

Sviluppo C++ multi-piattaforma con strumenti Open Source

Gli strumenti di programmazione liberi per tutti i sistemi operativi

Davide “CD-RON” Rondini

Montebelluna Linux User Group

25 ottobre 2008 – Castelfranco Veneto



Licenza d'utilizzo

Copyright © 2008, Davide "CD-RON" Rondini.

Questo documento viene rilasciato secondo i termini della licenza Creative Commons (<http://creativecommons.org>).

L'utente è libero di:

distribuire, comunicare al pubblico, rappresentare o esporre in pubblico la presente opera

alle seguenti condizioni:

Attribuzione Deve riconoscere la paternità dell'opera all'autore originario.

Non commerciale Non può utilizzare quest'opera per scopi commerciali.

No opere derivate Non può alterare, trasformare o sviluppare quest'opera.

In occasione di ogni atto di riutilizzazione o distribuzione, deve chiarire agli altri i termini della licenza di quest'opera.

Se ottiene il permesso dal titolare del diritto d'autore, è possibile rinunciare a ciascuna di queste condizioni. Le utilizzazioni libere e gli altri diritti non sono in nessun modo limitati da quanto sopra. Questo è un riassunto in lingua corrente dei concetti chiave della licenza completa (codice legale), reperibile sul sito Internet

<http://creativecommons.org/licenses/by-nc-nd/2.0/legalcode>



- 1 Introduzione
- 2 Il compilatore
 - Sistemi *NIX
 - Windows
- 3 gestione progetti
 - CMake
- 4 Librerie grafiche
- 5 Version Control System
 - Subversion
- 6 Installazione e distribuzione
 - CPack
 - NSIS
 - Pacchetti Linux (deb, rpm)
- 7 Bibliografia



Perché sviluppare multi-piattaforma

- Raggiungere un'utenza più ampia possibile
- Software di ampio utilizzo
- Unico sforzo di sviluppo
- Mettersi al riparo dalle variazioni di mercato
- Richieste diverse degli utenti
- Non appoggiare un monopolio



Perché C++?

- Prestazioni
- Base di librerie sterminata
- Accesso a basso livello
- Standardizzazione
(*ISO/IEC 14882:2003*)
- C'è almeno un compilatore per piattaforma
- Programmazione object-oriented

Regole generali

- Utilizzare più possibile gli standard C++: ANSI, STL
- Verificare che le API che si vogliono usare siano portabili (\simeq Open Source)
- Definire i sistemi operativi supportati in fase di progettazione
- Fare il debug in parallelo



Perché GCC

- Standard *de facto* in ambienti non-Windows
- Open Source
- Implementa correttamente gli standard
- Produce binari con buone prestazioni
- Affidabile
- Supporta decine di piattaforme hardware



Linguaggi

- C
- C++
- Objective-C
- Fortran
- Java
- Ada

Debugger

- `gdb`** Estremamente potente e flessibile, il compagno ideale di `gcc`.
- `valgrind`** Memory debugger, utile per scovare memory leak.

Profiler

- `gprof`** Per ottimizzare al massimo il proprio codice.

IDE

- `Eclipse`** IDE nato per Java, ma multi-linguaggio. Flessibile e potente, ma pesante.
- `kdevelop`** IDE di KDE, multi-linguaggio, orientato allo sviluppo KDE/Qt.
- `Anjuta`** IDE di Gnome, multi-linguaggio, orientato allo sviluppo Gnome/GTK.
- `vi/Emacs/joe`** Per i fan della riga di comando

The screenshot shows the KDevelop IDE interface. The main window displays the source code for `addexistingfilesdlg.cpp`. The code includes comments and logic for handling file lists. A tooltip is visible over the `append` method call, providing details about the `QValueListIterator` class.

```

// items added via button or drag 'n drop
KFileItemListIterator itemList ( m_importList );

// items already added to the importView
KFileItemListIterator importedList ( *importView->items() );

QListViewItem* child = m_titem->firstChild();

QStringList duplicateList;

while ( child )
{
    FileItem* curItem = static_cast<FileItem*> ( child );

    itemList.toFirst();

    for ( ; itemList.current(); ++itemList )
    {
        if ( (*itemList)->name() == curItem->name() )
        {
            duplicateList.append ( (*itemList)->name() );
        }
    }

    child = child->nextSibling();
}

importedList.toFirst();

for ( ; importedList.current(); ++importedList )
{
    itemList.toFirst();

    for ( ; itemList.current(); ++itemList )

