

Workshop on Essential Abstractions in GCC

Introduction and Opening Remarks

GCC Resource Center
(www.cse.iitb.ac.in/grc)

Department of Computer Science and Engineering,
Indian Institute of Technology, Bombay



July 2009

Outline

- Genesis and Objectives of GCC Resource Center
- Motivation behind the Workshop
- Philosophy and Pedogogy of the Workshop



Part 1

Genesis and Objectives of GRC

How Did It All Begin?

- An Informal Group
 - ▶ CSE faculty members at IITB: Uday Khedker
Amitabha Sanyal
Supratim Biswas

- A Desire



How Did It All Begin?

- An Informal Group
 - ▶ CSE faculty members at IITB: Uday Khedker
Amitabha Sanyal
Supratim Biswas
 - ▶ Reasonably long and deep experience of research in compilers
- A Desire



How Did It All Begin?

- An Informal Group

- ▶ CSE faculty members at IITB: Uday Khedker
Amitabha Sanyal
Supratim Biswas
- ▶ Reasonably long and deep experience of research in compilers

- A Desire

Performing research grounded in theory and corroborated by empirical evidence



How Did It All Begin?

- An Informal Group

- ▶ CSE faculty members at IITB: Uday Khedker
Amitabha Sanyal
Supratim Biswas
- ▶ Reasonably long and deep experience of research in compilers

- A Desire

Performing research grounded in theory and corroborated by empirical evidence

- ▶ Exploring research issues in **real** compilers
- ▶ Demonstrating the relevance and effectiveness of our research in **real** compilers



A Modest Start in 2003...

- Our Tool of Experiment

- Our Guinea Pigs



A Modest Start in 2003...

- Our Tool of Experiment The Gnu Compiler Collection
 - ▶ Compiler generation framework
 - Stable compiler generated for several dozen targets

- Our Guinea Pigs



A Modest Start in 2003...

- Our Tool of Experiment The Gnu Compiler Collection
 - ▶ Compiler generation framework
 - Stable compiler generated for several dozen targets
 - ▶ Millions of users

- Our Guinea Pigs



A Modest Start in 2003...

- **Our Tool of Experiment** The Gnu Compiler Collection
 - ▶ Compiler generation framework
 - Stable compiler generated for several dozen targets
 - ▶ Millions of users

- **Our Guinea Pigs**

Several unsuspecting M.Tech. students, external B.E. students, and project engineers



And Then in 2007...

Advanced GCC Workshop 2007 - Mozilla Firefox


File Edit View History Bookmarks Tools Help

http://www.cse.iitb.ac.in/~uday/gcc-workshop/?file=intro

Most Visited Getting Started Latest Headlines

Workshop on GCC Internals

Organised by
[Centre for Formal Design and Verification of Software](#) and
[Dept. of Computer Science & Engg.,](#)
[IIT Bombay.](#)



[Home](#)

[About GCC](#)

[Other workshops on GCC](#)

[The focus of this workshop](#)

[Take-aways from this workshop](#)

[Participation](#)

[Important Dates](#)

[Schedule](#)

[Reaching IIT Bombay](#)

Home

This workshop is a 3-day instructional workshop (and not a forum for contributed presentations) and involves lectures and laboratory exercises aimed at providing details of the internals of [GCC which is an acronym for GNU Compiler Collection](#). It is the de-facto standard compiler generation framework on GNU/Linux and many other variants of Unix/Linux on a wide variety of machines. In the last 20 years of its existence, it has seen a rapid growth and wide acceptability.

The [focus of this workshop](#) is different


News

(15 Aug 07). Our paper on [incremental construction of machine descriptions](#) has been accepted for presentation at the [GREPS 2007 workshop](#). This paper describes the methodology which was taught in our workshop at IIT Bombay.

(23 July 07). Slides and other workshop material is available at [the downloads page](#).

(10 July 07). There have been some delays in organizing the slides page and in sending certificates. I am tied up with a couple of deadlines. I hope to do the needful soon enough, perhaps on this week end. Will send a mail to all participants once this is done.

Prof. Uday Khedkar Sun Jan 25, 8:33 PM



And then in 2008...

Thanks to small seed grants from IITB and IBM Faculty Award...



And then in 2008...

The screenshot shows the GCC Resource Center website in a Mozilla Firefox browser window. The browser's address bar displays the URL `http://www.cse.iitb.ac.in/grc`. The website header features the IIT Bombay logo on the left, the text ".gccrc" in a large font, and the main title "GCC Resource Center Department of Computer Science & Engg." with a small GCC logo on the right.

The main content area has a green background and contains the following text:

Welcome to the GCC Resource Center at **I.I.T. Bombay**.

GCC is an acronym for GNU Compiler Collection. It is the de-facto standard compiler generation framework for a number of GNU/Linux and many other variants of Unix/Linux on a wide variety of machines and is one of the most dominant softwares in the free software community. Although it follows an **open, collaborative development methodology** and its source code is available to all for inspection and modification, not much effort has gone in bridging the gap between standard conceptual structure of a compiler and the GCC implementation.

