시스템 프로그래밍 실습

[Assignment1-1]

Class : [금요일 1, 2 교시]

Professor : [최상호 교수님]

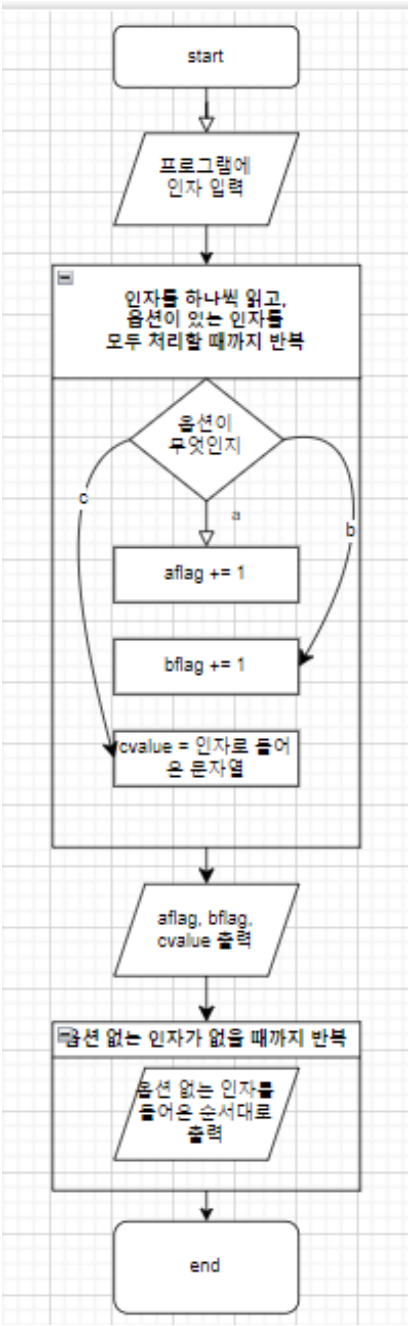
Student ID : [2020202034]

Name : [김태완]

Introduction

이번 과제는 ftp 서버 구현에 필요한 명령어 설계를 위해 옵션과 인자를 파싱하는 프로그램을 만드는 내용입니다. getotp 함수를 사용하여 옵션과 그에 대한 인자를 파싱하여 원하는 동작을 수행할 수 있도록 구현하는 것이 목표입니다. 이 과제를 수행함으로써, 다음 차시에서 명령어를 구현할 때 파싱 부분보다 기능 자체의 구현에 더욱 집중할 수 있습니다.

Flow chart



Pseudo code

```
main(integer argc, string array argv)
{
    integer aflag = 0
    integer bflag = 0
    string cvalue = NULL
    integer index
    character c

    opterr = 0

    repeat until read all optional arguments in argv
    {
        read a argument from argv
        parse command line options using getopt
         -a option aflag = aflag + 1
         -b option bflag = bflag + 1
         -c option cvalue = optarg(argument provided)
        point the next argument
    }

    print aflag, bflag, and cvalue


    repeat until read all non-optional arguments in argv
    {
        read a argument from argv
        print the non-option argument
        point the next argument
    }

    return 0
}
```

결과화면

```
kw2020202034@ubuntu:~/system_programming_1$ make
gcc -o kw2020202034_opt  kw2020202034_opt.c
```

먼저, make 를 실행해 소스파일을 컴파일하였습니다. 이때, -o 옵션을 주어 실행파일명을 -o 옵션의 인자로 주어진 kw2020202034_opt 로 만들어지도록 하였습니다.

```
kw2020202034@ubuntu:~/system_programming_1$ ./kw2020202034_opt -a -b -c value
aflag = 1, bflag = 1, cvalue = value
```

프로그램에 -a, -b, -c 옵션(인자는 value)를 하나씩 주어 결과를 확인하였습니다. 결과는 aflag 가 1 번, bflag 가 1 번, cvalue 는 -c 뒤에 인자로 들어온 value 로 입력과 출력이 잘 매칭됨을 볼 수 있습니다.

```
kw2020202034@ubuntu:~/system_programming_1$ ./kw2020202034_opt -abaab
aflag = 3, bflag = 2, cvalue = (null)
```

