CS333 - INTRODUCTION TO OPERATING SYSTEMS - 19APCS2
PROJECT 2

Sauce:
19125063 - Châu Hữu Phát
19125090 - Trương Quý Hải
19125119 - Trương Lăng Trường Sơn

Ideas:

1. **int Create(char *name):**
   - This function is used to create a file with a given name. It returns 1 for success or 0 for failure.
   - The function is implemented using the filesystem with the function filesystem → Create(filename, 0).

2. **OpenFileID Open(char *name), int Close(OpenFileID id):**
   - With the Open function, we use Openfileid Open(char* name, int type), the function given by the system. The ouput is 1 for success and 0 for failure.
   - As stated in the requirements, a table used to describe the files (10 files at maximum) is required. Thus, the is a table with 10 rows with each row describing a file system. We use the Table added in thread.h and thread.cc to do this task.
   - With the Close fuction, we use the fuction currentThread → gdTable.closeFile(idFile) with output 1 for success and 0 for failure.

3. **int Read(char *buffer, int size, OpenFileID id), int Write(char *buffer, int size, OpenFileID id):**
   - With the Read function, we use the function Read of the fileSystem if fileId is different from 0 and we use GetString from synchConsoleIn if fileId is 0. The output is 1 for success and 0 for failure.
   - With the Write function, if fileId is 1 => use PutString of synchConsoleOut, fileId != 1 => use fileSystem->Write. The output is 1 for success and 0 for failure.

4. **int Seek(int position, OpenFileID id):**
   - This function is used to change the position of the pointer in a file.
   - We use the information in the gTable to find the position of the pointer and move it to the required position.
   - We use OpenFile::Seek.

5. **int Remove(char *name):**
   - This function is used to remove a file.
   - This function should check if the file is closed before removing the file.
   - And then we call the kernel → fileSystem to remove the file.

6. **Program createfile:**
   - This program is used to create a file.

- This program is implemented using syscall ReadString to get filename from users and syscall Create.

7. **Program cat:**
   - This program is used to display the content of a file.
   - First, we use syscall ReadString to ask users to input the filename.
   - Then, we save the filename to memory and use syscall Open with type = 1.
   - If there is no error while reading, read the file with the max length and display it to the console.

8. **Program copy:**
   - This program is used to copy the data of a source file to a destination file.
   - First, we use syscall ReadString to ask users to input the name of the source file and destination file.
   - Then, we save these filenames to memory and use syscall Open with type = 1 for the source file and type = 0 for the destination file.
   - If there is no error while reading, read the source file with the max length and write to the destination file.

9. **Program delete:**
   - This program is used to delete a file.
   - This program is implemented using syscall ReadString to get filename from users and syscall Remove.

10. **Program concatenate:**
    - This program is used to concatenate the contents of two source files and output them to the destination file.
    - First, we use syscall ReadString to ask users to input the name of both source files and one destination file.
    - Then, we assign the content of the first file to the beginning of the string buffer and assign the content of the second file to the next part of the string.
    - Therefore, we get the concatenated content of both files, and then write it to the destination file.

How to run:

1. In code/build.linux run:
   - make depend
   - make
2. Usage:
   - Run "nachos -u" for all command-line options.