山东大学 计算机科学与技术 学院

操作系统 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：201705130113 | 姓名：黄瑞哲 | | 班级：计科17.3 |
| 实验题目：线程和管道通信实验 | | | |
| 实验学时：2 | | 实验日期：2019.10.28 | |
| 实验目的：  通过 Linux 系统中线程和管道通信机制的实验，加深对于线程控制和管道通信 概念的理解，观察和体验并发进/线程间的通信和协作的效果 ，练习利用无名管道 进行进/线程通信的编程和调试技术。 | | | |
| 硬件环境：  Intel Core i5-8300H | | | |
| 软件环境：  Windows 10 Pro 1903  Docker for Windows  Ubuntu 18.04  VS Code | | | |
| 实验内容与设计：      pthread\_create 函数创建一个新的线程      函数 pthread\_join 用于挂起当前线程，直到 th 指定的线程终止运行为止。通过在pthread\_create调用创建一个线程时使用PTHREAD\_CREATE\_DETACHED属 性或者使用 pthread\_detach 可以让线程处于被分离状态。      pipe 建立一个无名管道，pipe\_id[0]中和 pipe\_id[1]将放入管道两端的描述符。其中pipe\_id[0]为读，pipe\_id[1]为写。      read 和 write 分别在管道的两端进行读和写。 pipe\_id 是 pipe 系统调用返回的管道描述符。Buf 是数据缓冲区首地址，count 说明数据缓冲区以 size\_t 为单位的长度。read 和 write 的返回值为它们实际读 写的数据单位。其中通信方式默认为同步读写，如果管道中没有数据则读者堵塞直至数据到达，如果管道有数据则写者阻塞直至数据取走。  独立实验： | | | |
| 结论分析与体会：  在引入线程的操作系统中，线程是调度的基本单位，进程是资源分配的基本单位。线程可以访问其附属进程的资源。进程间通信需要进程同步和互斥手段的辅助，以保证数据的一致性，而线程间可以直接读写进程的数据段来进行通信。  管道pipe 是进程间通信最基本的一种机制,两个进程可以通过管道一个在管道 一端向管道发送其输出,给另一进程可以在管道的另一端从管道得到其输入.管道以 半双工方式工作,即它的数据流是单方向的。  在独立实验我使用了两种写法，分别是线程与进程，其结果如下（上图：线程 下图：进程）      使用线程的时候首先建立4个无名通道和3个线程。1号线程负责计算函数和，2好线程负责计算斐波那契，3号线程负责计算阶乘。4个管道中有两个负责1号线程与2号线程通信，一条1号线程写x，2号线程读x；另一条2号线程写答案，1号线程读答案，另外两条同理，负责1号线程与3号线程的通信。  使用进程的时候也是要建立4个无名通道然后在fork出两个子进程，分别负责斐波那契和阶乘的计算，其余与线程相同。但是值得注意一点的是，由于子进程先继承后分离的特性，在每个子进程被创建的时候会继承父进程的管道，因此需要关闭无用管道的端，避免数据传递错误。 | | | |