

2018. 3. 22 목 – 21 회차

과정 : TI, DSP, Xilinx Zynq FPGA, MCU 기반의 프로그래밍 전문가 과정

Prof. 이상훈

gcccompil3r@gmail.com

Stu. 정상용

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

## Linux 운영체제 \_ 4

### 1. ls(숨긴파일을 제외한 ls 기능)

```
#include <stdio.h>
#include <dirent.h>

int main()
{
    DIR *dp; // 파일 디렉토리(파일 포인터의 개념)
    int i = 0;
    struct dirent *p; // 디렉토리내에 있는 파일 & 디렉토리 따위의 내용
    dp = opendir("."); // "."안에 있는 dir 를 오픈, return : 오픈한 디렉토리의 주소값
    // dir 또한 open 을 해야만 사용가능
    while(p = readdir(dp)) // readdir 은 dp(디렉토리) 안의 리스트를 읽음, 첫번째 내용을 포인터, 두번째
        내용을 포인터 .... 마지막은 NULL
    {
        if(p->d_name[0] == '.') /* linux 에서 숨긴파일을 생성할 경우, 파일이름 앞에 "."을 기입한다.
                                첫 번째 파일의 이름의 첫번째 문자를 파악
                                => 숨긴파일을 찾는 중 */

            continue; // /* continue 시, i++까지 제낀다.
                       => 숨긴파일은 아무 작업없이 그냥 넘어가도록 한다. */

        printf("%-16s ", p->d_name); // %-16s : 문자열(s)를 16 자리에서 왼쪽부터 출력
        if((i+1) % 5 == 0) // 다섯개 출력시 개행
            printf("\n");
        i++;
    }
    printf("\n");
    closedir(dp);
    return 0;
}
```

→ Keep in mind

1. #include <dirent.h> : DIR, struct dirent, opendir, readdir, closedir
2. DIR \*valuable : valuable 은 dir 를 가리키는 포인터로 설정
3. struct dirent \*valuable : valuable points the files and directories in directory

4. readdir, closedir 은 dir 단위의 read, close
5. readdir(dir)의 return 은 dir 내의 파일 또는 디렉토리를 순차적으로 하나씩 가리키는 포인터
6. opendir 의 중요한 의미 : dir 또한 open 후 사용가능

## 2. getopt : 원하는 옵션 출력하기

```
#include <stdio.h>
```

```
#include <unistd.h> //getopt
```

```
int main(int argc, char **argv[])
{
    int cmd;
    while((cmd = getopt(argc, argv, "ab")) > 0)
    {
        printf("cmd = %c\n", cmd); // a, b 옵션을 제외하고는 ?로 출력
        switch(cmd)
        {
            case 'a' :
                printf("a option\n");
                break;
            case 'b' :
                printf("b option\n");
                break;
            default:
                printf("unknown option\n");
        }
    }
    return 0;
}
```

→ keep in mind

1. #include <unistd.h> : getopt
2. 출력시 ./a.out “-” 반드시 - 추가 후 작성해야함(Ex. ./a.out -acd -dfaz)
3. 주어진 옵션외에는 cmd = ? 로 인식
4. 위 프로그램의 단점: 옵션이 문자하나밖에 받지 못함(‘abc’처럼 단어 따위의 것들을 옵션으로 인식불가)

## 3. 1 & 2 의 결합한 형태 - 조건에 따라 숨긴파일도 출력(밑 예제의 조건 : -a)

```
#include <stdio.h>
```

```
#include <dirent.h>
```

```
#include <unistd.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') // 옵션이 a 가 아닌 경우
```

```

{
    if(p->d_name[0] == '.') // 숨긴파일은 출력하지 않음 → 옵션이 a 인 경우, 숨긴파일도 출력
        continue;
}
printf("%-16s", p -> d_name);
if((i+1)%5 == 0)
    printf("\n");
i++;
}
printf("\n");
closedir(dp);
return 0;

```

#### 4. 예제 3 과 비슷한 기능(But, 숨긴 파일을 보는 조건을 flag 로 제시)

```

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

```

```

int main(int argc, char **argv)
{
    DIR *dp;
    int i = 0;
    int cmd;
    struct dirent *p;
    int flag = 0;
    while((cmd = getopt(argc, argv, "alRi")) > 0)
    {
        switch(cmd)
        {
            case 'a':
                flag |= 1;
                break;
            case 'l':
                flag |= 2;
                break;
            case 'R':
                flag |= 4;
                break;
            case 'i':
                flag |= 8;
                break;
        }
    }

    dp = opendir(".");
    while(p = readdir(dp))
    {
        if(!(flag&1))
        {
            if(p->d_name[0] == '.')
                continue;
        }
        printf("%-16s", p->d_name); //16 이 넘을 경우, 문자가 잘리지 않고 한 줄에 5 개가 아닌 4 개가 출력된
        다.
        if((i+1)%5 == 0)

```

→ need to study

1. 비트연산자

[illegible]