

Windows CE 自定义 Shell

覃 朗 雷跃明 (重庆大学 软件学院 重庆 400044)

摘 要: Shell 是用户访问操作系统的接口。Shell 开发在 Windows CE 操作系统开发中占据一定的重要性。对 Windows CE Shell 进行了概述,并通过建立 Shell 模型,讲述如何自定义 Windows CE Shell 和定制 Shell 必须注意的问题,对 Windows CE Shell 的定制进行深入研究。

关键词: Windows CE Shell; 自定义 Shell; Shell 模型; 模块; 消息管理

Customizing Windows CE Shell

QIN Lang, LEI Yue-Ming

(Department of Software Engineering, Chongqing University, Chongqing 400044, China)

Abstract: Shell is the interface for users to access to operating system. Shell development is of importance among Windows CE operating system development. This paper presents an overview of Windows CE Shell by establishing Shell model. It discusses how to customize Windows CE Shell and lists problems to be considered during customizing Shell. The establishment of Windows CE Shell has been investigated in depth.

Keywords: Windows CE shell; customizing shell; shell model; modular; communication management

1 引言

Windows CE Shell 为用户运行设备商的应用程序和管理 Windows CE 操作系统对象提供了一个接口,这些对象既可以是位于目标设备上的实际对象,如文件和文件夹,也可以是虚拟对象,如回收站。Windows CE Shell 是 Windows CE 操作系统主要的图形用户接口,是 Windows CE 启动后的第一个图形界面窗口。标准的 Windows CE Shell 包括众所周知的,如任务栏和开始菜单的 Windows 组件。由于不同嵌入式设备对系统和应用的要求不同,因此嵌入式设备需要定制系统。Windows CE 允许开发者对 Shell 进行定制,允许开发者为自己的目标设备实现简单的命令行接口,或是完全定制的图形用户接口 Shell。由于 Windows CE Shell 的定制是 Windows CE 系统定制中最具特色的一部分,Shell 的定制尤为重要。本文就如何自定义 Windows CE Shell 提出了一种解决方案。

2 Windows CE Shell 的加载^[1]

了解 Windows CE 系统启动时各模块的加载顺

序,以及各模块的作用,可以加深对 Shell 与各模块的依赖关系的理解,对自定义 Shell 有很大帮助。下面介绍 Windows CE 系统的启动流程。

当系统复位或启动时,CPU 将跳转到 NK.exe 的入口,NK.exe 是 Windows CE 的核心模块。当 NK.exe 初始化完成,内核将启用中断并调用调度程序来安排在系统中运行的第一个线程,此时,内核将查找并运行应用程序 FileSys.exe。FileSys.exe 在此阶段是管理文件系统、数据库函数和注册表的最重要进程。一旦 FileSys 初始化完毕,就可以进行系统初始化。此时,内核需等待来自注册表的数据以继续启动过程。

在注册表的:[HKEY_LOCAL_MACHINE]\Init 键下的值,提供了应该作为启动过程的一部分而加载的一组进程的名称、次序以及相关性的。

图 1 展示了 Windows CE 系统启动的流程。

由图中可以看到 Windows CE 系统启动的基本流程(这里的 shell.exe 是 CESH 的 Windows 端,是基于命令行的监视程序,属于命令行外壳。而我们所说

收稿时间:2009-09-15;收到修改稿时间:2009-10-26

的定制 shell 是针对图 1 中的 explorer.exe 来说的)。在 FileSys 启动后,可以通过修改注册表来加载自己的程序,这里有一定限制,如果在你的程序中用到 device 和 gwes 模块中提供的服务,就必须在这两个模块加载完毕后,再加载自己的模块。因此我们在定制 shell 时需注意哪些功能可用。当定制完成自己的 Shell 应用程序后,我们可以修改注册表中的值,将 explorer.exe 用自己的 Shell 模块代替即可。

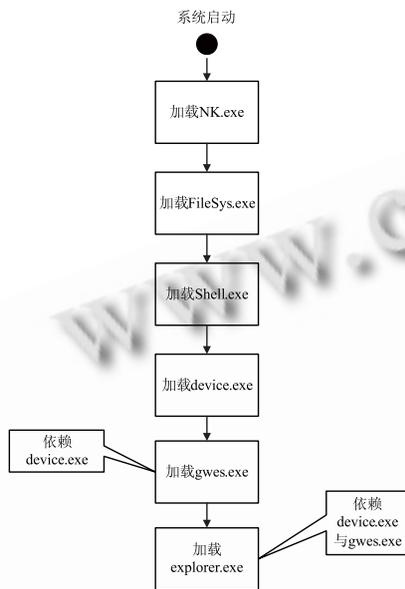


图 1 Windows CE 启动流程

3 定制Windows CE Shell

3.1 必要的功能^[2]

一个定制的 Shell 一般需要以下四项组成:

(a) Shell 应用程序;

