CS311 Project 1: Unix File I/O

Ryley Herrington January 19, 2012

1 Introduction

This project was supposed to copy an arbitrary file from one location to another. It also checks for an existing file before you overwrite and if one exists then it will prompt the user for comfirmation. Fairly simple, but in the following few paragraphs I will explain the code, and provide a plot of runtimes with different buffersizes.

2 Code Explanation

Besides the comments in the code, this program is simple to explain. It uses one external method outside of main called err_and_exit which basically prints out an error message. The main reason being that I have enough error checking that I would have too much code duplicated all over the place. I did pass in stderr though, that way we knew what made the code "blow up."

Inside the main portion of the program, we first check to see how many arguments were called for the program. If it's not the correct amount, we print out an error message with the usage. Then we set the parameters from the input as source and destination files and set our block size too (the third parameter).

We try and open the source file, and if we can't we throw an error. Then we check and see if the destination file is already there, and if it is then asks to overwrite the destination file.

We set flags and permissions as variables. Open the destination file descriptor, malloc space enough to make a buffer the size of your block size, and then we hit the while loop.

The loop will run as long as there is more to read in the loop. It copies it a little bit at a time, however many bytes you specified in the parameters, and writes it out to the destination file that many bytes as well.

Then we finally free the buffer, and close our destination and source files. That's all there is to it.

3 Revision Control Log

Revision 1.5- 2012/01/20 05:47 lines: +13 -9. This revision got the access bug fixed, and added the dialog to overwrite a file.

Revision 1.4 date: 2012/01/19 23:55:41; lines: +29 -18. Finally got segfault errors out due to wrong gcc, and got rid of extraneous for loop.

Revision 1.3 date: 2012/01/16 00:24:42; lines: +4 -3. More error checking, fixed read bug.

Revision 1.2 date: 2012/01/15 22:44:15; lines: +24 -16. Added most error checking, got rid of fopens because I forgot we used sys calls instead.

Revision 1.1 date: 2012/01/15 22:13:03; Initial revision.

4 Time Trials

This was interesting to see; for some reason I expected a bell curve and it was not a bell curve at all. If my copy size was 1 byte at a time, it took somewhere between 45 seconds and 1 minutes. As my buffer grew though the time became faster and faster and faster. The graph shows the speed versus byte

size, and is scaled on a semi-log plot. It shows that it continually gets faster and faster, not as I originally expected. See graph: