

CS220 - Computer System II
Project 2

Due: 11/01/2017, 11:59pm

1 Introduction

In this project, you will implement a student database system using linked lists. You are allowed to work in groups of no more than 2 people. Register the groups using the Github Classroom group feature. *Do not contact the TAs asking for group matchmaking.*

2 Goal

The goal of this project is to implement a shared library that contains the functionality to manage a student database system. Your library will support several functions including:

- Addition and removal of students.
- Addition and removal of courses for students.
- Writing and reading database information to files in text format.
- Writing and reading database information to files in binary format.

3 Included files

You are provided with the following files:

parser.h A header which contains the required structures and the prototypes for all the exported functions. You will implement the listed functions.

libparser.a A static library that contains the implementation. You may use it to test for expected output for different inputs. It has been tested on remote.cs.binghamton.edu. It may or not work on other machines.

driver.c A sample driver program that invokes functions in the library.

sample_input.txt A sample text input file.

output.txt An output text file generated by the driver.

output.bin A binary output file generated by the driver. Run it using hexdump on remote to see its contents.

4 Deliverables

You are to provide the following:

`parser.c` Well commented source file. You may also add other source files as desired.

Makefile that generates the shared library `libparser.so`, the driver program `driver` using `libparser.so`, as well as any other targets you desire. A framework has been provided in the repository.

README.md Lists team members' information, and provides a description of how the responsibilities were divided between group members including exact line numbers that were coded by individual members.

NOTE: You may use the following functions for your implementation. You may petition to have other functions included into this list if you find a need.

- The `printf` family.
- the `scanf` family.
- File manipulation functions `fopen`, `fclose`, `fread` and `fwrite`.
- String utility functions `strlen`, `strstr`, `strdup`, `strcmp`.
- Memory fill function `memset`.
- The `fgets` family.
- The `malloc` family.
- String/Numeric conversion functions `atoi`, `atof`, etc.

5 Some useful tips

- Maintain a global list of students and update it as and when students are added/removed.
- Although you are required to implement the functions in `parser.h`, implement additional functions that make your job easier. Declare these functions as static.

- Use a state machine to keep track of parsing. Change state as and when you parse a line.
- Most importantly, start early and ask for help when you need it!

6 Submitting the result

Create `proj2.tar.gz` comprising of `parser.h`, `parser.c` and the Makefile to build the program. Submit it on mycourses.