13. Hola Mundo con WinAPI

Monday, November 16, 2020 5:10 PM

Haremos un Hola Mundo! Usando la WinAPI, aprovechando el header (programa escrito en C++) winuser.h que tiene la función MessageBoxA.

La función MessageBoxA despliega una caja de diálogo que contiene un ícono del sistema, un conjunto de botones, y un breve mensaje específico de aplicación, el cual tiene un estatus o información de error.

Esta cuadro de mensaje devuelve un valor entero que nos indica el botón que el usuario accionó.

Sintaxis

C++ Copy

```
int MessageBoxA(
  HWND   hWnd,
  LPCSTR lpText,
  LPCSTR lpCaption,
  UINT   uType
);
```

Parámetros

- `hWnd`
  Tipo: HWND - Un handle de ventana
  Manejador (handle) de la ventana propietaria del cuadro de mensaje que se va a crear. Si este parámetro es NULL, el cuadro de mensaje no tiene propietario.

- `lpText`
  Tipo: LPCTSTR - Cadena tipo string
  El mensaje que se va a mostrar. Si la cadena consta de más de una línea, puede separar las líneas mediante un retorno de carro y/o un carácter de salto de línea entre cada línea.

- `lpCaption`
  Tipo: LPCTSTR - Cadena tipo string
  Título del cuadro de diálogo. Si este parámetro es NULL, el título predeterminado es Error.

- `uType`
  Tipo: UINT - Entero sin signo que va desde 0 hasta 4294967295.
  El contenido y el comportamiento del cuadro de diálogo. Este parámetro puede ser una combinación de indicadores de los siguientes grupos de indicadores. Para saber que valor indicar, consulta la documentación.

NOTA: Un handle es un puntero inteligente que se usa cuando un programa hace referencia a bloques de memoria u objetos controlados por otros sistemas, tales como una base de datos o un sistema operativo.

Valor de retorno

Si un cuadro de mensaje tiene un botón Cancelar, la función devuelve el valor IDCANCEL, también si se pulsa la tecla ESC o se selecciona el botón Cancelar. Si el cuadro de mensaje no tiene ningún botón Cancelar, presionar ESC no tendrá ningún efecto, a menos que haya un botón de MB_OK. Si se muestra un botón de MB_OK y el usuario presiona ESC, el valor devuelto será IDOK.
Si se produce un error en la función, el valor devuelto es cero. Para obtener información de error extendida, llama a la función GetLastError.
Si la función se realiza correctamente, el valor devuelto es uno de los siguientes valores de elemento de menú.

Valor	Botón presionado

3	Abortar
2	Cancelar
11	Continuar
5	Ignorar
7	No
1	Ok
4	Reintentar
10	Intentar de nuevo (Try Again)
6	Si

HelloWorld.py

HelloWo...

Sección	Código
1	`import ctypes`
2	`dll_handle = ctypes.WinDLL("User32.dll")` `k_handle = ctypes.WinDLL("Kernel32.dll")`

3	`hWnd = None`
	`lpText = 'Hello World'`
	`lpCaption = 'Hello Students!'`
	`uType = 0x00000001`

4	`response = dll_handle.MessageBoxW(hWnd, lpText, lpCaption, uType)`

5	`error = k_handle.GetLastError()`
	`if error != 0:`
	`  print("Error Code: {0}".format(error))`
	`  exit(1)`

6	`if response == 1:`
	`  print("Clicked OK!")`
	`elif response == 2:`
	`  print("User Exited!")`

Explicación

1. Importamos ctypes
2. Creamos un handle para User32.dll y para Kernel32.dll
   a. El User32.dll nos permite invocar el cuadro de mensaje
   b. El Kernel32.dll nos permite monitorear los mensajes de error que se almacenan en el sistema.
3. Antes de invocar una función se recomienda declarar los parámetros que vamos a enviar. Una buena práctica cuando trabajamos con la WinAPI es usar los mismos nombres para los argumentos que vamos a enviar que el nombre que tienen como parámetros al ser pasados a la función o método correspondiente.
   a. hWnd - No definiremos propietario
   b. uType - El mensaje tendrá los botones: **Ok** y **Cancelar**.
4. Llamamos a la API de Windows para invocar la caja de texto. El resultado de esta acción se almacenará en la variable response.
5. Se revisa si ocurrió algún error. Si la variable error tiene un valor de 0, implica que no hubo error, en otro caso, nos indica el error ocurrido. Las claves de error las puedes consultar en System Error Codes.
6. Verificamos la respuesta obtenida. Por el valor de uType, los únicos valores posibles de salida con 1 y 2.