```

Tooltip content:

```

Container: QValueList<QString>
Kind: Function
Access: public
File: /usr/lib64/qt3/include/qvaluelist.h
Line: 528 Column: 4

```

The IDE interface includes a menu bar (File, Edit, View, Project, Build, Debug, Scripts, Bookmarks, Tools, Settings, Window, Help), a toolbar, a File Selector on the left, a File List, a File Tree, and a right-hand pane with Documentation, Code Snippets, and Automake Manager. The status bar at the bottom shows "Line: 172 Col: 18 INS NORM *".

The screenshot displays the Eclipse IDE interface. At the top, the title bar reads "Java - SwingFrame.java - Eclipse SDK". The menu bar includes "File", "Edit", "Source", "Refactor", "Navigate", "Search", "Project", "Run", "Window", and "Help". Below the menu is a toolbar with various icons for file operations and development actions.

The main workspace is divided into several panes:

- Package Explorer:** Shows the project structure with "SwingApp" containing "org.eclipse.swingapp", "SwingFrame.java", and "JRE System Library [SunJava2-1.4.2]".
- SwingFrame.java:** The main editor showing a graphical window titled "Information" with two text input fields labeled "Name:" and "Description:". Below the window is the source code for the `getJTextArea()` method.
- Palette:** A sidebar on the right containing a "Swing Components" palette with various UI controls like `JButton`, `JCheckBox`, `JRadioButton`, `JToggleButton`, `JLabel`, `JTextField`, `JPasswordField`, `JTextArea`, `JSlider`, `JScrollBar`, `Swing Containers`, `Swing Menus`, and `AWT Controls`.
- Outline:** Located at the bottom left, it shows a tree view of the "Information" window's components, including `ContentPane`, `Label`, `JTextField`, `Label`, `ScrollPane`, and `JTextArea`.
- Properties:** A table at the bottom right showing the properties of the selected component.

```
private JTextArea getJTextArea () {
    if (jTextArea == null) {
        jTextArea = new JTextArea();
    }
    return jTextArea;
} // @jvcc:decl-index=0:visual-const:win=10,10"
```

Property	Value
background	Cob:white
bounds	0,0,334,75
columns	0
componentOrientation	UNKNOWN
editable	true
...	...

At the bottom of the IDE, the status bar shows "Writable", "Smart Insert", "125 : 1", and "In Sync".



anjuta - Anjuta

File Edit View Goto Project Build Tools Debug Subversion Settings Help

New Open Save Undo Redo

uri_normalize [453]

Breakpoints

Enb	ID	File	Line	Function
✓	2	gbf-am-project.c	3535	impl_add_source
✓	3	gbf-am-project.c	496	uri_normalize

Documents

gbf-am-project.c main.c anjuta-children.c

```

489     if (length > 0) {
490         p = normalized_uri + length - 1;
491         if (*p == '/')
492             *p = '\0';
493     }
494
495     return normalized_uri;
496 }
497
498 /**
499  * uri_get_chrooted_path:
500  * @root_uri: the root uri (or NULL)
501  * @uri: the uri which must be inside @root_uri for which
502  *       root-changed path is wanted
503  *
504  * E.g.: uri_get_chrooted_path ("file:///foo/bar", "file://
505  *
506  * Return value: a newly allocated chrooted path
507  */
508 static gchar *
509 uri_get_chrooted_path (const gchar *root_uri, const gchar
510 - {
511     gchar *root_path, *path;
512     gint root_length;
513     gchar *retval = NULL;
514
515     return retval;
516 }

```

Registers

Register	Value
eax	0x093d6b39
ecx	0x00000061
edx	0xfefeff60
ebx	0xb21428a0
esp	0xbfafae12d0
ebp	0xbfafae12f8
esi	0x093d6b18
edi	0x09242880
eip	0xb21372d5
eflags	0x00200212
cs	0x00000073
ss	0x0000007b
ds	0x0000007b
es	0x0000007b
fs	0x00000000
gs	0x00000033
st0	0x0000000000000000
st1	0x0000000000000000
st2	0x0000000000000000
st3	0x8000000000000000
st4	0x8000000000000000
st5	0x0000000000000000
st6	0x3ff8000000000000
st7	0x3ff8000000000000
fctrl	0x0000037f
fsbase	0x00000023

