实验6：OpenMP并行编程

1. 实验目的

（1）掌握OpenMP编程模型和应用场景，使用OpenMP并行化DGEMM

（2）理解OpenMP和pthreads编程方法的区别

2. 实验内容

（1）运行示例代码，掌握OpenMP的基本使用。

（2）在框架代码how-to-optimize-gemm中集成OpenMP实现DGEMM的版本，并记录相关数据。

3. 实验步骤

1. 新增openmp_gemm.c，在实现的naive gemm的代码的基础上，增加openmp编译指导语句
2. 修改makefile，增加编译选项，使得能够处理openmp的编译指导语句，修改变量NEW的赋值，修改环境变量调整线程数目，使得线程数与CPU的物理核心数一致
3. 运行make run 生成openmp版本的并行化可执行文件并执行测试

4. 实验报告及要求

汇总前面基于框架代码how-to-optimize-gemm不同方式的DGEMM实现，报告中要有以下内容：

1. 列出实验环境：操作系统版本，编译器版本，CPU的物理核数、频率
2. 每种方式的简单介绍，实现的核心代码
3. gflops曲线图，naive、openblas、pthreads、openmp四种情况画在一张图里面，矩阵规模至少包括8，32，256，512，1024，4096，8192。 如果矩阵规模/CPU核心数不是整数，可以调整将矩阵规模调整，比如1000，4000。画图可以用plotAll.py或者excel，并对数据做汇总分析说明。
4. 开启openmp时cpu利用率和能体现OpenMP开启多个线程的运行截图，OpenMP碰到的问题及解决过程，前面lab3和lab5要求记录和回答的内容。