This web site is an attempt to give you an insight into ideas and concepts that go behind a practical, industry strength compiler. Visit the **GCC Internals Documents** page for more information about the internal structure and operation of GCC that we have uncovered. We also plan to have other **activities** at the center.

Interesting Aspects of GCC

Historically, GCC has been one of the first projects of the **Free Software Foundation (FSF)** to provide a **free** compiler for its GNU Operating System. It started as C compiler, and was the acronym for "GNU C Compiler" in the early days. Over the years, it has been continuously upgraded to support a number of backend machines. Similarly, on the front end side, it has grown to support a number of front end languages like C++, Objective C, Java, and Fortran to name a few. As a consequence, it has been renamed as "GNU Compiler

The left sidebar contains a search box labeled "GCC Docs" with a "Google Search" button, and a list of navigation links: Home, About GRC, Activities, GCC Internals Docs, GCC Related Links, Downloads, People, IIT Bombay, and CSE. At the bottom of the page, there are logos for W3C HTML 4.0 and W3C CSS, and a footer with the email address `uday[at]cse[at]iitb[DOT]ac[DOT]in`, copyright notice "© 2008, GCC Resource Center", and the date "Last Site Update: July 27, 2008". The browser's status bar at the bottom right shows "Prof. Uday Khedkar Sun Jan 25, 7:50 PM".



Finally in 2009...

- A generous grant from the Department of Information Technology, Ministry of Communication and Information Technology, Government of India.



Plan for July 2009...

Essential Abstractions in GCC '09, Workshop on GCC Internals - Konqueror

Essential Abstractions in GCC '09
A Workshop on GCC Internals by GCC Resource Center

Department of Computer Science & Engineering
Indian Institute of Technology, Bombay

Home Updates Coverage Schedule Registration How to Reach Downloads FAQ

This workshop is a 3-day instructional workshop (and not a forum for contributed presentations) and involves lectures and laboratory exercises aimed at providing details of the internals of **GCC which is an acronym for GNU Compiler Collection**. It is the de-facto standard compiler generation framework on **GNU/Linux** and many variants of Unix. In the last 20 years of its existence, it has seen a rapid growth and wide acceptability.

Take-aways from the Workshop

After attending this workshop

- A teacher of compiler construction will be able to take examples of real compilation processes to illustrate the difference phases of compilation
- A compiler developer wanting to retarget GCC to a new machine will know how to write machine descriptions systematically
- A researcher exploring retargetable compilation will be exposed to real issues in an industry strength compiler
- A researcher exploring machine independent optimizations will be able to add data flow analysis based optimization passes to GCC
- A software engineer will be exposed to the architecture of a very large and very successful software

Who should attend this workshop?

Anybody who has done at least a first level undergraduate course in compiler construction and has some experience of either working in compilers or teaching compilers. A sound understanding of the process of compilation is a must. Familiarity with Unix/Linux (particularly, the command line style of working) is absolutely necessary.

About GCC

GCC, an acronym for GNU Compiler Collection, is a compiler generation framework which generates production quality optimizing compilers from descriptions of target platforms. It follows an **open development model** whereby its source is available for all for inspection and modification. It supports a wide variety of source languages and target machines (including operating system specific variants) in a ready-to-deploy form. Besides, new machines can be added by describing instruction set architectures and some other information (eg. calling conventions).

Novices may want to see the **Wikipedia introduction to GCC**. For experts, the **GCC page** contains a wealth of information including **installation instructions, reference manuals** (which include users' guides as well as details of GCC internals), a **set of frequently asked questions**, a **wiki page** for

Applications Places System Prof. Uday Khedkar Sat May 23, 9:20 AM



Objectives of GCC Resource Center

1. To support the open source movement

Providing training and technical know-how of the GCC framework to academia and industry.



Objectives of GCC Resource Center

1. To support the open source movement

Providing training and technical know-how of the GCC framework to academia and industry.

2. To include better technologies in GCC

Whole program optimization, Optimizer generation, Tree tiling based instruction selection.



Objectives of GCC Resource Center

1. To support the open source movement

Providing training and technical know-how of the GCC framework to academia and industry.

2. To include better technologies in GCC

Whole program optimization, Optimizer generation, Tree tiling based instruction selection.

3. To facilitate easier and better quality deployments/enhancements of GCC

Restructuring GCC and devising methodologies for systematic construction of machine descriptions in GCC.



Objectives of GCC Resource Center

1. To support the open source movement

Providing training and technical know-how of the GCC framework to academia and industry.

2. To include better technologies in GCC

Whole program optimization, Optimizer generation, Tree tiling based instruction selection.

3. To facilitate easier and better quality deployments/enhancements of GCC

Restructuring GCC and devising methodologies for systematic construction of machine descriptions in GCC.

