COM S 413/513 Project 1: Understanding bugs

Learning Objectives

1. Understanding the concepts related to bugs
2. Understanding the challenge of bug understanding and the need of automatic tool support
3. Improving problem solving skills on code inspection and bug diagnosis
4. Gaining hands-on experiences on navigating through open source code and real-world bugs

Description

In this homework, you are going to exercise your code inspection skills to analyze and understand bugs. As a first step, you will download the dbg benchmark located here: https://dbgbench.github.io/ This benchmark documented a set of real-world bugs from unix utilities *find* and *grep*. Here are some guidance on how you can explore this website effectively:

1. You can start under "use and reproduce", and follow "Step-by-Step instructions". The website also lists a set of tutorial materials (Click "tutorial materials" under "setup and infrastructure") that can help you further understand the two programs.
2. Click "Data for find" and "Data for grep" on the website, you will find detailed explanations of the bugs, including the root causes, symptoms and the tests that can trigger the bugs. There are also a set of patches for each bug developed by different participants of the research as well as the patch from the developer. You can find further information about the benchmark following the paper "Where is the bug and how is it fixed? an experiment with practitioners" https://dl.acm.org/citation.cfm?id=3106255

In this assignment, you are going to study and reproduce the bug Find [find.dbcb10e9].

Here are the suggested steps to follow:

1. (10 pt) Study "simplified bug report" listed in the bug description and use the failure-inducing input to reproduce the bug. Create a report for your study, including the fields of "program, bugid, bug type, bug location, fault location, fault condition (assertion),

patch location, test input that triggers the bug". Note that there are multiple patches available for each bug. Please use a correct patch to analyze the patch location. The fault location can be different among different answers. We accept all the correct answers as long as the fault location and fault condition match and you should add explanations on why you use this fault location. The location can be expressed using directory/file names, function names, and line numbers. Also, the terminologies of bug and fault in the dbg benchmark may be different from the ones taught in our class. Please use the terminologies taught in our class.

   a. Fault location and fault condition (assertion): 4 pt
   b. The rest: 6 pt (1 pt each item)

2. (12 pt) Minimize the buggy program. Here you are asked to create an executable, as small as possible program that can reproduce the bug in the original program. Here are some possible approaches to address the goal: you can either 1) delete irrelevant statements in the original code, 2) select relevant statements from the original code, or 3) combine both. You are welcomed to use any tools you find or just create the program manually. The resulting faulty program should aim to be executable and as small as possible.
   a. Reproduce the bug in a debugger, e.g., gdb, providing the stack when the failure is triggered (2 pt)
   b. Print a trace (a sequence of calls) for the failure-inducing input (you can insert a printf to print the function names or using an instrumentation tool like pin) (2 pt)
   c. Make the reduced program executable (2 pt): turn in the screenshot
   d. Reproduce the same bug using the reduced program (2 pt). Please add explanations on why you believe your generated program reproduced the bug in the original program.
   e. The reduced program should be small (2 pt). Please add explanations on why you believe your generated program cannot be further minimized.
   f. Two sets of c/cpp files and their sizes (2 pt) : you will need to submit two sets of .c/.cpp files, the original buggy files and the files that contain the generated program. You will also report the sizes, in terms of the lines of code for the original program and for your generated program. Please use the tool I upload to the course website under the directory of *tools.*

Fun point: if you are interested to evaluate your ability of bug diagnosis, you can also fill the questionnaire located https://dbgbench.github.io/questionnaire.pdf Then you can compare your answers with the survey results the researchers had generated over a set of experienced participates in the paper.

Deliverables (22 pt)

Please zip the following files and submit the zipped file to canvas under the "project 1: understanding bugs" assignment.

1. A report (.pdf) that consists of all the answers for Q1 and Q2
2. Two sets of .c/.cpp files for Q2

The homework is due Sept 22, Thur 11:59pm. This project can be challenging if it is your first time to handle open source code and real-world bugs. Start early!

Appendix

Frequently used docker commands for your references

docker ps
docker images
docker attach ContainerID
docker run -ti -v ⟨pathInHost⟩:⟨pathInDocker⟩ ⟨Image⟩ *copy a file from host to container*
docker cp foo.txt ContainerID:/foo.txt
copy a file from Docker container to host
docker cp ContainerID:/foo.txt foo.txt
exit docker without kill it
ctrl+p, ctrl+q