

Software Engineering



Name: Soham Das

Section: A1

Roll No: 002311001004

Assignment - 1

IT-UG2

Assignments-1 on GDB

Consider the program in the folder assigned.

First file - a.c

```
#include <f.h>
```

```
int f0(int *p)
{
    int x, cntr = 1;
    printf("Enter a number between 2 and 6 (non-inclusive): \n");
    scanf("%d", &x);
    while ((x <= 2) || (x >=6)) {
        printf("You have entered %d which is wrong.Please Reenter:\n",x);
        scanf("%d", &x);
        cntr++;
        if (cntr > 5) {
            printf("Max number of Invalid input reached.Program will QUIT...\n");
            return 1;
        }
    }
    printf("You have entered %d\n",x);
    *p = x;
    return 0;
}

int main()
{
    int cntr = 0 , i;
    if (f0(&cntr) != 0 )
        return 0;
    for(i =0 ;i < cntr; i++) {
        int a,b;
        a = 100 / (i+2);
        b = 327 / (i+2);
        f1(a,b);
        f2(&a,&b);
        printf("After operation %d ", i+1);
        f1(a,b);
    }
}
```

Second file - b.c

```
void f1(int x,int y)
{
    printf("The numbers are : ");
    printf("< %d, %d>\n",x,y);
}
void f2(int*p ,int *q)
{
    *p = (*p) + (*q);
    *q = (*p) -(*q);
    *p = (*p) - (*q);
}
```

Third file - f.h

```
#include<stdio.h>
extern void f1(int x, int y);
extern void f2(int *x, int *y);
```

a. Compile it so that it compiles with debugging symbols [using proper option]

Ans. gcc a.c b.c -g -o out -I./

```
[be2304@localhost assign1]$ gcc a.c b.c -g -o out -I./
```

b. Put breakpoint to function f1.

Ans. break f1

```
Reading symbols from /home/usr/student/ug/yr23/be2304/softwareEngg/Assignments/assign1/out...done.
(gdb) break f1
Breakpoint 1 at 0x400756: file b.c, line 4.
```

c. Put breakpoint to line 10 of b.c

Ans. break b.c:10

```
(gdb) break b.c:10
Breakpoint 2 at 0x40079e: file b.c, line 10.
(gdb) |
```

d. Run the program until it finishes. Which commands are you using to take it to completion?

Ans. run
continue

```
(gdb) run
Starting program: /home/usr/student/ug/yr23/be2304/softwareEngg/Assignments/assign1/./out
Enter a number between 2 and 6 (non-inclusive):
3
You have entered 3

Breakpoint 1, f1 (x=50, y=163) at b.c:4
4      printf("The numbers are : ");
Missing separate debuginfos, use: debuginfo-install glibc-2.17-157.el7_3.2.x86_64
(gdb) continue
Continuing.
The numbers are : < 50, 163>

Breakpoint 2, f2 (p=0x7fffffff194, q=0x7fffffff190) at b.c:10
10     *q = (*p) -(*q);
(gdb) continue
Continuing.

Breakpoint 1, f1 (x=163, y=50) at b.c:4
4      printf("The numbers are : ");
(gdb) continue
Continuing.
After operation 1 The numbers are : < 163, 50>

Breakpoint 1, f1 (x=33, y=109) at b.c:4
4      printf("The numbers are : ");
(gdb) continue
Continuing.
The numbers are : < 33, 109>

Breakpoint 2, f2 (p=0x7fffffff194, q=0x7fffffff190) at b.c:10
10     *q = (*p) -(*q);
(gdb) continue
Continuing.

Breakpoint 1, f1 (x=109, y=33) at b.c:4
4      printf("The numbers are : ");
(gdb) continue
Continuing.
After operation 2 The numbers are : < 109, 33>

Breakpoint 1, f1 (x=25, y=81) at b.c:4
4      printf("The numbers are : ");
(gdb) continue
Continuing.
The numbers are : < 25, 81>

Breakpoint 2, f2 (p=0x7fffffff194, q=0x7fffffff190) at b.c:10
10     *q = (*p) -(*q);
(gdb) continue
Continuing.

Breakpoint 1, f1 (x=81, y=25) at b.c:4
4      printf("The numbers are : ");
(gdb) continue
Continuing.
After operation 3 The numbers are : < 81, 25>
[Inferior 1 (process 31663) exited with code 03]
```

e. How many times breakpoint “1” is hit in one run of the program ?

