

시스템 프로그래밍 실습

# [Assignment1-1]

Class : [D]

Professor : [최상호교수님]

Student ID : [2020202060]

Name : [홍왕기]

# Introduction

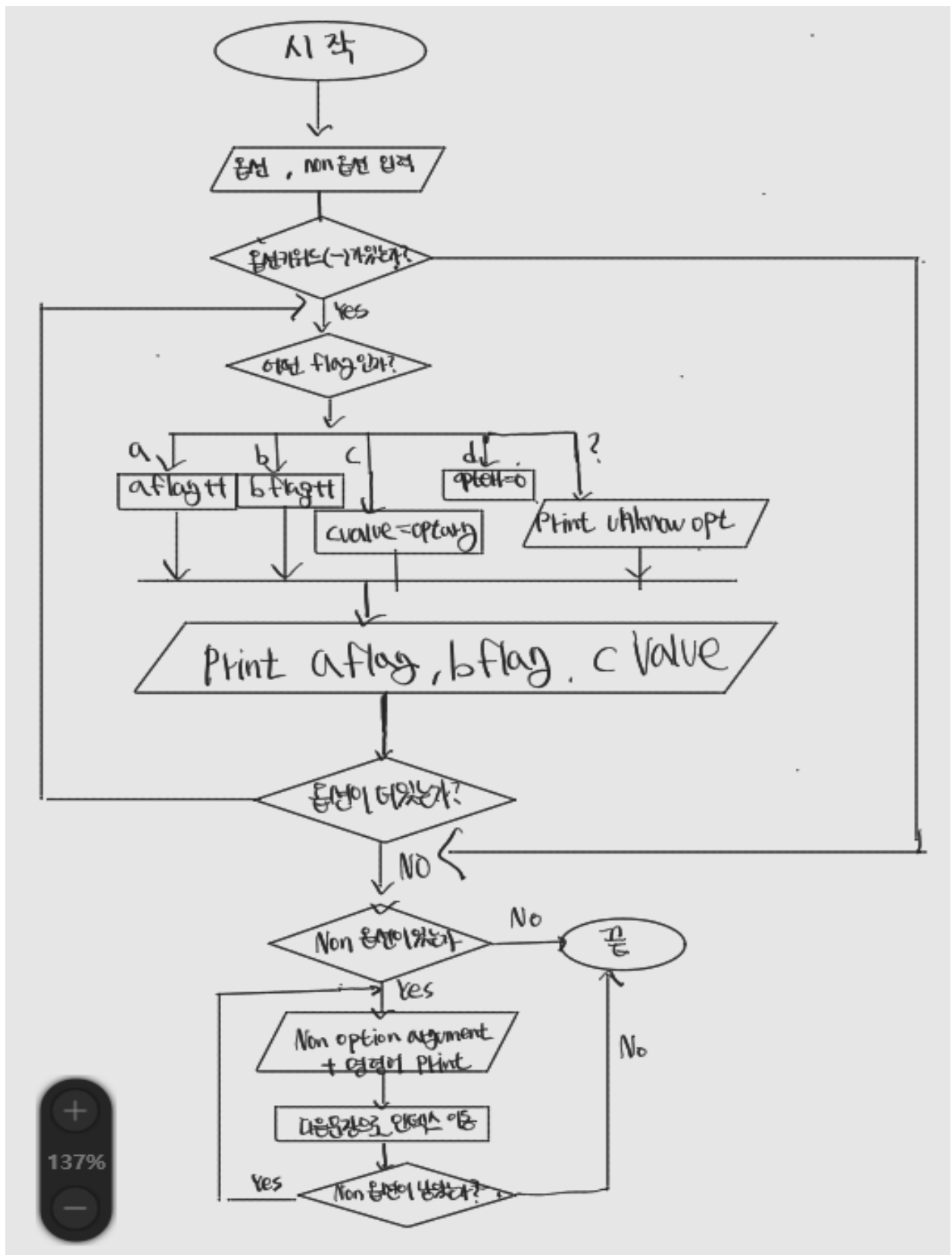
소프트웨어 개발에서 명령줄 인수를 처리하는 것은 매우 중요하다. 이를 효율적으로 수행하기 위해 `getopt()` 함수는 매우 유용한 도구이다. 우리는 최종적으로 완료할 FTP 서버를 개발할 것이다. 이를 위해 이번 프로젝트에선 `getopt()` 함수를 활용하여 사용자가 프로그램을 실행할 때 제어할 수 있는 다양한 옵션들을 효과적으로 처리할 것이다.

