

Jak to začalo?

oo

What's GNU? Gnu's Not Unix!

ooooooooooooooo

AMD64 port a SUSE

oooooooooooo

GCC dnes

ooooo

GCC

Jan Hubička

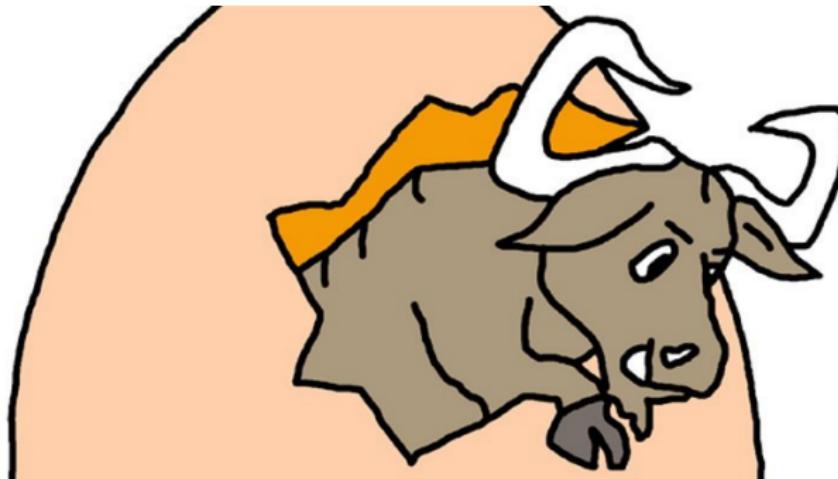
Katedra aplikované matematiky

Překladače 2021

Co je GCC

GNU Compiler collection:

- Překladač C, C++, obj-C, obj-C++, Fortranu, Ady, Go,...
- Podporující více než 50 procesorů a několik set targetů (kombinací procesor+OS)
- Vyvjený od 80. let jako jeden ze základních balíků GNU projektu
- První verze vydaná v roce 1987
- 10m řádek, cca 100-150 aktivních vývojářů, odhadem 2000 člověkolet



Richard Mathew Stallman (RMS) vs Xerox 9700



Jak to začalo?



What's GNU? Gnu's Not Unix!



AMD64 port a SUSE



GCC dnes



Richard Mathew Stallman (RMS) vs Xerox 9700



Richard Mathew Stallman (RMS) vs Xerox 9700



Hacker na MIT od 1971. V září 1983 oznamuje GNU operating system. 1984 ochází a věnuje se GNU projektu. Začal vyvíjet základní nástroje Emacs, GCC, GDB, make, ...

23. září 1983: RMS oznamuje GNU projekt

Starting this Thanksgiving I am going to write a complete Unix-compatible software system called GNU (for Gnu's Not Unix), and give it away free to everyone who can use it. Contributions of time, money, programs and equipment are greatly needed.

To begin with, GNU will be a kernel plus all the utilities needed to write and run C programs: editor, shell, C compiler, linker, assembler, and a few other things. After this we will add a text formatter, a YACC, an Empire game, a spreadsheet, and hundreds of other things. We hope to supply, eventually, everything useful that normally comes with a Unix system, and anything else useful, including on-line and hardcopy documentation.

GNU will be able to run Unix programs, but will not be identical to Unix. We will make all improvements that are convenient, based on our experience with other operating systems. In particular, we plan to have longer filenames, file version numbers, a crashproof file system, filename completion perhaps, terminal-independent display support, and eventually a Lisp-based window system through which several Lisp programs and ordinary Unix programs can share a screen. Both C and Lisp will be available as system programming languages. We will have network software based on MIT's chaosnet protocol, far superior to UUCP. We may also have something compatible with UUCP. [...]

I consider that the golden rule requires that if I like a program I must share it with other people who like it. I cannot in good conscience sign a nondisclosure agreement or a software license agreement.



- 1976: První Emacs
- 1984: GNU Emacs distribuovaný včetně zdrojáků (za 150 dolarů)
- 1985: založena Free Software Foundation, GNU Manifesto, Emacs General public licence
- 1987: GNU C compiler (GCC), 110k řádek
- 1989: GNU General Public Licence (GPL)
- 1991: Linux



22. března 1983: RMS vydává GCC 0.9

Date: Sun, 22 Mar 87 10:56:56 EST

From: rms (Richard M. Stallman)

The GNU C compiler is now available for ftp from the file /u2/emacs/gcc.tar on prep.ai.mit.edu. This includes machine descriptions for vax and sun, 60 pages of documentation on writing machine descriptions (internals.texinfo, internals.dvi and Info file internals).

This also contains the ANSI standard (Nov 86) C preprocessor and 30 pages of reference manual for it.

This compiler compiles itself correctly on the 68020 and did so recently on the vax. It recently compiled Emacs correctly on the 68020, and has also compiled tex-in-C and Kyoto Common Lisp. However, it probably still has numerous bugs that I hope you will find for me.

I will be away for a month, so bugs reported now will not be handled until then.