Ans. Breakpoint 1 is hit 6 times.

f. How many times breakpoint “2” is hit in one run of the program

Ans. Breakpoint 2 is hit 3 times.

g. How can you see details about a breakpoint ?

Ans. info breakpoints 1

```
breakpoint already hit 5 times
(gdb) info breakpoints 1
Num      Type      Disp Enb Address      What
1        breakpoint keep y  0x0000000000400756 in f1 at b.c:4
        breakpoint already hit 6 times
(gdb) info breakpoints 2
Num      Type      Disp Enb Address      What
2        breakpoint keep y  0x000000000040079e in f2 at b.c:10
        breakpoint already hit 3 times
(gdb) |
```

h. How you can see details about all breakpoints ?

Ans. info breakpoints

```
[inferior 1 (process 31885) exited with code 03]
(gdb) info breakpoints
Num      Type      Disp Enb Address      What
1        breakpoint keep y  0x0000000000400756 in f1 at b.c:4
        breakpoint already hit 6 times
2        breakpoint keep y  0x000000000040079e in f2 at b.c:10
        breakpoint already hit 3 times
(gdb) |
```

i. What is the value of variable x in f1 when breakpoint “1” is hit for 3 rd time ? How can you examine it ?

Ans. 33

```

(gdb) run
Starting program: /home/usr/student/ug/yr23/be2304/softwareEngg/Assignments/assign1/./out
Enter a number between 2 and 6 (non-inclusive):
3
You have entered 3

Breakpoint 1, f1 (x=50, y=163) at b.c:4
4      printf("The numbers are : ");
(gdb) c
Continuing.
The numbers are : < 50, 163>

Breakpoint 2, f2 (p=0x7fffffff194, q=0x7fffffff190) at b.c:10
10     *q = (*p) -(*q);
(gdb) c
Continuing.

Breakpoint 1, f1 (x=163, y=50) at b.c:4
4      printf("The numbers are : ");
(gdb) c
Continuing.
After operation 1 The numbers are : < 163, 50>

Breakpoint 1, f1 (x=33, y=109) at b.c:4
4      printf("The numbers are : ");
(gdb) print x
$1 = 33

```

j. Rerun the program.put a breakpoint at function f0. list 5 lines where it has stopped with breakpoint 3 for first time.

Ans.

```

http://www.gnu.org/software/gdb/bugs/...
Reading symbols from /home/usr/student/ug/yr23/be2304/softwareEngg/Assignments/assign1/out...done.
(gdb) break f0
Breakpoint 1 at 0x4005f9: file a.c, line 6.
(gdb) run
Starting program: /home/usr/student/ug/yr23/be2304/softwareEngg/Assignments/assign1/./out

Breakpoint 1, f0 (p=0x7fffffff198) at a.c:6
6      int x, cnt = 1;
Missing separate debuginfos, use: debuginfo-install glibc-2.17-157.el7_3.2.x86_64
(gdb) list
1
2      #include "f.h"
3
4      int f0(int *p)
5      {
6          int x, cnt = 1;
7          printf("Enter a number between 2 and 6 (non-inclusive): \n");
8          scanf("%d", &x);
9          while ((x <= 2) || (x >= 6)) {
10             printf("You have entered %d which is wrong.Please Reenter:\n",x);
(gdb) set listsize 5
(gdb)
(gdb) list
11             scanf("%d", &x);
12             cnt++;
13             if (cnt > 5) {
14                 printf("Max number of Invalid input reached.Program will QUIT...\n");
15                 return 1;
(gdb) |

```

Explore : Complete this rerun. Now see what is the change in details of breakpoint s by using the command used in “h”.

```
(gdb)
(gdb) info breakpoints
Num      Type          Disp Enb Address                What
1        breakpoint    keep y   0x00000000000400756 in f1 at b.c:4
2        breakpoint    keep y   0x0000000000040079e in f2 at b.c:10
3        breakpoint    keep y   0x000000000004005f9 in f0 at a.c:6
breakpoint already hit 1 time
(gdb) |
```