**ASSIGNMENT – 1**

**SE LAB**

Name:Swapnadeep Mishra

Roll: 002211001115

1. **Consider the program in folder assign1**
2. Compile it so that it compiles with debugging symbols

Command:gcc –g a.c –o a

[be22115@localhost ~]$ gcc -g a.c -o a

[be22115@localhost ~]$ gdb a

GNU gdb (GDB) Red Hat Enterprise Linux 7.6.1-94.el7

Copyright (C) 2013 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-redhat-linux-gnu".

For bug reporting instructions, please see:

<http://www.gnu.org/software/gdb/bugs/>...

Reading symbols from /home/usr/student/ug/yr22/be22115/a...done.

**B. put breakpoint to function 1**

Command: b f1

(gdb) b f1

Breakpoint 1 at 0x4005fb: file b.c, line 4.

**C. put breakpoint to line 10 of b.c**

Command: b b.c:10

(gdb) b b.c:10

Breakpoint 2 at 0x400643: file b.c, line 10.

**D. Run the program until it finishes,which command are you using to take it to completion?**

Command: r(to start the execution of the program)

(gdb) r

Starting program: /home/usr/student/ug/yr22/be22115/a

Enter a number between 2 and 6 (non-inclusive):

4

You have entered 4

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) c

Continuing.

The numbers are : < 50, 163>

Breakpoint 2, f2 (p=0x7fffffffe314, q=0x7fffffffe310) 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) c

Continuing.

The numbers are : < 33, 109>

Breakpoint 2, f2 (p=0x7fffffffe314, q=0x7fffffffe310) at b.c:10

10 \*q = (\*p) -(\*q);

(gdb) c

Continuing.

Breakpoint 1, f1 (x=109, y=33) at b.c:4

4 printf("The numbers are : ");

(gdb) c

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) c

Continuing.

The numbers are : < 25, 81>

Breakpoint 2, f2 (p=0x7fffffffe314, q=0x7fffffffe310) at b.c:10

10 \*q = (\*p) -(\*q);

(gdb) c

Continuing.

Breakpoint 1, f1 (x=81, y=25) at b.c:4

4 printf("The numbers are : ");

(gdb) c

Continuing.

After operation 3 The numbers are : < 81, 25>

Breakpoint 1, f1 (x=20, y=65) at b.c:4

4 printf("The numbers are : ");

(gdb) c

Continuing.

The numbers are : < 20, 65>

Breakpoint 2, f2 (p=0x7fffffffe314, q=0x7fffffffe310) at b.c:10

10 \*q = (\*p) -(\*q);

(gdb) c

Continuing.

Breakpoint 1, f1 (x=65, y=20) at b.c:4

4 printf("The numbers are : ");

(gdb) c

Continuing.

After operation 4 The numbers are : < 65, 20>

[Inferior 1 (process 8186) exited with code 04]

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

8 times breakpoint 1 hit

Num Type Disp Enb Address What

1 breakpoint keep y 0x00000000004005fb in f1 at b.c:4

breakpoint already hit 8 times

2 breakpoint keep y 0x0000000000400643 in f2 at b.c:10

breakpoint already hit 4 times

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

4 times breakpoint 2 hit

Num Type Disp Enb Address What

1 breakpoint keep y 0x00000000004005fb in f1 at b.c:4

breakpoint already hit 8 times

2 breakpoint keep y 0x0000000000400643 in f2 at b.c:10

breakpoint already hit 4 times

**G. How you can see details about a breakpoint?**

Command:

info b 1

info b 2

(gdb) info b 1

Num Type Disp Enb Address What

1 breakpoint keep y 0x00000000004005fb in f1 at b.c:4

breakpoint already hit 8 times

(gdb) info b 2

Num Type Disp Enb Address What

2 breakpoint keep y 0x0000000000400643 in f2 at b.c:10

breakpoint already hit 4 times

**H. How you can see details about all breakpoints?**

Command: info break

(gdb) info break

Num Type Disp Enb Address What

1 breakpoint keep y 0x00000000004005fb in f1 at b.c:4

breakpoint already hit 8 times

2 breakpoint keep y 0x0000000000400643 in f2 at b.c:10

breakpoint already hit 4 times

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

Value of x is 33 when breakpoint hits 3rd time

(gdb) r

Starting program: /home/usr/student/ug/yr22/be22115/a

Enter a number between 2 and 6 (non-inclusive):

4

You have entered 4

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) c

Continuing.

The numbers are : < 50, 163>

Breakpoint 2, f2 (p=0x7fffffffe314, q=0x7fffffffe310) 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) c

Continuing.