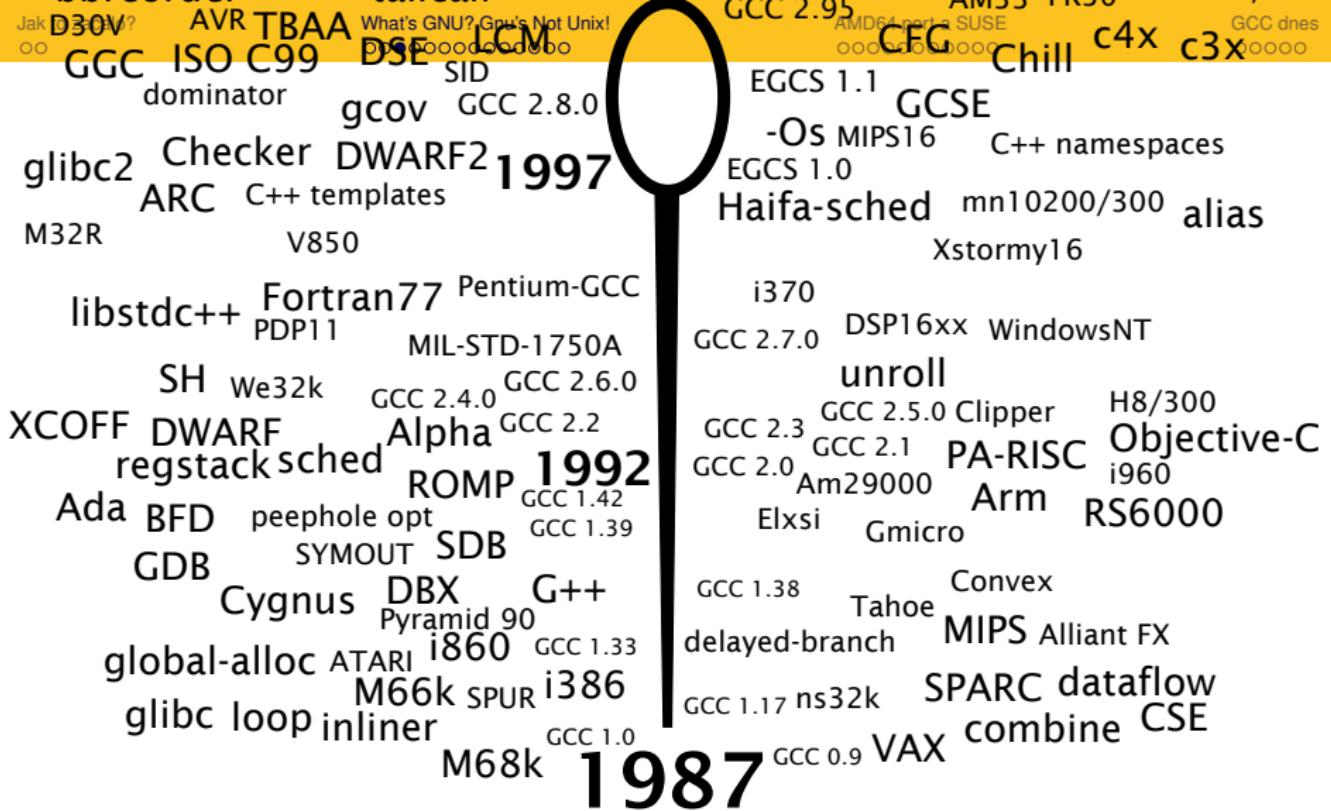
If you can't ftp, you can order a compiler beta-test tape from the Free Software Foundation for \$150 (plus 5% sales tax in Massachusetts, or plus \$15 overseas if you want air mail).

Free Software Foundation
1000 Mass Ave
Cambridge, MA 02138

GNU Compiler Collection 7.1 2017 GNU Tools Cauldron, Prague

store merging BRIG RISC-V
 HSA NPVTX DWARFS lpa-vrp
 OpenACC 2.0a C++ Concepts
 Fortran2015 C++14 **GCC 5.1**
 NDS32 Nios II IPA-ICF Auto-FDO
 IPA-devirt function reordering CilkPlus
 C++11 **Aarch64** **GCC 4.8.0**
 MSP430 partial-PRE DWARF4 -Og **2012**
OpenMP 3.1 C6X Tile-GX/Pro
 RL78 Ada 2012 Objective-C 2.0
 gimple-fold MeP **WHOPR** GCC 4.5.0
 LTO
Graphite Fortran2008 LatticeMicro32
 optimize attribute ACML **GCC 4.3.0**
 early inliner
 dataflow branch IPA-SSA **2007**
 Java 1.5 C++0x **OpenMP**
 sel-sched PHI-opt VRP reassoc
 Morphosys S/390 **GCC 4.0.0**
 vectorizer DSE SMS **Fortran95**
 new C++ parser trace scheduling
PCH profilebootstrap **GCC 3.3**
DFA scheduler **GCC Summit**
 Visium Ada95 MMIX CRIS **2002**
 FDO SHS XScale **GCC 3.0**
 CGEN
 bbreorder DCE tailcall IA-64
 D30V AVR TBAA DSE LCM
 GGC ISO C99 DSE SID
 dominator gcov **GCC 2.8.0**
 glibc2 Checker DWARF2 **1997**
 ARC C++ templates
 M32R V850
 libstdc++ Fortran77 Pentium-GCC
 PDP11 MIL-STD-1750A
 SH We32k **GCC 2.4.0** **GCC 2.6.0**
XCOFF DWARF Alpha **GCC 2.2**
 regstack sched **1992**
 Ada BFD peephole opt ROMP **GCC 1.42**
 SYMOUT SDB **GCC 1.39**
 GDB Cygnus DBX G++
 Pyramid 90 **GCC 1.33**
 global-alloc ATARI I860 **M68k SPUR**
 glibc loop inliner **M68k** **i386**
M68k **1987** **GCC 0.9** VAX

GCC 6.1 IPA-bit-CP spell-check
 C++17 diagnostic ranges
 OpenACC libgccjit MPX
GCC 4.9.0 color diags OpenMP 4.0
 FT32 ubsan
 LRA ThreadSanitizer NiosII
 GCC 4.7.0 NDS32 AddressSanitizer
GCC 4.6.0 CR16 **GNU Cauldron**
GCC 4.6.0 C1X Epiphany
 Microblaze Go
 IPA-SRA -Ofast partial inlining
 RX
 GCC 4.4.0 software pipelining picochip
 Moxie call-DCE IRA SPU Moxie
 Fortran2003 Gold Renesas RX
 GCC 4.2.0 Forwprop
GIMPLE tuples
 GCC 4.1.0 SCCVN IP-CP Blackfin
 IPA-pure-const IPA structalias
 SRA M32C tail-recuse SCCP
 GCC 3.4.0 ivopts SCEV mudflap
 unit-at-a-time IPA inliner CFG loop
 GCC 3.2 TREE-SSA tracer webizer
 GCC 3.1 IQ2000 PTA
 x86-64 MCore210/240 SSA-PRE
 libcpp Xtensa regrename Java
 GCC 2.95 SSA AM33 FR30 68HC11/2
 CFG Chill c4x c3x
 EGCS 1.1 GCSE
 -Os MIPS16 C++ namespaces
 EGCS 1.0 Haifa-sched mn10200/300 alias
 Xstormy16
 i370
 GCC 2.7.0 DSP16xx WindowsNT
 unroll
 GCC 2.3 GCC 2.5.0 Clipper H8/300
 GCC 2.0 PA-RISC Objective-C
 Am29000 Arm RS6000
 Elxi Gmicro
 GCC 1.38 Convex
 Tahoe MIPS Alliant FX
 delayed-branch
 GCC 1.17 ns32k SPARC dataflow
 combine CSE



RMS o GCC

Shortly before beginning the GNU project, I heard about the Free University Compiler Kit, also known as VUCK. (The Dutch word for "free" is written with a V.) This was a compiler designed to handle multiple languages, including C and Pascal, and to support multiple target machines. I wrote to its author asking if GNU could use it.

