

2주차 결과보고서

전공: 컴퓨터공학과

학년: 2학년

학번: 20231632

이름: Jumagul

Alua

1. 목적

실습 과정에 개발한 fmt에 대하여 결과 보고한다

2. 실습1

```
cse20231632@cspro:~/실습1$ make
cc -c -o main.o main.c
cc -c -o blackcow.o blackcow.c
cc -c -o dog.o dog.c
cc -c -o turtle.o turtle.c
gcc -o animal main.o blackcow.o dog.o turtle.o
cse20231632@cspro:~/실습1$ ls
animal  animal.h  blackcow.o  dog.o  main.o  phone  turtle.o
animal.exe  blackcow.c  dog.c  main.c  Makefile  turtle.c
cse20231632@cspro:~/실습1$ ./animal
Hey guys! Here is my story
We were raising a cow years ago
And I wanted to get a dog
But got a turtle instead
cse20231632@cspro:~/실습1$
```

3개의 'dog.c', 'blackcow.c', 'turtle.c' 라는 소스 파일을 만들어 'animal.h'에 메스도하고 'main.c' 에
서 다시 불러 온다. Makefile 로 통해 'animal' 출력을 하게끔 컴파일 한다.

---->뒤쪽에 계속

실습2

```
cse20231632@cspiro:~/실습2$ gcc -g -o file babo.c
cse20231632@cspiro:~/실습2$ gdb file
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.5) 7.11.1
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from file...done.
(gdb) l
1      #include <stdio.h>
2
3      int main(void)
4      {
5          int i;
6          float num;
7
8          for (i=0; i<5; i++){
9              num=i/2 + i;
10             printf("num is %f \n", num);
(gdb) b 2
Breakpoint 1 at 0x40052e: file babo.c, line 2.
(gdb) run
Starting program: /sogang/under/cse20231632/실습2/file

Breakpoint 1, main () at babo.c:8
8          for (i=0; i<5; i++){
(gdb) display num
1: num = 0
(gdb) n
9              num=i/2 + i;
1: num = 0
(gdb) n
10             printf("num is %f \n", num);
1: num = 0
(gdb) n
num is 0.000000
8          for (i=0; i<5; i++){
1: num = 0
(gdb) n
```

```

(gdb) n
9      num=i/2 + i;
1: num = 0
(gdb) n
10     printf("num is %f \n", num);
1: num = 1
(gdb) n
num is 1.000000
8     for (i=0; i<5; i++){
1: num = 1
(gdb) n
9      num=i/2 + i;
1: num = 1
(gdb) n
10     printf("num is %f \n", num);
1: num = 3
(gdb) n
num is 3.000000
8     for (i=0; i<5; i++){
1: num = 3
(gdb) n
9      num=i/2 + i;
1: num = 3
(gdb) n
10     printf("num is %f \n", num);
1: num = 4
(gdb) n
num is 4.000000
8     for (i=0; i<5; i++){
1: num = 4
(gdb) n
9      num=i/2 + i;
1: num = 4
(gdb) n
10     printf("num is %f \n", num);
1: num = 6
(gdb) n
num is 6.000000
8     for (i=0; i<5; i++){
1: num = 6
(gdb) n
12    }
1: num = 6
(gdb) n
__libc_start_main (main=0x400526 <main>, argc=1, argv=0x7fffffff558,
    init=<optimized out>, fini=<optimized out>, rtd_fini=<optimized out>,
    stack_end=0x7fffffff548) at ../csu/libc-start.c:325
325    ../csu/libc-start.c: No such file or directory.

```

주어진 코드에서 틀렸던 부분은 i 변수가

int였기 때문이다. 출력은 ".5" 라는 값을 가지지 못해서 int를 float형식으로 바꿔 출력이 맞게 나올 것이다. 여기는 디버깅을 해, 일단 breakpoint을 2줄에 넣어 출력을 하나하나 확인해보는 작업을 했다.

실습3

```

cse20231632@cspro: ~/실습3
#include "Header.h"

void Remove_Blanks_At_The_End( char *line ) {
    int i, k, newline_flag = 0;

    // 전체 문장에 대하여 line[k] 가 줄바꿈이면 flag를 1로 set, '\0'이면 ~
    for ( k = 0; ; k++ ) {
        if(line[k] == '\n'){
            newline_flag=1;
            break;
        } else if(line[k] == '\0'){
            break;
        }
    }
    // 전체 문장에 대하여 line[i]가 space가 아니라면 break,
    for ( i = k-1; i >= 0; i-- ) {
        if (line[i] != ' '){
            break;
        }
    }
    i++;
    // flag가 1일때 문장의 마지막은 줄바꿈&space, flag가 1이 아니면 '\0'
    if ( newline_flag == 1 ) {
        line[i] = '\n';
    }
    else {
        line[i] = '\0';
    }
}
}

```

String_manipulation.c, 'Remove_Blanks_At_The_End' function

```

cse20231632@cspro:~/실습3$ make
gcc -W -g -c -o main.o main.c
gcc -W -g -c -o string_manipulation.o string_manipulation.c
gcc -W -g -c -o output.o output.c
gcc -W -g -o main main.o string_manipulation.o output.o
cse20231632@cspro:~/실습3$ ls
Header.h  main  main.c  main.o  Makefile  output.c  output.o  string_manipulation.c  string_manipulation.o  textfile.txt
cse20231632@cspro:~/실습3$ ./main textfile.txt
Here is the perfect system for cleaning your room. First move all of the
items that do not have a proper place to the center of the room. Get rid
of at least five things that you have not used within the last year.Take
out all of the trash, and place all of the dirty dishes in the kitchen
sink. Now find a location for each of the items you had placed in the
center of the room. For any remaining items, see if you can squeeze them
in under your bed or stuff them into the back of your closet. See, that
was easy

```

Makefile usage and result text

해당 함수에는 주어진 문자열에서 뒤에 있는 공백 문자들을 제거하거나 줄 바꿈 문자'\n'를 유지하거나 문자열의 끝 널 문자 '\0'으로 대체하여 반환하는 역할이다. make를 통해 'main.o string_manipulation.o output.o'가 생가며 컴파일 할 수 있게 된다. 주어진 textfile.txt을 불러주며 출력을 한다.