**《并行计算》课程实验报告**

**实验2：基于共享内存的并行程序设计**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 |  | | 院系 | |  | | | 学号 | |  | |
| 任课教师 | |  | | | | 指导教师 |  | | | | |
| 实验地点 | |  | | | | 实验时间 |  | | | | |
| 实验课表现 | | 出勤、表现得分 | |  | | 实验报告  得分 |  | | 实验总分 | |  |
| 操作结果得分 | |  | |
| **一、实验目的** | | | | | | | | | | | |
| 1. 了解华为云环境的使用过程 2. 熟练掌握C++语言； 3. 掌握PThread 开发多线程程序； 4. 掌握PThread 中互斥机制的使用方式 5. 掌握 OpenMP 的基本功能、构成方式、句法； 6. 掌握 OpenMP 体系结构、特点与组成； 7. 掌握采用 OpenMP 进行多核架构下多线程编程的基本使用方法 | | | | | | | | | | | |
| **二、实验内容** | | | | | | | | | | | |
| 实验2.1 华为云实验环境说明   1. 购买并配置服务器 2. 使用CloudShell远程登录ESC 3. 创建用户       实验2.2并行环境下Pthread程序的编译和运行   1. hello-world程序 2. 创建程序源码            1. 编译和执行      1. 利用中值积分定理计算Pi值 2. 创建程序源码        1. 编译和运行      1. 互斥量的使用 2. 创建程序源码            1. 上传输入文件InFile1.txt到Pthread目录下 2. 编译和运行     实验2.3 并行环境下OpenMP程序的编译和运行   1. hello-world程序 2. 创建程序源码          1. 编译和执行      1. 利用中值积分定理计算Pi值 2. 创建程序源码 | | | | | | | | | | | |
| 1. 编译和执行      1. 在源程序代码的主程序体中进行omp方式优化            1. PI值蒙特卡洛算法 2. 创建串行代码          1. 编译和执行      1. 在源程序代码的主程序体中进行omp方式优化 | | | | | | | | | | | |
| **三、实验结果** | | | | | | | | | | | |
| 实验2.2并行环境下Pthread程序的编译和运行   1. hello-world程序      1. 利用中值积分定理计算Pi值      1. 互斥量的使用     实验2.3 并行环境下OpenMP程序的编译和运行   1. hello-world程序      1. 利用中值积分定理计算Pi值     在源程序代码的主程序体中进行omp方式优化后：     1. PI值蒙特卡洛算法     在源程序代码的主程序体中进行omp方式优化后： | | | | | | | | | | | |
| **四、思考题** | | | | | | | | | | | |
| 思考题1：串行程序采用PThread进行线程化基本步骤？   1. 定义线程函数。 2. 创建线程。 3. 等待线程结束 4. 同步线程   思考题2：PThread有几种共享数据互斥机制可以采用，如何应用？   1. 互斥锁：用于保护共享资源，确保同一时间只有一个线程可以访问 2. 条件变量：用于线程间的协调，当特定条件发生时通知线程 3. 读写锁：允许多个读线程同时访问，但写线程会独占访问 4. 信号量：控制对共享资源的访问数量   思考题3：OpenMP的主要功能，基本构成体有哪些？  OpenMP主要功能：可将代码块标记为并行执行；可将循环或代码段的执行分配给多个线程；可管理线程的私有数据和共享数据；提供线程同步机制；创建可并行执行的任务；控制循环迭代的分配方式；提供锁定机制；通过环境变量控制并行行为；  OpenMP的基本构成体包括：指令、子句、运行时库、环境变量、编译器支持  思考题4：试分析如何使用OpenMP实现多线程并行运算，提高系统运算效能，其引入环节应如何选取？   1. 环境准备，设置环境变量来控制线程数量 2. 使用#pragma omp parallel指令来标记并行区域 3. 进行工作分担，可将循环或代码段的执行分配给多个线程 4. 进行数据管理 5. 同步机制 6. 任务并行，创建可并行执行的任务。 7. 编译指令   引入环节的选取：循环是并行化的首选、对于可分解为多个独立数据块处理的算法进行数据并行、先对程序中计算量最大的部分进行并行化再逐步扩展到其他部分  思考题5：对本次实验最开始时的pthread\_hello\_world.c修改后编译并多次运行程序，记录线程执行顺序，分析线程程序执行顺序是否不可预见及其产生原因。  线程的执行顺序是不可预见的；  产生原因：操作系统调度；如果处理器有多个核心，线程会在不同的核心上并行执行；线程在执行过程中可能会因种原因被挂起或阻塞，影响线程的执行顺序；线程的执行还受到处理器缓存和内存访问速度的影响。  思考题6：根据你的实验，OMP\_NUM\_THREADS是不是设置得越大越好？说明理由。  不是。线程数不应超过处理器的物理核数，线程数过多会导致频繁的上下文切换，消耗CPU时间。 | | | | | | | | | | | |
| **五、实验心得体会** | | | | | | | | | | | |
| 在这次并行编程实验中，学习了PThread和OpenMP两种并行技术。通过编写和运行hello-world程序、计算Pi值以及使用互斥量，认识到合理选择线程数量和优化并行策略的重要性。并行化可以显著提高程序性能，但也需要注意线程同步和资源竞争问题。 | | | | | | | | | | | |
| 指导教师评语：  日期： | | | | | | | | | | | |