为了创建一个定制的 Shell,必须首先创建一个 Shell 应用程序,Shell 应用程序是由两个主要窗口——桌面窗口和任务条窗口组成的。在设计 Shell 应用程序时,必须为桌面窗口和任务栏窗口创建一个 WindowProc 函数来处理窗口消息。

(b) 桌面窗口;

桌面窗口用来显示桌面壁纸并放置快捷方式图标,它总是设备的最底层窗口。在自定制时可以选择自己的应用程序窗口作为桌面窗口。

(c) 任务条窗口;

一个 Shell 应用程序可以通过一个窗口句柄调用 RegisterTaskBar 函数注册一个任务条窗口,该窗口

接收其它应用程序窗口的产生、关闭和激活的通知,同时,它也会接收用于特定功能的键序列,如开始菜单、任务管理等,用户可以根据需要选择实现其部分功能。

(d) 任务相关的消息处理器。

任务条窗口接收有 GWES 的窗口管理器发送的 WM_SHELLNOTIFY 消息通知代码,并处理发送通知的窗口。

3.2 自定义的 shell 模型^[3,4]

Windows CE 定制 Shell 每个嵌入式产品都需要修改或者编写自己的 Shell,一是因为不同场合需要不同的人机接口;二是都希望自己的产品有差异性。但很多基本的东西不会改变,Shell 与各个进程之间的通信,对进程的管理,这些都是必须的。在很多情况下,Windows CE 只运行一个应用程序,在这种情况下,该应用程序就可以直接作为该系统的 Shell;或者运行一个程序界面,通过界面的按钮调用其它的程序。

在下面自定义的 Shell 模型中,把 Shell 分成了六个模块,CAItTab、CMinDesktop、CHibernateMonitor、CMinShell、CMinTask 与 CTaskManager,如图 2。

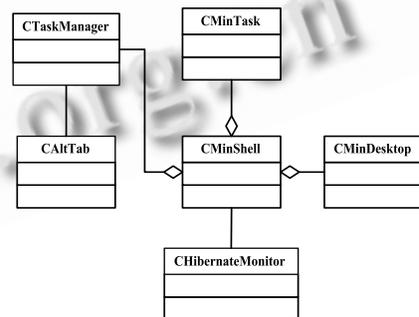


图 2 自定义 Shell 的六个模块

CAItTab 模块封装了任务管理器对话框,用户可以通过任务管理器来运行一个程序,或关闭一个正在运行的程序。该模块并非必须的,但会使调试更加方便,因此作为单独的模块便于取舍。

CTaskManager 模块负责对进程进行管理;包括管理应用程序的启动、关闭、对话框的显示、隐藏与销毁。该模块定义了一系列进程 ID 分别对应各应用程序模块,同时定义一系列用于 Shell 与各应用程序模

块通信的消息和通知命令。由于各应用程序生存期不同，如某些应用模块随 Shell 启动而启动，并且常驻内存，其他应用模块只在需要被调用时才载入内存执行。为了简化 Shell 对各应用模块的管理，定义了应用程序和 Shell 通信的消息(下面是三个最主要的消息)：

```
extern UINT WM_PROCESS_START
extern UINT WM_PRO_CREATE
extern UINT WM_PRO_DESTROY
```

WM_PROCESS_START 是 Shell 专用的，用于当 Shell 主界面的应用程序启动后通知 Shell 主界面已启动完成。WM_PRO_CREATE 在应用程序成功初始化后通知 Shell，并告诉 Shell 自己的主窗口句柄，应用程序也可能产生子窗口，子窗口在创建并初始化时同样发送 WM_PRO_CREATE 消息给 Shell，并告诉其窗口句柄。为了便于 Shell 统一管理，顶层窗口不论是隐藏还是被销毁，Shell 只需向其(窗口句柄)发送 windows 标准的 WM_CLOSE 消息，相应的窗口收到该消息时会根据自己实际情况判断窗口是被隐藏还是被销毁。

CMinTask 负责对任务条的窗口界面进行管理。任务条窗口的创建和注册都在该模块中完成，用户可以修改该模块定制合适的任务条界面。CMinDesktop 封装了一系列与桌面窗口相关的属性和方法。该模块负责桌面窗口的注册、创建，实现对桌面窗口相关消息的处理。CHibernateMonitor 负责对系统资源的监管；内存消耗、电池电量的相关信息处理在该模块中完成，并在适当时机通知系统进入睡眠状态。最后 CMinShell 模块将一系列与 Shell 相关模块组织起来并初始化工作。用户还可以根据需要扩充自己的 Shell 功能。

3.3 Shell 对进程的管理^[5]

Shell 对进程的管理包括对应用程序进程本身的管理和对进程通信的管理。在上述定制 Shell 中，Shell 对进程的管理主要体现在进程的启动、关闭。应用程序在启动成功时，将自身的句柄告诉给 Shell，完成对进程的注册，Shell 维护一张表对进程进行管理，这张表对应记录了进程的句柄，应用程序名称等等。应用程序退出时，也会发送通知给 Shell，Shell 将该程序对应的记录删除。

