

Università degli Studi di Padova
Corso di Laurea in Matematica
A.A. 2008-2009

INTRODUZIONE ALLA
PROGRAMMAZIONE
Barbara Di Camillo

Parte del materiale di questa lezione è stato elaborato dal
contenuto del sito
<http://www.science.unitn.it/~fiorella/guidac/indexc.html>

PARTE VII
Introduzione a C

Perché è utile

- È un linguaggio che, come il Pascal ed il Fortran (i suoi predecessori), permette di salvare i valori in variabili, di strutturare il codice, di convogliare il flusso del programma utilizzando istruzioni di ciclo, istruzioni condizionali e funzioni, di eseguire operazioni di input/output, di salvare dati in array o strutture
- diversamente da questi linguaggi
 - permette di **controllare in modo più preciso le operazioni di input/output**,
 - è un linguaggio più **sintetico** e permette di scrivere programmi piccoli e di facile comprensione.

Perché è utile

- esistono già **migliaia di righe di codice scritte** che permettono di risolvere quasi tutti i problemi legati alla programmazione
- **i sistemi operativi sono sviluppati per la maggior parte in Assembler ed in C** (uno su tutti il Kernel Linux), anche se viene incluso sempre di più codice scritto in C++ (C plus plus)
- **il C insegna a programmare**, a pensare a come impostare il codice, a risolvere i problemi che ci vengono messi di fronte
- una volta imparato il C **troverete molto più facile imparare un altro linguaggio di programmazione** (le basi insegnate verranno ritrovate in linguaggi come il C++, il Java, il PHP, il JavaScript, ecc.)

Peculiarità

- **Dimensioni del codice ridotte** - Il codice sorgente di un programma in C ha un peso (in Kb) relativamente piccolo
- **Dimensioni dell'eseguibile ridotte** - Anche una volta compilato, un programma in C, risulta molto piccolo e quindi di più facile diffusione; un programma in C potrà essere eseguito solamente sul medesimo Sistema Operativo per cui è stato compilato.

Peculiarità

- **Implementazione dei puntatori** - Il C fa un largo uso di puntatori per le operazioni riguardanti la memoria, gli array, le strutture e le funzioni.
- **Efficienza dei programmi** - Un programma scritto in C, è in grado di **gestire efficientemente la memoria** e risulta particolarmente efficiente (tempo e allocazione di memoria).

Peculiarità

- **Può essere compilato su una vasta gamma di computer** - Ogni computer può differire dagli altri per almeno due cose
 - l'architettura
 - il sistema operativo
- un programma scritto in C può però essere compilato su computer con diverse caratteristiche data l'alta disponibilità di compilatori per diverse piattaforme (il fatto di essere **sulla scena da molti anni** e la sua enorme diffusione ne fanno, di fatto, uno strumento altamente portabile).

Peculiarità

- **è un linguaggio di alto livello** - Un linguaggio di programmazione viene definito di alto livello tanto più **si avvicina alla terminologia umana**, inversamente si dice che un linguaggio è di basso livello se il suo codice si avvicina al linguaggio macchina (quello formato da 0 ed 1); tipico esempio di linguaggio a basso livello è l'Assembler, mentre linguaggi ad alto livello sono, oltre al C, il C++, il Java e molti altri.
- La particolarità dei linguaggi ad alto livello è quella di **avere una semplice sintassi in cui si usano parole della lingua inglese per descrivere comandi corrispondenti a decine di istruzioni in assembler o centinaia di istruzioni in linguaggio macchina.**

Peculiarità

- **Può maneggiare attività di basso livello** - è il linguaggio di più basso livello tra i linguaggi di alto livello. Questo è dovuto al fatto che ha **poche istruzioni, gestisce in maniera efficiente la memoria ed è possibile inserire direttamente all'interno di un file in C del codice Assembler**.

Programmare in C

- Per programmare in C sono necessari due strumenti:
 - un buon editor di testi. È possibile usare qualsiasi editor di testi disponibile sul proprio sistema operativo, tuttavia ve ne sono alcuni pensati ad hoc per la programmazione.
 - un "sistema di compilazione" (un tool di programmi, come compilatore, linker, debugger, ecc.) che controlla gli errori e traduce il sorgente in C in un eseguibile (composto da istruzioni di codice macchina)

- Facciamo una digressione su editor e compilatori...

- Dal sito:

<http://www.science.unitn.it/~fiorella/guidac/indexc.htm>

!

riporto una lista di compilatori ed editor disponibili per i più diffusi sistemi operativi.

- Si veda il sito sopra citato per ulteriori dettagli e link utili

Compilatori in Linux

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

- È l'acronimo di "GNU Compiler Collection" e permette di compilare codice scritto in **C**, **C++**, **Fortran**, **Java**, **Pascal**, ecc. Spesso con il termine *GCC* si intende "GNU C Compiler", mentre vengono riservati altri nomi quando si intende l'utilizzo per linguaggi come il Fortran (*GNAT*) o il C++ (*G++*);

Imparare ad usarlo è abbastanza facile, ad esempio per compilare un programma in **C** basta digitare dalla shell dei comandi:

```
gcc programma.c
```

Traduzioni di questa pagina



GNU Operating System

Iscrivetevi a *Free Software Supporter*

Aggiornamenti mensili su GNU e la FSF

email@example.com

Cos'è il Progetto GNU?

Il Progetto GNU è stato lanciato nel 1984 per sviluppare un sistema operativo Unix-compatibile completo che fosse [software libero](#): il sistema GNU.



Il kernel di GNU non è ancora stato completato, quindi GNU è usato con il kernel Linux. La combinazione di GNU e Linux è il

sistema operativo GNU/Linux, ora usato da milioni di persone.

Talvolta tale combinazione è chiamata impropriamente [Linux](#). Ci sono molte varianti, o "distribuzioni", di GNU/Linux.

Servono sviluppatori per GNUstep

GNUstep è un ambiente desktop orientato agli oggetti pienamente funzionale. Occorrono sviluppatori che scrivano applicazioni GNUstep affinché possiamo offrire un ottimo prodotto agli utenti. Si veda <http://www.gnustep.org/> per ulteriori informazioni.

Freedom Fry



[Audio/Video GNU](#)

[GNU GPL](#)

[GNU LGPL](#)

[GNU GFDL](#)

[GNU AGPL](#)

[Licenze di Software Libero](#)

[Pacchetti GNU](#)

[Manuali GNU](#)

[Mailing list di GNU](#)

Editor in Linux

Emacs (abbreviazione di Editor MACroS)

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

- Le sue potenzialità sono enormi, dalla colorazione del testo, all'indentazione automatica, alla possibilità di interagire in modo trasparente con la documentazione, con il compilatore (ed eventuale debugger) e la totale funzione, volendo, anche di terminale Linux.
- È disponibile (allo stesso sito) anche per Windows e Mac
- Ha una interfaccia per ambienti grafici chiamata XEmacs.

Editor in Linux

Kdevelop

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

- Riconosce automaticamente i percorsi dove sono installati i programmi necessari (come *GCC*) e le relative guide di supporto
- è possibile creare semplici file in *C* o in *C++*, ma anche progetti completi "a finestre" per Linux.
- È uno dei migliori programmi disponibili per Linux, ma, forse, troppo difficile da usare per il principiante che vuole scrivere semplici programmi in *C*.

Editor in Linux

VIM ("Vi IMproved")

<http://vim.sourceforge.net/>

- Insieme ad Emacs è uno dei migliori editor disponibili per Linux ed è disponibile (allo stesso sito) anche per MAC;
- tra le sue molte caratteristiche menzioniamo l'uso di finestre multiple, aiuto in linea, scrolling orizzontale, formattazione del testo, supporto per il mouse, linguaggio di scripting incluso, uso dei plugin, indentazione automatica, colorazione del testo
- Forse un po' difficile da usare all'inizio
- E' disponibile per praticamente tutti i sistemi operativi

Editor in Linux

GNU Nano

<http://www.nano-editor.org/>

- Versione avanzata del famoso Pico, questo semplice ma potente editor è per coloro che vogliono scrivere codice dalla shell, senza entrare nell'ambiente grafico
- è considerato uno degli editor non grafici migliori disponibili per Linux e vale la pena di usarlo se si vuole semplicità e velocità.

