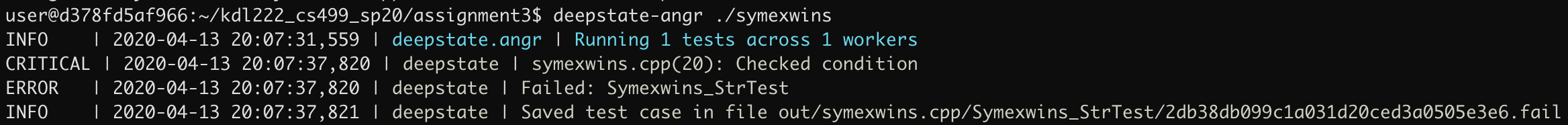
For this assignment, I started off with writing a program where symbolic execution wins. For symexwins.cpp, I used two integers generated by Deepstate and checked if 5 \* num1 + 3 == num2. I chose this equality because it will take longer for fuzzers to generate input that causes this to be true. As for symbolic execution, it should be very quick since it is a very small math problem to solve. First, I ran deepstate-angr to see how symbolic execution would do, and my intuition was correct. Deepstate-angr ran for about 6 milliseconds, as seen by the output below:



The input that angr found was num1=1 and num2=8.

Next, I tried to run AFL. Again, my intuition was correct. It took AFL about 4 minutes to find two numbers that caused the assert to fail, even though I let it run for 10 minutes. The input that AFL found was num1=-1 and num=-2. The output can be seen below:

A black and silver text

Description automatically generated

Next, I let LibFuzzer run for 18 minutes and gave it a corpus directory to save the failed input. In that time, it found 2 failing inputs:

* Num1=721420287, num2=-16777216
* Num1=-6710887, num2=-33554432

I am not 100% sure when the first failure was found, but definitely took longer than angr to find.

Finally, I ran the executable with –fuzz. I expect this fuzzer to take longer to find an input that causes the test to fail because of the fuzzer being very “basic” and not as “smart” when it generates random input. Again, my inuition was correct. 820913 tests were ran but all of them passed. I am not surprised that no failures were found since it would be difficult to make the test fail accidentally and not generating “smarter” input. The output can be seen below:

A picture containing computer

Description automatically generated