

Laboratorio reti AA 2008/2009

Dott. Matteo Roffilli

roffilli@csr.unibo.it

**Ricevimento in ufficio
dopo la lezione**

Laboratorio reti AA 2008/2009

Per esercitarvi fate SSH su:

alfa.csr.unibo.it

si-tux00.csr.unibo.it

....

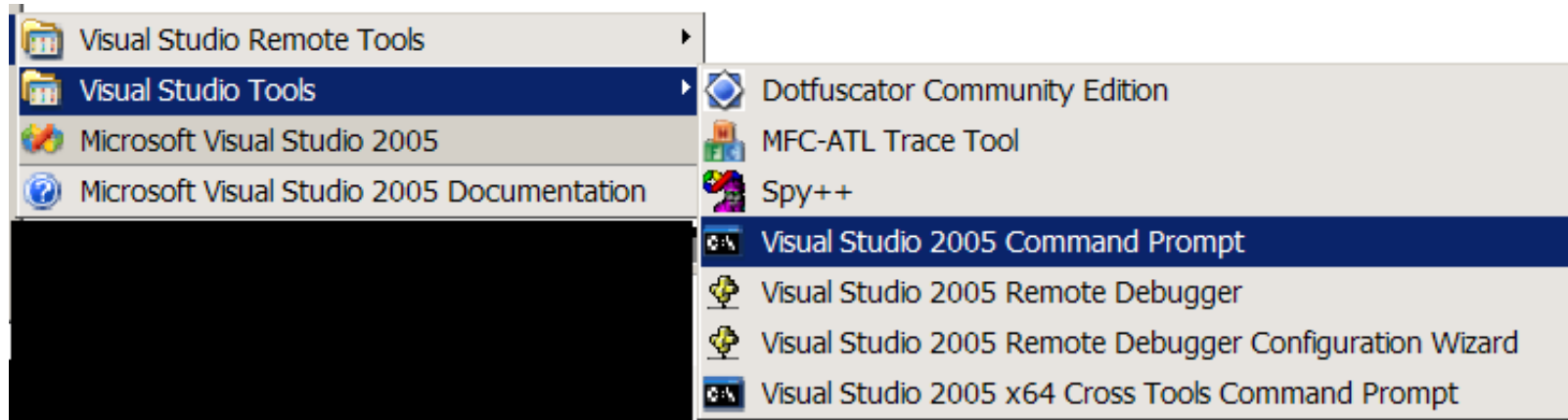
si-tux15.csr.unibo.it

Eventuali variazioni di orario/giorno verranno comunicate in anticipo via mail.

Laboratorio reti AA 2008/2009

- **Marzo**
- 5 Intro,SSH,VI/VIM,GCC base
- **12 Richiami di C e Compilazione**
- 19 Socket e Co.

Command line .NET



```
C:\WINDOWS\system32\cmd.exe - cl -?
D:\BIBLIO\BIBLIOTECHE\Reti_calcolatori_2006-07\lab2>cl -?
Microsoft (R) 32-bit C/C++ Optimizing Compiler Version 14.00.50727.42 for 80x86
Copyright (C) Microsoft Corporation. All rights reserved.

C/C++ COMPILER OPTIONS

-O1 optimize for speed
-O2 optimize for space
-O3 optimize for code size
-O4 optimize for code speed
-O5 optimize for code size and speed
-O6 optimize for code size and speed
-O7 optimize for code size and speed
-O8 optimize for code size and speed
-O9 optimize for code size and speed
-Oa optimize for code size and speed
-Ob<n> inline expansion (default n=0)
-Og enable global optimization
-Os favor code space
-Ox maximum optimizations
-O2 maximize speed
-Od disable optimizations (default)
-Oi[-] enable intrinsic functions
-Ot favor code speed
-Oy[-] enable frame pointer omission

-CODE GENERATION-
/GF enable read-only string pooling
/Gm[-] enable minimal rebuild
/GS[-] enable security checks
/GX[-] enable C++ EH (same as /EHsc)
/EHs enable C++ EH (no SEH exceptions)
/EHa enable C++ EH (w/ SEH exceptions)
```