Compilatori in Windows

DJGPP

<http://www.delorie.com/djgpp/zip-picker.html>

- Porting di *GCC* (compilatore per macchine unix/linux), permette di compilare programmi scritti in *C* ed in *C++*;
- risulta un po' difficile da installare
- è completamente gratuito sia per uso personale che commerciale.

Compilatori in Windows

Lcc-win32

<http://www.cs.virginia.edu/~lcc-win32/>

- Questo ambiente di sviluppo ha al suo interno tutto quello di cui c'è bisogno, compilatore (assemblatore, linker, ecc.), interfaccia visuale per scrivere il codice, correggerlo, ecc., il manuale utente e documentazione tecnica.
- L'uso per scopi personali è gratuito, mentre per scopi commerciali è necessario comprare la licenza.

Compilatori in Windows

DevC++

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

- Anche questo prodotto è un IDE (Integrated Development Enviroment) di sviluppo e racchiude al suo interno un ottimo editor per windows e un porting del GCC come compilatore.

Editor in Windows

1st Page 2000

<http://www.html.it/1stpage/>

- Uno dei migliori editor disponibili per Windows, è abbastanza veloce, permette di aprire più file contemporaneamente, inoltre ha l'highlighting del testo, il conteggio del numero di righe e colonne
- pensato soprattutto come editor html, si rivela uno strumento utile anche per scrivere programmi in C, C++ e Java

Editor in Windows

jEdit

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

- Ottimo editor multiplatforma che servirebbe per scrivere codice in Java, ma che può essere usato tranquillamente per scrivere codice in C.
- La sua potenza risiede, oltre che nelle funzionalità avanzate che un editor deve avere (numero di riga, testo con highlighting, ecc.), anche nell'alto livello di configurabilità e nella sicurezza di poter usare JEdit con quasi tutti i sistemi operativi (è disponibile anche per MAC e Linux).

Compilatori in MAC

GCC

- In MAC, il compilatore *GCC* è incluso
- basta aprire una console ed è possibile iniziare ad usarlo; la compatibilità con quello per Linux dovrebbe essere praticamente assoluta.

Scelta di editor e compilatore

L'accoppiata migliore per iniziare è formata da:

- un editor a scelta tra **jEdit**, **Emacs** e **VIM**, perché tali editor sono disponibili per tutti i sistemi operativi (inoltre se passate da un sistema all'altro i vostri file sono portabili)
- **GCC** come compilatore perché in questo modo si imparano a compilare i programmi da console, con l'enorme vantaggio di esperienza rispetto all'utilizzo di sistemi automatici (IDE) come Dev-C++ o Kdevelop.

- In laboratorio utilizzeremo *GCC* come compilatore e *Emacs* come editor, lavorando da *Linux*
- Per esercitarsi sul proprio *PC* conviene installare *Emacs* e, se il sistema operativo è *Windows* e quindi non si ha a disposizione *GCC*, si hanno 4 diverse possibili opzioni:
 1. Installare *Linux* sul proprio *PC* come sistema alternativo a *Windows*;
 2. Installare un emulatore sulla propria partizione *Windows* e installarvi una copia di *Linux*;
 3. Installare *Cygwin* (<http://www.cygwin.com>), un sistema che mette a disposizione l'ambiente di sviluppo *GNU* tipico di *Linux*
 4. Installare altri tipi di ambienti di sviluppo. Tra quelli che mettono a disposizione una versione dell'ambiente *GNU* per il sistema operativo *Windows*. Ad esempio *MinGW* (<http://www.mingw.org>)

Cygwin

Cygwin

<http://www.cygwin.com>

- Emulatore della shell di Linux per Windows
- Porting di GCC (compilatore per macchine unix/linux), permette di compilare programmi scritti in C ed in C++ usando i comandi di gcc
- è completamente gratuito sia per uso personale che commerciale.
- È un po' difficile da installare

MingW

mingw (Minimalist GNU for Windows)

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

- Porting di *GCC* (compilatore per macchine unix/linux), permette di compilare programmi scritti in C ed in C++ usando i comandi di *gcc*
- è completamente gratuito sia per uso personale che commerciale.

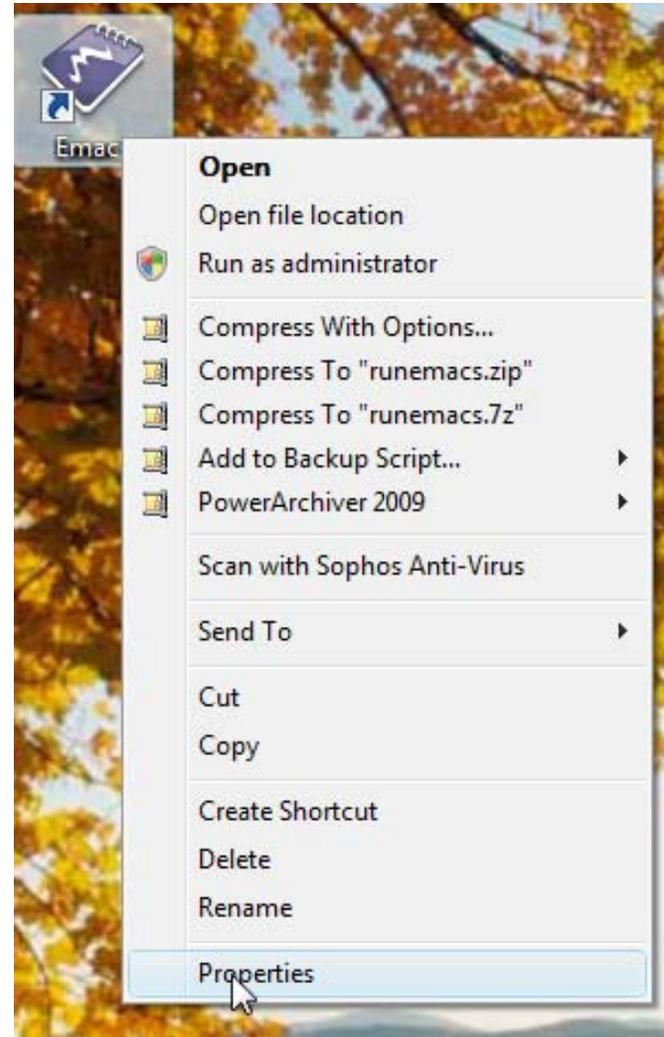
- Consiglio le opzioni 3 o 4, ma, viste le difficoltà riportate da molti studenti rispetto all'installazione di Cygwin, mi concentrerò sull'opzione 4.
- Vediamo prima l'installazione di emacs e poi quella di Cygwin in Windows

