// Keaton Spiller

// CS302

// 02/02/2024 Winter 2024

// Program # 1

Debugger Writeup

One particular case why the debugger was helpful was when I was deleting next in the Nodes deconstructor. Gdb was able to show the program infinitely jumping back and forth between deconstructor (Node::~Node()) and delete next, and I was able to see where the function was breaking. Gdb is crucial because if we try to debug complex problems with cout statements we can sometimes be tricked by the buffer, seeing things that aren’t apart of where the program broke.

Being able to track what is stored in each variable as it’s flowing through the code is crucial to understanding the error in the code we think we have written, and how the actual code is navigating through the logic.

I mostly knew how to use tmux and gef to walk through the code, but the basic gdb layout src does a great job at simplifying the code in ways that are overcomplicated with the tools I’m familiar with implementing.

Setting Breakpoint into the functions I’m interested in and walking through the code with next, or step, printing the variables along the way have been pretty useful to visualize the code. I still get lost in the wrong functions as I’m debugging but I’m making a lot of improvement with understanding how to navigate and trace when the error has occurred.