
운영체제 실습 보고서

GDB Debugger 실습

제출일	2017/03/26
학 번	201000287
소 속	일어일문학과
이 름	유다훈

1 문제 설명

- make 를 통하여 나온 assign_01 실행 시 무한 루프 발생
- assign_01.c 의 getNodeItem(Node, int)함수의 두 번째 인자 값을 변경하여 문제를 해결해야함

2 문제 파악 절차

```
#include <stdio.h>
#include <stdlib.h>
#include "ds.h"

int main(void){
    Node *NodeList1 = (Node*)malloc(sizeof(Node));
    Node *NodeList2 = (Node*)malloc(sizeof(Node));
    NodeList1 = NULL;
    NodeList2 = NULL;

    //MakeNodeList1
    insertNode(&NodeList1, 36);
    insertNode(&NodeList1, 38);
    insertNode(&NodeList1, 41);
    insertNode(&NodeList1, 42);
    insertNode(&NodeList1, 35);
    //Make NodeList2
    insertNode(&NodeList2, 32);
    insertNode(&NodeList2, 42);
    insertNode(&NodeList2, 37);
    insertNode(&NodeList2, 47);
    insertNode(&NodeList2, 39);

    //You may correct getNodeItem Function?
    if(getNodeItem(&NodeList1, 0) * getNodeItem(&NodeList2, 5) == 1599)
        printf("This is Assign 1\n");

    while((getNodeItem(&NodeList2, 0) != 39) ||
          (getNodeItem(&NodeList1, 0) > 36)){
        //You Can't Escape
    }
    printf("You've Passed Assign 1\n");
}
```

텍스트 편집기를 이용하여 확인한 assign_01.c 의 코드

```
Line number 34 out of range; assign_01.c has 33 lines.
(gdb) b 25
Breakpoint 1 at 0x40078a: file assign_01.c, line 25.
(gdb) r
Starting program: /home/parallels/Dahun/4/assign_01/assign_01

Breakpoint 1, main () at assign_01.c:25
25         if(getNodeItem(&NodeList1, 0) * getNodeItem(&NodeList2, 5) == 15
```

GDB Debugger 를 이용하여 브레이킹포인트 설정

- 1 문제가 되는 25 번째줄 if 문에서 문제 확인을 위하여 브레이킹포인트 설정

```
(gdb) s
getNodeItem (ListHead=0x7fffffffddde8, index=0) at ds.c:23
23         Node *current = *ListHead;
```

브레이킹 포인트에서 함수 내부로 진입

- 2 Step(s)을 이용하여 브레이킹 포인트에서 함수 내부로 진입하여 어떻게 작동하는지 확인

```
(gdb) finish
Run till exit from #0  getNodeItem (ListHead=0x7fffffffddf0, index=5) at ds.c:23

Program received signal SIGSEGV, Segmentation fault.
0x000000000400698 in getNodeItem (ListHead=0x7fffffffddf0, index=5) at ds.c:29
29         return current->item;
```

finish 를 통해 리턴값 확인

- 3 Finish 를 이용하여 리턴값을 확인하려 하였으나 오류 발생

```
(gdb) print getNodeItem(&NodeList2, 5)

Program received signal SIGSEGV, Segmentation fault.
0x000000000400698 in getNodeItem (ListHead=0x7fffffffddf0, index=5) at ds.c:29
29         return current->item;
The program being debugged was signaled while in a function called from GDB.
GDB remains in the frame where the signal was received.
To change this behavior use "set unwindonsignal on".
Evaluation of the expression containing the function
(getNodeItem) will be abandoned.
When the function is done executing, GDB will silently stop.
(gdb) █
```

print 를 이용하여 함수의 리턴 값 확인 시도

- 4 Print 를 이용하여 함수의 리턴 값 확인을 시도하였으나 오류 및 경고문 확인
 - &NodeList2 의 현재 최대값은 4 로, index 값인 5 는 존재하지 않음을 if 문 상위 노드 생성 및 노드 값 입력 부분에서 유추할 수 있음.

```
Breakpoint 1, main () at assign_01.c:28
28         while((getNodeItem(&NodeList2, 0) != 39) ||
(gdb) print getNodeItem(&NodeList2, 0)
$1 = 32
(gdb) print getNodeItem(&NodeList1, 0)
$2 = 36
```

while 문에서 브레이킹포인트 설정

- 5 While 문에서 브레이킹포인트를 설정하고, print 를 이용하여 리턴값 확인
 - While 문의 조건에서 &NodeList2 의 값이 39 가 아니면서 &NodeList1 의 값이 36 보다 크지 않기 때문에 무한루프에 빠지는 것을 확인

3 문제 해결 절차

```
(gdb) print getNodeItem(&NodeList1, 2)
$1 = 41
(gdb) print getNodeItem(&NodeList2, 4)
$2 = 39
```

곱해서 1599 가 나오는 노드 값을 확인

- 1 If 문의 조건을 만족시키기 위한 노드 값을 확인

```
(gdb) print getNodeItem(&NodeList2, 4)
$1 = 39
(gdb) print getNodeItem(&NodeList1, 4)
$2 = 35
```

39 와 값이 같고, 36 과 값이 작은 노드값을 확인

- 2 While 문을 탈출하기 위한 조건을 만족시키기 위한 노드 값을 확인

```
//You may correct getNodeItem Function?
if(getNodeItem(&NodeList1, 2) * getNodeItem(&NodeList2, 4) == 1599)
    printf("This is Assign 1\n");

while((getNodeItem(&NodeList2, 4) != 39) ||
      (getNodeItem(&NodeList1, 4) > 36)){
    //You Can't Escape
}
```

무한 루프의 문제가 되는 인자 값들을 수정

4 문제 해결 결과

```
parallels@ubuntu:~/Dahun/4/assign_01$ ./assign_01
This is Assign 1
You've Passed Assign 1
```

값이 정상적으로 출력되는 것을 확인

5 수정한 소스코드

assign_01.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include "ds.h"
4
5 int main(void){
6     Node *NodeList1 = (Node*)malloc(sizeof(Node));
7     Node *NodeList2 = (Node*)malloc(sizeof(Node));
8     NodeList1 = NULL;
9     NodeList2 = NULL;
10
11     //MakeNodeList1
12     insertNode(&NodeList1, 36);
13     insertNode(&NodeList1, 38);
14     insertNode(&NodeList1, 41);
15     insertNode(&NodeList1, 42);
16     insertNode(&NodeList1, 35);
17     //Make NodeList2
18     insertNode(&NodeList2, 32);
19     insertNode(&NodeList2, 42);
20     insertNode(&NodeList2, 37);
21     insertNode(&NodeList2, 47);
22     insertNode(&NodeList2, 39);
23
24     //You may correct getNodeItem Function?
25     if(getNodeItem(&NodeList1, 2) * getNodeItem(&NodeList2, 4) == 1599)
26         printf("This is Assign 1\n");
27
28     while((getNodeItem(&NodeList2, 4) != 39) ||
29           (getNodeItem(&NodeList1, 4) > 36)){
30         //You Can't Escape
31     }
32     printf("You've Passed Assign 1\n");
33 }
```