

20875 Software Engineering – Assignment 2

Due Sunday, November 24th 2024, 23:59

The assignment consists in finding and fixing bugs in the SCIP project. Specifically, we are targeting its latest stable version, SCIP 9.1.1, which can be downloaded directly using this link:

<https://scipopt.org/download/release/scipoptsuite-9.1.1.tgz>

SCIP is an open-source library for solving optimization problems. For example, it can solve linear programming (LP) and mixed-integer programming (MIP) problems (just like GLPK). It is accompanied by an executable, `scip`, that allows one to read a problem instance from a file and solve it. The input file must be in one of multiple supported formats. The parsing code for some of the formats contains bugs. Your task is to find such bugs, explain them, and propose fixes.

Write your answers directly on Blackboard (under “Assignment 2”) by the end of November 24th.

Part I. First, we target the parsing code for the “MPS” format. Find a bug that, when SCIP is compiled with assertions disabled, causes a crash, a memory leak, or a wrong result. A bug that only causes an error message is not enough for full marks. However, when SCIP is compiled with assertions enabled, it is acceptable that the bug just triggers an assertion failure.

Q1. [1 mark] Create or find a file that triggers that bug when parsed as an MPS-formatted file. If this file is smaller than 100000 bytes, it is enough to upload it. Otherwise, describe the specificities of this file that make it trigger the bug.

Q2. [1 mark] Explain the *consequences* of the bug and where they happen in the source code of SCIP (in which file and at which line) when assertions are disabled.

Example: “A crash happens at line 1234 in `example/example.c` in the function `cleanup()` because the pointer variable `p` is `NULL` and the expression `p[i]` dereferences a `NULL` pointer.”

Q3. [2 marks] Determine the *causes* of the bug. Specifically, explain the chain of events that happens, starting from the particularities of the input file, and ending with the bug consequences described above.

Example: “When an index is negative in the input file, the allocation at `example/example2.c` line 5678 in the function `initialize()` attempts to allocate a negative amount, fails, and returns `NULL`. This `NULL` pointer is then stored in `struct DATA *d`, specifically in the `d->pointer` field. When `cleanup()` is called, that pointer is stored in `p`, yielding the `NULL` pointer dereference.”

Be as general as possible in your description of the range of circumstances that can lead to the bug. For example, “when an index is negative” is more general than “when an index is -1”.

Be concise. The objective is that a reader familiar with SCIP understands just enough to then propose a bug fix. Between 2 and 5 sentences should suffice.

Q4. [2 marks] Propose a *fix* for the bug. Explain your strategy to address the root cause of the problem. You can include proposed changes to the source code (using `diff -u` or `git diff`). As an alternative, detail your proposed changes in plain English.

The fix must be general, and correctly address the underlying problem.

Be concise. The objective is that a reader familiar with SCIP understands just enough to then apply your proposed code changes. Between 1 and 3 sentences should suffice.

Part II. Now, we target the parsing code for the “LP” format. Like in Question 1, find a bug that, when SCIP is compiled with assertions disabled, causes a crash, a memory leak, or a wrong result. A bug that only causes an error message is not enough for full marks. However, when SCIP is compiled with assertions enabled, it is acceptable that the bug just triggers an assertion failure.

Q5. [1 mark] Create or find a file that triggers a bug in SCIP when parsed as an LP file. If this file is smaller than 100000 bytes, it is enough to upload it. Otherwise, describe the specificities of this file that make it trigger the bug.

Q6. [1 mark] Explain the *consequences* of the bug and where they happen in the source code of SCIP when assertions are disabled. Same as Q2.

Q7. [2 marks] Determine the *causes* of the bug. Same as Q3.

Q8. [2 marks] Propose a *fix* for the bug. Same as Q4.

Part III: Bonus question.

Find a bug in any part of SCIP that causes a *crash* (as opposed to an assertion failure) *even when assertions are enabled*. Note: We have not found such a bug in the MPS or LP parsing code of SCIP, but (i) there may still be one, and (ii) you can target other parts of the SCIP code.

Q9. [2 bonus marks] Fully describe and fix this bug just like in Parts I and II.

Rules. This is an individual assignment. You are allowed (and encouraged) to talk about the assignment with your classmates. However, you must find input that triggers bugs yourself, on your own computer. You must also write your answers on your own. As a consequence, you will be able reproduce and explain all bugs upon request. **Please do not contact the developers of SCIP.**

Hints: There may be technical hurdles involved when compiling SCIP and looking for bugs. We will share as many hints as possible to overcome such issues, at this address:
<https://www.poirrier.ca/courses/softeng-2024/hw02/>