FileSort.c

AUTHORS

Rakshay Kanthadai

Krishna Prajapati

NAME

filesort.c - c file

a.out - compiled binary executable

DESCRIPTION

The filesort program creates a binary executable that takes in a flag and a text file consisting of either strings or integers separated by commas and sorts the file with either a quicksort or insertion sort algorithm specified by the flag. White space in the text file is automatically removed (this includes spaces, tabs, and newlines). Function does all sorting through linked lists and tokenizing through the use of linked lists and prints out the final sorted linked list of tokens. Function also frees all use of heap memory prior to termination.

USE

```
gcc filesort.c
```

```
./a.out <flag> <filePath>
```

<flag> - a flag that indicates which sorting algorithm to use. Must be either "-q" or "-i" indicating quicksort or insertion sort respectively

<filePath> - an input file path to a file that contains either strings or integers to be sorted (cannot consist of both strings and integers). Each token must be comma-separated. Whitespace is ignored.

ERRORS

Fatal Errors: These are errors that will kill the program.

1. Not enough arguments- program will terminate and print a Fatal Error statement if the number of arguments is incorrect.

2. Not a valid flag - program will terminate and print a Fatal Error statement if the sorting flag is not "-q" or "-i".

3. Not a valid file - program will terminate and print a Fatal Error statement if the file cannot be opened/found. Check errno for more information.

4. Not enough space - program will terminate and print a Fatal Error statement if there is not enough memory space on the heap to allocate an initial linked list struct for even the first token or prior to accepting tokens.

5. Read error - program will terminate and print a Fatal Error statement if there is a problem reading the file. Check errno for more information.

6. Quicksort memory error - program will terminate and print a Fatal Error statement if there is not enough memory to allocate in the quicksort method.

Errors: These are errors that are unexpected but will not necessarily end the program

1. Not enough space - program will print an Error statement if there is not enough memory to store all elements from the file. Program will attempt to sort with any currently completed tokens.

Warnings: These are clarifications for the user in cases that seem incorrect, but the program can still run

1. Empty file - program will print out a warning if it recognizes that a file is empty or entirely composed of whitespace characters.

2. No non-empty tokens - program will print out a warning if it recognizes that a file only contains commas (and possibly whitespace characters).

RETURN VALUE

Program will print out the content that was read in alphabetical order if the input contained Strings or ascending order if the input contained integers. Each token will be printed on a new line.

NOTES

File is allowed to have empty tokens. If the program detects that Strings are being sorted, then these empty tokens will show up as new lines. If the program detects that integers are being sorted, then these empty tokens are converted to 0. If the file is composed of entirely empty tokens, it will treat them as strings.