**中山大学计算机院本科生实验报告**

**（2024学年春季学期）**

**课程名称：并行程序设计 批改人：**

|  |  |  |  |
| --- | --- | --- | --- |
| 实验 | **1-MPI矩阵乘法** | 专业（方向） | **计算机科学与技术** |
| 学号 | **21307174** | 姓名 | **刘俊杰** |
| Email | **liujj255@mail2.sysu.edu.cn** | 完成日期 | **2024/3/28** |

# **1.实验目的**

使用 MPI 点对点通信实现并行矩阵乘法，以探究并行计算在提高矩阵乘法效率方面的作用。

研究不同进程数量和矩阵规模下程序的运行时间，分析并行性能随进程数量和矩阵规模的变化。

## 讨论:

## 在内存受限情况下，如何进行大规模矩阵乘法计算？

## 如何提高大规模稀疏矩阵乘法性能？

# **实验过程和核心代码**

## 2.0 实验思路

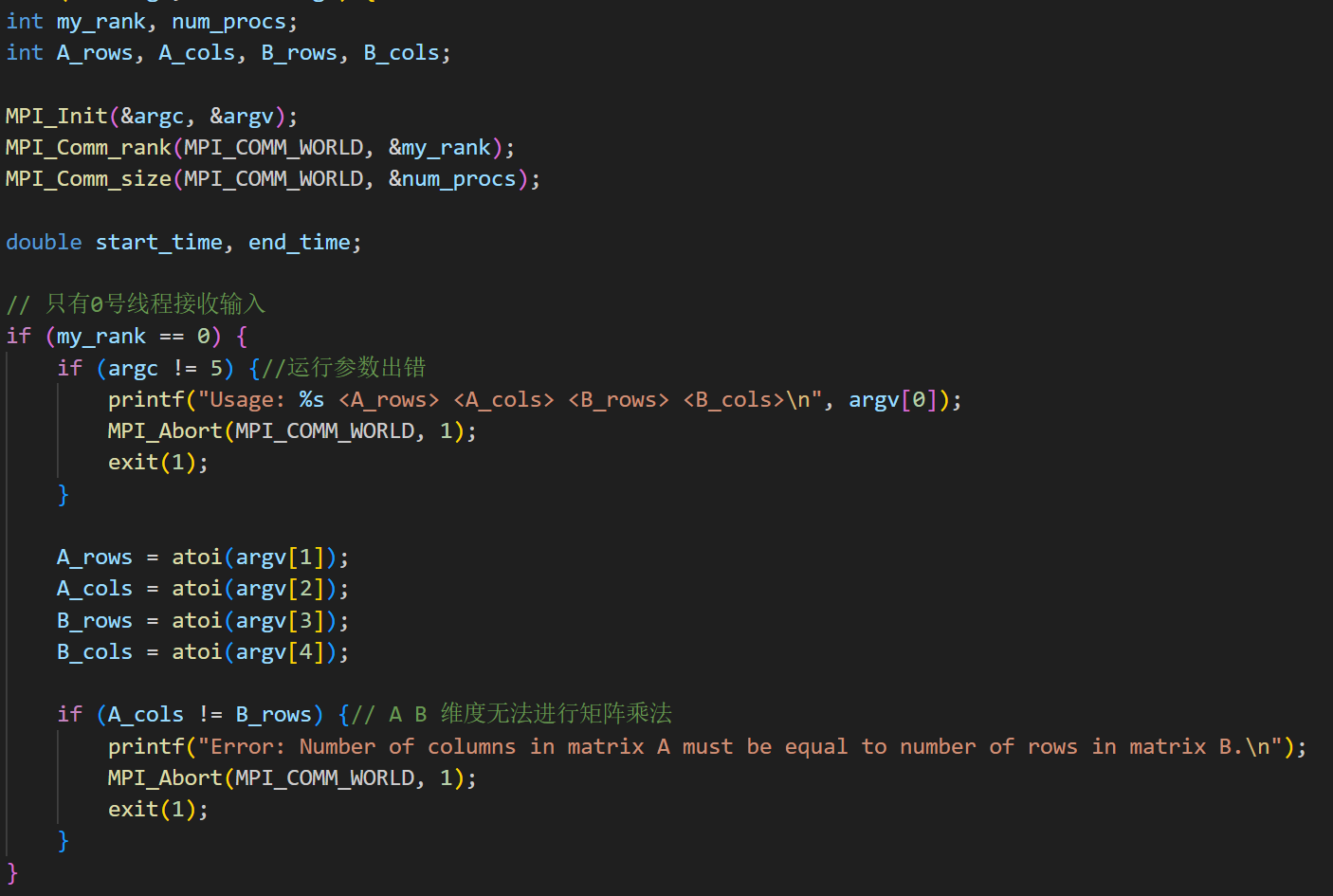
输入矩:阵 A\ B

结果矩阵: C

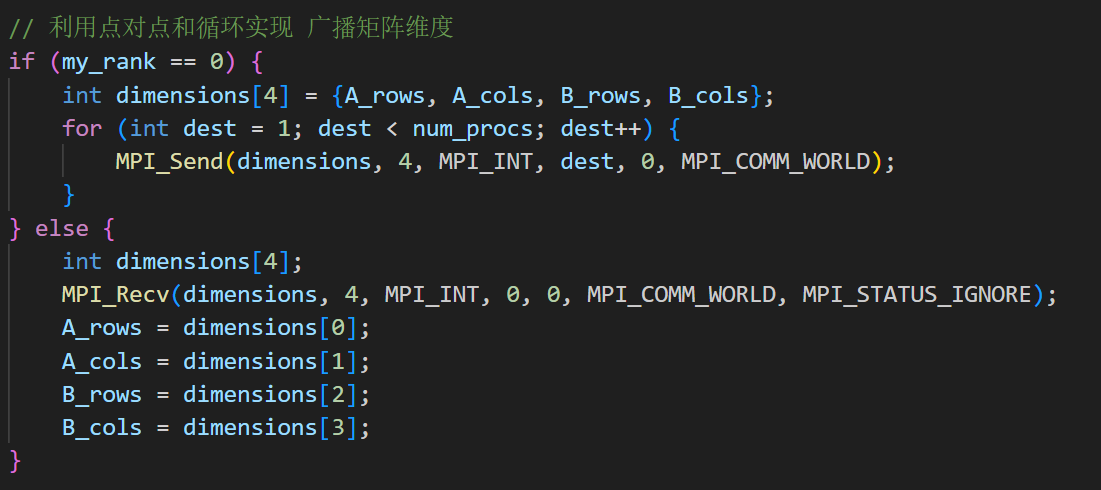
将矩阵A的不同行和整个矩阵B发送给

