Vadim Torgashov

I picked the archiving final project.

Identify the Problem: The problem is being able to go through a directory and writing all the metadata and data to a file correctly. Assuring that the metadata is written to the right place in the file and its actual data is correctly pointed to. The other problem will be creating a program that will go through the metadata and reconstruct the directory with files and subdirectories being place in the right spots.

Identify strategies: With C I can easily create a recursive function that will go through a directory and its subdirectories and list all the files. The function has a loop that simply goes through the directory contents and if it encounters another directory it will cause the function to be called for that directory. If its just a file then the function is called and instead of looping it will simply store the metadata and actual data based on the offsets it is given. The metadata for a file is length of name, name, offset to the data, and size of data. When the loop is done the file’s cursor has to be moved back and the directory’s meta data has to be written: number of files and subdirectories, name length and name.

Evaluate solution: I already have the recursion part done and the code to read back the metadata so its completely feasible. The reasoning for going through the directory and writing to the file in one function is to cut down on looping. If the metadata was simply saved to a linked list of structs I would then have to make another function to go through the list and write that data to file.

Evaluate outcome: the program does work as intended. There is one edge case I did not consider when archiving: having two files with the same name. The metadata is written first and then the program looks through the folders again and matches it with the metadata that was written to write the actual file data. When relooking at the folders it compares it to the file name in the metadata but doesn’t account for its directory. So sending the path of the file ang making sure it matches is how I could fix that edge case. One improvement to make is instead of adding name length I could instead add a null byte at the end of the name. In the format I have every other piece of metadata is needed.