**《操作系统原理》实验报告二**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 | 王粟鹏 | 学号 | U201817044 | 专业班级 | 软工1803 | 时间 | 2020.4.1 |

1. **实验目的**
2. 理解操作系统线程的概念和应用编程过程。
3. 理解线程的同步概念和编程。
4. **实验内容**

[1,2中任意1题和3,4,5或3,4,6 共计4道题] 在此我选择 1,3,4,5

1） 在 Ubuntu 或 Fedora 环境使用 fork 函数创建一对父子进程，分别输出各自的进程号和提示信息串。

2） 在Ubuntu或Fedora环境下使用pthread\_create函数创建2个线程A和B。线程A在屏幕上使用while循环顺序递增的输出1-1000的自然数；线程B在屏幕上用while循环顺序递减的输出1000-1之间的自然数。为避免输出太快，每隔0.5秒输出一个数。

3） 在 windows 环境下，利用高级语言编程环境（限定为 VS环境或 VC 环境）调用 CreateThread 函数实现2）的功能

4） 在Windows环境下，利用高级语言编程环境（限定为VS环境或VC环境）调用CreateThread函数和相关同步函数，模拟实现“生产者-消费者”问题。

5）在Windows环境下，利用高级语言编程环境（限定为VS环境或VC环境）调用CreateTHread函数实现“并发地画圆和画方”。圆的中心、半径、颜色，正方形的中心，边长，颜色等参数自己确定，合适就行。圆和正方形的边界上建议取720个点。为直观展展示绘制的过程，每个点绘制后睡眠0.2~0.5秒。

6） 在Windows环境下，利用高级编程环境（限定为VS环境或VC环境）调用CreateThread函数实现“文件拷贝小工具”。功能如下：1、具有一个编辑框，让用户任意指定源目录或文件，2、具有一个编辑框，让用户指定目的目录或文件；3、具有“开始拷贝”按钮；4、具有“停止拷贝”按钮；5、具有显示拷贝进度的lebal，当为目录拷贝时以文件数来统计进度，当为文件拷贝时以字节数来统计进度。

**三、实验过程**

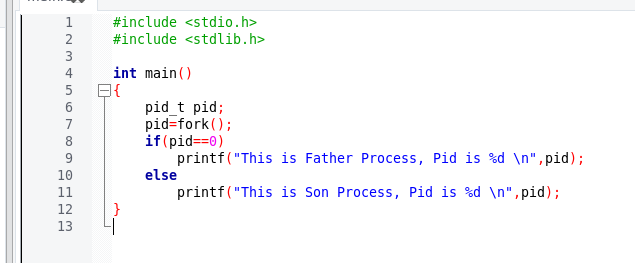
**（一）实验步骤**

1. **Ubuntu下的fork()函数。**

编程环境：Ubuntu系统下，使用CodeBlocks编程。

开发过程：首先编写实现该功能的代码，build之后再执行该程序，观察输出的结果是否正确。

核心代码如下：



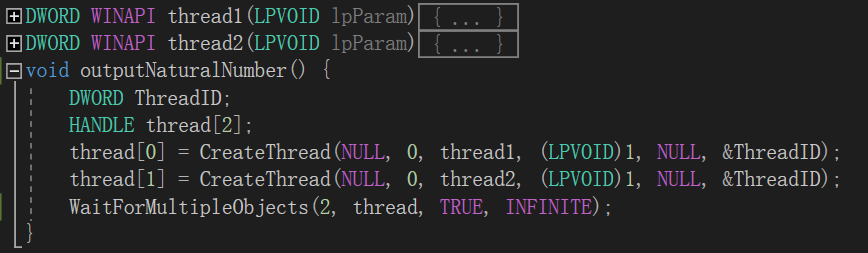
先创建一个进程号类型的对象pid，然后使用fork函数创建一个子进程。接下来根据pid进程号来判断是父进程还是子进程，然后根据不同的判断，输出进程号以及不同的提示信息。