## 2.1 MPI初始化和矩阵维度初始化

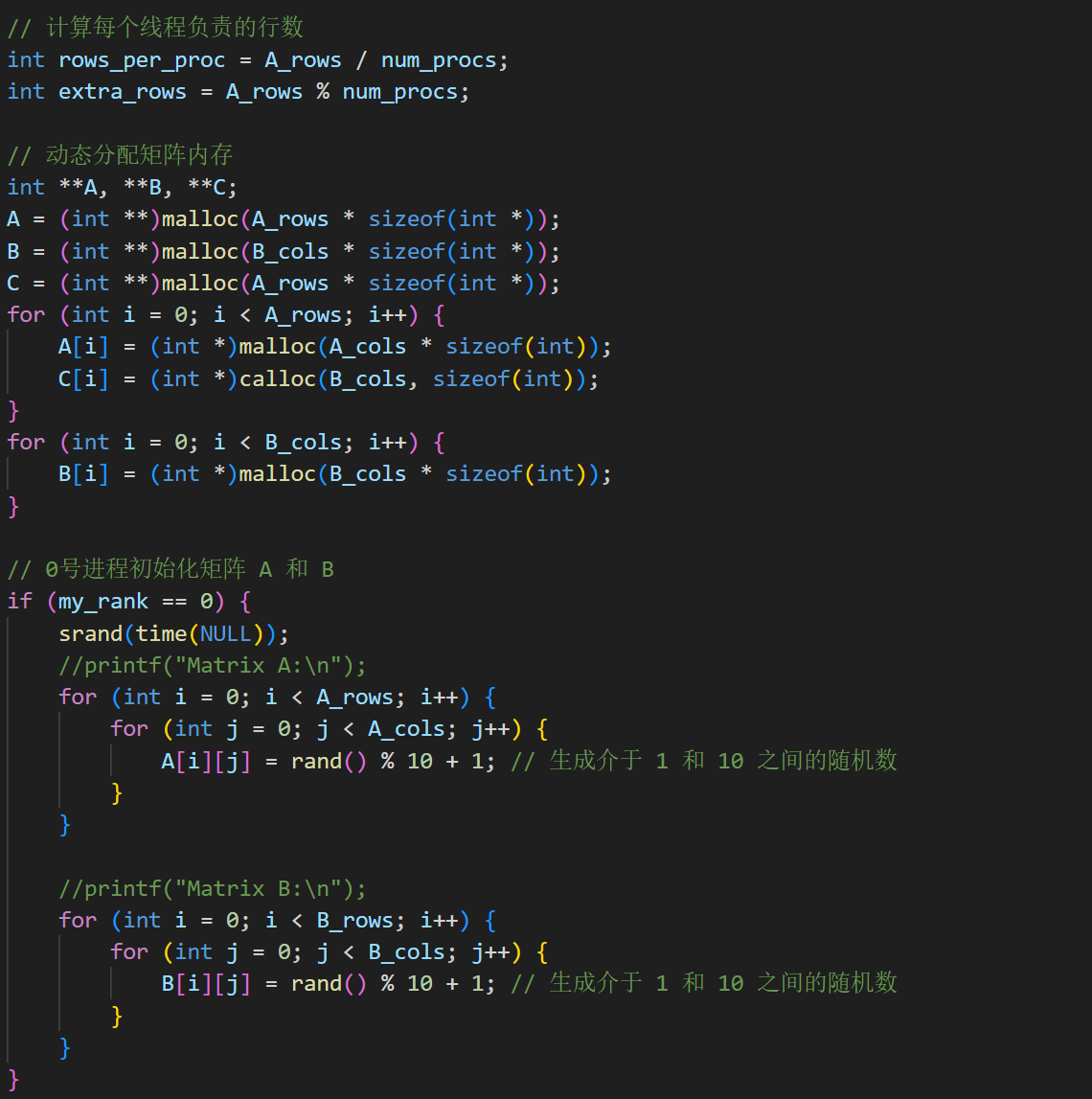
使用0号进程从运行参数中获得矩阵维度



## 2.2 利用MPI\_Send和MPI\_Recv 将矩阵维度信息广播到进程

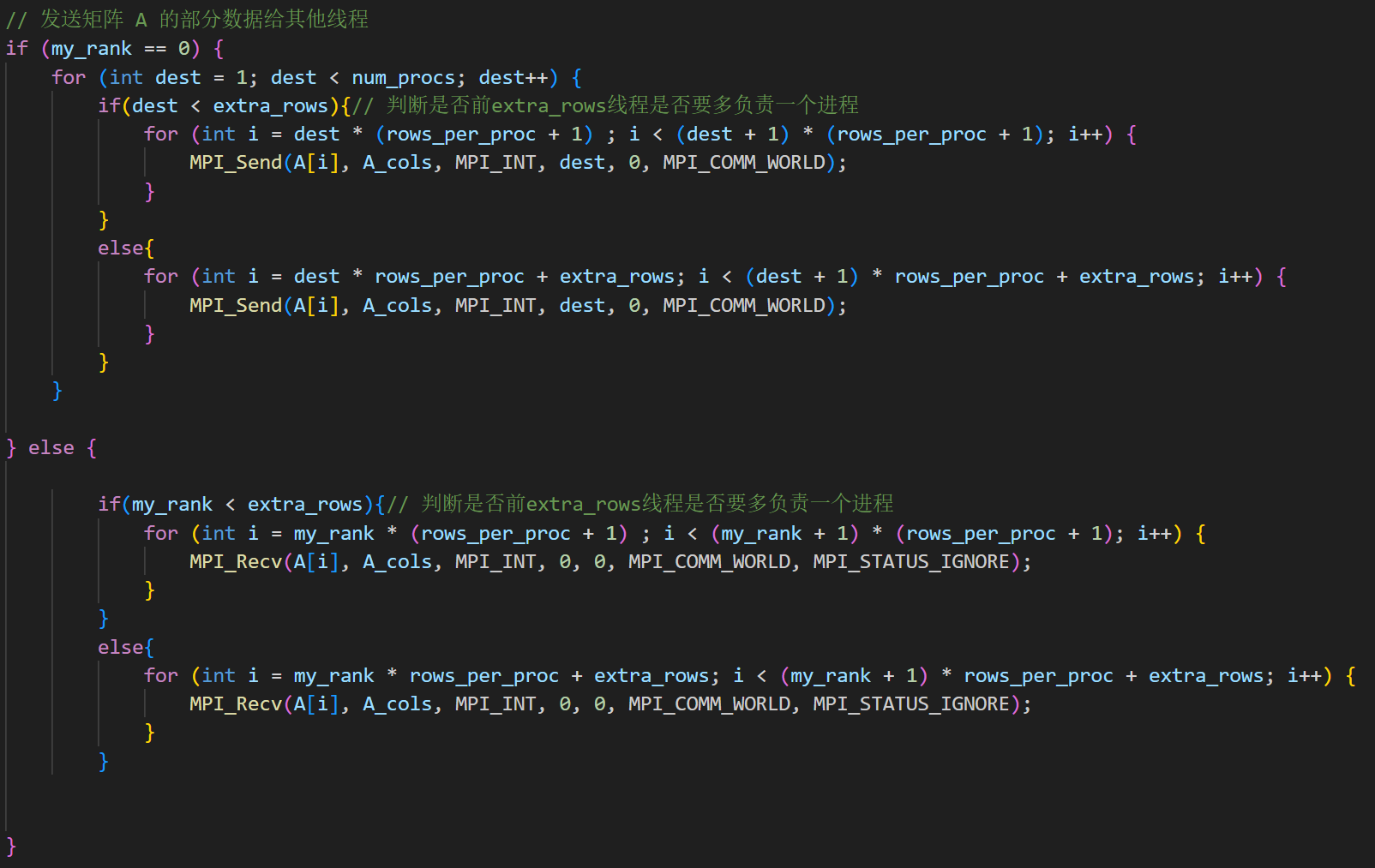


## 2.3 计算每个进程平均负责的A 的行数并初始化矩阵A、B

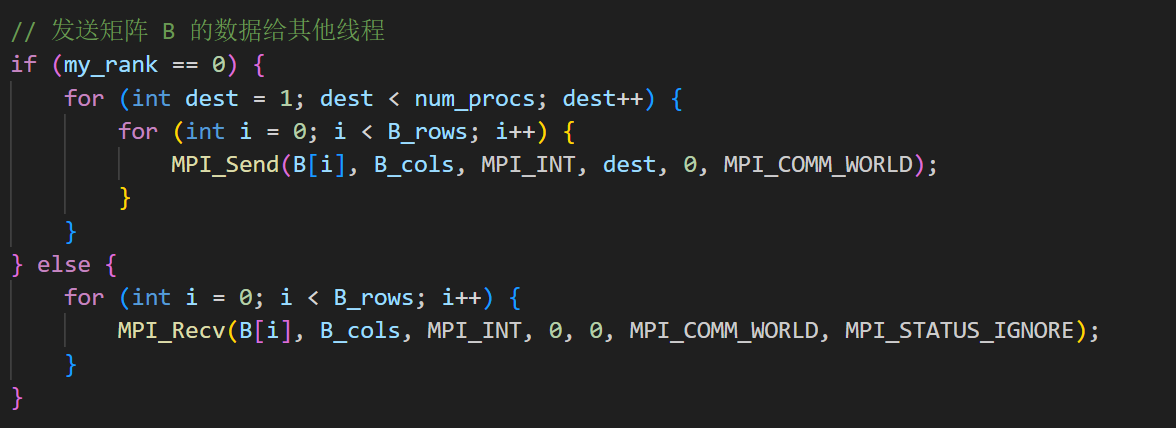