4. To bridge the gap between academic research and practical implementation

Designing suitable abstractions of GCC architecture



Broad Research Goals of GCC Resource Center

- Using GCC as a means
 - ▶ Adding new optimizations to GCC
 - ▶ Adding flow and context sensitive analyses to GCC (In particular, pointer analysis)
- Using GCC as an end in itself
 - ▶ Changing the retargetability mechanism of GCC
 - ▶ Cleaning up the machine descriptions of GCC
 - ▶ Systematic construction of machine descriptions
 - ▶ Facilitating optimizer generation in GCC



GRC Training Programs

Title	Target	Objectives	Mode	Duration
Workshop on Essential Abstractions in GCC	People interested in deploying or enhancing GCC	Explaining the essential abstractions in GCC to ensure a quick ramp up into GCC Internals	Lectures, demonstrations, and practicals (experiments and assignments)	Three days
Tutorial on Essential Abstractions in GCC	People interested in knowing about issues in deploying or enhancing GCC	Explaining the essential abstractions in GCC to ensure a quick ramp up into GCC Internals	Lectures and demonstrations	One day
Workshop on Compiler Construction with Introduction to GCC	College teachers	Explaining the theory and practice of compiler construction and illustrating them with the help of GCC	Lectures, demonstrations, and practicals (experiments and assignments)	Seven days
Tutorial on Demystifying GCC Compilation	Students	Explaining the translation sequence of GCC through gray box probing (i.e. by examining the dumps produced by GCC)	Lectures and demonstrations	Half day



GRC Training Programs

Title	Target	Objectives	Mode	Duration
Workshop on Essential Abstractions in GCC	People interested in deploying and enhancing GCC	Explaining the essential abstractions in GCC to ensure a quick ramp up into GCC Internals 3, 4, and 5 July, 2009 IIT Bombay, Mumbai	Lectures, demonstrations, and practicals (experiments and assignments)	Three days
Tutorial on Essential Abstractions in GCC	People interested in knowing about issues in deploying or enhancing GCC	Explaining the essential abstractions in GCC to ensure a quick ramp up into GCC Internals	Lectures and demonstrations	One day
Workshop on Compiler Construction with Introduction to GCC	College teachers	Explaining the theory and practice of compiler construction and illustrating them with the help of GCC	Lectures, demonstrations, and practicals (experiments and assignments)	Seven days
Tutorial on Demystifying GCC Compilation	Students	Explaining the translation sequence of GCC through gray box probing (i.e. by examining the dumps produced by GCC)	Lectures and demonstrations	Half day



Part 2

Motivation Behind this Workshop

Motivation Behind this Workshop

- To understand GCC well :-)



Motivation Behind this Workshop

- To understand GCC well :-)
- Reasonably quickly



Why is Understanding GCC Difficult?

Some of the obvious reasons:

- *Comprehensiveness*
GCC is a production quality framework in terms of completeness and practical usefulness.
- *Open development model*
Leads to heterogeneity of the design.
However now the main plan is vetted by the steering committee.
- *Rapid versioning*
GCC maintenance is a race against time.



Comprehensiveness of GCC 4.3.1: Wide Applicability

- Input languages supported:
C, C++, Objective-C, Objective-C++, Java, Fortran, and Ada
- Processors supported in standard releases:



Comprehensiveness of GCC 4.3.1: Wide Applicability

- Input languages supported:
C, C++, Objective-C, Objective-C++, Java, Fortran, and Ada
- Processors supported in standard releases:
 - ▶ Common processors:
Alpha, ARM, Atmel AVR, Blackfin, HC12, H8/300, IA-32 (x86), x86-64, IA-64, Motorola 68000, MIPS, PA-RISC, PDP-11, PowerPC, R8C/M16C/M32C, SPU, System/390/zSeries, SuperH, SPARC, VAX



Comprehensiveness of GCC 4.3.1: Wide Applicability

- Input languages supported:
C, C++, Objective-C, Objective-C++, Java, Fortran, and Ada
- Processors supported in standard releases:
 - ▶ Common processors:
Alpha, ARM, Atmel AVR, Blackfin, HC12, H8/300, IA-32 (x86), x86-64, IA-64, Motorola 68000, MIPS, PA-RISC, PDP-11, PowerPC, R8C/M16C/M32C, SPU, System/390/zSeries, SuperH, SPARC, VAX
 - ▶ Lesser-known target processors:
A29K, ARC, ETRAX CRIS, D30V, DSP16xx, FR-30, FR-V, Intel i960, IP2000, M32R, 68HC11, MCORE, MMIX, MN10200, MN10300, Motorola 88000, NS32K, ROMP, Stormy16, V850, Xtensa, AVR32



Comprehensiveness of GCC 4.3.1: Wide Applicability

- Input languages supported:
C, C++, Objective-C, Objective-C++, Java, Fortran, and Ada
- Processors supported in standard releases:
 - ▶ Common processors:
Alpha, ARM, Atmel AVR, Blackfin, HC12, H8/300, IA-32 (x86), x86-64, IA-64, Motorola 68000, MIPS, PA-RISC, PDP-11, PowerPC, R8C/M16C/M32C, SPU, System/390/zSeries, SuperH, SPARC, VAX
 - ▶ Lesser-known target processors:
A29K, ARC, ETRAX CRIS, D30V, DSP16xx, FR-30, FR-V, Intel i960, IP2000, M32R, 68HC11, MCORE, MMIX, MN10200, MN10300, Motorola 88000, NS32K, ROMP, Stormy16, V850, Xtensa, AVR32
 - ▶ Additional processors independently supported:
D10V, LatticeMico32, MeP, Motorola 6809, MicroBlaze, MSP430, Nios II and Nios, PDP-10, TIGCC (m68k variant), Z8000, PIC24/dsPIC, NEC SX architecture



