

Quiz 5 _ 디버깅 보고서

201724428 김승민

1. scanf 오류

```
(gdb) run
Starting program: /home/mimi9693/quiz5/stom
Enter the number of second
Enter 0 to end the program
1
Program received signal SIGSEGV, Segmentation fault.
0x00007ffff7a6cde5 in _IO_vfscanf_internal (s=<optimized out>, format=<optimized out>,
    argptr=argptr@entry=0x7fffffffdd28, errp=errp@entry=0x0) at vfscanf.c:1902
1902 vfscanf.c: No such file or directory.
(gdb) bt
#0 0x00007ffff7a6cde5 in _IO_vfscanf_internal (s=<optimized out>, format=<optimized out>,
    argptr=argptr@entry=0x7fffffffdd28, errp=errp@entry=0x0) at vfscanf.c:1902
#1 0x00007ffff7a785df in __isoc99_scanf (format=<optimized out>) at isoc99_scanf.c:37
#2 0x000000000040060b in main () at stom.c:12
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/mimi9693/quiz5/stom
Enter the number of second
Enter 0 to end the program
4
Program received signal SIGSEGV, Segmentation fault.
0x00007ffff7a6cde5 in _IO_vfscanf_internal (s=<optimized out>, format=<optimized out>,
    argptr=argptr@entry=0x7fffffffdd28, errp=errp@entry=0x0) at vfscanf.c:1902
1902 vfscanf.c: No such file or directory.
(gdb) list
1897     in vfscanf.c
```

(그림1_1)

stom을 실행했을 때 오류가 발생했고 backtrace와 list를 이용해 scanf ("%d", sec)에서 sec 앞에 "&"가 붙지 않은 오류임을 인지했다 따라서 scanf("%d", &sec)로 코드를 다시 작성했다.

```
#include <stdio.h>
#define S_TO_M 60

int main(void)
{
    int sec, min, left;

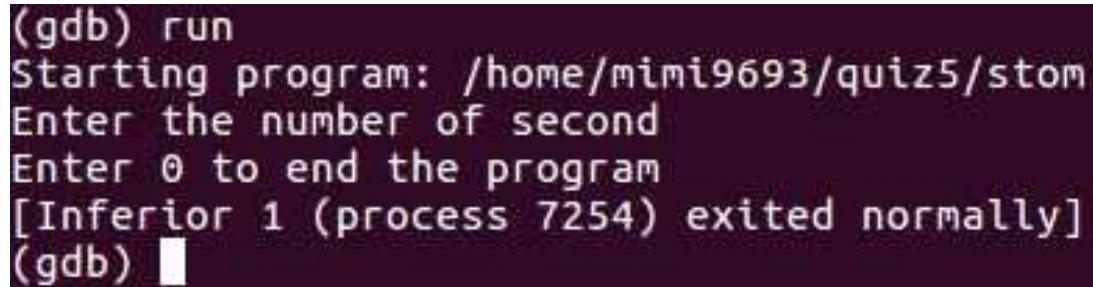
    printf("Enter the number of second\n");
    printf("Enter 0 to end the program\n");

    while(sec > 0) {
        scanf("%d", &sec);
        min = sec / S_TO_M;
        left = sec % S_TO_M;
        printf("%d sec is %d min, %d sec\n", sec, min, left);
        printf("Next input?\n");
    }

    return 0;
}
```

(그림1_2)