He responded derisively, stating that the university was free but the compiler was not. I therefore decided that my first program for the GNU project would be a multi-language, multi-platform compiler.

Hoping to avoid the need to write the whole compiler myself, I obtained the source code for the Pastel compiler, which was a multi-platform compiler developed at Lawrence Livermore Lab. It supported, and was written in, an extended version of Pascal, designed to be a system-programming language. I added a C front end, and began porting it to the Motorola 68000 computer. But I had to give that up when I discovered that the compiler needed many megabytes of stack space, and the available 68000 Unix system would only allow 64k.

I then realized that the Pastel compiler functioned by parsing the entire input file into a syntax tree, converting the whole syntax tree into a chain of "instructions", and then generating the whole output file, without ever freeing any storage. At this point, I concluded I would have to write a new compiler from scratch. That new compiler is now known as GCC...



1980

1990

2000

2010

Cygnus solutions: Cygnus your GNU support



John Gilmore, Michael Tiemann, David Henkel-Wallace

Cygnus solutions: Cygnus your GNU support



John Gilmore, Michael Tiemann, David Henkel-Wallace

Založena 1989 jako první plně free softwarová firma. Portovala GCC a GNU Toolchain na víc než 170 platforem. Za prvních 10 let investovala 250 mil. dolarů do GCC.

Roku 1999 mergovala s Red Hatem. V té době měla 150 vývojářů (výrazně více, než Red Hat).

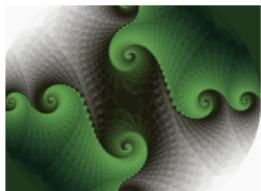
Moje setkání s GCC



V roce 1991 rodiče koupili Unixovou wosktation HP Apollo 425e
(Motorola 68040, 16MB ram, 1280x1024 256barev, 200MB SCSI disk)

HP-UX měl pouze překladač K&R C (pro komplikaci kernelu) a vývojový balík byl drahý. Rozhodl jsem se portovat GCC z Amigy.

První kontakt s GCC vývojáři



Proč jsem začal hackovat GCC?

- Dj Deorie portoval GCC na MS-DOS
(GCC generovalo lepší kód než Borland/Watcom)
- Demoscéna: chtěl jsem ukázat, že s GCC nemusím psát v assembleru
- Martin Mareš mi řekl, že GCC nerozumí.



První kontakt s GCC vývojáři

Jak to šlo:

- 1995 jsem poslal první patch go GCC ML.
- 1996 jsem dostal odpověď od Richarda Kennera (mantanera GCC 2) že musím napřed podepsat copyright assignment s FSF
- Napsal jsem copyright, podepsal se Stallmanem. Dostal nálepku a tričko.

První kontakt s GCC vývojáři

Jak to šlo:

- 1995 jsem poslal první patch go GCC ML.
- 1996 jsem dostal odpověď od Richarda Kennera (mantanera GCC 2) že musím napřed podepsat copyright assignment s FSF
- Napsal jsem copyright, podepsal se Stallmanem. Dostal nálepku a tričko.
- koncem roku 1996 jsem dostal odpověď od Richarda Kennera že můj patch je pravděpodobně obsoletní a dostal heslo k FTP pro tajné aktuální zdrojáky GCC

První kontakt s GCC vývojáři

Jak to šlo:

- 1995 jsem poslal první patch go GCC ML.
- 1996 jsem dostal odpověď od Richarda Kennera (mantanera GCC 2) že musím napřed podepsat copyright assignment s FSF
- Napsal jsem copyright, podepsal se Stallmanem. Dostal nálepku a tričko.
- koncem roku 1996 jsem dostal odpověď od Richarda Kennera že můj patch je pravděpodobně obsoletní a dostal heslo k FTP pro tajné aktuální zdrojáky GCC
- patch byl stále aktuální a posla jsem ho zpět

První kontakt s GCC vývojáři

Jak to šlo:

- 1995 jsem poslal první patch go GCC ML.
- 1996 jsem dostal odpověď od Richarda Kennera (mantanera GCC 2) že musím napřed podepsat copyright assignment s FSF
- Napsal jsem copyright, podepsal se Stallmanem. Dostal nálepku a tričko.
- koncem roku 1996 jsem dostal odpověď od Richarda Kennera že můj patch je pravděpodobně obsoletní a dostal heslo k FTP pro tajné aktuální zdrojáky GCC
- patch byl stále aktuální a posla jsem ho zpět
- Richard Kenner reagoval tím, že chybí dokumentace

První kontakt s GCC vývojáři

Jak to šlo:

- 1995 jsem poslal první patch go GCC ML.
- 1996 jsem dostal odpověď od Richarda Kennera (mantanera GCC 2) že musím napřed podepsat copyright assignment s FSF
- Napsal jsem copyright, podepsal se Stallmanem. Dostal nálepku a tričko.
- koncem roku 1996 jsem dostal odpověď od Richarda Kennera že můj patch je pravděpodobně obsoletní a dostal heslo k FTP pro tajné aktuální zdrojáky GCC
- patch byl stále aktuální a posla jsem ho zpět
- Richard Kenner reagoval tím, že chybí dokumentace
- Nevěděl jsem, co dokumentace je. Ukázalo se, že ChangeLog

První kontakt s GCC vývojáři

Jak to šlo:

- 1995 jsem poslal první patch go GCC ML.
- 1996 jsem dostal odpověď od Richarda Kennera (mantanera GCC 2) že musím napřed podepsat copyright assignment s FSF
- Napsal jsem copyright, podepsal se Stallmanem. Dostal nálepku a tričko.
- koncem roku 1996 jsem dostal odpověď od Richarda Kennera že můj patch je pravděpodobně obsoletní a dostal heslo k FTP pro tajné aktuální zdrojáky GCC
- patch byl stále aktuální a posla jsem ho zpět
- Richard Kenner reagoval tím, že chybí dokumentace
- Nevěděl jsem, co dokumentace je. Ukázalo se, že ChangeLog
- Roku 1997 jsem dostal první recenzi od Jeffa Law. Podstatný bod: za tečkou se píšou dvě mezery.

Jak to začalo?

oo

What's GNU? Gnu's Not Unix!

oooooooo●ooooo

AMD64 port a SUSE

oooooooooooo

GCC dnes

ooooo



Stallman postupně předává projekty kamarádům a věnuje se politice



Stallman postupně předává projekty kamarádům a věnuje se politice



Richard Kenner, maintainer GCC 2.x

Po vydání GCC 2.7.2 rozhoduje, že další verze bude po dokončení plné podpory C++



Stallman postupně předává projekty kamarádům a věnuje se politice



Richard Kenner, maintainer GCC 2.x

Po vydání GCC 2.7.2 rozhoduje, že další verze bude po dokončení plné podpory C++

GNU FORTRAN



Forky GCC — Pentium gcc, Fortran, atd.

