ASSIGNMENT 4 WRITEUP

Chris Moon

February 2023

1 WHAT I LEARNED

STRUCTS AND ADTS

-structs are a little like objects or classes in other programming languages: they have a definition and can include values inside of them

-for example, I could create a "bucket" struct, with values for material and volume.

-structs include a constructor, and can include other, helper functions.

-the constructor is used to declare a specific instance of a struct, similar to how one might declare an int variable, as well as allocate memory for the struct

-Struct name = structconstructor(parameters); vs int x = 3

-structs can be used to create data types, like matrices, that don't exist natively in C.

-in order to use struct values inside the struct's source file, you can use arrow syntax.

-EX: structname-¿ value

-however, outside of the source file, this isn't valid, and in order to access these values, you need to use a helper fuction.

-to use a struct in another file, include the header to its source file.

-a matrix can be represented as an array of arrays in C, which is achievable by using a double pointer

FILES

-Access files using pointers to them.

-EX: FILE *filename;

-files have to be opened before any writing or reading, using open("file.txt", "r/w");

-"r" = read, "w" = write

-using optarg, and getopt, you can have the user input the name of a file

-EX: FILE *file = optarg;

-fscanf() works like a reverse printf: it looks for a pattern first

-It can be used to access values from a file.

-Files can be written to by using the fprintf() function.

-files have to be closed, similar to how memory has to be freed, at the end of the program.