

利用 C 语言的 SDK 开发测试管理系统

赵思奇 王仕秀

摘要 介绍以 C 语言的 SDK 方式来开发软件工程的过程，讲述其编程实现及开发技巧。本方法以窗口为核心，以事件驱动为动力，巧妙地将程序代码与用户界面分开，很值得开发人员借鉴。

关键词 SDK，事件驱动，资源文件

一、引言

Win32 是指支持 32 位计算机的 Windows 环境。在此环境下，多种开发语言都可通过调用 API（应用程序接口）函数，实现微软风格的窗口化界面的编程。这其中，基于 Microsoft VC 的编译环境，以 C++/MFC 或 C/SDK 的方式编程，是应用极广的常用方式。实际上，C 或 C++ 都可以使用 SDK 或 MFC 来设计 Win32 程序，区别在于 SDK 是直接调用 API 函数，而 MFC 一般是通过类的包装后由 MFC 对象来调用 API。本文举例的测试工程师管理系统项目，是利用 Microsoft Visual C++ 6.0 的编译环境，以 C/SDK 的方式编写的。

二、具体实现

1. 需求分析

关于这个项目的需求分析，简化起来有以下几点：

主菜单，其中菜单项有“信息管理”、“薪水计算”、“排序”、“生成报表”、“关于”。

除“薪水计算”外各菜单项均有子菜单，如信息管理子菜单包括输入、查询、删除、修改、导入、导出、屏显、结束等功能选项。排序子菜单包括按照编号、姓名、工龄的升降序排序，生成报表子菜单包括信息和薪水报表，关于子菜单包括关于和帮助。

点击“薪水计算”后弹出对话框，如图 1 所示。用户正确输入后，测试工程师的薪水被计算出。

在信息管理子菜单中，选择“导出信息”可以保存测试工程师信息为二进制文件或 ASCII 码文件。导入信息可以将这些外部文件重新输入给系统；选择“退出系统”可提示用户先保存再退出，如图 2 所示。

查询信息功能提供有按编号或姓名两种查询方式，如图 3 所示。删除功能也提供按编号或姓名删除。“关于”子菜单中提供有联机帮助。

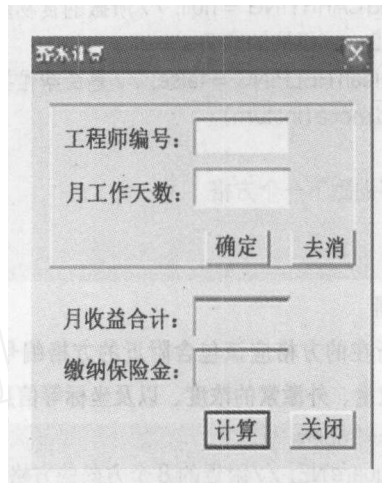


图 1 薪水计算对话框

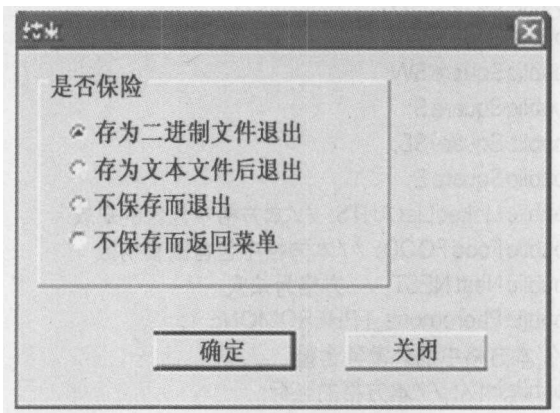


图 2 退出系统对话框

2. 项目主框架

Win32 编程是以窗口为核心，以事件驱动为动力，将程序代码与用户界面分开，开发时在资源文件 .rc 中描述所需要的资源。编译时资源编译器将所有用户资源集中制作出一个 .res 文件，再与程序代码结合在一起，生成 .exe 文件。明白了需求后，可以建立工程了。主框架如下：

(1) 主程序文件 WinMain.c

包括两个函数：

```
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE
```

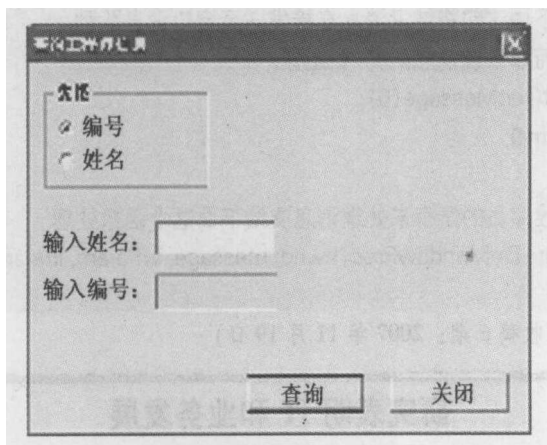


图 3 查询信息对话框

hPrevInstance, LPSTR lpCmdLine, int nCmdShow)

