

基于 Windows 平台的实时测试系统软件的设计与实现

石雨荷¹, 张会生²

(1. 空军工程大学 电讯工程学院, 陕西 西安 710077; 2. 西北工业大学 电子工程系, 陕西 西安 710072)

摘要:在 Windows 平台下利用多线程技术对实时测试系统软件进行了设计和实现。通过分析多线程及其同步机制,结合实例给出实时测试系统软件的总体实现框图及思路;描述了基于事件同步对象的系统软件具体实现方法;给出线程在激光路面平整度测试仪系统软件实际测试中的工作流程。上述思路、方法与流程已被充分应用于对激光路面平整度测试仪系统软件的开发中,并取得了非常好的效果。

关键词:Windows 平台;实时测试系统软件;多线程;事件

中图分类号:TP274 **文献标识码:**A **文章编号:**1009-3516(2002)02-0077-04

在开发实时测试系统软件时,Windows 操作系统是一个良好的开发平台。其具有多窗口,多任务等诸多特性,使在此平台下开发出的实时测试系统软件不但具有友好的界面,而且还可实现在实时采集及处理的同时完成人机交互、图形数据显示、计算分析及打印等任务。同时,大量涌现的 Windows 环境下的软件开发工具,如 Visual C++、Borland C++ Builder 等,不但可为实时测试应用软件设计出漂亮的界面,而且可以充分利用 Windows 的多任务和多线程的能力设计并实现快速、稳定的实时测试系统软件。

1 多线程及其同步机制

所谓多线程,就是通过系统调度使几个具有不同功能的程序流(即线程)同时并行地运行^[1]。多线程是在程序内部实现多任务的能力,程序可由若干单独执行的线程构成,这些线程从用户角度看是同时运行的。利用多线程可实现任务内的各子任务并行运行,提高事务处理效率^[2]。特别是在实时测试软件开发方面,这种多线程能力,大大增加了基于 PC 测试的数据采集卡的性能,可防止基于 PC 的测试软件与插入式数据采集卡通信时产生的堵塞现象。只要将用户界面显示与数据采集分配在不同的线程上,就可降低数据采集与用户界面显示间的干扰,使每个线程能以最快速度独立地运行,充分提高系统的测试速度^[3]。但多线程也有不利的一面,线程越多,系统开销越大,故大多数情况下以 2 个或者 3 个线程为宜。

一个应用程序如果使用多线程,那么通常情况下它就必须以某种方式或机制来确保应用程序内的多个相关线程能够相互安全地、协调一致地工作,这种方式或机制就是多线程的同步机制。实现多线程的同步机制一般是通过同步对象来完成的,Windows 95 提供了许多同步对象,有临界区、互斥、信号量和事件等^[4]。其中事件对象是同步对象的最基本形式,它是用来发信号以表示某一操作已经完成。

线程有 3 个基本状态,分别是运行状态、准备运行状态及休眠状态。当一个线程处于休眠状态时,一般调用 Wait For Single Object 函数等待同步信号由无效变为有效,然后可以执行下面的代码,否则线程就处于等待的状态,操作系统也就不会分配机器时间片给该线程^[4]。

2 实时测试系统软件的总体设计与实现

系统软件设计的主要思路是把应用程序分为前台线程和后台线程两部分。

收稿日期:2001-05-18

作者简介:石雨荷(1972-),女,河南宝丰人,讲师,硕士,主要从事通信导航与测控系统技术研究。

前台线程为主控线程。其主要负责生成界面、设置在测试中所需要的各种参数、控制测试过程、在测试前设置 A/D 卡及初始化各传感器、生成后台数据处理和采集线程并对它们进行适时的控制、数据转换与管理以及打印等事后的各种相关处理工作。

后台线程为工作线程。它们分别是数据处理线程和数据采集线程,这两个线程由主控线程生成,其生存时间由主控线程控制,当主控线程发出结束测试的命令消息后,后台线程随即消亡。数据采集线程负责系统控制中心与外部采集设备之间的数据传输,A/D 卡在内/外触发的作用下,采集各传感器传送的模拟信号,将其转换为数字信号后存储在 A/D 卡中的 FIFO 存储器中。数据采集线程以一定的数据结构接收来自 A/D 卡的数据,发消息通知数据处理线程,使之开始处理数据。数据处理线程是本系统的主要程序实现部分,它负责实时计算分析、存储、显示等工作,程序的主要算法将在此实现。数据处理和数据采集两线程的相互配合与同步工作可用同步对象——事件(Event)来完成。

上述系统软件总体框图是在 Borland C++ Builder 软件开发环境中具体实现的。与 Visual C++ 等比较,使用 Borland C++ Builder 更简洁、方便,故它是 32 位 Windows 应用程序开发的最有力的工具之一。在具体的软件编程中,不但要创建和执行线程,而且要恰当地使用事件同步对象。

