Win32 下异步通信的一种实现方法

何王合 杨凡德 李伟华 (西安西北工业大学 710072)

摘要:本文探讨了如何利用 NetBios 实现在 32 位操作系统下的异步通信的原理和方法,给出两种实现途径:一是调用中断例程,二是使用异步事件句柄。我们在人机接口和嵌入程序中,利用该原理和方法成功地实现了异步通信。

关键词:线程 异步通信 句柄 事件

一、前言

网络通信的基础是网络通信协议,目前常用的几种通信协议不外乎 IPX/SPX、NETBIOS 和 TCP/IP 等。其中 NETBIOS 以其简单的接口,良好的兼容性得到广泛的应用。但如何利用 32 位操作系统的强大功能和 NetBios 实现异步通信,还是有待于探讨的技术。本文是从对 WIN32 中的 NetBios 从分析到具体的设计于实现,并给出了两种实现方法。作者在系统开发中,利用该方法和原理成功地在人机接口和嵌入程序中实现了异步通信。

二、WIN32 的多任务支持

WINDOWS95 是一个抢先式多任务的 32 位操作系统,可以让多个任务真正同时运行。因此在 WINDOWS95 这样的 32 位操作系统中,以多线程支持的多任务是实现异步通信的关键。

在 WIN32 中线程的创建是由 CreateThread 函数进行的,事件的创建事件是通过 CreateEvent 函数来完成。一旦创建了事件和线程后,在线程的运行模块中加入 WaitForSingleObject 函数,它允许无限期地等待一个对象由非信号状态转变为信号状态,或者指定一个超时间隔,超过这个时间间隔,等待则终止。WIN32 中的 NetBios 异步请求可以指定一个异步对象,当异步请求完成后,NetBios 将这个异步对象设置为信号状态。

三、WIN32 对 NETBIOS 的支持

WIN32 对通信的支持,一般有下列几种技术:利用 WNET 函数、NETBIOS、通信槽、管道、插座(SOCKET)等。本文仅讨论如何利用 NETBIOS 来实现异步通信,NetBios 是 IBM 公司为局域网通信开发的协议,支持多种通信协议,如 TCP/IP、MAP/TOP 和 IEEE 等,并且适合于 DOS、UNIX、OS/2 和 WINDOWS 等多种环境。NetBIos 位于 OSI 参考模型的会话层和表示层之间,把 NetBios 安排在较高的层次,有利于把用户程序和 NetBios 之间的通信同具体的网络实现方式隔离开来,因此,NetBios 应用程序有很好的兼容性和移植性。WIN32 的 API 中支持 NETBIOS 的仅有一个函数 NetBios,而且它仅有一个参数,即一个指向网络控制块(NCB)结构的一个指针。网络控制块存放所有的服务语意内容。NCB 包括一个命令,一个返回代码,网络环境的信息和指向用作消息或关于网络未来数据的缓冲区的指针。在 WIN32 的 NETBIOS 函数的实现中包括了一些扩展,并不是 IBM NETBIOS 3.0 说明书的一部分,与其也有一些不同的地方。这样使 POST 例程能在 C 中被调用,并允许使用事件对象来完成通知单。这些主要是由于网络控制块中增加了一个 ncb-event 字段。ncb-event 是用来说明一个 WIN32 事件句柄,当接受到一个异步 NetBios 命令时,ncb-event 说明的事件由系统设置为非接收信号状态;当命令结束时,NETBIOS 将其设置为收到信号状态。当 NETBIOS 函数返回一非零值,就向该事件发信号。如果 ncb-event 没有设置为 ASYNCH 值或 ncb-post 为非零,则 ncb-event 必须为零,否则就会返回 NRC—ILLCMD。换句话说,不能要求多于一个的关于 NCB 需求,要完成的通知单或是同步的,或者向一事件发信号,或者调用一个完成程序。