如同 DOS 下 C 语言程序的入口点是 main 函数，Windows 程序的入口点是 WinMain 函数。其中主要包含五大部分：填充窗口类结构、注册窗口、建立窗口、显示更新窗口、消息循环。

LRESULT CALLBACK WndProc (HWND hwnd, UINT message, WPARAM wParam, LPARAM lParam)

主窗口消息处理函数，比如滚动条、屏幕客户区重绘的消息处理，还有调用各对话框消息处理函数。

(2) 其他 .C 文件

About.c (关于对话框处理函数)

EndProc.c (结束对话框处理函数)

InputProc.c (输入对话框处理函数)

DeleteProc.c (删除对话框处理函数)

SearchProc.c (查询对话框处理函数)

ReportProc.c (信息报表函数、薪水报表函数)

ImportProc.c (导入信息对话框处理函数)

(3). RC 文件 (资源文件)

绘制出各对话框(约 10 个)、主菜单、加速键表、字符串表、Logo 图标。

(4) WinMain.h 文件

测试工程师结构体定义、大部分全局变量、主要函数声明。

3. 主程序文件讲解

下面主要讲解一下主程序文件 WinMain.c 中的两个函数：

(1) 入口函数

```
int WINAPI WinMain (HINSTANCE hInstance,
//当前实例句柄
HINSTANCE hPrevInstance, //前一个实例句柄
LPSTR lpCmdLine, //命令行字符串
int nCmdShow //窗口显示方式)
{
```

```
MSG msg; //定义消息结构体变量
WNDCLASS wndclass; //定义窗口结构体变量
HACCEL hHotkey; //加速键表句柄
HDC hdc; //定义客户区变量
...
//填充窗口类结构
wndclass.style = CS_HREDRAW | CS_VREDRAW;
//表明窗口含有
wndclass.lpfnWndProc = WndProc;
//表明主窗口消息处理函数是 WndProc
wndclass.hIcon = LoadIcon(hInstance, MAKEINTRESOURCE
(IDI_ICON1)); //加载 Logo
wndclass.hCursor = LoadCursor(NULL, IDC_ARROW);
//加载光标
wndclass.hbrBackground = (HBRUSH) GetStockObject
(WHITE_BRUSH); //使用系统默认画刷
wndclass.lpszMenuName = MAKEINTRESOURCE
(IDR_MAINMENU); //加载主菜单
...
//注册窗口类
if (!RegisterClass(& wndclass))
{
    MessageBox(NULL, "本程序需要 Windows XP/2000 操
作系统!", szAppName, MB_CONERROR);
    return 0;
} //创建窗口
hwnd = CreateWindow(szAppName, //指定窗口类名
TEXT("北大青鸟软件测试工程师管理系统"),
//窗口左上角标题
WS_OVERLAPPEDWINDOW | WS_VSCROLL |
WS_HSCROLL, //窗口样式,有垂直和水平滚动条
//(-----省略部分参数)
hInstance, //当前实例的句柄
NULL //指向一个传递给窗口的参数值指针
);
//显示窗口
ShowWindow(hwnd, nCmdShow);
//低优先级,一会儿才显示
UpdateWindow(hwnd);
//高优先级,立即显示加载加速键
hHotkey = LoadAccelerators(hInstance,
MAKEINTRESOURCE(IDR_MAINACCR));
//消息循环
while (GetMessage(& msg, NULL, 0, 0))
//从系统消息队列取消息
{ //若不是加速键消息,则调用 WndProc() 进行处理,
//即主窗口消息处理函数
if (!TranslateAccelerator(hwnd, hHotkey, & msg))
```

```

{
    TranslateMessage(& msg);
    //把虚拟键消息翻译为字符消息
    DispatchMessage(& msg);
    //把消息分配到相应窗口的消息队列
}
}
return msg.wParam;
}

