Lecture

18 & 19

Review of Last Lecture

Review of Today’s Lecture

.rc resource file (text file containing many resource statement)

|

Compile to .res file (using Resrouce Compiler)

|

Link with other files to make final EXE (using linker)

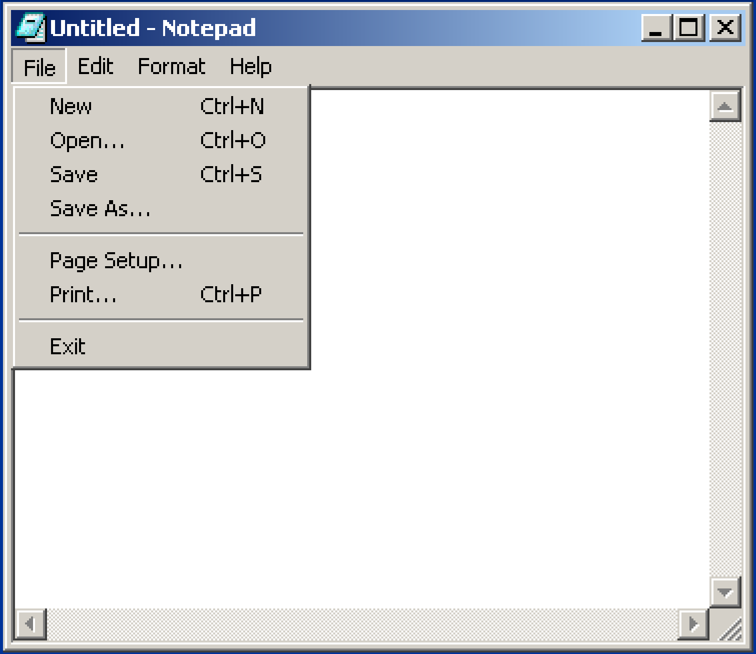
Menu

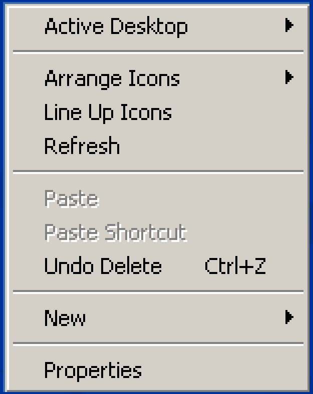
Windows 98/2000 include menu animation feature.

