

线程池



日成蝶—Windows 高级编程

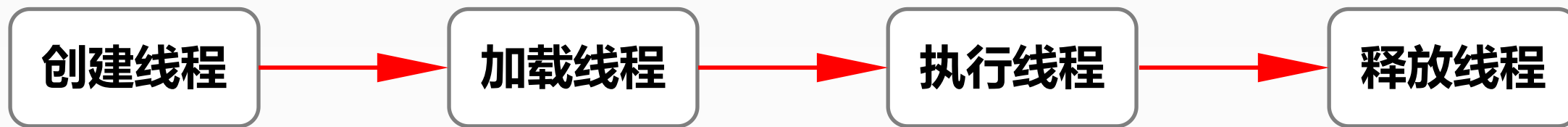
七日做茧，一朝成蝶！



主讲：袁春旭

线程使用流程

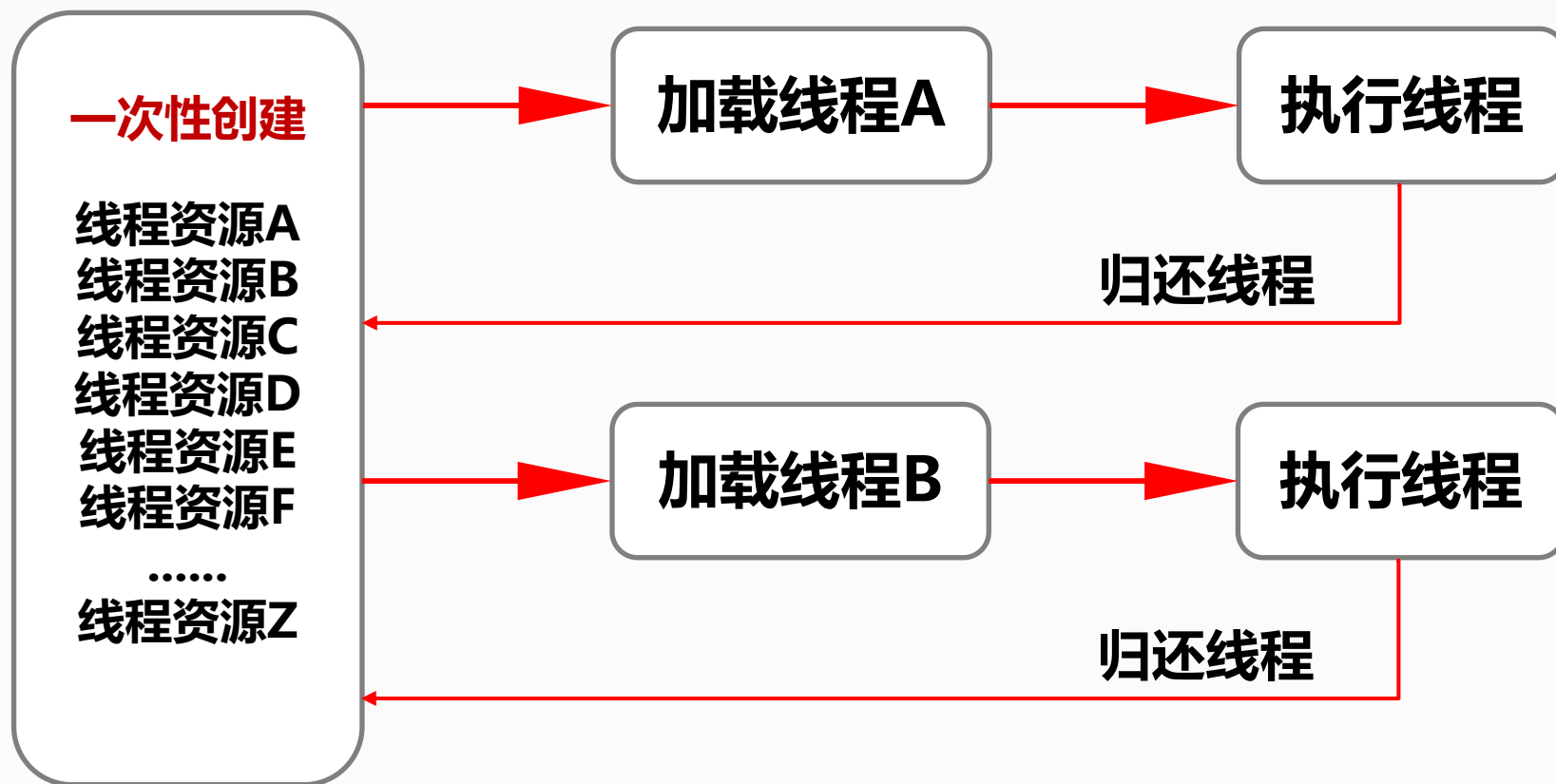
线程使用流程



思考：存在的问题，改进方法

线程池原理

线程池原理



API函数

API的设计对比

创建线程

加载线程

执行线程

CreateThread

TrySubmitThreadpoolCallback

首次调用API才
创建线程池，创
建后与进程共存
亡

释放线程

CloseHandle

优点：形式上省去创建和释放线程的调用，大规模多线程调用时相对效率更高。思考：何时不适用。
缺点：不断创建工作项，仍存在改进空间。

线程池异步函数调用

CreateThreadpoolWork

**创建线程池及工作项
线程池不会重复创建**

SubmitThreadpoolWork

执行线程

CloseThreadpoolWork

**仅销毁工作项
线程池与进程共存亡**

线程池周期性调用

CreateThreadpoolTimer

**创建线程池及计时器
线程池不会重复创建**

SetThreadpoolTimer

设定调用周期

CloseThreadpoolTimer

**仅销毁计时器
线程池与进程共存亡**

线程池内核对象触发调用

CreateThreadpoolWait

**创建线程池及等待对象
线程池不会重复创建**

SetThreadpoolWait

绑定内核对象

CloseThreadpoolWait

**仅销毁等待项
线程池与进程共存亡**

线程池IO完成调用

CreateThreadpoollo

**创建线程池及IO对象
线程池不会重复创建**

StartThreadpoollo

与IO完成端口关联

CloseThreadpoollo

**仅销毁项IO对象
线程池与进程共存亡**

编码实战