Installazione di emacs

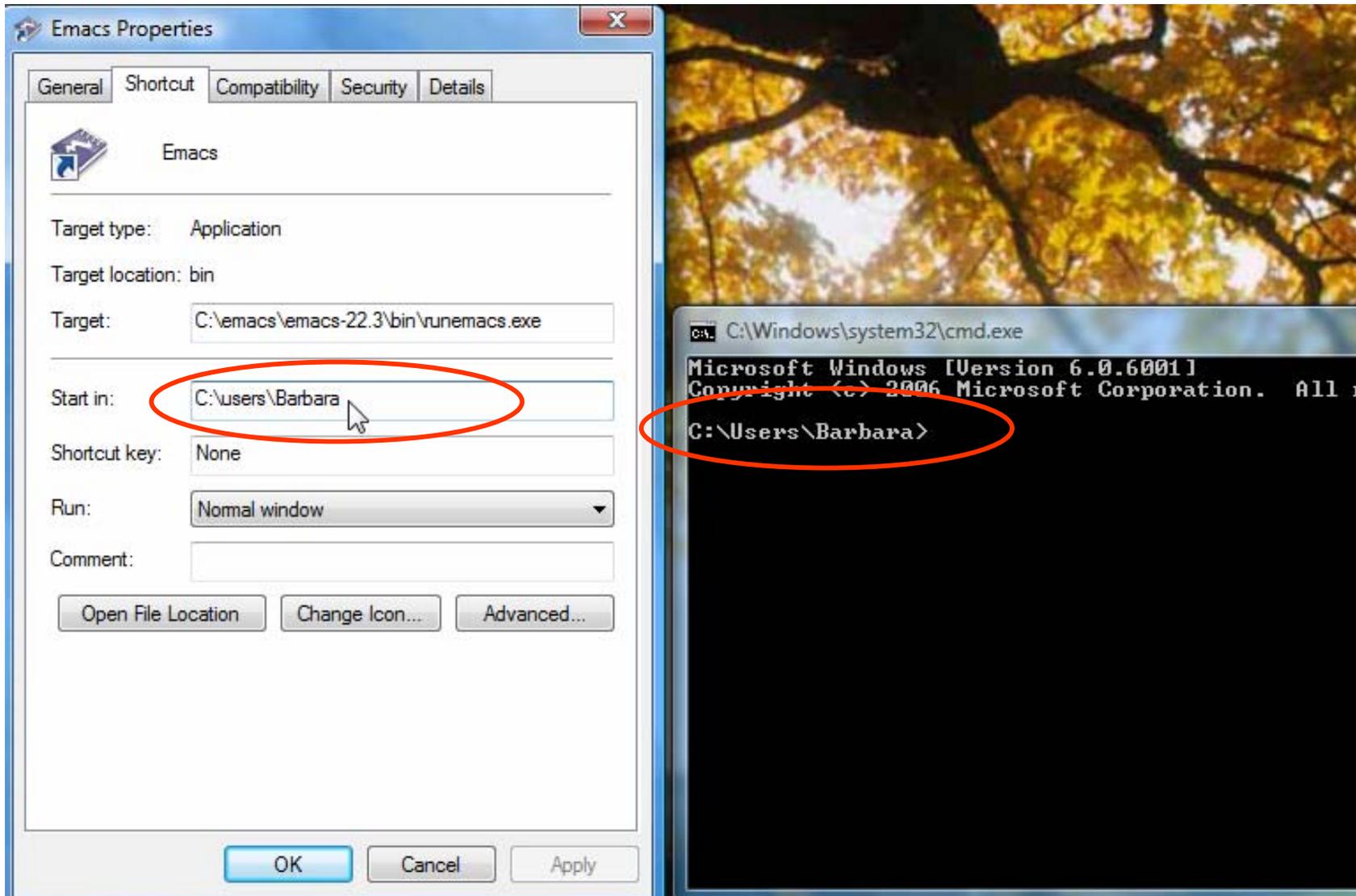
- Creare una directory "emacs" in C
`C:\emacs`
- Andare al sito:
<http://ftp.gnu.org/pub/gnu/emacs/windows/> e scaricare il file `emacs-22.3-bin-i386.zip` nella directory emacs
- Dezippare il file; apparirà la cartella `emacs-22.3`
- Andare nella directory `C:\emacs\emacs-22.3\bin` e fare doppio click sul file `addpm.exe`
- L'esecuzione del file `addpm.exe` provocherà la creazione di un'icona per lanciare emacs. Copiare l'icona sul desktop

Installazione di emacs

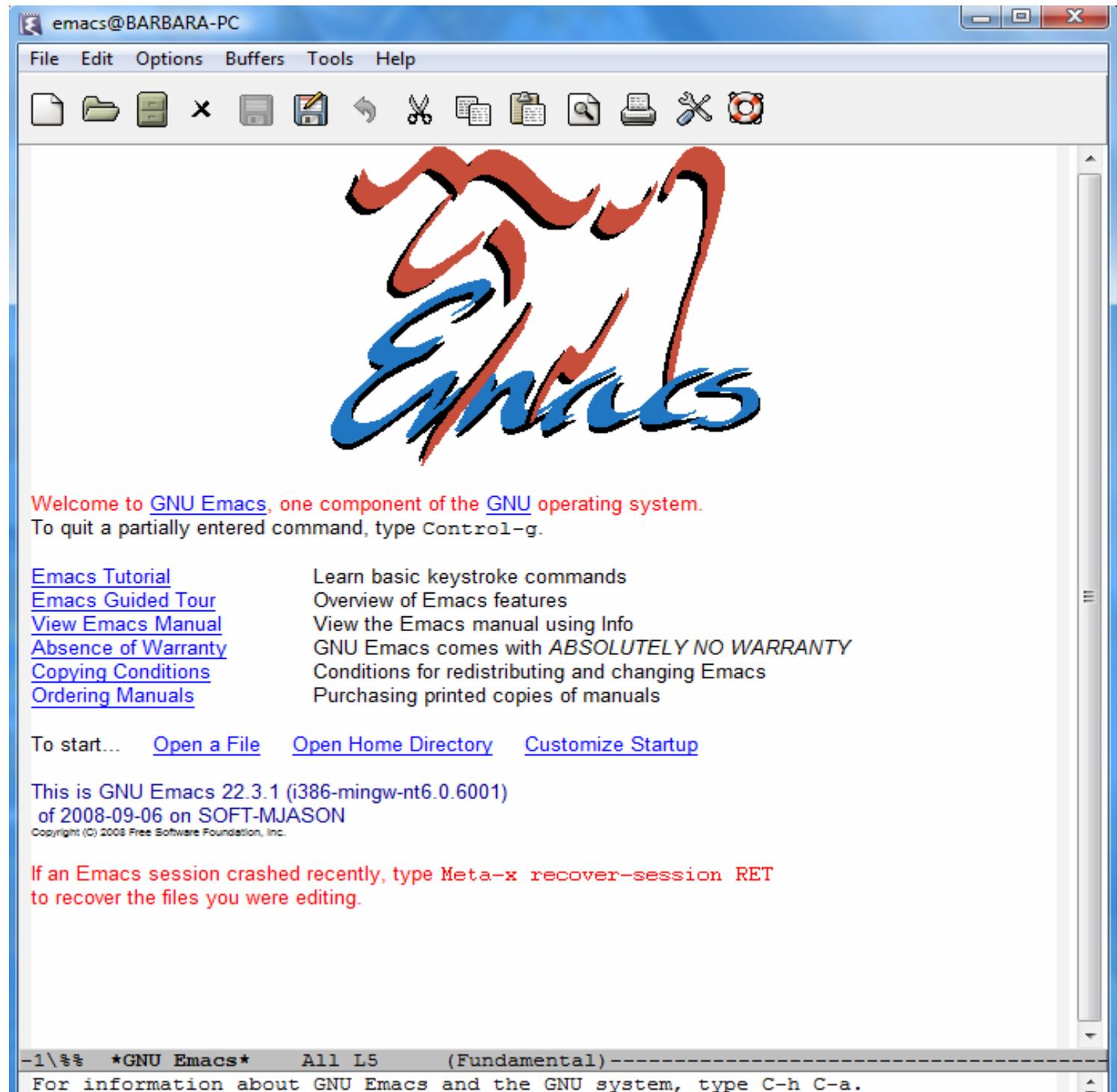
- Fare click col tasto destro del mouse sull'icona e selezionare Properties.



- Selezionare Shortcut e impostare il programma in modo che di default venga lanciato nella directory di lavoro desiderata



- Provare a lanciare emacs (doppio click sull'icona sul desktop)



Slide aggiunta

Installazione di emacs

- Se qualcosa non funziona provare ad installare emacs 21.3 seguendo dettagliatamente le istruzioni che trovate sul sito:
<http://www.claremontmckenna.edu/math/alee/emacs/emacs.html>
- Al sito <http://www.gnu.org/software/emacs/> si possono trovare la documentazione di emacs e le istruzioni relative all'installazione su diversi sistemi operativi.
- In Linux e Mac emacs dovrebbe comunque già essere installato

Installazione di MinGW

- Andare sul sito www.mingw.org e selezionare (in basso a sinistra) "[HOWTO Install the MinGW \(GCC\) Compiler Suite](#)"
- Selezionare https://sourceforge.net/project/showfiles.php?group_id=2435&package_id=240780 e scaricare [MinGW-5.1.4.exe](#)

MinGW | Minimalist GNU for Windows - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.mingw.org/

Most Visited Getting Started Latest Headlines

About Documentation Recent posts News aggregator

MinGW

Minimalist GNU for Windows

Search

Administrative Update

The Old MinGWiki pages are again offline as a result of action from SF due to flock issues with the phpwiki application. We'll be trying a different approach.