Disassembly

```

0xb21372d5 add $0x1c,%esp
0xb21372d8 mov %esi,%eax
0xb21372da pop %ebx
0xb21372db pop %esi
0xb21372dc pop %edi
0xb21372dd pop %ebp
0xb21372de ret
0xb21372df mov 0xfffffe8(%ebp),%eax
0xb21372e2 test %eax,%eax
0xb21372e4 je 0xb2137364 <uri_normalize+0x110>
0xb21372e6 mov 0xfffffe8(%ebp),%eax
0xb21372e9 call 0xb2133fe0 <uri_normalize+0x100>
0xb21372ee mov %eax,%edi
0xb21372f0 movl $0x0,0x8(%esp)
0xb21372f8 mov 0xfffffec(%ebp),%eax
0xb21372fb mov %edi,(%esp)
0xb21372fe mov %eax,0x4(%esp)
0xb2137302 call 0xb2131da4 <g_free+0x10>
0xb2137307 mov %edi,(%esp)
0xb213730a mov %eax,%esi
0xb213730c call 0xb2132af4 <g_free+0x10>
0xb2137311 mov 0xfffffec(%ebp),%eax
0xb2137314 mov %eax,(%esp)
0xb2137317 call 0xb2132af4 <g_free+0x10>
0xb213731c mov %esi,0xfffffec(%ebp)
0xb213731f jmp 0xb21372a4 <uri_normalize+0x100>
0xb2137321 movb $0x0,0xfffffff(%ebp)
0xb2137325 add $0x1c,%esp

```

Locals

Variable	Value	Type
uri	<value optimized out>	const gchar *
base_uri	0x0	const gchar *
absolute_path	0x93c6658 "\360~\te/naba/projects/anjuta"	gchar *
normalized_uri	0x93d6b18 "file:///home/naba/projects/anjuta"	gchar *

Tasks Documents Inheritance Graph Memory

Col: 000 Line: 0497 Project: anjuta Mode: INS Zoom: 0 Debugger: Stopped



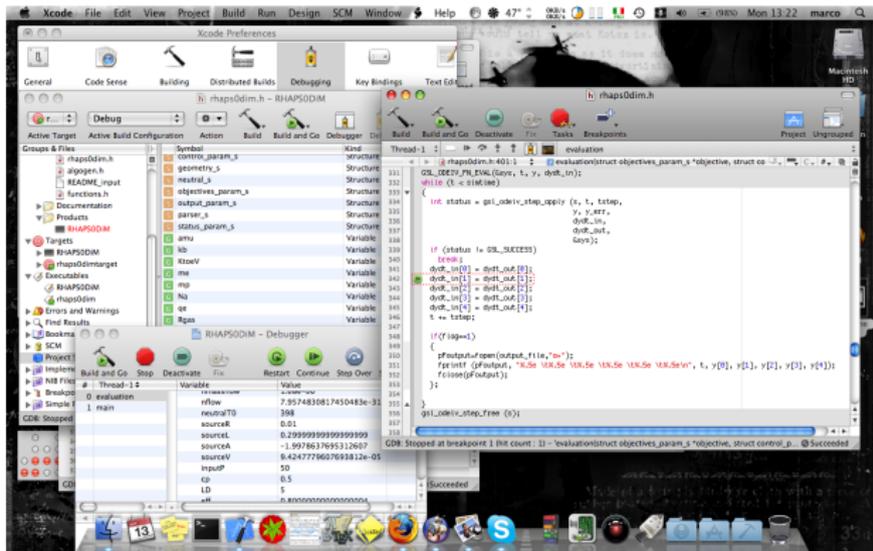
Apple MacOS

Tools

Apple utilizza gcc come compilatore, e quindi molti dei tool collegati, come gdb.

Tools

MacOS fornisce un IDE preinstallato, **XCode**.



Compilatori Open Source per Windows

MinGW

In pratica un port di gcc per Windows, ma più limitato. Spesso serve installare librerie GNU per farlo funzionare.

Dev-C++

Fondamentalmente un IDE per MinGW, basato su strumenti GNU (gdb, Cygwin).

OpenWatcom

Compilatore prima commerciale, poi reso libero. Nato nel 1965 come compilatore FORTRAN, molto diffuso negli anni '80-'90, oggi molto meno.



