

WHAT CAN COMPILER EXPLORER DO FOR GCC

Marc Poulhiès, FOSDEM 2024



WHAT

The image shows the Compiler Explorer web application. At the top, there is a navigation bar with the logo, 'Add...', 'More', and 'Templates' menus. A notification box asks for suggestions or bug reports. The main interface is split into two panes. The left pane shows the C source code for a 'square' function. The right pane shows the assembly output for the same function, compiled with x86-64 gcc 13.2. The assembly code includes stack frame setup, argument passing, multiplication, and return.

```
COMPILER EXPLORER Add... More Templates Do you have any suggestions, requests or bug reports? Feel free to contact us at anytime Share Policies
```

C source #1 x x86-64 gcc 13.2 (Editor #1) x

A Save/Load + Add new... Vim C

```
1 /* Type your code here, or load an example. */
2 int square(int num) {
3     return num * num;
4 }
5
```

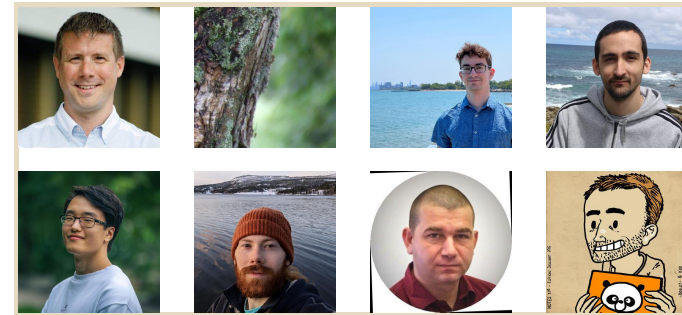
x86-64 gcc 13.2 Compiler options...

A Output... Filter... Libraries Overrides + Add new... Add tool...

```
1 square:
2     push    rbp
3     mov     rbp, rsp
4     mov     DWORD PTR [rbp-4], edi
5     mov     eax, DWORD PTR [rbp-4]
6     imul   eax, eax
7     pop     rbp
8     ret
```

NUMBERS

- Started ~10 years ago
- Team of 8
- 2000+ compilers
- 60+ languages
- 4M jobs/week
- ~2000\$/month



WHY THIS PRESENTATION?

- Showcase a (tiny!) subset of Compiler Explorer features
- Get feedback/ideas

BASIC USE CASES

- Single file/L
- Multiple files/L

```
Save/Load + Add new... Vim C
#include <stdio.h>

int main (int argc, char **argv) {
    printf ("Hello FOSDEM 2024\n");
    return 0;
}
```

```
x86-64 gcc 13.2 Compiler options...
Output... Filter... Libraries Overrides + Add new... + Add tool...
1 .LC0:
2     .string "Hello FOSDEM 2024"
3 main:
4     push    rbp
5     mov     rbp, rsp
6     sub     rsp, 16
7     mov     DWORD PTR [rbp-4], edi
8     mov     QWORD PTR [rbp-16], rsi
9     mov     edi, OFFSET FLAT:.LC0
10    call   puts
11    mov     eax, 0
12    leave
13    ret

Output (/0) x86-64 gcc 13.2 i - cached (3966B) ~254 lines filtered Compiler License
Output of x86-64 gcc 13.2 (Compiler #1)
A Wrap lines Select all
Compiler returned: 0
```

```
2
3 int main (int argc, char **argv) {
4     printf ("Hello FOSDEM 2024\n");
5     return 0;
6 }
7
```

```
1 .LC0:
2     .string "Hello FOSDEM 2024"
3 main:
4     push    rbp
5     mov     rbp, rsp
6     sub     rsp, 16
7     mov     DWORD PTR [rbp-4], edi
8     mov     QWORD PTR [rbp-16], rsi
9     mov     edi, OFFSET FLAT:LC0
10    call    puts
11    mov     eax, 0
12    leave
13    ret
```

Output (/0) x86-64 gcc 13.2 - 842ms (3966B) ~254 lines filtered Compiler License

Output of x86-64 gcc 13.2 (Compiler #1) ✕

A Wrap lines Select all

```
ASM generation compiler returned: 0
Execution build compiler returned: 0
Program returned: 0
Hello FOSDEM 2024
```

C x86-64 gcc 13.2 Compiler options...