1997: EGCS Extended/Experimental GNU Compiler system

From: gumby@cygnus.com (D.V. Henkel-Wallace)
Subject: A new project to merge the existing GCC forks
Date: Fri, 15 Aug 1997 16:31:29 -0700
To: egcs@egcs.cygnus.com

A bunch of us (including Fortran, Linux, Intel and RTEMS hackers) have decided to start a more experimental development project, just like Cygnus and the FSF started the gcc2 project about 6 years ago. Only this time the net community with which we are working is larger! We are calling this project 'egcs' (pronounced 'eggs').

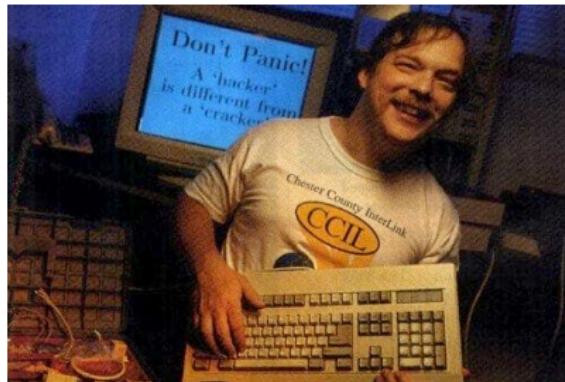
Why are we doing this? It's become increasingly clear in the course of hacking events that the FSF's needs for gcc2 are at odds with the objectives of many in the community who have done lots of hacking and improvement over the years. GCC is part of the FSF's publicity for the GNU project, as well as being the GNU system's compiler, so stability is paramount for them. On the other hand, Cygnus, the Linux folks, the pgcc folks, the Fortran folks and many others have done development work which has not yet gone into the GCC2 tree despite years of efforts to make it possible.

This situation has resulted in a lot of strong words on the gcc2 mailing list which really is a shame since at the heart we all want the same thing: the continued success of gcc, the FSF, and Free Software in general. Apart from ill will, this is leading to great divergence which is increasingly making it harder for us all to work together – It is almost as if we each had a proprietary compiler! Thus we are merging our efforts, building something that won't damage the stability of gcc2, so that we can have the best of both worlds.

...

Come join us! David Henkel-Wallace

1997: The Cathedral and the Bazaar



Eric Steven Raymond

[EGCS] We now have history on a project that, in several ways, may provide a more indicative test of the bazaar premise than fetchmail; EGCS (<http://egcs.cygnus.com/>), the Experimental GNU Compiler System.

This project was announced in mid-August of 1997 as a conscious attempt to apply the ideas in the early public versions of “The Cathedral and the Bazaar”. The project founders felt that the development of GCC, the Gnu C Compiler, had been stagnating. For about twenty months afterwards, GCC and EGCS continued as parallel products – both drawing from the same Internet developer population, both starting from the same GCC source base, both using pretty much the same Unix toolsets and development environment. The projects differed only in that EGCS consciously tried to apply the bazaar tactics I have previously described, while GCC retained a more cathedral-like organization with a closed developer group and infrequent releases.

Steering committee



- Na fotce: David Edelsohn (IBM), Ramana Radhakrishnan (ARM), Gerald Pfeifer (SUSE), Jim Wilson (SiFive) Ian Lance Taylor (Google), Jason Merrill (Red Hat)
- Nedorazili: Toon Moene (Koninklijk Nederlands Meteorologisch Instituut), Kaveh R. Ghazi, Jeffrey A. Law (Red Hat), Marc Lehmann (nethype GmbH), David Miller (Red Hat), Joseph Myers (CodeSourcery / Mentor Graphics), Joel Sherrill (OAR Corporation),

Technické zázemí



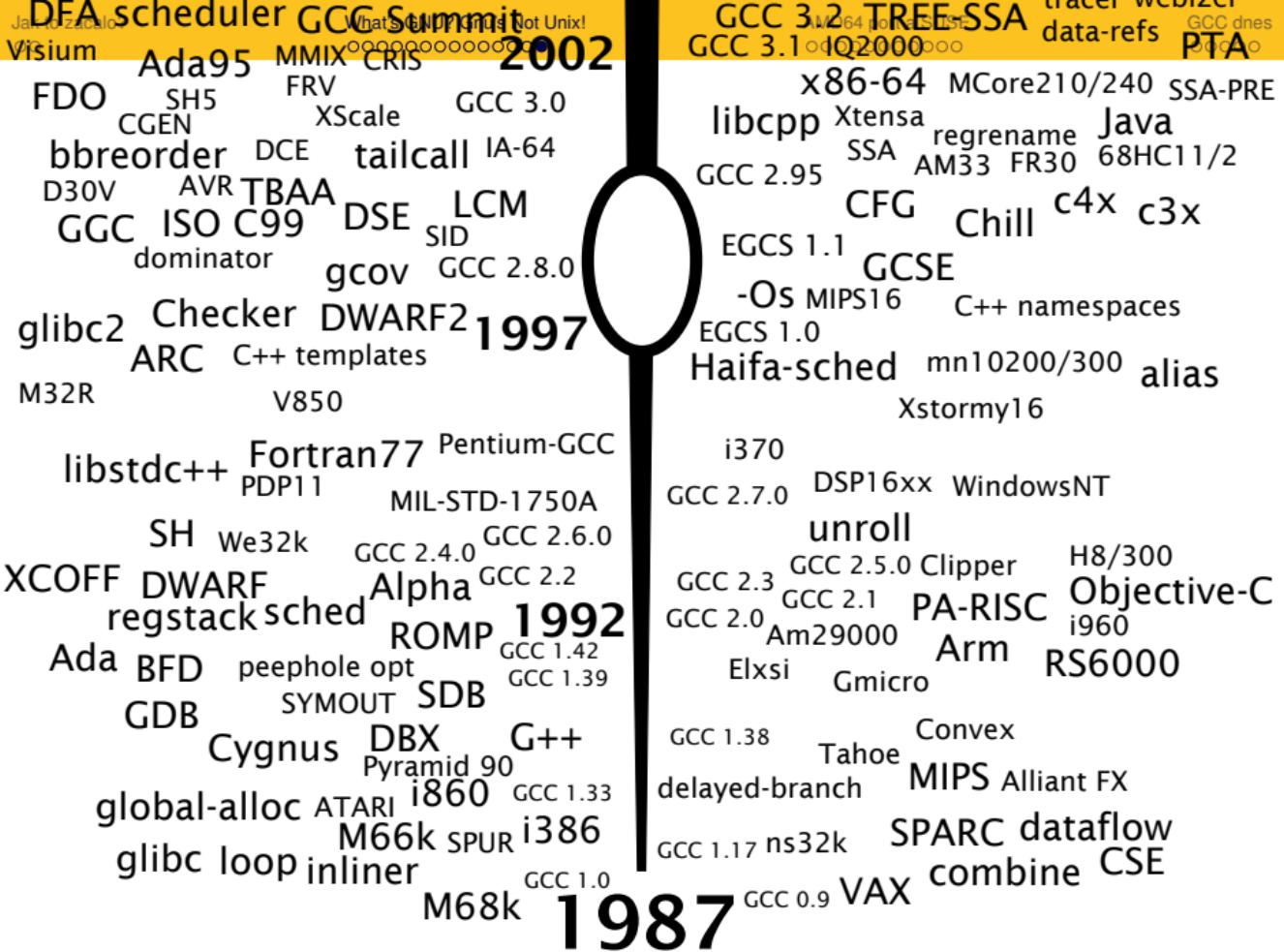
- ➊ Release managers: Richard Biener, Jakub Jelínek
Dříve: Jeff Law, Mark Mitchell, Joseph Myers, Bernd Schmit

