Software Engineering

Assignments-2 on gdb

Name - Anuska Nath

Department - Information Technology

Roll No - 002311001003

Section - A1

Assignments-2 on gdb

a> Consider the program in folder assign2.

The program code is:

```c
#include <stdio.h>


int main()
{
   int i,j,k;
   for ( i= 0 ; i< 10; i++)
    for ( j = 0; j< 200; j++)
        for (k = 0; k< 3000; k++)
        {
            int t1,t2;
            t1=i;
            t2=j;
            if ((k !=0) && (k%1000 == 0))
                printf ("you have reached [%d][%d][%d]-th iteraion \n",t1,t2,k);
        }
}

```

a. Put a breakpoint in 1st executable line of the innermost loop.

Ans> The first executable line in the innermost loop is line 12 in d.c (t1 = i;).
So the command is “break d.c:12” .

```
(gdb) break d.c:12
Breakpoint 1 at 0x400553: file d.c, line 12.
```

b. If you run and continue ,how many times it is supposed to stop at breakpoint 1?

Ans> breakpoint is hit for 10 times, 200 times and 3000 times for the 1st, 2nd and 3rd loop respectively.

So total no of times breakpoint 1 will be hit = 10*200*3000 = 6000000.

Command: run, c 6000000

```
(gdb) run
Starting program: /home/usr/student/ug/yr23/be2303/SE/assign2/prog

Breakpoint 1, main () at d.c:12
12                  t1=i;
Missing separate debuginfos, use: debuginfo-install glibc-2.17-157.el7_3.2.x86_64
(gdb) c 6000000
Will ignore next 5999999 crossings of breakpoint 1.  Continuing.
you have reached [0][0][1000]-th iteraion
you have reached [0][0][2000]-th iteraion
you have reached [0][1][1000]-th iteraion
you have reached [0][1][2000]-th iteraion
you have reached [0][2][1000]-th iteraion
you have reached [0][2][2000]-th iteraion
you have reached [0][3][1000]-th iteraion
```

•
•
•

```
you have reached [9][197][2000]-th iteraion
you have reached [9][198][1000]-th iteraion
you have reached [9][198][2000]-th iteraion
you have reached [9][199][1000]-th iteraion
you have reached [9][199][2000]-th iteraion
[Inferior 1 (process 11475) exited with code 0347]
(gdb)
```

```
(gdb) info breakpoint 1
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x0000000000400553 in main at d.c:12
        breakpoint already hit 6000000 times
(gdb)
```

c. How will you continue so that it stops at 1000th iteration of innermost loop ?

Ans> ignore <breakpoint Number> <no. of times>.

Command: ignore 1 1000

```
(gdb) ignore 1 1000
Will ignore next 1000 crossings of breakpoint 1.
(gdb) run
Starting program: /home/usr/student/ug/yr23/be2303/SE/assign2/prog

Breakpoint 1, main () at d.c:12
12                  t1=i;
(gdb) c
Continuing.
you have reached [0][0][1000]-th iteraion
```

d. How you can condition your breakpoint , so that the loop stops at every 1000th iteration of innermost loop ?

Ans> condition 1 k>0 && (k-999)%1000 == 0, run, continue or c, continue or c, info breakpoints

```
(gdb) condition 1 k>0 && (k-999)%1000 == 0
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/usr/student/ug/yr23/be2303/SE/assign2/prog

Breakpoint 1, main () at d.c:12
12                  t1=i;
(gdb) c
Continuing.
you have reached [0][0][1000]-th iteraion

Breakpoint 1, main () at d.c:12
12                  t1=i;
(gdb) c
Continuing.
you have reached [0][0][2000]-th iteraion

Breakpoint 1, main () at d.c:12
12                  t1=i;
(gdb) info breakpoints
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x0000000000400553 in main at d.c:12
        stop only if k>0 && (k-999)%1000 == 0
        breakpoint already hit 3 times
```

e. Put a breakpoint in the 1st line of outermost loop.

Ans> The first line of outermost loop is line 8.
Command: break d.c:8

```
(gdb) break d.c:8
Breakpoint 2 at 0x400541: file d.c, line 8.
```

f. Disable breakpoint “1”

Ans> disable breakpoint <breakpoint Number>
Command: disable breakpoint 1, info breakpoints

```
(gdb) disable breakpoint 1
(gdb) info breakpoints
Num     Type           Disp Enb Address            What
1       breakpoint     keep n   0x0000000000400553 in main at d.c:12
        stop only if k>0 && (k-999)%1000 == 0
        breakpoint already hit 3 times
2       breakpoint     keep y   0x0000000000400541 in main at d.c:8
```

g. Add a command to breakpoint 2 so that it prints the value of “i” at each hit.

Ans> Command 2, print i, end, run, continue

```
(gdb) command 2
Type commands for breakpoint(s) 2, one per line.
End with a line saying just "end".
>print i
>end
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/usr/student/ug/yr23/be2303/SE/assign2/prog

Breakpoint 2, main () at d.c:8
8           for ( j = 0; j< 200; j++)
$1 = 0
(gdb) c
Continuing.
you have reached [0][0][1000]-th iteraion
you have reached [0][0][2000]-th iteraion
you have reached [0][1][1000]-th iteraion
you have reached [0][1][2000]-th iteraion
you have reached [0][2][1000]-th iteraion
you have reached [0][2][2000]-th iteraion
```

-
-
-

```
you have reached [0][194][1000]-th iteraion
you have reached [0][194][2000]-th iteraion
you have reached [0][195][1000]-th iteraion
you have reached [0][195][2000]-th iteraion
you have reached [0][196][1000]-th iteraion
you have reached [0][196][2000]-th iteraion
you have reached [0][197][1000]-th iteraion
you have reached [0][197][2000]-th iteraion
you have reached [0][198][1000]-th iteraion
you have reached [0][198][2000]-th iteraion
you have reached [0][199][1000]-th iteraion
you have reached [0][199][2000]-th iteraion

Breakpoint 2, main () at d.c:8
8           for ( j = 0; j< 200; j++)
$2 = 1
```

h. Delete breakpoint 2.

Ans> Delete breakpoint <breakpoint number>

Command: delete breakpoint 2