

Win32 API (Application Programming Interface).

Programación 2

MGTI Alejandro Villarreal Mojica

- Un control (objeto) envía un mensaje al sistema operativo cada vez que sucede algo nuevo.
- Porque un control puede enviar varios mensajes y porque muchos controles pueden enviar varios mensajes, hay una fórmula para cada mensaje o casi cada uno de ellos debe seguir ciertas condiciones.
- Así como hay reglas en una oficina postal que se deben seguir con el fin de enviar una carta.

- Un mensaje para Windows debe proporcionar cuatro elementos de información:
 - ¿Quién envió el mensaje?
 - ¿Cuál mensaje?
 - Primera parte del mensaje
 - 32 bits
 - Segunda parte del mensaje
 - 32 bits

- ¿Quién envió el mensaje?
 - Cada objeto necesitará en el programa, al igual que todo, un nombre.
 - El sistema operativo necesita este nombre para identificar cada objeto, para cualquier razón.
 - Un objeto en Windows se identifica como un “Handle”.
 - Para los controles de Windows, la variable debe ser del tipo HWND

- ¿Cuál Mensaje?
 - El objeto que envía un mensaje debe indicar al sistema operativo qué mensaje se está enviando.
 - Hay varios tipos de mensajes para diferentes circunstancias.
 - Sin embargo, para hacer las cosas un poco más sencillas, cada mensaje es un número entero constante positivo (unsigned int) identificado con un nombre particular.
 - Por lo tanto, el identificador de mensaje se pasa como UINT.

- El contenido del mensaje
 - Debido a que hay muchos tipos de mensajes, se debe proporcionar dos piezas adicionales de información para ayudar a procesar el mensaje.
 - Estos dos elementos dependen del tipo de mensaje y podrían ser cualquier cosa.
 - El primer elemento que acompaña es un tipo de 32 bits (unsigned int) llamada WPARAM (parámetro WORD; en otras palabras, es un WORD (unsigned int) argumento).
 - El segundo elemento que acompaña es un tipo de 32 bits de valor (long) llamado LPARAM (siglas de parámetro LONG; en otras palabras, es un argumento long).
 - Recuerde que estos dos pueden ser diferentes cosas para diferentes mensajes.

Ejemplo del uso de los parámetros:

```
LRESULT CALLBACK proc(HWND hWnd, UINT message, WPARAM  
wParam, LPARAM lParam)  
{  
    switch (message)  
    {  
        case WM_COMMAND:  
            if((HIWORD(wParam) == BN_CLICKED) &&  
(LOWORD(wParam) == MY_BUTTON))  
            {  
            }  
            break;  
    }  
}
```

SendMessage()

- La función **SendMessage** envía el mensaje.
- La función llama al CALLBACK procedure de la ventana especificada y no retorna hasta que el procedimiento de ventana ha procesado el mensaje.

SendMessage()

```
SendMessage(  
    HWND hwnd, // manipulador de la  
                // ventana de destino  
    UINT uMsg, // mensaje a enviar  
    WPARAM wParam, // primer parámetro  
                // del mensaje  
    LPARAM lParam // segundo parámetro  
                // del mensaje  
);
```

LRESULT CALLBACK procedure

- Para gestionar los mensajes enviados a Windows, que se comunican a través de una función llamada procedimiento de Windows.
- El nombre de la función no es importante, pero debe devolver un entero de 32 bits, long, o Win32 LONG.
- Por lo tanto, se declara como LRESULT (LONG Result).
- Debido a que este es un puntero a función , debe ser declarado y se define como CALLBACK .
- Los mensajes pueden ser transportados en una función definida de la siguiente manera :

```
LRESULT CALLBACK MessageProcedure ( HWND hWnd ,  
    UINT uMsg , WPARAM wParam , LPARAM lParam ) ;
```

SetWindowText

- Se utiliza para cambiar el título de una ventana, pero también sirve para cambiar el texto de un control estático (**Edit Control**).
- Se puede modificar el texto dentro de un control estático mediante la función **SetWindowText** o mediante el mensaje **WM_SETTEXT**.

SetWindowText

- static HWND var = GetDlgItem(hwnd, ID_EST);
- **SetWindowText**(var, "Nuevo texto");
- **SendDlgItemMessage**(hwnd, ID_EST, **WM_SETTEXT**, 0, (**LPARAM**)"Nuevo texto");
- **SendMessage**(GetDlgItem(hwnd, ID_EST), **WM_SETTEXT**, 0, (**LPARAM**)"Nuevo texto");

GetWindowText

- Copia el texto de la barra de título de la ventana especificada (variable tipo HWND), (si es que la tiene) en un buffer.
- Si la variable tipo HWND especificada es un control, se copia el texto del control.

GetWindowText

GetWindowText(HWND hWnd, LPTSTR lpString, int nMaxCount)

- HWND hWnd, // manipulador de ventana o control con texto
- LPTSTR lpString, // dirección de buffer para el texto
- int nMaxCount // número máximo de caracteres a copiar

GetWindowText

- static HWND var = GetDlgItem(hwnd, ID_EST);
- **GetWindowText**(var, varchar, size);
- **SendDlgItemMessage**(hwnd, ID_EST, WM_GETTEXT, size, (LPARAM)varchar);
- **SendMessage**(GetDlgItem(hwnd, ID_EST), WM_GETTEXT, size, (LPARAM)varchar);