

Unix Lab Assignment

GNU Project Debugger(GDB)

Name:G Saiteja

Rollno:422149

Sec: "A"

Code: (Finding circularity of a linked list)

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

```
#include <stdlib.h>
```

```
struct node {  
    int data;  
    struct node*next;  
    int visit;  
};
```

```
int init(struct node**head) {  
    *head=NULL;  
    return 1;  
}
```

```
int destroy(struct node**head) {  
    *head=NULL;  
    return 1;  
}
```

```
int insertcll(struct node**head,int data,int predata) {  
    struct node*newnode=(struct node*)malloc(sizeof(struct node));  
    if(newnode==NULL) {  
        return 0;  
    }
```

```

newnode->data=data;
struct node*ptr=*head;
if(*head==NULL) {
    *head=newnode;
    newnode->next=newnode;
}
while(ptr!=NULL && ptr->data!=predata) {
    ptr=NULL;
    ptr=ptr->next;
}
if(ptr==NULL) {
    return 0;
}
else {
    newnode->next=ptr->next;
    ptr->next=newnode;
    return 1;
}
}

```

```

int traverse(struct node*head) {
    if(head==NULL) {
        return 0;
    }
    struct node*ptr=head;
    printf("%d ",ptr->data);
    ptr=ptr->next;
    while (ptr!=NULL && ptr!=head) {
        printf("%d ",ptr->data);
        ptr=ptr->next;
    }
    printf("\n");
    return 1;
}

```

```

int isCircular(struct node*head) {
    if(head==NULL) {
        return 0;
    }
    struct node*ptr=head;
    while (ptr!=NULL && ptr->visit!=1) {
        ptr->visit=1;
        ptr=ptr->next;
    }
    if(ptr==NULL) {
        printf("It is not circular \n");
        return 0;
    }
    else {
        printf("It is circular \n");
        return 1;
    }
}

```

```

int main() {
    struct node* head =(struct node*)malloc(sizeof(struct node));
    init(&head);
    int n;
    int data1;
    int predata;
    printf("enter the no.of elements:");
    scanf("%d",&n);
    // inserting as circular linear linked lists
    for (int i=0;i<n;i++) {

        printf("enter the element:");
        scanf("%d",&data1);
        insertcll(&head,data1,predata);
        predata=data1;
    }
}

```

```

    traverse(head);
    isCircular(head);
    return 0;
}

```

Debugging:

```

student@ai-HP-ProDesk-600-G4-MT:~/9221409/unix$ gcc -g isCircular.c
student@ai-HP-ProDesk-600-G4-MT:~/9221409/unix$ gdb ./a.out
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04) 9.2
Copyright (C) 2020 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 ./a.out...
(gdb) run
Starting program: /home/student/9221409/unix/a.out
enter the no.of elements:3
enter the element:1
enter the element:2
enter the element:3

Program received signal SIGSEGV, Segmentation fault.
0x000055555555529f in insertcll (head=0x7fffffffde60, data=3,
    predata=2) at isCircular.c:32
32          ptr=ptr->next;
(gdb) list
27          *head=newnode;
28          newnode->next=newnode;
29      }
30      while(ptr!=NULL && ptr->data!=predata) {
31          ptr=NULL;
32          ptr=ptr->next;
33      }
34      if(ptr==NULL) {
35          return 0;
36      }
(gdb) list
37      else {
38          newnode->next=ptr->next;
39          ptr->next=newnode;
40          return 1;
41      }

```

```
Activities Terminal Mar 13 16:55 student@al-HP-ProDesk-600-G4-MT: ~/9221409/unix
(gdb) list
37         else {
38             newnode->next=ptr->next;
39             ptr->next=newnode;
40             return 1;
41         }
42     }
43
44     int traverse(struct node*head) {
45         if(head==NULL) {
46             return 0;
47         }
48         struct node*ptr=head;
49         printf("%d ",ptr->data);
50         ptr=ptr->next;
51         while (ptr!=NULL && ptr!=head) {
52             printf("%d ",ptr->data);
53             ptr=ptr->next;
54         }
55         printf("\n");
56         return 1;
57     }
58
59     int isCircular(struct node*head) {
60         if(head==NULL) {
61             return 0;
62         }
63         struct node*ptr=head;
64         while (ptr!=NULL && ptr->visit!=1) {
65             ptr->visit=1;
66             ptr=ptr->next;
67         }
68         if(ptr==NULL) {
69             printf("It is not circular \n");
70             return 0;
71         }
72         else {
73             printf("It is circular \n");
74             return 1;
75         }
76     }
77 }
(gdb)
```

```
Activities Terminal Mar 13 16:55 student@al-HP-ProDesk-600-G4-MT: ~/9221409/unix
73         printf("It is circular \n");
74         return 1;
75     }
76 }
(gdb)
77
78 int main() {
79     struct node* head =(struct node*)malloc(sizeof(struct node));
80     init(&head);
81     int n;
82     int data1;
83     int predata;
84     printf("enter the no.of elements:");
85     scanf("%d",&n);
86     // inserting as circular linear linked lists
(gdb)
87     for (int i=0;i<n;i++) {
88         printf("enter the element:");
89         scanf("%d",&data1);
90         insertc1l(&head,data1,predata);
91         predata=data1;
92     }
93     traverse(head);
94     isCircular(head);
95     return 0;
96 }
(gdb)
97 }
(gdb)
Line number 98 out of range; isCircular.c has 97 lines.
(gdb) break 91
Breakpoint 1 at 0x5555555548d: file isCircular.c, line 91.
(gdb) break 95
Undefined command: "break95". Try "help".
(gdb) break 95
Breakpoint 2 at 0x555555554bf: file isCircular.c, line 95.
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/student/9221409/unix/a.out
enter the no.of elements:3
enter the element:1
Breakpoint 1, main () at isCircular.c:91
91         insertc1l(&head,data1,predata);
(gdb) next
```

```
Activities Terminal Mar 13 16:55 student@ai-HP-ProDesk-600-G4-MT: ~/9221409/unix
Start it from the beginning? (y or n) y
Starting program: /home/student/9221409/unix/a.out
enter the no.of elements:3
enter the element:1

Breakpoint 1, main () at isCircular.c:91
91         insertc1l(&head,data1,predata);
(gdb) next
92         predata=data1;
(gdb) print head
$1 = (struct node *) 0x55555559ae0
(gdb) next
87         for (int i=0;i<n;i++) {
(gdb) print head->data
$2 = 1
(gdb) continue
Continuing.
enter the element:2

Breakpoint 1, main () at isCircular.c:91
91         insertc1l(&head,data1,predata);
(gdb) next
92         predata=data1;
(gdb) print head->next
$3 = (struct node *) 0x55555559b00
(gdb) print (head->next)->data
$4 = 2
(gdb) next
87         for (int i=0;i<n;i++) {
(gdb) next
89             printf("enter the element:");
(gdb) next
90             scanf("%d",&data1);
(gdb) next
enter the element:3

Breakpoint 1, main () at isCircular.c:91
91         insertc1l(&head,data1,predata);
(gdb) next

Program received signal SIGSEGV, Segmentation fault.
0x00005555555529f in insertc1l (head=0x7fffffffde00, data=3, predata=2)
    at isCircular.c:32
32         ptr=ptr->next;
(gdb) 
```

Segmentation fault in insertc1l line number : 32
ptr=ptr->next;