시스템 프로그래밍 실습

Assignment1-1

Class : 금 1, 2 분반

Professor : 최상호 교수님

Student ID : 2020202031

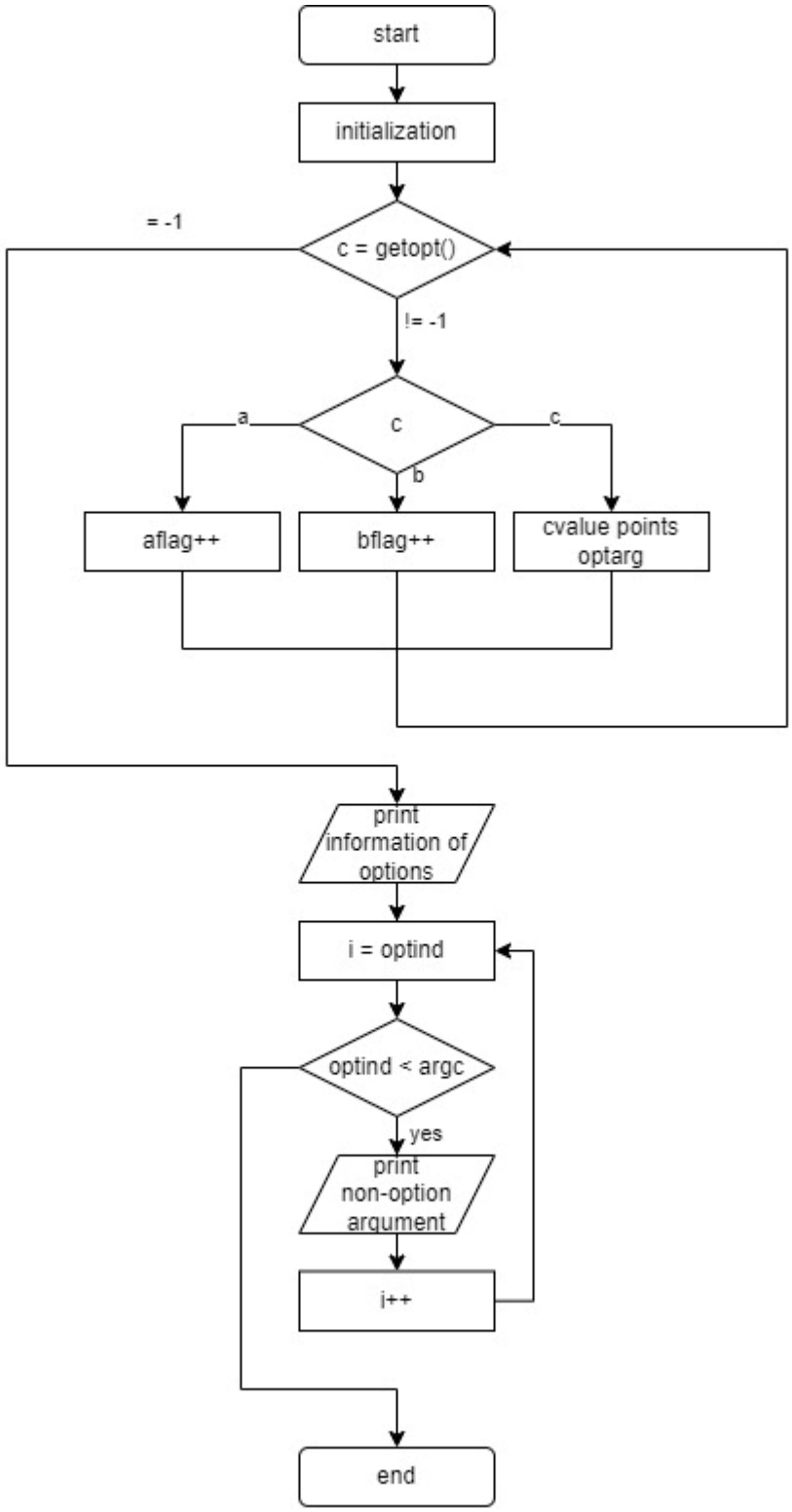
Name : 김재현

Introduction

C 언어에서, getopt() 함수는 명령어에서 -기준으로 옵션을 파싱하고 이를 프로그램에서 사용할 수 있는 형태로 처리합니다. 이를 통해 프로그래머는 사용자의 입력을 쉽게 이해하고 적절히 대응할 수 있습니다.