Microsoft Visual C++ Express

- Ambiente di sviluppo completo Microsoft
- Comprende compilatore, debugger, IDE, API (versione 2008)
- Dalla versione 2005 le versioni Express sono *gratuite* (non Open Source)

Perché usarlo

- Soluzione nativa Microsoft
- Ottima integrazione
- Debugger e auto-completamento di alto livello
- IDE evoluto e multi-linguaggio



Visual C++ Express

The screenshot displays the Visual C++ Express 2008 IDE with the following components:

- Menu Bar:** File, Edit, View, Project, Build, Debug, Tools, Window, Help.
- Toolbox:** Standard Windows development tools.
- Process:** [5208] samx.exe, Thread: [5084] Main Thread, Stack Frame: samx.exe:samsWindow::OnC...
- Solution Explorer:** Shows the project structure for 'samx_logger' (3 projects), including 'datastream', 'Source Files', and 'samx'.
- Code Editor:** Displays the source code for 'sams_main.cpp'. Key sections include:
 - Initialization and error handling for timer and channel matching.
 - A conditional compilation block for 'USE_DATA_LOG' that sets up a log file and prints acquisition status.
 - Configuration of a 'Mx FIFO' and channel setup.
 - Acquisition start logic with error handling for channel setup information.
- Watch Window:** Lists variables such as 'aCHID' (array of integers), 'aScaleMin' (float), and 'aScaleMax' (float).
- Output Window:** Shows the debug output, including the loading of various system DLLs like 'kernel32.dll', 'user32.dll', 'GDI32.dll', etc.



Progetti nativi

*NIX

Makefile

File di testo interpretati dal programma `make`

Windows

vcproj/dsp/sln

File XML generati e gestiti da Visual Studio.

nmake

Analogo a `make` per Windows.

Mac

xcodeproj/nib

Xcode gestisce i propri file di progetto...

Makefile

...ma essendo *NIX è compatibile con `make`.

È possibile utilizzare `Makefile` su tutte le piattaforme, con qualche accortezza.



CMake è un sistema di generazione di file di progetto per diversi compilatori e piattaforme.

Caratteristiche

- Genera i file di progetto *nativi* per altri tool
- Usa una sintassi unificata
- Open Source
- Supporta moltissimi compilatori
- Sviluppato da Kitware Inc.
- Usato da KDE 4.x
- GUI o CLI

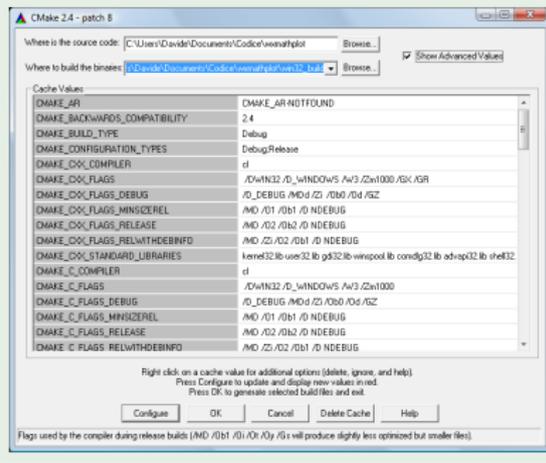


Come funziona CMake

Come è strutturato

- Gli script di progetto si chiamano `CMakeLists.txt`
- Ogni sotto-cartella ha un suo `CMakeLists.txt`
- CMake separa i file di compilazione dai sorgenti
- Le librerie più comuni vengono caricate facilmente, tramite script già forniti con CMake

CMake sotto Windows



Cartella principale

```
# Nome progetto "HELLO". 1
# Cartella dei sorgenti: ${HELLO_SOURCE_DIR} 2
# Cartella dei file di compilazione: ${HELLO_BINARY_DIR}. 3
cmake_minimum_required (VERSION 2.6) 4
project (HELLO) 5
6
# Indica le sottocartelle da compilare 7
# Ciascuna deve contenere il proprio CMakeLists.txt 8
add_subdirectory (library) 9
add_subdirectory (main) 10
```



Cartella `library`

```
# Crea una libreria chiamandola Hello 1
# Include il file hello.cpp 2
# Il nome del file output e' automatico 3
add_library (Hello hello.cpp) 4
```