GCC

- La GNU Compiler Collection (solitamente abbreviata in GCC) è un insieme di compilatori creato inizialmente dall'hacker americano **Richard Stallman** come parte del Sistema GNU, un sistema operativo libero compatibile con Unix.
- Le versioni recenti di GCC, sviluppate dalla comunità **Open Source**, sono incorporate nelle principali distribuzioni del sistema operativo GNU/Linux, e di molti altri sistemi basati su fondamenta Unix, come per esempio Mac OS X.
- Nata inizialmente come un compilatore per il **linguaggio C** (il nome in origine era GNU C Compiler), GCC dispone oggi di vari front end per altri linguaggi, tra cui Java, C++ , Objective_C ,Fortran e Ada, ed è in grado di generare eseguibili per molte architetture, tra le quali x86 (intel compatibili), x86 a 64 bit (AMD/intel a 64 bit), PowerPC, s390 (IBM), Sparc (Sun Microsystems).
- L'ultima versione rilasciata è la **4.3.3**

GCC homepage

[Trova nella pagina](#) [Trova successivo](#)

Modalità autore [Tutte le immagini](#) [Adatta alla larghezza](#) 120%

GCC, the GNU Compiler Collection

The GNU Compiler Collection includes front ends for C, C++, Objective-C, [Fortran](#), [Java](#), and Ada, as well as libraries for these languages ([libstdc++](#), [libgcj](#),...).

We strive to provide regular, high quality [releases](#), which we want to work well on a variety of native and cross targets (including GNU/Linux), and encourage everyone to [contribute changes](#) and [help testing](#) GCC. Our sources are readily and freely available via [SVN](#) and [weekly snapshots](#).

Major decisions about GCC are made by the [steering committee](#), guided by the [mission statement](#).

About GCC

- [Mission Statement](#)
- [Releases](#)
- [Snapshots](#)
- [Mailing lists](#)
- [Contributors](#)
- [Steering Committee](#)

Documentation

- [Installation](#)
- [Platforms](#)
- [Testing](#)
- [Manual](#)
- [FAQ](#)
- [Wiki](#)
- [Further Readings](#)

Download

- [Mirror sites](#)
- [Binaries](#)

"Live" Sources

- [SVN read access](#)
- [Rsync read access](#)
- [SVN write access](#)

Development

- [Development Plan](#)
- [Tentative Timeline](#)
- [Contributing](#)
- [Why contribute?](#)
- [Open projects](#)
- [Front ends](#)
- [Back ends](#)
- [Extensions](#)

News

January 27, 2009

The GCC Steering Committee, along with the Free Software Foundation and the Software Freedom Law Center, is pleased to announce the release of a new [GCC Runtime Library Exception](#).

This license exception has been developed to allow various GCC libraries to upgrade to GPLv3. It will also enable the development of a plugin framework for GCC. ([Rationale document and FAQ](#))

Status

Current release series: [GCC 4.3.3](#)

Status: [2009-01-24](#) (regression fixes and docs only).

[Serious regressions](#). [All regressions](#).

Previous release series: [GCC 4.2.4](#)

Status: [2008-05-19](#) (regression fixes and docs only).

[Serious regressions](#). [All regressions](#).

Active development: GCC 4.4.0 ([changes](#))

Status: [2009-02-16](#) (regression fixes and docs only).



GCC

Il gcc fornisce al programmatore un controllo esteso del processo di compilazione.

Il processo di compilazione può comprendere fino a quattro stadi distinti:

1. Fase di preprocessing
 2. Compilazione vera e propria
 3. Assemblaggio
 4. Linking
- E' possibile interrompere il processo di compilazione in uno qualsiasi di questi stadi per esaminare il risultato della compilazione.
 - Con gcc è anche possibile controllare la quantità e il tipo di informazioni di **debugging** così da includerle nel file binario prodotto.
 - Come la maggior parte dei compilatori consente di eseguire l'ottimizzazione del codice.
 - gcc è anche un ***cross compilatore***, per cui è possibile sviluppare codice utilizzando un particolare processore e farlo poi girare su un altro.

