Abstract

The file management commands provided with MS-DOS are unsatisfactory when it comes to managing thousands of files and subdirectories. These MS-DOS commands provide minimal functionality and are tedious and time-consuming to use. Third party file management programs provide greater functionality and ease of use, however, most allow the user to operate on only one disk drive at a time.

This project is a user friendly file management system that operates in an MS-DOS environment. The system allows the user to select files for management from multiple drives and directories simultaneously. The file management functions provide the ability to copy or delete files and entire directory structures, change the attributes of or move files, and view or edit the contents of files. The user interface features pull-down menus, hot-key menu option selection, on-line help, and mouse support.

The program is named Flex for the flexibility it allows the user when selecting files for management.
List of Figures

2.1 Search Specifications Menu ......................... 5
2.2 Attributes Menu ..................................... 6
2.3 Sort-Keys Menu Display ................................ 6
2.4 File Management Menu ................................. 8
1. Introduction

1.1 Overview of the Problem

As the owner of a 110 Megabyte hard disk at home, I manage 3,500 files in 150 directories. As the administrator of a large (over 120 nodes) Local Area Network, I manage over 10,000 files on network servers. Many of these files are similarly named setup and configuration files residing on more than one drive and in many different directories.

MS-DOS file management commands such as "copy" and "delete" are tedious and sometimes frustrating to use when trying to selectively manage a large number of files. The cryptic messages provided by MS-DOS sometimes inadequately reflect the operation performed. The need to type long path names over and over is time-consuming and leaves room for errors. Following are some examples of the problems MS-DOS commands present.

Suppose one issues the command "copy a:\*.* c:\wpdata". If "c:\wpdata" is not an existing directory, all the files from drive A are copied into one file, named "wpdata", on drive C. MS-DOS gives no indication that the files were copied to a single file instead of a directory.

As another example, suppose one wants to delete a directory with two levels of subdirectories under it. A minimum of six DOS commands must be issued (three "del" and three "rmdir" commands). Realistically though, one may want to first issue three "dir" commands to see what files are going to be deleted. If read-only files exist in those subdirectories one must use an additional three "attrib" commands to remove the read-only file attribute before the files can be deleted with the "del" command. Altogether, a total of twelve commands must be issued to remove the one directory and its subdirectories.

The shortcomings of MS-DOS has led to a large market of third-party file management software packages. Typically these products allow the user to work with files on one drive at a time. The past two years have seen the emergence of file management packages that support multiple drives. Although I did not perform an exhaustive search, in my readings of computer journals I have not found a file management package that allows the user to specify that only certain files from the available drives be displayed for management. This need to selectively list and manage files from chosen subdirectories and drives led me to pursue this project.
1.2 Overview of the Flex Program

The Flex program allows the user to input a list of search inclusion and/or exclusion specification strings. The inclusion strings specify which files to include in a list of files displayed for management. The exclusion strings exclude files from being displayed. A specification string may contain drive, path, and filename specifications, for example "c:\data\wp\*.doc". The wildcard character '*' may be used in place of the drive specification to cause all drives to be searched. The DOS wildcard characters '*' and '?' may be used in both the path and the filename specifications. When a search inclusion specification string ends with "/S", all subdirectories below the given directory are also searched. The user may select the file attributes to search for from a list of DOS file attributes. The user may also specify how the displayed file-list is sorted.

Once the files are selected, they are displayed in a list format in a window. This window contains information about the name, drive, path, date, time, attributes and size of the files. The user may scroll through the list and select (mark) the files to be managed. Once selected, the user may copy, move, delete, change the attributes of, view, or edit the files. Some of the special features of Flex include the ability to delete entire directory structures including read-only and hidden files; copy entire directory structures; and load any text editor to edit a file.

The Flex program is a menu driven program that includes pull-down menus, hot-key selection of menu options, user input boxes, windows, mouse support and context sensitive on-line help.
1.3 Resources

I selected C as the programming language for this project because of its power and flexibility. I selected the Borland Turbo C Compiler and Borland Turbo Debugger because of their ease of use and powerful features.

To provide an efficient and friendly user interface, I chose a commercial C library called the Human Interface Manager (HIM). The HIM library supplies routines for windows, pull-down menus, on-line help and mouse support.

The Microsoft Mouse Programmer's Reference provided the mouse driver library to interface with the Human Interface Manager Library.

The Programmer's Guide to the IBM PC by Peter Norton supplied valuable information about MS-DOS function calls, the DOS directory structure, and other programming related information about the IBM family of microcomputers.
2. Flex Physical Characteristics

2.1 Program Environment

This project runs under MS-DOS version 3.0 or greater. It requires a minimum of 256K of memory, although more memory will allow for larger file-lists. It has been tested under MS-DOS versions 3.3 and 4.01. Although Flex was not written to support a specific network environment, it should run under any network operating system that: 1) supports DOS interrupts and function calls and 2) maps server subdirectories to drive letters. A Microsoft compatible mouse is optional.