Kyle Zalewski

CS202

11/2/2018

# Programming Project 2 Debugger Analysis

For this assignment, gdb was a very useful tool in tracking down some strange errors I received while testing my code. I was able to track down the exact function that was causing a fault and print variable and object contents to see where data was being accessed that wasn’t available. Since this assignment required a lot of functions calling other functions, this ability was indispensable.

I ran into a strange issue in gdb where I was reaching a segmentation fault when a display function was called after removal of a node. The removal function appeared successful, but in certain test cases, the list display function would fault when moving to the abstract base class function. I unfortunately was not able to trace the reason for the fault, because using gdb to backtrace showed a function call with a strange (read: not real) name. I unfortunately had to omit some functionality to remove the segmentation fault. Given more time, I would hopefully be able to find the source of the fault, but my current knowledge of gdb doesn’t give me any clues as to how.

Valgrind was useful for tracking down memory leaks, as there were several parts of my code that used dynamic memory. I learned to more effectively use the leak test with the full flag to track down the exact line of code that was causing the leak.