Cartella `main`

```
# Include gli header della libreria Hello 1
include_directories (${HELLO_SOURCE_DIR}/Hello) 2
# Aggiunge il percorso di link della libreria hello 3
link_directories (${HELLO_BINARY_DIR}/Hello) 4
# Crea l'eseguibile 5
add_executable (helloMain main.cpp) 6
# Fa il linking di hello nel main 7
target_link_libraries (helloMain Hello) 8
```

Cos'è e a cosa serve un toolkit grafico

- Oggigiorno *quasi* tutte le applicazioni hanno una GUI
- Per il multi-piattaforma la scelta del toolkit grafico è critica.
- È essenziale minimizzare lo sforzo di porting con una libreria comune
- Per l'utente è importante un aspetto grafico coerente tra le diverse applicazioni

Open Source

- wxWidgets
- GTK+

Dual licensing

- Qt





wxWidgets

- wxWindows License
- Ha decine di bindings e di port
- Compila più o meno ovunque esista un compilatore ANSI C++.
- Usa le API native della piattaforma

Pro

- Open Source
- Documentazione
- Facilità di programmazione
- Piattaforme e compilatori supportati

Contro

- Multi-threading poco evoluto
- Poco note
- Editor form poco evoluti o commerciali



Qt

- GPLv2 per Open Source, Proprietaria per sviluppo commerciale
- Alla base di KDE
- Sviluppato da Trolltech, oggi Nokia
- Framework completo (supporto del sonoro)

Pro

- Integrazione con KDE
- Molto evoluto e completo
- Innovative
- Qt designer
- Possibile supporto commerciale

Contro

- Licenze commerciali costose
- Supporto Visual Studio solo commerciale



GTK+/gtkmm

- LGPL 2.1
- Alla base di GNOME
- Derivate dal progetto GIMP
- Fanno parte del progetto GNU
- API C con wrapper C++ (e altri)

Pro

- Open Source
- Ottimo supporto Linux
- Integrazione con GNOME
- Essenziale e semplice
- Supporto per l'accessibilità

Contro

- Solo toolkit grafico
- Compilazione su Windows non immediata
- Compilabili su Windows solo con MinGW

Il controllo di versione

Software per gestire le modifiche al codice di un progetto. Un VCS consente di:

- Tenere traccia delle modifiche
- Poter tornare indietro e recuperare vecchie versioni
- Gestire più programmatori che lavorano in parallelo sul codice
- Ramificazioni e sviluppo parallelo di più versioni

VCS e multi-piattaforma

Nello sviluppo multi-piattaforma il controllo di versione diventa essenziale.

Permette di mantenere una versione di riferimento unificata per tutte le piattaforme.

Software Open Source

- CVS
- Subversion
- Bazaar
- Mercurial
- Git



Come funziona il VCS 1/2

Il repository

- I file vengono gestiti da un server centrale, chiamato repository.
- Ciascun utente può creare una o più copie di lavoro



Il commit

- Una volta completate le modifiche, l'utente le invia al server
- Il software confronta i file e produce una nuova revisione.



Come funziona il VCS 2/2

L'update

- l'utente confronta i propri file col repository
- Normalmente i file vengono fusi (merge)



La risoluzione di conflitti

- Se ci sono conflitti, li deve risolvere l'utente, poi eseguire il commit



Subversion (SVN)

Caratteristiche

- Gestisce le versioni anche delle cartelle
- Forza a immettere un commento
- Backup e ripristino facile del repository
- Gestione efficiente dei file binari
- Il *commit* è una operazione atomica
- Comandi avanzati di analisi dati

Protocolli supportati

- File (cartella locale)
- WebDAV (http/https)
- ssh

kdesvn

The screenshot shows the kdesvn application window. The top part displays a file tree with columns for 'Nome', 'Stato', 'Revisione dell'ultima modifica', and 'Ultimo autore'. The bottom part shows a log of transfer statistics.

