CS202

GDB Writeup

Program #2

Hyunchan Kim

During this assignment, I got to use gdb again. Basically, gdb helped in debugging memory leaks and semantically checking the program for behavioral correctness whenever I got errors and unexpected results. Running the program and running the stack backtrace helped in finding the memory leaks and segmentation faults. Further stepping through main to find the semantic errors in the program was considerably more difficult in order to find the break points to set. A particular area where gdb was used was when the insertion was being done in the BST to check the balance of the tree. This processed was used to modify how the data was sent into the tree in particular modification of the mapping function to be more random inorder to match the data. gdb was used to step into some of the insert and remove functions of the data structures to find whether a memory leak had occured during the operation. In particular, the commands used were the step and next functions. The code breaking was found with using the debugger in the graphical code interface using the tui command. The memory leaks in the program were checked using valgrind, since gdb has no functionality to do so. And moreover, break point and print functions were use a lot when verifying errors at the exact point and the correct values and pointers were using. This is probably the most common functionality I get to use in the future.