

GCC Source Code: An Internal View

Uday Khedker

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



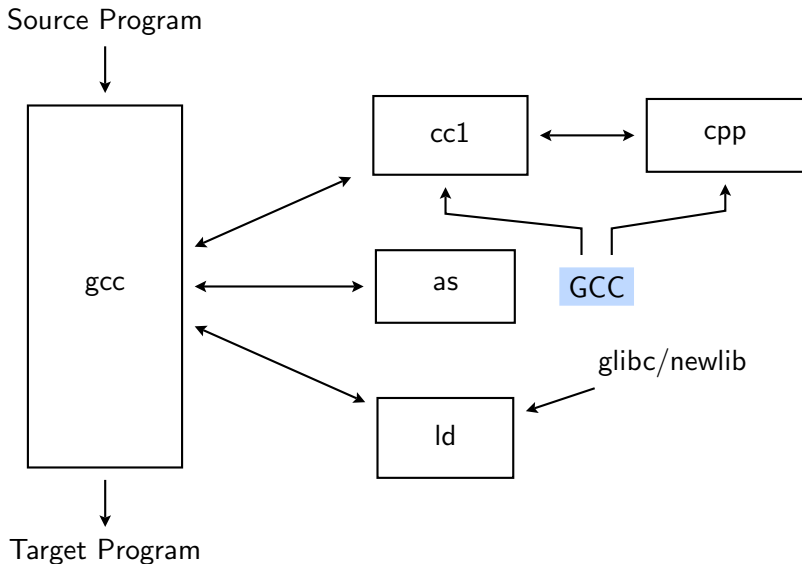
Feb 2010

Outline

- A summary of GCC architecture
- Walking the maze of a large code base
- An Internal View of GCC code