Types of Menu Items

drop-down menu

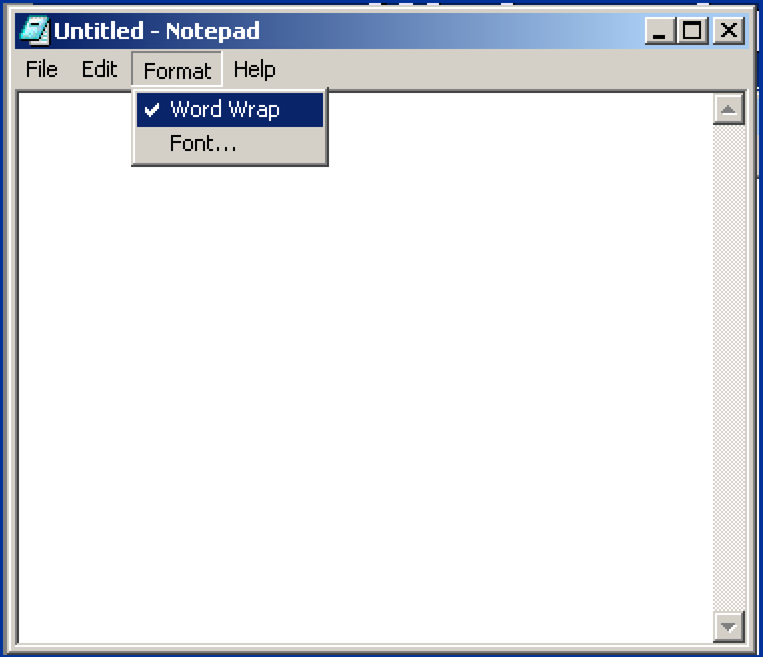
Popup menu



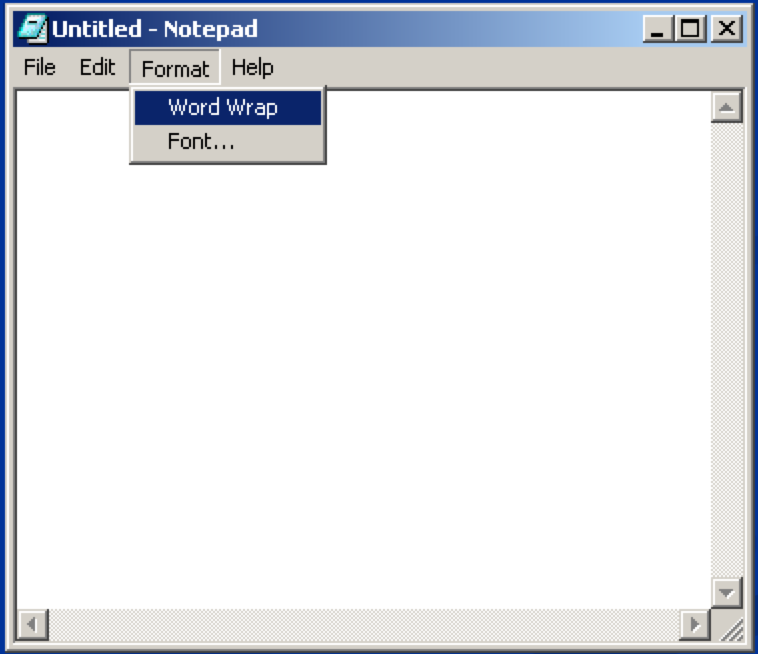


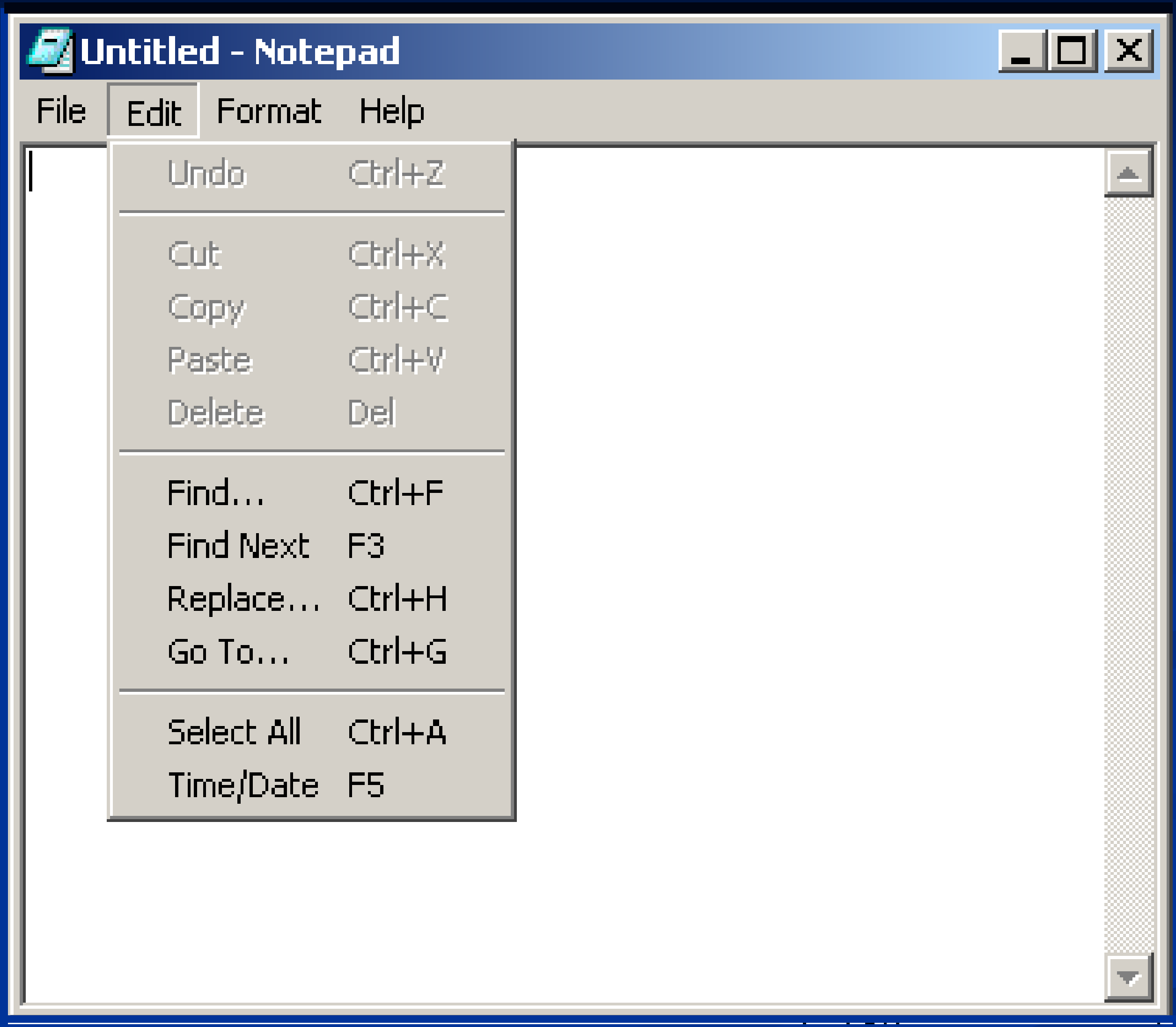
States of menu items

Checked



Unchecked




Untitled - Notepad
File
Edit
Format
Help
Undo Ctrl+Z
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete Del
Find... Ctrl+F
Find Next F3
Replace... Ctrl+H
Go To... Ctrl+G
Select All Ctrl+A
Time/Date F5