GCC comandi base

gcc

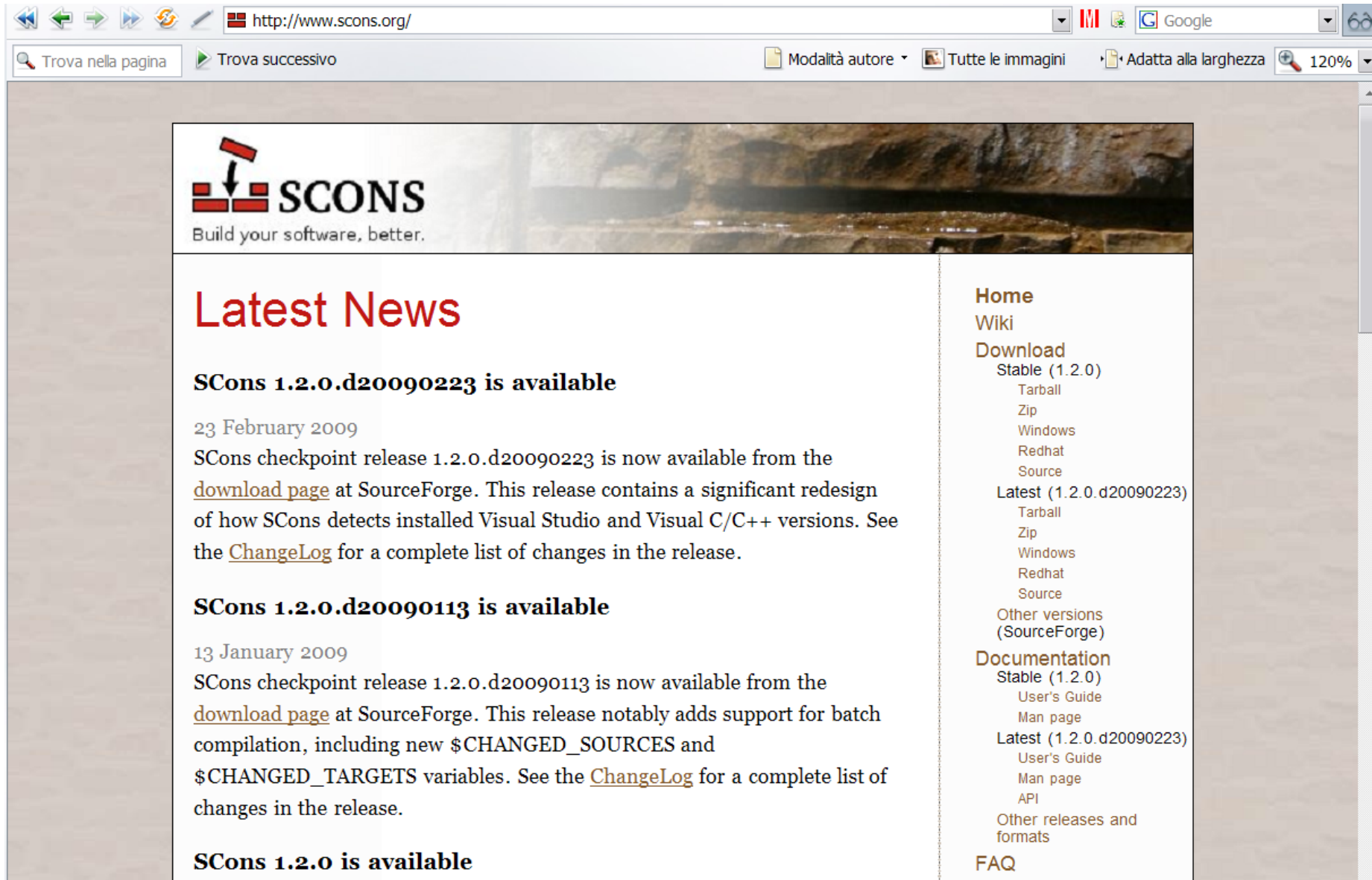
ld

ar

make

gdb

SCons – alternativa a Make



The screenshot shows a web browser window with the address bar displaying `http://www.scons.org/`. The browser's search bar contains the text "Trova nella pagina" and "Trova successivo". The page features the SCons logo, which consists of a red square with a white arrow pointing down to a red square, followed by the text "SCONS" and the tagline "Build your software, better." below it. The main content area is titled "Latest News" in red. It contains three news items, each with a date, a title, and a description. The first item is dated "23 February 2009" and titled "SCons 1.2.0.d20090223 is available". The second item is dated "13 January 2009" and titled "SCons 1.2.0.d20090113 is available". The third item is titled "SCons 1.2.0 is available". On the right side of the page, there is a sidebar with a table of contents. The sidebar is divided into two columns. The left column contains links to "Home", "Wiki", "Download", and "Documentation". The right column contains links to "Stable (1.2.0)", "Latest (1.2.0.d20090223)", "Other versions (SourceForge)", "User's Guide", "Man page", "API", and "Other releases and formats".

