

Problem

Code a program, (WAL), to manage a list of Web addresses for Netscape Navigator[®] bookmark files.

Discussion

This program will allow a user to create and manage lists of bookmark folders and Uniform Resource Locators, (URLs), for an on-line Web Address Management system. In addition to allowing a user to create new entries, the program will also allow a user to perform management operations upon folder and URL information of the Web address lists.

The interface for this program will be simple one-line text menus. A user chooses an operation by entering the first letter of the operation. After the operation completes a menu line is then (re)displayed. The initial menu line will provide a user with the following operations:

WAL: (N)ew, (O)pen, or (Q)uit?

The New operation allows the user to enter the name of an existing Netscape Navigator[®] bookmark file. (See program 1, "Web Address Checker", specifications for a description of the Netscape Navigator[®] bookmark file format.) The bookmark file is read, an error report is produced and the WAL list(s) is built, (see the Data Structures section below). This involves building a double-linked list(s) of all of the folder and URL data derived from the bookmark file, ordered upon the entry indexes. After the WAL linked-list(s) is built, the second menu line below is displayed. The Open operation allows a user to enter an existing WAL bookmark list file name, (for a discussion of the WAL bookmark list file format see the Input section below). After the WAL bookmark list file information has been read, it is automatically opened and the second menu line displayed. The Quit operation queries the user to be sure they wish to leave the system, (basically to annoy them ☺), and then issues a brief termination message before exiting. The second menu line(s) below will provide a user with the following operations:

WAL Folder: *folder name*

(A)dd, (D)elete, (M)odify, (L)ist, (N)avigate, (S)tate, (V)iew or (C)lose?

WAL URLs: *folder name*

(A)dd, (D)elete, (M)odify, (L)ist, (S)tate, (V)iew or (C)lose?

By default, the initial current folder is the root folder. The State operation will toggle WAL between folder mode and URL mode. When in folder mode, the first menu line above is displayed, and all of the menu commands operate upon folders. When in URL mode, the second menu line above is displayed, and the commands operate upon the URLs of the current folder. Only the first 50 characters of the folder or URL title/name need be displayed. The Navigate operation, available only in Folder mode, queries the user to allow movement up or down the bookmark folder hierarchy. When moving up, the current folder is replaced by it's parent folder. This results in an error from the root folder. When moving down, the user enters the (L)ist number a folder in the current folder, which then becomes the current folder. Folder (L)ist numbers entered by the user that are not present in the current folder results in an error. After a folder change, the folder menu line is updated with the current *folder name*.

The Add operation allows a user to create an entry to be inserted into the current folder of the open WAL list(s). The user is queried for the appropriate entry, (folder or URL depending on the mode), information for the entry type. (Folder: {title, date and optional one-line comment}; URL: {name, link, add date and optional one-line comment}.) For added URLs, the last visit and modified dates are set equal to the add date. The user is not queried for folder/entry indexes; these are assigned automatically by WAL. The Delete operation allows the user to remove a folder or URL, (mode dependent). The folder/URL (L)ist number is input and the corresponding entry is then removed from the current WAL list(s). In the case of a folder, only the current folder is deleted, after user confirmation. All of the folder's contained URLs, are removed, and all of the sub-folders are *moved* to the parent folder. Their folder-index becomes the folder-index of the folder being deleted. (If the entered folder/URL (L)ist number is not in the current folder an appropriate error message must be given.) The Modify operation queries the user for the (L)ist number of a folder/URL, (depending on the mode of course). It then gives the user the option of changing the title/name of the entry, the date or the entry comment. In the case of a URL, the only date that can be modified is the add date. The List operation simply displays a numbered listing of the folders or URLs of the current folder, depending upon the mode. It should display the folder contents in a sequential positive integer numbered and labeled tabular format. Displaying 10 to 20 list elements at a time, allowing the user to hit the return key to list the next 10/20 elements. More than one line may be used to display an entry. The View operation is a future operation. Selection of the operation should inform the user that it is **not** currently implemented in the current version of the system. (When implemented later, it will allow a user to see a labelled, semi-hierarchical view of all of the bookmark folders and URLs. It begins by showing

