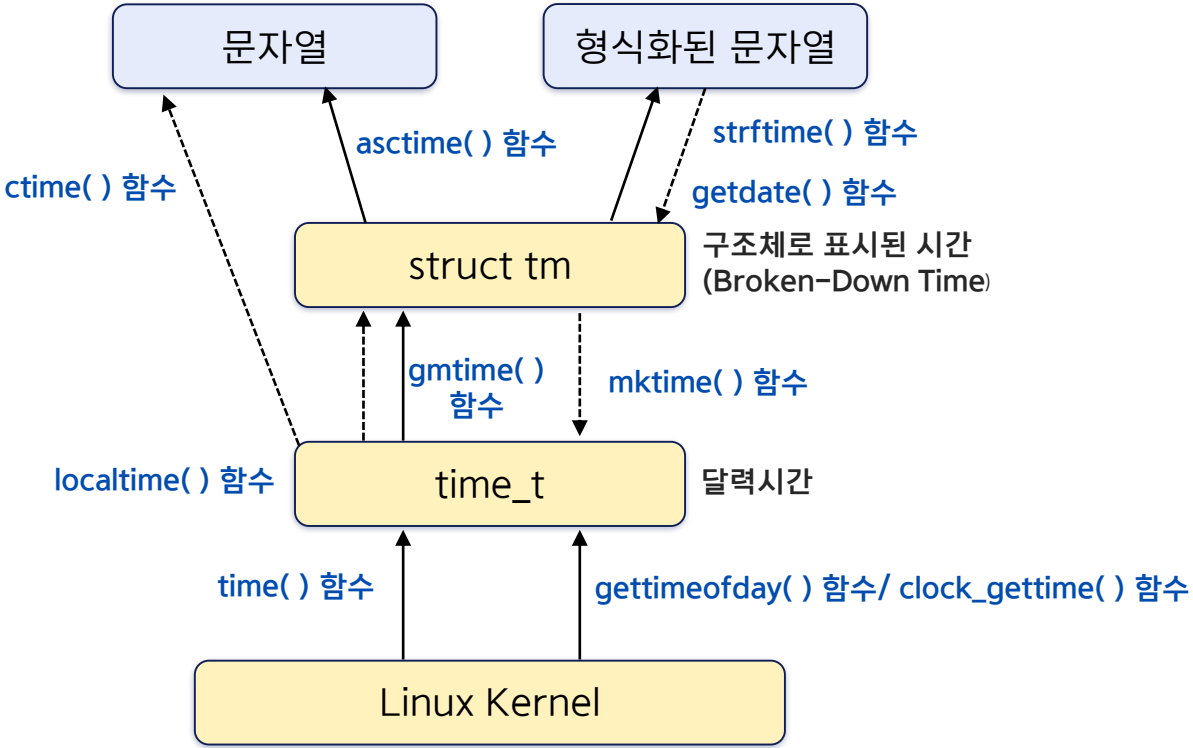
실행 확인

S_ISUID 04000 실행하는 사람이 권한을 가진다.
S_ISGID 02000 실행하는 그룹이 권한을 가진다.
S_ISVTX 01000 스티키 비트 설정

```
$ chmod 7777 1.txt
$ ls -l 1.txt
-rwsrwsrwt 1 freetime freetime 0 Jul 18 21:37 x1.txt
$ chmod 4444 2.txt; ls -l 2.txt
-r-Sr--r-- 1 freetime freetime 0 Jul 18 21:38 2.txt
$ chmod 2444 3.txt; ls -l 3.txt
-r--r-Sr-- 1 freetime freetime 0 Jul 18 21:38 3.txt
$ chmod 1444 4.txt; ls -l 4.txt
-r--r--r-T 1 freetime freetime 0 Jul 18 21:38 4.txt
```

시간 처리

유닉스의 시간 관련 함수들



출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

strftime( ) 함수의 서식 지정자

형식	내용	형식	내용
%A	완전한 요일 이름	%a	간략한 요일 이름
%B	완전한 월 이름	%b	간략한 월 이름
%H	24시간으로 표시한 시간(00 ~ 23)	%I	12시간으로 표시한 시간(01 ~ 12)
%j	년 중의 지나간 합산 일(001 ~ 365)	%d	월 중의 일(01 ~ 31)
%m	월(01 ~ 12)	%p	AM / PM
%M	분(00 ~ 59)	%S	초(00 ~ 61/윤초)
%X	시간(12:21:13)	%x	날짜(02/13/74)
%Y	세기 있는 년도(1974)	%y	세기 없는 년도(74)

출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

strftime( ) 함수의 서식 지정자

형식	내용	형식	내용
%W	월요일 주 수(00 ~ 53)	%w	요일(0=일요일 ~ 6)
%U	일요일 주 수(00 ~ 53)	%Z	시간대 이름(KST)
%c	날짜와 시간(Wed Feb 13 12:21:13 1974)		

출처 서영진, 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리 파이, 제이펍, 2020

따라하기 time.c


```
#include <stdio.h>
#include <time.h>
#include <sys/time.h>                /* getimeofday( ) 함수에서 사용 */
#include <stdlib.h>

int main(int argc, char **argv)
{
    int i, j;
    time_t rawtime;
    struct tm *tm;
    char buf[BUFSIZ];
    struct timeval mytime;
```

따라하기 time.c


```
time(&rawtime);                    /* 현재의 시간 구하기 */
printf("time : %u\n", (unsigned)rawtime);   /* 현재의 시간을 화면에 출력 */

gettimeofday(&mytime, NULL);       /* 현재의 시간 구하기 */
printf("gettimeofday : %ld/%ld\n", mytime.tv_sec, mytime.tv_usec);
printf("ctime : %s", ctime(&rawtime));      /* 현재의 시간을 문자열로 바꿔서 출력 */

putenv("TZ=PST3PDT");              /* 환경 변수를 설정한다. */
tzset();                           /* TZ 변수를 설정한다. */
tm = localtime(&rawtime);          /* tm = gmtime(&rawtime); */
printf("asctime : %s", asctime(tm));       /* 현재의 시간을 tm 구조체를 이용해서 출력 */
```

```
    /* 사용자 정의 문자열 지정 */
    strftime(buf, sizeof(buf), "%a %b %e %H:%M:%S %Y", tm);
     printf("strftime : %s\n", buf);

     return 0;
}
```

따라하기 time.c

실행 확인

```
$ gcc -o time time.c
$ ./time
time : 1688798672
gettimeofday : 1688798672/743017
ctime : Sat Jul  8 15:44:32 2023
asctime : Sat Jul  8 04:44:32 2023
strftime : Sat Jul  8 04:44:32 2023
```

요약정리

01 • 파일 소유자와 그룹 변경

02 • chmod()

03 • 시간 처리