

시스템 프로그래밍 실습

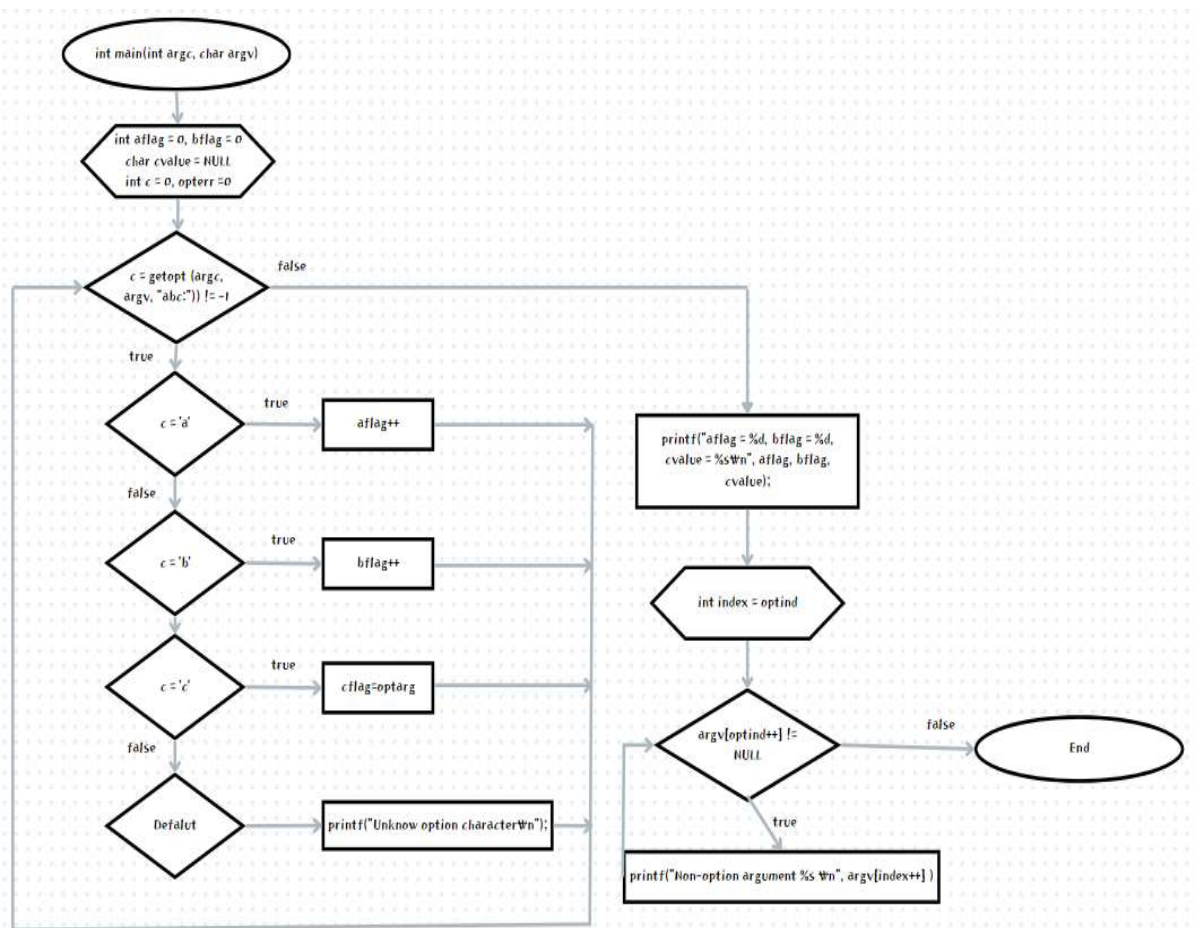
# [Assignment1-1]

Class : D 반(실습 2 금 56)  
Professor : [담당 교수님 명]  
Student ID : 2022202104  
Name : 김유찬

# Introduction

FTP 서버를 만들어보는 시간을 갖을 것이다. FTP 에서도 리눅스 환경처럼 instruction 과 option 에 따라 서로 다른 동작을 할 수 있게 만들 것이다. 하지만 이를 구별하기 위해서는 unistd 헤더 파일에 있는 외부 전역변수와 함수를 이용하여 각각의 옵션을 확인할 수 있고 옵션에 해당하는 string 으로 저장할 수 있고 또 그 옵션을 입력 받았을 때 특정한 기능을 할 수 있도록 만들 것이다.