the user the name of root folder, followed by its contents indented. The contents of nested folders will also be indented, with the indentation increasing and decreasing depending upon the level of nesting encountered. Only ten to twenty folder/URL entries are displayed on the screen at a time, allowing the user to hit the return key to list the next ten/twenty lines). The Close operation first queries to user to be sure they wish to close the current WAL bookmark list file, (and to continue to really annoy them ☺). It then writes out the WAL bookmark list file, (with a '.wal' extension), (if it has been changed), destroys the internal WAL bookmark list, closes all open files and returns control to the WAL, (initial), menu line.

This program will later be incorporated into an on-line Web Address Management system. Note: The standard double linked-list operations, (insert, delete, etc.), from the text/notes may be used, but must be properly referenced. Code from other sources must be approved by the instructor before it can be used. (To use code from other sources without prior approval is an Honor Code violation.)

Execution

The program should initially present the user with a startup screen displaying the name of the program, a brief 3-4 line explanation of the program and the programmer's name, and email address. After the user hits the return key, the startup screen is to be cleared. A brief help screen should appear explaining the program, which is cleared upon the pressing of the return key. (The startup and help screen may be combined if desired.) Following the brief help screen the WAL, (initial), menu line is displayed and execution continues as described above in the discussion section until the user Quits the system.

Structure Charts

An initial structure chart design must be submitted on **Monday Feb. 16th**. A final structure chart reflecting design changes and corresponding to the code must also be produced and submitted with the executable. The initial chart will be compared against the final chart to determine the quality of the design.

Input

The layout for the WAL bookmark list file is an internal format. No defined format specification is imposed. WAL is only responsible for being able to read WAL bookmark list files that have been produced internally.

Data Structures

The WAL list(s) must be double linked-list(s), implemented with a pointer representation. A design decision has to be made to implement two lists, (folders and URLs) or one list containing **both** folder and URL information. If only one list is implemented, unions could be used, but are NOT required. One struct declaration can be designed to hold all the necessary fields for both folders and URLs. In this case, an enumerated type field, {FolderNode, UriNode}, should be maintained in each list information record.

Assumptions

It may NOT be assumed that a bookmark list will contain any maximum number of entries limit. Error checking must be performed whenever possible, (file existence, illegal menu operation selection, etc.). WAL bookmark list files should have a default file name the same as the Navigator[®] bookmark.htm file, except with a '.wal' extension. Note – don't waste time on a snappy interface; no brownie points will be given for one and no points will be lost for doing one ☺.

Grading

Separate compilation, and an abstract double linked-list data structure, (pointer implementation), are requirements of this program. Failure to utilize these coding features in the program will result in severe grade penalties. Turn in hard copies of the source code, output files, initial and final structure charts. Submit a diskette, (system labeled: Win 95 or NT), with files containing: ASCII source code, executable image, all project workspace files, I/O files, and a brief ASCII *readme* file with execution instructions. The files on the disk may be zip compressed in a self extracting archive. The executable image must be called *wal.exe*. All deliverables must be placed in an sealed 9"x12" envelope, neatly labeled. Only 3 1/2 inch DOS (FAT)1.4 MB disks are to be submitted.

Everyone is required to demonstrate their program to the GTAs. Instructions for the McBryde 116/118 CS lab demo times will be placed on the course Web site. To receive partial credit for programs that are non-working, or are not fully functional, a brief one or two paragraph description of the problem(s) must be included in the assignment envelope. The location, identified to a specific function minimally, must also be listed along with possible corrections that need to be made. Non-working/non-implemented menu operations must not bomb the program when selected. They must display a message informing the user that the features are not currently implemented.