`getopt()` 함수는 C 프로그램에서 명령줄 옵션을 처리하는 데 사용된다. 이 함수는 명령줄에서 옵션을 읽어와서 해당하는 동작을 수행하고, 필요한 경우 옵션에 대한 인수를 처리할 수 있다. `getopt()` 함수를 이용하면 프로그램이 유연하게 동작하면서도 사용자가 제공하는 입력을 정확하게 처리할 수 있는 장점이 있다. `getopt()` 함수를 사용하여 옵션 관리를 구현함으로써 최종적인 FTP 서버를 구현하는 과정에서 다음과 같은 이점을 얻을 수 있다.

먼저, 사용자가 FTP 프로그램을 실행할 때 원하는 옵션을 지정할 수 있다. 예를 들어, `-u` 옵션을 사용하여 사용자 이름을 지정하거나, `-p` 옵션을 사용하여 비밀번호를 설정할 수 있다. `getopt()` 함수를 이용하여 프로그램이 명령줄에서 전달받은 옵션을 처리하면서 프로그램의 흐름을 조절할 수 있다. 이를 통해 사용자의 요구에 따라 다른 동작을 수행할 수 있게 된다. 마지막으로 `getopt()` 함수를 통해 FTP 프로그램에 간결하고 일관된 명령줄 인터페이스를 제공할 있게 된다. 이는 사용자가 프로그램을 보다 쉽게 이해하고 사용할 수 있도록 도와준다.

결국 `getopt()` 함수를 사용하여 FTP 프로그램을 개발하면 명령줄 인수를 효과적으로 처리하고 프로그램의 유연성을 높일 수 있다. 이를 통해 사용자가 FTP 프로그램을 보다 효율적으로 활용할 수 있으며, 파일 전송 및 관리 작업을 보다 편리하게 수행할 수 있을 것이다.

## Flow chart



## Pseudo code

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

int main(int argc, char **argv) {
    Declare variables:
        aflag = 0, bflag = 0
        cvalue = NULL
        index, c
    Set opterr to 0

    while ((c = getopt(argc,argv,"abdc: ") != -1)
        Switch on the option character:
            Case 'a':
                aflag++
            Case 'b':
                bflag++
            Case 'c':
                cvalue = optarg;
            Case 'd':
                opterr=0;
            Case '?':
                Print "Unknown option character"

    Print the values of aflag, bflag, and cvalue

    Loop through non-option arguments:
        Print each non-option argument

    Return 0
}
```

## 결과화면

먼저 ./testopt 실행 명령어를 alias 를 통해 testopt=./testopt 으로 설정하였다.