1. **Windows双线程输出自然数。**

编程环境：Windows下使用VS编程环境

开发过程：首先编写两个不同线程的函数，然后在main函数中创建这两个线程函数的线程，执行线程，然后等待线程结束，观察输出的过程以及结果。

核心代码如下：



其中的thread1和thread2分别为两个的线程函数，lpParam为进程传递的参数。函数体内分别实现了正序输出自然数和逆序输出自然数，每输出一个数字便使用Sleep()函数睡眠0.5秒。在outputNaturalNumber函数当中，创建了一个进程数组thread，然后为thread1和thread2分别创建了一个线程。最后使用WaitForMultipleObjects函数等待两个线程的结束。

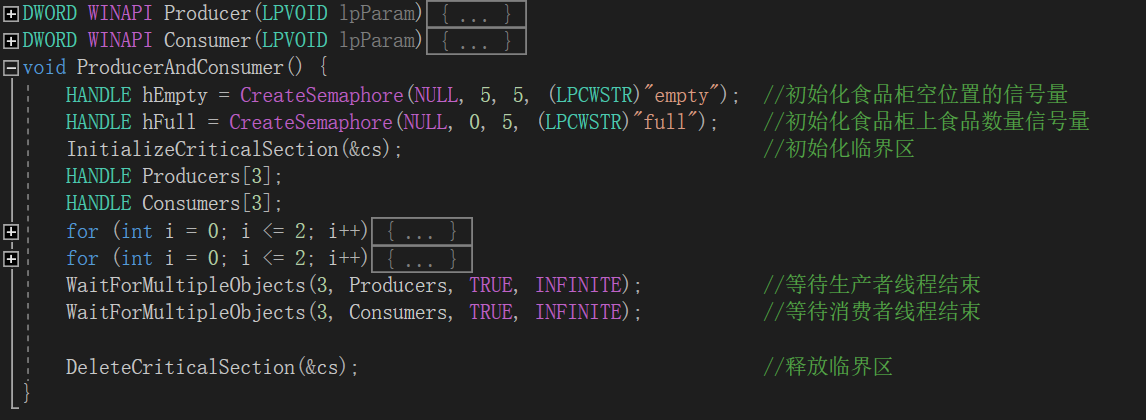
在main函数当中调用outputNaturalNumber函数，即可看到输出的过程以及结果。

1. **“生产者-消费者”。**

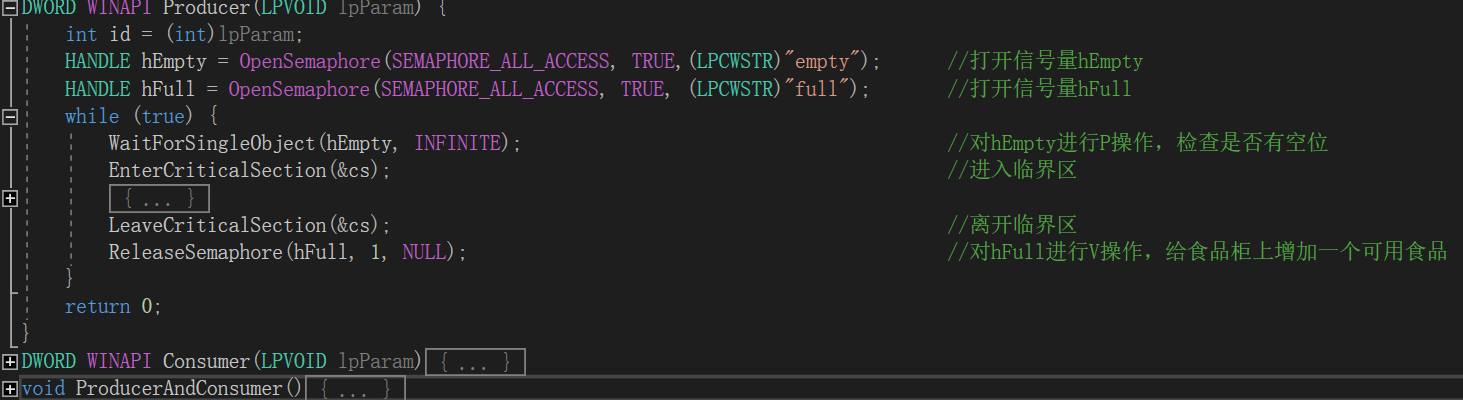
编程环境：Windows中的VS编程环境。

开发过程：食品柜上最大数量为5首先分析出该程序需要两个信号量，一个指示商品柜上的空位，还有一个指示商品柜上的商品数。除此之外，还要有一个临界区。一共需要两个线程函数，一个用来代表生产者，另一个用来代表消费者。首先声明全局变量临界区cs，在主函数中创建信号量、初始化临界区，然后编写“生产者”和“消费者”的线程函数，最后在主函数当中对“生产者”和“消费者”分别创建线程函数，随后可观察两者的运行情况。

