**中央财经大学**

**实 验 报 告**

**实验项目名称**

**所属课程名称**

**实 验 类 型**

**实 验 日 期**

**班 级**

**学 号**

**姓 名**

**成 绩**

**实验室**

|  |
| --- |
| **实验概述：** |
| **【实验目的及要求】**  **掌握Linux下生产者/消费者问题算法的实现**  **【实验原理】**  1.clone系统调用：  功能：创建一个轻进程或线程  用法：  int clone (int (\*fn)(void \*arg),void \*stack,int flag,void \*arg);  其中，  ★fn是轻进程所执行的函数；stack是轻进程所使用的栈  ★flag是CLONE\_VM,CLONE\_FS,CLONE\_FILES,CLONE\_SIGNAND,CLONE\_PID的组合  ★arg是调用过程的对应参数  clone的关键是flag的设定。  ☆CLONE\_VM表示子进程共享父进程的内存  ☆CLONE\_FS表示子进程共享父进程的文件系统  ☆CLONE\_SIGNAND表示子进程共享父进程的消息处理机制  ☆CLONE\_PID指子进程继承父进程的id号  2.sem\_wait和sem\_post系统调用：  sem\_wait(&s)和sem\_post(&s)分别相当于信号灯的P操作和V操作。其中，s是类型为sem\_t的信号灯。初始化函数sem\_init(s,0,8)。  3.pthread\_mutex\_lock和pthread\_mutex\_unlock系统调用：  pthread\_mutex\_lock(&mutex)和pthread\_rmutex\_unlock(&mutex)分别用于加锁和解锁。参数为mutex定义的互斥锁。初始pthread\_mutex\_init(&mutex,NULL)。  **【实验环境】（使用的软件）**  **Linux虚拟机，gcc编译器** |
| **实验内容：** |
| **【实验方案设计】**  利用clone系统调用实现线程创建、利用sem\_wait和sem\_post系统调用完成PV操作、利用pthread\_mutex\_lock和pthread\_mutex\_unlock实现进程互斥  **【实验过程】（实验步骤、记录、数据、分析）**  #define \_GNU\_SOURCE  #include "sched.h"  #include "pthread.h"  #include "stdio.h"  #include "stdlib.h"  #include "semaphore.h"  #define BUFFER\_LEN 8 //定义缓冲区长度  void producer(void \*args);  void consumer(void \*args);  pthread\_mutex\_t mutex;  sem\_t product;  sem\_t warehouse;  char buffer[BUFFER\_LEN][4];  int in,out; //产品进、出缓冲区指针  int tot=0; //缓冲区中的产品总数  main(int argc,char \*\*argv){  pthread\_mutex\_init(&mutex,NULL); //对互斥信号量进行初始化  sem\_init(&product,0,0); //对标识产品个数的信号量进行初始化  sem\_init(&warehouse,0,BUFFER\_LEN); //对标识缓冲区中空位置个数的信号量进行初始化  in=0;  out=0;  int i0=0,i1=1;  int clone\_flag, retval;  char \*stack;  clone\_flag=CLONE\_VM|CLONE\_SIGHAND|CLONE\_FS|CLONE\_FILES;  //创建两个生产者和两个消费者线程  stack=(char \*)malloc(4096);  retval=clone((void \*)producer,&(stack[4095]),clone\_flag,(void \*)&i0);  stack=(char \*)malloc(4096);  retval=clone((void \*)consumer,&(stack[4095]),clone\_flag,(void \*)&i0);  stack=(char \*)malloc(4096);  retval=clone((void \*)producer,&(stack[4095]),clone\_flag,(void \*)&i1);  stack=(char \*)malloc(4096);  retval=clone((void \*)consumer,&(stack[4095]),clone\_flag,(void \*)&i1);    exit(1);    }  void producer(void \*args){  int id=\*((int \*)args);  int i;  for(i=0;i<10;i++){  sleep(i+1);  sem\_wait(&warehouse);  pthread\_mutex\_lock(&mutex);  if(id==0)  strcpy(buffer[in],"aaa\0");  else  strcpy(buffer[in],"bbb\0");  printf("producer %d produces %s in %d\n",id,buffer[in],in);    in=(in+1)%BUFFER\_LEN;  tot++; //当前产品总数加1  printf("\*\*\*\*\*the number of products:%d\*\*\*\*\*\n",tot);  pthread\_mutex\_unlock(&mutex);  sem\_post(&product);  }  printf("producer %d is over!\n",id);  }  void consumer(void \*args){  int id=\*((int \*)args);  int i;  for(i=0;i<10;i++){  sleep(10-i);  sem\_wait(&product);  pthread\_mutex\_lock(&mutex);  printf("consumer %d get %s in %d\n",id,buffer[out],out);  out=(out+1)%BUFFER\_LEN;  tot--; //当前产品总数减1  printf("\*\*\*\*\*the number of products:%d\*\*\*\*\*\n",tot);  pthread\_mutex\_unlock(&mutex);  sem\_post(&warehouse);  }  printf("consumer %d is over!\n",id);  }  **【结论】（结果）**    **图一**    **图二**    **图三：程序一直运行** |
| **【小结】**   1. 第一次修改后：运行多次后，可以发现程序会出现死锁现象。分析原因如下：   在生产者函数中，sem\_wait(&warehouse)函数用于等待缓冲区中有空位置，pthread\_mutex\_lock(&mutex)函数用于保护缓冲区的访问。如果这两个函数的顺序交换，就会出现死锁的情况。当生产者线程A获取到互斥锁后，由于缓冲区已满，它会等待消费者线程来取走产品，此时线程A被阻塞。而消费者线程B由于缓冲区已空，它会等待生产者线程A来生产新产品，此时线程B也被阻塞。这样，两个线程都被阻塞，就出现了死锁现象。  因此，正确的顺序应该是先等待互斥锁，再等待信号量，即先调用pthread\_mutex\_lock(&mutex)，再调用sem\_wait(&warehouse)。这样可以避免死锁的发生。   1. 第二次修改运行结果可能会出现以下两种情况：    1. 程序一直运行，没有输出任何结果。这是因为生产者线程在等待仓库空间的时候需要获取互斥锁，而消费者线程在等待仓库有产品的时候也需要获取互斥锁。由于两个操作的顺序被交换了，所以生产者线程会占用互斥锁并等待仓库空间，而消费者线程则会占用互斥锁并等待仓库有产品，最终导致死锁。    2. 程序正常运行，但是输出的结果可能会出现生产者连续生产两个产品或消费者连续消费两个产品的情况。这是因为生产者线程在获取互斥锁之后会直接生产产品，而不会检查仓库是否有空间。同样的，消费者线程在获取互斥锁之后也会直接消费产品，而不会检查仓库是否有产品。这样可能会导致仓库满或空的情况，从而出现连续生产或消费的情况。   综上所述，正确的操作顺序应该是先获取互斥锁，再等待信号量。这样可以保证线程之间的同步和互斥，避免死锁和连续操作的情况。   1. 收获：   3.1 更深入地理解了线程同步和互斥的概念，以及信号量和互斥锁的应用。  3.2 学会了如何使用pthread库中的函数来创建线程，并使用信号量和互斥锁来实现线程之间的同步和互斥。  3.3通过实验中的错误操作，更加清晰地认识到了正确的操作顺序对于线程同步和互斥的重要性。   1. 建议：   加强对实验的讲解，让学生更加清楚地了解实验的目的和要求，以及实验中需要注意的问题。  提供更加详细的实验指导书和实验代码，让学生可以更加轻松地完成实验，并且可以更好地理解实验的内容和原理。  加强实验的交互性，让学生可以更加积极地参与实验，提高实验的效果和效率。  提供更加完善的实验环境和实验设备，让学生可以更加方便地进行实验，并且可以更好地掌握实验技能。 |
| **指导教师评语及成绩：** |
| **评语：**  **成绩： 指导教师签名：**  **批阅日期：** |

