**《操作系统原理》实验报告三**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 | 王粟鹏 | 学号 | U201817044 | 专业班级 | 软工1803 | 时间 | 2020.4.17 |

1. **实验目的**
2. 理解线程/进程的通信机制和编程；
3. 理解线程/进程的死锁概念和如何解决死锁。
4. **实验内容**

（2）（考虑信号通信机制）在Ubuntu或Fedora环境创建父子2个进程A，B。进程A不断获取用户从键盘输入的字符串或整数，通过信号机制传给进程B。如果输入的是字符串，进程B将其打印出来；如果输入的是整数，进程B将其累加起来，并输出该数和累加和。当累加和大于100时，结束子进程，子进程输出“My Work done!”后结束，然后父进程也结束。

（4）在Windows环境使用创建一对父子进程，使用管道(pipe)的方式实现进程间的通信。父进程提供数据（1-100，递增），子进程读出来并显示。

（5）在Windows环境下，利用高级语言编程环境（限定为VS环境或VC环境或QT）调用CreateThread函数哲学家就餐问题的演示。要求：（1）提供死锁的解法和非死锁的解法；（2）有图形界面直观显示哲学家取筷子，吃饭，放筷子，思考等状态。（3）为增强结果的随机性，各个状态之间的维持时间采用随机时间，例如100ms-500ms之间。

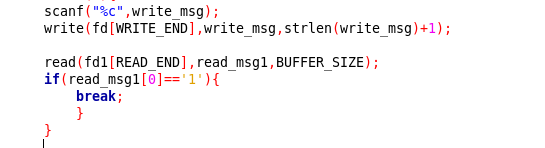
1. **实验过程**
   * 1. **实验步骤**
2. **Ubuntu下的父子进程通信**

环境说明：Linux-5.5.7

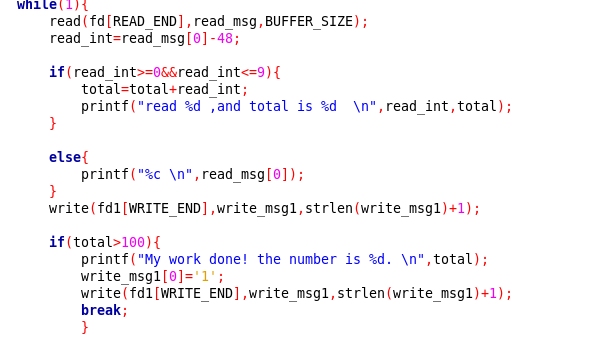
开发过程：

选择使用父子进程之间的管道通信传递信息。首先要在main函数当中声明相关的缓冲区以及文件描述符fd[2]和fd1[2]，其中fd控制父进程写子进程读取，fd1控制子进程写父进程读取。然后创建两个管道，实现双向通信。使用fork()函数创建子进程，用pid记录进程号。

