



Operating System Principle, OS

《操作系统原理实验》

Linux+ Windows综合版本

华中科技大学

2022年03月-2022年05月

实验二：第4章进程管理，第5章死锁

● 一、实验目的

- (1) 理解进程/线程的概念和应用编程过程；
- (2) 理解进程/线程的同步机制和应用编程；

● 二、实验内容

- 1) 在Linux下创建一对父子进程。
- 2) 在Linux下创建2个线程A和B，循环输出数据或字符串。
- 3) 在Windows下创建线程A和B，循环输出数据或字符串。
- 4) 在Linux下创建一对父子进程，实验wait同步函数。
- 5) 在Windows下利用线程实现并发画圆/画方。
- 6) 在Windows或Linux下利用线程实现“生产者-消费者”同步控制
- 7) 在Linux下利用信号机制实现进程通信。
- 8) 在Windows或Linux下模拟哲学家就餐，提供死锁和非死锁解法。

● 三、实验要求

- 2,6,8必做,其余任选2。课前提前预做，老师机房检查和答疑。

实验二：第4章进程管理，第5章死锁

● 四、实验指南

■ 1) 在Linux下创建一对父子进程。

- ◆提示1：分别输出各自的进程号，父进程号和特别的提示字符串信息
- ◆提示2：让父进程提前结束或后结束，观察子进程的父进程ID。
- ◆提示3：使用PS命令查看进程列表信息，核对进程号，父进程号

■ 2) 在Linux下创建2个线程A和B，循环输出数据或字符串。

- ◆提示1：使用pthread线程库
- ◆提示2：线程A递增输出1-1000；线程B递减输出1000-1。为避免输出太快，每隔0.2秒（可自行调节）输出一个数。
- ◆提示3：输出数据时，同时输出A或B以标示是哪个线程输出的，并注意格式化输出信息。例如：

A:1000

A:0999

B:0001

A:0998

B:0002

.....

实验二：第4章进程管理，第5章死锁

● 四、实验指南

■ 3) 在Windows下创建线程A和B，循环输出数据或字符串。

- ◆提示1：使用CreateThread创建线程
- ◆提示2：线程A递增输出1-1000；线程B递减输出1000-1。为避免输出太快，每隔0.2秒输出一个数。
- ◆提示3：输出数据时，同时输出A或B标示是哪个线程输出的，并注意格式化输出信息。例如：

A:1000

A:0999

B:0001

A:0998

B:0002

.....

■ 4) 在Linux下创建父子进程，实验wait同步函数，理解父子进程同步

- ◆提示1：子进程休眠5秒，父进程不休眠。子进程用exit返回参数。
- ◆提示2：父进程调用wait等待子进程先结束，并分析子进程返回参数。
- ◆提示3：父进程输出子进程的返回信息。

实验二：第4章进程管理，第5章死锁

● 四、实验指南

■ 5) 在Windows下，利用线程实现并发画圆画方。

- ◆提示1：圆心，半径，颜色，正方形中心，边长，颜色自己确定。
- ◆提示2：圆和正方形边界建议都取720个点。为直观展示绘制过程，每个点绘制后睡眠0.2秒~0.5秒。
- ◆提示3：建议使用VS和MFC或QT对话框类型程序来绘制窗口和图形。



实验二：第4章进程管理，第5章死锁

● 四、实验指南

■ 6) 在Windows或Linux下利用线程实现“生产者-消费者”同步控制

- ◆提示1：使用数组（10个元素）代替缓冲区。2个输入线程产生产品（随机数）存到数组中；3个输出线程从数组中取数输出。
- ◆提示2： Windows使用[临界区对象](#)和[信号量对象](#)，主要函数
EnterCriticalSection | LeaveCriticalSection | WaitForSingleObject |
ReleaseSemaphore
- ◆提示3： Linux使用[互斥锁对象](#)和[轻量级信号量对象](#)，主要函数：
sem_wait(), sem_post(), pthread_mutex_lock(),
pthread_mutex_unlock()
- ◆提示4： 生产者1的数据：1000-1999 (每个数据随机间隔100ms-1s)，生产者2的数据：2000-2999 (每个数据随机间隔100ms-1s)
- ◆提示5： 消费者每休眠100ms-1s的随机时间消费一个数据。
- ◆提示6： 屏幕打印（或日志文件记录）每个数据的生产和消费记录。

实验二：第4章进程管理，第5章死锁

● 四、实验指南

■ 7) 在Linux下利用信号机制实现进程通信

- ◆提示1：父进程创建子进程，并让子进程进入死循环。
- ◆提示2：子进程每隔2秒输出“I am Child Process, alive !\n”
- ◆提示3：父进程询问用户“To terminate Child Process. Yes or No? \n”
要求用户从键盘回答Y或N.若用户回答N，延迟2秒后再提问。
- ◆提示4：若用户回答Y，向子进程发送用户信号，让子进程结束。
- ◆提示5：子进程结束之前打印字符串：“Bye,Wolrd !\n”
- ◆提示6：函数：kill(), signal(), 利用用户信号，编写信号处理函数

实验二：第4章进程管理，第5章死锁

● 四、实验指南

■ 8) 在Windows或Linux下模拟哲学家就餐，提供死锁和非死锁解法。

- ◆提示1：同时提供可能会带来死锁的解法和不可能死锁的解法。
- ◆提示2：可能会带来死锁的解法参见课件。Windows尝试使用临界区对象（EnterCriticalSection, LeaveCriticalSection）；Linux尝试使用互斥锁(pthread_mutex_lock, pthread_mutex_unlock)
- ◆提示3：完全不可能产生死锁的解法，例如：尝试拿取两只筷子，两只都能拿则拿，否则都不拿。Windows 尝试使用 WaitForMultipleObjects, WaitForSingleObject和互斥量对象 ReleaseMutex等相关函数) Linux尝试使用互斥锁pthread_mutex_lock, pthread_mutex_trylock等函数。
- ◆提示4：[可选]图形界面显示哲学家取筷，吃饭，放筷，思考等状态。
- ◆提示5：为增强随机性，各状态间维持100ms-500ms内的随机时长。