Technické zázemí



- ① Release managers: Richard Biener, Jakub Jelínek
Dříve: Jeff Law, Mark Mitchell, Joseph Myers, Bernd Schmit
- ② Global reviewers: Richard Earnshaw, Richard Biener, Jakub Jelinek, Richard Kenner, Jeff Law, Michael Meissner, Jason Merrill, David S. Miller, Joseph Myers, Bernd Schmidt, Ian Lance Taylor, Jim Wilson
- ③ Maintainers: 116 dalších vývojářů

CVS/SVN/Git repozitář, otevřené emailové konference, wiki, aktivní IRC kanál



Spolupráce s AMD

From: Jan Hubicka [SMTP:hubicka@atrey.karlin.mff.cuni.cz]

Sent: Monday, July 12, 1999 11:06 AM

Subject: K6 and K7 specific optimization for GNU C compiler

Dear sir,

Recently the Cygnus solutions sponsored by Intel has donated major rewrite of Intel IA-32 backend for GNU Compiler. (free open-source compiler used as primary one for Linux, FreeBSD and BeOS platforms and ported to many others). See
<http://egcs.cygnus.com>

...

As interesting side-effect, the new backend got clean, extensible design making it much easier to tune for new CPUs. This week I've done retuning of this backed for AMD-K6 CPU. ... after first passrought machine description file, AMD-K6 specific optimizations seems to bring noticeable speed improvements. Sensible small graphics loops are accelerated many times mainly because of movzx acceleration, decoder scheduling and alignment issues. Average code is approx. 10% faster.

...

Thats why I would like to ask AMD to donate me the technical documentation and one K7 chip with motherboard for neccesary testing. Then I would make K7 chip support in two months and include it into development tree of gcc2.96.0 that can be released this year. Development versions are publically available and many users use them even before official release. So in advance to optimized motherboards and BIOS, K7 can have an optimized compiler.

Thank you very much,

Jan Hubicka (Student of computer science on Charles University, Prague)

From: "Alex Dreyzen (AMD)"<v-alexdr@microsoft.com>
To: "hubicka@atrey.karlin.mff.cuni.cz" <hubicka@atrey.karlin.mff.cuni.cz>
Cc: "Wayne Meretsky (E-mail)"<wayne.meretsky@amd.com>
Subject: RE: gcc beta survey
Date: Thu, 20 Jan 2000 22:56:44 -0800

Hi Jan,

I am a compiler developer at AMD. Currently I work at Microsoft assisting the MSVC team with AMD related tasks.

The gcc work that you are describing sounds promising. We are very interested in improving gcc generated code performance on Athlon. One of the most important benchmarks for us is SPEC2000 (www.spec.org)

On the spec web site you can find the SPECint2000 results submitted for PIII 733. These numbers were achieved with Intel 4.5 compiler which is not released yet. We would be very interested in working with you to try to approach or exceed these numbers using gcc on a similarly configured Athlon system.

< donated major rewrite of Intel IA-32 backend for GNU Compiler>

I would like to learn more about the gcc ia-32 backend you are describing. I would appreciate if you can provide me some links and news group names. I browsed the <http://egcs.cygnus.com> link that you referred to but could not find any related documentation or discussions.

I would also like to look at some of your code patches for k6-2 specific optimizations. Are they available for download?

If you are interested in optimizing gcc for Athlon SPEC2000 we could arrange to send you a k7 system. You can find Athlon optimization guide at <http://www.amd.com/products/cpg/athlon/techdocs/pdf/22007.pdf>

Alex Dreyzen

From: alex.dreyzen@amd.com
To: hubicka@atrey.karlin.mff.cuni.cz
Subject: Summer Work
Date: Thu, 30 Mar 2000 18:33:53 -0600

Honza,

I talked to Fred Weber today. He is the Sledgehammer (k8) architect. We need someone to help with implementing x86-64 extensions in gnu tools (assembler, linker, gcc). AMD has an office in Redmond and gnu work can be done here.

Would you consider coming to the US (Redmond, WA) to do some gnu work as a summer student, and possibly, if you like it here becoming an AMD employee, shortly after that?

What do you think?

Alex Dreyzen

From: alex.dreyzen@amd.com
To: hubicka@atrey.karlin.mff.cuni.cz
Subject: Summer Work
Date: Thu, 30 Mar 2000 18:33:53 -0600

Honza,

I talked to Fred Weber today. He is the Sledgehammer (k8) architect. We need someone to help with implementing x86-64 extensions in gnu tools (assembler, linker, gcc). AMD has an office in Redmond and gnu work can be done here.

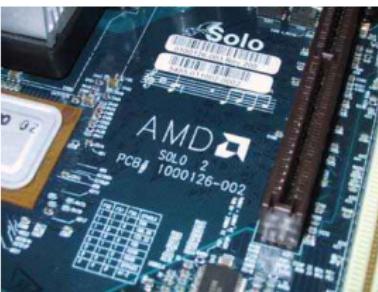
Would you consider coming to the US (Redmond, WA) to do some gnu work as a summer student, and possibly, if you like it here becoming an AMD employee, shortly after that?

What do you think?

