第六次作业: 写计时类

2100011046 杨家骅

项目结构

项目目录分为类库和测试程序两部分，采用CMake进行多文件编译的组织。

- 计时类`Timer`的代码在目录TimerClass/src中，分为"Timer.h"和"Timer.cpp"两个文件生成一个静态库。该类定义在名称空间`HwaUtil`中。
- 对该类的测试程序`TimerTest`的代码在目录TimerClass/test中，生成一个可执行文件。
- 测试利用了上次作业的代码, 程序引用了之前编写好的`ArgumentReader`类和`Mat_Demo`两个类,代码在目录TimerClass/src中，各自生成一个类库。这两个类都定义在名称空间`HwaUtil`中,并已经经过修改,嵌入了对计时类`Timer`的调用。
- input.txt文件分别用于测试时给定输入内容，与`Mat_Demo`类测试时相同,编译时会同步拷贝到输出目录。

`Timer` 类使用说明

为实现计时功能，只需在每段需要计时的函数开始处和所有可能返回的语句前分别调用`Timer::start()`和`Timer::stop()`即可。计时结果会自动记录到全局存在的`Timer::time`中。
程序结束时调用print_time_usage，输出程序运行的总时间,并以表格的形式列出各函数用时情况。

`Timer` 类的实现方法概述

通过一静态哈希表`func_time_info`将函数名映射到每个函数对应的FuncTimeInfo实例，FuncTimeInfo类记录一个函数每次调用的开始时间和结束时间，存储在TimerRecord结构的vector中。计时采用C++标准库中的`std::chrono::high_resolution_clock`类，该类提供了`now()`函数，返回一个`std::chrono::time_point`对象表示当前的时间。两个`std::chrono::time_point`对象之差为`std::chrono::duration`对象，表示经历的时间间隔。

`Timer`类还支持递归调用的情况, 利用一个`n_recursion`变量跟踪记录当前还未返回的函数实例数, 仅在最外层的递归返回时认为一次调用结束, 最终得到的调用次数也是互不包含的调用"群组"数量。

`TimerTest` 测试程序

`main`函数中,首先进行了Fibbonacci数列前15项的计算,以测试计时类对递归程序的支持;进而执行了矩阵类和输入类的程序,并在运行全程进行统计.

编译

要求C++20支持.

```
mkdir cmake-build
cd cmake-build
cmake ..
cmake --build .
```

运行

```
cd ./bin/TimerTest
./TimerTest
```

输出如下：

```
❯ ./TimerTest
Now calculating the first 15 Fibonacci numbers:
0
1
1
2
3
5
8
13
21
34
Now doing MatDemoTest
calculation: max
0.999458
Matrix printed:
7.82637e-06 0.131538 0.755605 0.45865 0.532767 0.218959 0.0470446 0.678865 0.679296 0.934693 0.383502 0.519416
0.830965 0.0345721 0.0534616 0.5297 0.671149 0.00769819 0.383416 0.0668422 0.417486 0.686773 0.588977 0.930436
0.846167 0.526929 0.0919649 0.653919 0.415999 0.701191 0.910321 0.762198 0.262453 0.0474645 0.736082 0.328234
0.632639 0.75641 0.991037 0.365339 0.247039 0.98255 0.72266 0.753356 0.651519 0.0726859 0.631635 0.884707
0.27271 0.436411 0.766495 0.477732 0.237774 0.274907 0.359265 0.166507 0.486517 0.897656 0.909208 0.0605643
0.904653 0.504523 0.516292 0.319033 0.986642 0.493977 0.266145 0.0907329 0.947764 0.0737491 0.500707 0.384142
0.277082 0.913817 0.529747 0.464446 0.94098 0.050084 0.761514 0.770205 0.827817 0.125365 0.0158677 0.688455
0.868247 0.629543 0.736225 0.725412 0.999458 0.888572 0.233195 0.306322 0.351015 0.513274 0.591114 0.845982
0.412081 0.841511 0.269317 0.415395 0.537304 0.467917 0.287212 0.178328 0.15372 0.571655 0.802406 0.0330538
0.53445 0.49848 0.955361 0.748293 0.554584 0.890737 0.624849 0.84204 0.159768 0.212752 0.71471 0.130427
0.0909903 0.274588 0.0029996 0.414293 0.0268763 0.70982 0.937897 0.239911 0.180896 0.31754 0.886991 0.652059
0.150335 0.681346 0.385815 0.387725 0.499741 0.147533 0.587187 0.845576 0.590109 0.955409 0.556146 0.148152

Program total time: 941 microseconds
|CLASS_NAME----------------|FUNC_NAME------------------|TIME(Sec)------|CALLS----|AVG(Sec)-----|PER%--------|
                        --                      TOTAL     0.000941166         1   0.000941166         100%
          HwaUtil::Mat_Demo                       mmax        2.25e-06         1      2.25e-06         0.2%
            HwaUtil::(root)       operator<<(Mat_Demo)      0.00010025         1     0.00010025        10.7%
          HwaUtil::Mat_Demo                         ()        4.25e-06         1      4.25e-06         0.5%
    HwaUtil::ArgumentReader                   ReadArgs      6.1708e-05         1    6.1708e-05         6.6%
    HwaUtil::ArgumentReader                     AddArg      1.7041e-05         5    3.4082e-06         1.8%
    HwaUtil::ArgumentReader                     GetArgV      1.1792e-05         5    2.3584e-06         1.3%
            HwaUtil::(root)                         fib     0.000522417        10   5.22417e-05        55.5%
            HwaUtil::(root)                        main        0.000914         1      0.000914        97.1%
-------------------------------------------------------------------------------------------------------------
```