Comprehensiveness of GCC 4.3.1: Size

- Pristine compiler sources (download tarball)
 - ▶ Lines of C code : 2122047
 - ▶ Lines of MD code : 245933
 - ▶ Lines of total code : 2367980
 - ▶ Total authors (approx) : 65
 - ▶ Backend directories : 35
- Generated source for i386 (input language: c)
 - ▶ Total lines of code : 439703
 - ▶ Total lines of .c files code : 334855
 - ▶ Total number of .c files : 16
 - ▶ Total lines of .h files : 104848
 - ▶ Total number of .h files : 274



Open Source and Free Software Development Model

- The Cathedral and the Bazaar
Eric S Raymond, 1999.



Open Source and Free Software Development Model

- The Cathedral and the Bazaar
Eric S Raymond, 1999.
- Cathedral: Total Centralized Control
Design, implement, test, release



Open Source and Free Software Development Model

- The Cathedral and the Bazaar
Eric S Raymond, 1999.
- Cathedral: Total Centralized Control
Design, implement, test, release
 - ▶ + Facilitates a homogeneous, coherent, and well-planned design
 - ▶ – Could be restricted by the vision of a few people



Open Source and Free Software Development Model

- The Cathedral and the Bazaar
Eric S Raymond, 1999.
- Cathedral: Total Centralized Control
Design, implement, test, release
 - ▶ + Facilitates a homogeneous, coherent, and well-planned design
 - ▶ – Could be restricted by the vision of a few people
- Bazaar: Total Decentralization
Release early, release often, let users fix bugs



Open Source and Free Software Development Model

- The Cathedral and the Bazaar
Eric S Raymond, 1999.
- Cathedral: Total Centralized Control
Design, implement, test, release
 - ▶ + Facilitates a homogeneous, coherent, and well-planned design
 - ▶ – Could be restricted by the vision of a few people
- Bazaar: Total Decentralization
Release early, release often, let users fix bugs
 - ▶ – Could lead to heterogeneity in the design
 - ▶ + Is not restricted by the vision of a few people



The Bazaar Approach

Release early, release often, let users fix bugs

- Brooks' law (The Mythical Man Month, 1975)

A combination of the two seems more sensible



The Bazaar Approach

Release early, release often, let users fix bugs

- Brooks' law (The Mythical Man Month, 1975)
 - ▶ 12 man month effort

A combination of the two seems more sensible



The Bazaar Approach

Release early, release often, let users fix bugs

- Brooks' law (The Mythical Man Month, 1975)
 - ▶ 12 man month effort
 - ▶ 1 person working for 12 months
OR
12 persons working for 1 month?

A combination of the two seems more sensible



The Bazaar Approach

Release early, release often, let users fix bugs

- Brooks' law (The Mythical Man Month, 1975)
 - ▶ 12 man month effort
 - ▶ 1 person working for 12 months
OR
12 persons working for 1 month?
- Bazaar approach believes that the two somewhat equivalent in internet-based distributed development.

A combination of the two seems more sensible



The Bazaar Approach

Release early, release often, let users fix bugs

- Brooks' law (The Mythical Man Month, 1975)
 - ▶ 12 man month effort
 - ▶ 1 person working for 12 monthsOR
12 persons working for 1 month?
- Bazaar approach believes that the two somewhat equivalent in internet-based distributed development.
- “Given enough eyeballs, all bugs are shallow”.
Code errors, logical errors, and architectural errors.

A combination of the two seems more sensible



The Current Development Model of GCC

- GCC Steering Committee: Free Software Foundation has given charge
 - ▶ Major policy decisions
 - ▶ Handling Administrative and Political issues
- Release Managers:
 - ▶ Coordination of releases
- Maintainers:
 - ▶ Usually area/branch/module specific
 - ▶ Responsible for design and implementation



The Current Development Model of GCC

- Proposing changes
 - ▶ Extensive discussions over mailing lists
 - ▶ Submissions to `gcc-patches@gcc.gnu.org`
 - ▶ Major changes are forked off as an independent branch which is later merged with the main code
- Reviewers:
 - ▶ Can be general/global or area/branch/module specific
 - ▶ Can approve changes suggested by others
 - ▶ Need approval of other reviewers for their own changes
- Maintainers:
 - ▶ Can approve changes suggested by others
 - ▶ Do not need approval of reviewers for their own changes in their area/branch/module



Why is Understanding GCC Difficult?

