Practice of Debugging via gdb

Homework Assignment #1

ELEC462-003: System Programming

(Instructor: Prof. Suh, Young-Kyoon)

Due: 13:59:59 pm, Thursday, March 16, 2023

In this assignment, you will practice code debugging with `gdb`. More specifically, you will learn how to find a bug(s) in C program via `gdb`. Finding the bug itself is not the main purpose of the assignment, so you are welcome to ask classmates for help finding them. Your main purpose here is to "understand" the functionality of `gdb`, and demonstrate that you are able to use it. Now let's start the debugging. Please keep the following instructions.

1. Get given code, `factorization_org.c`, from LMS.

`factorization_org.c`: C code to factorize for a given quadratic equation.

** Note that this code has some bugs. **

2. Compile `factorization_org.c` to create an executable `factorization_org`.

3. Run the executable.

```
yksuh@macan:~/courses/ELEC462/homeworks/hw1$ gcc -g -o factorization_org factorization_org.c -lm
yksuh@macan:~/courses/ELEC462/homeworks/hw1$ ./factorization_org
aX^2 + bX + c = 0
a: 7
b: -77
c: 126
Doesn't make sense
yksuh@macan:~/courses/ELEC462/homeworks/hw1$
```

Note that because of some bugs, "Doesn't make sense" is shown. **Hint**) There are "two" buggy lines.

3. Start `gdb` and turn on logging.

```
yksuh@macan:~/courses/ELEC462/homeworks/hw1$ gdb factorization_org
GNU gdb (Ubuntu 8.1.1-0ubuntu1) 8.1.1
Copyright (C) 2018 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-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from factorization_org...done.
(gdb) set logging file gdb.hw1.log
(gdb) set logging on
Copying output to gdb.hw1.log.
(gdb) set history filename gdb.hw1.history
(gdb) set history save
(gdb) run  // start your debugging from here
Starting program: /home/yksuh/courses/ELEC462/homeworks/hw1/factorization_org
aX^2 + bX + c = 0
a: 7
b:
```

// turn on logging

You can use *any* `gdb` command for the debugging. I require you to have to try these commands: ***b*** (for breakpoint), ***c*** (for continue), ***p*** (for print), ***n*** (single step), and ***where*** (for showing the current line). Any other command can be used as well.

It is important to note that `gdb` breaks before executing the line. If a line modified a variable, you would be looking at the state of variables before that variable was modified.

4. Based on your findings, correct the buggy code. Only two lines in `factorization_org.c` will be fixed. Don't touch any other line. Also, you are NOT allowed to add or delete any code. Rename the code to `factorization_fixed.c`. Once all the bugs are correctly fixed, then the updated code will be able to produce the following output:

```
yksuh@macan:~/courses/ELEC462/homeworks/hw1$ gcc -o factorization_fixed factorization_fixed.c -lm
yksuh@macan:~/courses/ELEC462/homeworks/hw1$ ./factorization_fixed
aX^2 + bX + c = 0
a: 7
b: -77
c: 126
7(x-9.00)(x-2.00) = 0
yksuh@macan:~/courses/ELEC462/homeworks/hw1$
```

5. Submit your `gdb` log and history files and fixed code: `gdb.hw1.log`, `gdb.hw1.history`, and `factorization_fixed.c`.

[Submission]
- Using your student ID, create a directory like: '`hw1_s`<*StudentID*>'. If your student number is '2022123456', your directory name is '`hw1_s2012123456`'.
- Put the log, history, and fixed code files into that directory.
- Zip the directory like '`hw1_s2022123456.zip`'.
- Upload the zip file into LMS.

Q&A

You can ask questions to TA or me by email.

Late Day Policy

All exercises are due at 13:59 pm on the assigned due date. A grading penalty will be applied to late assignments. Any assignment turned in late will be penalized 50% per late day.

Plagiarism

No plagiarism will be tolerated. Once determined, you'll get penalized severely.