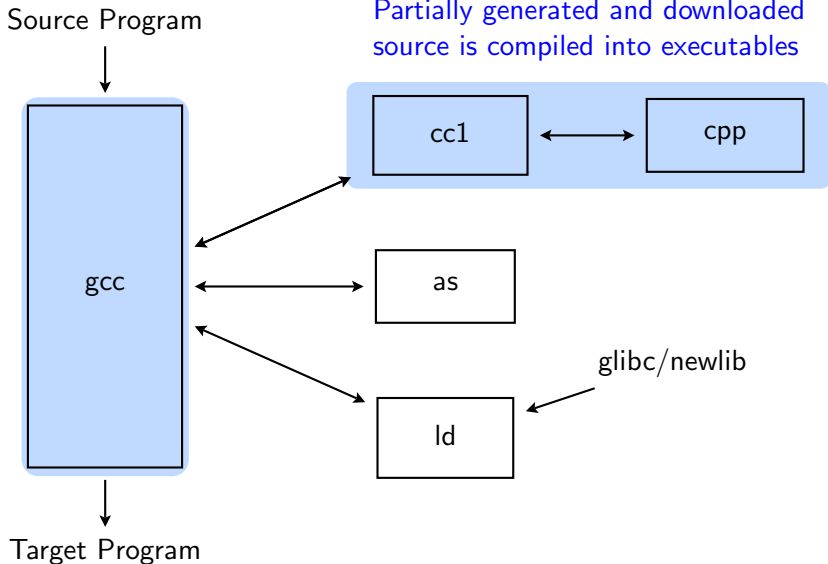


The Gnu Tool Chain

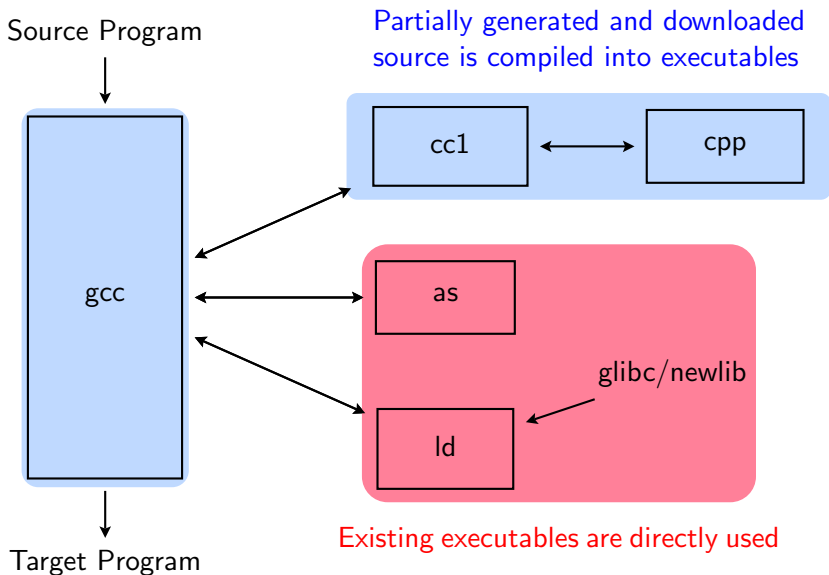


The Gnu Tool Chain

Partially generated and downloaded source is compiled into executables

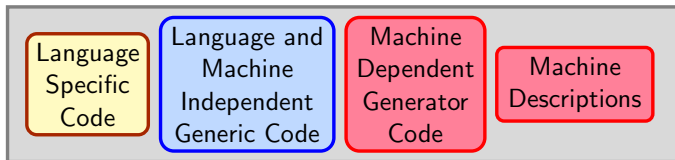


The Gnu Tool Chain



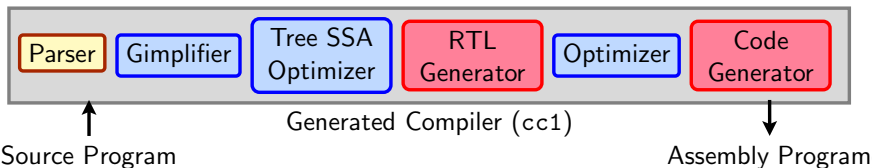
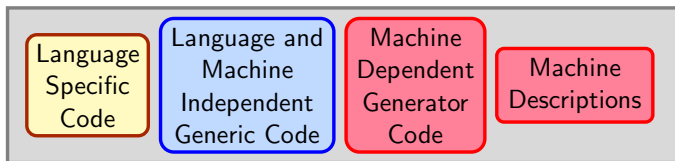
The Architecture of GCC