Nome	Stato	Revisione dell'ultima modifica	Ultimo autore
mathplot		47	cdron77
CMakeFiles	Senza revisione		
html	Senza revisione		
lib		22	jblanco
samples		46	cdron77
www		12	mrhill
CMakeCache.txt	Senza revisione		
CMakeLists.txt		42	cdron77
Changelog		42	cdron77

148 Byte di 148 Byte trasferiti.
472 Byte di 472 Byte trasferiti.
300 Byte di 300 Byte trasferiti.
724 Byte di 724 Byte trasferiti.
357 Byte di 357 Byte trasferiti.
1.24 kByte trasferito.
1.57 kByte trasferito.
Controllo degli aggiornamenti completato
Pronto

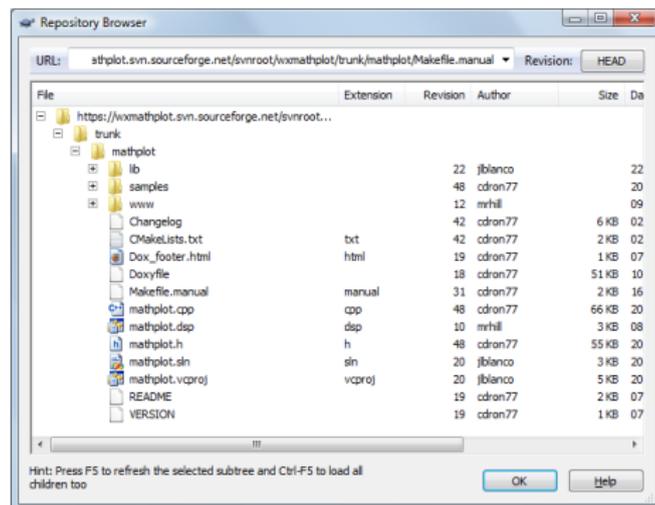
Subversion per Windows: TortoiseSVN



TortoiseSVN

TortoiseSVN

- Client SVN per Windows
- Si integra in Esplora Risorse
- Interamente gestito da menù contestuale



Distribuire il proprio programma

l'ultimo passo

- Non sempre è sufficiente distribuire i sorgenti compressi
- Può essere necessario configurare il sistema dove installare
- Problema della gestione delle dipendenze
- Integrazione con il sistema operativo

Il fattore chiave: l'utente

- Quante installazioni si prevedono?
- Chi esegue l'installazione? Che esperienza ha?
- Che canale di distribuzione si vuole usare?



CPack...

- È il “fratello” di CMake per la pacchettizzazione
- Consente con una sintassi unificata di creare pacchetti per differenti piattaforme e sistemi di installazione
- Può funzionare con o senza CMake

...ma...

Non funziona *ancora* bene come CMake, è completo solo da CMake 2.6

Cosa produce

TGZ tar + gzip

STGZ tar.gz auto-estraenti

NSIS Script NSIS

ZIP zip

TBZ2 tar + bzip2

TZ tar + Z

PackageMaker Mac OSX Package Maker.

OSXX11 Mac OSX X11 Bundle.

RPM Red Hat Package

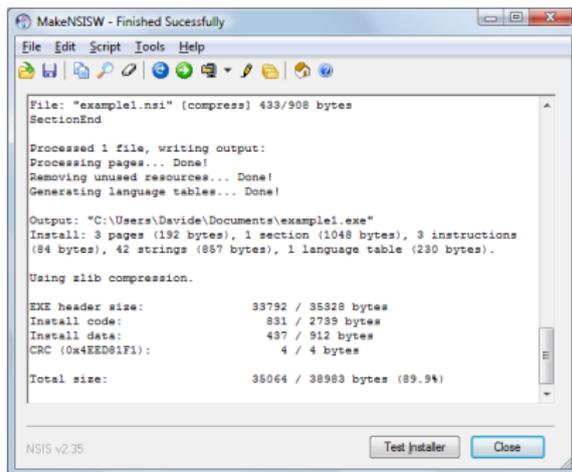
DEB Debian Package



- Linguaggio di scripting per installatori Windows
- Nato per fare l'installatore di WinAmp
- Potente e flessibile

Cosa fa

- Crea eseguibili autoinstallanti
- Esegue altri programmi
- Imposta chiavi del registro di sistema
- Crea un disinstallatore automatico
- Gestisce la licenza
- E molto altro...



```
MakeNSISW - Finished Successfully
File: "example1.nsi" (compress) 433/908 bytes
SectionEnd

Processed 1 file, writing output:
Processing pages... Done!
Removing unused resources... Done!
Generating language tables... Done!

Output: "C:\Users\Davide\Documents\example1.exe"
Install: 3 pages (192 bytes), 1 section (1048 bytes), 3 instructions
(84 bytes), 42 strings (887 bytes), 1 language table (230 bytes).

Using zlib compression.

EXE header size:          33792 / 35328 bytes
Install code:             831 / 2789 bytes
Install data:             437 / 912 bytes
CRC (0x4EED81F1):         4 / 4 bytes

Total size:               35064 / 38983 bytes (89.94)
```