Alex Dreyzen



K8: IT'S HAMMER TIME!



AMD64 port

- ① GCC, Binutils: GNU Linker, Assembler, disassembler (společně s Michaelem Matzem, Zdeňkem Dvořákem, Bo Thorstenem)
- ② Kernel (Andi Kleen, Martin Mareš, Vojta Pavlík, Mavel Machek ...)
- ③ Gdb
- ④ glibc (Andreas Jaeger)
- ⑤ Xorg
- ⑥ ...

**System V Application Binary Interface
AMD64 Architecture Processor Supplement
(With LP64 and ILP32 Programming Models)**
Draft Version 0.99.8

Edited by

H.J. Lu¹, Michael Matz², Milind Girkar³, Jan Hubička⁴,
Andreas Jaeger⁵, Mark Mitchell⁶

June 17, 2016

¹hongjiu.lu@intel.com

²matz@suse.de

³milind.girkar@intel.com

⁴jhub@cesk.cz

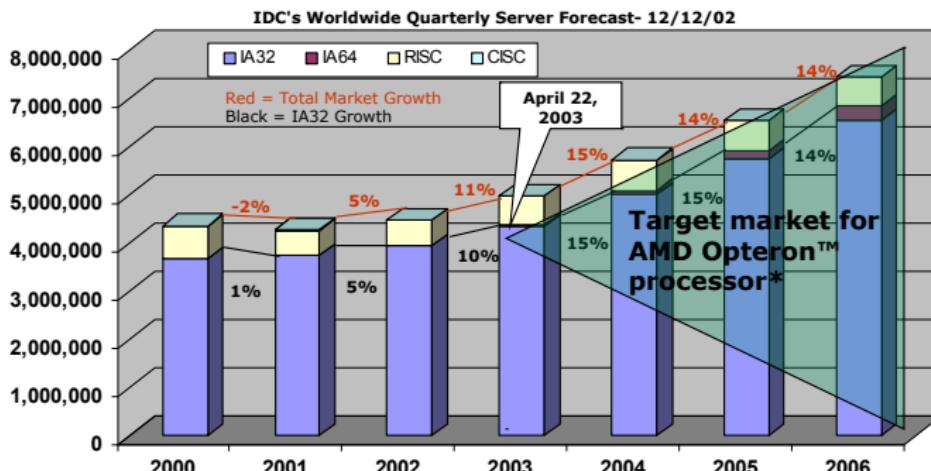
⁵aj@suse.de

⁶mark@codesurgery.com

Figure 3.4: Register Usage

Register	Usage	Preserved across function calls
%rax	temporary register; with variable arguments passes information about the number of vector registers used; 1 st return register	No
%rbx	callee-saved register	Yes
%rcx	used to pass 4 th integer argument to functions	No
%rdx	used to pass 3 rd argument to functions; 2 nd return register	No
%rsp	stack pointer	Yes
%rbp	callee-saved register; optionally used as frame pointer	Yes
%rsi	used to pass 2 nd argument to functions	No
%rdi	used to pass 1 st argument to functions	No
%r8	used to pass 5 th argument to functions	No
%r9	used to pass 6 th argument to functions	No
%r10	temporary register, used for passing a function's static chain pointer	No
%r11	temporary register	No
%r12–%r14	callee-saved registers	Yes
%r15	callee-saved register; optionally used as GOT base pointer	Yes
%xmm0–%xmm1	used to pass and return floating point arguments	No
%xmm2–%xmm7	used to pass floating point arguments	No
%xmm8–%xmm15	temporary registers	No
%mm0–mm7	temporary registers	No
%k0–%k7	temporary registers	No
%st0,%st1	temporary registers; used to return long double arguments	No
%st2–%st7	temporary registers	No
%fs	Reserved for system (as thread specific data register)	No
%rcxw	SSE2 control and status word	partial
%x87_SW	x87 status word	No
%x87_CW	x87 control word	Yes

Enterprises want legacy compatibility **AMD**



- The growth in today's server market is toward technology based on the x86 platform.
- Competing 64-bit products are niche, proprietary technologies that are expensive and difficult to integrate with existing systems.

*Based on AMD market trend research

AMD Opteron™ Processor Ecosystem

Operating Systems



Operating System	Type	Available	Comments
Windows 2000 Server editions	32-bit	✓	http://www.microsoft.com/windows2000/ Users will need to get Opteron chipset drivers from AMD
Red Hat Professional 8.0	32-bit	✓	http://www.redhat.com/software/linux/professional/
SuSE Linux 8.1 editions	32-bit	✓	http://www.suse.com/us/private/products/suse_linux/i386/
Solaris 9 for x86	32-bit	✓	http://www.sun.com/software/solaris/x86/index.html
UnitedLinux Version 1	32-bit 64-bit	✓	http://www.unitedlinux.com/pdfs/UL1_0ProdSpecSheet.pdf Consortium includes SuSE, The SCO Group, Connectiva, Turbolinux
Linux 2.4 kernel patches	32-bit	✓	http://www.x86-64.org/
Mandrake Linux Corporate Server 2.1	32-bit 64-bit	4/22/03	http://www.mandrakesoft.com/products/range?wslang=en
SuSE Linux Enterprise Server (SLES) 8	32-bit 64-bit	4/22/03	http://www.suse.com/us/business/products/server/sles/index.html Beta available from AMD for OEMs, ISVs, and IHBs
Windows Server 2003	32-bit	4/24/03	http://www.microsoft.com/windowsserver2003/default.mspx
NetBSD	64-bit	4/22/03	Development underway by Open Source community
Beowulf Syscl Operating System	64-bit	4/22/03	Linux-based cluster operating system
Turbolinux	64-bit	4/22/03	RC1 candidate available at launch
Red Hat Advanced Server 3.0	64-bit	Stay Tuned	Support announced 08/02 but no release schedule announced.
Windows for AMD64	64-bit	Stay Tuned	http://msdn.microsoft.com Pre-alpha available from Microsoft for OEMs, ISVs, IHBs at
FreeBSD, OPENBSD	64-bit	Stay Tuned	Development underway by Open Source community.