Deeper reason: GCC is not a *compiler* but a *compiler generation framework*

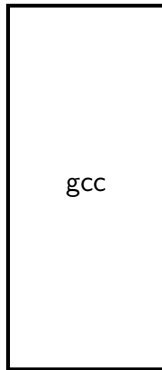
There are two distinct gaps that need to be bridged:

- Input-output of the generation framework: The target specification and the generated compiler
- Input-output of the generated compiler: A source program and the generated assembly program



The Gnu Tool Chain

Source Program



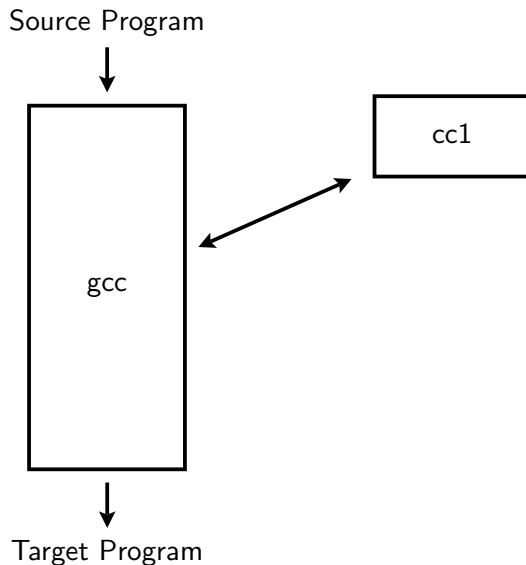
gcc



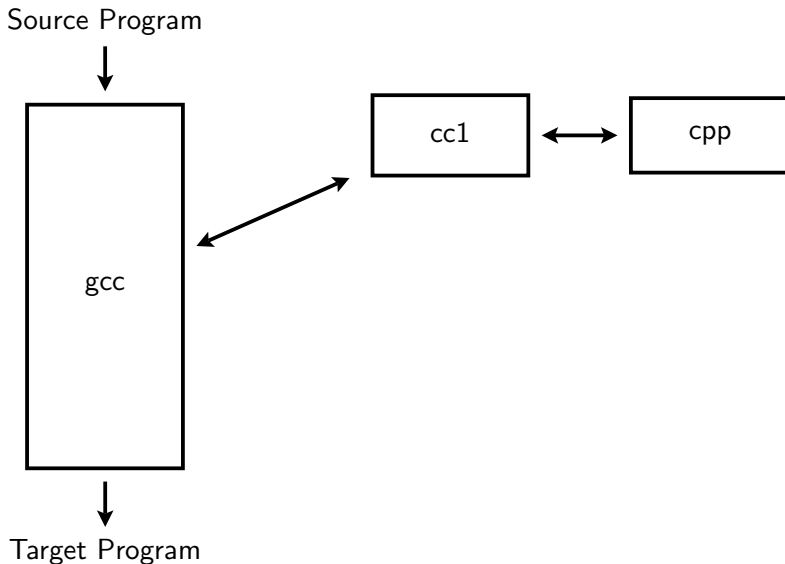