이 실습에서는 getopt() 함수를 사용하여 사용자로부터 입력 받은 명령어에 사용된 옵션의 개수와 옵션의 옵션을 출력하고, 옵션이 아닌 parameter 는 따로 출력해주는 프로그램을 작성합니다.

Flow chart


start
initialization
= -1
c = getopt()
!= -1
c
a
b
c
aflag++
bflag++
cvalue points optarg
print information of options
i = optind
optind < argc
yes
print non-option argument
i++
end

Pseudo code

Start function main

aflag = 0

bflag = 0

cvalue = NULL.

Declare integer c.

opterr = 0 (disable error message printing).

Start parsing command-line options:

    Call getop(argc, argv, "abc:"), store the result in c.

    While c is not equal to -1:

        Switch on c

            Case 'a':

                Increment aflag.

            Case 'b':

                Increment bflag.

            Case 'c':

                Assign cvalue to optarg.

    Print option information:

        Print aflag, bflag, cvalue

    Print non-option arguments:

        for (from optind to argc - 1):

            Print argv[i].

End function main.

결과화면

```
kw2020202031@ubuntu:~/sys_programing/1-1$ ./kw2020202031_opt
aflag = 0, bflag = 0, cvalue = (null)
```

어떠한 옵션도 입력되지 않았기 때문에

aflg = 0, bflag = 0, cvalue = (null)이 출력된 것을 확인할 수 있습니다.

```
kw2020202031@ubuntu:~/sys_programing/1-1$ ./kw2020202031_opt -a -b
aflag = 1, bflag = 1, cvalue = (null)
```

옵션이 a, b 한 개씩 입력됐기 때문에

aflg = 1, bflag = 1, cvalue = (null)이 출력된 것을 확인할 수 있습니다.

```
kw2020202031@ubuntu:~/sys_programing/1-1$ ./kw2020202031_opt -ab
aflag = 1, bflag = 1, cvalue = (null)
```

옵션이 a, b 한 개씩 입력됐기 때문에

aflg = 1, bflag = 1, cvalue = (null)이 출력된 것을 확인할 수 있습니다.

```
kw2020202031@ubuntu:~/sys_programing/1-1$ ./kw2020202031_opt -c foo
aflag = 0, bflag = 0, cvalue = foo
```

옵션이 c 한 개만 입력됐고, c 옵션 뒤에 foo 문자열이 입력됐으므로 foo 는 c 의 옵션으로서 동작하는 것을 알 수 있습니다. 따라서

aflg = 0, bflag = 0, cvalue = foo 가 출력된 것을 확인할 수 있습니다.

```
kw2020202031@ubuntu:~/sys_programing/1-1$ ./kw2020202031_opt -cfoo
aflag = 0, bflag = 0, cvalue = foo
```

옵션이 c 한 개만 입력됐고, c 옵션 뒤에 foo 문자열이 입력됐으므로 foo 는 c 의 옵션으로서 동작하는 것을 알 수 있습니다. c 옵션 뒤의 문자열은 공백의 여부에 상관없이 c 의 옵션으로 인식됩니다. 따라서

aflg = 0, bflag = 0, cvalue = foo 가 출력된 것을 확인할 수 있습니다.

```
kw2020202031@ubuntu:~/sys_programing/1-1$ ./kw2020202031_opt arg1
aflag = 0, bflag = 0, cvalue = (null)
Non-option argument arg1
```

