

76 How to Deal with OPTIONAL Data in C++

#将不同类型的数据存储在单变量中系列1/3

很多时候，我们有一个返回数据的函数，比方说我们正在读取一个文件，但是如果这个文件不能被读取会发生什么？它可能不存在，或者是数据不是我们期望的格式，我们仍然需要从函数中返回一些东西。在这个特定的情况下，可能只会返回一个空字符串，但这没有多大意义。意思是，如果读取文件是空的，我们应该有办法看到数据是否存在。

这就是要用到 `std::optional` 的地方了，这是C++17标准的新东西。

```
C++

#include <fstream>

std::string ReadFileAsString(const std::string& filepath)
{
    std::ifstream stream(filepath);
    if(stream)
    {
        std::string result;
        // Read file
        stream.close();
        return result;
    }
    return std::string();
}

int main()
{
    std::string data = ReadFileAsString("data.txt");
    if(data!="")
    {
        // 但这种检查是否打开成功的方式不是很好
    }
}
```

我们更喜欢的应该是能够知道数据是否被成功读取的方法：

```
std::string ReadFileAsString(const std::string& filepath, bool& outSuccess) // 判断读取成功的参数
{
    std::ifstream stream(filepath);
    if(stream)
    {
        std::string result;
        // Read file
        stream.close();
        outSuccess = true;    // 读取成功赋值
        return result;
    }
    outSuccess = false;    // 读取失败赋值
    return std::string();
}

int main()
{
    bool fileOpenSuccessfully;
    std::string data = ReadFileAsString("data.txt", fileOpenSuccessfully);
    if(fileOpenSuccessfully)
    {

    }
}
```

这显然比判断空字符串好得多，但仍然不够好。
这就要用到 `std::optional` 了，它在头文件 `<optional>` 中。

```

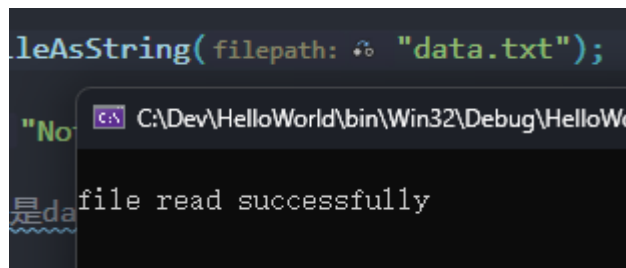
#include <fstream>
#include <optional>

std::optional<std::string> ReadFileAsString(const std::string& filepath)
{
    std::ifstream stream(filepath);
    if(stream)
    {
        std::string result;
        // Read file
        stream.close();
        return result;
    }
    return {};
}

int main()
{
    std::optional<std::string> data = ReadFileAsString("data.txt"); // 可以auto
    if(data) // 本来可以写data.has_value(), 但是data对象有一个bool运算符, 所以这样写更简洁
    {
        std::cout << "file read successfully\n";
    }
    else
    {
        std::cout << "file could not read\n";
    }
    std::cin.get();
}

```

可以在sln文件的同目录下创建一个data.txt文档，里面写点东西后运行代码，可以得到读取成功提示。



还可以通过 `value_or` 来实现存储值不存在时返回传入的默认值：

```

std::string value = data.value_or("Not present");
std::cout << value << "\n";

```

```
ng(filepath: "dat2a.txt");  
se C:\Dev\HelloWorld\bin\Win32\Debug\  
Not present  
file could not read  
_
```

`std::optional::value_or` 是一个成员函数，它返回存储的值或者在存储的值不存在时返回传入的默认值。在你的例子中，如果 `data` 存在值（即文件成功读取），`value_or` 就返回文件的内容；否则，它返回 "Not present"。