States of menu items

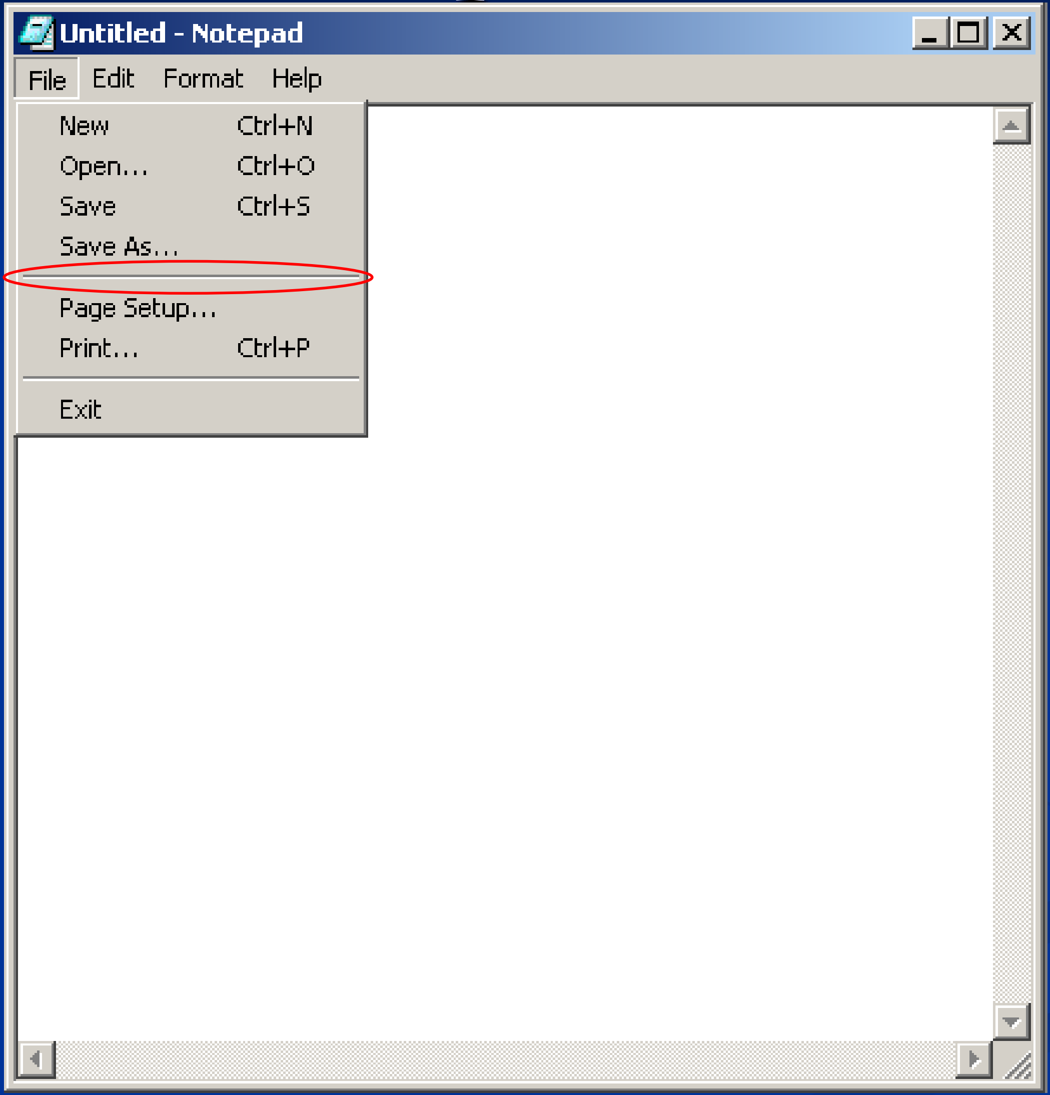
Grayed

while selected, clicking it has no effect

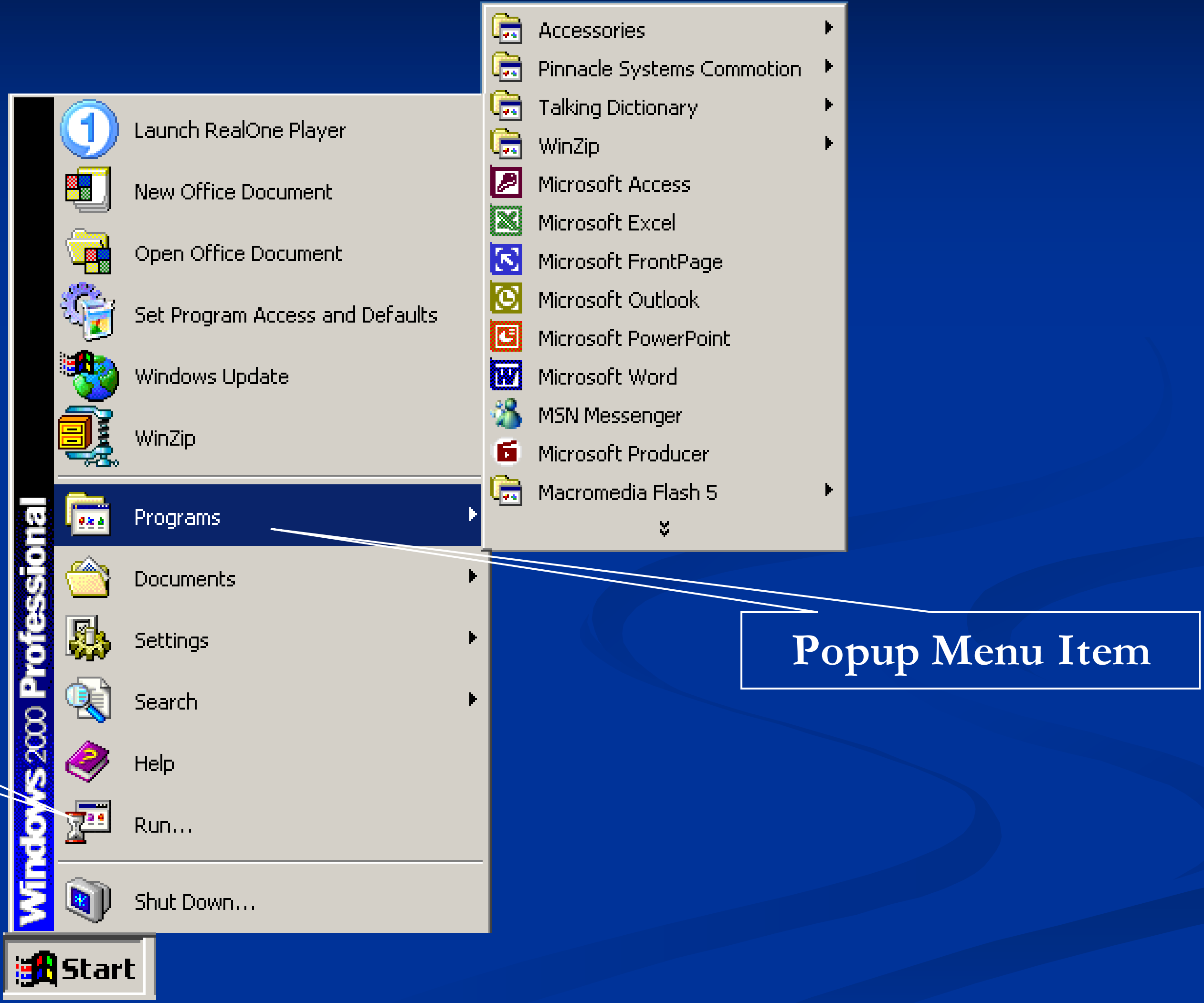
Inactive

can not be selected

Separator


Untitled - Notepad
File
Edit
Format
Help
New Ctrl+N
Open... Ctrl+O
Save Ctrl+S
Save As...
Page Setup...
Print... Ctrl+P
Exit

