

实验二 进程创建及进程间通信

一、实验目的

1. 掌握 Linux 进程的创建方法，加深对进程概念的理解，明确进程和程序的区别。
2. 认识进程并发执行的实质。
3. 学习控制进程同步的方法。
4. 分析进程竞争资源的现象，学习解决进程互斥的方法。
5. 了解管道通信的特点，掌握管道通信的使用方法。

二、实验内容

1. 利用 fork 函数创建进程。
2. 利用 wait() 来控制进程执行顺序, wait() 给我们提供了一种实现进程同步的简单方法, 。
3. 用 lockf() 来给每一个进程加锁, 以实现进程之间的互斥。
4. 用 pipe() 来实现进程的管道通信。

三、实验步骤

1. 进程的创建。编写一段 C/C++ 程序, 使用系统功能调用 fork() 创建两个子进程。当此程序运行时, 在系统中有一个父进程和两个子进程活动。让每一个进程在屏幕上显示一个字符; 父进程显示字符 “a”, 子进程分别显示字符 “b” 和 “c”。试观察记录屏幕上的显示结果, 并分析原因。
2. 修改 1. 的程序, 要求子进程先输出, 然后父进程再输出。
3. 修改 1. 的程序, 每一个进程循环十次, 每次循环: 子进程 1 将 “daughter i” 输出到文件 to_be_locked.txt 文件中, 子进程 2 将 “son i” 输出到文件 to_be_locked.txt 文件中, 父进程将 “parent i” 写入到文件 to_be_locked.txt 文件中。用 lockf() 来给每一个进程加锁, 以实现进程之间的互斥。使用 cat

to_be_locked.txt 查看输出结果，并分析原因。

4. 编写程序：实现进程的管道通信，用系统调用 `pipe()` 建立一管道，两个子进程 P1 和 P2 分别向管道各写一句话：

Child 1 is sending a message!

Child 2 is sending a message!

父进程从管道中读出两个来自子进程的信息并显示（要求先接收 P1，后 P2）。