

CSI 3336 Systems Programming

Homework 6

We have several tools available to help us develop sound C code. This homework introduces you to several of these tools. This is not an exhaustive list of tools available, but it does introduce you to a good set of tools that you can use throughout the rest of the semester.

1. Work through the Valgrind tutorial found at <http://cs.baylor.edu/~donahoo/tools/valgrind/>. Print the source code used in the tutorial for submittal. Execute your executable using Valgrind and redirect the output to file named valTutorial.txt. Print valTutorial.txt for submittal.
2. Work through the GDB tutorial found at <http://cs.baylor.edu/~donahoo/tools/gdb/tutorial.html>. Printout the corrected code for broken.cpp and core.cpp. Notice that there are a few inconsistencies within the core dump tutorial. First, you will most likely need to use the ulimit command to store a core dump (ulimit -c unlimited - will set the maximum size of a core dump to unlimited size). Second, once a core dump is created you will need to list the directory to find the proper name of the core file (my file was named "core.*" where * is some numerical value).
3. Work through the Electric Fence tutorial found at <http://cs.baylor.edu/~donahoo/tools/efence/>. Turn in a printout of the code used in the example along with the redirected output.
4. Work through the RCS tutorial found at <http://cs.baylor.edu/~donahoo/tools/rcs/>. Do a directory listing redirecting output to a file named rcsTutorial.txt. Print this file to turn in.
5. Create a makefile for the program in the next question. Make sure to use at least one macro and provide a clean target.
6. Write a linked list that stores characters. Create the files list.h, list.c, and driver.c. Write functions that add at the head, remove from the head, print the list, print the list in reverse order, and destroys of the list. Make sure to write a driver that demonstrates your linked list working properly.