GCC 3.3.x
Vs
.Net2003

Speaker: Dr. Matteo Roffilli
Ph.D student in Computer Science

roffilli@csr.unibo.it
Introduction

- Actually best language?
- C/C++ is used in about 50% of projects
- Support x86, x86_64, MMX, SSE, SSE2, SSE3
- … and clones
- Embedded systems: StrongArm, PXA, …
Popular compilers

- Borland C/C++ 5.6
- Digital Mars C/C++ 8.34
- GNU C/C++ 3.3.x
- Intel C/C++ 7.0
- Metrowerks CodeWarrior 8.3
- Microsoft Visual C++ 6.0
- Microsoft Visual C++ .NET2003
- Watcom C/C++ 12.0
Welcome to the GCC home page!

GCC is the GNU Compiler Collection, which currently contains front ends for C, C++, Objective-C, Fortran, Java, and Ada, as well as libraries for these languages (libstdc++, libgcj,...). Further frontends are available.

Major decisions about GCC are made by the steering committee, guided by the mission statement.

We encourage everyone to contribute changes and help testing GCC, and we provide access to our development sources with anonymous CVS and weekly snapshots.

We strive to provide regular, high quality releases, which we want to work well on a variety of native (including GNU/Linux) and cross targets. To that end, we use an extensive test suite and automated regression testers as well as various benchmark suites and automated testers to maintain and improve quality.

Current release series: GCC 3.3.2 (released 2003-10-17)
   Branch status: 2003-10-29 (open for regression fixes).

Previous release series: GCC 3.2.3 (released 2003-04-25)
   This is the last release from the 3.2.x series, the branch has been closed after the release.

Active development (mainline): will become GCC 3.4 (current changes)
   Stage 3; open for all maintainers.

News/Announcements

October 18, 2003
    Bernardo Innocenti of Develer S.r.l. has contributed the m68k-uclinux target and improved support for ColdFire cores, based on former work by Paul Dale (SnapGear, Inc.) and Peter Barada (Motorola, Inc.).

October 17, 2003
    GCC 3.3.2 has been released.
What we are looking for?

- Compilation Time
- Speed of generated code
- Size of generated code
- Language support
- Features
- Developing tools
- Automatic Vectorization with SWAR
- Expense
How testing? KNOPPPIX!
Closed because of "Software-Patents"

In the next few days, the European Parliament will decide about the legalisation and adoption of so-called "software patents" in Europe, which are already used by large companies in other countries to put competitors out of business. This can lead to the termination of many software projects such as KNOPPIX, at least within Europe, because the holders of the over 30,000 already granted "software patents" (currently without a legal foundation) can claim exclusive rights and collect license fees for trivial things like "progress bars", "mouseclicks on online order forms", "scrolling within a window" and similar. That way, software developers will have to pay the "software-patentholders" for using these features, even in their own, completely self-developed applications, which can completely stall the development of innovative software for small and medium companies. Apart from this, the expense for patent inquiries and legal assistance is high, for even trying to find out if the self-developed software is possibly violating "software-patents", if you want to continue to market your software. Contrary to real patents, "software-patents" are, in the current draft, monopolization of business ideas and methods, even without any tangible technical implementation.

More about the current major problem at http://swpat.ffii.org/index.en.html

© KNOPPER.NET 2003
### Compilation Time

Some test results…

Compilation Time in *ms*

<table>
<thead>
<tr>
<th>Source:</th>
<th>GCC</th>
<th>.NET</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>1368</td>
<td>243</td>
</tr>
<tr>
<td>C2</td>
<td>886</td>
<td>1640</td>
</tr>
<tr>
<td>C3</td>
<td>182</td>
<td>336</td>
</tr>
<tr>
<td>Pch</td>
<td>39234</td>
<td>3343</td>
</tr>
<tr>
<td>whereis</td>
<td>3261</td>
<td>1523</td>
</tr>
<tr>
<td>zlib</td>
<td>4890</td>
<td>2703</td>
</tr>
</tbody>
</table>

