

## Workshop on Essential Abstractions in GCC

### Introduction and Opening Remarks

GCC Resource Center  
([www.cse.iitb.ac.in/grc](http://www.cse.iitb.ac.in/grc))

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



July 2010

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



### 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



#### Part 1

### Genesis and Objectives of GRC

## 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 2008...

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



## And Then in 2007...



## And then in 2008...



## Finally in 2009...

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

- National Resource Center for F/OSS, Phase II
- Participating agencies:  
CDAC Chennai (coordinating agency), CDAC Mumbai, CDAC Hyderabad, [IIT Bombay](#), IIT Madras, Anna University
- IIT Bombay's focus: GCC

## July 2009...

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

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

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).

Noivces 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 the developers of GCC, additional reading material, and several mailing lists for more detailed issues and queries.

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

## 2009...

**GCC Resource Center**  
Department of Computer Science & Engineering  
Indian Institute of Technology, Bombay

Welcome to GCC Resource Center at IIT Bombay

**NEWS & EVENTS**

- Course CS 715 (Design and Implementation of Gnu Compiler Generation Framework)
- Tutorial on GCC for Parallelization as part of ACM PPoPP 2010
- Workshop on Compiler Construction with Introduction to GCC (7th to 13th December 2009)
- Workshop on Essential Abstractions in GCC (3rd to 5th July 2009)

You are visitor number **000876**

**About GCC**

GCC is an acronym for GNU Compiler Collection. It is the de-facto standard compiler generation framework for all distros on GNU/Linux and many other variants of Unix on a wide variety of machines and is one of the most dominant softwares in the free software community. It supports several input languages for a variety of operating systems on more than 30 target processors. More back ends can be added by describing new target processors using the specification mechanism provided by GCC.

Noivces 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 the developers of GCC, additional reading material, and several mailing lists for more detailed issues and queries.

**About GCC Resource Center**

This Center has been established at IIT Bombay with the twin goals of (a) spreading the know-how of GCC by building suitable abstractions of GCC internals, and (b) improving GCC by introducing new technologies. It was initiated with a seed grant from IIT Bombay and an IBM Faculty Award for Prof. Uday Khedkar. Currently, this center is supported by a generous grant from Department of Information Technology (DIT), Ministry of Communication and Information Technology (MCIIT), Govt. of India, under the second phase of the National Resource Center for Free/Open Source Software (NRCFOSS II).

**Interesting Aspects of GCC**

Update: GCC has been one of the first projects of the Free Software Foundation (FSF) to provide a free compiler for the ARM.

Mon Jan 18, 12:09 AM

## July 2010...

**Essential Abstractions in GCC '10**  
A Workshop on GCC Internals by GCC Resource Center  
July 5-8, 2010, IIT Bombay

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

This workshop is a 4-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 (GNU Compiler Collection). GCC 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.

**What is 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).

Noivces 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 the developers of GCC, additional reading material, and several mailing lists for more detailed issues and queries. An excellent description of GCC internals can also be found on Wikipedia. Some notes on GCC internals are available at Washington University.

**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. We have identified some homework exercises to be done before the participants arrive for the workshop.

**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 machine independent optimizations will be able to add data flow analysis based optimization passes to GCC
- A researcher exploring retargetable compilation will be exposed to real issues in an industry strength compiler
- A software engineer will be exposed to the architecture of a very large and very successful software

**Structure of the workshop**

This offering of the workshop has a lot of new material and all the material has been upgraded to GCC 4.5.0. Besides, based on the feedback from

**UPDATES**

- June 8, 2010: A detailed schedule of the workshop is now available.
- June 1, 2010: The first set of selected candidates has been put up.

Past updates...

