

Laboratorio 1. INF 100

Ambiente de trabajo con librerías en C

Profesor: Msc. Ariel Vernaza G.

Requerimientos:

- Compilador de C (Visual C++, DevCpp, Mingw32)
- Librerías SDL
- Imagen de Mario Bros extension .bmp

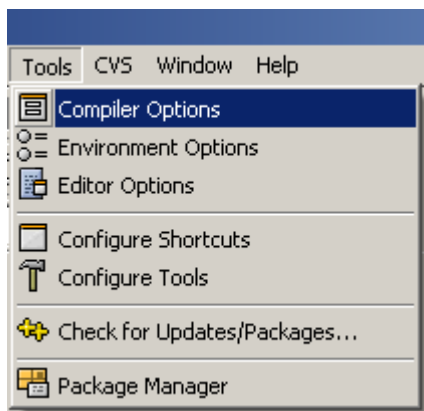
Parte 1. Configurar ambiente de trabajo

Lo primero que se debe tener es el compilador de C, y las librerías de SDL.

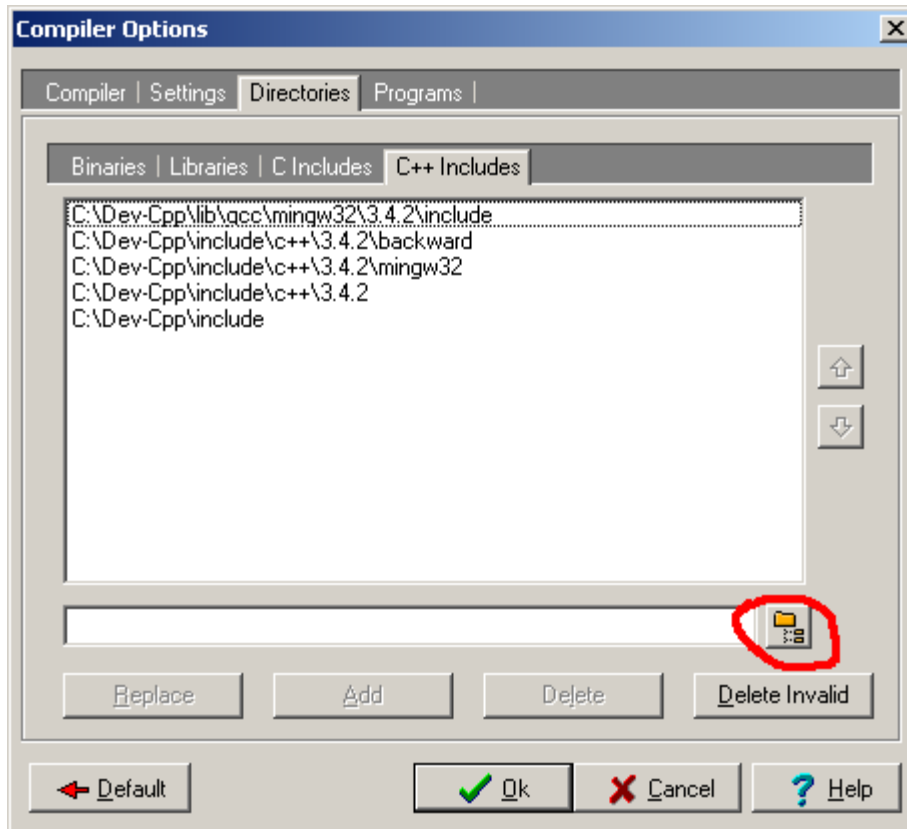
Las pueden descargar de la misma pagina sdl.org solo seleccionen la plataforma donde van a trabajar.

SDL (simple direct layer) lo soportan sin problemas los siguientes sistema operativo: linux windows, macosx android y steam.