## 2.4 将每个进程负责矩阵A的对应行发送出去

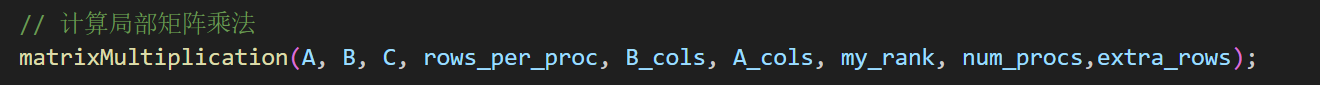
需要注意行数除进程数除不尽的情况,前extra\_rows的进程要多负责一行

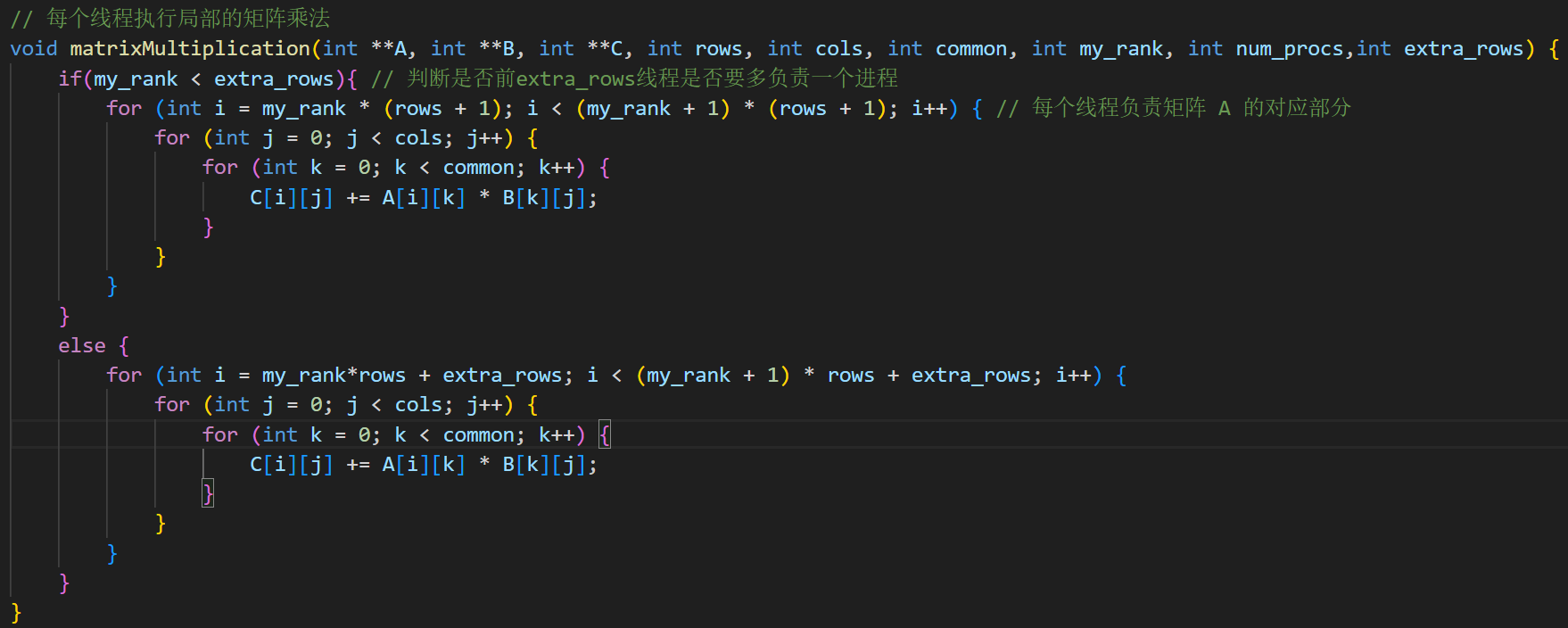


## 2.5 将整个矩阵B发给其他进程

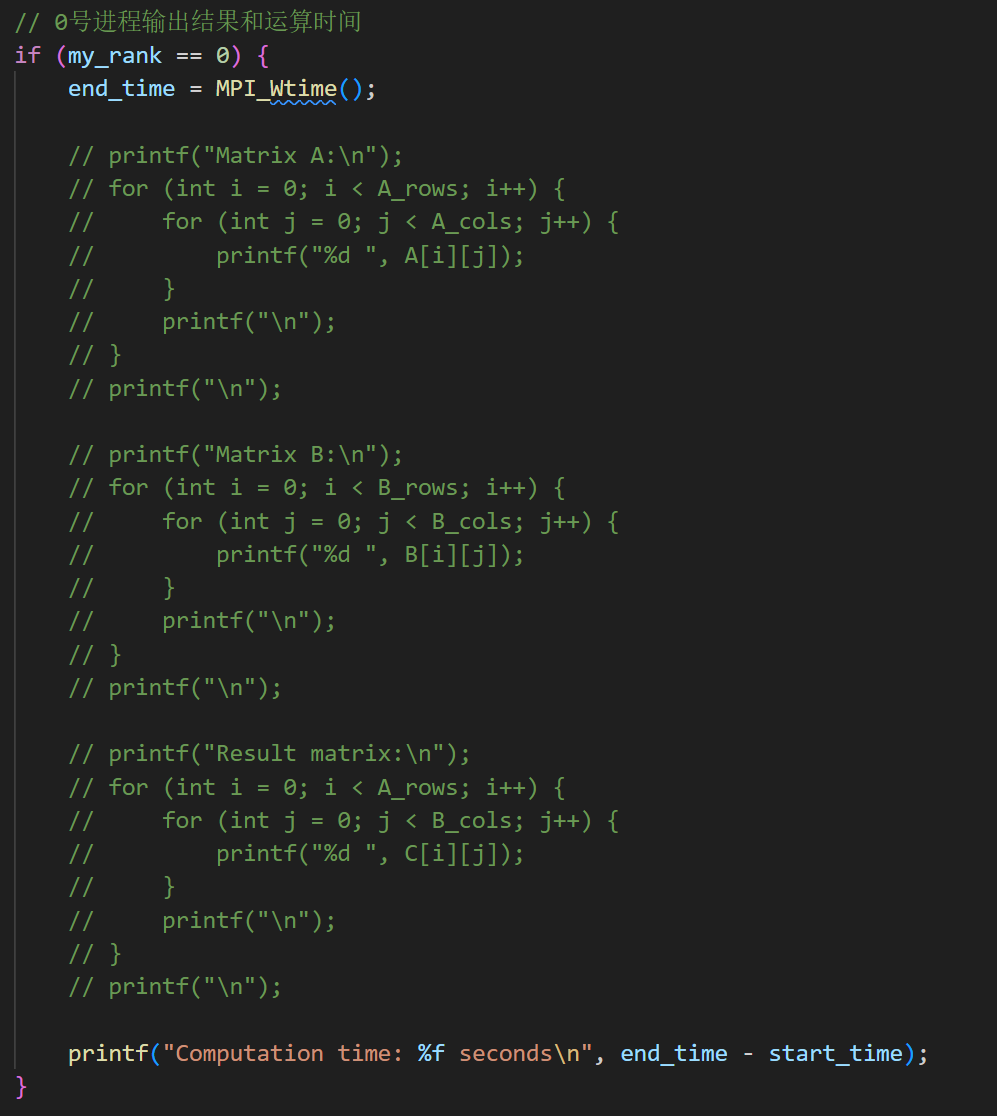


## 2.6 每个进程执行矩阵乘法

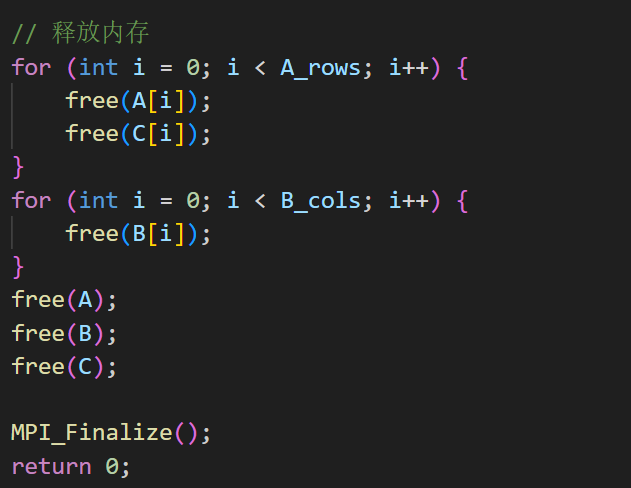