SCONS
Build your software, better.

Latest News

SCons 1.2.0.d20090223 is available
23 February 2009
SCons checkpoint release 1.2.0.d20090223 is now available from the [download page](#) at SourceForge. This release contains a significant redesign of how SCons detects installed Visual Studio and Visual C/C++ versions. See the [ChangeLog](#) for a complete list of changes in the release.

SCons 1.2.0.d20090113 is available
13 January 2009
SCons checkpoint release 1.2.0.d20090113 is now available from the [download page](#) at SourceForge. This release notably adds support for batch compilation, including new `$CHANGED_SOURCES` and `$CHANGED_TARGETS` variables. See the [ChangeLog](#) for a complete list of changes in the release.

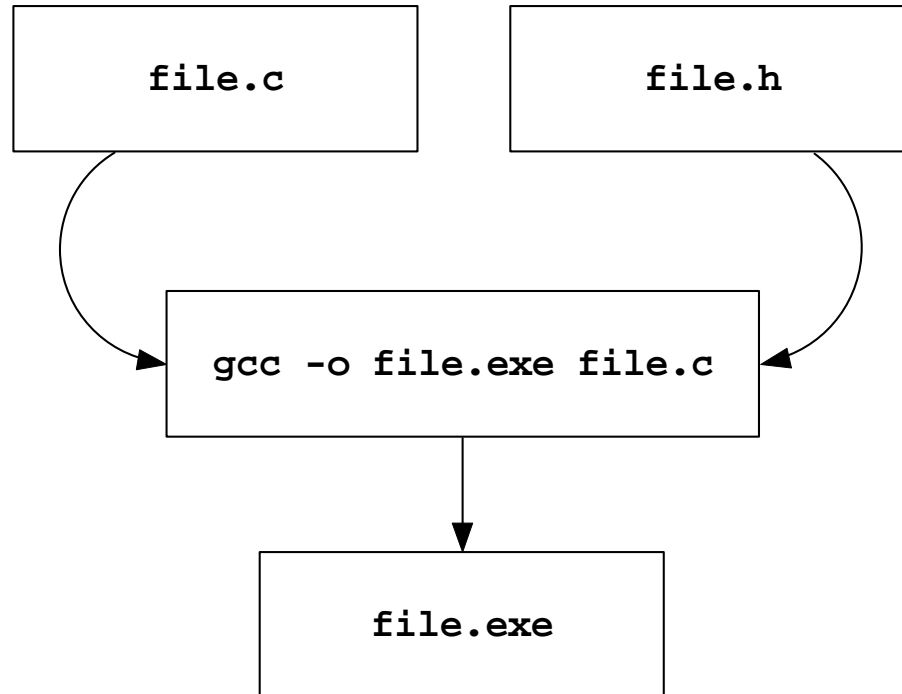
SCons 1.2.0 is available

Home
Wiki
Download
Stable (1.2.0)
Tarball
Zip
Windows
Redhat
Source
Latest (1.2.0.d20090223)
Tarball
Zip
Windows
Redhat
Source
Other versions (SourceForge)
Documentation
Stable (1.2.0)
User's Guide
Man page
Latest (1.2.0.d20090223)
User's Guide
Man page
API
Other releases and formats
FAQ