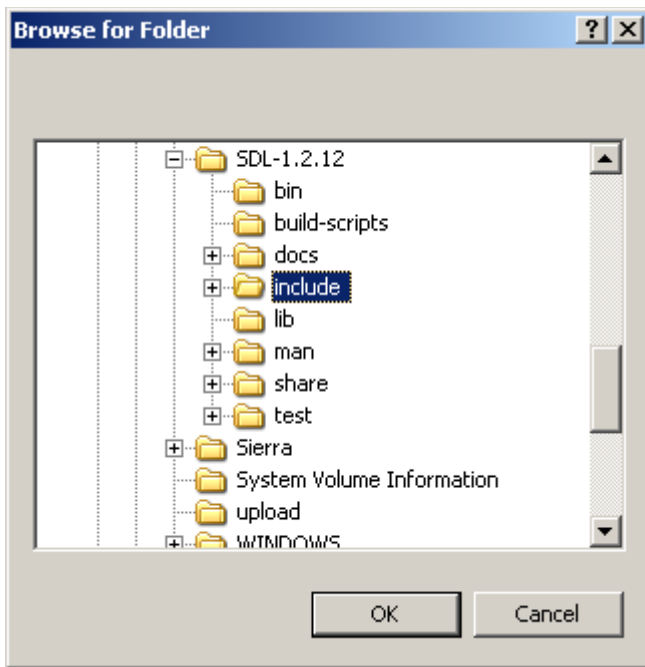
Para trabajar con SDL en C hay que configurar el compilador,. En herrameientas, seleccionar opciones de compilacion realizar los siguientes paso:

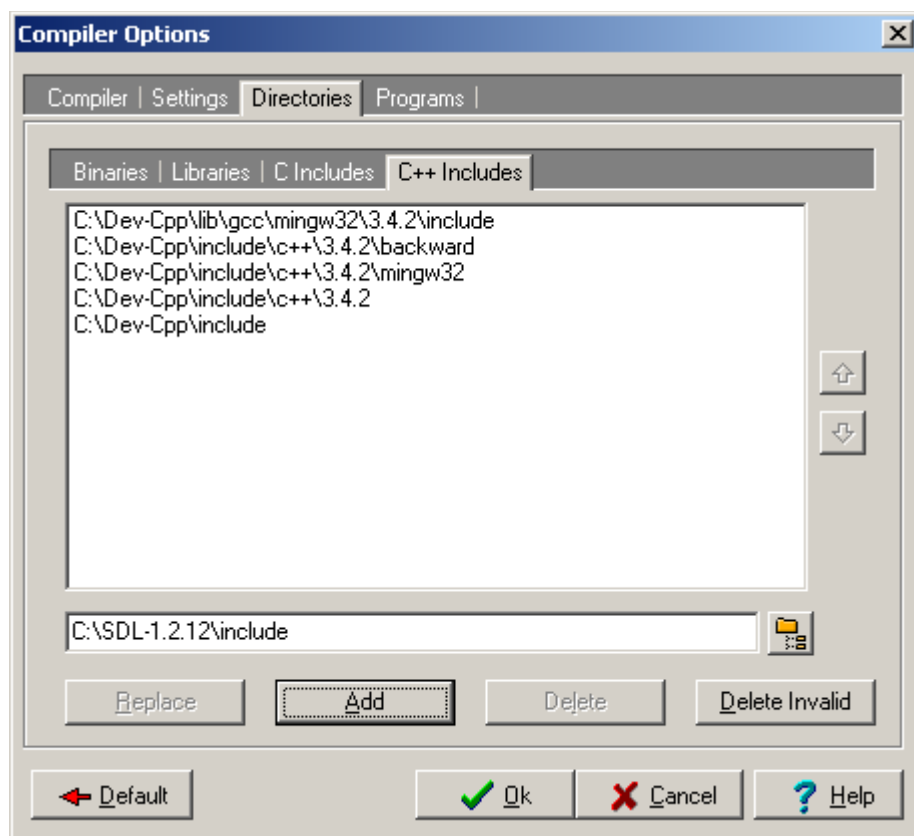


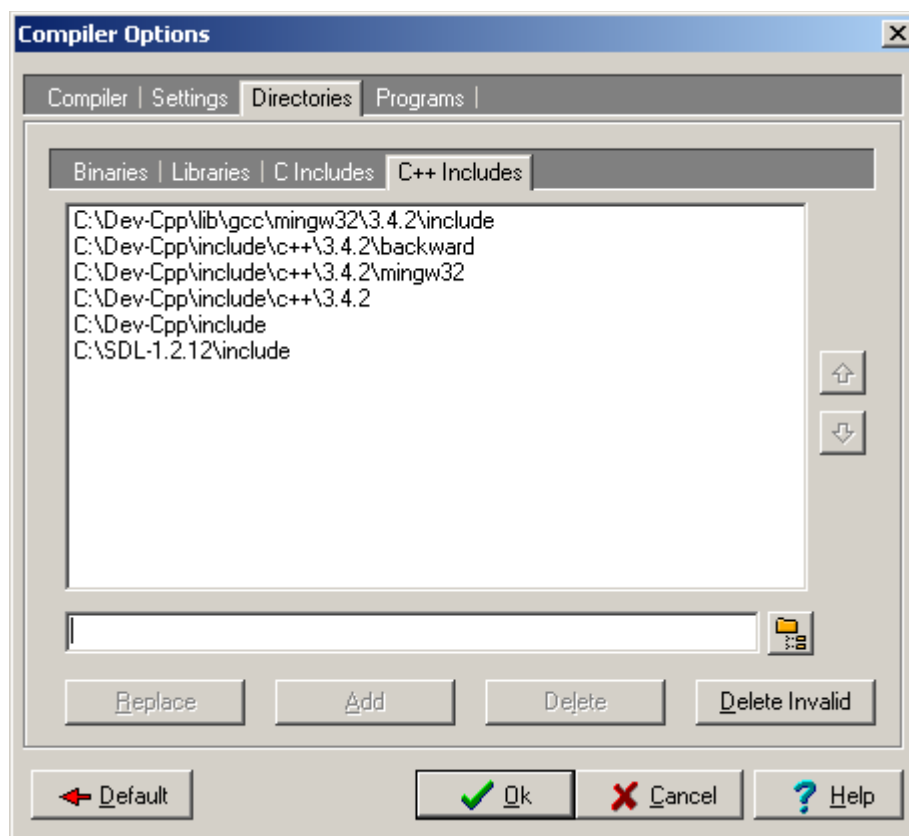
Ir al tab de directorios y luego dentro de el al tab de librerías de c++. En este caso, vamos importar librerías de C++, C es capaz de manejarlas sin problemas.



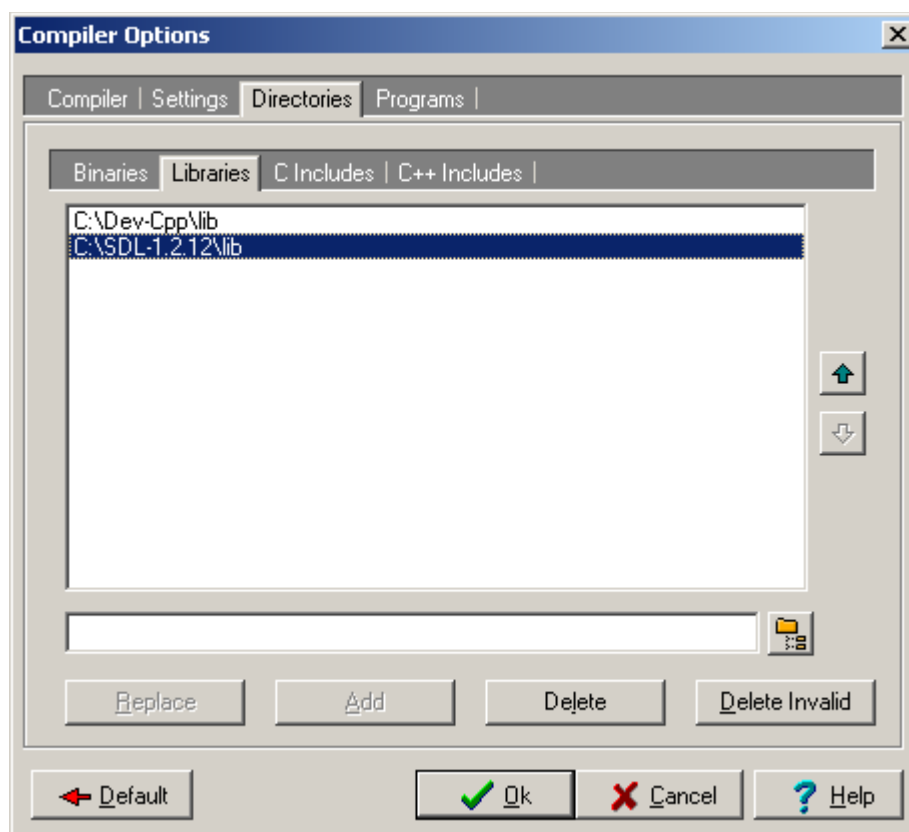
Debemos agregar la carpeta de include dentro de SDL.