```

(2)最重要的函数，主窗口消息处理函数

```

LRESULT CALLBACK WndProc(HWND hwnd, UINT message,
WPARAM wParam, LPARAM lParam)
{
    //定义变量,包括滚动条的页高行高、保存文件时的可选文件
    //类型等.
    //开始消息处理
    switch(message)
    {
        case WM_CREATE: //清屏、滚动条的字长宽取值
        case WM_SIZE: //滚动条的结构 SCROLLINFO 的填充
        case WM_VSCROLL: //垂直滚动条移动处理
        case WM_HSCROLL: //滚动条水平移动处理
        case WM_KEYDOWN: //键盘控制垂直、水平滚动条
        case WM_PAINT: /* 清屏、无效区重绘(分类调用本系统中
所有要用的屏幕输出函数),这样别的地方想向屏幕输出时,只
需要发此 WM_PAINT 命令就可以了.最重要的消息之一,*/
        if(ShowAllStatus == 1)
        ShowAllInfo(hwnd); //输出所有的测试工程师信息函数
        if(ShowOneStatus == 1)
        ShowOneInfo(hwnd); //输出某个特定测试工程师信息函数
        case WM_COMMAND:
        switch(LOWORD(wParam))
        //点击主菜单及子菜单各项时的处理
        {
            case IDM_INPUT: /* 弹出输入信息对话框, InputInfoProc
(1)是输入信息对话框消息处理函数,位于 InputProc.c 文件,
当在主菜单中选择信息管理后再选择输入信息时,系统就会调用
InputInfoProc()函数进行处理.*/
            DialogBox(hInstance, MAKEINTRESOURCE
(IDD_ADDINFO), hwnd, InputInfoProc);
            return TRUE;
        case IDM_SEARCH: //在此处调用查询对话框消息处理函数
        case IDM_DELETE: //在此处调用删除对话框消息处理函数
        ...
        case IDM_SR: //薪水报表,保存通用对话框的结构填充
        case WM_DESTROY: /* 选择关闭对话框时的处理,包括按
下 ALT + F4,点右上角关闭,及对话框中的关闭按键.*/
        case WM_DESTROY:
    
```

```

//下面这句将结束消息直接发送至窗口消息队列,
//而非系统消息队列,更迅速.
PostQuitMessage(0);
return 0;
}
//程序员的所有未处理消息交给下面这个函数处理
return DefWindowProc(hwnd, message, wParam, lParam);
}

```

(收稿日期:2007年11月19日)

研究表明 IT 和业务发展 不一致会对财务产生严重影响

研究表明终端用户体验监控对于有效的服务等级协议、IT 性能和成本控制至关重要

康普科伟迅公司 (Compuware Corporation, NASDAQ 交易代码: CPWR) 今天发布由 Compuware 公司发起, Forrester Consulting 进行的一项调查结果。该结果表明: 81% 的机构已采用正式的服务等级协议 (SLA), 平均 74% 的时间与该协议相一致。许多公司的报告也表明, 拙劣的应用性能会导致成本增加和收益损失。在该研究中, Forrester Consulting 得出的结论是, IT 机构与服务等级协议 (SLA) 不相符的首要原因是, 业务部门的期望值远远超过了 IT 所能达到的程度。

导致期望值不匹配的关键原因是采用了以 IT 为中心, 不符合业务目标的服务等级参数。Forrester 研究表明, 41% 的受访者认为他们对服务等级的认识都处在基础层面, 他们也不经常向他们的上级领导提供符合服务等级协议的信息。此外, 40% 的受访者认为他们的服务等级报告缺少他们的上级领导所要求的信息。

康普科伟迅公司 IT 服务管理副总裁 Steve Tack 表示: “仅仅依赖专注于技术的参数, 很难实现 IT 与业务的有效对话和沟通, 因此很难推动积极有效的服务管理。”

Tack 表示: “终端用户体验 (EUE) 监控从终端用户的角度出发, 通过提供服务质量可视的 IT, 最终应对这些挑战。终端用户体验 (EUE) 有利于 IT 与业务更有效地沟通, 建立现实和一致的期望值, 并根据业务影响有效优化其努力。”

Forrester Research 的副总裁兼首席分析师 Jean - Pierre Garbani 表示: “IT 与业务之间契合度的最终评判者是终端用户。如果将二者的契合度定义为可用性、性能、易用性及准确性与用户期望值的一致程度, 那么监控终端用户性能是 IT 人士所知的满足这些期待值的唯一方法。” (来自 Forrester Wave: 基于应用的终端用户体验监控, 2007 年第二季度, Forrester Research, Inc., 2007 年 6 月)。