Compiler Generation Framework

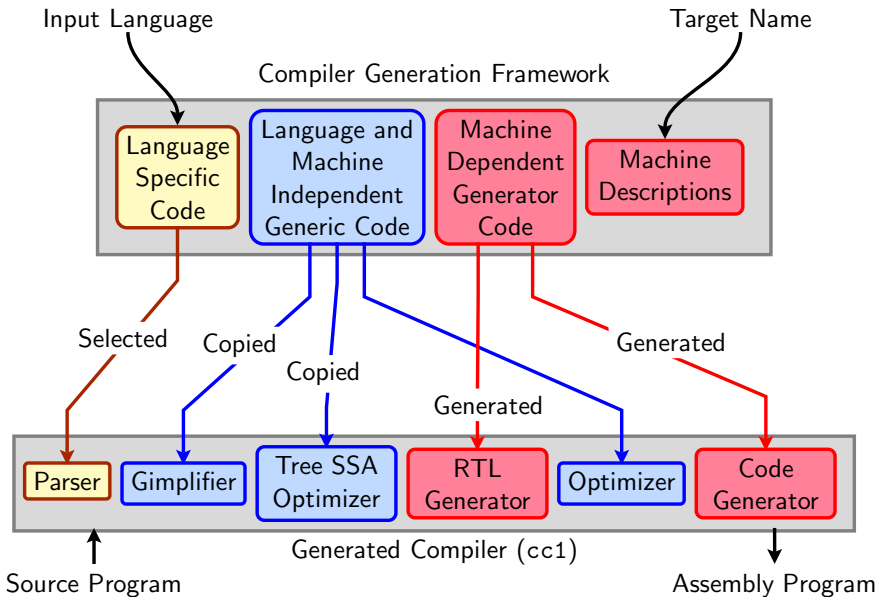


The Architecture of GCC

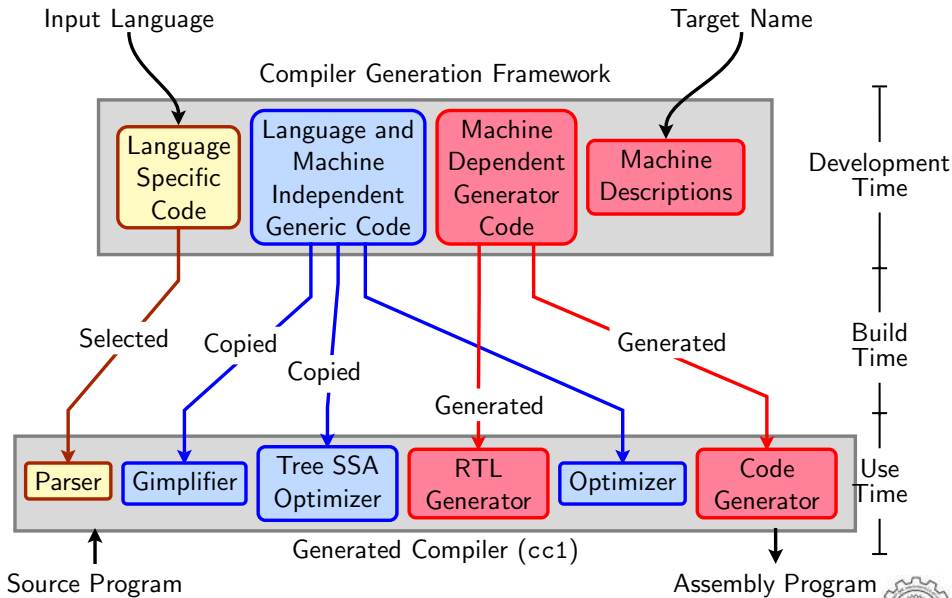
Compiler Generation Framework



The Architecture of GCC



The Architecture of GCC



An Example of The Generation Related Gap

- Predicate function for invoking the loop distribution pass

```
static bool
gate_tree_loop_distribution (void)
{
    return flag_tree_loop_distribution != 0;
}
```



An Example of The Generation Related Gap

- Predicate function for invoking the loop distribution pass

```
static bool
gate_tree_loop_distribution (void)
{
    return flag_tree_loop_distribution != 0;
}
```

- There is no declaration of or assignment to variable `flag_tree_loop_distribution` in the entire source!



An Example of The Generation Related Gap

- Predicate function for invoking the loop distribution pass

```
static bool
gate_tree_loop_distribution (void)
{
    return flag_tree_loop_distribution != 0;
}
```

- There is no declaration of or assignment to variable `flag_tree_loop_distribution` in the entire source!
- It is described in `common.opt` as follows

```
ftree-loop-distribution
Common Report Var(flag_tree_loop_distribution) Optimization
Enable loop distribution on trees
```



An Example of The Generation Related Gap

- Predicate function for invoking the loop distribution pass

```
static bool
gate_tree_loop_distribution (void)
{
    return flag_tree_loop_distribution != 0;
}
```

- There is no declaration of or assignment to variable `flag_tree_loop_distribution` in the entire source!
- It is described in `common.opt` as follows

```
ftree-loop-distribution
Common Report Var(flag_tree_loop_distribution) Optimization
Enable loop distribution on trees
```