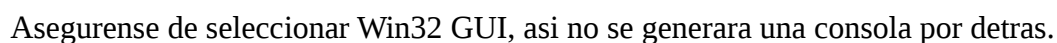
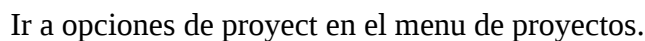


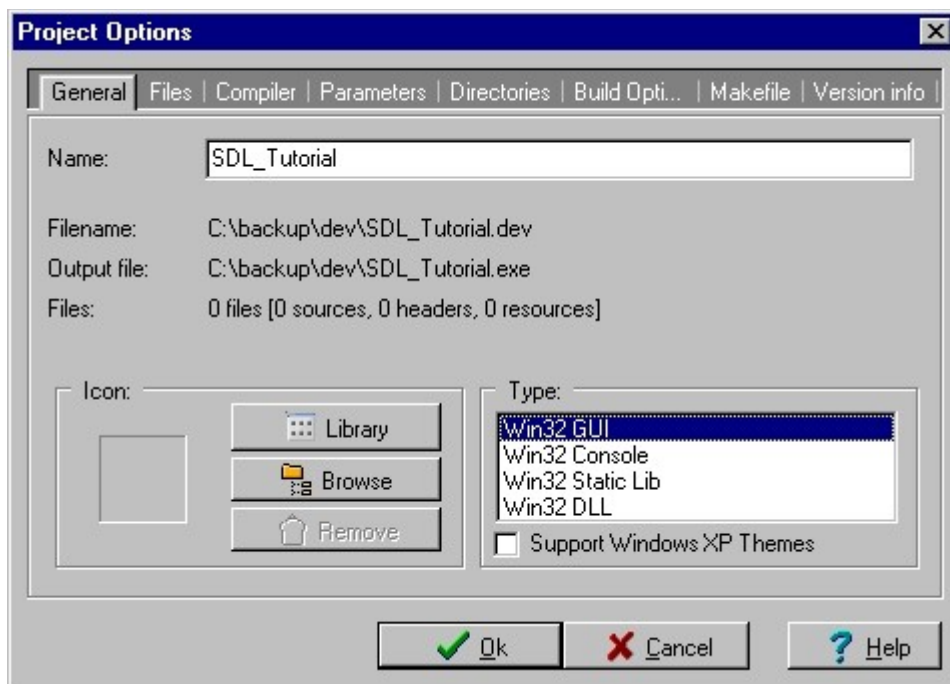
Luego en el tab de Librerías o Library, debemos agregar la carpeta de lib de SDL.



Una alternativa a eso es colocar el archivo SDL.dll en la carpeta C:\WINDOWS\SYSTEM32 si tu maquina es 32 bits, o colocarlo en la carpeta C:\Windows\SysWOW64 si tu maquina es 64 bits.

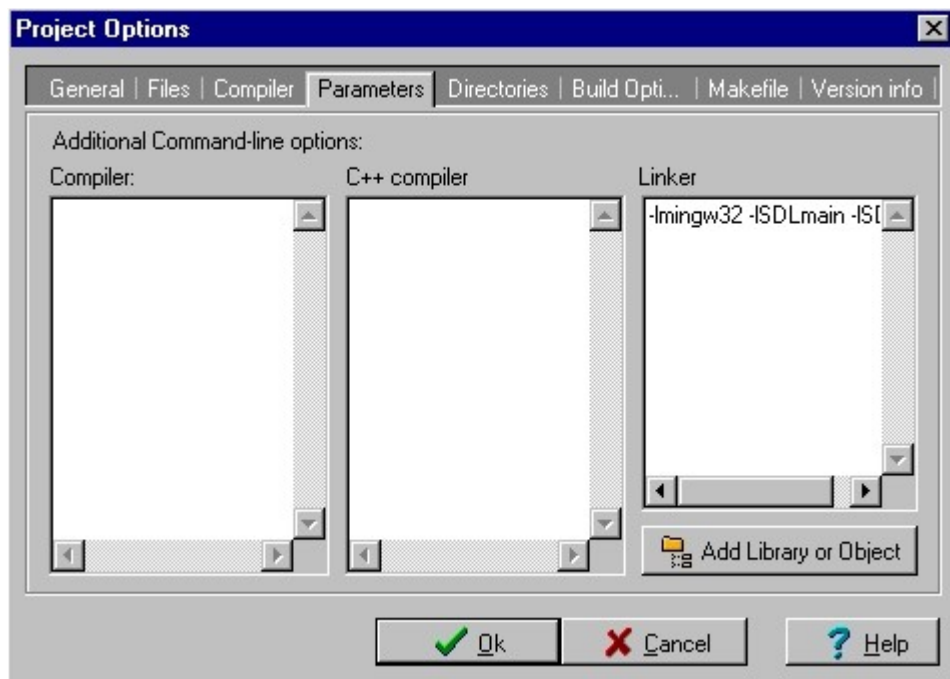
Para trabajar entonces crear un proyecto en blanco