User login

Login/Register

Navigation

- About
 - Contact
 - History
 - Licensing Terms
 - Mailing Lists
 - Privacy
 - Reporting Bugs

About

MinGW: A collection of freely available and freely distributable Windows specific header files and import libraries, augmenting the GNU Compiler Collection, (GCC), and its associated tools, (GNU binutils). MinGW provides a complete Open Source programming environment for Windows. It is a complete replacement of native Windows programs that do not depend on any 3rd-party C++ runtime DLLs.

- Licensing Terms
- Mailing Lists
- Privacy
- Reporting Bugs
- Submit Patches
- Downloads
- Sitemap

▶ Documentation

○ Recent posts

▶ News aggregator

MSYS: A Minimalist GNU for Windows (MinGW) compatible Bourne shell environment, with a small collection of UNIX command line tools. Primarily used for building Open Source software, but also useful for running native Windows programs.

Mailing List: You are encouraged to join, and in which you are invited to participate. There are several mailing lists for general questions: please use it; do not create wiki pages or comments to the archives can be found on the [mailing lists](#) page. You may search the archives on the right hand side of every <http://www.mingw.org> page.

31555 reads

Who's online

There are currently 0 users and 87 guests online.

Popular content

Today's:

- HOWTO Install the MinGW (GCC) Compiler Suite
- MSYS
- MinGWiki
- Getting Started

Mail List Search

Choose Mail List:
MinGW-users

Require all words?

Search

Recent Project News

- Source Navigator NG4 [more](#)

Recent File Releases

- GNU Binutils GNU binutils-2.19.1 released (Tue, 03 Feb 2009 13:37:58 GMT)
- MSYS Base System Technology Preview: MSYS-1.0.11 released



Administrative Update

The Old MinGWiki pages are again offline as a result of action from SF due to flock issues with the phpwiki application. We'll be trying a different approach.

User login

[Login/Register](#)

Home

HOWTO Install the MinGW (GCC) Compiler Suite

[View](#)

[Revisions](#)

Posted July 25th, 2008 by mingwadmin

[getting started](#)
[install](#)
[mingw](#)

TODO: Convert to the MediaWiki formatting as discussed at [Converting Old MinGWiki Pages](#)

TODO: Revise untar instructions.

Using the (possibly "Proposed"/"Candidate") Installer

If you are new to MinGW, the recommended way to begin is to download an installer executable from https://sourceforge.net/project/showfiles.php?group_id=2435&package_id=240780 and run it (e.g. "MinGW-5.1.4.exe" -- currently there is only a 'Proposed' version). This package will install mingw-runtime, w32api, GCC, binutils and mingw32-make but it may not be the latest versions of the individual packages. Once you have all these packages (although mingw32-make is not necessary if you have MSYS or Cygwin) and extract them into the MinGW directory.

Warning: * For the same path as MSYS (but it can be a subdirectory as

Architettura a 32 bit !!!

Mail List Search

Choose Mail List:

MinGW-users ▾

Require all words?

You have selected Automated MinGW Installer

Please choose the file that best matches your architecture or operating system from the list of releases and files contained in this package.

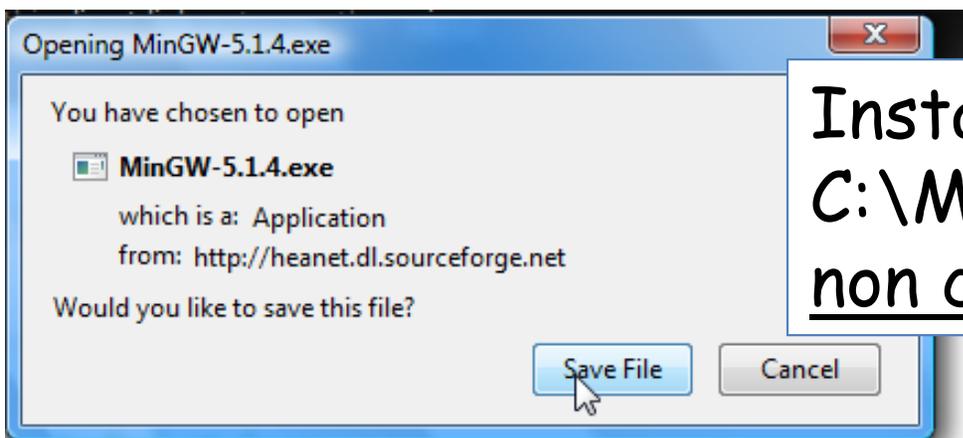
 [Options](#)

Package	Release	Filename	Size	Downloads	Architecture	Type
---------	---------	----------	------	-----------	--------------	------

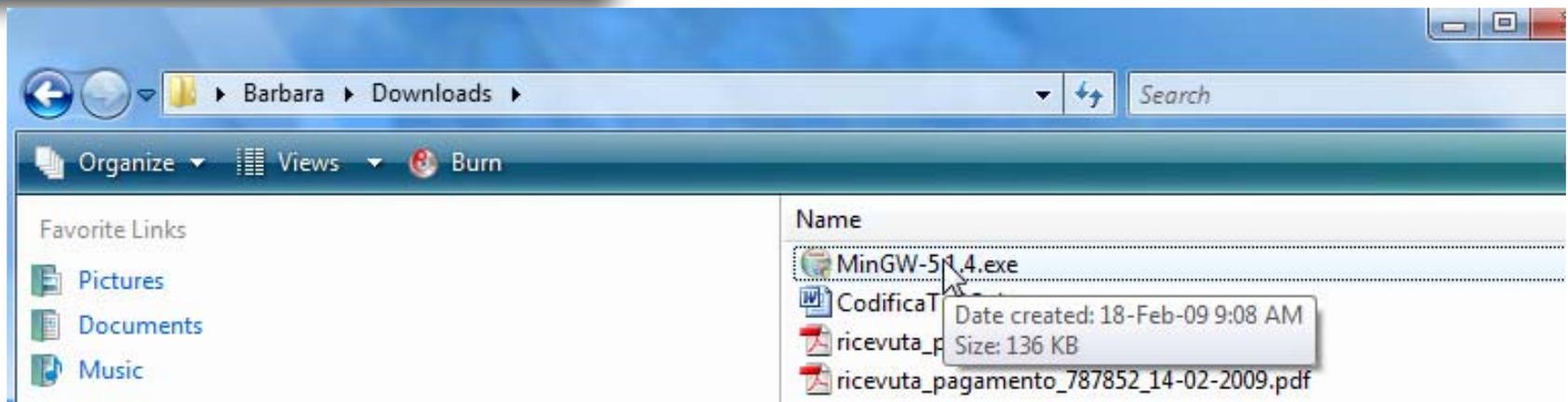
[Automated MinGW Installer](#)

Latest	MinGW 5.1.4 	(2008-04-26 21:51)				
	MinGW-5.1.4.exe 		140095	666981	i386	.exe (32-bit Windows)
	MinGW-5.1.4-src.tar.gz 		368476	43293	i386	Source .gz

Totals: 1 2 508571 710274



Installare il programma nella directory `C:\MinGW` o comunque in un path che non contenga spazi!!!



Durante l'installazione (che richiede un po' di tempo) seguire le istruzioni di default.

L'unica opzione su cui non è il caso di seguire il default è quella in cui vi viene chiesto se installare i compilatori di diversi linguaggi (ovviamente selezionate C, ed eventualmente i compilatori di altri linguaggi...)

