BREVE GUIDA ALL'UTILIZZO DI FILE WAV all'interno di Progetti DEV-CPP (utilizzando il linguaggio C oppure il linguaggio C++)

La funzione sndPlaySound

La funzione **sndPlaySound** esegue un suono in formato **WAV** specificato attraverso il nome del file relativo Questa funzione offre un sottoinsieme di funzionalità della funzione **PlaySound** ed è ancora a disposizione per ragioni di compatibilità all'indietro rispetto ai sistemi operativi meno recenti.

Sintassi

```
c++

BOOL sndPlaySound(
   LPCTSTR lpszSound,
   UINT    fuSound
);
```

Parametri

- ***lpszSound***

  Una stringa che specifica il suono da eseguire. Questo parametro conterrà il nome del file audio in formato WAV. Se questo parametro è uguale a **NULL** , qualsiasi suono in esecuzione verrà immediatamente fermato.

- ***fuSound***

  Insieme di **flag** che gestiscono la modalità di esecuzione del suono in formato WAV.

  Ecco un sottoinsieme di valori utilizzabili:

Valore	Significato
SND_ASYNC	Il suono viene eseguito in modo **ASINCRONO** e la funzione restituirà il controllo al programma chiamante immediatamente dopo averlo avviato (ossia senza aspettare la sua fine). Per terminare un suono eseguito in modalità ASINCRONA, occorre chiamare la funzione **sndPlaySound** con il parametro ***lpszSound*** posto al valore **NULL**.
SND_SYNC	Il suono viene eseguito in modalità **SINCRONA** e la funzione non restituirà il controllo al programma chiamante fino a quando il suono non terminerà la sua esecuzione
SND_LOOP	Il suono viene eseguito in continuazione fino a quando la funzione **sndPlaySound** non verrà chiamata nuovamente con il parametro ***lpszSound*** posto a **NULL**. E' obbligatorio specificare in questo caso anche il flag **SND_ASYNC** .
SND_NODEFAULT	Se non si riesce a trovare il file WAV contenente il suono, la funzione restituirà il controllo al programma chiamante senza eseguire il suono di default del sistema
SND_NOSTOP	Se nello stesso processo è in esecuzione un altro suono, la funzione restituirà immediatamente il controllo al programma chiamante restituendo il valore **FALSO** e senza eseguire il suono in essa specificato.

