

# KV Betriebssysteme

## Windowsprogrammierung

### KV Betriebssysteme Windowsprogrammierung Aufbau von Windows-Programmen

Michael Sonntag

Inst. f. Informationsverarbeitung und Mikroprozessortechnik (FIM)  
Johannes Kepler Univ. Linz, Altenbergerstr. 69, A-4040 Linz, Austria  
E-Mail: sonntag@fim.uni-linz.ac.at

Michael Sonntag

KV Betriebssysteme

1

### Grund-Aufbau

#### Code

- Der Programmcode, exkl. DLLs

#### Daten

- Statische Initialisierungsdaten für Variablen, ...

#### Ressourcen

- Daten zur Definition der sichtbaren Programmteile:
  - Dialogfenster, Bitmaps, Strings, ...
  - Zum leichteren Austausch, z. B. für Internationalisierung
- Ressourcen unabhängig vom Programm: Änderung durch Linken
- Standard-Ressourcen: Siehe nächste Folie
- Custom-Ressource: Beliebig von der Applikation definiert

Michael Sonntag

KV Betriebssysteme

2

### Arten von Ressourcen (1)

- **Dialogfenster**
  - Samt darin enthaltenen Controls (Listboxen, Buttons, ...)
  - Spezifizierung ist Geräteunabhängig
- **Menüs**
  - Beliebig verschachtelte Popup-Menüs
- **Bitmaps**
  - Rastergrafiken, die im Programm angezeigt werden können
- **Icons**
  - Bitmaps mit einer zusätzlichen Sichtbarkeitsmaske
- **Accelerators**
  - Zurodnung von Tastenkürzeln zu Menübefehlen
- **Stringtabellen**
  - Mit symbolischen Namen versehene Texte.

Michael Sonntag

KV Betriebssysteme

3

### Arten von Ressourcen (2)

- **Cursor**
  - Bitmaps für Cursors
- **Font**
  - Schriftart (Angabe des Dateinamens)
- **Message-Table**
  - Spezielle Strings für Ereignis-Logging und Formatierung von Ausgaben
- **Version**
  - Informationen zur Version des Programms (Produktversion, Dateiversion, Betriebssystem, ...)

Michael Sonntag

KV Betriebssysteme

4

### Erzeugen von Ressourcen (1)

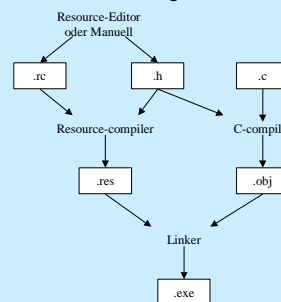
- Ressourcen können in Textdateien (.rc) beschrieben werden, ein Resource-Compiler erzeugt daraus eine binäre Datei (.res), die dann vom Linker zusammen mit dem Code und den Daten zum Exe-File gebunden wird.
- Die einzelnen Ressourcen werden über Nummern bzw. Namen angesprochen, die in einer gesonderten Datei gespeichert werden (meist „resource.h“).
- **LPTSTR MAKEINTRESOURCE(WORD wInteger)**  
Wandelt eine Resource-Identifikationsnummer in einen entsprechenden String um; für Resource-Management-Funktionen hilfreich

Michael Sonntag

KV Betriebssysteme

5

### Erzeugen von Ressourcen (2)



Michael Sonntag

KV Betriebssysteme

6

# KV Betriebssysteme

## Windowsprogrammierung

### Erzeugen von Ressourcen (3)

#### Händisch

- NUR für einfache Ressource wie Menüs, Stringtabellen, ... geeignet!

#### Tools

- Für alles andere sollte ein Ressourcen-Editor verwendet werden.
- Diese erlauben meist auch das Verändern von fertigen Programmen, da sie nicht nur mit Quell-Ressourcen (.rc), sondern auch compilierten (.res) und gelinkten arbeiten können.

Michael Sonntag

KV Betriebssysteme

7

### Aufteilung der Informationen

#### 1. Resource.h

- Definition der Namen der Ressourcen als Konstanten.

