

# TI DSP, MCU, Xilinx Zynq FPGA Based Programming Expert Program

**Instructor – Innova Lee(Sanghoon Lee)**

[gcccompil3r@gmail.com](mailto:gcccompil3r@gmail.com)

**Student – Hyungju Kim**

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



# System programming

## Implement "ls" instruction

```
howard@ubuntu: ~/HomeworkBackup/21th
1 #include <stdio.h>
2 #include <dirent.h>
3 #include <fcntl.h>
4
5 int main(void){
6     DIR* dp;
7     struct dirent* p;
8     int i = 1;
9     dp = opendir(".");
10    while(p = readdir(dp)){
11        if(p->d_name[0] == '.')
12            continue;
13
14        printf("%-16s ",p->d_name);
15        if(i%5 == 0)
16            printf("\n");
17        i++;
18    }
19    printf("\n");
20
21    return 0;
22 }
```

The dirent structure ,in dirent.h header file, is used for access to directory file. So, If you want to implement ls instructor, which works showing every files in the current directory.

## Adding some options

Think about an instructor with some options, f.e , ls -al. when you add an option, you'd write the option's parameter in the backside of instructor. So, argv[] would be used to implement it.

```
howard@ubuntu: ~/HomeworkBackup/21th
1 #include <stdio.h>
2 #include <fcntl.h>
3 #include <unistd.h>
4
5 int main(int argc, char* argv[]){
6     int cmd;
7     cmd = getopt(argc, argv, "abcd");
8     while((cmd = getopt(argc, argv, "abcd")) > 0){
9         switch(cmd){
10             case 'a':
11                 printf("a option added\n");
12                 break;
13             case 'b':
14                 printf("b option added\n");
15                 break;
16             case 'c':
17                 printf("c option added\n");
18                 break;
19             case 'd':
20                 printf("d option added\n");
21                 break;
22             default :
23                 printf("wrong option added\n");
24                 break;
25         }
26     }
27     return 0;
28 }
```





# System programming

## Adding some options

Adding an option with getopt() is not efficient. Let's suppose that there are a lot of options, about 800. Alphabet is not enough to include it. So, We have to set an instructor multiple letters like 'aa', 'bb' using 'flag' variant.

```
int flag = 0;
int cmd;
DIR* dp;
struct dirent* p;
int i = 1;

while((cmd = getopt(argc, argv, "abcd")) > 0){
    switch(cmd){
        case 'a':
            flag |= 1;
            break;
        case 'b':
            flag |= 2;
            break;
        case 'c':
            flag |= 4;
            break;
        case 'd':
            flag |= 8;
            break;
    }
}
```

## File status (-l option)

```
howard@ubuntu: ~/HomeworkBackup/21th
1 #include <sys/types.h>
2 #include <sys/stat.h>
3 #include <unistd.h>
4 #include <stdio.h>
5 #include <pwd.h>
6 #include <grp.h>
7 #include <time.h>
8 int main(int argc, char** argv){
9     struct stat buf;
10    struct dirent *p;
11    struct passwd* pw;
12    struct group* gr;
13    struct tm* tm;
14    char ch;
15    char perm[11]="-----";
16    char rwx[4] = "rwx";
17    char sst[4] = "sst";
18    int i;
19    stat(argv[1], &buf);
20
21    if(S_ISDIR(buf.st_mode))
22        perm[0] = 'd';
23    if(S_ISREG(buf.st_mode))
24        perm[0] = '-';
25    if(S_ISFIFO(buf.st_mode))
26        perm[0] = 'p';
27    if(S_ISSOCK(buf.st_mode))
28        perm[0] = 's';
29    if(S_ISCHR(buf.st_mode))
30        perm[0] = 'c';
31    if(S_ISBLK(buf.st_mode))
32        perm[0] = 'b';
33
34    for(i=0; i<9; i++){
35        if((buf.st_mode >> (8-i)) & 1)
36            perm[i+1] = rwx[i%3];
37        for(i=0; i<3; i++){
38            if((buf.st_mode >> (11-i)) & 1)
39                if(perm[(i+1)*3] == '-')
40                    perm[(i+1)*3] = sst[i]^0x20;
41            else
42                perm[(i+1)*3] = sst[i];
43        }
44    }
```