如果pid>0，则证明为父进程，通过循环不断获取用户输入，然后通过管道写入缓冲区，同时读取另一个管道中的信息，检查子进程是否结束。



如果pid==0，则证明为子进程，不断循环从管道中读取数据，并对数据进行操作。如果数字之和大于100，则传递给父进程信息，然后依次结束进程。



1. **Windows下的匿名管道通信**

环境说明：Windows下的VS环境。

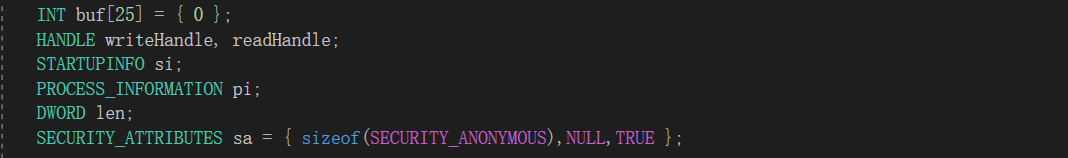
开发过程：

子进程：先编写子进程，完成子进程所需要实现的功能。

父进程：首先在父进程中定义缓冲区、以及管道和子进程所需要的参数，然后建立管道，创建子进程，在父进程当中向管道中写数据，由子进程读出数据

核心代码：

父进程



该部分主要定义了缓冲区，以及创建管道和创建子进程所需要的参数。

]N$BBXFXV4@~S%U55T4I6WF

创建管道，读写句柄分别为 readHandle 和 writeHandle

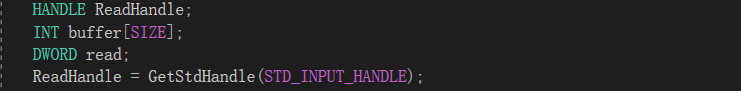
4WX8ATSJFKK%MF9$8PEWNGI

使用CreateProcess创建子进程，第五个参数设置为TRUE，允许子进程访问父进程创建的管道，如果成功创建了子进程，则输出”Make Process sucessfully”。

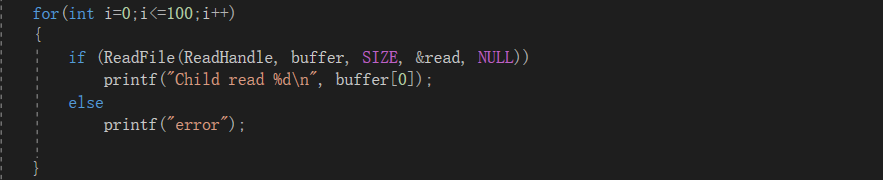


使用循环向缓冲区中写入数据，然后使用WriteFile写入到管道当中。等循环结束之后，使用WaitForSingleObject()函数等待子进程的结束。

子进程



定义和配置子进程读取管道内容所需要的参数



使用循环，每次循环都读取管道中的内容，然后输出。

1. **哲学家就餐问题**

环境说明：Windows下的VS环境。除此之外，使用到了EasyX。函数库为<graphics.h>。

实验过程：

会发生死锁的情况

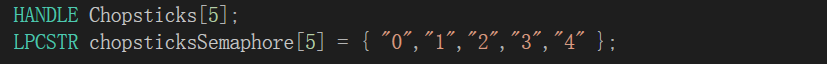
哲学家线程函数的编写。根据传入的参数确定哲学家的编号，然后函数遵从一下步骤：首先访问左手边筷子的信号量、访问右手边筷子的信号量、吃饭、释放左手边筷子的信号量、释放右手边筷子的信号量、思考。每个步骤都随机Sleep时间，使用圆代表哲学家，使用不同的颜色代表哲学家所处于的不同的状态。

主线程函数。创建五个信号量，存储在一个数组当中，便于根据编号访问信号量代表的筷子。然后创建五个哲学家进程。调用WaitForMultipleObjects()函数等待线程结束。

不会发生死锁的情况

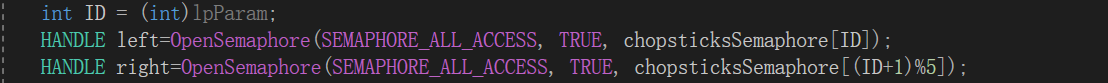
该部分和会发生死锁的代码差别不大。只是在哲学家函数的编写当中，增加了一个步骤检测左边的左边的筷子信号量，然后接着释放。因此，当前四个哲学家都拿到自己左边的筷子时，第五个哲学家检测到左边的左边的筷子被取走之后就不会取自己左边的筷子，这样就给第一个哲学家创造了取自己右边筷子的机会，实现了资源的合理分配。