Installazione di MinGW

Perché MinGW possa funzionare dovete settare le variabili d'ambiente di Windows in modo che il sistema operativo possa trovare il programma, sappia cioè in che PATH cercarlo.

Le istruzioni per settare le variabili d'ambiente si trovano sul sito di MinGW in
Documentation -> HOWTO

Cercare la parola chiave "system variable" e selezionare "Here"

MinGWWiki | MinGW - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.mingw.org/wiki/MinGWiki

Most Visited Getting Started Latest Headlines

About **Documentation** Recent posts News aggregator

MinGW

Minimalist GNU for Windows

Home

MinGWiki

Posted May 8th, 2007 by earnie

These pages are freely editable and are a community effort to display and give direction. We are currently in the process of moving the data from the [previous wiki](#), so you may wish to check there as well. Feel free to join in the effort.

You are welcome to add new pages, and to update or to add relevant comments to existing pages. However, you should not treat this as a forum, for asking arbitrary questions about [MinGW](#) or [MSYS](#); those should be directed to the [mailing list](#).

MinGWiki Pages How to add or edit MinGWiki pages.

Getting Started What to get depending on your needs, where to get it, and how to install it properly.

Package Guide An explanation of the packages that make up MinGW: what they do, where to get them, and how to use and compile them.

FAQ [Answers to questions](#) related to compiling issues, installation problems, debugging tips and MinGW cautions. Some answers are expanded on more comprehensively in [HOWTO](#) articles.

HOWTO Comprehensive articles explaining specifics of compiling issues, installation problems, debugging tips, MinGW cautions and more.

Source Examples Community supplied source code examples.

Community Supplied Links Links to resources used by MinGW community members, such as tutorials, utilities and programs.

Unanswered Please list here all questions that you would like documented by someone else.

Administrative Update

The Old MinGWiki pages are again offline as a result of action from SF due to flock issues with the phpwiki application. We'll be trying a different approach.

User login

Login/Register

Navigation

- About
 - Contact
 - History
 - Licensing Terms
 - Mailing Lists
 - Privacy
 - Reporting Bugs
 - Submit Database

Mail List Search

Choose Mail List:

MinGW-users

Require all words?

Search

MinGW for First Time Users HOWTO

[View](#)

[Revisions](#)

Posted January 17th, 2009 by iddy

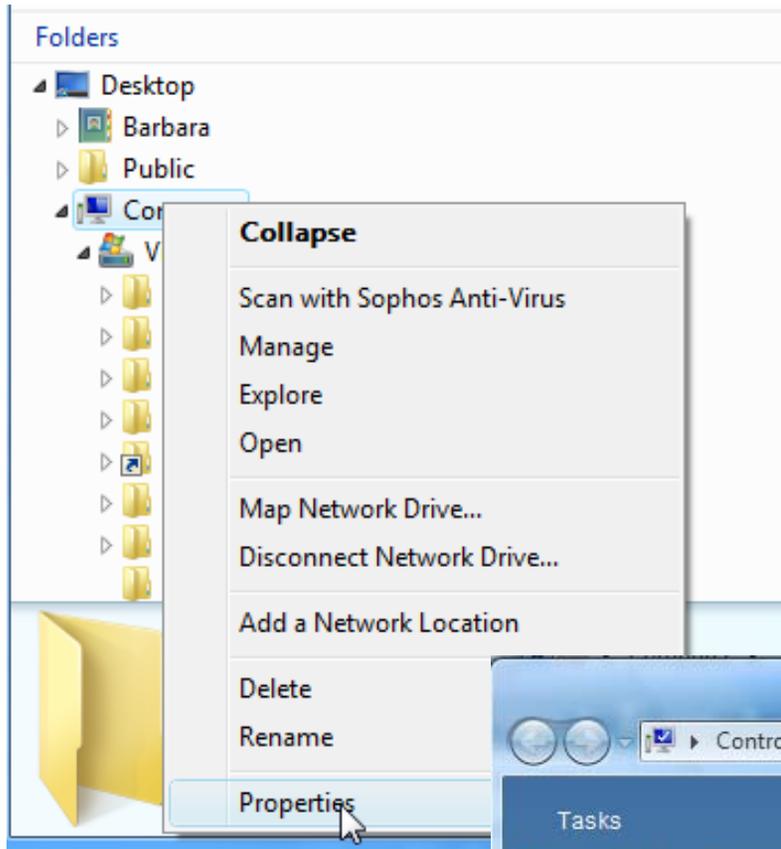
 [how to](#)

For those who are just starting out, this will explain how to use MinGW to compile a C++ executable, in a command prompt environment.

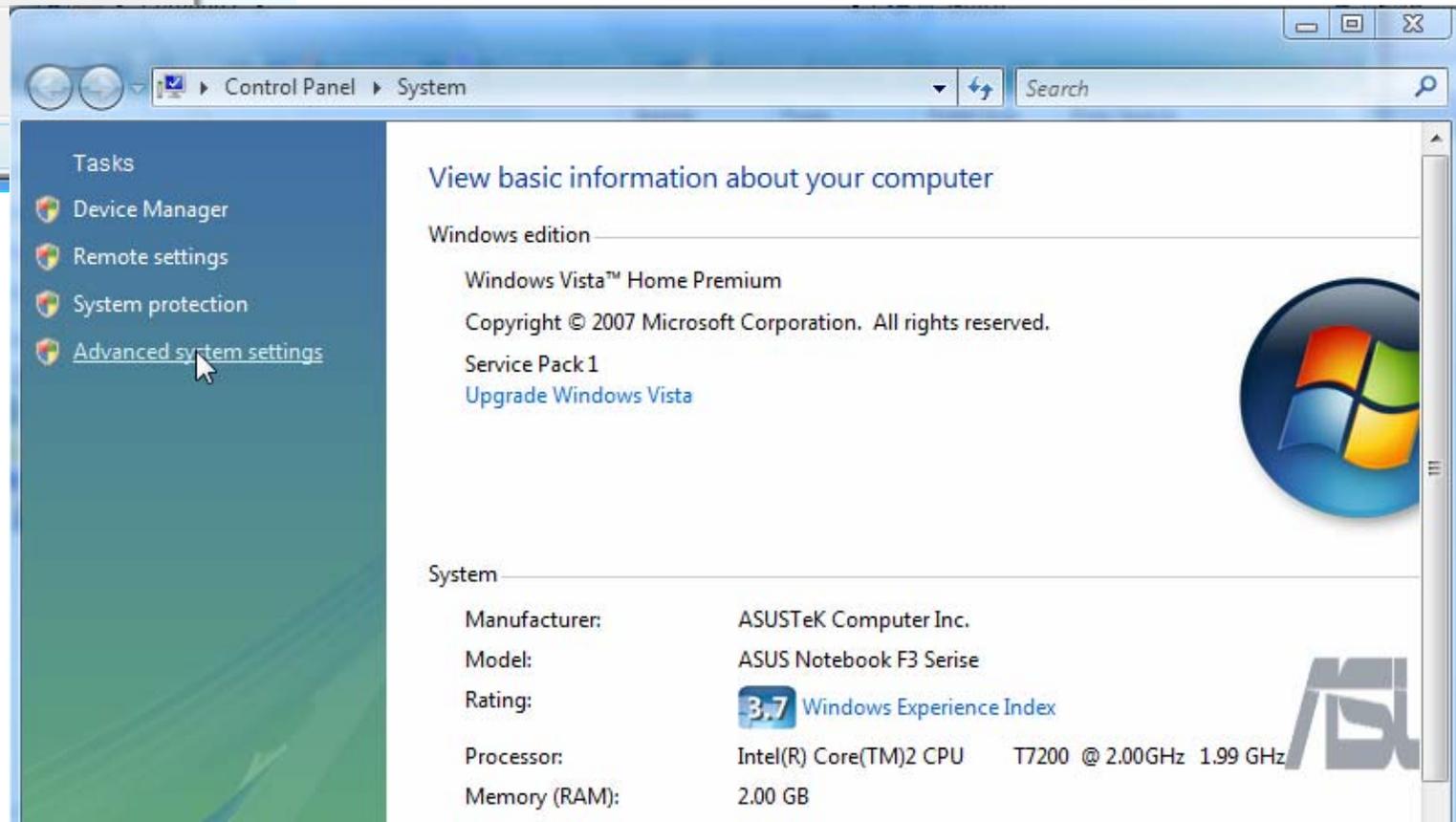
First you must make sure you have installed MinGW and set the System Variable for Path is set to express the location of MinGW. This is explained [Here](#).

