简易任务管理器

姓名：高圣迪

0. Introduction

项目名称**简易任务栏管理器的实现。**在之前的基础上利用输入的指令来控制更新信息的线程的开关。我写的代码中，“b”表示打开线程，"e"表示关闭线程。

其他指令： show: 打印信息，quit: 退出程序, 其他任意指令: 打印“Wrong Command”并给出应该打什么指令。

程序开始运行时，更新信息的线程并没有启动，需要“b”来启动。

1. Functions

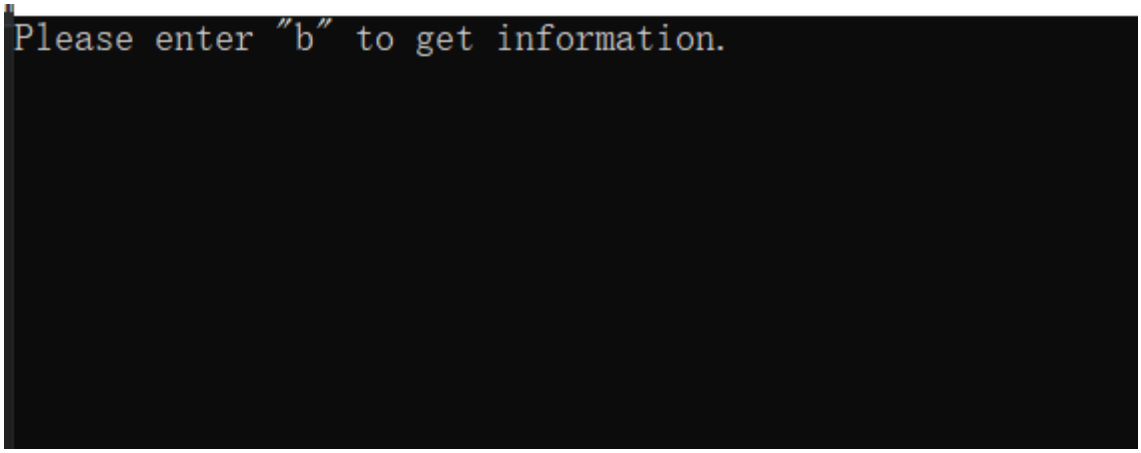
一开始我的想法是，既然有`handle = CreateThread()`,则在接到结束指令时直接调`CloseHandle()`API就行，结果发现没有用，这个函数的意思是我关闭了这个线程的句柄对象，表示我不再使用这个句柄了，不对此句柄对应线程做干预，并没有结束线程。

之后查一些资料，发现了`TerminateThread()`函数，这个函数确实可以把线程关闭，但我担心的点在于这个函数是强制关闭线程，可能会造成内存泄漏；或者我创的线程正在更新，存数据的list因写入被锁住，这时候线程被关了，锁却没有解锁。虽然自己实际跑了几次，没有出现问题，正常运行，但是还是不敢用这个函数。

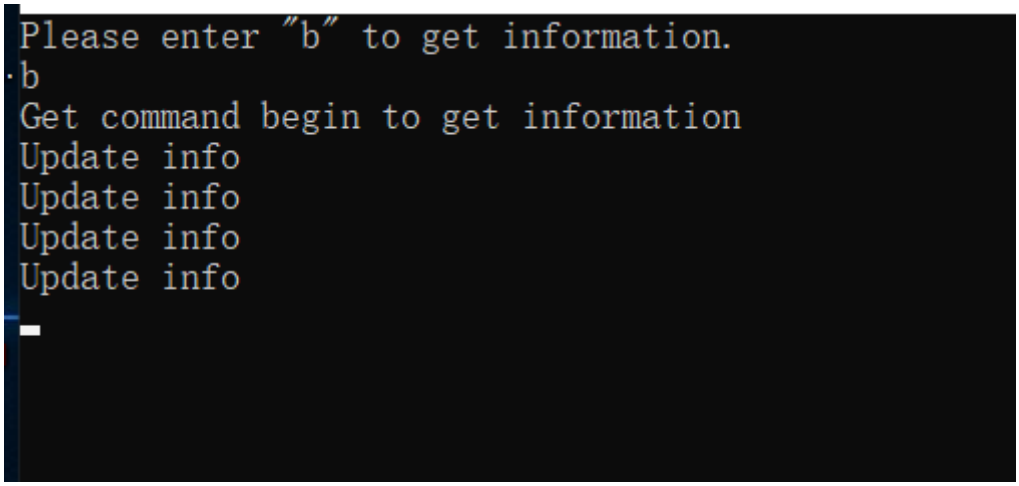
最后我想的是让线程自然退出。所以我用了一个全局变量，`signalThread`，当创建出线程时把它设为true，关闭线程时设为false(初始值也为false)。当次线程存在并判断到signalThread为false时，关闭定时器，退出线程，关闭句柄。如以下代码。

```
MSG msg;
while (GetMessage(&msg, NULL, 0, 0)){
    if (msg.message == WM_TIMER)
        DispatchMessage(&msg);
    if (signalThread == false) {
        KillTimer(hwnd, NULL);
        break;
    }
}
```