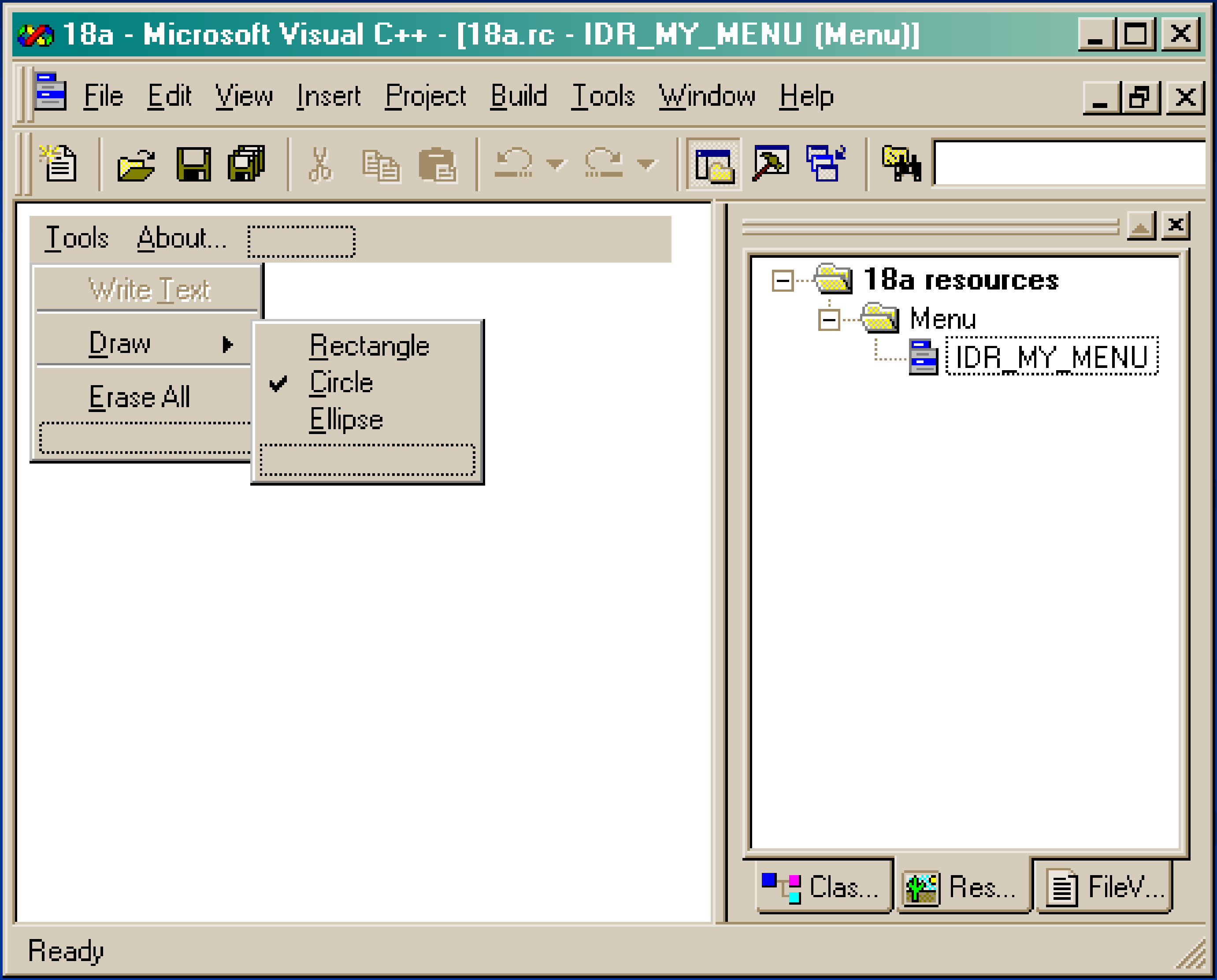
Start Menu



Menu Item

Popup Menu Item

Example Menu



MENU resource definition statement

T is *mnemonic*

```
IDR_MY_MENU MENU DISCARDABLE
BEGIN
    POPUP "&Tools"   BBar
    BEGIN
BBar     MENUITEM "Write &Text",     ID_TOOLS_WRITE_TEXT, GRAYED
BBar    MENUITEM SEPARATOR
BBar    POPUP "&Draw"
         BEGIN
BBar    MENUITEM "&Rectangle",  ID_TOOLS_DRAW_RECTANGLE
BBar    MENUITEM "&Circle",     ID_TOOLS_DRAW_CIRCLE, CHECKED
BBar    MENUITEM "&Ellipse",    ID_TOOLS_DRAW_ELLIPSE
         END
BBar    MENUITEM SEPARATOR
BBar    MENUITEM "&Erase All",      ID_TOOLS_ERASE_ALL, INACTIVE
    END
BBar MENUITEM "&About...",           ID_ABOUT
END
```

Resource definition

```
MENUITEM "&Rectangle",   ID_TOOLS_DRAW_RECTANGLE
```

Clicking on “Rectangle” menu item sends a message

WM_COMMAND

wParam: low word: `ID_TOOLS_DRAW_RECTANGLE` **(the menu ID)**

high word: 0

lParam: NULL

Loading a menu

```
HMENU LoadMenu(
   HINSTANCE hInstance,   // handle to module
   LPCTSTR lpMenuName     // menu name or resource id
);
```

- Can be an *integer ID* or a *string name* of a menu
- Call MAKEINTRESOURCE() macro to convert integer id to a LPTSTR pointer.

In Win32

all bits in high word of a 32-bit pointer are non-zero

But

all bits in high word of an integer < 65536 is 0

```
LPTSTR MAKEINTRESOURCE (WORD wInteger);
```

ASCII version of the macro

```
(LPSTR)((DWORD)((WORD)(i)))
```

Specifying default class menu for a window class

```
WNDCLASS wndClass;
… … …
… … …

wndClass. lpszMenuName = menu resource name or
                                   resource identifier
```

