山东大学 软件 学院

**操作系统** 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202000300125 | 姓名：贾星宇 | | 班级：2020级5班 |
| 实验编号：实验四 | | | |
| 实验题目：进程同步实验 | | | |
| 实验学时：4 | | 实验日期：2022年4月30日 | |
| 实验目的：  加深对并发协作进程同步与互斥概念的理解，观察和体验并发进程同步与互斥 操作的效果，分析与研究经典进程同步与互斥问题的实际解决方案。了解 Linux 系 统中 IPC 进程同步工具的用法，练习并发协作进程的同步与互斥操作的编程与调试 技术。 | | | |
| 硬件环境：  宿主机：lntel(R) Core(TM) i5-10210U CPU @ 1.60G Hz 2 .1 1 G Hz  虚拟机：ubuntu-20.04.1-desktop-amd64 | | | |
| 软件环境：  宿主机：Windows 10 家庭中文版  虚拟机：Code::Blocks | | | |
| 实验步骤与内容：  1.分析并编写示例实验代码，并运行:  ipc.h:    ipc.c              producer.c:        consumer.c:        运行结果如下：  首先在文件所在位置输入：  gcc -g -c producer.c ipc.c  gcc producer.o ipc.o -o producer  gcc -g -c consumer.c ipc.c  gcc consumer.o ipc.o -o consumer  随后在当前终端窗体中启动执行速率为 1 秒的一个生产者进程：  ./producer 1:    可以看到生产者进程在写入8个字节后因为缓冲区满而阻塞  随后打开一个每三秒执行一次的生产者进程：  ./producer 3:    因为缓冲区满，所以新打开的进程被阻塞  再打开另外两个终端窗体，进入当前工作目录，从中启动执行速率为 2 和 4 的两个消费者进程:  ./consumer 2  ./consumer 4    可以看到由于消费者进程读出了写入缓冲区的字符，生产者从新被唤醒继续向  读过的缓冲区单元中同步的写入字符。  利用ctrl+c中断两个生产者进程:    两消费者进程在读空缓冲区后而阻塞  在此打开两生产者进程，随后中断两消费者进程：  两生产者进程在写满缓冲区后而阻塞  经验证，发现实验结果和实验大纲结果一致。  2.分析独立实验要求：  抽烟者问题。假设一个系统中有三个抽烟者进程，每个抽烟者不断地卷烟并抽 烟。抽烟者卷起并抽掉一颗烟需要有三种材料：烟草、纸和胶水。一个抽烟者有烟 草，一个有纸，另一个有胶水。系统中还有两个供应者进程，它们无限地供应所有 三种材料，但每次仅轮流提供三种材料中的两种。得到缺失的两种材料的抽烟者在 卷起并抽掉一颗烟后会发信号通知供应者，让它继续提供另外的两种材料。这一过 程重复进行。 请用以上介绍的 IPC 同步机制编程，实现该问题要求的功能。  编写独立实验代码：  ipc.h与ipc.c同上  0：代表拥有烟草的抽烟者  1：代表拥有纸的抽烟者  2：代表拥有胶水的抽烟者  Producer.c:        Smokertoba.c      Smokerglue.c    Smokerpaper.c | | | |
| 结论分析与体会：  【对大纲代码的理解】  首先分析生产者代码consumer.c：  首先是一些基本的定义:  进程休眠时间（控制进程执行速度）、设置共享内存参数（缓冲区起始位置、长度、指针数、权限）、缓冲区首地址指针、放置产品位置指针、信号量（生产者/消费者 同步/互斥）等等  随后进入循环： 首先利用down函数请求操作，如果缓冲区满或者另一生产者在放置产品，则进程阻塞，如果没有，则--后此进程开始执行  执行过程是向缓冲区写入字符，字符按照‘A’加上指针数来形成B/C/D……  执行完后指针指向的存放位置下移，随后利用up函数++，告知本进程结束，阻塞的生产者或者消费者可以开始。  睡眠设定时间后重新开始上述流程。  消费者代码consumer.c的执行流程大致与上面相同，区别在于：  生产者执行前阻塞生产者，执行后唤醒生产者；  生产者执行前受缓冲区是否满影响；  生产者执行后唤醒消费者（告知缓冲区已有值）（缓冲区自己并没有信号量）  消费者执行前阻塞消费者，执行后唤醒消费者；  消费者执行前受缓冲区是否空影响；  消费者执行后唤醒生产者（告知缓冲区非满）  【独立实验分析】  独立实验与上述代码基本相同，区别在于：  生产者生产时要在0~2中生产，分别代表生产胶水纸、胶水烟草和纸烟草。生产了不同类后应该打开不同类的消费者进程  消费者进程在读取的时候要判断来源是否是自己需要的，然后在进行是否读取的判断  【结论分析】  1. 操作系统中提供的并发进程同步机制是怎样实现和解决同步问题的？  在Linux操作系统中，可以使用PV操作来解决同步问题。例如本次实验中的up和down。通过设置信号量为1，随后如果有进程开始进入临界区，则将信号量-1变为0；如果有其他进程尝试进入临界区，则会发现信号量为0，则会进入循环等待，直到其他进程释放临界区资源，将信号量++；  2.信号灯的初值：  初值代表了可用的资源量。例如本次实验中的缓冲区，设置了其初始值为8，代表了缓冲区可以存放8个大小的数值。  每次放入一个数值就将信号量减1，如果信号量为0代表存满，则进入阻塞状态。  每次读取信号量，就将信号量加1，如果有阻塞的写进程则被释放 | | | |