获得mutex对象所属权的线程才能进入临界区，未获得的线程在临界区外等待。

mutex和monitor相比，能够跨应用，能同步不同应用程序（同步两个exe文件）。

mutex是非静态类，使用时需要创建对象。

mutex分两种：局部互斥体和系统互斥体；局部互斥体只能在创建它的应用中使用，系统互斥体能在不同应用程序中使用；

局部互斥体使用方法：

mutex m = new mutex(false)

false表示创建该互斥体的线程不具有所有权，true则相反。

系统互斥体使用方法：

mutex m = new mutex(false, 系统名)；

不同应用程序根据系统名，辨别是不是同一个互斥体。

WaitOne（） RealeaseMutex（）

class Program {

static void Main(string[] args) {

Thread t1 = new Thread(delegate () {

Mutex fileMutex = new Mutex(false, "timeFileMutex");

string fileName = "C:\\Users\\soft\\Desktop\\time.txt";

for (int i = 1; i < 10; ++i) {

try {

fileMutex.WaitOne();

File.AppendAllText(fileName, "t1 " + DateTime.Now + "\r\n");

}

catch (ThreadInterruptedException) {

Console.WriteLine("线程t1被中断");

}

finally {

fileMutex.ReleaseMutex();

}

Thread.Sleep(1000);

}

});

t1.Start();

}

}

}