

1 Design

My program is a state machine which uses getopt to parse through commands. When it reads the "q" option, it puts itself in APPEND mode. This causes it to iterate through the further arguments to the program and parse them as input files. When it reads the "x" option, it puts itself in EXTRACT mode. This causes it to read the arfile argument and create files from that archive. Etc etc etc.

2 Worklog

July 11th 10:00am - 11:00am: Read through assignment details, coded skeleton and getopt stuff.

July 11th 8:00pm - 9:00pm: Got getopt stuff to work, began with code to read in files and create an archive file.

July 12th 8:00am - 11:00am: Got program to successfully read in files and save them to a custom archive file (NOT the ar-specified archive file format)

July 13th 2:00pm - 4:00pm: Got program to successfully read in files and save them to an archive file somewhat reminiscent of the ar-specified format.

July 13th 7:30pm - 12:00am: Got program to successfully read in files, save them to the ar-specified format, and print their contents kind of. Still lots to do.

July 17th 3:00pm - 1:00am: Fixed bugs in outputting archive file properly (newline-padding odd outputs). Spent a long time debugging delete option problem.

July 18th 4:00pm-8:00pm: Fixed delete option problem. Checked for errors, fixed minor ones. Finished worklog.

3 Challenges

It was difficult for me to figure out how the ar-specified file format worked, until I thought to Google it. I was surprised that there was a good wikipedia page about it.

I also spent a considerable amount of time on the delete code. I was having problems with my pointer references and knowing where I was in the file. I spent an overly long amount of time trying to fix a bug, then eventually rewrote the entire section of code in about 20 minutes. I should have done that sooner. I think that for me, it is much easier to rewrite code that involves strings and pointers than to debug existing code.

4 Questions

Point of assignment: To get us used to the level of difficulty of programming assignments, to give an introduction to system programming and file IO. To show how complicated even simple Linux tools are.

Correctness testing: By running the ar command and checking for exact duplication of behavior. Also by checking to see if output from myar could be run with ar and vice versa. I ran most of my tests with simple 3-character .txt files. The program works with more complicated text files; I can't imagine it working differently for larger files than 3-character ones.

Learn: I learned next time I need to start earlier. Also I learned lots about IO and string manipulation.