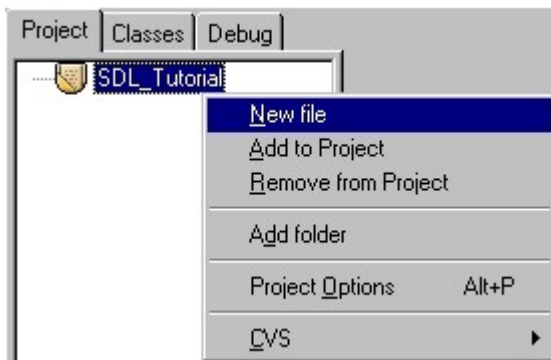
Ahora en el tab de parametros colocar este texto en el linker (si lo escriben mal no va a funcionar.)

-lmingw32 -lSDLmain -lSDL



Eso le dira a el compilador de C que debe usar estas librerias para compilar el programa.

Ahora deben crear un archivo de texto nuevo



Agreguen el siguiente codigo.

```
#include "SDL/SDL.h"

int main( int argc, char* args[] )
{
    //Start SDL SDL_Init( SDL_INIT_EVERYTHING );

    //Quit SDL

    SDL_Quit();

    return 0;
}
```

Al compilar el programa deberia funcionar correctamente.

Parte 2: Aparecer una imagen en la pantalla.

Para hacer esto deben utilizar la estructura

SDL_Surface

Se deben crear dos una para mario y otra para la pantalla.

La forma correcta de llamar esto es de esta forma, dentro del programa principal (**main**) crear dos punteros a sdl_surface

```
SDL_Surface* screen = NULL;
```

```
SDL_Surface* mario = NULL;
```

Una estructura en C, es un tipo de dato que permite agrupar diferentes tipos de datos en una misma variable.

El puntero es representado por *, es un tipo de variable que permite, acceder a la memoria para almacenar y leer información es muy útil al momento de trabajar con dispositivos, por decirlo es la herramienta más poderosa al trabajar en C. Cuando un puntero apunta a vacío no está apuntando a ninguna parte de la memoria.

Inicializamos la pantalla

```
//Start SDL
```

```
SDL_Init( SDL_INIT_EVERYTHING );
```

```
//Set up screen
```

```
screen = SDL_SetVideoMode( 640, 480, 32, SDL_SWSURFACE );
```

```
// Load mario
```

```
mario = SDL_LoadBMP( "mario.bmp" );
```

SDL_SetVideoMode, nos da acceso directo a la memoria de la tarjeta gráfica, y nos permite escribir sobre ella, en la superficie indicada de 640 x 480, con resolución de 32 bits. Cuando le decimos **SDL_SWSURFACE** le estamos diciendo que la memoria la vamos a gestionar por software.

El otro método, *SDL_LoadBMP*, carga una imagen BMP y nos devuelve una matriz que es realmente el mapa de bits de la imagen. Esto es una forma antigua de hacerlo, pero lo suficientemente directa para permitirnos manipular la imagen.

Finalmente colocamos a mario sobre la pantalla

```
//Apply image to screen
```

```
SDL_BlitSurface( mario, NULL, screen, NULL );
```

```
//Update Screen
```

```
SDL_Flip( screen );
```

```
//Pause
```

```
SDL_Delay( 2000 );
```

El método *SDL_BlitSurface*, escribe sobre la pantalla, la imagen que tenemos precargada.

Una vez tenemos hacemos *SDL_Flip*, que manda una orden a la sistema para que refresque la pantalla y así mario apareciera en la pantalla.

Para este caso de prueba, usamos *SDL_Delay* que dormirá la pantalla por 2 segundos, antes de continuar.

```
//Free the loaded image
```

```
SDL_FreeSurface( mario );
```

```
SDL_FreeSurface( screen );
```

```
//Quit
```

```
SDL SDL_Quit();
```

Finalmente liberamos la memoria de los punteros, sino lo hacemos podemos provocar stackoverflow, llenandola con información huérfana.

Al compilar este programa ya debe aparecer mario en la pantalla.