Target Program



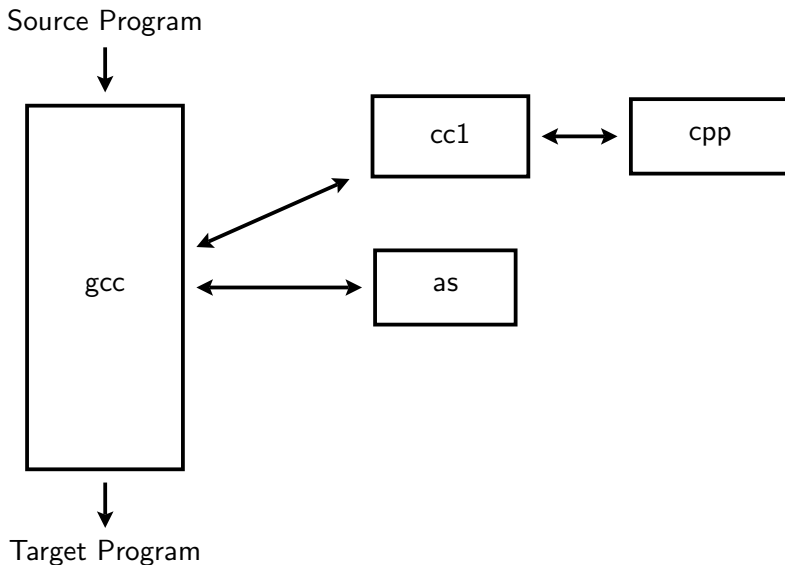
The Gnu Tool Chain



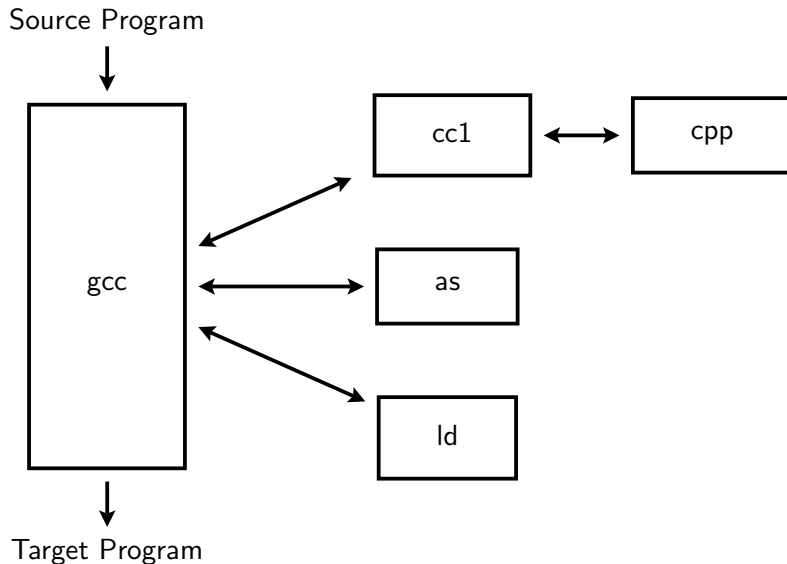
The Gnu Tool Chain



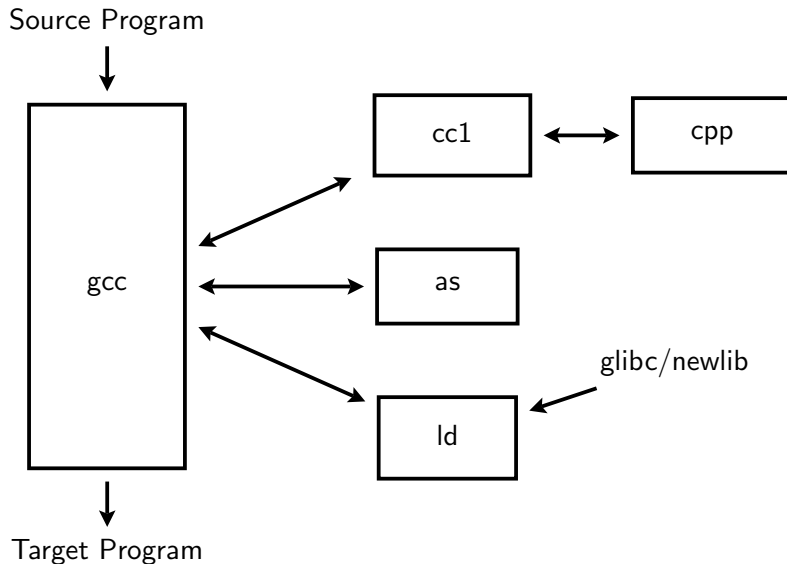
The Gnu Tool Chain



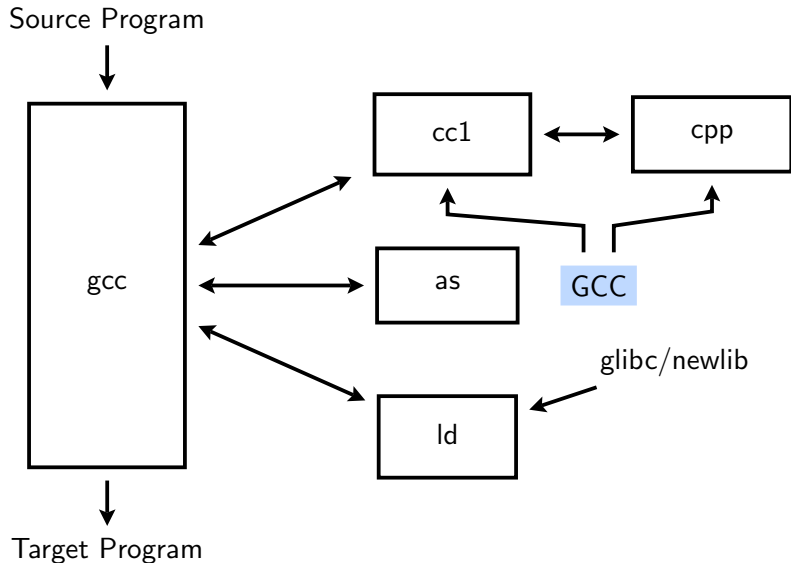
The Gnu Tool Chain



The Gnu Tool Chain

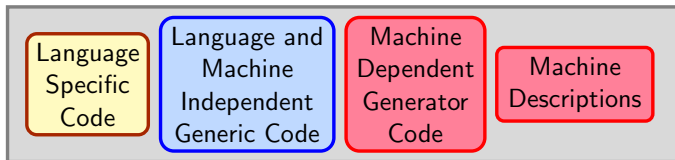


The Gnu Tool Chain



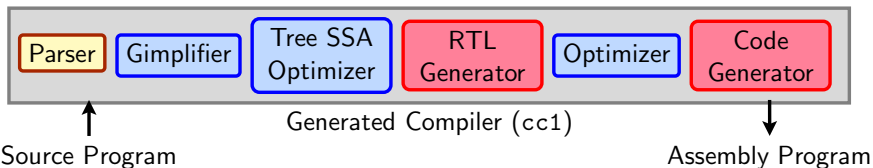
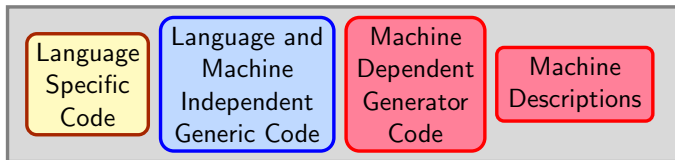
The Architecture of GCC