NSIS v2.35

Test Installer Close

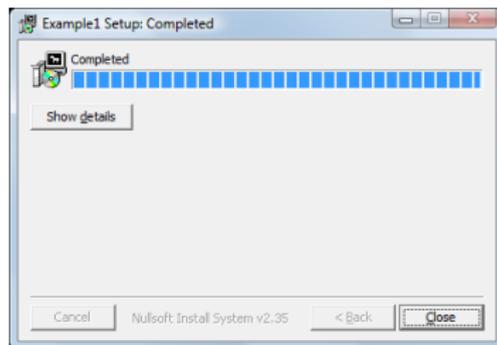
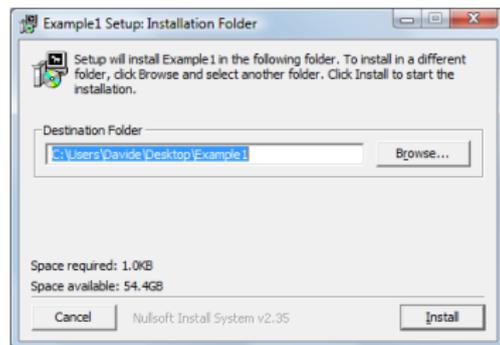


Example1.nsi

```
; Semplice script NSIS 1
; Chiede un percorso di installazione , 2
; poi ci installa una copia di se stesso 3
4
Name "Example1" ; Nome dell' installatore 5
OutFile "example1.exe" ; Il nome del file pacchetto 6
InstallDir $DESKTOP\Example1 ; Cartella di installazione di default 7
RequestExecutionLevel user ; Richiede i privilegi su Windows Vista 8
Page directory ; Imposta le pagine da installare 9
Page instfiles 10
Section " Installazione " ; Cosa installare 11
    SetOutPath $INSTDIR ; Percorso di destinazione 12
    File example1.nsi ; File da installare 13
SectionEnd ; Fine sezione 14
```



NSIS: Example1 in azione



I pacchetti in Linux

- I programmi vengono separati dalle librerie
- La gerarchia delle cartelle è stabilita da uno standard (**FHS - Filesystem Hierarchy Standard**)
- Ogni pacchetto contiene i riferimenti alle proprie dipendenze
- I pacchetti vengono raccolti e distribuiti da server in rete (*repository*)

Vantaggi

- Nessuna duplicazione di librerie
- Unico sistema per installazione e aggiornamento
- Reperibilità e sicurezza per l'utente

Svantaggi

- Più complesso per lo sviluppatore
- Più soluzioni, incompatibili tra loro



Cosa serve

- il pacchetto tar.gz con i sorgenti compilabili
- Un makefile con le istruzioni di installazione
- Un file *nomepacchetto.spec* con:
 - 1 Nome e info pacchetto
 - 2 Versione
 - 3 Dipendenze
 - 4 Istruzioni di compilazione

Come si crea

Usando `rpmbuild`.





Cosa serve

- I sorgenti in una cartella
- Una sotto-cartella `debian` con i file:
 - 1 `changelog`
 - 2 `README`
 - 3 `README.Debian`
 - 4 `copyright`
 - 5 `control`
 - 6 `rules`

Come si crea

- Usando i pacchetti `debuild`, `debhelper` e `dh-make`.
- `dh_make` è in grado di creare automaticamente i file necessari
- `debhelper` ha delle utility per gestirli in maniera automatizzata
- `debuild` genera il pacchetto





[Cplusplus.com](http://www.cplusplus.com/)

<http://www.cplusplus.com/>



[C++ su Wikipedia](#)

Inglese: <http://en.wikipedia.org/wiki/C%2B%2B>

Italiano: <http://it.wikipedia.org/wiki/C%2B%2B>



[Brian W. Kernighan, Dennis M. Ritchie](#)

Il linguaggio C. Principi di programmazione e manuale di riferimento. Nuova edizione italiana, Pearson Education Italia, 2004 (ISBN 88-7192-200-X)



[Bjarne Stroustrup](#)

