Jonathan Armstrong

Analysis and Design

Analysis

The wash program utilizes the entering of user input from the command line in order to perform file I/O functions based on the letter entered. Due to C not being an object oriented language, a chain of if statements was necessary to sift through each possible option (all 11 different letters) and the most logical structure for each command was for it to have its own method.

The list function (c command) prints the contents of the file to the screen.

The copy function (d command) opens the file being copied for reading and the file to be copied into for writing and writes to the write file until the read file has no more bytes left to be read.

The giveName function (r command) prompts the user for a new file name and replaces the old name with it.

The delete function (d command) unlinks the file using unlink().

The zero function (t command) deletes the file and then creates a new file with the same name, effectively truncating it.

The append function (a command) performs the same actions as copy except the file being written to keeps its content.

The display function (l command) opens the file for reading then lseeks to the end of the file -100 (so 100 bytes from the end) and prints the contents to the end.

The chPermission function (m command) prompts the user what to change the permissions to and changes the permissions with chmod.

The chTime function (x command) sets the file’s access time to the current time.

The next function (n command) moves to the next file that can be processed (given at the start of program) and queries the user for commands.

The quit function (q command) exits the program.

Design

To build a program that performs file I/O operations based on user input we need initial files to work with, so the user must enter those when running the program. The commands entered will be performed on the first file entered unless the user enters the ‘next’ command or quits. The user can enter any number of commands before quitting because the chain of if statements that determine which command to issue is within an infinite while loop that only exits with a quit (q command).

* list
  + Create a new FILE file
  + **If** file is null
    - print error
  + **Else**
    - Set c to fgetc(file)
    - While there is more file
      * Print more file
* copy
  + declare variables for reading and writing descriptors, a buffer, a pointer, and number of chars to be read and written
  + Prompt user for the name of file to be copied to
  + **if** there is an error with opening the file to be copied
  + **if** there is an error with opening the file to be copied to
  + **while** 1
    - **if** there is more to be read
      * set pointer equal to buffer
      * **while** there is more to read
        + **if** there is an error with writing

print error

* + - * + Subtract the number of chars written from chars to be read
        + Add the number of chars written to the pointer
    - **else if** there is no more to be read
      * **break**

**else**

* + - * print error
* giveName
  + prompt user for new name
  + use rename function to set the file’s new name
* delete
  + unlink the current file using unlink function
* zero
  + Save the file’s name to a variable
  + Delete the file
  + Create a new file with the same name
* Append
  + Create two FILEs
  + Prompt user for destination of the append
  + Open file 1 for reading
  + Open file 2 for appending
  + **If** file 1 is NULL
    - Print error
  + **Else**
    - **While** there is more file to be appended
      * Append more file
  + close the appended file
* display
  + **if** there is an error with open
    - print error
  + use lseek to go to the end of the file -100
  + read the 100 bytes to the buffer
  + print the buffer
* chPermission
  + prompt user for what mode to change it to
  + use chmod to change the permission of the file
* chTime
  + create a struct utimbuf utimeStruct
  + set actime to 0
  + if there is an error with changing the time
    - print error
* next
  + Set the currentFile string equal to the next file being held in args[]
* Quit
  + Exit the program