实验报告说明

**1．实验项目名称：**要用最简练的语言反映实验的内容。要求与实验指导书中相一致。

**2．实验类型：**一般需说明是验证型实验还是设计型实验，是创新型实验还是综合型实验。

**3．实验目的与要求**：目的要明确，要抓住重点，符合实验指导书中的要求。

**4．实验原理：**简要说明本实验项目所涉及的理论知识。

**5．实验环境**：实验用的软硬件环境（配置）。

**6．实验方案设计（思路、步骤和方法等）**：这是实验报告极其重要的内容。概括整个实验过程。

对于**操作型实验**，要写明依据何种原理、操作方法进行实验，要写明需要经过哪几个步骤来实现其操作。

对于**设计型和综合型实验**，在上述内容基础上还应该画出流程图、设计思路和设计方法，再配以相应的文字说明。

对于**创新型实验**，还应注明其创新点、特色。

**7．实验过程（实验中涉及的记录、数据、分析）：**写明具体上述实验方案的具体实施，包括实验过程中的记录、数据和相应的分析。

**8．结论（结果）：**即根据实验过程中所见到的现象和测得的数据，做出结论。

**9．小结：**对本次实验的心得体会、思考和建议。

**10．指导教师评语及成绩：**指导教师依据学生的实际报告内容，用简练语言给出本次实验报告的评价和价值。

**注意：**

* 实验报告将记入实验成绩；
* 每次实验开始时，交上一次的实验报告，否则将扣除此次实验成绩。