

# TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 – Innova Lee( 이상훈 )

gcccompil3r@gmail.com

학생 – 윤연성

[whatmatters@naver.com](mailto:whatmatters@naver.com)

- 1.ls\_module
- 2.퍼미션 rwx 변환
- 3.sticky bit

Touch a.txt b.txt 만들면 생성됨

```
./a.out -a  
./a.out -b a  
./a.out -b  
./a.out -abc
```

-a 옵션  
옵션분할

ls -a = 모든파일출력 (숨김파일 포함 )  
ls -l = 일반파일을 자세하게 보여줌  
ls -al 은 숨김파일 포함 모든파일을 자세하게 보여줌  
ls /home/aa\* = 파일명이 aa 로 시작하는 모든 파일목록을 출력  
ls /home/\*.txt = .txt 끝나는 모든파일 목록을 출력

ls -R 디렉토리를 순회하면서 보여줌  
ls -S 파일의 사이즈를 기준으로 정렬  
ls -t = 수정한 날짜를 기준으로 정렬

drwxrwxr-x ( 디렉터리)  
-rwxrwxr-x (일반파일)  
- = 일반파일  
d =디렉터리  
l =심벌링 링크  
b =블록장치파일  
c =문자장치파일  
p =파이프파일  
s =소켓파일

man ls

////////// ls module

```
-----
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>

int main(int argc, char **argv)
{
    int cmd;
    while((cmd = getopt(argc, argv, "ab")) > 0)
    {
        switch(cmd)
        {
            case 'a':
                printf("a option\n");
                break;
            case 'b':
                printf("b option\n");
                break;
            default:
                printf("unknown option\n");
        }
    }
    return 0;
}
```

```
-----
./a.out -a
a option
./a.out -b
b option
./a.out -ab
a option
b option
-----
```

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

```

#include <unistd.h>
#include <stdio.h>

int main(int argc, char **argv)
{
    struct stat buf; //stat 스탯 = 정보를 본다
    char ch;
    stat(argv[1], &buf); // argv[1]정보를 보고 buf 에 저장
    //argv 는 파일명이나 디렉토리명이 와야한다

    if(S_ISDIR(buf.st_mode))           //디렉토리냐? buf.st_mode 에 들어있음
        ch = 'd';
    if(S_ISREG(buf.st_mode))           //레귤러=일반파일   ./a.out ls_module5.c
        ch = '-';                      치면 일반파일이라 - 라고 나옴
    if(S_ISFIFO(buf.st_mode))          //파이프
        ch = 'p';
    if(S_ISLNK(buf.st_mode))           //심벌릭링크 = 바로가기
        ch = 'l';
    if(S_ISSOCK(buf.st_mode))          //소켓(네트워크 관련장비)
        ch = 'S';
    if(S_ISCHR(buf.st_mode))           //캐릭터디바이스
        ch = 'C';
    if(S_ISBLK(buf.st_mode))           //블럭디바이스
        ch = 'b';
    printf("%c\n",ch);
    return 0;
}

```

---

```

#include <sys/types.h>
#include <dirent.h>
#include <unistd.h>
#include <stdio.h>
#include <fcntl.h>

int main(int argc, char**argv)
{
    DIR *dp;
    int i = 0, cmd;
    struct dirent *p;
    cmd = getopt(argc, argv, "a");
}

```

```

dp = opendir(".");
while(p = readdir(dp))
{
    if(cmd != 'a')
    {
        if(p->d_name[0] == '.')
            continue;
    }
    printf("%-16s ",p->d_name);
    if((i + 1)%5 ==0)
        printf("\n");
    i++;
}
printf("\n");
closedir(dp);
return 0;
}

```

-----

./a.out

2.c	a.out	a.txt	1.c	c.txt
3.c	b.txt			

---

```

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

```

```

int main(int argc, char **argv)
{
    struct stat buf;                //   파일의종류
    char ch;                        //   나   rwx   rwx   rwx
    char perm[11] = "-----";      //   사용자   그룹권한   제 3 자
    char rwx[4] = "

                                //perm 9 개를뽑아내기 위해
                                //rwx 파일권한설정할때 8 진수

    int i;
    stat(argv[1], &buf);

    if(S_ISDIR(buf.st_mode));

