실습 093 - filesystem

제출 마감일: 2023-06-01 23:59
업로드 가능한 파일 수: 2
제출 방식: 개인

목적

이 실습은 C++17 표준에서 지원하는 filesystem 을 사용하는 연습을 합니다.

설명

C++17 표준부터 파일시스템 지원 라이브러리가 추가됐습니다.

사용하려면 #include <filesystem> 헤더 파일을 인클루드 해야 합니다.

(CLION 에서는 CMakeLists.txt 파일의 하단에 target_link_libraries([자신의 프로젝트명] stdc++fs) 를 추가해 주셔야 합니다)

파일시스템 관련 클래스 정의는 std::filesystem 네임스페이스에 포함되어 있습니다.

즉, using namespace std::filesystem; 을 이용하면 짧게 클래스명 만으로도 관련 클래스를 사용할 수 있습니다.

주요 클래스로는 path, directory_entry 가 있습니다.

path 클래스는 경로를 추상화한 클래스입니다.

```
path p1 ("C:\Users") ;        p1/= "me" ;        cout << p1 << endl;
path p2 ("C:\Windows") ;    p2 += "\Fonts" ;    cout << p2 << endl;

for(const auto& sub_path : p1)
        cout << sub_path << endl;
```

directory_entry 클래스는 디렉토리나 파일을 추상화한 클래스입니다.

is_directory(), is_regular_file() 등의 멤버 함수를 제공합니다.

```
directory_entry dir (p1);

if ( dir.exists() and dir.is_regular_file() )

        cout << " file size: " << dir.file_size() << endl;
```

또한, 디렉토리에 속한 파일이나 서브 디렉토리를 순회하는 directory_iterator, recursive_directory_iterator 이터레이터도 제공합니다.

```
//현재 디렉토리의 파일명을 출력함

for (const auto& file : fs::directory_iterator("./")) {
   if (is_regular_file(file)) {
         cout << file.path() << endl;
   }
}
```

```
//현재 디렉토리의 파일명 및 서브 디렉토리의 파일명을 모두 출력함

auto begin = fs::recursive_directory_iterator(p);
auto end = fs::recursive_directory_iterator();
for(auto it = begin; it != end; ++it){
   const string blank(it.depth()*2, ' ');
   auto& entry = *it;
   if(fs::is_regular_file(entry)){
      cout << blank << " F " << entry;
      cout << " (" << file_size(entry) << ") bytes " << endl;
   } else if (fs::is_directory(entry)){
      cout << blank << " D " << entry << endl;
   }
}
```

참고: Filesystem library

문제

현재 디렉토리 및 서브 디렉토리의 파일명을 출력하는 프로그램을 작성하시오

(단, 컴파일에 성공하면 제출이 된 것으로 간주함)

<참고>

- 컴파일에 필요한 헤더 파일 및 namespace

```
#include <filesystem>
#include <iostream>
#include <string>

using namespace std;
namespace fs = std::filesystem;
```

...

- 윈도우에서는 별다른 옵션 설정 없이 CLion 에서 실행 가능함
- PLATO 코딩 서버 (Linux) 에서는 다음 명령어로 컴파일 함

```
$>  g++ -fno-diagnostics-color -std=c++17 -o target -lm -lutil -lstdc++fs
```

입력

없음

출력

출력을 비교하지 않음

제출파일

main.cpp
93.csv