## 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)
- ▶ Translation validation of GCC

### • 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



## Broad Areas of Interests

- Program Analysis and Optimization
- Translation Validation
- Retargetable compilation
- Parallelization and Vectorization for SIMD and MIMD Architectures

General explorations applied in the context of 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 GCC 3, 4, and 5 July, 2009 IIT Bombay, Mumbai	Explaining the essential abstractions in GCC 5-8 July, 2010 IIT Bombay, Mumbai	Lectures, demonstrations, and assignments)	Three days
Tutorial on Essential Abstractions in GCC	People interested in knowing about issues in deploying or enhancing GCC (modified version) 9 Jan 2010 ACM PPOPP, Bangalore	Explaining the essential	Lectures and demonstrations	One day
Workshop on Compiler Construction with Introduction to GCC	College teachers 7-13 Dec 2009, IIT Bombay, Mumbai	Explaining the theory and practice of compiler construction and helping them with help of GCC	Lectures, demonstrations, and assignments)	Seven days
Tutorial on	Students	Explaining the translation process of GCC	Lectures and	Half day
20 Jan 2010, Cummins College, Pune	20 Feb 2010, IIITDM, Jabalpur	06 March 2010, SGGS IET, Nanded	27 March 2010, RSCoE, Pune	25 Apr 2010, Punjabi Univ., Patiala



Part 2

## Workshop Plan

## GRC Training Programs

CS 715: The Design and Implementation of GNU Compiler Generation Framework

- 6 credits semester long course for M.Tech. (CSE) students at IIT Bombay
- Significant component of experimentation with GCC
- Introduced in 2008-2009



## Motivation Behind this Workshop

- To understand GCC well :-)
- Reasonably quickly



## Philosophy and Pedogogy

Twin goals of this workshop:

- *Learning how to learn GCC*

Our focus will be on

- ▶ giving you some core information
- ▶ showing you how to discover more information

- *Striking a balance between theory and practice*

Our focus will be on showing you how to

- ▶ discover concepts in a large code base and build abstractions
- ▶ take concepts and update a large code base
- ▶ relate the class room concepts of compilers to an industry strength compiler



## 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 program analysis in GCC
- ▶ 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 passes to GCC
- ▶ Add a new machine description and systematically enhance it



## Our Canvas

- Version: GCC 4.5.0
- Language: C
- Targets: i386, spim (mips simulator)



## 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.



**Day 1 (Monday 5 July 2010)**

09:45 to 10:00	Inaguration, Welcome note by the Head, CSE	
10:00 to 10:30	<i>Theme:</i> Introduction <i>Topic:</i> Introductory remarks, Workshop plan	Lecture
10:30 to 11:00	<i>Theme:</i> Introduction <i>Topic:</i> An overview of compilation and GCC	Lecture
11:00 to 11:30	Tea break	
11:30 to 13:00	<i>Theme:</i> An external view of GCC <i>Topic:</i> First level gray box probing of GCC	Lecture
13:00 to 14:00	Lunch	
14:00 to 15:30	<i>Theme:</i> An external view of GCC <i>Topic:</i> Gray box probing for machine independent optimizations	Lecture
15:30 to 16:00	Tea break	
16:00 to 17:30	<i>Theme:</i> An external view of GCC <i>Topic:</i> Gray box probing of GCC	Lab
TBD	Optional: ctags, cscope, ddd, shell, make, screen, patch files <i>Participants can continue to do the lab work until dinner (20:00)</i>	

**Day 3 (Wednesday 7 July 2010)**

09:30 to 10:15	<i>Theme:</i> Program analysis and optimization <i>Topic:</i> Introduction to data flow analysis	Lecture
10:15 to 11:00	<i>Theme:</i> Program analysis and optimization <i>Topic:</i> Introduction to GDFA	Lecture
11:00 to 11:30	Tea break	
11:30 to 12:15	<i>Theme:</i> Program analysis and optimization <i>Topic:</i> Introduction to parallelization and vectorization	Lecture
12:15 to 13:00	<i>Theme:</i> Program analysis and optimization <i>Topic:</i> Parallelization and vectorization in GCC	Lecture
13:00 to 14:00	Lunch	
14:00 to 15:00	<i>Theme:</i> Program analysis and optimization <i>Topic:</i> Parallelization and vectorization in GCC	Lecture
15:00 to 15:30	<i>Theme:</i> Program analysis and optimization <i>Topic:</i> Using GDFA	Lab
15:30 to 16:00	Tea break	
16:00 to 17:30	<i>Theme:</i> Program analysis and optimization <i>Topic:</i> Parallelization and vectorization in GCC	Lab