AMD Opteron™ Processor Ecosystem

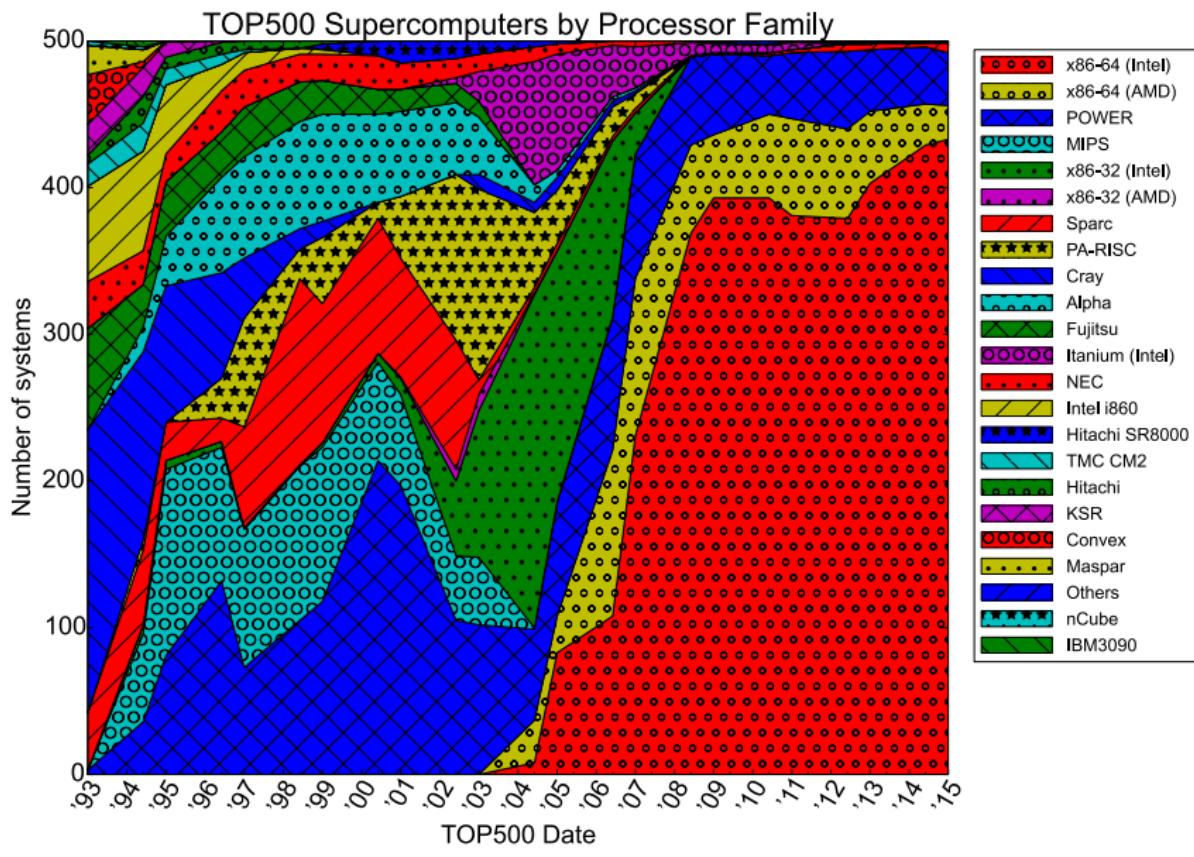
Open Source Development Tools



64-bit Tools	Type	Available	Comments
ATLAS 3.5.0 Developer Release	Library	✓	http://math-atlas.sourceforge.net/ Optimized BLAS (Basic Linear Algebra Subroutines) library
Blackdown Java Platform 2 Version 1.4.2	Linux JAVA	✓	http://www.blackdown.com/java-linux/java2-status/jdk1.4-status.html SUN Java products ported to Linux by Blackdown group
GNU binutils	Utilities	✓	http://www.gnu.org/software/binutils/ GNU collection of binary tools including GNU linker, GNU assembler
GNU C++ (g++) 3.2 GNU C (gcc) 3.2 GNU C (gcc) 3.3 (optimized)	Compilers	✓	http://gcc.gnu.org/ GNU Collection of Compilers (gcc) is a full-featured ANSI C compiler
GNU Debugger (GDB)	Debugger	✓	Analysis tool for debugging programs - included with SuSE SLES 8
GNU glibc 2.2.5 GNU glibc 2.3.2 (optimized)	C Library	✓	http://www.gnu.org/software/libc/libc.html GNU C Library
Other GNU Tools	Various	✓	bash, csb, ksb, strace, libtool - included with SuSE SLES 8
MPICH	Library	✓	Open Source message passing interface for Linux clusters
PERL, Python, Ruby, Tcl/Tk	Language	✓	Scripting languages - included with SuSE SLES 8



GNU means GNU's Not UNIX™ and is the primary project of by the Free Software Foundation (FSF), a non-profit organization committed to the creation of a large body of useful, free, source-code-available software.



RL78 Ada 2012 Objective-C 2.0
gimple-fold WHOPR GCC 4.5.0
LTO MeP Fortran2008 LatticeMicro32
Graphite switch-conv indirect inlining
optimize attribute ACML GCC 4.3.0
early inliner dataflow branch IPA-SSA
Java 1.5 C++0x OpenMP 2007
sel-sched PHI-opt VRP reassoc
Morphosys S/390 GCC 4.0.0
vectorizer DSE SMS Fortran95
new C++ parser trace scheduling
PCH profiledbootstrap GCC 3.3
DFA scheduler GCC Summit
Visium Ada95 MMIX CRIS 2002
FDO SH5 FRV XScale GCC 3.0
bbreorder DCE tailcall IA-64
D30V AVR TBAA DSE LCM
GGC ISO C99 DSE SID
dominator gcov GCC 2.8.0
glibc2 Checker DWARF2 1997
ARC C++ templates
M32R V850
libstdc++ Fortran77 Pentium-GCC
PDP11 MIL-STD-1750A
SH

Link to zscale? What's GNU? Gnu's Not Unix!
GCC 4.1.0 Microblaze a SUSE CTX Go
GCC 4.4.0 software pipelining picochip
Moxie call-DCE IRA SPU Moxie
Fortran2003 Gold Forwprop Renesas RX
GCC 4.2.0 Forwprop
GCC 4.1.0 GIMPLE tuples
SCCVN IP-CP Blackfin
IPA-pure-const IPA structalias
SRA M32C tail-recurse SCCP
GCC 3.4.0 ivopts SCEV mudflap
unit-at-a-time IPA inliner CFG loop
GCC 3.2 TREE-SSA tracer webizer
GCC 3.1 IQ2000 data-refs PTA
x86-64 MCore210/240 SSA-PRE
libcpp Xtensa Java
SSA regrename 68HC11/2
GCC 2.95 AM33 FR30
CFG Chill c4x c3x
EGCS 1.1 GCSE
-Os MIPS16 C++ namespaces
EGCS 1.0 Haifa-sched mn10200/300 alias
Xstormy16
i370
GCC 2.7.0 DSP16xx WindowsNT
unroll

Jak to začalo?

