Fondamenti di Informatica e Laboratorio T-AB e Fondamenti di Informatica T1 Ingegneria Elettronica e Telecomunicazioni e Ingegneria dell'Automazione a.a. 2011/2012

# Lab 01 Introduzione a Codelite

# Costruzione di un'Applicazione

Per costruire un'applicazione occorre:

- compilare il file (o *i* file se più d'uno) che contengono il testo del programma (file sorgente) Il risultato sono uno o più file oggetto.
- collegare i file oggetto l'uno con l'altro e con le librerie di sistema.

# Compilazione di un'Applicazione

- 1) Compilare il file (o *i* file se più d'uno) che contengono il testo del programma
  - File *sorgente*: estensione . c
  - File oggetto: estensione .oo .obj



f1.obj: Una versione tradotta che però <u>non è autonoma</u> (e, quindi, non è direttamente eseguibile).

# Collegamento (Linking) di un'Applicazione

- 2) Collegare il file (o *i* file) oggetto fra loro e con le librerie di sistema
  - estensione .oo .obj – File oggetto:
  - File eseguibile: estensione . exe o nessuna



Collegamento (Linking) di un'Applicazione

#### **LIBRERIE DI SISTEMA**:

insieme di componenti software che consentono di <u>interfacciarsi col sistema operativo</u>, <u>usare le risorse da esso gestite</u>, e realizzare alcune "istruzioni complesse" del linguaggio

# Ambienti Integrati

#### Oggi, gli <u>ambienti di lavoro integrati</u> automatizzano la procedura:

- compilano i file sorgente (se e quando necessario)
- invocano il linker per costruire l'eseguibile

#### <u>ma per farlo devono sapere:</u>

- *quali file sorgente* costituiscono l'applicazione
- *il nome dell'eseguibile* da produrre.

Progetti

# È da queste esigenze che nasce il concetto di *PROGETTO*

- un contenitore concettuale (e fisico)
- che elenca i file sorgente in cui l'applicazione è strutturata
- ed eventualmente altre informazioni utili.

Oggi, *tutti* gli ambienti di sviluppo integrati, *per qualunque linguaggio*, forniscono questo concetto e lo supportano con idonei strumenti.

#### Download an installer which includes codelite IDE + MinGW suite (GNU toolchain + WinAPI)

🛃 CodeLite - Revision: 5377							
File Edit View Search	Workspace Build Debug Plugins Perspective Settings Help						
i 📑 💼 🕐 i 🖉 i	New Workspace						
i 💽 😣 🔂 i 🍪	Close Workspace						
Workspace View	Reload Workspace Tromito il monu						
Workspace Explorer	Workspace Settings						
	Workspace Editor Preferences Workspace scegliere						
Program Files (x86)	Import MS Visual Studio solution						
▶ - 📄 7-Zip	New Project						
A Game of Throi	Add an existing project arkspace						
<ul> <li>Apple Software l</li> <li>CodeLite</li> </ul>	Open Active Project Settings Alt-F7						
⊳ 🦲 config	Retag Workspace ( Full )						
Debuggers	Retag Workspace ( Quick )						
<ul> <li>ipc</li> <li>iexers</li> <li>iexers<!--</th--><th>No workspaces found.</th></li></ul>	No workspaces found.						

ck L	inks:	
Lite W	New Workspace	
<u>e a N</u>	Workspace Name:	
_	TestWorkspace	
_	Workspace Path:	
	C:\Users\alessiobonfietti\Desktop\TestWks	
-	Create the workspace under a separate directory	
urkena	File Name:	Inserire il nome del
лкарс	C:\Users\alessiobonfietti\Desktop\TestWks\TestWorkspace.	
		Workspace ed il percorso
-		
	Create	Cancel
es four	u.	

Si consiglia di lavorare sempre in c:\temp

Codel ite - Revision: 533

#### Ogni workspace può contenere uno o più progetti

File I	Edit	View	Search	Wor	kspace )	Build	Debug	Plugins	Perspective	Setting	gs Help
					New W Open V Close V	'orkspac Vorkspa Vorkspa	е се се			7	🗼 i 🎻 i 🖻 🕞 i 🔍
Worksp	ace V	iew			Reload	Worksp	ace			- 1	
Wo	orkspa	ce E	plorer		Worksp Worksp Import	oace Sett oace Edit MS Visu	tings tor Prefer Jal Studio	ences solution f	ïle		
👩 Te	stWor	kspace		$\boldsymbol{<}$	New Pr	oject					
					Add an	existing	project				
					Open A	Active Pr	oject Sett	tings	Alt-	-F7	
					Retag \	Vorkspa	ce ( Full )			_	
					Retag \	Vorkspa	ce ( Quic	k)			

New project	
Categories:	Project name:
Console	Esempio01
Templates:	Project path:
Туре	C:\Users\alessiobonfietti\Desktop\TestWks Browse
<ul> <li>Simple executable (g++)</li> <li>Simple executable (Cobra)</li> <li>Simple executable (wxWidgets enabled)</li> <li>Simple executable (gcc)</li> </ul>	Compiler type: gnu gcc Create the project under a separate directory File name:
Simple executable (MSVC)	C:\Users\alessiobonfietti Inserire il nome del
Selezionare la	Note that this project is se gcc)
ategoria ' <b>Console</b> ' d il template ' <b>gcc</b> '	OK Cancel