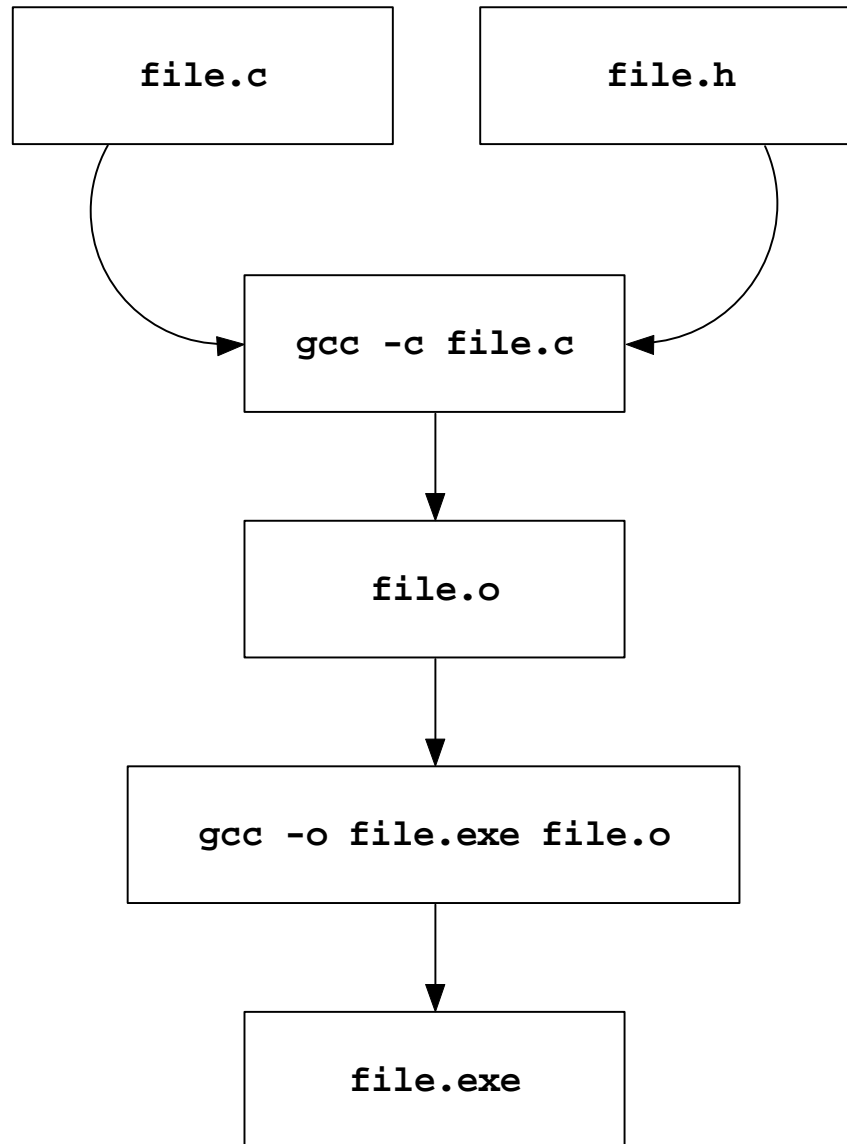
Caro buon vecchio C

```
/* first.c */  
  
#include <stdio.h>  
int main()  
{  
    printf("Hello, world\n");  
    return 0;  
}
```

