

# **DLLs**

Antonio Espín Herranz

# Librerías dinámicas

- Librería dinámica (\*.DLL): En el caso de una librería de vinculación dinámica las funciones no se copian dentro del ejecutable.
- Desde nuestro programa C++, haremos referencia a la función que nos interese de la DLL.
- Para poder usar la DLL, tenemos que cargarla en memoria, y a través de un puntero a una función podremos ejecutar la función.
- La función previamente habrá sido exportada en la DLL.

# Creación

- Creamos la DLL. Será un fichero con extensión CPP.
- Dentro de la DLL tenemos que especificar que funciones vamos a exportar.
- Desde una aplicación solo se podrán utilizar las funciones que hayamos exportado.

# Creación

- Incluir el fichero windows.h
- En la cabecera de las funciones que queramos exportar pondremos:

```
#if BUILDING_DLL
    #define DLLIMPORT __declspec(dllexport)
#else
    #define DLLIMPORT __declspec(dllimport)
#endif
```

```
DLLIMPORT func1();
DLLIMPORT func2();
```

# Uso de librerías dinámicas

- Desde una aplicación C++ (de consola).
- Incluir el fichero: windows.h.
- Definimos un tipo nuevo que representará un puntero a la función que queremos cargar.
- Habrá que especificar el tipo devuelto y los parámetros de entrada. Según el prototipo de nuestra función.  
`typedef void (CDECL* cfunc1)(char *, int *);`
- Definir una variable de este tipo.  
`cfunc1 ConexionDLL1;`

# Uso de librerías dinámicas

- Cargar la DLL.

// A través de esta variable cargaremos la DLL.

HINSTANCE hLib;

// Cargamos la DLL.

hLib = LoadLibrary("ejemploDLL.dll");

// Se debe hacer una comprobación para ver si se ha cargado correctamente:

if (hLib != 0){ ... }

- Indicar cual es la función que vamos a ejecutar de esa DLL, lo almacenamos en el puntero que hemos definido antes.
- Llamar a la función a través del puntero.

# Uso de librerías dinámicas

- Cargamos la función que queremos ejecutar con la instrucción `GetProcAddress` indicando la instancia de la DLL y el nombre de la función que queremos ejecutar.

```
ConexionDLL1 = (void (__cdecl *) (char *, int *))  
GetProcAddress(hLib,"HelloWord");
```

- Ejecutamos la función a partir del puntero.  
`ConexionDLL1(s, numero);`

- Por último, descargar la librería.  
`FreeLibrary(hLib);`

# Ejemplo de DLL

- Esta DLL, exporta una función que recibe dos parámetros y presenta un mensaje por pantalla y muestra el contenido de los parámetros:

```
#ifndef _DLL_H_  
#define _DLL_H_
```

```
#if BUILDING_DLL  
#define DLLIMPORT __declspec(dllexport)  
#else  
#define DLLIMPORT __declspec(dllimport)  
#endif
```

```
DLLIMPORT void HelloWorld();  
#endif
```



# Ejemplo de uso de DLL

```
#include <stdio.h>
#include <windows.h>

void helloworldDLL();

// Definición del Tipo:
typedef void (CDECL* cfunc1)();

// Definición de la variable:
cfunc1 ConexionDLL1;

void main(int argc, char *argv[]){

    helloworldDLL();
}
```

```
void helloworldDLL(){
    HINSTANCE hLib;

    // Cargamos la DLL.
    hLib = LoadLibrary("dll.dll");

    if(hLib!=0){
        // Cargamos la función que queremos
        ejecutar.
        ConexionDLL1 = (void (__cdecl *) ())
        GetProcAddress(hLib,"HelloWorld");

        // Llamamos a la función con el puntero a
        función:
        ConexionDLL1();

    } else
        printf("\nNo se ha podido cargar DLL ...");
    // Liberamos la DLL.
    FreeLibrary(hLib);
}
```