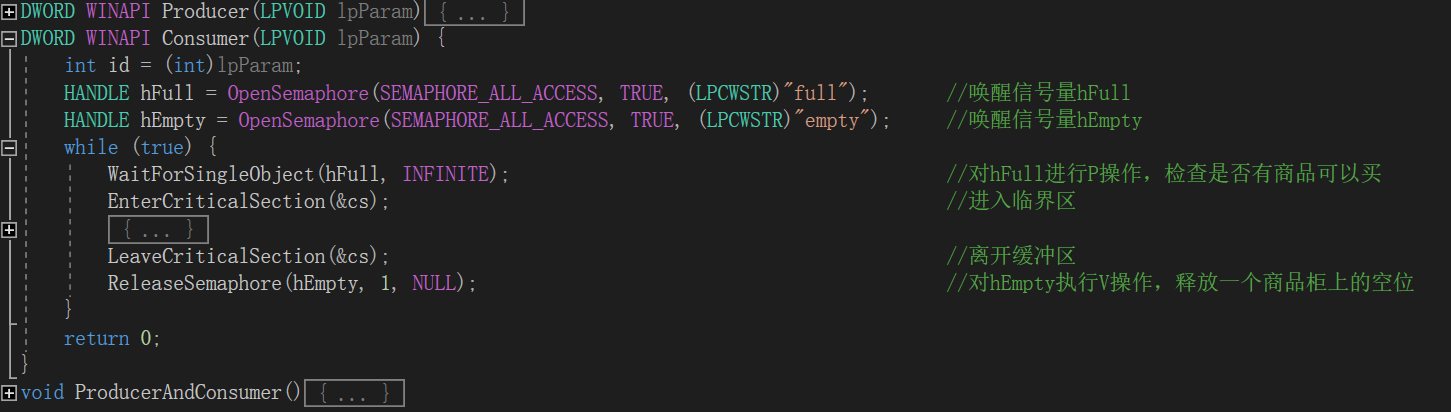
核心代码如下：



该部分为“生产者-消费者”的执行体。首先创建商品柜空位和商品柜商品位数量的信号量，然后初始化临界区。随后使用CreateThread创建生产者和消费者的线程函数，在创建线程函数的时候，给线程函数传递lpParam参数，用来表示不同的生产者和消费者。创建完线程之后使用WaitForMultipleObjects函数等待所有的线程结束，然后释放临界区。



该函数为“生产者”函数。首先获得当前生产者的id，然后将信号量empty和full打开。在while循环体内，首先对hEmpty执行P操作，检查是否还有空位置，如果有空位置则会进行执行下一步操作，进入临界区之后，输出相应的提示信息，然后退出临界区，最后对hFull执行V操作，增加一个可用商品。



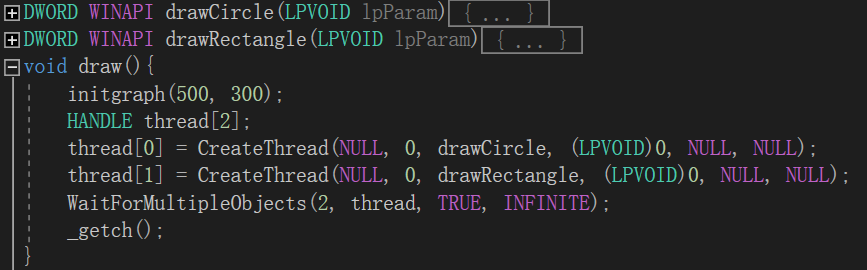
该部分为消费者函数。首先获得当前消费者的id，然后将信号量empty和full打开。在while循环体内，首先对hFull执行P操作，检查是否有商品可以购买，如果有商品则会进行执行下一步操作，进入临界区之后，输出相应的提示信息，然后退出临界区，最后对hEmpty执行V操作，增加一个商品空位。