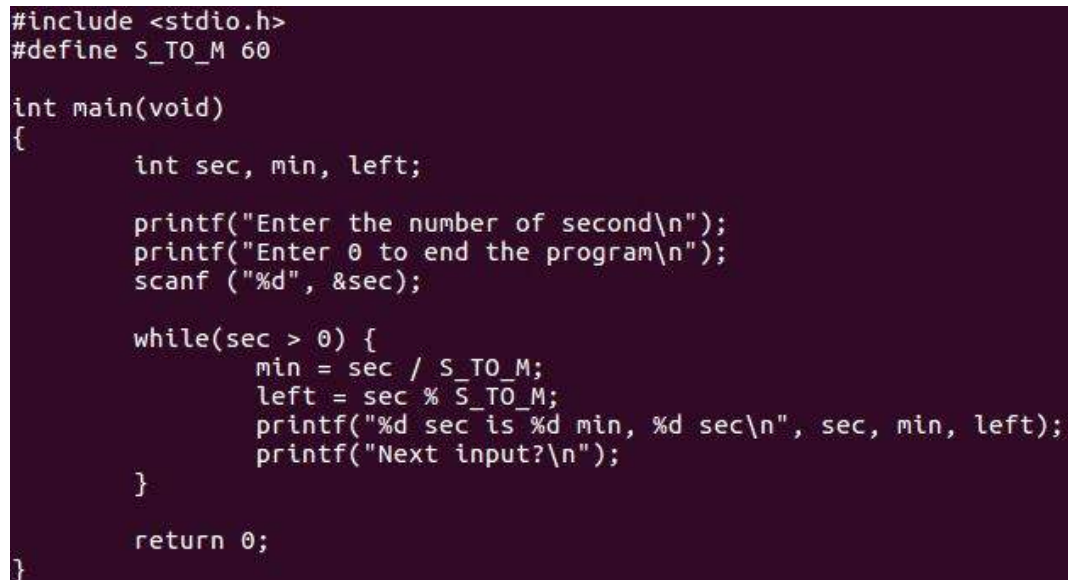
2. 변수값 미입력 오류



```
(gdb) run
Starting program: /home/mimi9693/quiz5/stom
Enter the number of second
Enter 0 to end the program
[Inferior 1 (process 7254) exited normally]
(gdb) █
```

(그림2_1)

일차 디버깅을 거친 후 다시 실행했을 때, 값이 입력되지 않은 채 프로그램이 종료가 되었다. line 6에 break를 한 후 n을 이용해 코드를 하나씩 실행해 오류를 찾았고 그 결과, sec에 대한 값이 미입력 되어서 프로그램이 while(sec>0)에서 false가 되었고 프로그램이 종료되었다. 따라서 scanf("%d", &sec)코드를 while문 밖으로 빼둠으로 오류를 해결했다.



```
#include <stdio.h>
#define S_TO_M 60

int main(void)
{
    int sec, min, left;

    printf("Enter the number of second\n");
    printf("Enter 0 to end the program\n");
    scanf ("%d", &sec);

    while(sec > 0) {
        min = sec / S_TO_M;
        left = sec % S_TO_M;
        printf("%d sec is %d min, %d sec\n", sec, min, left);
        printf("Next input?\n");
    }

    return 0;
}
```

(그림2_2)

3. 무한루프 오

```
Next input?
4 sec is 0 min, 4 sec
Next input?
4 sec is 0 min, 4 sec
Next input?
4 sec is 0 min, 4 sec
Next input?
4 sec is 0 min, 4 sec
Next input?
4 sec is 0 min, 4 sec
Next input?
4 sec is 0 min, 4 sec
Next input?
4 sec is 0 min, 4 sec
Next input?
4 sec is 0 min, 4 sec
Next input?
```

(그림3_1)

이차 디버깅을 끝낸 후 다시 실행하니 무한루프에 빠졌다.
이를 해결하기 위해 line 6에서부터 n을 눌러 차례대로
실행하니 sec의 값이 재입력 되지 않고 while문에서 항상
true인 값으로 실행함을 인지하였다. 따라서 line 16 아래에
scanf("%d", &sec)를 추가했고 0을 입력받았을 때
while문이 break되도록 코드를 짰다.

```
#include <stdio.h>
#define S_TO_M 60

int main(void)
{
    int sec, min, left;

    printf("Enter the number of second\n");
    printf("Enter 0 to end the program\n");
    scanf ("%d", &sec);

    while(sec > 0) {
        min = sec / S_TO_M;
        left = sec % S_TO_M;
        printf("%d sec is %d min, %d sec\n", sec, min, left);
        printf("Next input?\n");
        scanf("%d", &sec);
        if (sec==0) {
            break;
        }
    }

    return 0;
}
```

(그림3_2)

디버깅을 모두 마친 뒤 코드를 실행하였을 때 오류가 발생하지 않았다.

```
Enter the number of second
Enter 0 to end the program
0
[Inferior 1 (process 7298) exited normally]
(gdb) run
Starting program: /home/mimi9693/quiz5/stom
Enter the number of second
Enter 0 to end the program
104
104 sec is 1 min, 44 sec
Next input?
483
483 sec is 8 min, 3 sec
Next input?
0
```

(그림4)