核心代码：

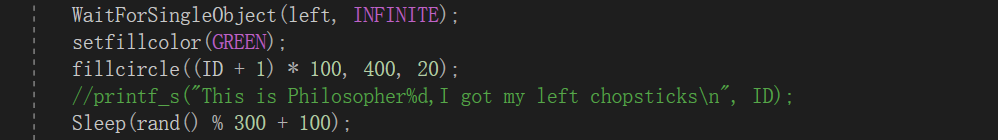


全局变量，第一个数组存储象征筷子的信号量。第二个数组用来辅助查找信号量。

哲学家线程函数



哲学家函数当中的准备参数，ID表示哲学家的编号。left表示左手边的筷子,right表示右手边的筷子。



访问左手边筷子的信号量，然后进行图形界面的相关变化。右手边的处理类似。

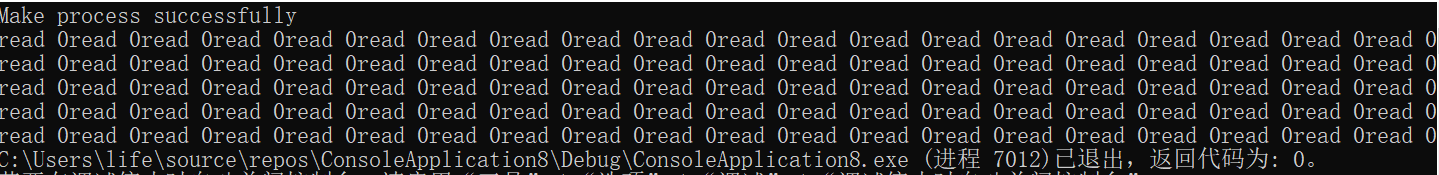
* + 1. **解决错误和优化**

1. 错误类型：运行错误

错误原因：在Windows管道通信当中，子进程并没有编写管道读取代码，而是使用scanf()函数获取管道中的内容。



错误现象:可以成功创建进程，但是无法正确输出结果。



解决办法：重写子进程函数，使用ReadFile从管道中获取数据。

1. 错误类型：运行错误

错误原因：通过管道传输的是char类型，在输出的时候没有转换为int类型

的值，而是输出char类型。

[S4UAHG``ZXX0YGZIR}Z~KX

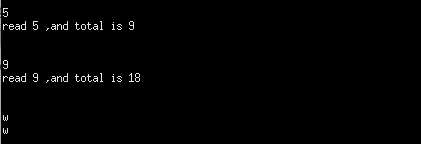
错误现象：无法输出正确结果，会输出字符。



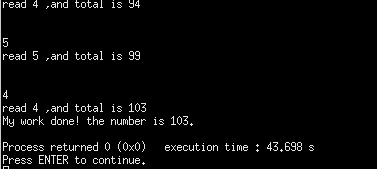


解决办法：将输出语句中的%c 改为 %d。

1. **实验结果**
2. **Ubuntu下的父子进程**

****

能够正确读出数字和字符，并可以正确进行不同操作。



当输入数字之和大于100时，可以正确输出提示信息，然后子进程父进程一次结束。

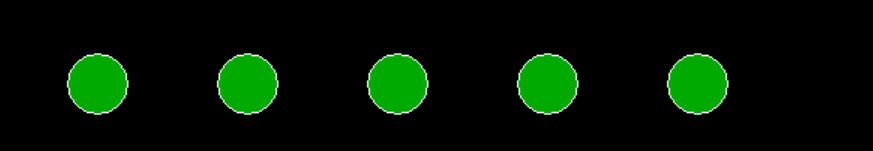
1. **Windows下的匿名管道通信**



子进程可以正确的输出从管道中读取的信息。

1. **哲学家就餐问题。**

会发生死锁的情况：所有的哲学家都卡绿色的状态，说明所有的哲学家都拿到了自己左手边的筷子，但都无法进行下一步操作。



不会发生死锁的情况：每个哲学家线程函数都可以正确的运行，不会发生阻塞的情况。





1. **体会**

在这次上机过程中，感觉收获很大。上一次上机实验虽然有管道通信的选项，但我并没有做，所以在这次实验当中补了回来。不得不承认刚开始写管道通信的时候费了很大的力气。当时是在网络上找的博客，一边学习一边写。但博客上写的比较乱，所以自己走了很多弯路。后来还是看书，在现代操作系统第九版中的实例很好的帮我认识了如何使用匿名管道进行通信。