在main函数当中调用ProducerAndConsumer函数，即可实现“生产者-消费者”的功能，同时输出每一步的操作结果。

1. **同时画圆和正方形**

编程环境：Windows下的VS编程环境。除此之外，使用到了EasyX。函数库为<graphics.h> 执行过程：首先编写两个线程函数，一个用来画圆，另一个用来画方。然后在主函数中创建一个画布，然后分别调用CreateThread()为两个函数分别创建线程，使用\_getch()函数停留在绘图界面。观察绘图过程以及结果。

核心代码如下：



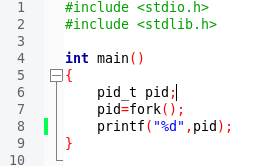
draw()函数为绘制的主体函数。首先调用initgraph()函数创建画布。然后创建画圆和画方的线程，使用WaitForMultipleObjects()函数等待两个线程结束。

drawCircle和drawRectangle函数分别为画圆和画方的函数，分别是将一个圆和正方形等分为720个点，然后每绘制一个点调用Sleep()函数睡眠1秒钟。

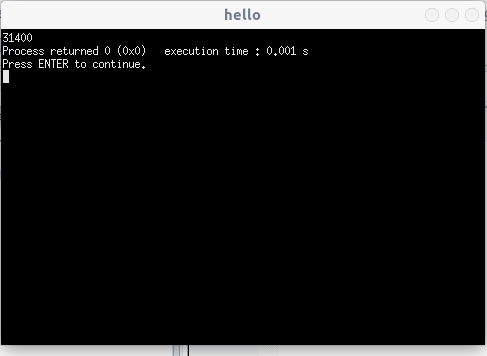
在main()函数中调用draw()函数，观察圆形和方形的绘制过程。

1. **解决错误和优化**
2. 错误类型：操作过程错误。

错误原因：没有对父子进程判断就输出进程号。

****

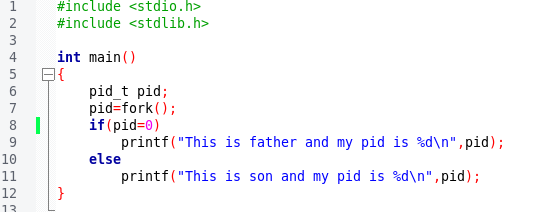
错误现象：只是输出一串数字，没有区分父子进程。



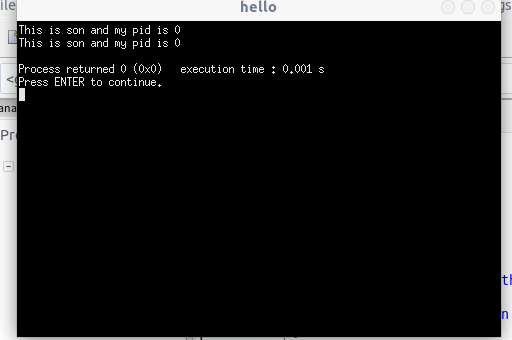
解决办法：在代码中添加父子进程的判断步骤，通过对pid的判断，执行不同的输出句柄，以此来分别父子进程。

