**《操作系统》实验报告**

实验序号：6　　　　　　　　实验项目名称：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 学　　号 | 1910705127 | 姓　　名 | 宋宇昕 | 专业、班 | 19软件1班 | |
| 实验地点 | 精工楼 | 指导教师 | 李远敏 | 实验时间 | 12.08 | |
| 一、实验目的及要求  1.加深对进程同步与通信概念的理解。  2.学习并掌握利用信号量进行相关编程，熟悉函数semget（）、semop（）和semctl（）的使用。  3.学习并掌握利用消息队列进行通信操作，熟悉函数msgget()、msgsnd()、msgrcv()、msggctl()等与消息有关的操作调用  4.学习并掌握利用共享内存进行相关应用，熟悉函数shmget（）、shmat（）、shmdt()和shmctl（）的使用二、实验设备（环境）及要求  1.搭建虚拟机VMware Workstation下的Ubuntu操作系统环境  2.搭建Ubuntu操作系统下的C语言编程环境，并用C语言实现N!程序，编译好并运行，采用函数调用的形式，模块化编程思想。  三、实验内容与步骤  实验一：信号量的使用  1.信号量操作：创建、设置、PV操作及删除。  2.使用vi将程序清单6-1程序输入。  3.利用gcc编译这个程序，写出编译命令和执行结果；如果不成功，尝试利用gdb调试。  4.运行6-1编译后的目标程序。  5.在主程序中注释中“semaphore\_p()”和“semaphore\_v()”两处语句，程序的运行结果有什么不同  6.仔细阅读程序，并画出程序流程图。  7.修改程序：将本程序中包含父子进程一个程序改为由两个程序来实现相同的功能  实验二：消息队列运用  1.使用vi将程序清单6-2-1程序输入。  2.使用vi将程序清单6-2-2程序输入  3.利用gcc分别编译这个程序，写出编译命令和执行结果；如果不成功，尝试利用gdb调试。  4.在后台运行6-2-1编译后的程序，在前台运行6-2-2编译后的目标程序。  5.仔细阅读程序，并画出程序流程图。  6.将本项目中两个程序合并为包含父子进程的一个程序来实现相同的功能  实验三：信号量和共享内存综合运用  父进程循环随机产生大写字母，并通过共享内存传递给子进程，子进程读取到该字母后将其转换为相应的小写字母并将该小写字母传递给父进程，最后父进程输出读取到的小写字母。该程序说明了共享内存和信号量结合的应用。  1. 使用vi将程序清单6-3程序输入。  2. 利用gcc编译这个程序，写出编译命令和执行结果；如果不成功，尝试利用gdb调试。  3.运行12-1编译后的目标程序的结果。  4.仔细阅读程序，并画出程序流程图。  5.程序分析：1）试说明父子进程都包括读和写，但它们的顺序有什么不同？  2）父进程读写分别各用一次信号量互斥访问，而子进程读写共用一次型号量互斥访问，试说明原因？  3）共享内存的第二个内存用特殊字符“#”或“@”有什么用途？  四、实验结果与数据处理  实验一：信号量的使用   1. 实验结果        1. 结果分析   运行结果中可以看到分别打印X和O，最后子进程和父进程分别结束打印出相应标志。  原因：我们从这个程序中可以看到，由于我们在调用进程的主函数里面加入了信号量控制的相关代码，在程序执行的过程中对这一段代码是互斥访问的，即一个进程执行完毕只有另一个进程才开始执行，所以我们就看到了X和O是交替打印的，这就是信号量互斥的原因。   1. 程序修改     因为我们没有使用信号量控制，所以我们的程序中循环打印语句执行的结果不像之前那样稳定，X和O的输出是随机的，并没有特殊的规律，说明两个进程在不断的争夺统一资源。   1. 总结：   由于信号量，只能进行两种操作，等待和发送信号，即P（s）和V（s），他们的行为是这样的。  P(s):如果s的值大于零，就给他减1，如果它的值为零，就挂起该进程的执行。  V(s)如果有其他进程，因等待s为挂起，就让他恢复执行。如果没有进程，因等待s而挂起，就给他加1。   1. 流程图     实验二：消息队列运用   1. 实验结果        1. 分析   消息队列提供了一种从一个进程向另一个进程发送一个数据块的方法，每个数据块都被认为含有一个类型接收进程，可以独立的接收有不同类型的数据结构，我们可以通过发送消息来避免命名管道的同步和阻塞问题。但是消息队列和命名管道一样，每一个数据块都有一个最大的限制。  消息队列和命名管道有不少的相同之处，通过与命名管道相同消息队列进行通讯的进程是不相关的进程，同时他们都是通过发送和接收的方式来传递数据的，在命名管道中发送数据用write，接收数据用read。则在消息队列中发送数据用msgsnd，接受数据用msgrcv，而且他们对每个数据都有一个最大的限制。   1. 流程图     实验三：信号量和共享内存综合运用   1. 实验结果        1. 分析   由程序代码我们可知，由父进程循环中随机产生大写字母，同时我们将产生的这些大写字母写入共享内存，然后再由子进程，从共享内存中读取该字母。然后将这个字母转写为小写字母，同时将这个小写字母写入共享内存中，再由父进程从共享内存中将这个小写字母读出，并输出在屏幕上。   1. 问题   ①在本实验中，由于子进程时先进行读取共享内存的操作，然后再进行写入的操作，而父进程是先进行写入共享内存的操作，然后再进行读取的操作，我们根据程序可知，父进程向共享内存中写入数据，然后子进程收到并修改之后，这时候父进程再次读取并显示在屏幕上，也就是说父进程的写入程序要在子进程的读程序之前，而父进程的读程序要在子进程的写程序之后，这样才能保证达到效果。  ②这是因为父进程在写完之后，根据程序我们可知父进程不能马上进行读取操作，要等到释放信号量之后，才能让子进程进行读取的操作并修改，所以我们根据这一现象可知，父进程的信号量要控制，要读写分离才合适。  ③从程序和打印结果来看，我们可知共享内存中的#或者@。是用来表示当前字符的大写或小写。#号表明当前字符为小写，那么这个字符就可以输出显示，@表示当前的字符为大写，这样字符就根据子进程修改则操作这样让区分程序逻辑性更强。   1. 流程图     五、分析与讨论  对上机实践结果进行分析，上机的心得体会。 | | | | | | |
| 六、教师评语  签名：  日期： | | | | | | 成绩 |