Software Engineering

Assignments-1 on gdb

Name - Anish Das

Department - Information Technology

Roll No - 002311001009

Section - A1

1. Consider the program in folder assign1

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

Ans: gcc -g a.c b.c -o prog

```
PS C:\Users\LENOVO\Downloads\Assignments\Assignments\assign1> gcc -g a.c b.c -o prog
b.c: In function 'f1':
b.c:4:5: warning: implicit declaration of function 'printf' [-Wimplicit-function-declaration]
     printf("The numbers are : ");
     ^~~~~~
b.c:4:5: warning: incompatible implicit declaration of built-in function 'printf'
b.c:4:5: note: include '<stdio.h>' or provide a declaration of 'printf'
PS C:\Users\LENOVO\Downloads\Assignments\Assignments\assign1> gdb prog
GNU gdb (GDB) 7.6.1
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 "mingw32".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from C:\Users\LENOVO\Downloads\Assignments\Assignments\assign1\prog.exe...done.
```

b> Put breakpoint to function f1.

Ans: break f1

```
(gdb) break f1
Breakpoint 1 at 0x4015d6: 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 0x401613: file b.c, line 10.
```

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

Ans: run , 4 (any no between 2 and 6 excluding both), continue , enter(till program is not being run)

```
(gdb) run
Starting program: C:\Users\LENOVO\Downloads\Assignments\Assignments\assign1/prog.exe
[New Thread 35260.0x6d14]
[New Thread 35260.0x8190]
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) continue
Continuing.
The numbers are : < 50, 163>

Breakpoint 2, f2 (p=0x61ff14, q=0x61ff10) at b.c:10
10          *q = (*p) -(*q);
(gdb)
Continuing.
After operation 1
Breakpoint 1, f1 (x=163, y=50) at b.c:4
4           printf("The numbers are : ");
(gdb)
Continuing.
The numbers are : < 163, 50>

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

Breakpoint 2, f2 (p=0x61ff14, q=0x61ff10) at b.c:10
10          *q = (*p) -(*q);
(gdb)
Continuing.
After operation 2
Breakpoint 1, f1 (x=109, y=33) at b.c:4
4           printf("The numbers are : ");
(gdb)
Continuing.
The numbers are : < 109, 33>
```

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

Breakpoint 2, f2 (p=0x61ff14, q=0x61ff10) at b.c:10
10          *q = (*p) -(*q);
(gdb)
Continuing.
After operation 3
Breakpoint 1, f1 (x=81, y=25) at b.c:4
4           printf("The numbers are : ");
(gdb)
Continuing.
The numbers are : < 81, 25>

Breakpoint 1, f1 (x=20, y=65) at b.c:4
4           printf("The numbers are : ");
(gdb)
Continuing.
The numbers are : < 20, 65>

Breakpoint 2, f2 (p=0x61ff14, q=0x61ff10) at b.c:10
10          *q = (*p) -(*q);
(gdb)
Continuing.
After operation 4
Breakpoint 1, f1 (x=65, y=20) at b.c:4
4           printf("The numbers are : ");
(gdb)
Continuing.
The numbers are : < 65, 20>
[Inferior 1 (process 35260) exited normally]
(gdb)
The program is not being run.
```

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

Ans: breakpoint 1 is hit “8” times.

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

Ans: breakpoint 2 is hit “4” times.

g> How you can see details about a breakpoint ?

Ans: info breakpoint <breakpoint number>

Example : info breakpoint 1

```
(gdb) info breakpoints 1
Num     Type           Disp Enb Address    What
1       breakpoint     keep y   0x004015d6 in f1 at b.c:4
        breakpoint already hit 8 times
```

h> How you can see details about all breakpoints ?

Ans : info breakpoints

```
(gdb) info breakpoints
Num     Type           Disp Enb Address    What
1       breakpoint     keep y   0x004015d6 in f1 at b.c:4
        breakpoint already hit 8 times
2       breakpoint     keep y   0x00401613 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 ?

Ans: run, 4 (any number between 2 and 6), c, enter, enter, print x

```
(gdb) run
Starting program: C:\Users\LENOVO\Downloads\Assignments\Assignments\assign1/prog.exe
[New Thread 4976.0x9c14]
[New Thread 4976.0x37f8]
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)
(gdb) c
Continuing.
The numbers are : < 50, 163>

Breakpoint 2, f2 (p=0x61ff14, q=0x61ff10) at b.c:10
10          *q = (*p) -(*q);
(gdb)
Continuing.
After operation 1
Breakpoint 1, f1 (x=163, y=50) at b.c:4
4           printf("The numbers are : ");
(gdb)
Continuing.
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 : break f0,list, set listsize 5

```
(gdb) break f0
Breakpoint 3 at 0x401466: file a.c, line 6.
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
error return ../../gdb-7.6.1/gdb/windows-nat.c:1275 was 5
Starting program: C:\Users\LENOVO\Downloads\Assignments\Assignments\assign1/prog.exe
[New Thread 40468.0x6b10]
[New Thread 40468.0x413c]

Breakpoint 3, f0 (p=0x61ff18) at a.c:6
6           int x, cntr = 1;
(gdb) list
1
2       #include "f.h"
3
4       int f0(int *p)
5       {
6           int x, cntr = 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) list
11              scanf("%d", &x);
12              cntr++;
13              if (cntr > 5) {
14                  printf("Max number of Invalid input reached.Program will QUIT...\n");
15                  return 1;
```

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

Ans: info breakpoints

```
(gdb) info breakpoints
Num     Type           Disp Enb Address    What
1       breakpoint     keep y   0x004015d6 in f1 at b.c:4
2       breakpoint     keep y   0x00401613 in f2 at b.c:10
3       breakpoint     keep y   0x00401466 in f0 at a.c:6
        breakpoint already hit 1 time
```