1. 错误类型：特殊的语法错误

错误原因：在判断进程号pid是否等于0时语法错误，使用了错误语法pid=0



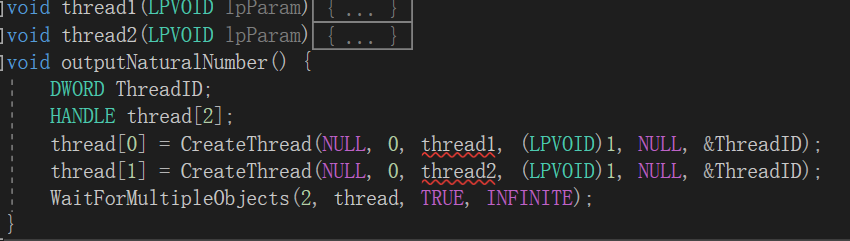
错误现象：输出的语句都相同，而且不会输出父进程的提示信息。



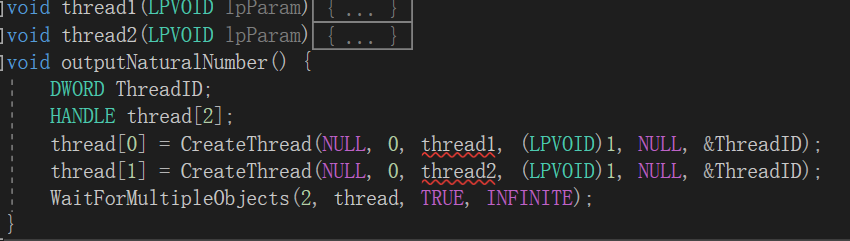
解决方法：把”pid=0”改为”pid==0”.

1. 错误类型：编译错误

错误原因：线程函数的返回值出错，返回值应该为一个函数的地址，我却返回void。



错误现象：线程函数无法正常调用，无法创建进程。会报错，参数无法转换。



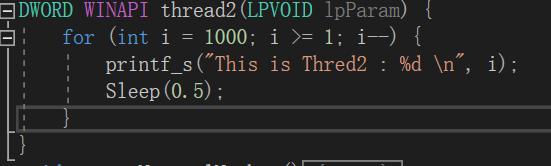
5UEDY71NJ~_P]02E})KPHAV

解决办法：将两个线程函数的返回值改为DWORD WINAPI。

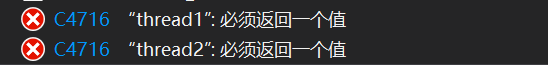
M~IGOXFY5EWNL(4[2WLW$5C

1. 错误类型：运行错误

错误原因：在进行双线程输出自然数的实验时，编写的线程函数缺少返回值。



错误现象：线程函数无法正常运行，编译器报错。

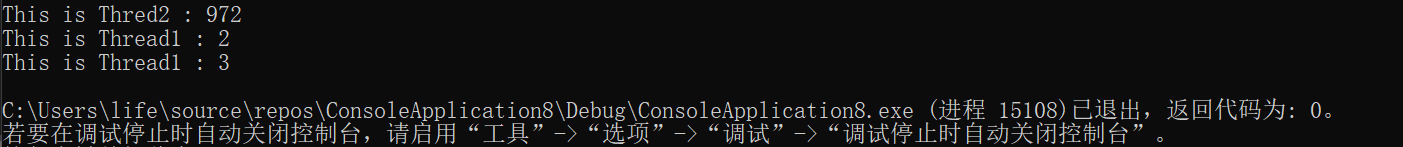


解决办法：给线程函数增加返回值，return 0;

1. 错误类型：操作过程错误

错误原因：在双线程输出自然数的实验中，没有使用WaitForMultipleObjects()等待线程的结束.