# System programming

## File status ( -l option)

When you entered `ls -al`, you can see a result like below.

```
total 88
drwxrwxr-x  3 howard howard 4096 Mar 22 05:23 .
drwxrwxr-x 19 howard howard 4096 Mar 21 17:37 ..
-rw-rw-r--  1 howard howard  304 Mar 22 02:37 1.c
-rw-rw-r--  1 howard howard   0 Mar 22 03:51 1.txt
-rw-rw-r--  1 howard howard  504 Mar 22 04:54 2.c
-rw-rw-r--  1 howard howard  609 Mar 22 05:10 3.c
-rwxrwxr-x  1 howard howard 8816 Mar 22 04:46 a.out
-rwSr--r--  1 howard howard   3 Mar 21 23:31 a.txt
-rwxrwxr-x  1 howard howard 9792 Mar 22 02:50 debug
-rw-rw-r--  1 howard howard  472 Mar 22 05:13 ls_l.c
-rw-rw-r--  1 howard howard  328 Mar 22 01:48 lsmodule1.c
-rw-rw-r--  1 howard howard  384 Mar 22 02:53 lsmodule2.c
-rw-rw-r--  1 howard howard  418 Mar 22 05:14 lsmodule3.c
-rw-rw-r--  1 howard howard  623 Mar 22 05:14 lstest2.c
-rw-rw-r--  1 howard howard 1048 Mar 22 05:14 lstest.c
-rw-rw-r--  1 howard howard  858 Mar 22 05:15 perm2.c
-rw-rw-r--  1 howard howard 1221 Mar 22 05:23 perm3.c
-rw-rw-r--  1 howard howard  646 Mar 22 05:15 perm.c
drwxrwxr-x  5 howard howard 4096 Mar 21 18:08 test
-rwSr--r--  1 howard howard   0 Mar 22 04:56 test.txt
```

10 letters in the left side stand for st\_mod, which consist of 16 bits, ( \_ \_ \_ \_ s,s,t,r,w,x,r,w,x,r,w,x). First 4 letter include information of file type. The reasons why 3 r,w,x exist is that there are 3 types of user(user(root), group, the others).

s,s,t means setuid(user id), setgid(group id), sticky. Sticky has 2 functions. When the file is directory, sticky bit lets the file become shared file. When it is execute file, stikcy bit lets the process do swapping(made by LRU algorithm). These permissions can be controlled by "chmod 0000" instructor.

## Paging & Swap

Analyzing debugging process, some address are displayed. But the address aren't real address. These are virtual address.

think about playing a computer game. Assume Game program occupies 100 Giga byte. If all memory is real, 100GB memory are needed to playing game. But, Full upgraded Computer has 64 giga byte RAM. It's impossible that executing process only with physical memory. So there is "Paging" and "swap" to use virtual memory and disc.

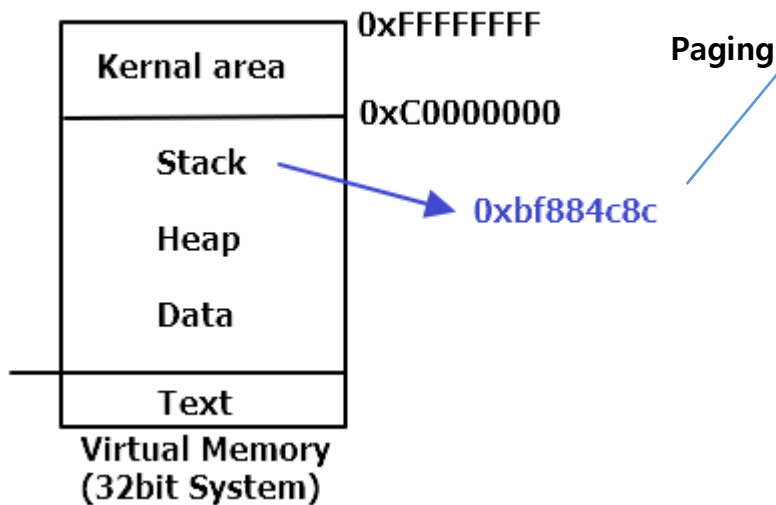




# System programming

## Paging & Swap

In 32bit System, virtual memory create like below in [task\_struct -> mm\_struct -> vm\_area\_struct]



Virtual address has 32bits. To analyze how to access to real memory, which means RAM, divide the address 3 parts, 10 bits, 10 bits, 12 bits.  
f.e, "0xbfb884c8c" is converted to  
"1011111110 0010000100 110010001100"

First 10 bit : page directory address  
Next 10 bit : page table address  
Last 12 bit : physical frame address in physical memory