○○

What's GNU? Gnu's Not Unix!

○○○○○○○○○○○○○○○○

AMD64 port a SUSE

○○○○○○○○○○○○

GCC dnes

●○○○

GCC Summit 2003-2010



Modernizace GCC

GCC is finally being dragged, kicking and screaming into 90's

(Richard Henderson, 2002)

Modernizace GCC

GCC is finally being dragged, kicking and screaming into 90's

(Richard Henderson, 2002)

- Implementace globálních optimizerů (Jeff Law, Doug Evans a další, 1998)
- Implementace CFG a profile feedbacku (jako softwarový projekt se Zdeňkem Dvořákem, Pavlem Nejedlým a Josefem Zlomkem, 2002–2003)
- Tree-SSA projekt (Diego Novillo, Nathan Sidwell a další 2001–2004)
- Dataflow branch (Kenneth Zadeck, Dan Berlin, a další 2005–2007)
- Interprocedurální optimalizace, unit-at-a-time (2003)
- Link-time optimalizace (2005–2017)
- Nový C++ parser (Codesourcery)
- OpenMP (Jakub Jelínek, ...)
- Podpora GPU offloadingu
- ...

Modernizace GCC

GCC is finally being dragged, kicking and screaming into 90's

(Richard Henderson, 2002)

- Implementace globálních optimizerů (Jeff Law, Doug Evans a další, 1998)
- Implementace CFG a profile feedbacku (jako softwarový projekt se Zdeňkem Dvořákem, Pavlem Nejedlým a Josefem Zlomkem, 2002–2003)
- Tree-SSA projekt (Diego Novillo, Nathan Sidwell a další 2001–2004)
- Dataflow branch (Kenneth Zadeck, Dan Berlin, a další 2005–2007)
- Interprocedurální optimalizace, unit-at-a-time (2003)
- Link-time optimalizace (2005–2017)
- Nový C++ parser (Codesourcery)
- OpenMP (Jakub Jelínek, ...)
- Podpora GPU offloadingu
- ...

Cygnus/Red Hat, SUSE, IBM, ARM, AMD, Intel, AdaCore, Synopsis, Mentor Graphics/Codesourcery/Samsung, Linaro, Cavium, Apple (až do přechodu na GPL3), Google (do 2014),...

Infrastruktura pro profilem řízené optimalizace v GCC

Zdeněk Dvořák, Jan Hubička, Pavel Nejedlý, Josef Zlomek

Problémy GCC

Zastaralý design:

- Heuristická řešení problémů
- Příliš nízká úroveň RTL
- Pouze lokální optimalizace

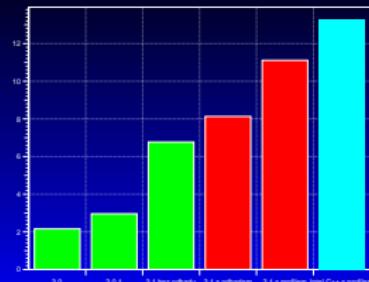
Další problémy:

- Duplikace kódu
- Nejasná interakce mezi jednotlivými optimalizačními průchody

Cíle projektu

- Zavést do GCC optimalizace řízené profilem
 - Naměřené chování
 - Statický odhad
- Sjednotit implementaci Control Flow Graphu
- Zajistit udržování této struktury
- Použít ji pro sběr a ukládání profilovacích informací
- Využít profil v některých optimalizacích
- Přidat nové optimalizace na nich založené

Srovnání s GCC 2.95.3



Total Lines : 9,527,876

Number of Languages : 32

Code Lines : 6,610,527

Total Comment Lines : 1,543,318

Total Blank Lines : 1,374,031

Percent Code Lines : 69.4%

Percent Comment Lines : 16.2%

Percent Blank Lines : 14.4%

Code, Comments and Blank Lines

Zoom 1yr 3yr 5yr 10yr All



Total Lines : 9,527,876

Number of Languages : 32

Code Lines : 6,610,527

Total Comment Lines : 1,543,318

Total Blank Lines : 1,374,031

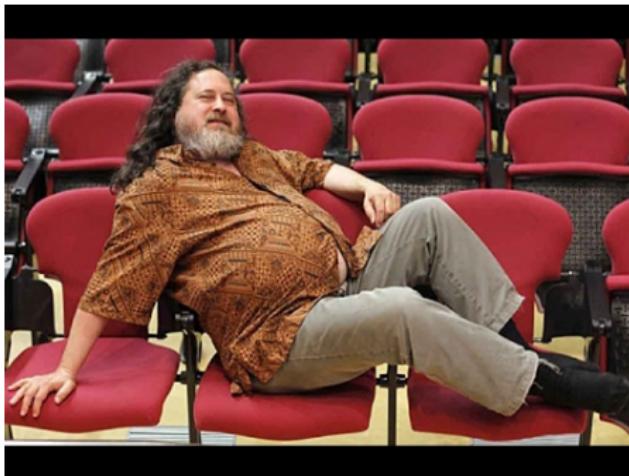
Percent Code Lines : 69.4%

Percent Comment Lines : 16.2%

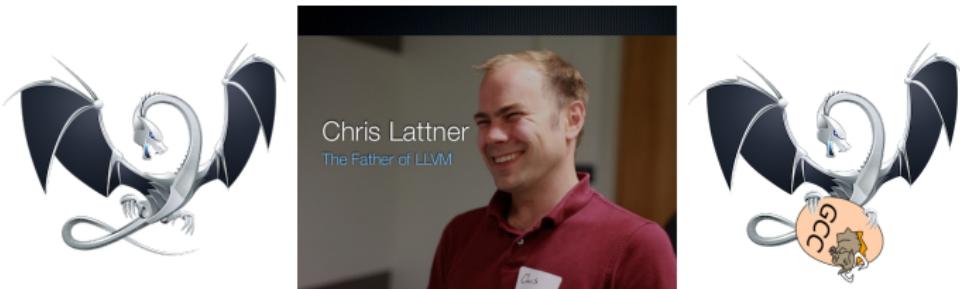
Percent Blank Lines : 14.4%

Code, Comments and Blank Lines

Zoom 1yr 3yr 5yr 10yr All



První skutečná konkurence: LLVM/Clang



2000 Lattner začíná pracovat na diplomce; 2003 první verze;
2005 Apple investuje do LLVM; 2007 clang; GPLv3; 2010 self-hosting;
2011 default pro MacOS; 2016 default pro android