2.1 基于事件同步对象的实现

在 Borland C++ Builder 中,TThread 对象描述了线程的概念。但是,TThread 对象并不完整,使用线程对象必须在子类中重载 Execute 方法。创建时,从 TThread 生成一个新类,确保重载 Execute 方法^[5]。执行时,除非线程要完成有限的工作,否则只要将线程的执行代码放在 while 循环中,直到 Terminated 变成 true 为止。

//在主线中创建数据采集和数据处理线程,由“测试(开始)”菜单项执行

```
void __fastcall TMainForm::MenuBeginClick(TObject *Sender)
```

```
{
```

```
hReceive = CreateEvent(NULL, false, true, NULL); //创建采集事件对象句柄
```

```
hCount = CreateEvent(NULL, false, false, NULL); //创建处理事件对象句柄
```

```
ReceivedSamples = new TReceivedThread(true); //创建数据采集线程
```

```
Counters = new TCounterThread(false); //创建数据处理线程
```

```
}
```

用 Windows API 函数 CreateEvent 创建两个内核事件对象分别用于数据采集和数据处理线程。

//重载 Execute() 函数,执行数据采集线程

```
void __fastcall TReceivedThread::Execute(void)
```

```
{
```

```
while (! Terminated)
```

```
{
```

```
WaitForSingleObject(hReceive, INFINITE); //休眠状态中,等待被数据处理线程唤醒
```

//数据采集过程

```
SetEvent(hCount); //置数据处理事件对象为通知状态,唤醒数据处理线程
```

```
}
```

```
}
```

//重载 Execute() 函数,执行数据处理线程

```
void __fastcall TCounterThread::Execute(void)
```

```
{
```

```
while (! Terminated)
```

```
{
```

```
WaitForSingleObject(hCount, INFINITE); //休眠状态中,等待被数据采集线程唤醒
```

//数据处理过程

```
SetEvent(hReceive); //置数据采集事件对象为通知状态,唤醒数据采集线程
```

2.2 线程在实际测试中的实现

如前所述,使用线程并非越多越好,线程使用的代价是增大系统开销。因此,可以根据实际情况将数据采集线程和数据处理线程适当地合并为一个线程,采集在前,处理在后,可省去了依靠事件对象同步的过程。这样整个系统软件的实现实际上是由前台主线程和后台数据采集处理线程构成的。结合激光路面平整度测试仪实例的系统软件运行流程图(图 1 所示)加以说明。