Valore di ritorno

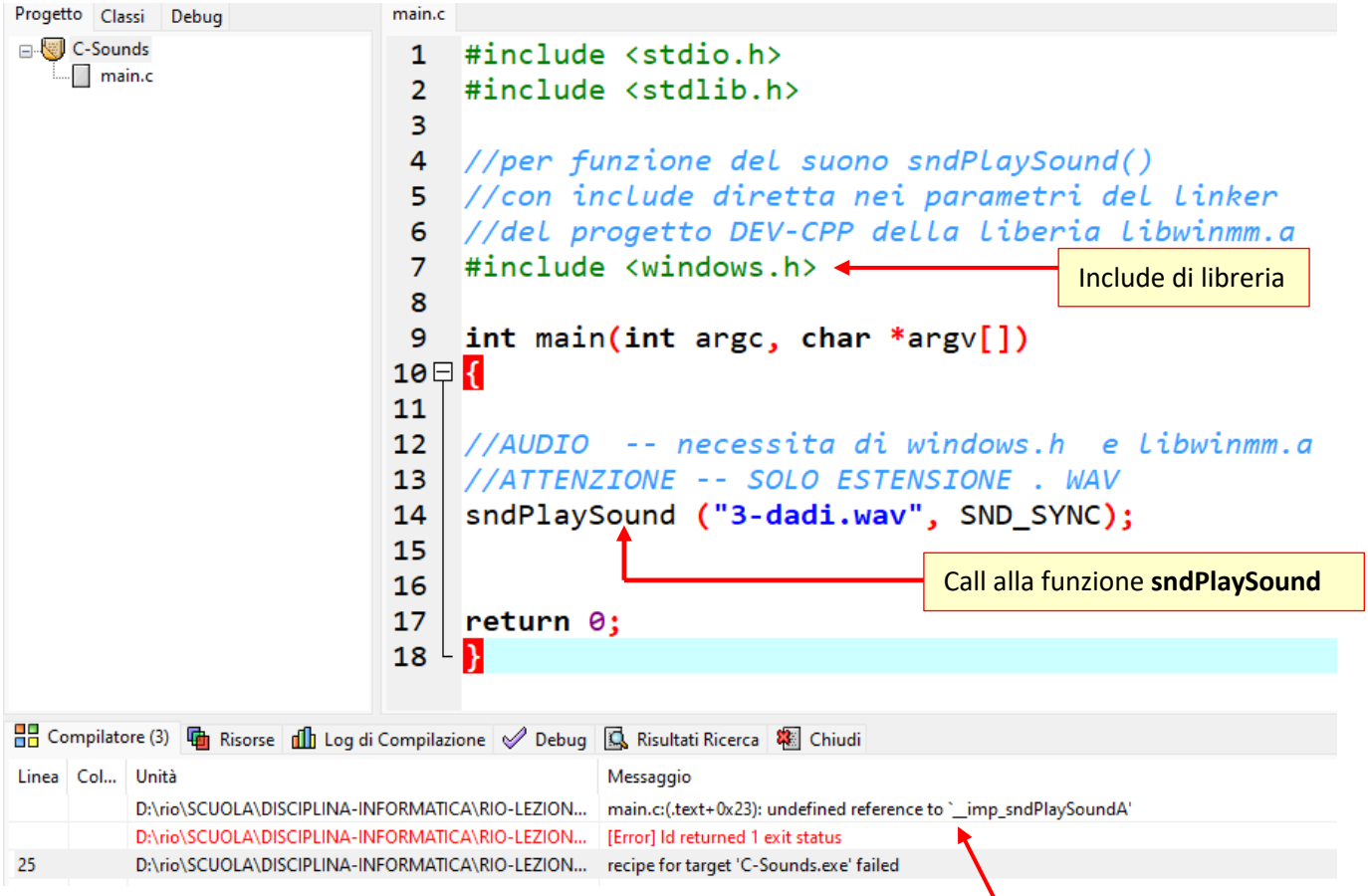
La funzione restituisce **TRUE** (ossia 1 per il linguaggio C) se ha successo oppure **FALSE** (ossia 0 per il linguaggio C) in caso di fallimento.

Requisiti minimi

Sistema client minimo supportato	Windows 2000 Professional [solo per app desktop]
Sistema server minimo supportato	Windows 2000 Server [solo per app desktop]
File header	Mmsystem.h (incluso in **Windows.h**)
Libreria	**Libwinmm.a**
Nomi unicode ed ANSI	**sndPlaySoundW** (Unicode) e **sndPlaySoundA** (ANSI)

COME UTILIZZARLA ALL'INTERNO DI UN PROGETTO DEV-CPP

In questo esempio abbiamo creato, scegliendo il linguaggio C, un progetto DEV-CPP chiamato **C-Sounds.dev** all'interno del quale proviamo ad utilizzare la funzione **sndPlaySound** per eseguire il file audio **3-dadi.wav** in modalità **ASINCRONA**



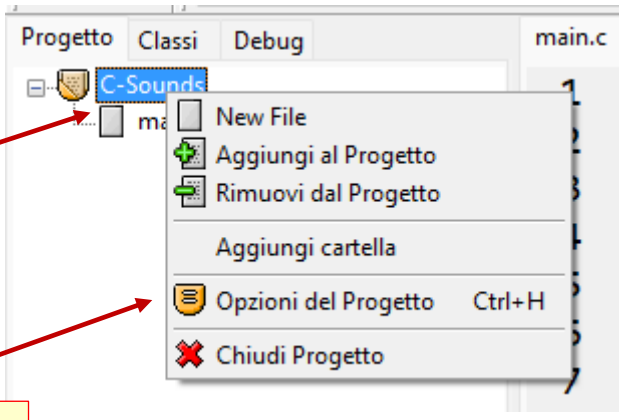
ATTENZIONE: non basta… infatti otterremo un errore in fase di LINKING