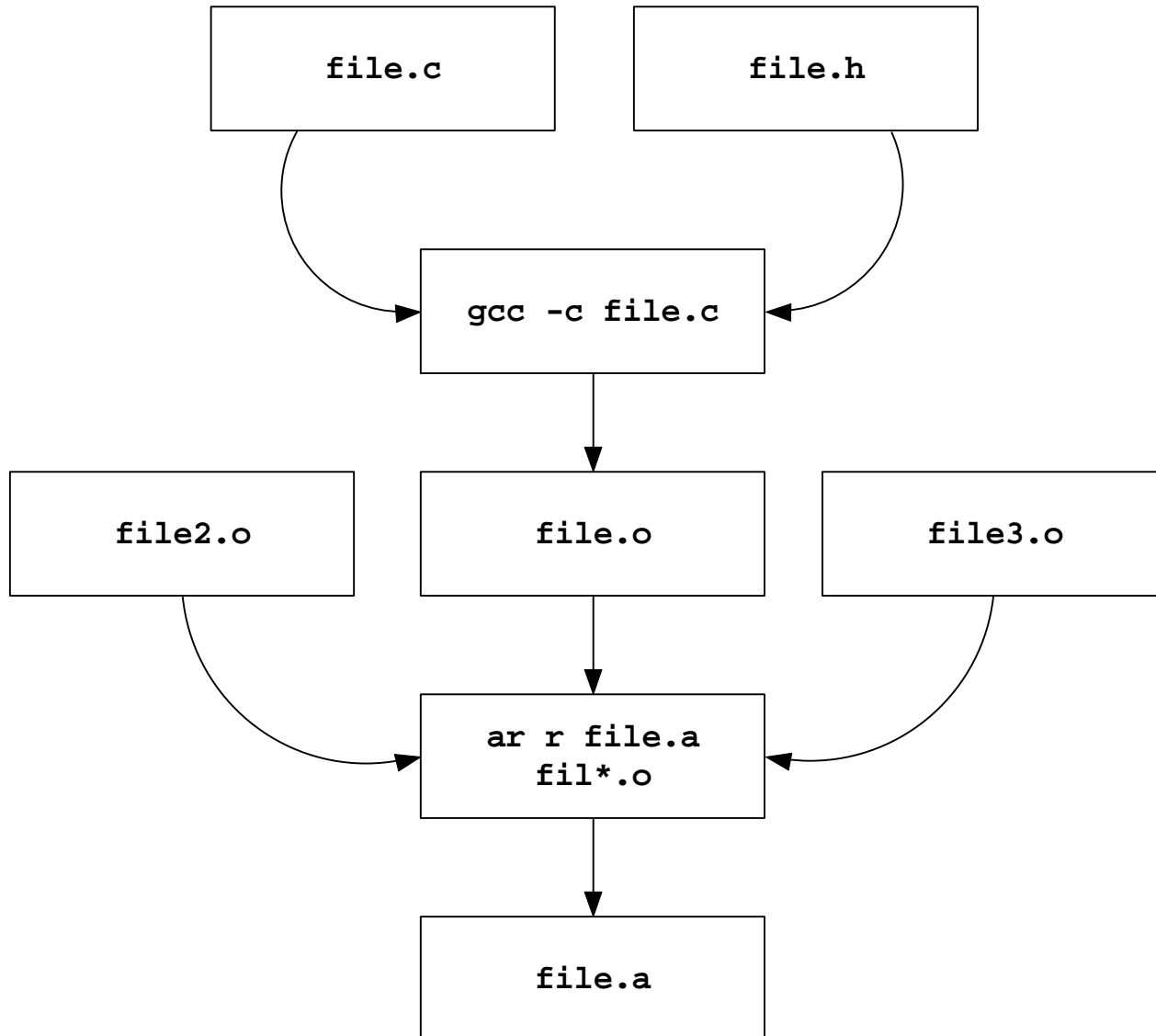
GCC compilazione



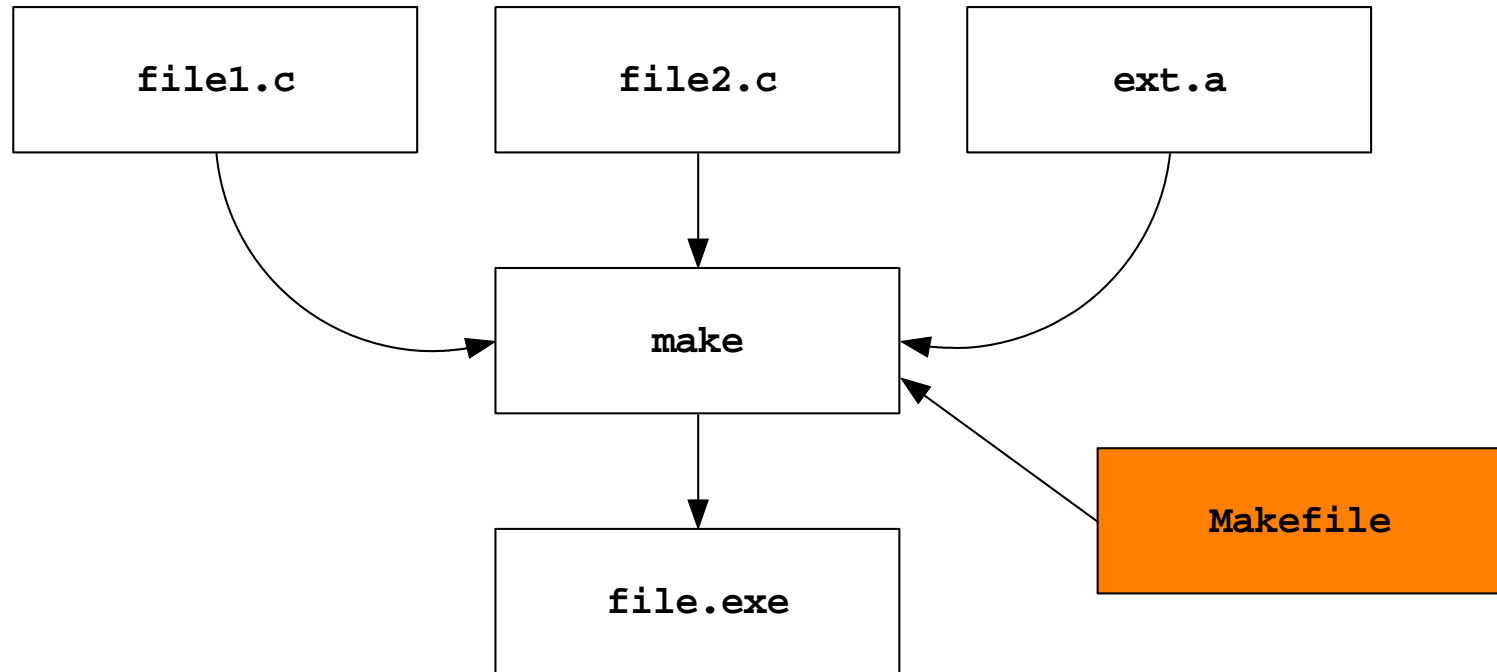
GCC linking



AR creare librerie



MAKE gestire un progetto



Un esempio di Makefile

```
# Makefile
COMPILER = gcc -Wall
LIBS = -lm
EXECUTABLE = first
OBJECT = first.o
$(EXECUTABLE): $(OBJECT)
$(COMPILER) -o $(EXECUTABLE) $(OBJECT)
$(LIBS)
%.o: %.c
$(COMPILER) -o $*.o -c $*.c
```

Un po' di esercizio

Goal:

realizzare un programma che legge a riga di comando 2 stringhe (A e B) e crea una nuova stringa C concatenando A a B: $C=AB$

Esempio:

1. Input: “capo” “ufficio”
2. Output: “capoufficio”

Tempo a disposizione:

15 minuti

Un po' di esercizio 2

Goal:

realizzare un programma che legge a riga di comando 2 numeri a virgola mobile in doppia precisione e stampa a video la loro somma

Requisiti:

1. La funzione di somma deve essere implementata in un file esterno chiamato **filext.c**
2. Con file filext.c compilato deve essere creata una libreria **functions.a**
3. Il file del main, chiamato **main.c**, deve essere compilato
4. Il main compilato e la libreria **functions.a** devono essere linkati in un eseguibile chiamato **somma.exe**
5. L'eseguibile deve funzionare!!!

Tempo a disposizione:

25 minuti

main.c

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

extern float somma(double a, double b);

int main(int argc, char *argv[])
{
    double a=0.0, b=0.0;
    if(argc!=3) exit(-1);

    a=(double)atof(argv[1]);
    b=(double)atof(argv[2]);
    printf("\n%2.3f + %2.3f = %2.3f\n", a, b, somma(a, b));

    return 0;
}
```

filext.c

```
doublet somma(double a, double b)
{
return(a+b);
}
```

sequenza comandi

```
gcc -Wall -c filext.c
```

```
gcc -Wall -c main.c
```

```
ar r functions.a filext.o
```

```
gcc -o somma.exe main.o functions.a
```