```

```

perm[0] = 'd';
if(S_ISREG(buf.st_mode));
perm[0] = '-';
if(S_ISFIFO(buf.st_mode));
perm[0] = 'p';
if(S_ISLNK(buf.st_mode));
perm[0] = 'l';
if(S_ISSOCK(buf.st_mode));
perm[0] = 's';
if(S_ISCHR(buf.st_mode));
perm[0] = 'c';
if(S_ISBLK(buf.st_mode));
perm[0] = 'b';

for(i = 0; i < 9; i++)                //i 가 0 부터 9 까지
    if(buf.st_mode >> (8 - i) & 1)    //
        perm[i + 1] = rwx[i % 3];
printf("%s\n",perm);
return 0;
}

```

-----rwx 퍼미션 권한 조종-----

**rwx : read, write, execute (실행권한)**

**421**

**rw** - root 사용자 권한

**r--** 그룹권한

**r--** 제 3 자에 대한 권한

**r** = 읽기 : 파일의 읽기 권한

**w** = 쓰기 : 파일의 쓰기권한

**x** = 실행 : 파일의 실행권한

**ex)**

**만약 rwxr-xr-x 면**

**소요자 : rwx**

그룹 : r-x  
공개 (3자): = r-x  
- 는 없다는 뜻

그리고 r = 4, w = 2, x = 1 퍼미션 기호 숫자

ex

4 + 2 + 1 | 4 + 0 + 1 | 4 + 0 + 1

7                5                5 합산하면 755

퍼미션변경

ls -al 로 확인 a.txt.변경

chmod 777 a.txt                                -----                퍼미션 변경 실행  
하면

ls -al 로 보면    rwxr-xr-x 에서  
rwxrwxrwx 가 된거를 확인할수있음

소유자 및 소유그룹변경

chown b.txt a.txt 하고

ls -al 하면

b.txt 로 바뀐거 확인할수있음

사용자는 읽고쓰기 가능하고 그룹과 제 3자는 읽기만 가능

파일종류가 처음으로 온다.

파일 권한설정하는데에 8 진사용 8 진수 사용할땐 첫번째에 0 을 넣음

getpwuid() - > 개인

getgrgid() - > 그룹

setuid, setgid, sticky bit

sticky 비트를 어떤경우 사용하는지 알아보자

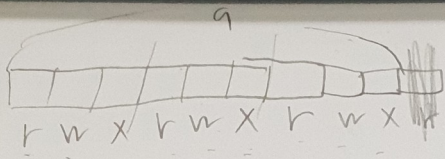
1 공유디렉토리에 붙게된다

2 추가적인 사항인데 swap 으로 사용하고자 할경우에도 붙인데 (즉 파일 )

변환하는 과정을 페이징이라함

32 비트냐 64 비트냐에 따라 다름

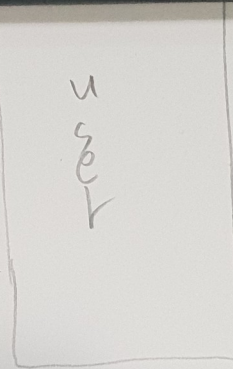




Swap이 실행  
LRM이 실행  
메모리

sticky bit가 실행

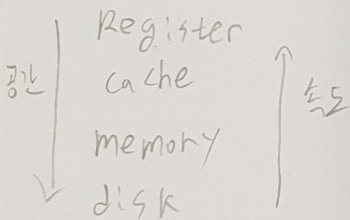
Swap = 디스크



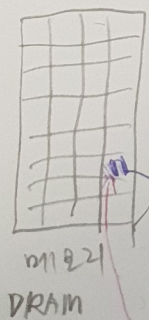
0x10000000  
10bit 10bit 10bit  
[1011 1111 1000 1000 0]

0x0

Vm(가상메모리)



out 실행  
sticky bit  
프로세스



4KB 단위

sticky bit가  
블록에 작용

if (특정 메모리 공간)

① 메모리 부족하면 데이터를 디스크로 옮기는 방법

메모리 부족하면

swap이 도움

디스크에 있는 영역

disk에 swap

가장 사용하지 않는 부분을 먼저  
메모리 양을 줄임

없는 queue

② 파일과 디스크



task\_struct

files\_struct

file

path

inode

super\_block

mm\_struct

vm\_area\_struct