Occorre inserire "a mano" la libreria specifica da utilizzare

Come inserire la libreria libwinmm.a nei parametri di linking del progetto

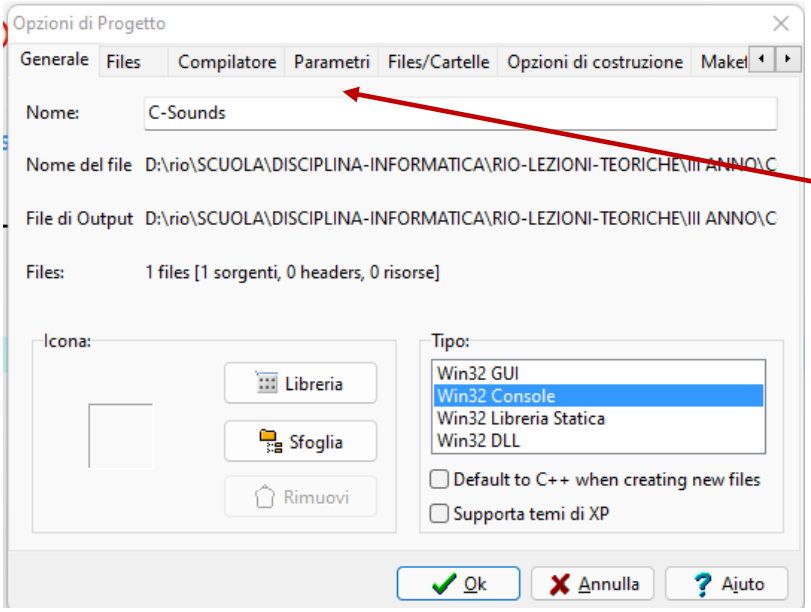
1

Tasto destro sul nome progetto



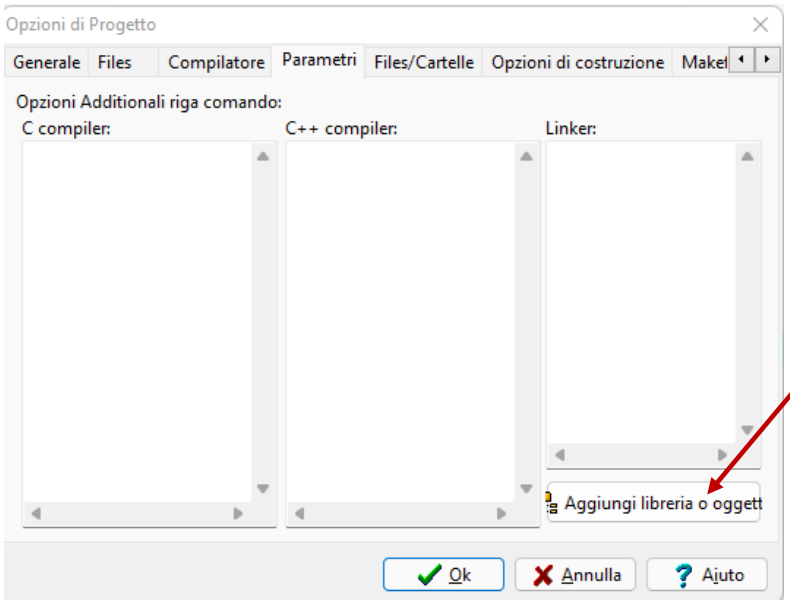
2

Poi **click** su "**Opzioni Progetto**"



