

Documentation for “File Search” App

Please resolve the task within 3 days (the implementation should be uploaded to Git. The link should be added to this form)

Application (App) to find file on Local Host:

1. App has to find file by "name".
2. Search has to be started from root "/" directory. (or Disk "C:\\" on Windows OS)
3. To make this search faster - App creates new Searching Thread for each sub-directory in root directory, but not more than 8 Execution Threads.
4. Thread who will find needed file has to print Full Path to the file and has to notify all the rest threads to stop searching.

=====

- Logic "to search file" has to be placed into static library.
- Application just uses this library to find files in a fast way.
- Static Library compilation has to be done separately from Application.
- Compiler: GCC or G++.
- Use Makefile for compilation/linking instructions.

=====

Things to Google: (Concurrency, Makefile, GCC, Static Library)

Program Code:

```
#include <iostream>
#include <filesystem>
#include <thread>
#include <vector>

namespace fs = std::filesystem;

void searchFiles(const std::string& directory, const std::string& targetFileName) {
    for (const auto& entry : fs::recursive_directory_iterator(directory,
fs::directory_options::skip_permission_denied)) {
        try {
            if (entry.path().filename() == targetFileName) {
                std::cout << "File found: " << entry.path() << std::endl;
            }
        }
        catch (const fs::filesystem_error& ex) {}
    }
}

void searchRootDirectories(const std::string& root, const std::string& targetFileName)
{
    std::vector<std::thread> threads;
    for (const auto& entry : fs::directory_iterator(root,
fs::directory_options::skip_permission_denied)) {
        try {
            if (entry.path().filename() == targetFileName) {
                std::cout << "File found: " << entry.path() << std::endl;
            }

            if (fs::is_directory(entry)) {
                threads.push_back(std::thread(searchFiles, entry.path().string(),
targetFileName));
            }
        }
        catch (const fs::filesystem_error& ex) {}
    }

    for (auto& thread : threads) {
        thread.join();
    }
}
```

```
int main() {
    std::string targetFileName;
    std::cout << "Enter the name of the file to search: ";
    std::cin >> targetFileName;

    std::string searchDirectory = "C:\\";
    searchRootDirectories(searchDirectory, targetFileName);

    system("pause"); return 0;
}
```

Creating an exe file using g++, running the program in the gcc compiler:

```
PS C:\WINDOWS\system32> cd C:\Users\Користувач\Downloads\test_folder
PS C:\Users\Користувач\Downloads\test_folder> ls

Directory: C:\Users\Користувач\Downloads\test_folder

Mode                LastWriteTime         Length Name
----                -
-a----          29.07.2023   22:42             1560 file_search.cpp

PS C:\Users\Користувач\Downloads\test_folder> g++ -std=c++17 file_search.cpp -o file_search
PS C:\Users\Користувач\Downloads\test_folder> dir

Directory: C:\Users\Користувач\Downloads\test_folder

Mode                LastWriteTime         Length Name
----                -
-a----          29.07.2023   22:42             1560 file_search.cpp
-a----          29.07.2023   22:43          233188 file_search.exe

PS C:\Users\Користувач\Downloads\test_folder> ./file_search
Enter the name of the file to search: Vagrant
File found: "C:\HashiCorp\Vagrant"
terminate called after throwing an instance of 'std::filesystem::__cxx11::filesystem_error'
  what():  filesystem error: directory iterator cannot advance: Invalid argument
PS C:\Users\Користувач\Downloads\test_folder> ./file_search
Enter the name of the file to search: FXSEXT.ecf
terminate called after throwing an instance of 'std::filesystem::__cxx11::filesystem_error'
  what():  filesystem error: directory iterator cannot advance: Invalid argument
PS C:\Users\Користувач\Downloads\test_folder> _
```

Enter the name of the file to search: Accounts.txt

terminate called after throwing an instance of 'std::filesystem::__cxx11::filesystem_error' what():
filesystem error: directory iterator cannot advance: Invalid argument

```
Enter the name of the file to search: Accounts.txt
terminate called after throwing an instance of 'std::filesystem::__cxx11::filesystem_error'
  what():  filesystem error: directory iterator cannot advance: Invalid argument
PS C:\Users\Користувач\Downloads\test_folder> |
```

As you can see, the program doesn't work correctly when it searches for a files with an extension, but this error is only in GCC, there is no such error in Visual Studio.

Pushing App to GitHub using Git Bash:

```
Користувач@Vazonok MINGW64 ~/Downloads/test_folder (main)
$ ls
file_search.cpp  file_search.exe*

Користувач@Vazonok MINGW64 ~/Downloads/test_folder (main)
$ git init
Initialized empty Git repository in C:/Users/Користувач/Downloads/test_folder/.git/

Користувач@Vazonok MINGW64 ~/Downloads/test_folder (main)
$ git add .

Користувач@Vazonok MINGW64 ~/Downloads/test_folder (main)
$ git commit -m "File_Searcher_V1"
[main (root-commit) 5ef82ca] File_Searcher_V1
 2 files changed, 48 insertions(+)
 create mode 100644 file_search.cpp
 create mode 100644 file_search.exe

Користувач@Vazonok MINGW64 ~/Downloads/test_folder (main)
$ git branch -M main

Користувач@Vazonok MINGW64 ~/Downloads/test_folder (main)
$ git remote add origin https://github.com/AndrewNFUU/File_Search_App.git

Користувач@Vazonok MINGW64 ~/Downloads/test_folder (main)
$ git push
fatal: The current branch main has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin main

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.

Користувач@Vazonok MINGW64 ~/Downloads/test_folder (main)
$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 46.45 KiB | 6.63 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/AndrewNFUU/File_Search_App.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

Користувач@Vazonok MINGW64 ~/Downloads/test_folder (main)
$
```