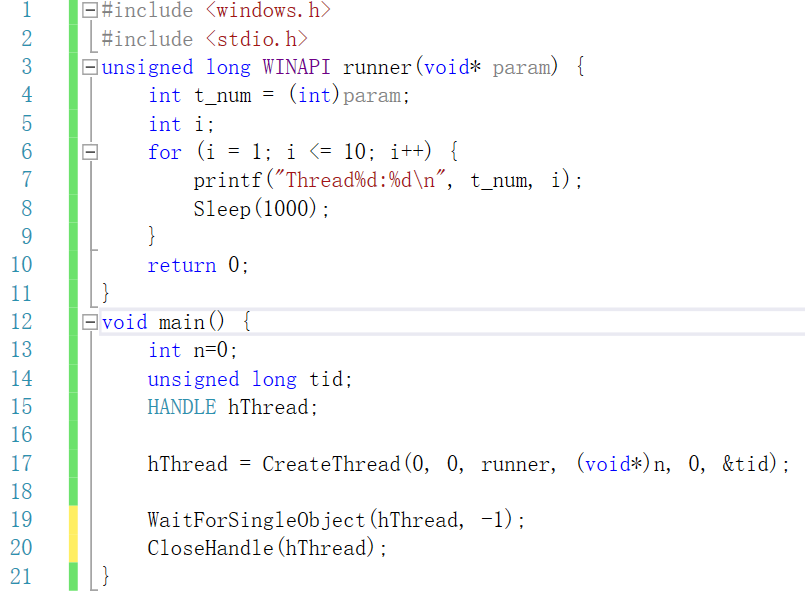
## 实验3：互斥

**完成下列操作，并将操作截图置于WORD文件中，文件命名：实验3-学号-姓名.docx”中，以及源代码文件压缩后一并上传“校网络教学平台”作业下的对应实验报告下。**

1. **Windows创建线程Win32API函数CreateThread**
2. 创建1个线程，每隔1秒输出1到10的数字。编辑、编译并执行下列代码（参考《操作系统精要(原书第2版)》P120，图4-11）,程序命名为winthread1.c。

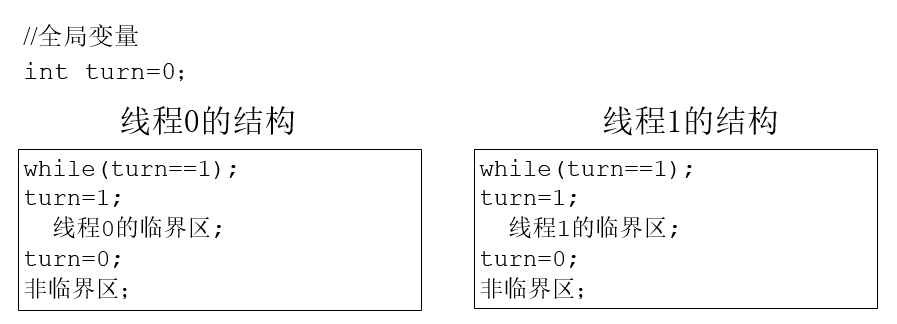


**要求与提示：**

1. 在Visual Studio下编辑、编译与运行
2. 查看CreateThread联网帮助：光标停于函数处🡪F1
3. 通过设置断点进行调试，查看进程号、线程号和线程数
   * 进程：（菜单）调试—>窗口—>进程
   * 线程：（菜单）调试—>窗口—>线程
4. 创建2个线程。改造上面的程序winthread1.c，新命名为**winthread2.c**。创建2个线程，每隔1秒输出1-10数字，2个线程并发执行。

**要求与提示：**

1. 通过设置断点进行调试，查看进程号、线程号和线程数
   * 执行到（但未执行）语句17、19、20时至少各有几个线程。
2. **使用“锁变量”实现互斥（有竞争条件）**



1. 在Windows下对上面2个线程的程序winthread2.c，以“锁变量”方式，实现2个线程互斥输出1-10数字，即当一个线程正在输出时，另一个线程阻塞等待。程序命名为winlockvar.c。

**要求与提示：**

1. 观察是否实现互斥输出到屏幕。并分析什么情况下出现竞争条件。
2. 在Linux下创建2个线程，各自每隔1秒输出1-10数字，二个线程并发执行。程序命名为**thread2.c**（参考“实验2：创建线程”）