File Edit View Search Workspace	Build Debug Plugins Perspective Settings Help C++
E 📄 🙆 💆 🗙 I	🔏 💼 💼 🦘 🛷 🧼 🌪 🚖 🕴 🖌 🗄 🗟 🖓 🗄 🗟 🖓 🐇 🖉 👘
i 💽 🙁 🗊 🎼 🏠 i 🕨	🔲 🗍 🥑 🔶 🔁 😵
Workspace View X	<global>  main(int argc, char **argy)</global>
Workspace Explorer Tabs Tab =	
🏳 🕀 🏤 👧 💀	main.c X
Debug 👻	
TestWorkspace	3 int main(int argc, char **argv)
a 📄 Esempio01	4 💌 {
⊿ 📄 src	<pre>5 printf("hello world\n"); 6 printf("hello world\n");</pre>
G main.c	7
	8
	Editor View
<b>TT</b> 7 <b>1 T</b> 7	
Workspace Vie	ew
Output View	
Build A Errors Q Search Q	Replace / 🔍 References / 📓 Output 🛛 / Trace / 🔂 Tasks / 🖾 BuildQ / 🖌 CppCheck / 🐒 CScope
Current working directory: C: Running program: le exec.exe	\Users\alessiobonfietti\Desktop\TestWks\Esempio01\Debug
Program exited with return co	de:
2	Output View
4	
71	
Send:	

File	Edit	View	Search	Worl	cspace	Build	Debu	ig P
		1	P	V	×	Ж		È
		J		(R)				C
Works	space \	/iew			×	:   <gl< td=""><td>obal&gt;</td><td></td></gl<>	obal>	
w	orksp	ace E	xplorer	Tabs	Tab ₹		nain.c	×
9	Ð 🖷		<b>S</b>				1	1 <b>±</b>
Debu	ig				-		2	
T	estWo	rkspace	2			11	3	i
	Ese	mpio01					4	▼ {
	4	src					5	
		C mai	n.c				6	
							7	}
						•		
Outpu	t View							
	🛛 Buil	ا 🔔 / ا	Errors	Q Sea	rch 🤉	Repla	ce 0	Refe
(	Curre Runni	nt wor ng pro	king d	irect le ex	ory: C	:\Usei ./Ese	s\ale	essic D1



Alla creazione di un progetto, l'IDE **Codelite** crea automaticamente il file principale contenente la funzione main del programma.

Da questa interfaccia è dedicata alla gestione dei file sorgente

Click destro sulla directoy 'src'									
er aggiungere un file sorgente									
Workspace View <global>         Workspace       Explorer       Tabş       Tab =         Imain.c       1       1       1         Debug       Imain.c       1       1         Imain.c       1       1       1<td>Image: Second Second</td></global>	Image: Second								
Output View           Build         Errors         Search         Replace         References         References           Current, working, directory:         C:\Users\alessichonfiett	OK Cancel								







:h Worksp	oace [	Build	Debug	Plugins	Perspectiv	/e Settings	Help	C++
	×		Configura	tion Mana	iger			
			Run			Ctrl-F5	]	Per Eseguire il programma
		:	Stop					
Tabe T	ah T		Build Proje	ect		F7	fargv	<i>v</i> )
			Compile C	Current File	2	Ctrl-F7		
<i>J</i>	_		Clean Proj	ect				
	_		Stop Build				**a	Build = Compile + Link
			Rebuild Pr	roject				
	_		Build and (	Run Projec	ct	Ctrl-F9	(n")	);
			Build Worl	kspace		Ctrl-Shift-B		
			Clean Wor	rkspace				
			Rebuild W	orkspace				
			Batch Buil	d				
			Next Build	Error		F4		
							_	



Una volta scritto, compilato e collegato il programma (ossia, costruito l'eseguibile)

occorre uno strumento che consenta di

- eseguire il programma <u>passo per passo</u>
- vedendo le variabili e la loro evoluzione
- e <u>seguendo le funzioni</u> via via chiamate.





- Sia Codelite sia altri ambienti di sviluppo incorporano un *debugger* con cui eseguire il programma,
- riga per riga
  - entrando anche dentro alle funzioni chiamate
  - oppure considerando le chiamate di funzione come una singola operazione
- oppure <u>inserendo breakpoints</u>



# Fase di Debugging

- Prima di iniziare la sessione di debugging e' possibile inserire i cosiddetti breakpoints
  - punti di interruzione nell'esecuzione del programma in cui il debugger fornisce una "fotografia" dello stato delle variabili
- Per inserire un breakpoint posizionare il cursore nel punto in cui si vuole fermare il debug e (alternative):
  - Utilizzare il comando da Menù
  - Premere F9
  - Singolo click a fianco del numero di riga



# Debugger: Come Procedere

- Nel menu Debug che compare quando il Debugger e' attivo ci sono alcune voci importanti:
  - <u>Execute</u>: esegue il programma fino al prossimo Debug
  - <u>Step in</u>: esegue passo passo le istruzioni di una funzione
  - Step Out: esegue l'istruzione e torna alla funziona chiamante
  - <u>Next</u>: esegue l'istruzione corrente
  - Show current line: permette di posizionare il cursore in una determinata posizione nel sorgente e esegue tutte le istruzioni fino ad arrestarsi al cursore.



·var3

int





New watch Expression to wa	tch:		Rappresenta non in stack	zione parti Watches
Debugger	OK	Cancel		×
I Cocals	\infty Watches 🔳 Ascii V	/iewer 🖯 📄 Call Stack 🦯 🥥 Bre	akpoints 🛛 式 Threads	💐 Memory 🛛 🐞 🕨
Expression	Value	Туре		New
······B[2]	12	int		
				Delete
				Delete All



# **Build: Warning**



#### **Build: Errors**