*Source: Dr. Dobb’s Journal Oct 2003*
### Speed of generated code

#### Some test results…

#### Execution Time in ms

<table>
<thead>
<tr>
<th></th>
<th>GCC</th>
<th>.NET</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>7714</td>
<td>7813</td>
</tr>
<tr>
<td>T2</td>
<td>1679</td>
<td>1843</td>
</tr>
<tr>
<td>T3</td>
<td>6809</td>
<td>2341</td>
</tr>
<tr>
<td>T4</td>
<td>385</td>
<td>406</td>
</tr>
<tr>
<td>T5</td>
<td>1590</td>
<td>887</td>
</tr>
<tr>
<td>zlib</td>
<td>11310</td>
<td>9117</td>
</tr>
</tbody>
</table>

**Source:**
Dr. Dobb’s Journal Oct 2003
## Size of generated code

**Some test results…**

**Size of code in Kbytes**

**.NET outperforms GCC!**

**Source:**
*Dr. Dobb’s Journal Oct 2003*

<table>
<thead>
<tr>
<th>Code Type</th>
<th>GCC</th>
<th>.NET</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>98</td>
<td>45</td>
</tr>
<tr>
<td>T2</td>
<td>95</td>
<td>40</td>
</tr>
<tr>
<td>T3</td>
<td>95</td>
<td>40</td>
</tr>
<tr>
<td>T4</td>
<td>458</td>
<td>53</td>
</tr>
<tr>
<td>T5</td>
<td>453</td>
<td>49</td>
</tr>
<tr>
<td>zlib</td>
<td>151</td>
<td>102</td>
</tr>
</tbody>
</table>
Language support

- GCC has the BEST language support.
- GCC supports AT&T and Intel Assembler syntax.
- .NET supports MS proprietary extensions:
  - MFC (Microsoft Foundation Class)
  - ATL (Active Template Library)
  - WTL (Windows Template Library)
  - Managed C++
Language support 2

A key question:
Do you need Microsoft Extensions

Consider:
SDL, wxWindows, QT
OpenGL, OpenRM, OpenSceneGraph, Demeter, ...
Automatic Vectorization .NET

.NET uses natively MMX,SSE,SSE2

In our project .NET with vectorization has gained a speedup of 2x over VS6+sp5 and GCC 3.3.x

try:-nologo -W4 -wd4244 -wd4018 -EHsc
-G7 -Ox -arch:SSE2

A bunch of error when moving C++ from VS6 to .NET2003 !!!

Very less with GCC 2.95.2 to GCC 3.3.x
Automatic Vectorization GCC

GCC 3.3.x declares to use vectorization but... we were not able to see difference!!!

Homeover try:


- ADVFLAGS = -march=pentium4 -mcpu=pentium4 -mmmx -msse -msse2 -mfpmath=sse -momit-leaf-frame-pointer -minline-all-stringops -mno-push-args -maccumulate-outgoing-args -m128bit-long-double
Online Resource

- DrDobb’s Journal [www.ddj.com](http://www.ddj.com)
- FSF and GCC homepage
- Microsoft homepage, MSDN
- SPRITE Linux Day Proceeding
Neural Network Lab

Chief
Prof. Renato Campanini

Ph.D student
Dr. Matteo Roffilli

Graduated Students
Dr. Enrico Angelini
Dr. Cristiano Campeggiani
Dr. Omar Schiaratura

Undergraduated Students
Daniele Quercioli
Erich Zangheri
Christian Zoffoli

wwfn.csr.unibo.it
Medical Imaging Group

**Full Professors**
Prof. Dante Bollini

**Associate Professors**
Prof. Renato Campanini

**PhD Students**
Dr. Danilo Dongiovanni
Dr. Mirko Gombia
Dr. Nico Lanconelli
Dr. Matteo Masotti
Dr. Giuseppe Palermo
Dr. Alessandro Riccardi
Dr. Matteo Roffilli

**Fellows**
Dr. Emiro Iampieri

www.bo.infn.it/mig
THANKS FOR YOUR ATTENTION!

Any question?

Speaker: Dr. Matteo Roffilli
roffilli@csr.unibo.it