Compiler Generation Framework

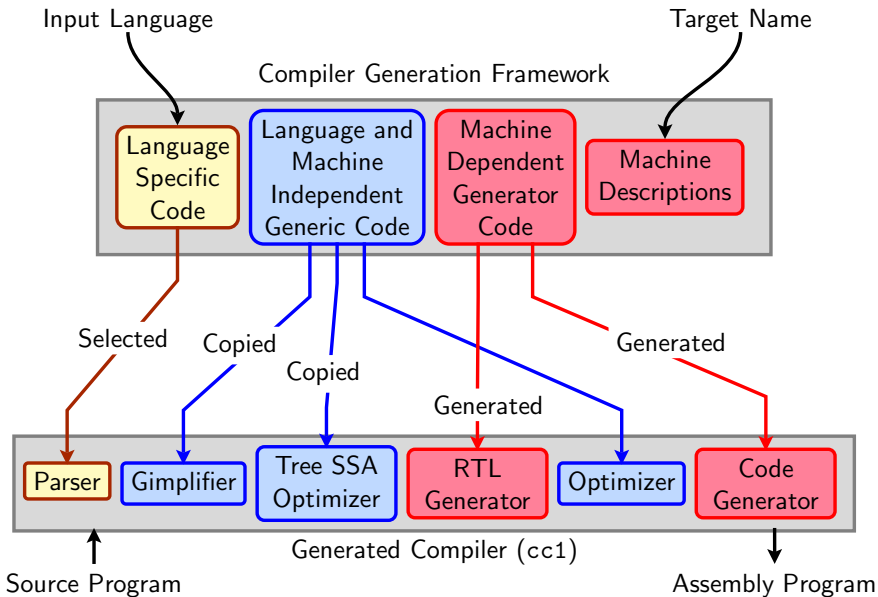


The Architecture of GCC

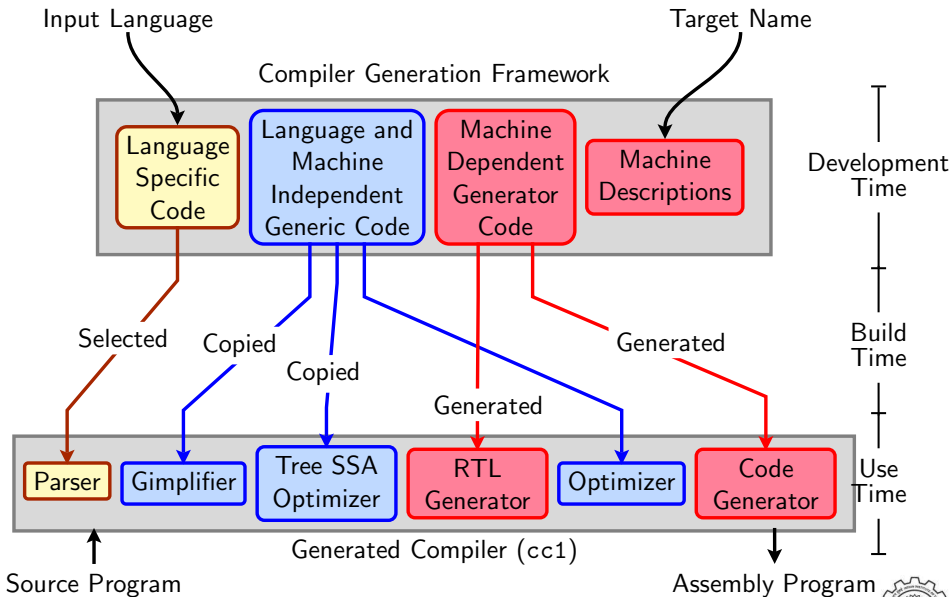
Compiler Generation Framework



The Architecture of GCC



The Architecture of GCC



Part 3

Our Philosophy and Pedogogy

Philosophy and Pedogogy

Twin goals of this workshop:

- Learning how to learn GCC
- Striking a balance between theory and practice



Philosophy and Pedogogy

- We will
 - ▶ Explain configuration and building of GCC

- You will



Philosophy and Pedogogy

- We will
 - ▶ Explain configuration and building of GCC
 - ▶ Explain essential abstractions related to compilation
The key intermediate representations and their manipulations

- You will



Philosophy and Pedogogy

- We will
 - ▶ Explain configuration and building of GCC
 - ▶ Explain essential abstractions related to compilation
The key intermediate representations and their manipulations
 - ▶ Explain essential abstractions related to generation of a compiler
The machine descriptions and their influence on compilation
- You will



