Kyle Zalewski

CS202

Debugger Writeup

GDB turned out to be a valuable asset when troubleshooting my code during testing. I was able to quickly track down troublesome functions and fix them with the use of the text user interface function of the debugger. This allowed me to visually step through each portion of my code and follow the flow of calls through the various classes. This made short work of locating the origin of segmentation faults (of which I ended up having several chained together at one point of my testing). Once I saw which functions were being called while assembling my data structures, I was able to ensure that those were correct and change them if they weren’t. In very short order I had a program that was behaving exactly as expected.

Valgrind was useful in finding and eliminating memory leaks throughout the process. Using the –leak-check=full option gave me specific insight into the exact lines of code that were the start of a memory leak (as there was no corresponding delete). This helped me to optimize my destructors and remove the leaks. There ended up being a couple that I was unable to track down, but the tool helped me to remove the vast majority. In the future I will dig deeper and ensure that all memory leaks are removed completely.