Output... Filter... Libraries Overrides Add new... Add tool...

- Compile to binary object
- Link to binary
- Execute the code
- Intel asm syntax
- Demangle identifiers

```
rbp
5
rbp, rsp
48 83 ec 10
4 sub    rsp, 0x10
89 7d fc
8 mov    DWORD PTR [rbp-0x4], edi
48 89 75 f0
b mov    QWORD PTR [rbp-0x10], rsi
bf 00 00 00 00
f mov    edi, 0x0
10      R_X86_64_32 .rodata
e8 00 00 00 00
14 call   19 <main+0x19>
15      R_X86_64_PLT32 puts-0x4
b8 00 00 00 00
19 mov    eax, 0x0
c9
1e leave
```



```
x86-64 gcc 13.2      Compiler options...  
A ▾   Output... ▾   Filter... ▾   Libraries   Overrides  + Add new... ▾   Add tool... ▾  
 Compile to binary object  
 Link to binary  
 Execute the code  
 Intel asm syntax  
 Demangle identifiers  
401128 48 83 ec 10  rbp  
40112a sub    rsp,0x10  
40112c 89 7d fc  rbp,edi  
40112e mov    DWORD PTR [rbp-0x4],edi  
401130 48 89 75 f0  rbp,rsi  
401131 mov    QWORD PTR [rbp-0x10],rsi  
401133 bf 04 20 40 00  
401135 mov    edi,0x402004  
401137 e8 f1 fe ff ff  
40113a call  401030 <puts@plt>  
40113d b8 00 00 00 00  
40113f mov    eax,0x0  
401141 c9  
401144 leave  
401146 c3
```


Project + Add new... CMake Ada

Included files

- example.adb
- input
- foo.adb
- foo.ads

x86-64 gnat 13.2 (Tree #1)

x86-64 gnat 13.2 -gnat2022

```

A Output... Filter... Libraries Overrides + Add new... Add tool...
1 .LC1:
2     .ascii "Bar printing: "
3 .LC0:
4     .long 1
5     .long 14
6 foo__bar:
7     push    rbp          #
8     mov     rbp, rsp     #,
9     push    r14         #

```

Output (/0) x86-64 gnat 13.2 - cached (10618B) ~474 lines filtered Compiler License

Output of x86-64 gnat 13.2 (Compiler #1)

A Wrap lines Select all

```

Execution build compiler returned: 0
Program returned: 0
  Bar printing: Part1: 1444896
  Bar printing: Part2: 404395

```

Save/Load + Add new... Vim Ada

```

183 -- Part1: 1444896
184 Foo.Bar ("Part1: " & At_Most_100k'Image
185 -- Part2: 404395
186 Foo.Bar ("Part2: " & Found'Image);
187 end Example;
188

```

input

A Save/Load + Add new... Vim Ada

```

982 dir dtpzsrfc
983 214055 srgzhp.nlr
984 $ cd dtpzsrfc
985 $ ls
986 142652 bhgwj
987

```


MANY OTHER FEATURES

- cmake
- GPU (Nvidia) support
- diff
- libraries
- env overrides
- ISA documentation (x86 and riscv)
- many more

COMPILER DEV ORIENTED FEATURES

CONFORMANCE VIEW

- e.g. PR 113141: LINK/L/F/L
- EMPTY/L

Save/Load + Add new... Vim CppInsights Quick-bench C++

```
2
3 struct ConvToRef {
4     operator int&() { return global_x; }
5 };
6
7 struct Foo { int& r; };
8
9 int main()
10 {
11     Foo bar{ { ConvToRef{} } };
12 }
13
```