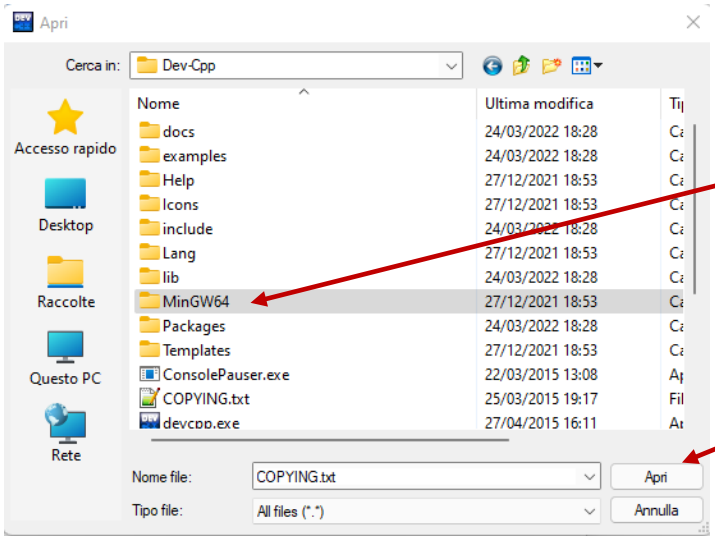
3

Click sul TAB "**Parametri**"



4

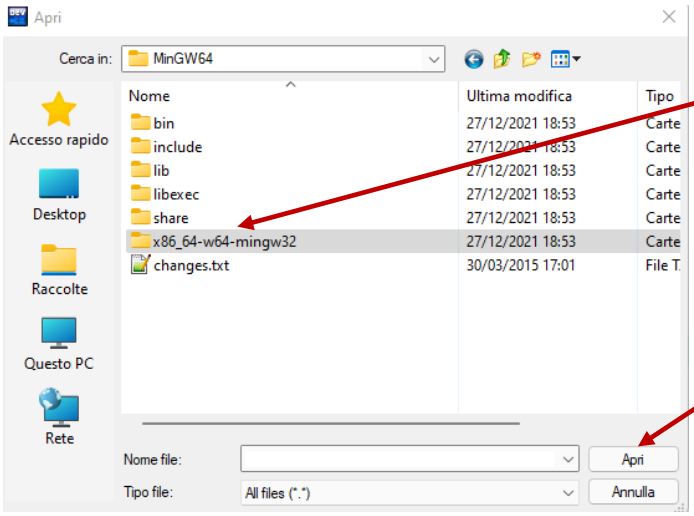
Click sul **"Aggiungi libreria o oggetti"**



5

Naviga fino alla sottodirectory **MingGW64** che si trova al di sotto della directory di istallazione **Dev-Cpp**

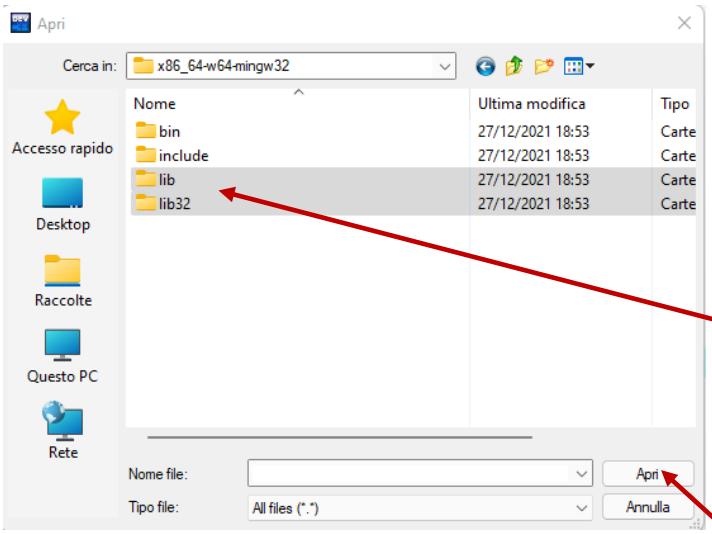
Selezioniamola e poi clicchiamo su "Apri"



6

Individua la sottodirectory **x86_64-w64-mingw32** che si trova al di sotto della directory **MingGW6**

Selezioniamola e poi clicchiamo su "Apri"

