



Tiny C Compiler

News

[Note: I am no longer working on TCC. Check the mailing list to get up to date information.]

(Feb 15, 2013) [TCC](#) version 0.9.26 is out thanks to Thomas Preud'homme ([Changelog](#)). Summary of the changes:

- Support for C99 VLA
- Generation of make dependencies (-MD/-MF)
- Support improved for various architectures (x86-64, arm, OSX, WinCE, kFreeBSD, Hurd)
- A bunch of bug fixes

(May 20, 2009) [TCC](#) version 0.9.25 is out thanks to Grischka ([Changelog](#)). TCC version 0.9.25 is the first that supports the x86-64 target. Thanks to Shinichiro Hamaji for this.

(Apr 1, 2008) [TCC](#) version 0.9.24 is out thanks to Grischka ([Changelog](#)). TCC now supports compilation from standard input and the arm eabi.

(Jun 17, 2005) [TCC](#) version 0.9.23 is out ([Changelog](#)). This is the first version with support for the Windows target.

(Nov 8, 2004) [TCC](#) version 0.9.22 is out ([Changelog](#)). Linux kernel compilation is 30% faster (10 seconds on a 2.4 GHz Pentium 4).

(Oct 25, 2004) [TCC](#) version 0.9.21 is out ([Changelog](#)). This version is the first one able to build a bootable Linux kernel with only a few patches to the kernel sources. As a demonstration, you can try the [TCCBOOT](#) boot loader. It is able to compile and boot a Linux kernel directly from its source code.

NOTE: if you want to compile the Linux kernel with TCC, you must use a custom build script as in [TCCBOOT](#). I never tried to compile the Linux kernel with TinyCC and the standard Linux Makefiles.

Features

- **SMALL!** You can compile and execute C code everywhere, for example on rescue disks (about 100KB for x86 TCC executable, including C preprocessor, C compiler, assembler and linker).
- **FAST!** tcc generates x86 code. No byte code overhead. Compile, assemble and link several times faster than gcc.
- **UNLIMITED!** Any C dynamic library can be used directly. TCC is heading toward full **ISOC99** compliance. TCC can of course compile itself.
- **SAFE!** tcc includes an optional **memory and bound checker**. Bound checked code can be

- mixed freely with standard code.
- Compile and execute C source directly. No linking or assembly necessary. Full C preprocessor and GNU-like assembler included.
- C script supported : just add '#!/usr/local/bin/tcc -run' at the first line of your C source, and execute it directly from the command line.
- With `libtcc`, you can use TCC as a backend for dynamic code generation.

Download

Compilation Speed

Compilation speed for the [Links Browser project](#). There are 76936 lines (including headers). 1950947 lines (67.2 MBytes) are compiled because the same headers are included in many files. TinyCC is about **9 times** faster than GCC.

| Compiler | Time(s) | lines/second | MBytes/second |
|---------------|---------|--------------|---------------|
| TinyCC 0.9.22 | 2.27 | 859000 | 29.6 |
| GCC 3.2 -O0 | 20.0 | 98000 | 3.4 |

Measures were done on a 2.4 GHz Pentium 4. Real time is measured. Compilation time includes compilation, assembly and linking.

More up to date tests are available: [1](#), [2](#), [3](#), [4](#).

Online Documentation

You want to help ?

Here are some suggestions:

- Report bugs to the mailing list (and eventually fix them).

Links

- [TinyCC mailing list](#)
- [Savannah project page and git repository](#)
- [OTCC - The smallest self compiling pseudo C compiler](#)
- [FFASN1](#) - My small but powerful ASN.1 compiler.
- [TinyCC fork](#) by Rob Landley
- [LLVM Compiler Infrastructure](#)
- [SmartEiffel](#) - With TCC you can compile your Eiffel code faster
- [C--](#) - An intermediate language for compilers
- [The GNU C Compiler](#)
- [The LCC Compiler](#)
- [The Small Device C Compiler](#)
- [Cyclone](#), A Safe Dialect of C
- [The D language](#)
- [Programming in C](#)
- The [Scriptometer](#) evaluates various scripting languages (including TCC).

License

TCC is distributed under the GNU Lesser General Public License.



Copyright (c) 2001-2009 Fabrice Bellard

Fabrice Bellard - <http://bellard.org/> - <http://www.tinycc.org/>