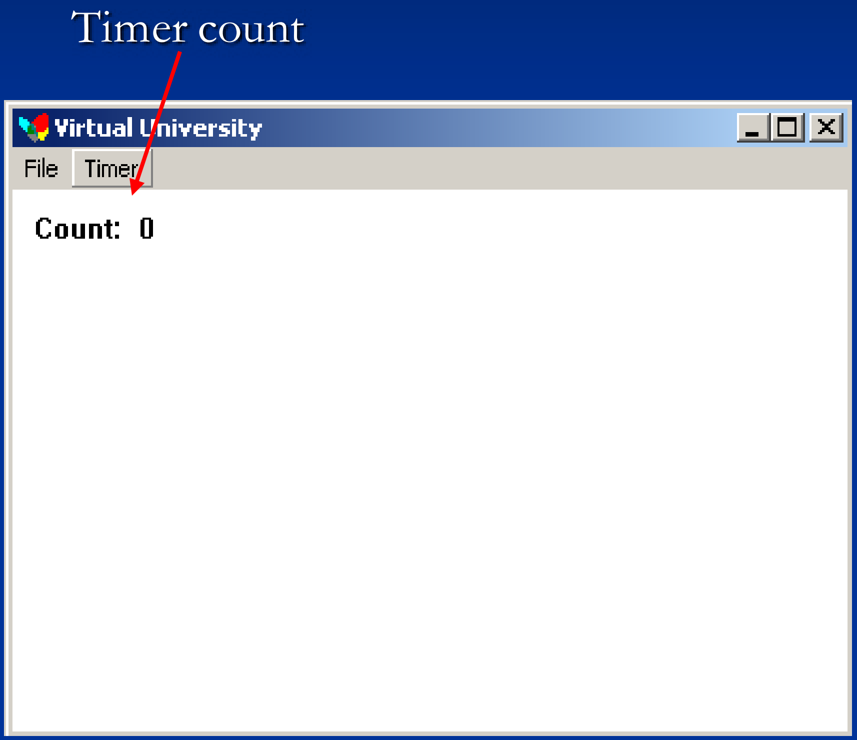
- Can be an *integer ID* or a *string name* of a menu
- Call MAKEINTRESOURCE() macro to convert integer id to a LPTSTR pointer.

Specifying a menu for a window

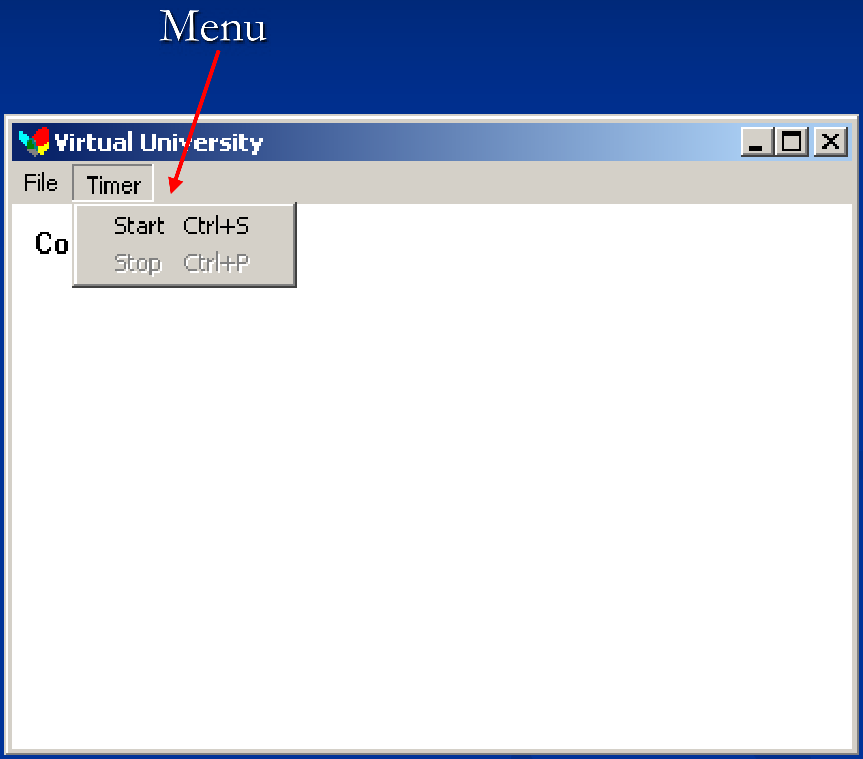
CreateWindow(…,…, hMenu, …,…);

handle to menu HMENU loaded by LoadMenu()

Menu specified in CreateWindow() overrides default class menu


Timer count
Virtual University
File
Timer

Count: 0


Menu
Virtual University
File
Timer
Start Ctrl+S
Stop Ctrl+P

Co

Resource definitions: String Table

```
#include “resource.h”

STRINGTABLE DISCARDABLE
BEGIN
    IDS_APP_NAME            "Virtual University"
    IDS_CLASS_NAME          "MyWindowClass"
END
```

Resource definitions: Application Icon

```
IDI_MAIN_ICON          ICON    DISCARDABLE     "VU.ICO"
```

Resource definitions: Application Menu

```
IDR_FIRST_MENU MENU DISCARDABLE
BEGIN
POPUP “&File"
    BEGIN
                MENUITEM "E&xit",     ID_FILE_EXIT
END

POPUP "&Timer"
    BEGIN
        MENUITEM "&Start",     ID_TIMER_START
        MENUITEM "Sto&p",      ID_TIMER_STOP, GRAYED
END
END
```

