山东大学 软件 学院

**操作系统** 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202200300092 | 姓名：郑睿杰 | | 班级：大数据2班 |
| 实验编号：实验2 | | | |
| 实验题目：进程控制实验 | | | |
| 实验学时：4 | | 实验日期：2024.4.23 | |
| 实验目的：  通过 Linux 系统中管道通信机制，加深对于进程通信概念的理解，观察和体验并发进程间的通信和协作的效果 ，练习利用无名管道进行进程通信的编程和调试技术 | | | |
| 硬件环境：  机型：ThinkBook-13x-ITG  CPU：11th Gen Intel(R) Core(TM) i7-1160G7 @ 1.20GHz  内存：16150988 kB | | | |
| 软件环境：  Ubuntu 9.4.0-1 ubuntu1~20.04.2 + vscode2022 | | | |
| 实验步骤与内容：  【实验模型描述】  1.在本实验中，使用了C语言编写的程序，通过 fork() 函数创建了两个子进程，分别用于计算 f(x) 和 f(y) 的值。  2.父进程使用管道进行与子进程之间的通信，将子进程计算得到的结果传递给父进程，父进程则进行 f(x,y) 的计算。  【主要算法代码分析】      1.首先，在主函数中通过 scanf() 函数获取用户输入的 x 和 y 的值。  2.创建了一个管道 pipefd，用于父子进程之间的通信。  3.使用 fork() 函数创建两个子进程，分别用于计算 f(x) 和 f(y)。  4.子进程1计算 f(x)，并将结果通过管道写入到管道中。  5.子进程2计算 f(y)，并将结果通过管道写入到管道中。  6.父进程关闭了管道的写端，并从管道中读取子进程计算得到的 f(x) 和 f(y) 的结果。  7.父进程等待子进程结束，然后对 f(x) 和 f(y) 的结果进行加法操作，得到f(x,y) 的结果，并输出。  【运行结果】    任意输入x,y，输出执行f(x),f(y),f(x,y)的结果  【实验结果的分析综合】  通过实验结果可以看出，子进程1计算得到的 f(x) 结果为 6，子进程2计算得到的 f(y) 结果为 5，父进程将这两个结果相加得到了f(x,y) 的值为 11。这说明通过使用进程间通信的方式，可以实现对多个函数的并行计算，并在父进程中进行进一步的整合处理，提高了程序的效率和性能。 | | | |
| 结论分析与体会：  【实验中遇到的问题和解决方法】  1. 进程间通信问题：初期实现时，存在数据在子进程和父进程间传输不一致的问题。这是由于对管道使用不熟悉，未能正确管理管道的开闭。解决方法是确保每个进程在适当的时候关闭了不需要的管道端。例如，子进程在写入数据后应立即关闭写端，父进程在读取完数据后关闭读端。  2. 管道阻塞问题：由于父进程的读取操作是阻塞的，如果子进程的写操作没有完成，则父进程会一直等待，导致程序假死。解决方法是通过在父进程中合理安排关闭写端的时机，确保所有数据写入完成后再进行读取操作。  【实验结果达到设计目标的程度】  实验成功实现了设计的目标，即通过进程间通信，实现了对两个不同数学函数的并行计算，并能够将计算结果正确返回给父进程进行汇总处理。最终的程序输出符合预期的计算结果。  【还可以进行哪些改进】  1. 性能优化：当前使用的是递归方式计算阶乘和斐波那契数列，这对于较大的输入值效率低下。可以改进为使用循环或者动态规划的方法来提高计算效率。  2. 错误处理：目前的程序中，对于管道创建失败或者进程创建失败的错误处理较为简单。可以增加更详细的错误信息输出，帮助定位问题所在。  3. 资源管理：增加更严格的资源管理，例如在子进程异常退出时能够及时释放已分配的资源，避免资源泄露。  【实验得到的收获和启发】  1. 并行计算的实用性：实验中通过子进程的并行工作，提高了计算的效率，这对于解决实际问题中遇到的大规模计算问题具有重要的启发意义。  2. 系统编程能力：通过实践学习了 UNIX 系统下的进程创建、进程间通信等系统编程知识，加深了对操作系统工作原理的理解。  3. 问题解决能力的提升：在遇到问题时，通过查阅资料和反复调试，提升了独立解决问题的能力，也锻炼了编程和逻辑思维能力。 | | | |