

HW11 多线程同步练习

截止 6月6日 23:59 **得分** 0 **提交** 一份上传文件
可用 5月23日 0:00 至 6月15日 23:59

此作业锁定于 6月15日 23:59。

实现

需求描述

在多线程编程中，有时会遇到这样的场景：某个需要初始化的资源需要被多个线程访问，该初始化必须只被执行一次，且初始化因为各种原因不能像static变量那样在进程开始还在单线程的状态下就完成。现在请你用C++实现一个这个功能。

参考API

你需要实现一个类 `waiting_once`，其中包含有 `call_once_waiting` 成员函数。该函数的实现需要保证：

1. 初始化函数只会调用一次，即同一个 `waiting_once` 对象的多次 `call_once_waiting` 中，只有一次运行了参数提供的函数（如果不同调用次数间参数内的函数不同，则运行其中任意一个函数）；
2. 调用者可确定函数返回时一定已初始化完毕，即每个 `call_once_waiting` 调用退出时，参数函数一定都已经执行完毕。

```
struct waiting_once
{
    // 如果想遵从C++23，可以用 std::move_only_function<void()>
    using init_function = std::function<void()>;

    void call_once_waiting(init_function f);
private:
    // 添加你需要的成员或函数
};
```

实现性能要求

朴素实现

一个显然的实现是使用mutex保护一个记录是否已经执行过 `call_once` 操作的变量，请实现。

进阶实现

由于初始化只有1次，当程序运行一定进度后，再每次都上互斥锁开销太大。请考虑如何优化。

报告

1. 请简述你的优化方法。
2. 请测试你两种实现的性能，至少包括已经完成初始化后 `call_once_waiting` 的吞吐量（即单位时间内能执行几次 `call_once_waiting` 函数调用）。

注意事项

- 本次作业含编程作业。
- 助教提供了**代码框架** ↓ (https://oc.sjtu.edu.cn/courses/64176/files/9175767/download?download_frd=1)，你可以依照自己需求对框架进行任意修改，包括接口、实现、测试与构建系统。
- 注意：助教设计API并实现要求功能的时间在半小时以内，需要几十行的代码（不包括代码框架和测试部分）。请同学们合理安排时间和工作进度，以完成课程的要求。
- 请将你的代码与报告提交到 Canvas。命名使用“学号+姓名+hw11”，如“522123456789+张三+hw11.pdf”等。
- 请勿抄袭！课后作业采用倒扣分制，如果有遗漏或者得分不足会在最终成绩酌情减分，同时课后作业的内容会体现在期末试卷中，对同学们也是一种练习。
- 本次作业的截止时间是 2024年6月6日 23:59，迟交将会酌情扣分。
- 有任何作业相关的问题可以询问 熊天磊、徐凯亮 助教。