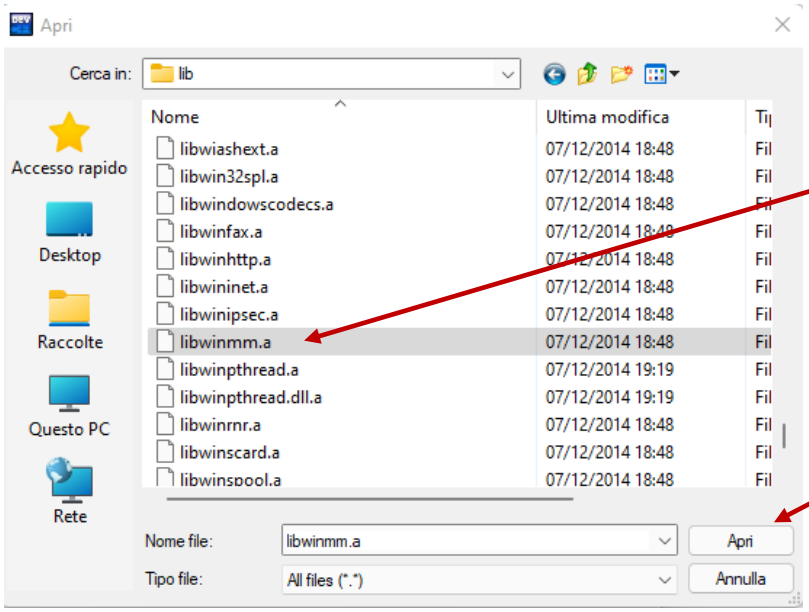


7

Individua le sottodirectory **lib e lib32** che si trovano al di sotto della directory **x86_64-w64-mingw32**
N.B. La libreria libwinmm.a che cerchiamo è presente in entrambe le directory ma a noi basta scegliere una delle due come origine

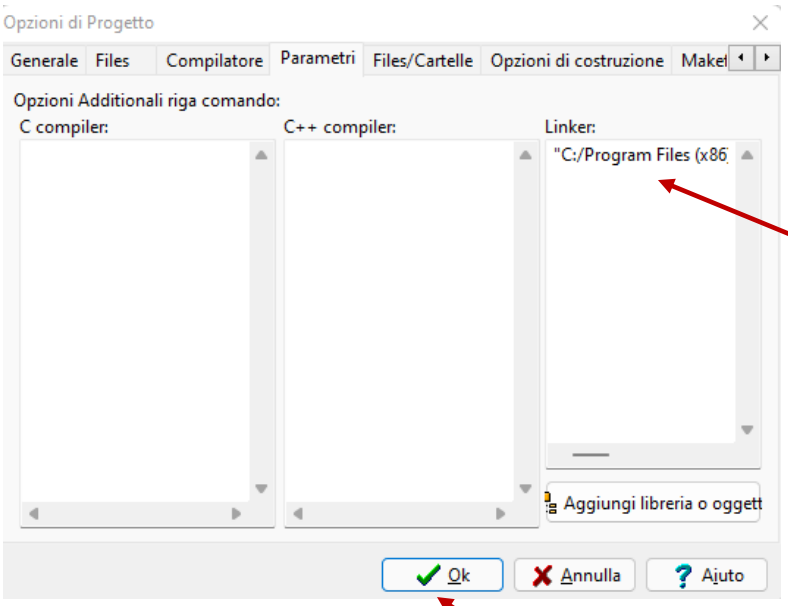
Selezioniamo una delle due e poi clicchiamo su **"Apri"**

8



Scorrere tutte le librerie presenti nella directory **lib** o **lib32** (in questo caso noi abbiamo scelto **la prima**) fino a raggiungere il file **libwinmm.a**

Selezioniamola e poi clicchiamo su "Apri"



9

Ora la nostra libreria **libwinmm.a** appare essere aggiunta alla sezione relativa al **"Linker"** del nostro progetto DEV-CPP

Non ci resta che validare il tutto cliccando su OK