##### Konvention:

- IDI\_???: Icons
- IDM\_???: Menüs
- IDB\_???: Bitmaps
- IDD\_???: Dialog
- IDS\_???: Strings

#### 2. <Programmname>.rc

- Eigentliche Definition der Ressourcen, bzw. Spezifikation des Namens der externen Datei

Michael Sonntag

KV Betriebssysteme

8

### Einbinden von Ressourcen

#### Verwendung im Programm

- FormatMessage(...) für Message-Tabellen
- LoadAccelerators(...) für Tastenkürzel
- LoadImage(...) für Icons, Cursors, Bitmaps und Metafiles
- LoadMenu(...) für Menüs
- LoadString(...) für Stringtabellen-Einträge

#### Beispiel:

```
#include "resource.h"
...
WNDCLASSEX wc;
wc.hIcon = LoadIcon(hInstCurr, MAKEINTRESOURCE(IDI_MAINICON));
```

Michael Sonntag

KV Betriebssysteme

9

### KV Betriebssysteme Windowsprogrammierung Arten von Ressourcen

Michael Sonntag

Inst. f. Informationsverarbeitung und Mikroprozessortechnik (FIM)  
Johannes Kepler Univ. Linz, Altenbergerstr. 69, A-4040 Linz, Austria  
E-Mail: sonntag@fim.uni-linz.ac.at

Michael Sonntag

KV Betriebssysteme

10

### Icons

nameID ICON dateiname

Definiert ein Icon als den Inhalt einer Datei.

#### Funktionen:

- HANDLE LoadImage(„IMAGE\_ICON,,“): Laden von Icons
- BOOL DrawIcon(HDC hDC, int x, int y, HICON hIcon): Icon anzeigen
- BOOL GetIconInfo(HICON hIcon, PICONINFO pInfo): Informationen über das Icon einholen (Hotspot, Zeichen-Bitmap, Masken-Bitmap).

Michael Sonntag

KV Betriebssysteme

11

### Menüs - Aufbau (1)

menuID MENU {...}

Eine horizontale Menüzeile mit beliebigen Einträgen; kann auch Popups enthalten.

#### Erlaubt:

- MENUITEM
- POPUP

POPUP: Ein vertikales Menü, das Menüeinträge und weitere Submenüs enthalten kann.

- POPUP text, optionlist {...}
- Optionen: CHECKED, GRAYED, HELP, INACTIVE, MENUBREAK, MENUBARBREAK, MENUITEM

Michael Sonntag

KV Betriebssysteme

12

# KV Betriebssysteme

## Windowsprogrammierung

### Menüs - Aufbau (2)

#### MENUITEM

- MENUITEM SEPARATOR Trennstich zwischen Einträgen
- MENUITEM text, id, optionlist
  - Optionen: Wie bei POPUP

• Sowohl das Menü als auch alle Menüeinträge (nicht aber die Popups!) benötigen eine Nummer: IDM\_???

Michael Sonntag

KV Betriebssysteme

13

### Menüs - Aufbau (3)

Mit einem "&" kennzeichnet man den folgenden Buchstaben als Tastenkürzel für die Bedienung über die Tastatur. Er wird automatisch unterstrichen angezeigt.

- Nicht mit Accelerators verwechseln!
- Diese werden mit "\t" angefügt, sind aber rein optisch als Hilfe für den Benutzer da. Ihre Funktion ist extra zu definieren und extra zu implementieren. Sie sind auch nicht zwingend an Menüeinträge gebunden.

Michael Sonntag

KV Betriebssysteme

14

### Menüs - Events

#### WM\_COMMAND

- Wenn ein Menüpunkt ausgewählt wurde
- Menü-ID ist in LOWORD(wParam) zu finden

#### WM\_INITMENU:

- Vor dem Anzeigen zur Aktualisierung durch den Benutzer

#### WM\_INITMENUPOPUP:

- Wie oben, aber für Popups

#### WM\_CONTEXTMENU:

- Bei Rechts-clicks. Zum öffnen von Kontextmenüs falls gewünscht; sonst an DefWindowProc! Siehe TrackPopupMenuEx(...).