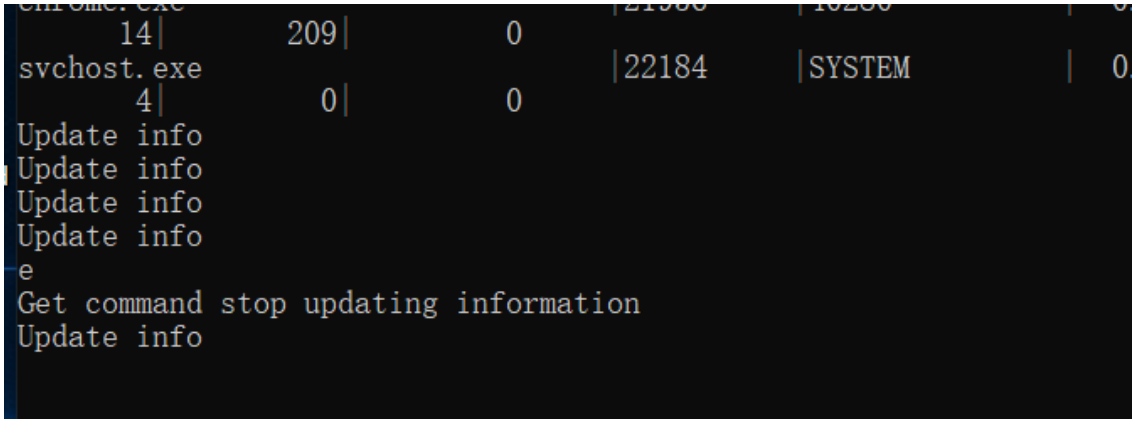
2. Result



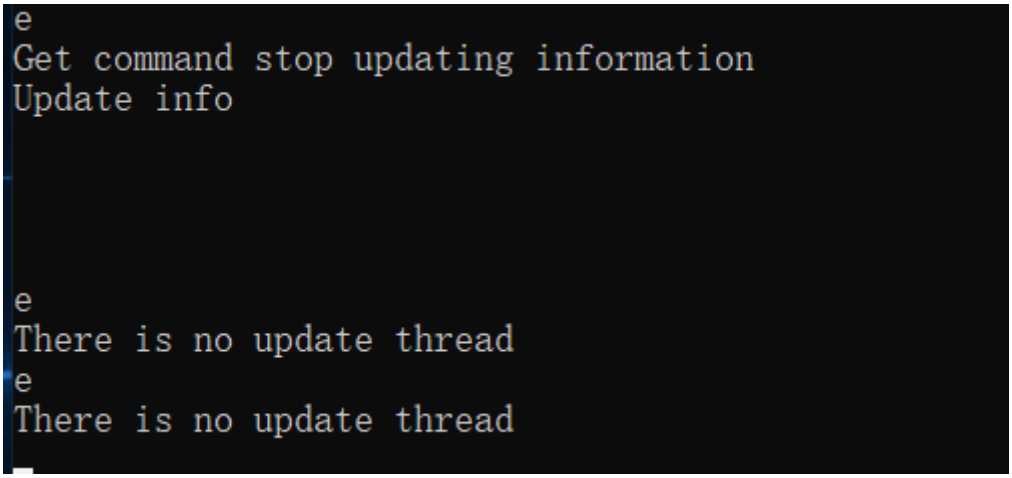
1. 一开始没有开启次线程就"show"的结果，提示用户输入“b”打开线程。



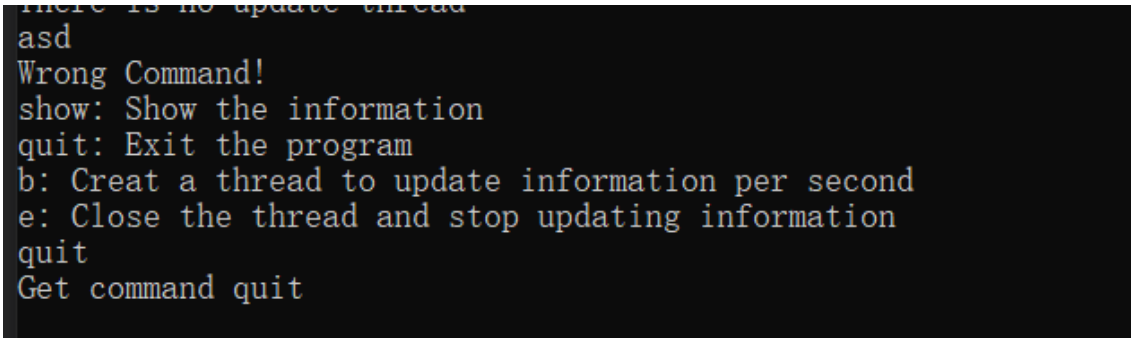
2. 输入“b”后线程开启，这张图里的“Update info"是为了更直观的看出线程确实打开了，我发给您的代码中这行print被注释掉了。



3. 这张图上半部分是show的结果，然后我输入“e"，线程确实退出了。



4. 当没有更新信息的线程，仍想关闭时，会有提示。打开线程同理，我没有写成可以开很多update info线程。（若想写的话，也是把判断删了即可）



5. 最后是随便输入的指令，有help提示出来。

3. Reference

CreateThread()之后又马上CloseHandle()的问题

https://blog.csdn.net/kofandlizi/article/details/6458011

两个终止线程函数：ExitThread 和 TerminateThread

https://blog.csdn.net/farrellcn/article/details/6423920