

Kunio Comey

CS202

GDB Writeup 1

Effectiveness of Debugger:

GDB helped with specifying the location of a segmentation fault. Without GDB, figuring out where the segmentation fault occurred in memory is difficult unless you set up a lot of cout statements throughout the program. It allowed less complexity on the actual program code and automatically did the jobs I would normally setup with output statements. For all memory problems, GDB works great.

Problems GDB helped solve:

Like I said above, GDB helped with solving problems with going into memory that didn't belong to me. It was a huge help for display functions inside of classes. GDB would show all of the functions called prior to the segmentation fault to help locate the problem code. This really helps with the programming process since I don't need to set up output statements to handle errors all throughout class function implementations.

Enhances Programming Experience?

I definitely learned to appreciate how much easier it is to debug GDB. The biggest problem with debugging without GDB in a complex class hierarchy would be figuring out where the error occurred in the hierarchy. If I had 4 classes in a hierarchy, I would have to look through 4 implementation files to solve it.

Feature to learn in the future

I would like to understand the output messages from GDB more so I wouldn't have to spend so much time debugging code. I would also like to learn how to use the GUI version of GDB because it seems a little more straight forward in finding where the errors in code occur.