在VC中可按照如下步骤结束正在运行的其他进程

1）取得进程的句柄（利用FindWindow函数得到）；

2）获取进程ID号（用GetWindowThreadProcessId函数获取）；

3）打开进程，获取处理该进程的句柄；

4）利用TerminateProcess函数结束进程，将该函数的第二个参数设为4.

int CStaticFunc::KillProcess(LPCSTR pszClassName, LPCSTR pszWindowTitle)

{

HANDLE hProcessHandle;

ULONG nProcessID;

HWND TheWindow;

TheWindow = ::FindWindow( NULL, pszWindowTitle );

::GetWindowThreadProcessId( TheWindow, &nProcessID );

hProcessHandle = ::OpenProcess( PROCESS\_TERMINATE,

​ FALSE, nProcessID);

return ::TerminateProcess( hProcessHandle, 4 );

}

如何创建一个线程

\#include<stdio.h>

\#include<string.h>

\#include<pthread.h>

\#include<stdlib.h>

\#include<unistd.h>

pthread\_t tid[2];

\*\*void\\* doSomeThing(void \\*arg)\*\*

{

unsigned long i = 0; pthread\_t id = pthread\_self();

if(pthread\_equal(id,tid[0])) { printf("first\n"); }

else { printf("Second\n");}

return NULL;

}

int main(void) {

int i = 0; int err;

while(i < 2) {

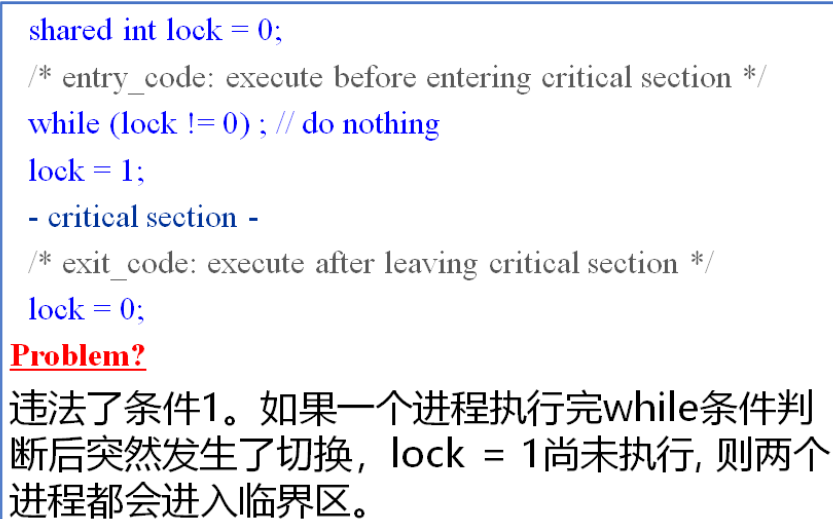
err = \*\*pthread\_create(&(tid[i]), NULL, &doSomeThing, NULL);\*\*

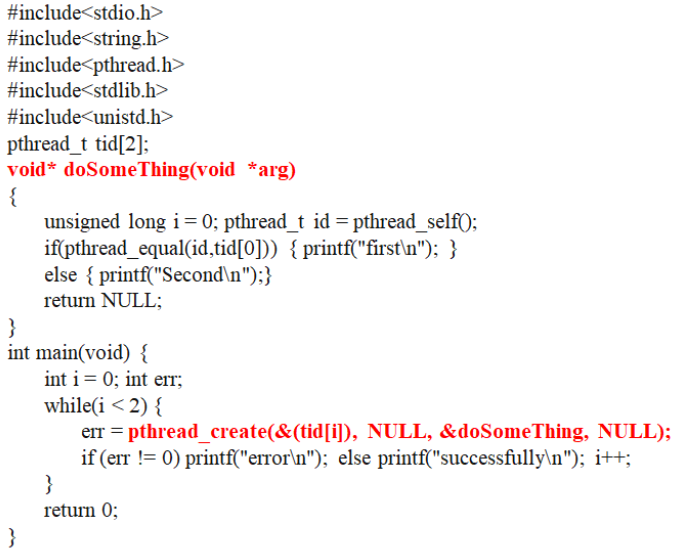
if (err != 0) printf("error\n"); else printf("successfully\n"); i++;