Example Application WNDCLASS

```
#define BUFFER_SIZE   128
TCHAR windowClassName[BUFFER_SIZE];


LoadString(hInstance, IDS_CLASS_NAME,
  windowClassName, BUFFER_SIZE);

wc.hIcon = LoadIcon(hInstance,
  MAKEINTRESOURCE(IDI_MAIN_ICON));

wc.lpszMenuName          =
  MAKEINTRESOURCE(IDR_FIRST_MENU);

wc.lpszClassName = windowClassName;
```

Example Application CreateWindow()

```
#define BUFFER_SIZE   128

TCHAR windowName[BUFFER_SIZE];

… … …

LoadString(hInstance, IDS_APP_NAME,
   windowName, BUFFER_SIZE);

hWnd = CreateWindow(windowClassName,
   windowName, ...
```

Window Procedure

```
static int count;

static BOOL bTimerStarted;

          .   .   .   .   .   .

case WM_CREATE:

          count=0;

          bTimerStarted=FALSE;

          .   .   .   .   .   .
```

Window Procedure: WM_COMMAND message

```
    case WM_COMMAND:
        switch( LOWORD(wParam) )
        {
  case ID_TIMER_START:
        SetTimer(hWnd, ID_TIMER, 1000, NULL);
    bTimerStarted=TRUE;
    hOurMenu = GetMenu(hWnd);
   EnableMenuItem(hOurMenu, ID_TIMER_START,
    MF_BYCOMMAND | MF_GRAYED);
   EnableMenuItem(hOurMenu, ID_TIMER_STOP,
    MF_BYCOMMAND | MF_ENABLED);
        DrawMenuBar(hWnd);
```

Embedded within the 0r’s sequence of slides

Getting a handle to the menu of a window

```
HMENU GetMenu(
  HWND hWnd  // handle to window
);
```

Redrawing the menu bar of a window

```
BOOL DrawMenuBar(
  HWND hWnd  // handle to window
);
```

Window Procedure: WM_COMMAND message

```
case ID_TIMER_STOP:
    KillTimer(hWnd, ID_TIMER);
    bTimerStarted = FALSE;
    hOurMenu = GetMenu(hWnd);

    EnableMenuItem(hOurMenu, ID_TIMER_STOP,
    MF_BYCOMMAND | MF_GRAYED);
    EnableMenuItem(hOurMenu, ID_TIMER_START,
    MF_BYCOMMAND | MF_ENABLED);
    DrawMenuBar(hWnd);

    break;
```

Window Procedure: WM_COMMAND message

```
case ID_FILE_EXIT:
        DestroyWindow(hWnd);
```

Window Procedure

```
case WM_TIMER:
        switch(wParam)
        {
        case ID_TIMER:
        ++count;
        count %= 10;
        GetClientRect(hWnd, &rect);
        InvalidateRect(hWnd, &rect, TRUE);
                break;
        }
        break;
```

Window Procedure

```
TCHAR msg[10];

      case WM_PAINT:
            hDC = BeginPaint(hWnd, &ps);
      wsprintf(msg, "Count: %2d", count);
      TextOut(hDC, 10, 10, msg, lstrlen(msg));
            EndPaint(hWnd, &ps);
            break;
```

Window Procedure

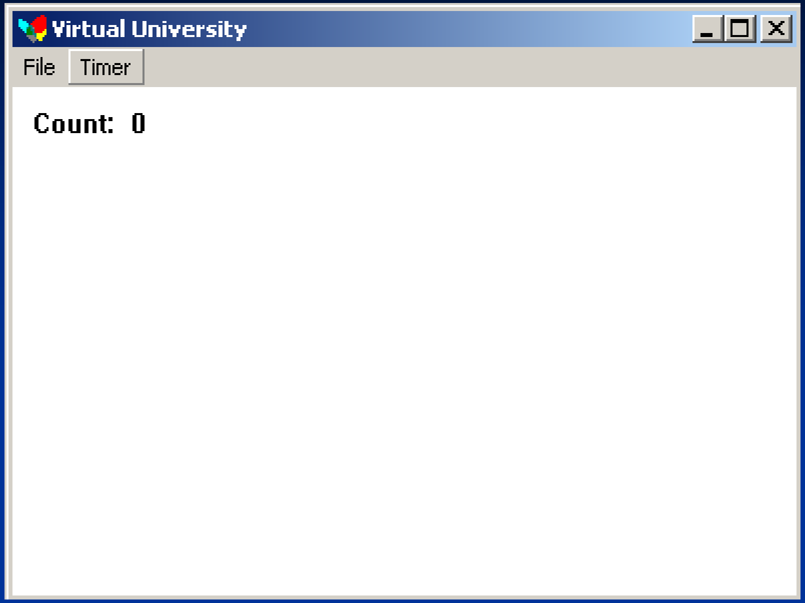
```
case WM_DESTROY:
    if(bTimerStarted)
        KillTimer(hWnd, ID_TIMER);
    PostQuitMessage(0);
    break;
```