-a 옵션과 -b 옵션을 번갈아서 주었는데, a 가 3 번, b 가 2 번 주어졌습니다. 이는 프로그램의 결과를 보면 aflag = 3, bflag = 2, cvalue 는 주어지지 않았으니 (null)로 잘 출력됨을 볼 수 있습니다.

```
kw2020202034@ubuntu:~/system_programming_1$ ./kw2020202034_opt -a -cval -b
aflag = 1, bflag = 1, cvalue = val
```

-a, -c 옵션(공백 없이 인자 val 을 추가), -b 옵션을 주었습니다. 결과는 a, b 한 번씩, c 의 인자 val 이 잘 들어갔음을 볼 수 있습니다.

```
kw2020202034@ubuntu:~/system_programming_1$ ./kw2020202034_opt -ba none
aflag = 1, bflag = 1, cvalue = (null)
Non-option argument none
```

옵션 -b, -a, 그리고 옵션을 주지 않은 인자 'none'을 프로그램에 인자로 주었습니다. 그 결과는 aflag = 1, bflag = 1, 그리고 non-option argument 'none'이 잘 출력되었습니다.

```
kw2020202034@ubuntu:~/system_programming_1$ ./kw2020202034_opt nono -c yes
aflag = 0, bflag = 0, cvalue = yes
Non-option argument nono
```

그 다음 줄에는 옵션 없는 인자 nono, -c 옵션과 그 인자 yes 를 주었는데, 주지 않은 a, b 옵션의 flag 는 0 이고, cvalue 는 yes, 그리고 non option 인자 nono 가 잘 들어갔습니다.

```
kw2020202034@ubuntu:~/system_programming_1$ ./kw2020202034_opt -v abcd - all -c
spe
aflag = 0, bflag = 0, cvalue = spe
Non-option argument abcd
Non-option argument -
Non-option argument all
```

존재하지 않는 옵션인 -v 옵션과 옵션 없는 인자 abcd, -(뒤에 문자가 없으니 문자열 취급), all, 그리고 -c 옵션과 그의 인자인 spe 를 주었습니다. 그 결과 주지 않은 aflag = bflag = 0, cvalue = spe, 그리고 주어진 순서대로 옵션 없는 인자들이 차례로 출력되었음을 볼 수 있습니다.

```
kw2020202034@ubuntu:~/system_programming_1$ ./kw2020202034_opt
aflag = 0, bflag = 0, cvalue = (null)
```

마지막으로, 인자를 주지 않았을 때 a,b,c 옵션이 각각 0, NULL 을 나타내어 인자가 없어도 오류가 나지 않음을 볼 수 있습니다.

고찰

예제코드를 바탕으로 프로그램을 완성시키기 위해 추가 작성한 코드는 크게 2 부분으로 나눌 수 있는데, switch 문 내에서 옵션이 있는 인자를 처리하는 부분, 그리고 반복문으로 옵션 없는 인자들을 출력하는 부분입니다. 저는 이 부분 들에서 인자들의 예외처리(-a, -b, -ab 처럼 두 옵션이 같이 붙어 있는 경우 처리, -c value, -cvalue 처럼 -c 옵션에 문자열이 붙어 있고 떨어져 있는 경우 예외처리, 등)를 해야 하는 줄 알았습니다. 그러나 실습자료를 통해 getopt 함수가 이에 대한 모든 예외처리를 해주어 제가 추가로 해줄 부분은 없다는 것을 배웠습니다. 추가로, getopt 함수가 argv 내에서 순서를 바꿔주어 옵션 있는 인자가 앞으로, 없는 인자가 뒤로 오게 해준다는 사실을 실습 자료로 공부해 이를 이용해 옵션 없는 인자를 반복문으로 처리할 때 다음으로 처리할 인자를 가리키는 인덱스 optind 를 이용해 들어온 순서대로 인자들을 처리할 수 있었습니다.

Reference

x