错误现象：自然数没有输出完，程序就已经结束了，无法获得完整的输出。

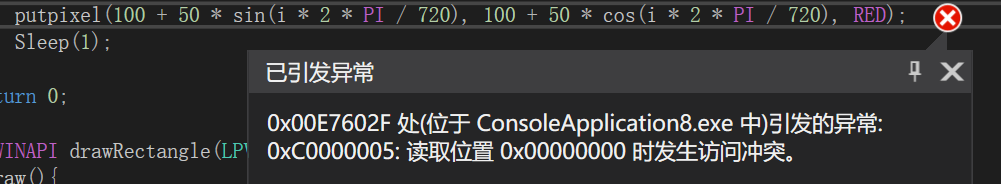


解决办法：使用WaitForMultipleObjects函数等待进程的结束。

1. 错误类型：运行错误

错误原因：在使用绘图函数之前，没有创建画布。

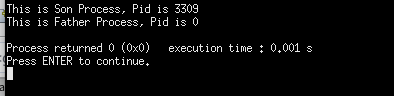
错误现象：使用绘图函数时，位置访问错误，无法画图。



解决办法：先在draw()函数当中调用initgraph()函数创建画布。

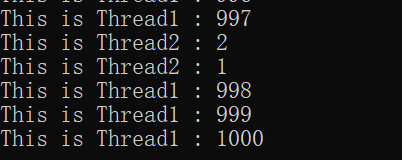
**四、实验结果**

1. **Ubuntu下的fork()函数**

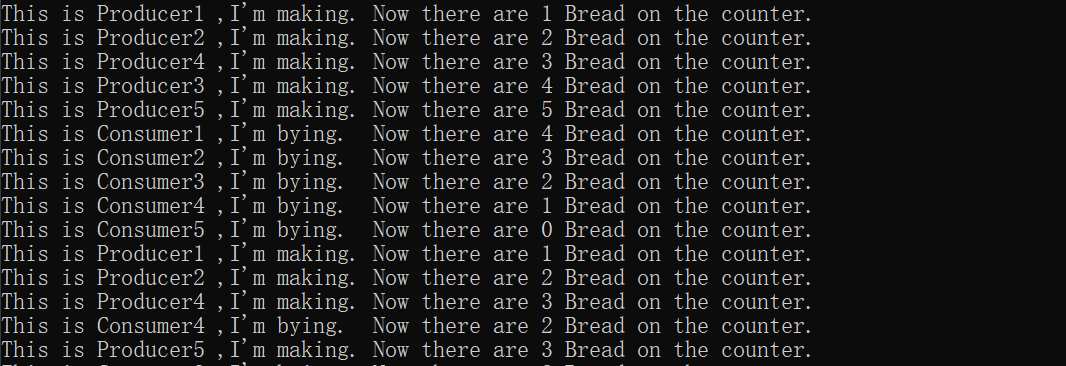
****

输出结果正确，能够根据pid成功的判断父子进程。Pid为0时，输出的为父进程提示信息；当pid>0时，输出为子进程提示信息。

1. **Windows双线程输出自然数**

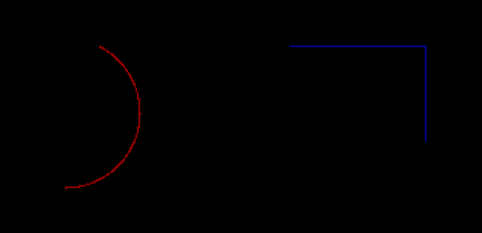
****根据输出结果观察到，线程1和线程2交替进行，最终均各自完成了相应的输出功能。

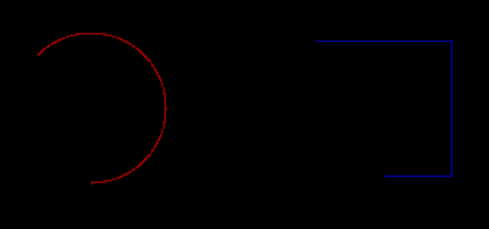
1. **“生产者-消费者”**

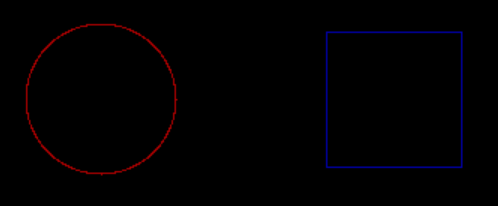
****

根据输出信息观察到，不同的消费者和生产者是交替进行，交替顺序没有规律。当食品数为0时，没有出现消费者继续买食品的情况；当食品柜数量为5时，即食品柜已满时，也不会出现生产者继续生产食品的现象。这说明两个信号量工作正常。我们观察到食品的数量是在自然数的基础上连续变化的，没有出现相邻数字不变化，或者变化数大于1的情况，因此临界区也发挥了作用。

1. **同时画圆和正方形**

****

****

****

通过观察可以发现，画圆和画方确实是同时进行的。

1. **体会**

理解了Linux系统下进程的创建、Windows系统下创建新的线程以及Windows系统下如何实现线程的同步和互斥。将关于信号量、锁机制以及线程的概念应用到了实践当中，对“如何使用信号量以及锁机制解决实际问题”进行了实践，在此过程中学会了如何编写一个线程函数以及如何创建一个新的线程。

在编程经验方面。在这次实验当中，我开始学着分析库函数的参数是什么含义、如何使用这些参数以及如何通过这些参数传递信息，有时候一些类型转换还是必要的。

在教训方面。一些比较基本的语法还是要注意，比如：pid==0和pid=0 。在今后的编程中更应该注意这些方面。