Keyboard Accelerators

Keyboard Accelerators

```
IDR_ACCELERATOR ACCELERATORS DISCARDABLE
BEGIN
    "P",  ID_TIMER_STOP,     VIRTKEY, CONTROL, NOINVERT
    "S",  ID_TIMER_START,    VIRTKEY, CONTROL, NOINVERT
   "X",  ID_FILE_EXIT,      VIRTKEY, ALT, NOINVERT
END
```

NOINVERT has no use in Win32

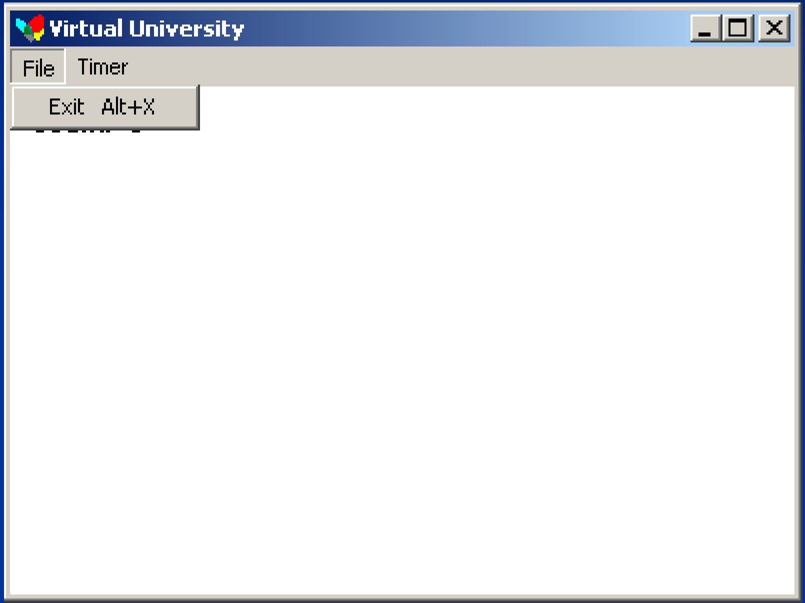


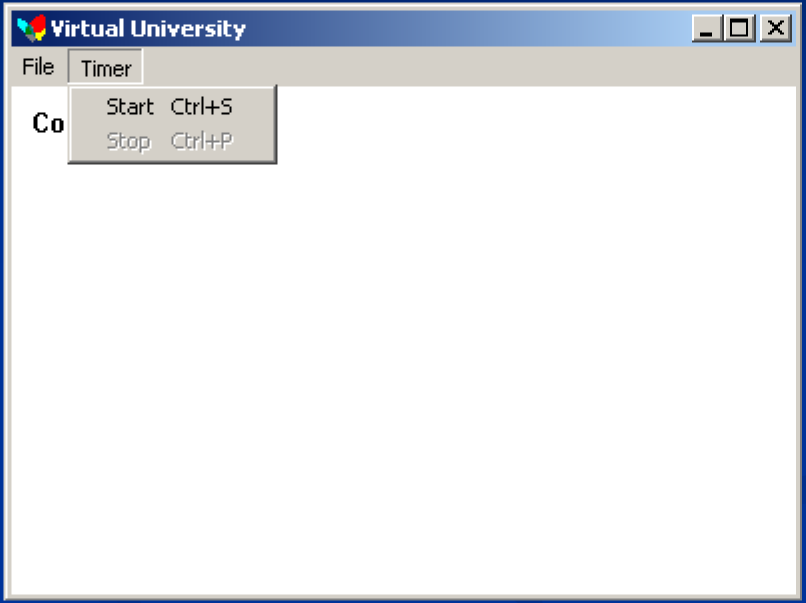
```
Count:  0
```

How do I know what accelerators to use to get rid of it!!!

Keyboard Accelerators Shortcuts

```
IDR_FIRST_MENU MENU DISCARDABLE
BEGIN
    POPUP "&File"
    BEGIN
        MENUITEM "E&xit\tAlt+X",      ID_FILE_EXIT
    END
    POPUP "&Timer"
    BEGIN
       MENUITEM "&Start\tCtrl+S",    ID_TIMER_START
       MENUITEM "Sto&p\tCtrl+P",     ID_TIMER_STOP, GRAYED
    END
END
```





I know ALT+X can save me from this application!!!

Message Loop

```
HACCEL hAccelerators;


hAccelerators = LoadAccelerators(hInstance,
   MAKEINTRESOURCE(IDR_ACCELERATOR));

while(GetMessage(&msg, NULL, 0, 0) > 0)

{

 if(!TranslateAccelerator(msg.hwnd, hAccelerators, &msg))

   {

         TranslateMessage(&msg);

         DispatchMessage(&msg);

   }

}
```

Do normal message dispatching only if it was not an accelerator Key stroke