Compliance viewer (Editor #1) 17:15

Compiler	Status	Options	Actions
x86-64 gcc (trunk)	✖	Compiler options...	✖ ↶ ↷
x86-64 gcc 13.2	✖	Compiler options...	✖ ↶ ↷
x86-64 gcc 13.1	✖	Compiler options...	✖ ↶ ↷
x86-64 gcc 12.3	✔	Compiler options...	✖ ↶ ↷
x86-64 gcc 12.2	✔	Compiler options...	✖ ↶ ↷
x86-64 gcc 12.1	✔	Compiler options...	✖ ↶ ↷
x86-64 gcc 11.4	✔	Compiler options...	✖ ↶ ↷
x86-64 gcc 11.3	✔	Compiler options...	✖ ↶ ↷
x86-64 gcc 11.2	✔	Compiler options...	✖ ↶ ↷
x86-64 gcc 11.1	✔	Compiler options...	✖ ↶ ↷
x86-64 gcc 10.5	✔	Compiler options...	✖ ↶ ↷

EXPLORER

C source #1

Save/Load + Add new... Vim C

```
1 /* Type your code here, or load an example. */
2 int f (){}
3
```

Conformance Viewer (Editor #1) 21/15

+ Add compiler Libraries

x86-64 gcc (trunk)				Compiler options...
x86-64 gcc 13.2				Compiler options...
x86-64 gcc 13.1				Compiler options...
x86-64 gcc 12.3				Compiler options...
x86-64 gcc 12.2				Compiler options...
x86-64 gcc 12.1				Compiler options...

GIMPLE

LINK

- Recent addition: Thanks Jeremy!
- Available for all GCC ≥ 9

GIMPLE

GIMPLE source #1

POWER64 gcc 13.2.0 (Editor #1)

Save/Load Add new... Vim

GIMPLE

POWER64 gcc 13.2.0

-O0 -fPIC

```

1 int __GIMPLE (ssa)
2 square (int num)
3 {
4     int D_1916;
5     int _2;
6
7     __BB(2):
8     _2 = num_1(D) * num_1(D);
9     goto __BB3;
10
11    __BB(3):
12    return _2;
13 }
14

```

kvx

KALRAY MPPA GCC

- kvx ACB 4.12.0 (GCC 11.3.0)
- kvx ACB 4.11.1 (GCC 10.3.1)
- kvx ACB 4.10.0 (GCC 10.3.1)
- kvx ACB 4.9.0 (GCC 9.4.1)
- kvx ACB 4.8.0 (GCC 9.4.1)
- kvx ACB 4.6.0 (GCC 9.4.1)

```

8     stw 9,112(31)
9     lwz 9,112(31)
10    mullw 9,9,9
11    extsw 9,9
12    mr 3,9
13    addi 1,31,64
14    ld 31,-8(1)
15    blr
16    .long 0
17    .byte 0,0,0,0,128,1,0,1

```

TREE/RTL DUMPS

LINK/L/F/L

- display all tree/rtl dumps
- more can be added (e.g. Ada dumps/L)

The image shows a screenshot of the GCC Tree/RTL Viewer interface. It is divided into several panes:

- Source Code Pane (Top Left):** Displays the C source code for a `square` function:

```
1 int square(int num) {  
2     return num * num;  
3 }  
4
```
- Assembly Pane (Bottom Left):** Shows the assembly code for the `square` function:

```
1 square:  
2     imul    edi, edi  
3     mov     eax, edi
```
- Passes List Pane (Middle):** A list of compiler passes is shown, with `fix_loops (tree)` selected. Other passes include `dse5 (tree)`, `dce3 (tree)`, `forwprop3 (tree)`, `phiopt3 (tree)`, `ccp3 (tree)`, `powcabs (tree)`, `bswap (tree)`, `laddress (tree)`, `lim2 (tree)`, `walloca2 (tree)`, `pre (tree)`, `sink1 (tree)`, `dse3 (tree)`, `dce4 (tree)`, `slp2 (tree)`, `veclower21 (tree)`, `switchlower1 (tree)`, `sincos (tree)`, `reassoc2 (tree)`, `slsr (tree)`, `split-paths (tree)`, and `fre5 (tree)`.
- Options Menu (Right):** A context menu is open over the `fix_loops (tree)` entry, listing various options:
 - Raw Dump
 - Slim Dump
 - All Options
 - GIMPLE Frontend Syntax
 - Addresses
 - Basic Blocks
 - Line Numbers
 - Pass Details
 - Pass Stats
 - Unique IDs
 - Virtual Operands
- RTL Viewer Pane (Right):** Shows the RTL (Register Transfer Language) for the `square` function, including instructions like `CS of "...`, `D.2739`, and `1(D) =`.

NIGHTLY GCC COMPILERS

- GCC/master
 - x86, RISC-V 32/64, ARM, Aarch64, BPF, PowerPC
- External repositories
 - GCC+Cobol, GCCrs
- rustc_cg_gcc: rustc + gcc (libjit)
- Topic branches

A screenshot of GitHub Actions workflow runs for various GCC branches. The table shows the workflow name, the branch it runs on, the status of the workflow, and the last build time.

Workflow Name	Branch	Status	Last OK	Last build
gcc_contracts	gcc lock3-contracts-trunk	passing	Last OK	no result
gcc_contracts_labels	gcc lock3-contracts-labels-trunk	passing	Last OK	2024-01-25T00:22:49+00:00
gcc_contracts_nonattr	gcc contracts-nonattr-trunk	passing	Last OK	2024-01-25T00:48:19+00:00
gcc_coroutines	gcc cxx-coroutines-trunk	passing	Last OK	2023-11-07T23:48:40+00:00
gcc_gccrs_master	gcc gccrs-master	passing	Last OK	2024-01-25T00:27:15+00:00
gcc_modules	gcc cxx-modules-trunk	passing	Last OK	no result

API

- Access basic (compile/execute) features from
 - shell scripts
 - application or plugins (IDE, cli tool, ...)
 - web page

```
→ ce-rs git:(main) x ./target/debug/ce-rs compile --source "int f(){}" --flags "-Wall" --instruction-set kvx --summary
✓ Compilation "K1C gcc 7.4" (0)
✓ Compilation "K1C gcc 7.4 (obsolete)" (0)
✓ Compilation "K1C gcc 7.5" (0)
✓ Compilation "K1C gcc 7.5 (obsolete)" (0)
✓ Compilation "KVX ACB 4.1.0 (GCC 7.5.0)" (0)
✓ Compilation "KVX ACB 4.1.0 (GCC 7.5.0)" (0)
✓ Compilation "KVX ACB 4.1.0-cd1 (GCC 7.5.0)" (0)
✓ Compilation "KVX ACB 4.1.0-cd1 (GCC 7.5.0)" (0)
✓ Compilation "KVX ACB 4.10.0 (GCC 10.3.1)" (0)
✓ Compilation "KVX ACB 4.10.0 (GCC 10.3.1)" (0)
✓ Compilation "KVX ACB 4.10.0 (GCC 10.3.1)" (0)
✓ Compilation "KVX ACB 4.11.1 (GCC 10.3.1)" (0)
✓ Compilation "KVX ACB 4.11.1 (GCC 10.3.1)" (0)
✓ Compilation "KVX ACB 4.11.1 (GCC 10.3.1)" (0)
✓ Compilation "KVX ACB 4.12.0 (GCC 11.3.0)" (0)
✓ Compilation "KVX ACB 4.12.0 (GCC 11.3.0)" (0)
```

CREATE LOCAL/PRIVATE INSTANCE

⚠ perfect remote-execution service ⚠

- Quick setup is easy
- Multi-user setup needs care

WHAT'S NEXT FOR GCC

- opt view for GCC?
- better diff?
- more mingw compilers?