옵션은 입력되지 않았고, arg1 이라는 문자열만이 입력된 것을 확인할 수 있습니다. 따라서

aflg = 0, bflag = 0, cvalue = (null)이 출력된 것을 확인할 수 있으며, 옵션이 아닌 입력 arg1 은 Non-option argument 로 출력됩니다.

```
kw2020202031@ubuntu:~/sys_programing/1-1$ ./kw2020202031_opt -a arg1
aflag = 1, bflag = 0, cvalue = (null)
Non-option argument arg1
```

옵션은 a 한 개만 입력됐고, arg1 이라는 문자열만이 입력된 것을 확인할 수 있습니다. 옵션 a 는 옵션을 가지지 않으므로 arg1 은 Non-option argument 입니다. 따라서

aflg = 1, bflag = 0, cvalue = (null)이 출력된 것을 확인할 수 있으며, 옵션이 아닌 입력 arg1 은 Non-option argument 로 출력됩니다.

```
kw2020202031@ubuntu:~/sys_programing/1-1$ ./kw2020202031_opt -c foo arg1
aflag = 0, bflag = 0, cvalue = foo
Non-option argument arg1
```

옵션이 c 한 개만 입력됐고, c 옵션 뒤에 foo 문자열이 입력됐으므로 foo 는 c 의 옵션으로서 동작하는 것을 알 수 있습니다. 그리고 arg1 라는 문자열이 입력됐습니다. arg1 은 Non-option argument 이므로

aflg = 0, bflag = 0, cvalue = foo 가 출력되며, arg1 은 Non-option argument 로 출력됩니다.

```
kw2020202031@ubuntu:~/sys_programing/1-1$ ./kw2020202031_opt -a -
aflag = 1, bflag = 0, cvalue = (null)
Non-option argument -
```

옵션이 a 한 개만 입력됐고, 그 뒤에 문자 '-'가 입력됐습니다. '-'은 Non-option argument 입니다. 따라서

aflg = 1, bflag = 0, cvalue = (null)이 출력되고, -은 Non-option argument 로 출력됩니다.

```
kw2020202031@ubuntu:~/sys_programing/1-1$ ./kw2020202031_opt -aa
aflag = 2, bflag = 0, cvalue = (null)
```

옵션이 a 두 개만 입력됐습니다.

aflg = 2, bflag = 0, cvalue = (null)이 출력됩니다.

```
kw2020202031@ubuntu:~/sys_programing/1-1$ ./kw2020202031_opt -d -a
aflag = 1, bflag = 0, cvalue = (null)
```

옵션이 a, d 각각 한 개씩만 입력됐습니다.getopt 에서 optstring 에 d 는 포함되지 않았지만 opterr 를 0 으로 설정했기 때문에 별다른 error message 가 발생하지 않는 것을 확인할 수 있습니다.

aflg = 1, bflag = 0, cvalue = (null)이 출력됩니다.

```
kw2020202031@ubuntu:~/sys_programing/1-1$ ./kw2020202031_opt foo foo
aflag = 0, bflag = 0, cvalue = (null)
Non-option argument foo
Non-option argument foo
```

아무런 옵션 없이 문자열 "foo"가 공백을 사이에 두고 연달아 두 번 입력됩니다. 이 두 문자열은 Non-option argument 로 취급됩니다. 따라서

aflg = 0, bflag = 0, cvalue = (null)이 출력되고, foo 는 Non-option argument 로 출력됩니다.

고찰

이번 실습을 통해 getopt() 함수를 반복문과 함께 사용하여 argv 를 끝까지 읽으면 Non-option arguments 는 뒤쪽으로 재정렬된다는 것을 알게 되었습니다. 또한 재정렬된 argv 에서 마지막 옵션의 다음 index 를 optind 가 저장하고 있어, 옵션을 제외한 나머지 문자열들에 접근할 수 있다는 것도 알게 되었습니다. 이를 통해 리눅스의 명령행 인터페이스(Command-Line Interface, CLI)가 동작하는 원리를 이해할 수 있었습니다.

Reference