Download the setup files and install the applications. Before you use them you need to tell Windows where to find them; specifically this means adding the install directory to your **PATH**. To do this carry out the following procedure (Warning - adding entries to the path is totally benign, however if you delete you mess up your path string then you could seriously compromise the functioning of your computer, please be careful):

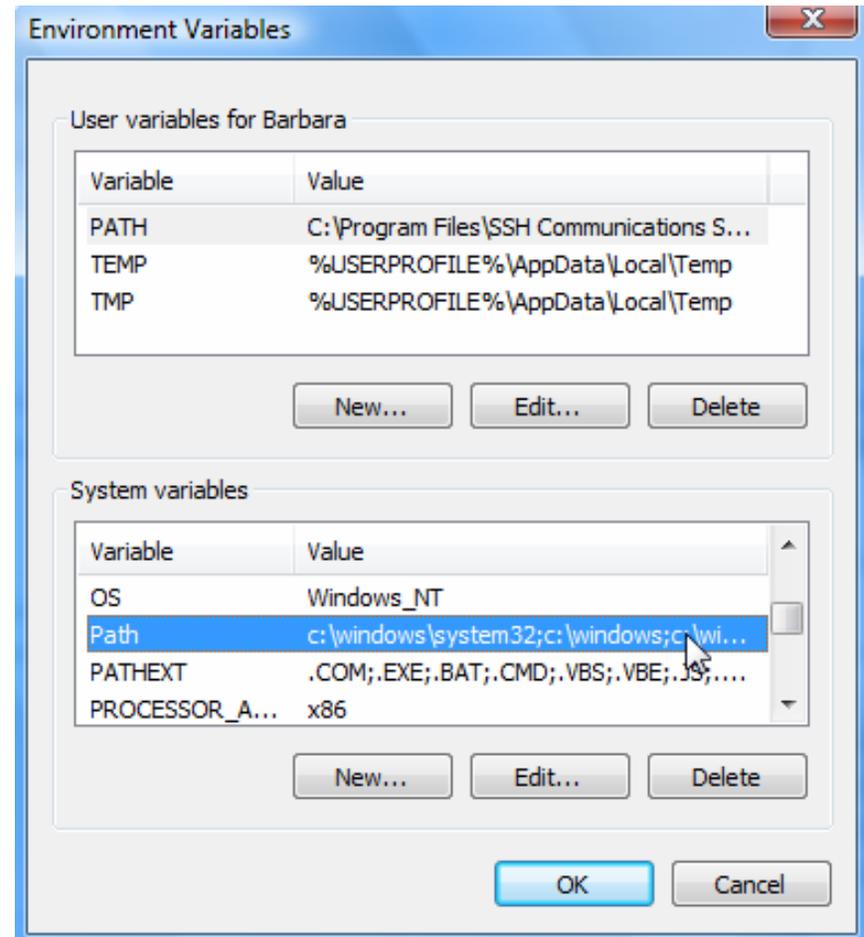
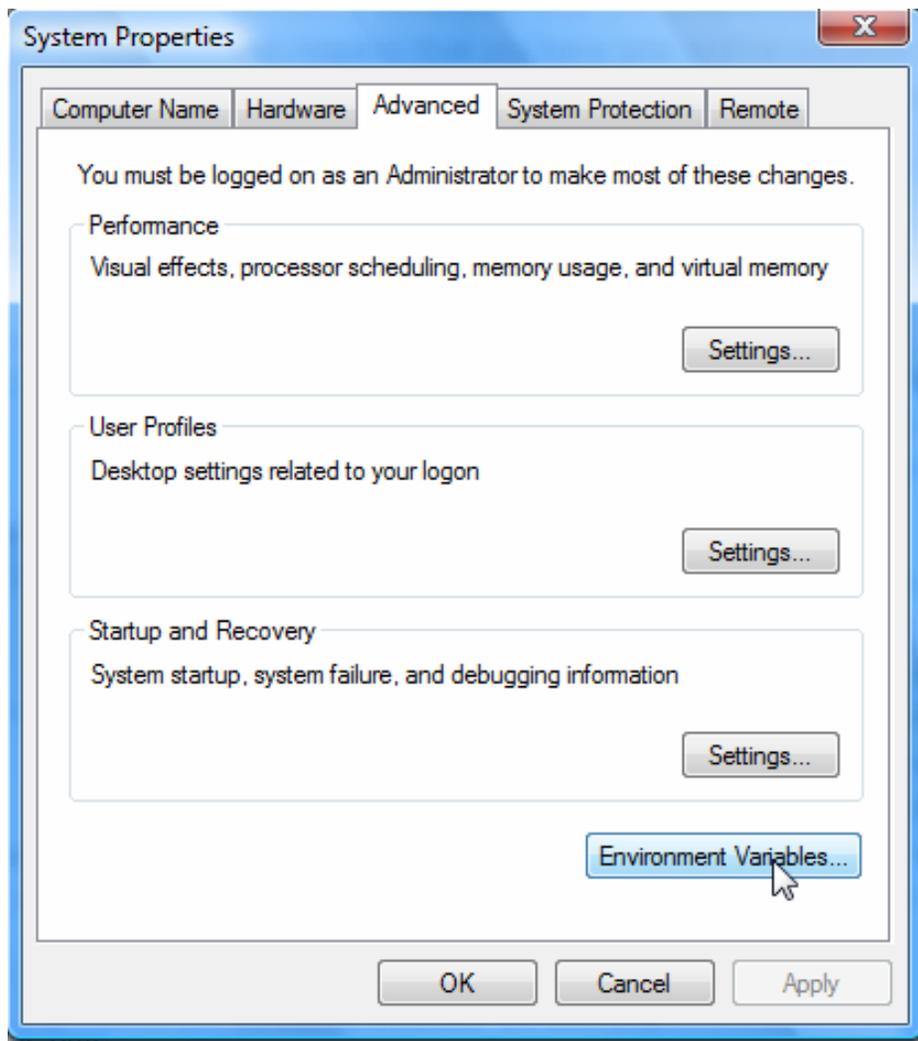
1. Right-click on "My Computer" and select "Properties".
2. Click Advanced -> Environment Variables.
3. In the box entitled "System Variables" scroll down to the line that says "Path" and double-click the entry.
4. You will be presented with a dialog box with two text boxes, the bottom text box allows you to edit the Path variable. It is *very* important that you do not delete the existing values in the path string, this will cause all sorts of problems for you!
5. Scroll to the end of the string and at the end add ";<installation-directory>\bin". Here <installation-directory> is the full address of the installation directory; if you accepted the defaults when installing tar and gzip then it will (probably) be "C:\Program Files\GnuWin32". Don't forget the semicolon, this separates the entries in the path.
6. press OK -> OK -> OK and you are done.