The numbers are : < 33, 109>

Breakpoint 2, f2 (p=0x7fffffffe314, q=0x7fffffffe310) at b.c:10

10 \*q = (\*p) -(\*q);

(gdb) c

Continuing.

(gdb) r

Starting program: /home/usr/student/ug/yr22/be22115/a

Enter a number between 2 and 6 (non-inclusive):

4

You have entered 4

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=0x7fffffffe314, q=0x7fffffffe310) 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) p x

$1 = 33

(gdb)

**J. Rerun the program. Put a breakpoint at function f0. List 5 lines where it has stopped with breakpoint 3 for first time.**

Commands:

Put breakpoint: b f0

Rerun: r(rerun if require press y again)

C(keep executing until the finishing of the program)

List 5 lines:s (executed n times for n lines here 5 times)

Breakpoint 3, f0 (p=0x7fffffffe318) at a.c:6

6 int x, cntr = 1;

(gdb) s

7 printf("Enter a number between 2 and 6 (non-inclusive): \n");

(gdb) s

Enter a number between 2 and 6 (non-inclusive):

8 scanf("%d", &x);

(gdb) s

s

9 while ((x <= 2) || (x >=6)) {

(gdb) s

10 printf("You have entered %d which is wrong.Please Reenter:\n",x);

(gdb) s

You have entered 4196381 which is wrong.Please Reenter:

11 scanf("%d", &x);

(gdb) c

Continuing.

2)

Commands:

i)b f1

[be22115@localhost ~]$ gdb a

GNU gdb (GDB) Red Hat Enterprise Linux 7.6.1-94.el7

Copyright (C) 2013 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-redhat-linux-gnu".

For bug reporting instructions, please see:

<http://www.gnu.org/software/gdb/bugs/>...

Reading symbols from /home/usr/student/ug/yr22/be22115/a...done.

(gdb) b f1

Breakpoint 1 at 0x4005fb: file b.c, line 4.

(gdb) b b.c:10

Breakpoint 2 at 0x400643: file b.c, line 10.

(gdb) r

Starting program: /home/usr/student/ug/yr22/be22115/a

Enter a number between 2 and 6 (non-inclusive):

4

You have entered 4

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) c

Continuing.

The numbers are : < 50, 163>

Breakpoint 2, f2 (p=0x7fffffffe314, q=0x7fffffffe310) 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>

**ii)**

b b.c:10

[be22115@localhost ~]$ gdb a

GNU gdb (GDB) Red Hat Enterprise Linux 7.6.1-94.el7

Copyright (C) 2013 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-redhat-linux-gnu".

For bug reporting instructions, please see:

<http://www.gnu.org/software/gdb/bugs/>...

Reading symbols from /home/usr/student/ug/yr22/be22115/a...done.

(gdb) b f1

Breakpoint 1 at 0x4005fb: file b.c, line 4.

(gdb) b b.c:10

Breakpoint 2 at 0x400643: file b.c, line 10.

(gdb) r

Starting program: /home/usr/student/ug/yr22/be22115/a

Enter a number between 2 and 6 (non-inclusive):

4

You have entered 4

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) c

Continuing.

The numbers are : < 50, 163>

Breakpoint 2, f2 (p=0x7fffffffe314, q=0x7fffffffe310) 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>

**iii)**

**r c**

Breakpoint 1, f1 (x=25, y=81) at b.c:4

4 printf("The numbers are : ");

(gdb) c

Continuing.

The numbers are : < 25, 81>

Breakpoint 2, f2 (p=0x7fffffffe314, q=0x7fffffffe310) at b.c:10

10 \*q = (\*p) -(\*q);

(gdb) c

Continuing.

Breakpoint 1, f1 (x=81, y=25) at b.c:4

4 printf("The numbers are : ");

(gdb) c

Continuing.

After operation 3 The numbers are : < 81, 25>

Breakpoint 1, f1 (x=20, y=65) at b.c:4

4 printf("The numbers are : ");

(gdb) c

Continuing.

The numbers are : < 20, 65>

Breakpoint 2, f2 (p=0x7fffffffe314, q=0x7fffffffe310) at b.c:10

10 \*q = (\*p) -(\*q);

(gdb) c

Continuing.

Breakpoint 1, f1 (x=65, y=20) at b.c:4

4 printf("The numbers are : ");

(gdb) c

Continuing.

After operation 4 The numbers are : < 65, 20>

[Inferior 1 (process 8186) exited with code 04]

**iv)info b1**

(gdb) info b 1

Num Type Disp Enb Address What

1 breakpoint keep y 0x00000000004005fb in f1 at b.c:4

breakpoint already hit 8 times

**v)info break**