

Name – Aarizkhan Shaikh
Roll No – 101075
Class – 1 (A4)
Topic – (LCA) PF Assignment 2

Assignment 2 – Debugging, Documentation & Pointers

1] Why debugger is needed?

Debugging tools (also known as debuggers) are used to detect coding errors at various stages of SDLC (Software Development Life Cycle). They are used to recreate the error conditions, then examine the program at that time to determine the root cause.

2] What is GDB?

GDB (GNU Debugger) is a powerful debugging tool. It allows you to explore inside your program while they are running and also see what happens when your program crashes. GDB works with executable files, which are binary files created during the compilation process.

3] How to prepare your program for debugging?

Steps to prepare your program for debugging –

1. Use this command – **gcc -o -g filename.exe filename.c**
2. Place **breakpoints** into functions using GDB for detailed analysis.
3. **Run** the program, perform single line inspection using **next & step** commands in GDB

4] Investigating File: **debugAssignment1.c**

Bug Type: Business Logic Error

- **FIXME - 1**

- On line 6, we can see in the function "**int maxOf()**" that there is a logic bug that stores the smaller integer's value into the bigger integers parameter i.e. "**int max**".
- I found this bug with the help of **gdb commands** such as **break, run, step, print, etc..**
- After **changing the less than** sign, the error is resolved!

- **FIXME - 2**

- On line 23, we can see in the **main function** that there are two missing variables in the "**printf()**" function that are **n1, n2**.
- I found this bug with the help of **gdb commands** such as **break, run, step, print, etc..**
- So we resolve the bug by placing the variables in the following format (n1 ,n2 , max) using VS Code Editor.

5] Investigating File: **debugAssignment2.c**

End Goal: Learn how pointers work!

Q1] **How to print variables X, Y, P?**


Ans - We'll open the program in **GDB**, then use the command "**break main**" to set a breakpoint. Then, we'll use the commands "**run, next**" to navigate through lines of code before printing the values of the given variables with the commands "**print x, print y, and print p.**"

Q2] Find Memory Address of x using pointer p.

Ans – We'll use the GDB command “**print &x**”,
Address of variable x is “**0x61ff14**”

Q3] Take a memory dump of x and dump 16 memory bytes. Take a note and put the 16 byte dump in your WORD file.

Ans – (gdb) **x/16xb &x**



0x61ff14:	0x0a	0x00	0x00	0x00	0x0a	0x00	0x00	0x00
0x61ff1c:	0x14	0xff	0x61	0x00	0x80	0x00	0x40	0x00

Q4] After *p = -10 is executed, take the same memory dump and put those 16 bytes in your WORD file.

Ans – (gdb) **x/16xb &x**



0x61ff14:	0xf6	0xff	0xff	0xff	0x0a	0x00	0x00	0x00
0x61ff1c:	0x14	0xff	0x61	0x00	0x80	0x00	0x40	0x00

Q5] Observations

Ans – Both memory dumps have different addresses because we changed the value in the same allocated memory assigned to the given variable "x."