C++ Linguaggio, libreria standard, principi di programmazione. III edizione, Pearson Education Italia, 2000 (ISBN 88-7192-078-3)



[Bruce Eckel, *Thinking in C++*](#)

<http://www.mindview.net/Books/TICPP/ThinkingInCPP2e.html>



Riferimenti sui compilatori



[GCC Home Page](http://gcc.gnu.org/)

<http://gcc.gnu.org/>



[William von Hagen](#)

The Definitive Guide to GCC, Second Edition, Apress, 2006 (ISBN 1-59059-585-8)



[MinGW](http://www.wxwidgets.org/wiki/index.php/Supported_Platforms)

http://www.wxwidgets.org/wiki/index.php/Supported_Platforms



[Dev-C++](http://www.bloodshed.net/devcpp.html)

<http://www.bloodshed.net/devcpp.html>



[OpenWatcom](http://www.openwatcom.org/index.php/Main_Page)

http://www.openwatcom.org/index.php/Main_Page



[Microsoft Visual Studio C++ Express](http://www.microsoft.com/express/vc/)

<http://www.microsoft.com/express/vc/>



Riferimenti sugli IDE

-  [Eclipse](http://www.eclipse.org/)
<http://www.eclipse.org/>
-  [kdevelop](http://www.kdevelop.org/)
<http://www.kdevelop.org/>
-  [Anjuta](http://anjuta.sourceforge.net/downloads)
<http://anjuta.sourceforge.net/downloads>
-  [XCode](http://developer.apple.com/leopard/overview/)
<http://developer.apple.com/leopard/overview/>





GNU Make

<http://www.gnu.org/software/make/>



CMake

<http://www.cmake.org/>



Ken Martin, Bill Hoffman

Mastering CMake 4th Edition, Kitware, Inc., 2008 (ISBN 978-1930934207)





[wxWidgets Home Page](http://www.wxwidgets.org/)

<http://www.wxwidgets.org/>



[Presentazione sulle wxWidgets](http://www.montellug.it/modules/mydownloads/visit.php?cid=6&lid=45)

<http://www.montellug.it/modules/mydownloads/visit.php?cid=6&lid=45>



[Julian Smart, Kevin Hock, Stefan Csomor](#)

Cross-Platform GUI Programming with wxWidgets, Prentice Hall, 2006
(ISBN 0-13-147381-6)



[GTK+](http://www.gtk.org/)

<http://www.gtk.org/>





Qt

<http://trolltech.com/>



Alan Ezust, Paul Ezust

An Introduction to Design Patterns in C++ with Qt 4, Prentice Hall, 2006
(ISBN 978-0131879058)

Publicato online:

<http://cartan.cas.suffolk.edu/oopdocbook/opensource/index.html>



Tutorial Qt in italiano

http://www.digitalfanatics.org/projects/qt_tutorial/it/index.html



Licenze dei framework

wxWidgets: <http://www.wxwidgets.org/about/newlicen.htm>

Qt: <http://trolltech.com/products/appdev/licensing>

GTK+: <http://www.gnu.org/licenses/old-licenses/lgpl-2.1.html>



 [Il controllo di versione](http://it.wikipedia.org/wiki/Controllo_versione)
http://it.wikipedia.org/wiki/Controllo_versione

 [Subversion](http://subversion.tigris.org/)
<http://subversion.tigris.org/>

 [Client Subversion](http://en.wikipedia.org/wiki/Comparison_of_Subversion_clients)
http://en.wikipedia.org/wiki/Comparison_of_Subversion_clients

 [TortoiseSVN](http://tortoisesvn.net/)
<http://tortoisesvn.net/>





CVS

<http://ximbiot.com/cvs/>



Bazaar

<http://bazaar-vcs.org/>



Mercurial

<http://www.selenic.com/mercurial/wiki/>



Git

<http://git.or.cz/>



Riferimenti sulla pacchettizzazione



CPack

http://www.cmake.org/Wiki/CMake:Packaging_With_CPack



NSIS

http://nsis.sourceforge.net/Main_Page



Maximum RPM

<http://www.redhat.com/docs/books/max-rpm/max-rpm-html/>



Wikipedia RPM

http://it.wikipedia.org/wiki/RPM_Package_Manager



Pacchetti Debian

http://www.debian.org/doc/FAQ/ch-pkg_basics



Debian New Maintainers' Guide

<http://www.debian.org/doc/maint-guide/>