## Flow chart



# Pseudo code

```
int main (int argc, char **argv)
{
    int aflag = 0, bflag = 0;
    char *cvalue = NULL;
    not error message;
    while (c is word with options)
    {
        switch (c)
        {
            case 'a':
                aflag++;
                break;
            case 'b':
                bflag++;
                break;
            case 'c':
                cvalue is arugment after option c ;
                break;
            case '?':
                printf("Unknow option character\n");
                break;
        }
    }
    printf("aflag = (aflag value), bflag = (bflag value), cvalue = (cflag value)");
    int index = optind;
    while(argv[index] is not null)
        printf("Non-option argument (argv[index++])");
}
```

# 결과화면

```
////////////////////////////////////
// File Name : kw2022202104_opt.c //
// Date : 2024/03/29 //
// OS : Ubuntu 20.04.6 LTS 64bits
//
// Author : Kim You Chan //
// Student ID : 2022202104 //
// -----//
// Title: System Programming Assignment #1-1 ( ftp server ) //
// Description : option도 함께 입력 받아 작성할 수 있게 한다. //
////////////////////////////////////
#include <unistd.h>
#include <stdio.h>
int main (int argc, char **argv)
{
    int aflag = 0, bflag = 0;
    char *cvalue = NULL; //c옵션 문자열을 입력받기 위해 출력
    int c=0;
    opterr = 0; //이 값이 0이 아니면 에러시 발생 메시지가 출력된다. 0이면 에러시 메시지가 출력되지 않는다. default =0
    //옵션과 문자열을 끝까지 확인한다.
    while ((c = getopt (argc, argv, "abc:")) != -1) //c:이기 때문에 c 뒤 문자열을 optarg로 받을 수 있다. getopt 함수는 전달된 인수를 옵션에 따라 구별할 수 있도록 한 함수이다.
    {
        switch (c)
        {
            case 'a': //옵션이 a일 때
                aflag++; //aflag 값이 1추가된다.
                break;
            case 'b': //옵션이 b일 때
                bflag++; //bflag 값이 1추가된다.
                break;
            case 'c': //옵션이 c일 때
                cvalue = optarg; //-c옵션 뒤에 있는 문자열을 optarg이 담겨있다.
                break;
            case '?':
                printf("Unknow option character\n"); //다른 옵션이 나오면 출력된다.
                break;
        }
    }

    printf("aflag = %d, bflag = %d, cvalue = %s\n", aflag, bflag, cvalue); // 잘 작동하는지 출력해본다.

    //optind는 옵션에 붙어있는 것까지 고려하여 입력 배열의 index를 알려주기 때문에 optind를 써야한다.
    int index = optind; //외부 전역함수를 함부로 바꾸면 안 되기 때문에 index라는 변수에 저장했다. 또한 이제부터는 옵션을 고려하지 않기 때문에 optind를 굳이 안 써도 된다.
    while(argv[index] != NULL) //더이상 입력값이 없으면 NULL이기 때문에 NULL까지 반복한다.
        printf("Non-option argument %s \n", argv[index++]); // 옵션을 제외한 나머지 입력값을 차례로 출력한다.
    return 0;
}
```

getopt(argc, argv, "[options]") : 전달된 인수를 옵션에 따라 구별할 수 있도록 한 함수이다. argv 를 읽는 과정에 option 이 없으면 -1 를 반환한다.