Shell 在管理进程通信时，主要用到两种方式：

- a) 两个进程之间通过 Shell 实现通信；
- b) Shell 告诉两个通信端各自的句柄，两个进程直接通信。

图 3 描述了 a)和 b)两种通信方式。

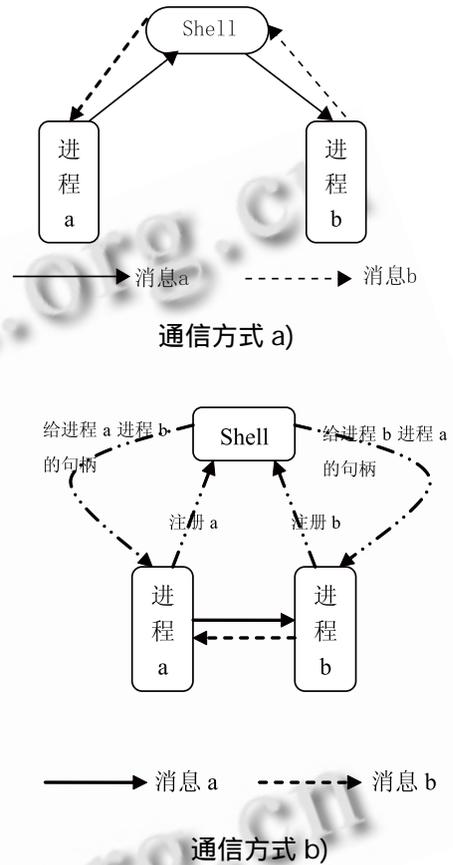


图 3 进程间通信方式

两种方式各有特点，对 a)方式而言，两个进程直接通过 Shell 建立通信，使得进程之间的耦合度降低，便于 Shell 的统一管理，但由于通信经过 Shell 转发，通信效率比 b)方式稍低。如电话模块的设计，在实时性要求不是很高时，来电监视模块可以单独运行而不启动电话的显示模块，在确认有来电后，Shell 启动相应的电话显示模块，并将数据传给显示模块。这种方式在某种程度上节约了系统资源。对 b)方式而言，进程间直接通信可以保证目标进程在第一时间收到消息，效率较高。如安防模块的设计，安防监视进程发现安全隐患时，直接发送消息给安防显示模块，显示模块及时以显示、或蜂鸣等方式引人注目，在这之前，Shell 必须告诉两个进程各自的句柄，以便于 a 与 b 之间直接通信。

3.4 Shell 工作流程

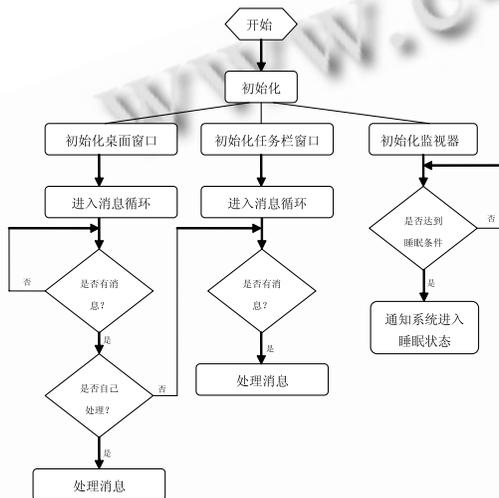


图 4 Shell 的消息循环过程

Shell 随系统的启动而启动并完成初始化。这里的 Shell 即是一个 win32 的应用程序, 遵循 win32 的消息循环。系统的各应用模块通过发送消息与 Shell 进行通信, Shell 的各模块接收消息并进行相应处理, 或发送消息给各应用模块, 通知其完成特定任务。上述

自定义 Shell 的工作流程如图 4 所示。

4 总结

定制 Windows CE Shell 是 Windows CE 嵌入式软件开发的一个重要环节。Microsoft 的 Platform Builder 提供了一系列的 Shell 模块供开发者选择, 开发者也可以根据需求定制自己的 Shell, 建立自己的消息管理机制, 实现对各个应用模块的管理。

参考文献

- 1 Douglas Boling. Windows CE 程序设计. 北京: 北京大学出版社, 1999. 727 - 735.
- 2 张冬泉, 谭南林, 王雪梅, 焦风川. Windows CE 实用开发技术. 北京: 电子工业出版社, 2006. 125 - 129.
- 3 Gamma E, Helm R, Johnson R, Vlissides J. 设计模式. 北京: 机械工业出版社, 2007.
- 4 张志华, 田英爱, 秦奕青. 模式在对象标识管理器中的应用. 微计算机信息, 2006, 8(1): 276 - 278.
- 5 杨俊. Windows Shell 编程机制分析及应用研究. 现代计算机, 2007. 3: 344 - 345.