

## “并行编程原理与实践”课程实验二、

### Openmp 实验指南

#### 一、实验目的

熟悉利用 Openmp 并行化技术，实现基础串行代码，利用 Openmp 实现算法的并行化处理。

#### 二、实验内容

针对以下问题，实现 Openmp 的串行化算法和并行化算法。

##### 1、稠密矩阵乘法 GEMM

GEMM 算法是一种矩阵乘法，其名字来源于 “General Matrix Multiply”。这个算法的作用是计算矩阵的乘积。在计算机视觉、深度学习和自然语言处理等领域，矩阵乘法是非常常见的操作。下面将介绍 GEMM 算法的实现步骤。

##### （1）矩阵乘法的定义

矩阵乘法是一种特殊的矩阵运算，其定义如下：

$$C = A * B$$

其中，A、B、C 均为矩阵。A 的列数必须与 B 的行数相等，否则无法进行矩阵乘法。

##### （2）矩阵乘法的算法

矩阵相乘的常用算法是基于三重循环的实现方式。假设 A 的大小为  $m \times k$ ，B 的大小为  $k \times n$ ，C 的大小为  $m \times n$ 。

##### 2、快速排序

快速排序（Quick Sort）是 C. A. R. Hoare 在 1960 年提出。它应用分治思想进行排序，通过对数据进行分区操作，并递归地对分区后的子

序列进行排序，从而达到整个序列有序的目的。

#### (1) 基本思想：

快速排序的核心思想是在待排序序列中选择一个基准值（pivot），然后将小于基准值的元素放在基准值的左边，大于基准值的元素放在基准值的右边，这样就找到了基准值在数组中的正确位置。之后，再分别对基准值左右两边的子序列进行同样的操作，直到整个序列有序。

#### (2) 排序流程：

快速排序算法通过多次比较和交换来实现排序，其排序流程大致如下：

a)选择基准值：在待排序序列中选取一个元素作为基准值。

b)分区操作：通过一趟排序将待排序的数据分割成独立的两部分，其中一部分的所有数据都比另一部分的所有数据要小，然后再按此方法对这两部分数据分别进行快速排序，整个排序过程可以递归进行，以此达到整个数据变成有序序列。

c)递归排序：对基准值左右两边的子序列递归地执行上述分区操作，直到子序列的长度为 1 或 0，即已经有序。

### 三、实验环境

代码实现采用 C 语言编程环境，Linux 编程环境。

### 四、实验步骤

实现上述程序代码，分析程序中可进行并行化部分，利用并行计算的主要模式，思考并行化的思路。

### 五、实验报告

实现实验分析报告。