Michael Sonntag

KV Betriebssysteme

15

### Menüs - Beispiel (1)

#### Program.rc:

```
#include "resource.h"
...
IDM_MAINMENU MENU {
    POPUP "&Game" {
        MENUITEM "&new\tCtrl+N", IDM_GAME_NEW
        MENUITEM SEPARATOR
        MENUITEM "E&xit", IDM_GAME_EXIT
    }
    POPUP "&Help" {
        MENUITEM "&About\tF1", IDM_HELP_ABOUT
    }
}
```

Michael Sonntag

KV Betriebssysteme

16

### Menüs - Beispiel (2)

#### Program.c:

```
LRESULT CALLBACK WndProc (HWND hWnd,
    UINT message, WPARAM wParam, LPARAM lParam) {
    switch (message) {
        case WM_COMMAND:
            switch (LOWORD(wParam)) {
                case IDM_GAME_EXIT: PostQuitMessage(0); break;
                case IDM_HELP_ABOUT: About(); break;
            }
            break;
        .....
    }
    int InitApplication(HINSTANCE hInstCurr) {
        wc.lpszMenuName=MAKEINTRESOURCE(IDM_MAINMENU);
```

Michael Sonntag

KV Betriebssysteme

17

### Menüs - API-Funktionen

#### Ausgewählte Funktionen:

- GetMenuItemInfo(): Menü-Infos (ID, Typ, State, Submenu, ...)
- SetMenuItemInfo(): Menü-Infos festlegen
- EnableMenuItem(): Enabled, Disabled, Grayed
- GetMenu(): Menü eines Fensters holen
- GetSubMenu(): Handle für Submenü holen
- LoadMenu(): Menü aus Ressourcen laden
- TrackPopupMenuEx(): Popup-Menü darstellen und durchführen

Michael Sonntag

KV Betriebssysteme

18

# KV Betriebssysteme

## Windowsprogrammierung

### Accelerators

- Accelerators sind Tastenkürzel für Menüeinträge und sonstige Funktionen. Auch wenn sie nicht in Menüs aufscheinen, werden sie identisch behandelt: WM\_COMMAND.
- Im Programm müssen Accelerators mit LoadAccelerators geladen werden.
- Die Hauptschleife ist wie zu verändern; siehe Beispiel. Wenn ein Accelerator-Tastendruck vorliegt (Rückgabe!= 0), darf TranslateMessage und DispatchMessage NICHT mehr aufgerufen werden!
- Die Tastenkombination ALT+F4 ist im Systemmenü des Hauptfensters als Kürzel zum Beenden des Programms definiert und kann daher nicht verwendet werden.

Michael Sonntag

KV Betriebssysteme

19

### Accelerators - Definition

```
accTableName ACCELERATORS {
    event, idValue, [type, options]
}
event: "character", ASCII-Code, virtuelle Bezeichnung
idValue: Bezeichner (Code für WM_COMMAND)
type: ASCII oder VIRTKEY
options: ALT, SHIFT, CONTROL, NOINVERT

•Bei WM_COMMAND ist HIWORD(wParam)=1, falls das Kommando von einem Accelerator ausgelöst wurde.
```

Michael Sonntag

KV Betriebssysteme

20

### Accelerators - Beispiel (1)

```
Program.rc:
IDA_SHORTCUTS ACCELERATORS {
    "N", IDM_GAME_NEW, VIRTKEY, CONTROL
    VK_F1, IDM_HELP_ABOUT, VIRTKEY
}

CTRL+N: WM_COMMAND+IDM_GAME_NEW
F1: WM_COMMAND+IDM_HELP_ABOUT
```

Michael Sonntag

KV Betriebssysteme

21

### Accelerators - Beispiel (2)

```
Program.c:
int WINAPI WinMain (HINSTANCE hInstCurr,
    HINSTANCE hInstPrev, LPSTR lpszCmd, int nCmdShow) {
    HACCEL hAccel;
    hAccel=LoadAccelerators(hInstCurr,
        MAKEINTRESOURCE(IDA_SHORTCUTS));
    while (GetMessage (&msg, NULL, 0, 0)>0) {
        if(!TranslateAccelerator(hWndMain, hAccel, &msg)) {
            TranslateMessage (&msg); DispatchMessage (&msg);
        }
    }
    return msg.wParam;
}
```

Michael Sonntag

KV Betriebssysteme

22

### Dialoge - Arten

Dialoge sind spezielle Fenster zur Interaktion mit dem Benutzer. Im Gegensatz zu Fenstern sind sie meist nur kurz sichtbar und dienen zum Einstellen von Parametern bzw. für Mitteilungen.