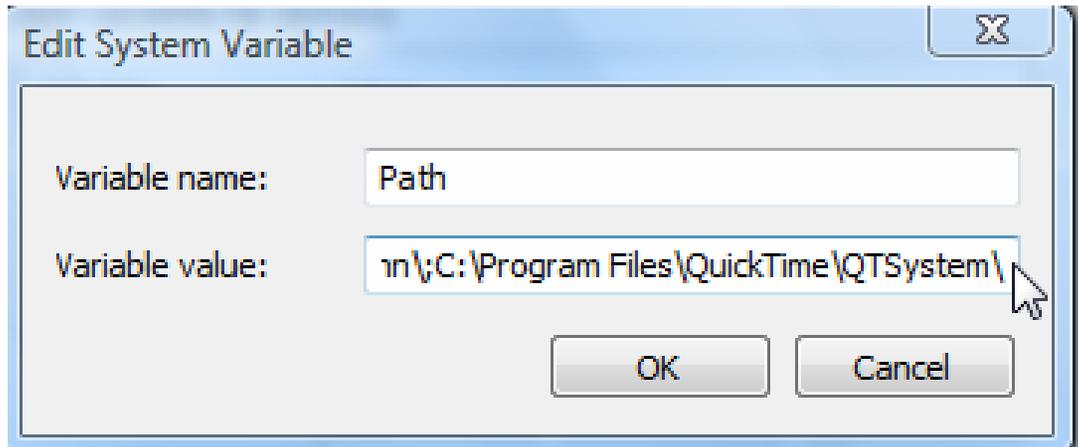
computer -> properties ->
advanced system settings



Slide modificata



Advanced -> Environment Variables -> System Variables



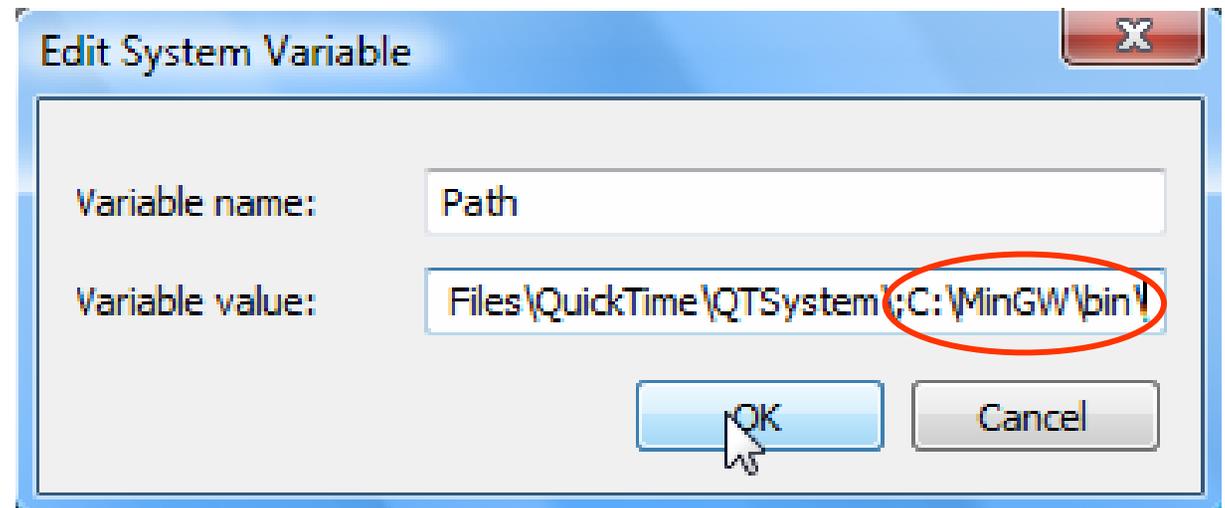
Posizionarsi alla fine della stringa Variable value

ATTENZIONE: non cancellare nulla altrimenti molti programmi o addirittura il SO potrebbero non funzionare più correttamente !!!

Scrivere:

**;C:\MinGW\bin\
e fare click su OK**

C:\MinGW è il path della directory dove avete installato MinGW



- Riprendiamo la trattazione del linguaggio C ...

HelloWorld in C

Il primo programma che scriveremo in C è il famoso "Hello World!" (Ciao Mondo) che ha soltanto lo scopo di stampare a video la scritta per spiegare la sintassi basilare del linguaggio.

Nome file: helloworld.c

```
#include<stdio.h>

main() {
    printf("Hello World!\n");
}
```

HelloWorld in C

- **#include** è una direttiva del preprocessore, un comando, che permette di richiamare le librerie standard del C. Senza librerie un programma non avrebbe a disposizione i comandi per eseguire anche le operazioni più semplici, come leggere un file o stampare a video una scritta.
- La funzione principale in un qualsiasi programma in C è la **main()** che, in questo caso, non ha parametri, ma può ricevere anche degli input da riga di comando. Il main è indispensabile ed unico e deve esserci sempre;

HelloWorld in C

- **Le parentesi graffe servono per delimitare blocchi di istruzioni**, o come vengono abitualmente chiamate "statement", che sono eseguite in ordine, da quella più in alto, giù fino all'ultima;
- **Il punto e virgola serve per "chiudere" un'istruzione**, per far capire che dopo quel simbolo inizia una nuova istruzione.
- la prima funzione che incontriamo (che è anche una delle più usate) è **printf()**, che stampa a video tutto quello che gli viene passato come argomento. Fa parte della libreria `<stdio.h>`, senza la cui inclusione, non avrebbe senso.

La libreria standard

- La Libreria standard del C è una raccolta di Header file e librerie di funzioni (come `stdio.h`) usate per implementare operazioni comuni, come l'input/output e l'elaborazione di stringhe nel linguaggio di programmazione C.
- Quasi tutti i programmi C si basano sulla libreria standard per funzionare.
- Un header file è un file che aiuta il programmatore nell'utilizzo di librerie durante la programmazione. Un header file di C è un semplice file di testo che contiene i prototipi delle funzioni definite nel relativo file.c. I prototipi permettono al compilatore di produrre un codice oggetto che può essere facilmente unito dal linker (e per questo in gergo "linkato") con quello della libreria in futuro, anche senza avere la libreria sottomano al momento.

La libreria standard

Input e Output	<stdio.h>
Test dei caratteri	<ctype.h>
Funzioni su stringhe	<string.h>
Funzioni Matematiche	<math.h>
Funzioni Utilita'	<stdlib.h>
Funzioni Diagnostiche	<assert.h>
Liste argomenti variabile	<stdarg.h>
Salti non locali	<setjmp.h>
Segnali	<signal.h>
Funzioni Date e Ore	<time.h>
Limiti dei tipi di variabile	<limits.h>

Stringhe e caratteri

- In C, le funzioni di output (come printf) hanno il compito di stampare a video la STRINGA racchiusa tra parentesi tonde.
- Una stringa è una sequenza di CARATTERI racchiusa tra virgolette.
- Si definiscono caratteri:
 - lettere (a, X, é...)
 - cifre (0, 1, 2...)
 - simboli (+, £, *...)
 - sequenze di escape (o di controllo)

Sequenze di escape

- In particolare, le sequenze di escape sono sequenze di caratteri che provocano particolari comportamenti della funzione `printf`. Ad esempio, la sequenza `'\n'` fa sì che il cursore si posizioni nella posizione iniziale della riga successiva dello schermo.

`\a` bell

`\b` backspace

`\f` formfeed (salto pg)

`\n` a capo

`\r` ritorno carrello

`\t` tabulazione orizzontale

`\v` tabulazione verticale

`\\` backslash

`\?` punto interrogativo

`\'` apice singolo

`\"` apice doppio

`\ooo` carattere ottale

`\xhh` carattere esadecimale

HelloWorld in C

- All'interno di un programma C possono essere inseriti dei **commenti**. Esistono due modi per farlo

- **//**

Tutto quello che sta a destra sulla medesima riga viene considerato commento e viene ignorato (ai fini dell'interpretazione del linguaggio) dal compilatore;

- **/* ... */**

Tutto quello che è compreso tra i due asterischi viene considerato commento; questa forma viene utilizzata per commenti su più righe.

HelloWorld in C

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

```
main() { // Il main è indispensabile ed  
// unico e deve esserci sempre;  
    printf("Hello World!\n"); /*Il punto e  
virgola serve per "chiudere"  
un'istruzione, per far capire che dopo  
quel simbolo inizia una nuova  
istruzione.*/  
}
```

Variabili

- Per poter proseguire è fondamentale definire il concetto di **variabile**
- Ad esempio, a quando salviamo un numero di telefono di un amico sul cellulare, se vogliamo chiamarlo basterà inserire il suo nome (nome della variabile) ed il cellulare comporrà automaticamente il numero di telefono (valore della variabile).
- Una variabile esiste in funzione del **nome** e del suo **valore** corrispondente.

