File System Test Interface Requirements

CSE325

For your filesystem manager, I am providing the specifications of a test interface. Your project should be able to accept commands, as defined in this document, as a way to exercise your filesystem manager. As other managers are added, additional commands may be introduced to enhance testing of the new functionality.

The commands listed are not necessarily all the commands you might need to test your design. I have only designed commands to test specific functionality that is required for the assignment. If you have other internal functionality, you might have to add new commands to be able to fully test your implementation.

Initialization Commands

1. *init_fs* device

   This command will initialize the filesystem manager such that it knows about the listed device. **Extra Credit**: Design, implementation, and testing for multiple devices (up to 5 points).

Status Commands

1. *list* {DEVICES | FILEINFO filename | DIRECTORY}

   The directive “list” without arguments will perform all lists. The argument DEVICES will instruct "list" to output all the devices known to have valid filesystems. The argument FILEINFO, with the specified name, will output information from the FCB, such as size and block numbers that hold the file contents. An argument of DIRECTORY will list information about all valid files in the filesystem.

Action Commands

1. *format* device fs_name blocksize

   This command will format the specified device into blocks of the specified size and associate the specified name with the device. It should return the number of blocks if successful or an error code for failure.

2. *mount* fs_name

   The command will allocate internal system structures for the named filesystem. It should return success or failure.

3. *open* filename [new | read-only | read-write]

This command will open a file with the specified name. The argument specified will determine the use of the file that is being opened: new says a new file is to be created, if an existing file of the same name exists then (choose one: delete old file, return error, or ???); read-only says the file must exist and will not be allowed to be changed; read-write says the file must exist and the contents may be changed. The command should return success or failure as well as the "handle" of the file.

4. *read* filehandle block_number buf_ptr

The command requires 3 arguments, the filehandle, block number of the file to read, and the buffer to place the data. Return an error for an invalid filehandle or invalid operation on the file (opened as new).

5. *write* filehandle block_number buf_ptr

This command requires 3 arguments, the filehandle, block number of the file to write, and the buffer to be written. Return an error for invalid filehandle or invalid operation on the file (opened as read-only).

6. *close* filehandle

The command will close the file, update any FCB information and return success or failure.

7. *delete* filename

This command will delete the FCB entry for the named file and return any file space to the free pool for use by other processes. It will return an error for an incorrect filename, or success.

Expected use of this test interface will be like:

- Format the filesystem
- Mount the filesystem
- Multiple “opens” to build up some files
- An occasional “list” to check for valid information