

Lab 6: Debugging with GDB

Due: April 17th 11:59pm

Assignment

Help! I had to write a Collection library and accompanying `main()` routine to demo the functionality. Unfortunately, I was up all night with a sick kid and my brain ended up fried. As a result my precious code is riddled with errors and I don't have time to track them down! Your job is to fix the mistakes in my `collection.c` and `driver.c` files so that the demo runs flawlessly. Use GDB to track down the problems. The `output.txt` file demonstrates what the output of the demo program should look like once you have tracked down all of the bugs. Oh. And another thing. I think there might be a memory leak. Or two. Whoops. You will also need to figure out where any memory leaks might be and fix those too. Use the provided `makefile` to compile the program. The demo code is compiled into an executable named `demo`. Once you have it running, you might also want to use GDB to step through the code and double check that everything is correct even if the output matches what is in the sample file. Just to be sure.. cause you never know. Please include a file named `changes.txt` where you enumerate the changes/fixes you made and what error the change addressed. To start GDB with the executable run:

```
gdb ./demo
```

To give you an idea of what the collection library does, it defines a data type called `Collection` and the set of operations that can be performed on the `Collection`. Check out `collection.h` to see the data type definitions and the function prototypes. The `Collection` object maintains a doubly-linked list (Each element in the collection maintains a pointer to both the previous element and the next element in the `Collection`). New elements are added to the end of the list.

Evaluation

- +15 submission compiles
- +20 submission runs to completion
- +50 submission runs and produces correct output
- +10 "hidden" problems not obvious to comparing to `output.txt` are corrected
- +5 memory leaks have been abolished

Hints & Tips

- GDB does not directly tell you about memory leaks, however, it does provide the tools needed to make well-educated guesses as to where they might be.
- Use GDB to your advantage. The lab can be completed with relatively minor code changes if you take the time to experiment with the tools GDB provides.