LINK

Opt Pipeline Viewer x86-64 clang (trunk) (Editor #1, Compiler #1) x86-64 clang (trunk) (Editor #1)

Function: square

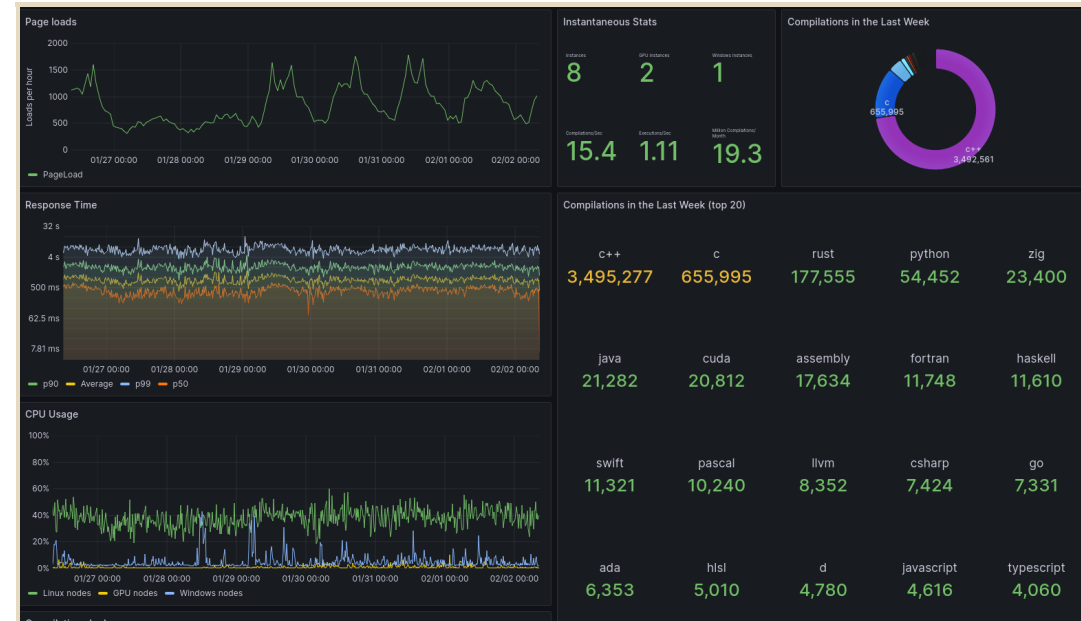
Passes:	Line	Code
Annotation2MetadataPass on [module]	1	1 define dso_local i32 @square(i32 noundef %num)
ForceFunctionAttrsPass on [module]	2	2 entry:
AssignmentTrackingPass on [module]	3	3 %num.addr = alloca i32, align 4
InferFunctionAttrsPass on [module]	4	4 store i32 %num, ptr %num.addr, align 4
CoroEarlyPass on [module]	5	5 %0 = load i32, ptr %num.addr, align 4
LowerExpectIntrinsicPass on square	6	6 %1 = load i32, ptr %num.addr, align 4
SimplifyCFGPass on square	7	7 %mul = mul nsw i32 %0, %1
SROAPass on square	3+	3+ tail call void @llvm.dbg.value(metadata i32, i32, DebugLoc: !0)
EarlyCSEPass on square	4+	4+ %mul = mul nsw i32 %num, %num
CallSiteSplittingPass on square	8	5 ret i32 %mul
OpenMPOptPass on [module]	9	6 }
IPSCCPass on [module]		
CalledValuePropagationPass on [module]		

CONCLUSION

- Showcased a (tiny!) subset of Compiler Explorer features
- Contributions are welcome (code, feature request, ...)

THANK YOU!

- dkm@kataplop.net
- IRC: dkm on libera.chat & oftc
- Discord: anyone in the team
- github.com/compiler-explorer



stats.compiler-explorer.com