Philosophy and Pedogogy

- We will
 - ▶ Explain configuration and building of GCC
 - ▶ Explain essential abstractions related to compilation
The key intermediate representations and their manipulations
 - ▶ Explain essential abstractions related to generation of a compiler
The machine descriptions and their influence on compilation
- You will
 - ▶ Build and run GCC



Philosophy and Pedogogy

- We will
 - ▶ Explain configuration and building of GCC
 - ▶ Explain essential abstractions related to compilation
The key intermediate representations and their manipulations
 - ▶ Explain essential abstractions related to generation of a compiler
The machine descriptions and their influence on compilation
- You will
 - ▶ Build and run GCC
 - ▶ Examine various IR dumps produced by GCC



Philosophy and Pedogogy

- We will
 - ▶ Explain configuration and building of GCC
 - ▶ Explain essential abstractions related to compilation
The key intermediate representations and their manipulations
 - ▶ Explain essential abstractions related to generation of a compiler
The machine descriptions and their influence on compilation
- You will
 - ▶ Build and run GCC
 - ▶ Examine various IR dumps produced by GCC
 - ▶ Add a new machine description and systematically enhance it



Takeaways from this Workshop

- A programmer will get a better compiler
- A compiler professional will be able to deploy and enhance GCC much more easily.
- A compiler researcher will be able to use GCC for research much better.
- A compiler teacher will be able to strike a better balance between theory and practice.
- A compiler student will be exposed to issues in real compilers.



Schedule: Day 1

09:00 to 09:30	Registration
09:30 to 10:00	Introduction and opening remarks
10:00 to 11:15	Getting started with GCC: Configuration and building
11:15 to 11:30	Tea break
11:30 to 13:00	(Lab) Configuration and building
13:00 to 14:00	Lunch
14:00 to 15:00	Gray Box Probing of GCC An introduction to IRs
15:00 to 15:45	Introduction to Gimple IR
15:45 to 16:00	Tea break
16:00 to 17:30	(Lab) Adding a gimple pass
19:00 to 20:00	Inspecting and debugging gcc code (Optional Session) (use of cscope, ctags, gdb etc.)
20:00 to 21:00	Dinner



Schedule: Day 2

09:30 to 10:15	Introduction to RTL
10:15 to 11:00	An overview of retargetability and an introduction to machine descriptions
11:00 to 11:15	Tea break
11:15 to 13:00	(Lab) Adding an RTL pass
13:00 to 14:00	Lunch
14:00 to 15:30	Spim machine descriptions: Level 0 and 1
15:30 to 15:45	Tea break
15:45 to 17:30	(Lab) spim machine descriptions
18:30 to 20:00	(Optional session) An Overview of research projects in GCC Resource Center
20:00 to 21:00	Dinner



Schedule: Day 3

09:30 to 10:10	spim machine descriptions levels 2 and 3
10:15 to 10:45	The retargetability model of GCC
10:45 to 11:00	Tea break
11:00 to 13:00	(Lab) spim machine descriptions
13:00 to 14:00	Lunch
14:00 to 15:30	The Generic Data Flow Analyzer in GCC
15:30 to 15:45	Tea break
15:45 to 17:30	Concluding session



Announcements and Questions

- Lab arrangements:
 - ▶ Location: Block A, first room on the left
 - ▶ Common local account on each machine:
Use the same machine for all lab assignment.
 - ▶ Login id: gccworkshop09, Password: workshop



Announcements and Questions

- Lab arrangements:
 - ▶ Location: Block A, first room on the left
 - ▶ Common local account on each machine:
Use the same machine for all lab assignment.
 - ▶ Login id: gccworkshop09, Password: workshop
- Accessing internet
 - ▶ Set proxy to proxy.cse.iitb.ac.in
 - ▶ Set no proxy for .iitb.ac.in
 - ▶ Login id: workshop, Password: workshop123



Announcements and Questions

- Lab arrangements:
 - ▶ Location: Block A, first room on the left
 - ▶ Common local account on each machine:
Use the same machine for all lab assignment.
 - ▶ Login id: gccworkshop09, Password: workshop
- Accessing internet
 - ▶ Set proxy to proxy.cse.iitb.ac.in
 - ▶ Set no proxy for .iitb.ac.in
 - ▶ Login id: workshop, Password: workshop123
- Tiffin/Tea at 5:30
- Dinner and breakfast arrangements
 - ▶ Breakfast available in the hostels
 - ▶ Dinner: Fri, Sat: FCKA Foyer
 - ▶ Dinner Sun, Breakfast Mon: in hostels on prior intimation



Announcements and Questions

- Certificates of participation
Can be collected from Nisha on Sunday afternoon
- Receipts of payments
Will be sent to your address by post



Announcements and Questions

- Certificates of participation
Can be collected from Nisha on Sunday afternoon
- Receipts of payments
Will be sent to your address by post
- Concluding session
 - ▶ Informal discussions
 - ▶ Brief (10 minute) presentations by participating organizations/individuals
 - ▶ Confirmed presentations from KPIT Cummins, and Acme Technologies
 - ▶ If you are interested, please contact me well today or latest tomorrow
 - ▶ Feedback forms