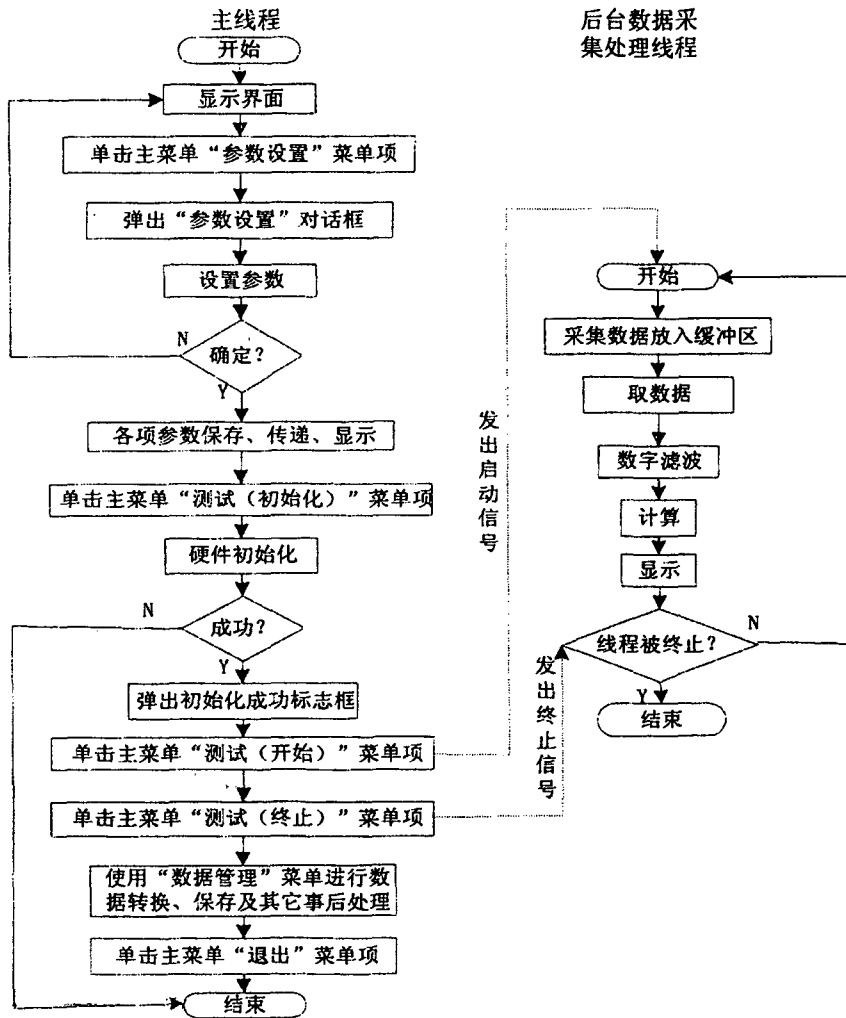


图 1 系统软件实现流程图

以上介绍的思路、方法与流程已成功应用于激光路面平整度测试系统的研制中。通过整个系统的硬件联调和多次实际上路进行路面测试,表明系统软件在 Windows 平台下工作正常,前台线程与后台线程运行协调且稳定,并与硬件系统所进行的工作配合良好,取得了非常好的效果。

参考文献:

- [1] Willims Stallings. Operating System Internals and Design Principles[M]. New Jersey:Prentice Hall Inc.,1998.
- [2] 李春华,徐明,周兴铭. 多线程的软件实现[J]. 计算机科学与工程,1999,21(4):17-21.
- [3] 金达,阙沛文. 当前计算机技术发展对基于 PC 机自动测试系统的影响[J]. 计算机自动测量与控制,1999,7(1):5-7.
- [4] Jeffrey Richter. Windows 高级编程指南[M]. 北京:清华大学出版社,1999.
- [5] John Miano, Tom Cabanski, Harold Howe. Borland C++ Builder 编程指南[M]. 北京:电子工业出版社,1998.

(编辑:门向生)

Design and Realization of Real - time Test System Software Based on Windows Platform

SHI Yu - he¹, ZHANG Hui - sheng²

(1. The Telecommunication Engineering Institute, Air Force Engineering University, Xi'an 710077, China; 2. Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: A real - time test system software under Windows platform is designed and realized by utilizing the technique of multithreading in this paper. First, combined with the examples, the general block diagram and thinking of real - time test system software are introduced after analyzing multithreading and its synchronization mechanism. Second, the concrete realization method of the system software based on synchronization event object is described. And last, the work flow of multithreading in the road surface roughness laser test system software is provided in detail. The above method, thinking and work flow have been fully applied to the development of the road surface roughness laser test system software, from which good results are achieved.

Keywords: windows platform; real - time test system software; multithreading; event

作者署名

署名是表示文责自负的承诺。所谓文责自负,即论文一经发表,署名者对作品负有责任,包括政治上、科学上和法律上的责任。如果文章中存在剽窃、抄袭的内容,或者有政治性、技术性错误,署名者即应负完全的责任。署名即表明作者愿意承担责任。

署名便于读者与作者联系。读者若需向作者询问、质疑或请教以求帮助,可以直接与作者联系。署名即表明作者有同读者联系的意愿。

署名原则:论文的署名者应具备下列条件:本人应是直接参加课题研究的全部或主要部分的工作,并做出主要贡献者;本人应为作品创作者,即论文撰写者;本人对作品具有答辩能力,并为作品的直接责任者。

有的人虽为课题组成员,参加了部分研究和实验工作,但由于其工作性质是辅助性的,不应列为作者;也有人研究工作确有贡献,并对成果有答辩能力,但未直接参加作品的创作工作,也不宜作为论文的作者。作者应是上述三原则的同时具备者。不够署名条件但对研究成果有所贡献者可以为“致谢”段中的感谢对象。

关于学位论文改写后在期刊上发表时的署名问题。由于第一,发表的是在答辩学位论文的基础上改写成文的;第二,学位论文及其报道的成果是在导师的指导下由学生完成的。所以,可以由学生和导师共同署名,一般是学生在前导师在后。

直接由个人创作的作品,由作者个人署名,个人作者为作品(论文)的著作权人。个人署名一般应使用真实姓名。

多位作者共同完成的作品联合署名时,署名顺序按对该文的贡献大小排列。第一作者为主要贡献者和直接创作者,同时又是作品的直接责任者,享有更多的权利,承担着更多的义务。除有特别声明外,第一作者就是第一权利、第一责任和第一义务者。

团体作者和执笔者。如果由一个组织机构或数人组成的团体对一篇作品(论文)承担责任,可以用该团体的名称来署名;或出于保密等其他的原因,也可以用虚拟的团体名称署名。由1个或几个人整理或改编的作品,可以署执笔者姓名。

(摘自陈浩元主编《科技书刊标准化18讲》)