**Day 2 (Tuesday 6 July 2010)**

09:30 to 10:00	<i>Theme:</i> Generating a compiler from GCC <i>Topic:</i> Configuration and building	Lecture
10:00 to 10:30	<i>Theme:</i> Generating a compiler from GCC <i>Topic:</i> Building a cross compiler	Lecture
10:30 to 11:00	<i>Theme:</i> An internal view of GCC <i>Topic:</i> GCC control flow and plugins	Lecture
11:00 to 11:30	Tea break	
11:30 to 12:00	<i>Theme:</i> An internal view of GCC <i>Topic:</i> GCC control flow and plugins	Lecture
12:00 to 13:00	<i>Theme:</i> An internal view of GCC: <i>Topic:</i> Manipulating GIMPLE and RTL IRs.	Lecture
13:00 to 14:00	Lunch	
14:00 to 15:30	<i>Theme:</i> Intermediate representations in GCC <i>Topic:</i> Adding GIMPLE interprocedural pass	Lab
15:30 to 16:00	Tea break	
16:00 to 17:30	<i>Theme:</i> Intermediate representations in GCC <i>Topic:</i> Adding RTL pass	Lab

**Day 4 (Thursday 8 July 2010)**

09:30 to 10:15	<i>Theme:</i> Retargetability issues <i>Topic:</i> Introduction to machine descriptions	Lecture
10:15 to 11:00	<i>Theme:</i> Retargetability issues <i>Topic:</i> spim machine descriptions levels 0 and 1	Lecture
11:00 to 11:30	Tea break	
11:30 to 12:15	<i>Theme:</i> Retargetability issues <i>Topic:</i> spim machine descriptions levels 2, 3, 4	Lecture
12:15 to 13:00	<i>Theme:</i> Retargetability issues <i>Topic:</i> Advanced issues in machine descriptions and retargetability model of GCC	Lecture
13:00 to 14:00	Lunch	
14:00 to 15:30	<i>Theme:</i> Retargetability issues <i>Topic:</i> spim machine descriptions	Lab
15:30 to 16:00	Tea break	
16:00 to 17:00	<i>Theme:</i> Concluding session Activity Summarization, interactive session, distribution of certificates	



## Announcements and Questions

### Lecture and lab schedule

- Flexible
- Duration, time may be changed dynamically based on how well things are being received
- Lab and lectures may be interchanged too



## Announcements and Questions

- Tea at 17:30
- Dinner and breakfast arrangements
  - ▶ Breakfast available in the hostels
  - ▶ Dinner: Mon, Tue, Wed: Available at 20:00 in foyer



## Announcements and Questions

### Lab arrangements:

- Assignments have to be done in groups of two.
  - ▶ Please use the tea time to finalize your group
  - ▶ A sheet will be circulated after the tea for group details
  - ▶ If you need a laptop, we will issue it during lunch. You will need to return it in the evening.
- Doing the assignments.
  - ▶ Do all exercises on your laptop, or
  - ▶ Use your laptop and log into our servers,
  - ▶ Use our laptop and log into our servers.
- How to connect to server?  
Separate information sheet has been provided.
- Teaching assistants will help you in doing the assignments



## Announcements and Questions

- Receipts of payments
  - ▶ If you have sent the DD earlier, your receipt may be ready.
  - ▶ Please collect from Nisha on Thursday during the lunch time.
- Concluding session
  - ▶ Informal discussions
  - ▶ Brief (10 minute) presentations by participating organizations/individuals  
If you are interested, please contact me today or latest tomorrow
  - ▶ Feedback forms