## 외부 전역함수

1. optind : 그 다음 작업할 argv 인덱스를 가리킨다. (char \* : 데이터 형)
2. optarg : 옵션 뒤에 있는 문자열을 담고 있다. (int : 데이터 형)
3. opterr : 이 값이 0 이면 에러 발생시 메시지를 출력한다. 1 이면 에러 시 메시지가 출력되지 않는다. default 는 0 이다. (int : 데이터 형)
4. optopt : 에러가 나면 원인을 제공한 문자를 저장하는 변수이다. (int : 데이터 형)
5. optarg : 옵션 뒤에 있는 인자를 받아옴

```

kw2022202104@ubuntu:~/2_ls$ ./kw2022202104_opt
aflag = 0, bflag = 0, cvalue = (null)
kw2022202104@ubuntu:~/2_ls$ ./kw2022202104_opt -a -b
aflag = 1, bflag = 1, cvalue = (null)
kw2022202104@ubuntu:~/2_ls$ ./kw2022202104_opt -ab
aflag = 1, bflag = 1, cvalue = (null)
kw2022202104@ubuntu:~/2_ls$ ./kw2022202104_opt -c foo
aflag = 0, bflag = 0, cvalue = foo
kw2022202104@ubuntu:~/2_ls$ ./kw2022202104_opt -cfoo
aflag = 0, bflag = 0, cvalue = foo
kw2022202104@ubuntu:~/2_ls$ ./kw2022202104_opt arg1
aflag = 0, bflag = 0, cvalue = (null)
Non-option argument arg1
kw2022202104@ubuntu:~/2_ls$ ./kw2022202104_opt -a arg1
aflag = 1, bflag = 0, cvalue = (null)
Non-option argument arg1
kw2022202104@ubuntu:~/2_ls$ ./kw2022202104_opt -c foo arg1
aflag = 0, bflag = 0, cvalue = foo
Non-option argument arg1
kw2022202104@ubuntu:~/2_ls$ ./kw2022202104_opt -a -
aflag = 1, bflag = 0, cvalue = (null)
Non-option argument -
kw2022202104@ubuntu:~/2_ls$ ./kw2022202104_opt -aa
aflag = 2, bflag = 0, cvalue = (null)
kw2022202104@ubuntu:~/2_ls$ ./kw2022202104_opt -d -a
aflag = 1, bflag = 0, cvalue = (null)
kw2022202104@ubuntu:~/2_ls$ ./kw2022202104_opt foo foo
aflag = 0, bflag = 0, cvalue = (null)
Non-option argument foo
Non-option argument foo
kw2022202104@ubuntu:~/2_ls$

```

명령어와 옵션을 적으면 argv 2 차원 배열에 입력이 된다. argv[0]에는 실행한 파일이 들어가고 두번째부터는 argv[1]에, 세번째부터는 argv[2]에 입력된다. -c foo 와 -cfoo 도 똑같이 인식해야한다. 그러기 위해서는 getopt 함수를 while 문과 switch 문을 이용하면 된다. 지정된 옵션 이외 인자들은 optind 를 다른 변수로 받아들이고 1 씩 증가해서 argv 배열에 접근하면 알아낼 수 있다. 위에 사진은 -a 이나 -b 옵션이면 aflag 이나 bflag 가 1 씩 증가하고 -c 옵션이면 뒤에 인자를 optarg 를 통해 받아들일 수 있다. opterr =0 으로 해서 결과화면에 에러가 있지만 에러메세지가 안 뜨는 것을 알 수 있다. 또한 마지막 명령어를 보면 option 이 아닌 문자를 차례로 알려주는 기능도 있다.

## 고찰

지금까지 c 언어로 코딩할 때는 scanf 로 입력을 받았는데 이번에는 실행파일을 cli 로 입력하면서 매개인자로 입력 받는 형태였다. 또한 unistd 헤더 파일에 제공하는 extern global variable 인 optind, optarg, opterr 등 변수 중 optind 가 처음엔 왜 필요한가 했지만 -c chell, -cchell 를 똑같이 인식하기 위해서는 꼭 있어야하는 extern global variable 이라는 것을 이번 과제를 통해 깨닫게 되었다. 이 함수를 통해 앞으로 명령어와

옵션을 구별하여 인식하고 그에 맞게 작동할 수 있게 되었다. 내가 구현한 함수 switch 문 안에 각 옵션에 맞게 동작하는 함수들을 만들어 FTP 서버 instruction 과 option 을 구현할 수 있을 것이다.

## Reference

시스템프로그래밍 실습 강의자료