```
kw2020202060@ubuntu:~/Downloads$ make
gcc -o testopt kw2020202060_opt.c
kw2020202060@ubuntu:~/Downloads$ alias testopt=./testopt
kw2020202060@ubuntu:~/Downloads$ testopt
aflag = 0, bflag = 0, cvalue = (null)
kw2020202060@ubuntu:~/Downloads$ testopt -a -b
aflag = 1, bflag = 1, cvalue = (null)
kw2020202060@ubuntu:~/Downloads$ testopt -ab
aflag = 1, bflag = 1, cvalue = (null)
kw2020202060@ubuntu:~/Downloads$ testopt -c foo
aflag = 0, bflag = 0, cvalue = foo
kw2020202060@ubuntu:~/Downloads$ testopt -cfoo
aflag = 0, bflag = 0, cvalue = foo
kw2020202060@ubuntu:~/Downloads$ testopt arg1
aflag = 0, bflag = 0, cvalue = (null)
Non-option argument arg1
kw2020202060@ubuntu:~/Downloads$ testopt -a arg1
aflag = 1, bflag = 0, cvalue = (null)
Non-option argument arg1
kw2020202060@ubuntu:~/Downloads$ testopt -c foo arg1
aflag = 0, bflag = 0, cvalue = foo
Non-option argument arg1
kw2020202060@ubuntu:~/Downloads$ testopt -a -
aflag = 1, bflag = 0, cvalue = (null)
Non-option argument -
kw2020202060@ubuntu:~/Downloads$ testopt -aa
aflag = 2, bflag = 0, cvalue = (null)
kw2020202060@ubuntu:~/Downloads$ testopt -d -a
aflag = 1, bflag = 0, cvalue = (null)
kw2020202060@ubuntu:~/Downloads$ testopt foo foo
aflag = 0, bflag = 0, cvalue = (null)
Non-option argument foo
Non-option argument foo
```

) makefile 실행  
) ./testopt 를 testopt 커맨드 설정  
) 아무 옵션이 없기에 flag, value 모두 X  
) -a 옵션 a, b를 받아 a, b flag 증가  
) -1가 입력되어 -안나오는 4점부터 옵션인식 (a↑ b↑)  
) -c로 foo라는 추가적인 인자를 받는 옵션 추가.  
) 위와 마찬가지로 -1가 들어와 옵션으로 foo추가  
) -1가 없기에 옵션으로 인식 X Non옵션  
) -a를 옵션으로 인식하 flag 증가 -c가 아니므로 arg1 = Non옵션  
) foo를 옵션으로 받아기에 arg1은 옵션이 X  
) a를 옵션 추가 후 -뒤에 옵션이 없으므로 Non옵션  
) -aa 즉 옵션 aflag 두번 증가.  
) -d는 opt에 1으로 설정하 aflag 증가.  
) (-)가 없기에 Non옵션인 foo 두번 출력

## 고찰

getopt 인 파싱 함수를 구현하면서 처음에는 getopt() 함수를 사용하여 명령줄 인자를 파싱하는 방법에 대해 어려움을 느꼈다. 처음 보는 함수이기에 와닿지 않았던 것이다. 하지만 강의자료를 토대로 이 함수를 활용함으로써 gteopt()라는 함수에 대해 깊이 알게 되었고 코드를 구현하니 이론 적인 부분에서 완벽하게 깨달았다. 각 옵션과 플래그를 어떻게 정의하고 활용할지 결정하는 과정에서는 프로그래밍 능력 또한 향상되었다. getopt 구조와 주석을 활용하여 최대한 쉽게 설계함으로써 다른 사용자가 프로그램의 기능을 명확히 이해할 수 있도록 노력했다.

프로젝트를 진행하면서는 getopt() 함수와 옵션 문자열의 작동 방식을 이해하는데 시간을 오래 걸렸다. 옵션 문자열에 대한 이해를 바탕으로 각 옵션의 동작을 구현하는 과정에서 많은 것을 배웠던 시간이었다. 또한, 프로그램의 요구 사항에 맞게 case 문을 활용해 옵션과 플래그를 조정하는 방법을 익혔다.

이 getopt() 함수를 기반으로 한 FTP 서버를 구현하는 것은 미래에 도움이 많이 되는 프로젝트일 것이다. 이를 통해 네트워크 프로그래밍과 시스템 설계에 대한 깊은 이해를 얻을 수 있을 것이다. FTP 서버를 구현하는 과정에서는 네트워크 프로토콜 및 보안에 대한 지식을 확장하고, 사용자가 파일을 효율적으로 관리할 수 있는 서버를 구축하는 경험을 쌓을 것이다. 이러한 경험을 통해 미래에 배운 것을 적용하여 훌륭한 프로그래머가 될 것이다.

## Reference

<https://opensource.apple.com/source/Libc/Libc-186/stdlib.subproj/getopt.c.auto.html>

(getopt 함수 원본)