## 2.7 0号进程输出结果

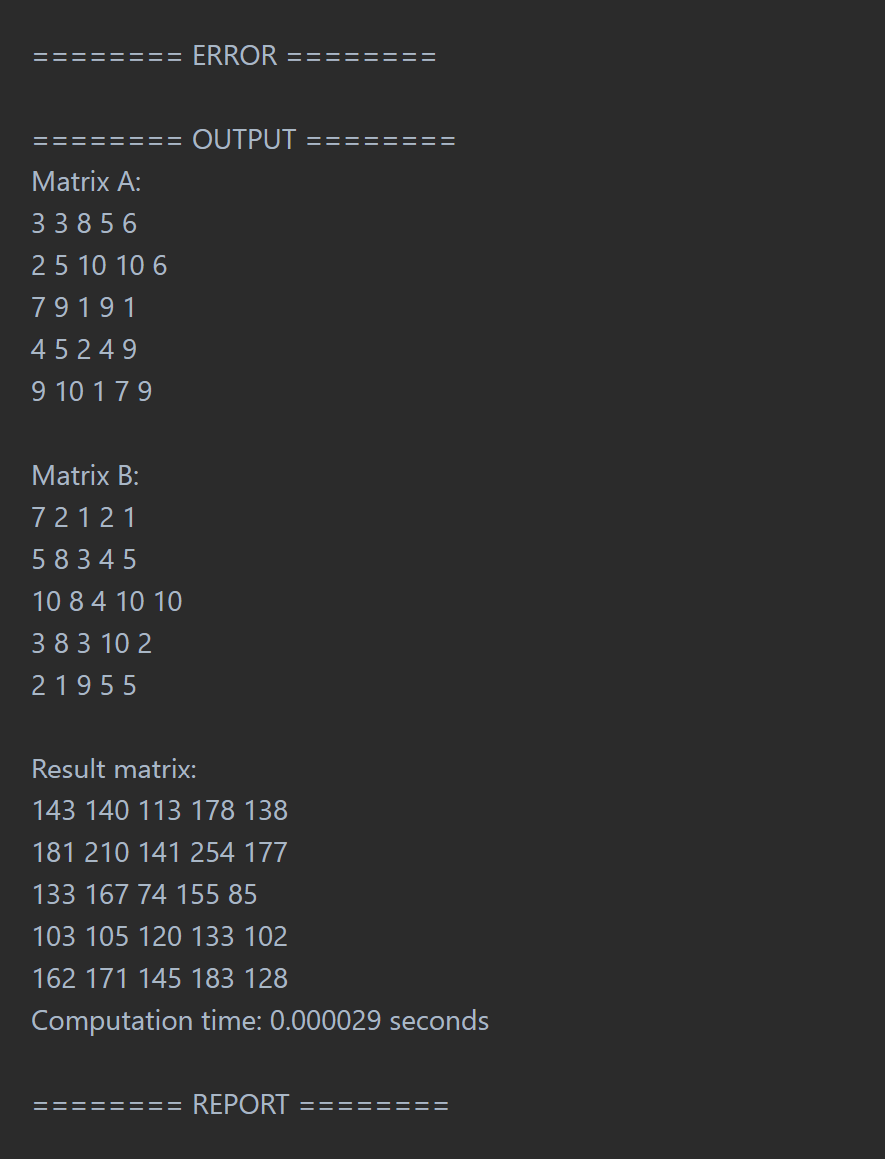


## 2.8 MPI\_Finalize()并释放内存



# **实验结果**

## 实验结果正确性验证(四个进程)

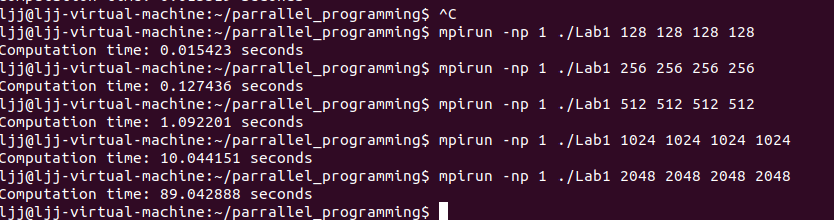


可以看到计算结果正确

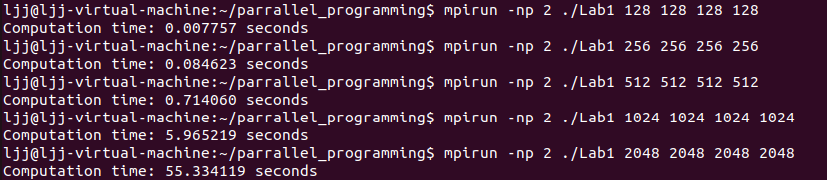
## 运行时间结果

由于虚拟机资源有限，最多可以允许4个进程

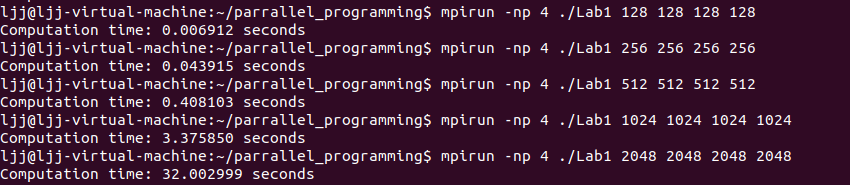