- The required C statements are generated during the build



Another Example of The Generation Related Gap

Locating the `main` function in the directory `gcc-4.4.2/gcc` using `cscope`



Another Example of The Generation Related Gap

Locating the main function in the directory gcc-4.4.2/gcc using cscope

File	Line	
0 collect2.c	766	main (int argc, char **argv)
1 fix-header.c	1074	main (int argc, char **argv)
2 fp-test.c	85	main (void)
3 gcc.c	6216	main (int argc, char **argv)
4 gcov-dump.c	76	main (int argc ATTRIBUTE_UNUSED, char **argv)
5 gcov-iov.c	29	main (int argc, char **argv)
6 gcov.c	355	main (int argc, char **argv)
7 gen-protos.c	130	main (int argc ATTRIBUTE_UNUSED, char **argv)
8 genattr.c	89	main (int argc, char **argv)
9 genattrtab.c	4438	main (int argc, char **argv)
a genautomata.c	9321	main (int argc, char **argv)
b genchecksum.c	65	main (int argc, char ** argv)
c gencodes.c	51	main (int argc, char **argv)
d genconditions.c	209	main (int argc, char **argv)
e genconfig.c	261	main (int argc, char **argv)
f genconstants.c	50	main (int argc, char **argv)



Another Example of The Generation Related Gap

Locating the main function in the directory gcc-4.4.2/gcc using cscope

```
g genemit.c          820 main (int argc, char **argv)
h genextract.c       394 main (int argc, char **argv)
i genflags.c         231 main (int argc, char **argv)
j gengentr1.c        350 main (int argc, char **argv)
k gengtype.c         3584 main (int argc, char **argv)
l genmddeps.c         45 main (int argc, char **argv)
m genmodes.c         1376 main (int argc, char **argv)
n genopinit.c         472 main (int argc, char **argv)
o genoutput.c        1005 main (int argc, char **argv)
p genpeep.c          353 main (int argc, char **argv)
q genpreds.c         1399 main (int argc, char **argv)
r genrecog.c         2718 main (int argc, char **argv)
s main.c             33 main (int argc, char **argv)
t mips-tdump.c       1393 main (int argc, char **argv)
u mips-tfile.c       655 main (void )
v mips-tfile.c       4690 main (int argc, char **argv)
w protoize.c         4373 main (int argc, char **const argv)
```

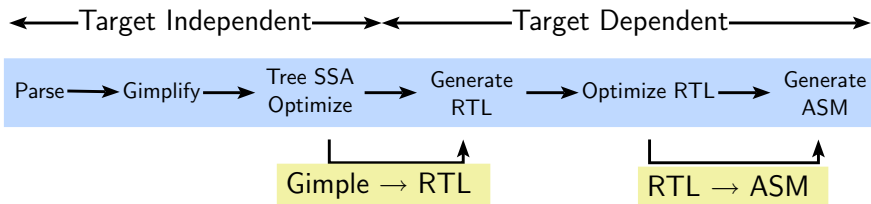


Transformation Passes in GCC

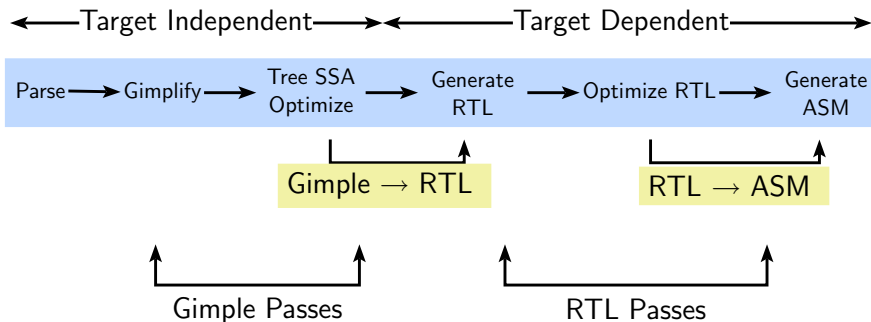
- A total of 196 unique pass names initialized in `$(SOURCE)/gcc/passes.c`
 - ▶ Some passes are called multiple times in different contexts
Conditional constant propagation and dead code elimination are called thrice
 - ▶ Some passes are only demo passes (eg. data dependence analysis)
 - ▶ Some passes have many variations (eg. special cases for loops)
Common subexpression elimination, dead code elimination
- The pass sequence can be divided broadly in two parts
 - ▶ Passes on Gimple
 - ▶ Passes on RTL
- Some passes are organizational passes to group related passes