**要求与提示：**

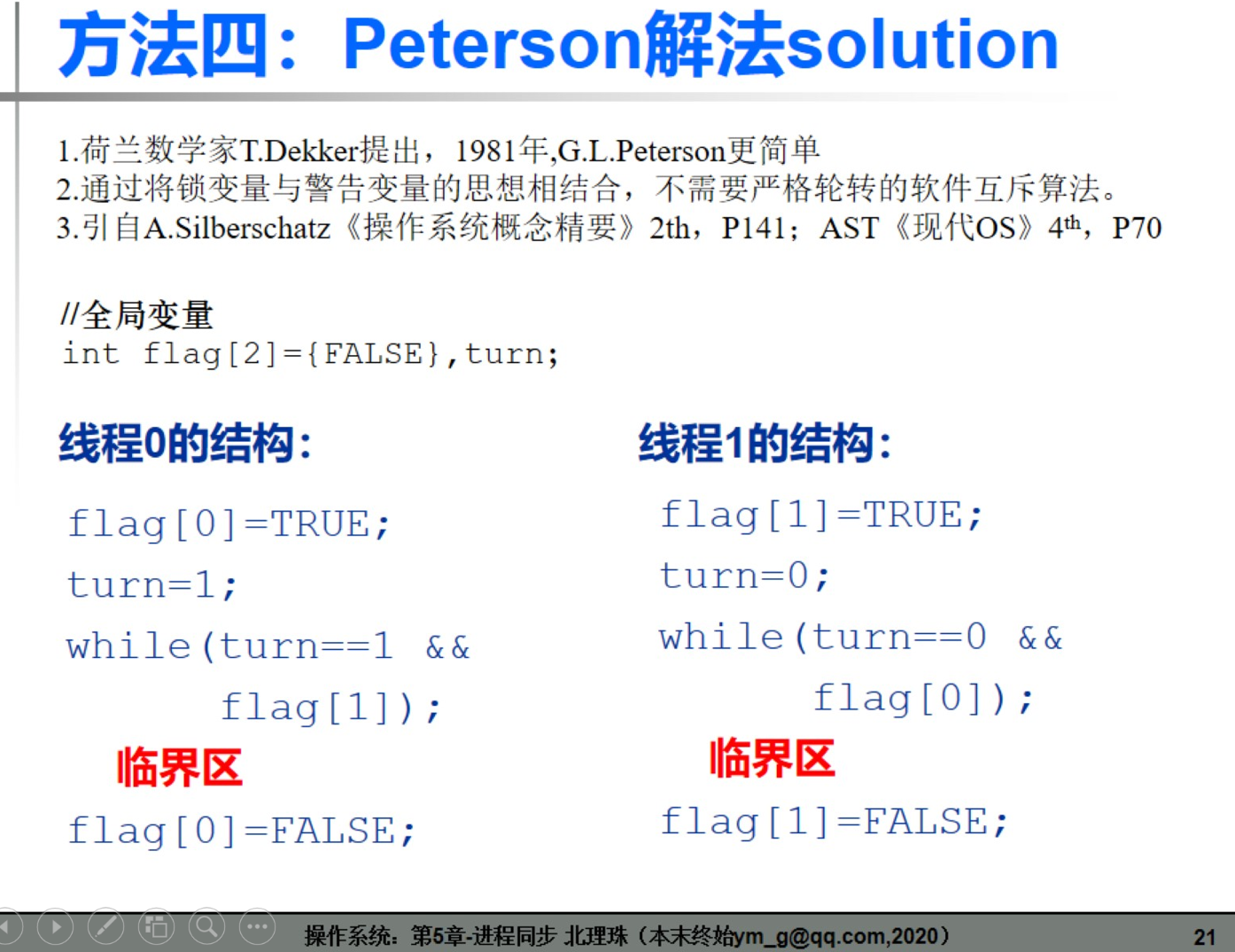
1. 观察执行的结果。
2. 修改**thread2.c**程序，以“锁变量”方式，实现2个线程互斥输出1-10数字，即当一个线程正在输出时，另一个线程阻塞等待。程序命名为lockvar.c。
   * 1. 观察执行的结果。
3. **使用“自旋锁”（严格轮转）实现互斥**



* 1. 在Linux下修改**thread2.c**，以“自旋锁”方式，实现2个线程互斥输出1-10数字程序命名为spinlock.c。

**要求与提示：**

1. 如果想要线程1先输出，如何修改程序
2. 如何实现三个线程的自旋锁，按照第3、2、1号线程的顺序输出
3. **使用Peterson解法实现互斥**



（参考《操作系统精要(原书第2版)》P141，图5-2）

1. 在Linux下修改**thread2.c**，以“Peterson”方式，实现2个线程互斥输出1-10数字程序命名为peterson.c。

**要求与提示：**

1. 如果线程0的turn=1改为turn=0，其它地方能否相应修改以实现互斥？
2. flag与turn的赋值语句互换，是否可行？
3. 如果线程0的while(turn==1 && flag[1])改为while(turn==0 && flag[1])，线程1的while(turn==0 && flag[0])改为while(turn==1 && flag[0])是否可行？
4. **使用互斥量实现互斥**

（参考《操作系统精要(原书第2版)》P159-160）

* **头文件：**

pthread.h

* **互斥量(或互斥锁)的数据类型：**

pthread\_mutex\_t

* **四个函数：**

pthread\_mutex\_init(互斥量,NULL)； //创建互斥量

pthread\_mutex\_destroy(互斥量); //消毁互斥量

pthread\_mutex\_lock(互斥量); //P操作，申请互斥量

pthread\_mutex\_unlock(互斥量); //V操作，释放互斥量

1. 在Linux下修改**thread2.c**，以互斥量方式，实现2个线程互斥输出1-10数字程序命名为mutex.c。

**要求与提示：**

1) 观察执行的结果。

1. **（附加）使用信号量实现互斥**

（参考《操作系统精要(原书第2版)》P160，无名信号量）

* **头文件：**

semaphore.h

* **信号量的数据类型：**

sem\_t

* **四个函数：**

sem\_init(信号量,0,信号量的初值)； //创建信号量

sem\_destroy(信号量); //消毁信号量

sem\_wait(信号量); //P操作

sem\_post(信号量); //V操作

1. 在Linux下修改**thread2.c**，以信号量方式，实现2个线程互斥输出1-10数字程序命名为semaphore.c。

**要求与提示：**

1) 观察执行的结果。