}

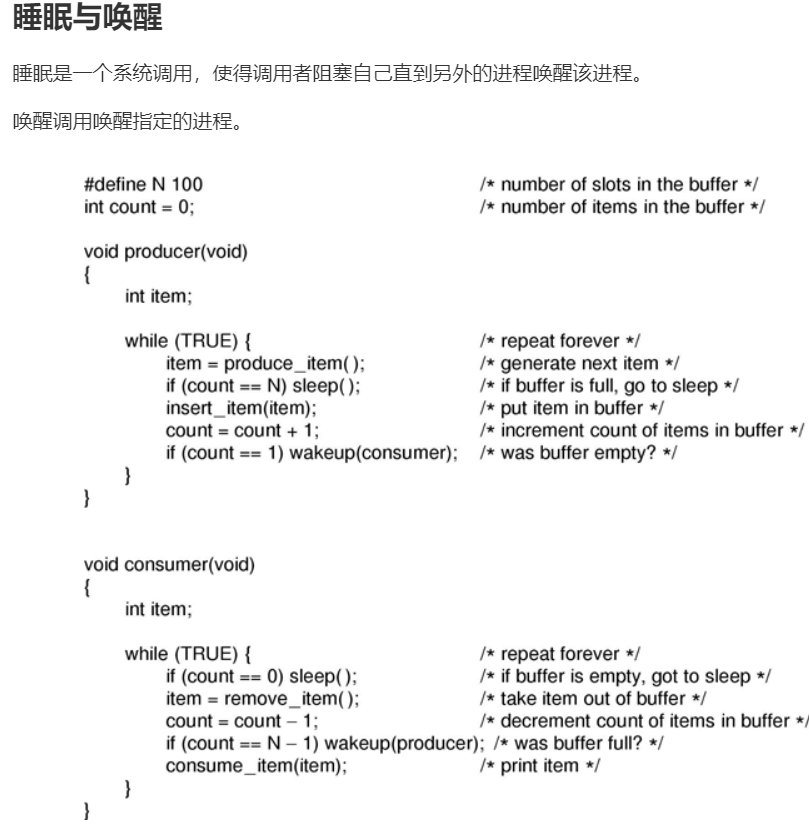
return 0;

}

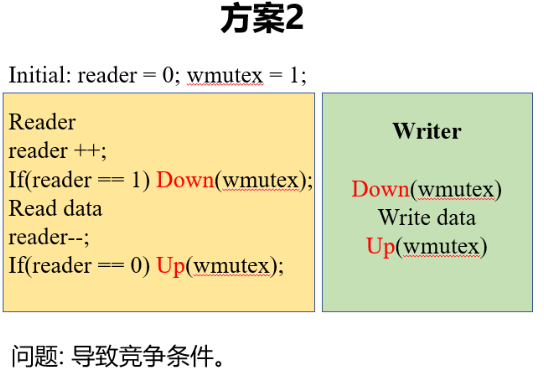
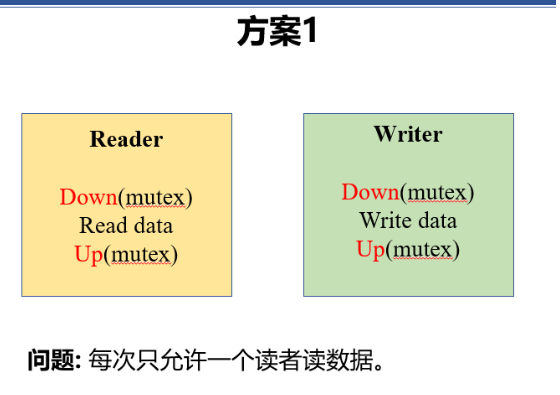


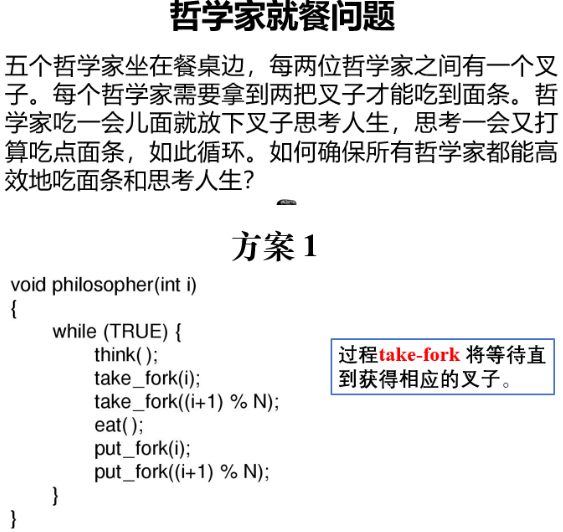


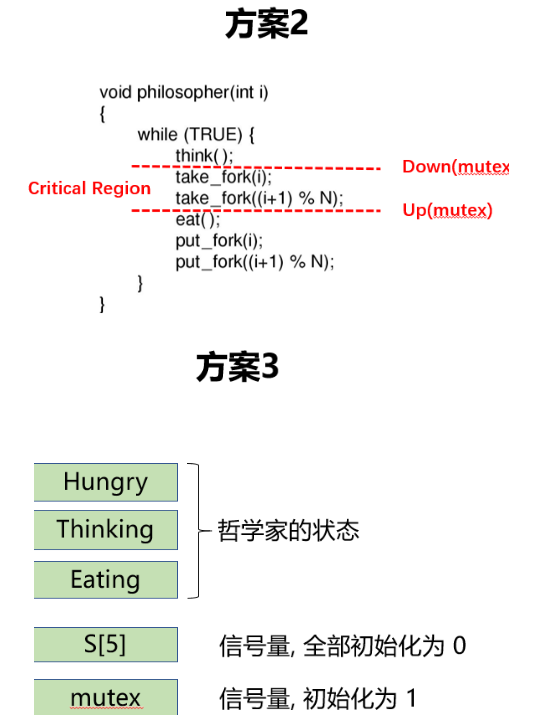
生产者消费者：



读者和写者



理发师

