VSFS MAN + REPORT

NAME

VSFS - Short for Very Simple File System

SYNOPSIS

VSFS list [notes file]

VSFS copyin [notes file] [IF] [EF]

VSFS copyout [notes file] [EF] [IF]

VSFS mkdir [notes file] [Directory name]

VSFS rm [notes file] [File name]

VSFS rmdir [notes file] [Directory name]

VSFS defrag [notes file]

DESCRIPTION

VSFS is a very simple file system, designed to be contained within a .notes file. It operates as a bare bones file system, allowing the containment of files and directories which are depicted within the notes file by their first character.

- **list - List all files within FS in traditional 'ls -r' format**
- **copyin - Copies an external file and its contents into the FS**
- **copyout - Creates a copy from the FS to external file system**
- **mkdir - Creates a directory within the FS**
- **rm - Removes a file within the FS**
- **rmdir - Removes a file and its contents within the FS**
- **defrag - Removes all deletend entries within the FS**

EXAMPLES

java Driver VSFS mkdir test.notes dir1

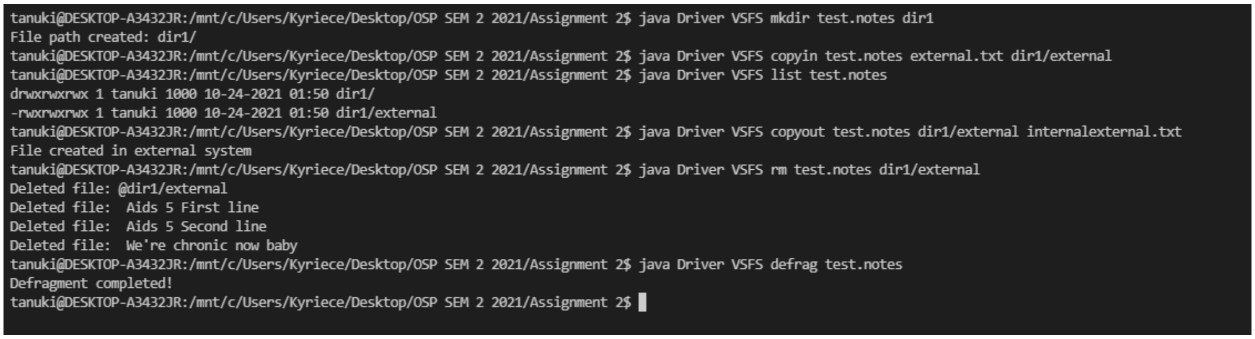
java Driver VSFS copyin test.notes external.txt dir1/external

java Driver VSFS list test.notes

java Driver VSFS copyout test.notes dir1/external internalexternal.txt

java Driver VSFS rm test.notes dir1/external

java Driver VSFS defrag test.notes



```
tanuki@DESKTOP-A3432JR:/mnt/c/Users/Kyriece/Desktop/OSP SEM 2 2021/Assignment 2$ java Driver VSFS mkdir test.notes dir1
File path created: dir1/
tanuki@DESKTOP-A3432JR:/mnt/c/Users/Kyriece/Desktop/OSP SEM 2 2021/Assignment 2$ java Driver VSFS copyin test.notes external.txt dir1/external
tanuki@DESKTOP-A3432JR:/mnt/c/Users/Kyriece/Desktop/OSP SEM 2 2021/Assignment 2$ java Driver VSFS list test.notes
drwxrwxrwx 1 tanuki 1000 10-24-2021 01:50 dir1/
-rwxrwxrwx 1 tanuki 1000 10-24-2021 01:50 dir1/external
tanuki@DESKTOP-A3432JR:/mnt/c/Users/Kyriece/Desktop/OSP SEM 2 2021/Assignment 2$ java Driver VSFS copyout test.notes dir1/external internalexternal.txt
File created in external system
tanuki@DESKTOP-A3432JR:/mnt/c/Users/Kyriece/Desktop/OSP SEM 2 2021/Assignment 2$ java Driver VSFS rm test.notes dir1/external
Deleted file: @dir1/external
Deleted file:  Aids 5 First line
Deleted file:  Aids 5 Second line
Deleted file:  We're chronic now baby
tanuki@DESKTOP-A3432JR:/mnt/c/Users/Kyriece/Desktop/OSP SEM 2 2021/Assignment 2$ java Driver VSFS defrag test.notes
Defragment completed!
tanuki@DESKTOP-A3432JR:/mnt/c/Users/Kyriece/Desktop/OSP SEM 2 2021/Assignment 2$ █
```

Language justification

The language I decided to use was java. This was more of a comfort choice as its the language I was most familiar with and have had the most experience with in programming.

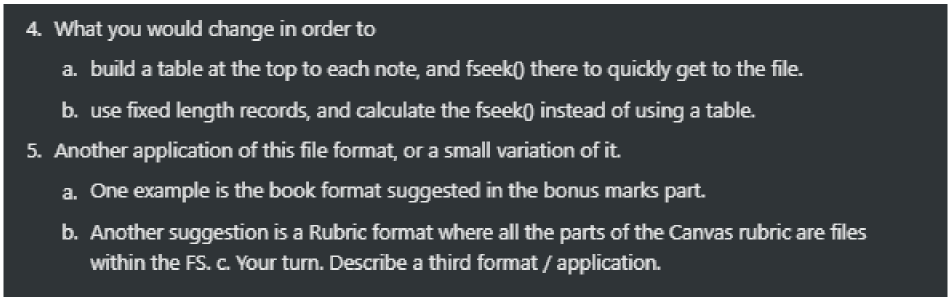
Additional Modifications

The design I implemented is very brute-force in the sense of working with the .notes file. Primarily I believe majority of the flaws come from the lack of OOP integration. Creating an array list of file objects, rather than just Strings (in reference to FSContent) when first reading the file would have made everything much more smoother.

Assuming the above emplementation is incorporated, we could now easily sort folders catagorically. This means we could create a table for every folder, which points to a table which contains all the files within that folder. This would allow for quick acess to any file.

For fixed length records, we would likely split each record using a specific char or empty line. Each record will have its contents pointers, which refers to the offset at which the content desired is located.

This type of application would be primarily used to split things 'catagorically', therefore I would suggest implementing something along the lines of an encyclapedia, which contains all the different types of animals. This could also be with previous suggestions, where instead of folders pointing to files, it'll be species pointing to subspecies and subspecies pointing to information about that animal.



4. What you would change in order to
    a. build a table at the top to each note, and fseek() there to quickly get to the file.
    b. use fixed length records, and calculate the fseek() instead of using a table.
5. Another application of this file format, or a small variation of it.
    a. One example is the book format suggested in the bonus marks part.
    b. Another suggestion is a Rubric format where all the parts of the Canvas rubric are files within the FS. c. Your turn. Describe a third format / application.