Variabili

- La comodità risiede (come nel cellulare) nel poter usare un nome per valori, che possono essere numeri o lettere, di grande entità o difficili da ricordare.
- Un altro vantaggio, non da sottovalutare, è la possibilità di usare il nome della variabile al posto del suo valore per eseguirvi sopra delle operazioni, con la possibilità, in seguito, di modificare il valore come e quante volte vogliamo

Variabili

- Le variabili non sono altro che dei contenitori, identificati da un nome **univoco**, di un qualsiasi **valore**, sia esso un numero o una stringa.
- Per poter fare qualcosa di concreto all'interno dei programmi dobbiamo utilizzare le variabili in cui il pregio di avere un corrispondente **nome-valore** permette di gestire i cambiamenti di valore ad esse associati, fornendo quindi quella **dinamicità necessaria** ad eseguire operazioni complesse e/o ripetitive con il minimo sforzo.
- Le variabili vengono definite da un **tipo** e da un **nome**.

Nome di una variabile

- Il nome per identificare una variabile viene comunemente riferito come **identificatore**.
- Un identificatore è costituito da una o più lettere, cifre o caratteri e
 - Non deve corrispondere ad una parola chiave
 - deve iniziare con una lettera o il carattere di sottolineatura (underscore "_")
 - la sua lunghezza massima dipende dal compilatore, ma generalmente non si possono superare i 31 caratteri
 - il C è case-sensitive, quindi si fa distinzione tra lettere maiuscole e lettere minuscole.

Parole chiave (Keyword) del C

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Tipo di una variabile

- Il tipo della variabile indica quale tipo di valori può assumere il contenuto della variabile stessa.
- Una variabile di tipo intero conterrà soltanto dei numeri interi, mentre una variabile di tipo carattere conterrà solamente lettere dell'alfabeto, numeri e simboli.
- Ad esempio l'intero **7** è estremamente diverso dal carattere **'7'**, infatti l'intero viene trattato come un numero e su di esso si possono svolgere le più svariate operazioni matematiche, mentre il carattere viene gestito come un simbolo (si tratti di una lettera o di un numero).

Tipo di una variabile

- Tutto ciò ha una spiegazione logica che risiede nella rappresentazione del numero stesso; se l'intero **7** viene rappresentato da un byte (otto bit) come 00000111, il carattere **'7'** viene rappresentato con un byte ma seguendo quella che è la codifica ASCII, quindi appare, a livello di bit come 00110111 che equivale ad una rappresentazione intera di 55.

Dichiarazione di variabile

- **Tutte le variabili**, prima di essere utilizzate, **devono essere dichiarate**, cioè **deve essere detto al compilatore il tipo della variabile ed il suo nome** (es. `int x`), **questo per permettergli di allocare la memoria necessaria alla variabile stessa**;
- la dichiarazione generalmente viene fatta all'inizio del programma, ma in programmi di grandi dimensioni può trovarsi anche in altre posizioni (o altri file)
- bisogna ricordare che comunque **la dichiarazione di una variabile può essere fatta una ed una sola volta**.

Inizializzazione di variabile

- Successivamente **la variabile deve essere inizializzata**, cioè le deve essere assegnato un valore, operazione che generalmente viene fatta contemporaneamente alla dichiarazione.

```
// solo dichiarazione  
int x;  
// inizializzazione  
x = 10;
```

```
// dichiarazione ed inizializzazione  
int y = 15;
```

Inizializzazione di variabile

Esempi:

- `int a;`
- `int a = 2;`
- `int a, b;`
- `float f = 2.345;`
- `char c = 'X';`

Facendo precedere il tipo dal qualificatore **const**, si ottiene una "variabile costante", il cui valore non potrà essere cambiato

- `const double pi = 3,141592653;`

Tipi di Dato

- In C esistono **vari tipi di variabili**, questo per venire incontro sia all'esigenze di **rappresentabilità** di grossi numeri sia al **maggior risparmio di memoria** possibile, utilizzando di volta in volta il tipo più adeguato ad una specifica situazione.
- Nella tabella seguente sono mostrati i vari tipi, con la parola che in C ne consente l'uso (char per carattere, int per intero, ecc...), ed il numero di bit necessari, all'interno del C, per la loro rappresentazione. La tabella si riferisce ad un'architettura a 32 bit

Tipi di Dato

Tipo di dato	Parola chiave	bit
Singolo carattere	char	8
Intero corto	int o signed short	16
Intero corto senza segno	unsigned short	16
Intero lungo	signed int	32
Intero lungo senza segno	unsigned int	32
Intero lungo	signed long	32
Intero lungo senza segno	unsigned long	32
Reale in virgola mobile (singola precisione)	float	32
Reale in virgola mobile (doppia precisione)	double	64
Reale in virgola mobile (precisione estesa)	long double	80

Tipo char

- Il tipo char può contenere un qualsiasi carattere definito secondo lo standard ASCII e quindi potrà contenere qualsiasi lettera (maiuscola o minuscola), cifra (da 0 a 9) e simbolo previsto dalla codifica.
- Per dichiarare ed inizializzare una variabile char, ad esempio inizializzandola con la lettera 'r', basta scrivere:

```
// dichiarazione ed inizializzazione  
char a = 'r';
```

Tipo char

- Il tipo **char** è adibito a contenere uno ed un solo **carattere**; questa imposizione, scomoda quando vogliamo memorizzare una successione di caratteri, è stata risolta in alcuni linguaggi adottando il tipo **string** (stringa)
- in **C** questa situazione viene risolta utilizzando un **array di char**, l'array è, a tutti gli effetti, un contenitore di variabili dello stesso tipo, che vedrete nel dettaglio durante il corso di programmazione

```
char c = 'X';  
char messaggio[] = "Questa e' una stringa";
```

Tipo int

- Il tipo `int` è quello che permette di contenere dei numeri; il tipo `int` ha le sue due varianti che sono `short` e `long`, anche se in realtà **un tipo `int` è già di per se' un tipo `short`**, mentre il `long` permette di estendere (utilizzando due byte in più) il range dei valori che il **tipo `int` può assumere** per venire incontro all'esigenza di lavorare con grandi numeri.
- Il tipo `int` contiene numeri appunto interi e cioè senza la virgola e parti frazionate.

Tipo int

- Nel pezzo di codice qui di seguito vengono dichiarate ed inizializzate variabili int. Per far capire come il fatto che int rappresenti i numeri interi comporti una certa attenzione, verrà eseguita l'operazione di divisione tra due interi (utilizzando l'operatore di divisione /)

```
int x = 7;
```

```
int y = 3;
```

```
int z;
```

```
z = x / y;
```

```
/* z vale 2, cioè la parte intera della  
divisione tra 7 e 3 */
```

Tipo float e double

- I tipi **float** e **double** sono chiamati anche numeri in virgola mobile, cioè quelli che in matematica vengono chiamati numeri reali, e quindi possono essere usati per contenere tutti quei numeri che hanno parti frazionarie.
- La differenza tra i due sta solamente nei bit che sono riservati per la loro rappresentazione, che si va a riflettere, non solo nel range di rappresentazione, ma anche nel numero di cifre dopo la virgola che possono essere rappresentate. Risulta, quindi più accurata la rappresentazione utilizzando i double.

Tipo float e double

- Qui di seguito viene presentato un semplice pezzo di codice (utilizzando i double) per far vedere come, a differenza degli int, venga rappresentata la divisione tra due numeri in virgola mobile:
- Da notare che la notazione usata per rappresentare la virgola è quella inglese, cioè quella in cui si usa un punto (e non una virgola) per dividere la parte intera da quelle frazionaria.

```
double x = 7.0  
double y = 2.0  
double z;
```

```
z = x / y           // z vale 3.5
```

Conversioni di Tipo

Quando un'espressione coinvolge valori di tipo diverso, essi vengono convertiti ad un tipo comune:

- automaticamente ad un tipo "più grande", che quindi non fa perdere informazione
- esplicitamente, mediante un'operazione di **casting**. Obbligatoria se non è possibile una conversione implicita automatica.