Basic Transformations in GCC



Basic Transformations in GCC



Passes On Gimple

Pass Group	Examples	Number of passes
Lowering	Gimple IR, CFG Construction	12
Interprocedural Optimizations	Conditional Constant Propagation, Inlining, SSA Construction	36
Intraprocedural Optimizations	Constant Propagation, Dead Code Elimination, PRE	40
Loop Optimizations	Vectorization, Parallelization	24
Remaining Intraprocedural Optimizations	Value Range Propagation, Rename SSA	23
Generating RTL		01
Total number of passes on Gimple		136



Passes On RTL

Pass Group	Examples	Number of passes
Intraprocedural Optimizations	CSE, Jump Optimization	15
Loop Optimizations	Loop Invariant Movement, Peeling, Unswitching	7
Machine Dependent Optimizations	Register Allocation, Instruction Scheduling, Peephole Optimizations	59
Assembly Emission and Finishing		03
Total number of passes on RTL		84



Comprehensiveness of GCC 4.4.2: Size

Source Lines	Number of lines in the main source	2,187,216
	Number of lines in libraries	1,633,558
Directories	Number of subdirectories	3794
Files	Total number of files	62998
	C source files	13968
	Header files	9163
	C++ files	4191
	Java files	6340
	Makefiles and Makefile templates	163
	Configuration scripts	52
	Machine description files	206

(Line counts estimated by the program `sloccount` by David A. Wheeler)



Walking the Maze of a Large Code Base

- Use cscope

```
cd $SOURCE
cscope -R
```
- Use ctags

```
cd $SOURCE
ctags -R
```

Make sure you use `exeburant-ctags`



gcc-4.4.2 Control Flow

```
main
  validate_all_switches
  lookup_compiler
  do_spec
    do_spec_2
      do_spec_1 /* Get the name of the compiler */
  execute
    pex_init
    pex_run
      pex_run_in_environment
        obj->funcs->exec_child
```



cc1-4.4.2 Control Flow

```

main
  toplev_main
  decode_options
  do_compile
  compile_file
    { lang_hooks.parse_file => c_common_parse_file
      { c_parse_file
        c_parser_translation_unit
        c_parser_declaration_or_fndef
        finish_function
        c_genericize
        gimplify_function_tree
        gimplify_body
        gimplify_stmt
        gimplify_expr
        cgraph_finalize_function
        pop_file_scope
        cgraph_finalize_compilation_unit
        cgraph_analyze_functions
        cgraph_analyze_function
        cgraph_lower_function
        tree_lowering_passes
        execute_pass_list (&all_lowering_passes)
      }
      lang_hooks.decls.final_write_globals => c_write_global_declarations
      { cgraph_optimize
        cgraph_analyze_functions
        cgraph_analyze_function
        cgraph_lower_function
        tree_lowering_passes
        execute_pass_list (&all_lowering_passes)
        ipa_passes
        cgraph_expand_all_functions
        cgraph_expand_functions
        tree_rest_of_compilation
        execute_pass_list (&all_passes)
      }
      targetm.asm_out.file_end
    }
  }
  finalize
}

```



cc1-4.4.2 Control Flow: Lowering Passes

```
lang_hooks.parse_file => c_common_parse_file
  c_parse_file
    c_parser_translation_unit
      c_parser_declaration_or_fndef
        finish_function
          c_genericize
            gimplify_function_tree
              gimplify_body
                gimplify_stmt
                  gimplify_expr
cgraph_finalize_function
pop_file_scope
cgraph_finalize_compilation_unit
  cgraph_analyze_functions
    cgraph_analyze_function
      cgraph_lower_function
        tree_lowering_passes
          execute_pass_list (all_lowering_passes)
```



cc1-4.4.2 Control Flow: Optimization and Code Generation Passes

```
lang_hooks.decls.final_write_globals => c_write_global_declarations
{
  cgraph_optimize
    cgraph_analyze_functions
    cgraph_analyze_function
    cgraph_lower_function
    tree_lowering_passes
    execute_pass_list (&all_lowering_passes)
  ipa_passes
  cgraph_expand_all_functions
  cgraph_expand_functions
  tree_rest_of_compilation
  execute_pass_list (&all_passes)
}
```