#### Typen:

- **Modal:** Das Programm läuft erst weiter, wenn der Dialog geschlossen wird (Keine Interaktion mit Hauptfenster möglich).
  - System-Modal: Das gesamte Windows-System wird gesperrt und NUR dieser Dialog ist zugänglich (z. B. Shutdown-Dialog)
- **Nicht-Modal:** Es kann beliebig zwischen Hauptfenster und Dialog gewechselt werden, Z. B. Such-Dialoge
  - Diese sind um einiges komplexer, da meist wechselseitiges Update erforderlich ist!

Michael Sonntag

KV Betriebssysteme

23

### Dialoge - Elemente

#### Windows-Elemente:

- CTEXT: Zentrierter statischer Text
- PUSHBUTTON: Normaler Button
- RADIOBUTTON, CHECKBOX: Radio/Check-Button
- GROUPBOX: Rechteck zur Gruppierung (Radio-Buttons!)
- COMBOBOX, LISTBOX: Auswahl aus einer Liste
- EDITTEXT: Texteingabe (Ein-/Mehrzeilig)
- .....

#### “Custom-Controls”:

- Werden als OCX oder sonstige Dateien (z. B. ActiveX, ...) zur Verfügung gestellt. Viele Hersteller bieten Bibliotheken an.

#### Tip: Dialog-Editor benutzen!

Michael Sonntag

KV Betriebssysteme

24

# KV Betriebssysteme

## Windowsprogrammierung

### Dialoge - Events

#### WM\_COMMAND:

- Wird von den einzelnen Controls an die Dialog-Fensterprozedur geschickt. LOWORD(wParam) ist die ID des Controls, welches die Nachricht ausgelöst hat (z. B. IDD\_ABOUT\_OK), HIWORD(wParam) den Subtyp (Control-spezifisch; z. B. BN\_CLICKED).

#### WM\_INITDIALOG:

- Wird an den Dialog gesendet unmittelbar bevor er angezeigt wird. Typischerweise werden hier die anzuzeigenden Werte initialisiert (Text in Edit-Controls, Radio-/Checkboxes setzen, Listboxen füllen, ...).
- TRUE zurückgeben, damit Windows den Fokus automatisch setzt (selbst ein Control aktiviert: FALSE).

Michael Sonntag

KV Betriebssysteme

25

### Dialoge - Programmierung (1)

#### Dialog erzeugen (nur modale!):

- `int DialogBox(HINSTANCE hInst, LPCTSTR lpTemplate, HWND hWndParent, DLGPROC lpDlgFnc)`
  - `hInst`: Instanz des Programmes, zu dem der Dialog gehört
  - `lpTemplate`: Dialog-Ressource (MAKEINTRESOURCE)
  - `hWndParent`: Übergeordnetes Fenster
  - `lpDlgFnc`: Dialog-Fensterprozedur
- Die Funktion kehrt erst zurück, wenn der Dialog geschlossen wurde (Modaler Dialog!). Rückgabewert: Von `EndDialog`

Michael Sonntag

KV Betriebssysteme

26

### Dialoge - Programmierung (2)

#### Dialog-Fensterprozedur:

- Wie jedes Fenster braucht auch ein Dialog eine Fensterprozedur:
- `BOOL CALLBACK DialogProc(HWND hWndDlg, UINT uMsg, WPARAM wParam, LPARAM lParam)`
- Außer bei `WM_INITDIALOG` sollte nur dann 0 zurückgegeben werden, wenn die Nachricht nicht behandelt wurde.
- `DefWindowProc` darf NICHT aufgerufen werden! Bei einem Rückgabewert von 0 wird die Nachricht von Windows selbst behandelt.