利用 ncb-event 来提出异步请求比用 ncb-post 节省资源,而且 ncb-event 是非零的,如果在异步请求处理以前线索就已终止,则这个异步请求将会被撤消。而用 ncb-post 发送的异步请求就不能这样。特别要注意的一点是仅有手工设置的事件才可被 NETBIOS 使用,并且一个给定的事件不应与多个活动的异步 NETBIOS 命令相关

© 中国科学院软件研究所 http://www.c-s-a.org.cn

联。

四、异步通信的实现

在 WIN32 中进行异步通信的方法一般有两种:第一种是调用中断例程;第二种是使用异步事件句柄。

1. 调用中断处理例程

我们假设申请用户名为 USER,则:

```
USER.ncb-command = NETBIOS-ADDNAME|ASYNCH;
USER.ncb-post = (void(CALLBACK *)(NCB *))newPost;
NetBios(&USER);
对应的中断处理函数代码:
NCB—POST newPost(NCB * Ncbptr)
{//在此设置 MsgFlag 为真
//在此插入响应事件代码
//也可用 SengMessage 或 PostMessage 等方式向消息队列中插入对应的消息
}
```

2. 使用异步事件句柄

在 Win32s 系统中进行异步请求时,一般是通过改变一个事件的状态(由非信号状态到信号状态)来通知异步通信的完成,同时由一个跟踪此事件状态改变的线程来通知主线程(往 WINDOWS 的消息循环队列中加入对应的消息)或直接插入事件响应代码。如果有多于一个的异步通信请求要求同时执行时,可以把它们分配给几个不同的线程;当然也可以由一个线程同时轮流跟踪多个事件,来完成不同的响应。在 ncb-event 字段中赋以事件的句柄前应先建立一个置为人工复位且初始化为非信号状态的事件,然后用一线程跟踪它。

具体作法如下:

step 1. 创建一个 NetEvent 的事件:

```
hnewNetEvent = CreateEvent (NULL, TRUE, FALSE, "newNetEvent);//创建该事件;
//返回的句柄 hnewNetEvent 就是用来执行 NetBios 异步命令
//时 ncb)event 所要的值。
```

step 2. 创建一跟踪线程:

```
DWORD ThreadnewID; hThreadnew = CreateThread (NULL, 0, (LPTHREAD-START-ROUTINE)newNet-Routine,
//说明线程的起始地址
NULL, CREATE—SUSPENDED, //说明线程的挂起
&ThreadnewID ); ResumeThread ( hThread-new);//运行
```

step 3. 执行异步 NetBios 命令:

```
myNcb.ncb-event = hnewNetEvent;
//传送事件的句柄
myNcb.ncb-command = NETBIOS-ADDNAME | ASYNCH; //指定命令以异步方式执行
NetBios(&myNcb);
```

step 4. 新建线程的运行模块

```
void newNetRoutine(void *) {WaitForSingleObject (hnewNetEvent, -1);
//以无超时等待 hnewNetEvent 指定的事件由非信号状态转变为信号态,并设置消息标志为真。
//在此可以插入响应代码。
//在此也可向消息队列中发一个消息。以便通知主线程进行必要的处理。
ExitThread(0);
}
```

从上面可以看出,使用异步事件句柄进行通信的关键是对异步命令完成消息的捕捉以及将此消息通知给主线程。

五、结束语

两种实现方法各有优点。从程序结构上考虑,用第二种方法更为合理,用户可以随时控制通信的进行。如果通信是用菜单命令来实现的,那么用第一种更好。在程序中,随程序中的某一事件的产生而需进行通信的情况,则用第二种更合适。

参考文献

[1] 《C Program's Guide to Serial Commununications》 JoeCampbell Sams Pubishing

[2] 《Windows NT 高级编程技术》[美] Jeffrey Richter 清华大学出版社

[3] 《NetBios C 程序员指南》[美] W. David Schwaderer 著

[4] 《Windows 高级编程》李力等 海洋出版社

(来稿时间:1997 年 6 月)

© 中国科学院软件研究所 http://www.c-s-a.org.cn