### 1个进程



### 2个进程



### 4个进程



## 3.3 综合对比

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **进程数** |  | | | | |
| 128 | 256 | 512 | 1024 | 2048 |
| 1 | 0.015423s | 0.127436s | 1.092201s | 10.044151s | 89.042888s |
| 2 | 0.007757s | 0.084623s | 0.714060s | 5.965219s | 55.334119s |
| 4 | 0.006912s | 0.043915s | 0.408103s | 3.375850s | 32.002999s |

# **4.实验感想**

## 4.1 感想

通过实验，我深入理解了 MPI 点对点通信的概念和实现，并学会了如何利用 MPI 实现并行矩阵乘法。

在观察不同进程数量和矩阵规模下程序的运行时间后，我发现并行计算能够有效提高计算效率，特别是在处理大规模矩阵时。通过分析并行性能，我了解到在选择进程数量时需要考虑到计算节点的数量、通信开销等因素，并且需要权衡计算能力和通信开销之间的关系。

## 4.2 讨论

## 4.2.1在内存受限情况下，方法进行大规模矩阵乘法计算的方法：

①使用分布式存储：将矩阵分块存储在不同的计算节点上，通过并行计算和通信来实现大规模矩阵乘法，减少单个节点的内存压力。

②压缩存储：对矩阵进行压缩存储，例如利用稀疏矩阵的特点进行存储，减少内存占用，从而提高大规模矩阵乘法计算的效率。

**4.2.2 提高大规模稀疏矩阵乘法性能的方法包括：**

①矩阵分块：将稀疏矩阵划分为若干块，只对非零元素进行计算，减少计算量，从而提高计算效率。

②使用专门的稀疏矩阵存储格式：选择合适的稀疏矩阵存储格式，例如压缩稀疏矩阵存储格式，可以减少存储空间，并提高计算效率。