Michael Sonntag

KV Betriebssysteme

27

### Dialoge - Programmierung (3)

#### Dialog beenden:

- `BOOL EndDialog(HWND hDlg, int nResult)`
  - `hDlg`: Handle des Dialogs
  - `nResult`: Wert, der von `DialogBox` zurückgegeben werden soll
- Bei Erfolg wird ein Wert ungleich Null zurückgegeben.

#### Dialog-Ressourcen:

- Es gibt viele Controls; siehe Online-Hilfe oder Dokumentation
- Größe und Position werden in logischen Einheiten spezifiziert

Michael Sonntag

KV Betriebssysteme

28

### Dialoge - Programmierung (4)

#### Zugriff auf Controls:

- `HWND GetDlgItem(HWND hDlg, int nIDDlgItem)`
  - `hDlg`: Handle des Dialogs
  - `nIDDlgItem`: ID des gewünschten Elements (z. B. `IDD_RUN`)
- Über das zurückgegebene Handle können dann Control-spezifische Funktionen aufgerufen werden:
  - `GetWindowText`, `SetWindowText`: Text von Buttons und statischem Text holen bzw. setzen.
  - Status von Check- und Radio-Buttons holen bzw. setzen:
  - `Checked=SendMessage(handle,BM_GETCHECK,0,0);`
  - `SendMessage(handle,BM_SETCHECK,checked,0);`

Michael Sonntag

KV Betriebssysteme

29

### Dialoge - Beispiel (1)

#### Dialog.rc:

```
IDD_ABOUT_DIALOG 10,10,150,70
CAPTION "About Dialog"
STYLE DS_MODALFRAME|WS_POPUP|WS_CAPTION|WS_SYSMENU
{
    PUSHBUTTON "&Ok", IDD_ABOUT_OK,30,45,30,14,
        WS_CHILD | WS_VISIBLE | WS_TABSTOP
    PUSHBUTTON "&Cancel", IDD_ABOUT_CANCEL,90,45,30,14,
        WS_CHILD | WS_VISIBLE | WS_TABSTOP
    CTEXT "Version 1.0", -1, 0,24,150,8,
        WS_CHILD | WS_VISIBLE | WS_GROUP
    CTEXT "Ein einfaches Windows-Programm mit Dialog-Box", -1,09,9,150,8,
        WS_CHILD | WS_VISIBLE | WS_GROUP
}
```

Michael Sonntag

KV Betriebssysteme

30

# KV Betriebssysteme

## Windowsprogrammierung

### Dialoge - Beispiel (2)

```
BOOL CALLBACK AboutDialogProc(HWND hwndDlg,
    UINT uMsg, WPARAM wParam, LPARAM lParam) {
    switch(uMsg) {
        case WM_INITDIALOG: return TRUE;
        case WM_COMMAND:
            if(wParam==IDD_ABOUT_OK) {
                EndDialog(hwndDlg, 0); return TRUE; }
            if(wParam==IDD_ABOUT_CANCEL) {
                EndDialog(hwndDlg, 1); return TRUE; }
            break;
        case WM_CLOSE: EndDialog(hwndDlg, 1); return TRUE;
    }
    return FALSE;
}
```

Michael Sonntag

KV Betriebssysteme

31

### Dialoge - Beispiel (3)

```
LRESULT CALLBACK WndProc (HWND hWnd, UINT message,
    WPARAM wParam, LPARAM lParam) {
    switch (message) {
        case WM_COMMAND:
            switch(LOWORD(wParam)) {
                case IDM_HELP_ABOUT:
                    res=DialogBox(hInst, MAKEINTRESOURCE(IDD_ABOUT),
                        hWndMain,AboutDialogProc);
                    sprintf(str,"Der Dialog gab den Wert \"%d\" zurück!",res);
                    MessageBox(hWnd,str,"Rückgabewert des Dialogs",
                        MB_OK | MB_ICONINFORMATION);
                    return res;
            }
    }
```

Michael Sonntag

KV Betriebssysteme

32

### Dialoge - Beispiel (4)

**Ergebnis:**



Michael Sonntag

KV Betriebssysteme

33