《操作系统》课程实验报告

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 实验名称 | Linux安装及虚拟机的使用 | | | | | 实验序号 | 1 |
| 姓 名 | 陈小贤 | 系院专业 | 信管 | 班 级 | 18 | 学 号 | 10207180583 |
| 实验日期 | 2012年12月 | | 指导教师 | 丘文峰 | | 成 绩 |  |
| **一、实验目的和要求**  1.加深对进程概念的理解，  2.明确进程和程序的区别  3.了解并发执行的实质，分析进程争用资源的现象  4.尝试学习解决进程互斥的方法  5.了解Linux系统中进程通信的基本原理 | | | | | | | |
| **二、实验预习内容**  1.阅读Linux的sched.h源码文件  2.阅读Linux的fork.c源码文件  3.《实验指导书》 | | | | | | | |
| **三、实验设备**  1. 操作系统：Linux red hat  2. 安装软件： VirtualBox、red hat | | | | | | | |
| **四、实验内容及步骤**  1.进程的创建  编写一段程序，使用系统调用fork()创建两个子进程。当此程序运行时，在系统中有一个进程 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"和字符"c"。试观察记录屏幕上的显示结果，并分析原因。  2进程的控制  修改已编写的程序，将每个进程输出一个字符改为每个进程输出一句话，在观察程序行时屏幕上 出现的现象，并分析原因。  3.如果在程序中使用系统调用lockf()来给每一个进程加锁，可以实现进程之间的互斥，观察并分析 出现的现象    4. 进程的管道通信  编制一段程序，实现进程的管道通信。使用系统调用pipe()建立一条管道栈；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!  而父进程则从管道中读出来自于两个子进程的信息，显示出屏幕上。 要求父进程先接  进程P1发来的消息，然后再接收子进程P2发来的消息。 | | | | | | | |
| **五、实验结果、分析与心得**  **1.实验结果**  **1.** 进程创建  将while(p1=fork()==-1)语句改为while(（p1=fork()）==-1)语句后  2.进程控制    2.    3.加锁后  1.  .              4.任务2  5.通道通信  **2．分析结果**  1.由于执行第一句while(p1=fork()==-1)，大多数情况都会创建了一个子进程，然而子进程的fork（）函数返回0，父进程的fork（）函数返回子进程的id号.>0,因此经过while(p1=fork()==-1)后p1=0,则当子进程和父进程并发执行时都是只执行 if (p1==0) putchar('b')语句，输出两个b，并不会在父进程种2创建第二个子进程；有只输出一个b的情况，应该是子进程创建失败，只执行父进程，且不会再创建第二个子进程，只执行一次if (p1==0) putchar('b')语句输出一个b。如果要创建两个子进程，并输出bca，则应该在第一句while(p1=fork()==-1)语句改为while(（p1=fork()）==-1)语句。  改进之后：  输出的序列既可以是bca，bac等序列是由于三个进程间随机并发执行的原因，而出现ba，ca等序列是由于第一个子进程和第二个子进程没有被成功建立的结果。    2.输出的child、son、daughter的顺序就是随机的了，原因是输出child的子进程，daughter的子进程都已被建立，则三个进程之间是并发执行的。    3.child、son、daughter之间的程序是随机切换的，但是每当执行任何一个进程时都会一次执行到底，不会执行到一半就由于并发转去执行另外一个进程，这是由于对每个进程进行了加锁的结果。由于加锁，使得各进程间的执行时互斥的，因此才不会像进程控制那样，不但进程间的切换时随机的，进程之间的执行步骤也是交叉并被执行的。    **3．心得**  敲写代码要细心。 | | | | | | | |
| **教师评语：**  **成绩:**  **教师签字：** | | | | | | | |