

Cross Compiler - Entwickeln von Windows-Anwendungen unter Linux (und mehr)

Wolfgang Dautermann

22. Dezember 2010

1 Motivation

Open-Source Software wird – wie der Name schon sagt – (auch) im Sourcecode zur Verfügung gestellt und kann meist recht einfach auf andere Plattformen portiert werden. Aber wie stellt man Binaries für andere Plattformen unter Linux her – und warum will man sowas überhaupt machen?

2 Geplante Struktur des Vortrags

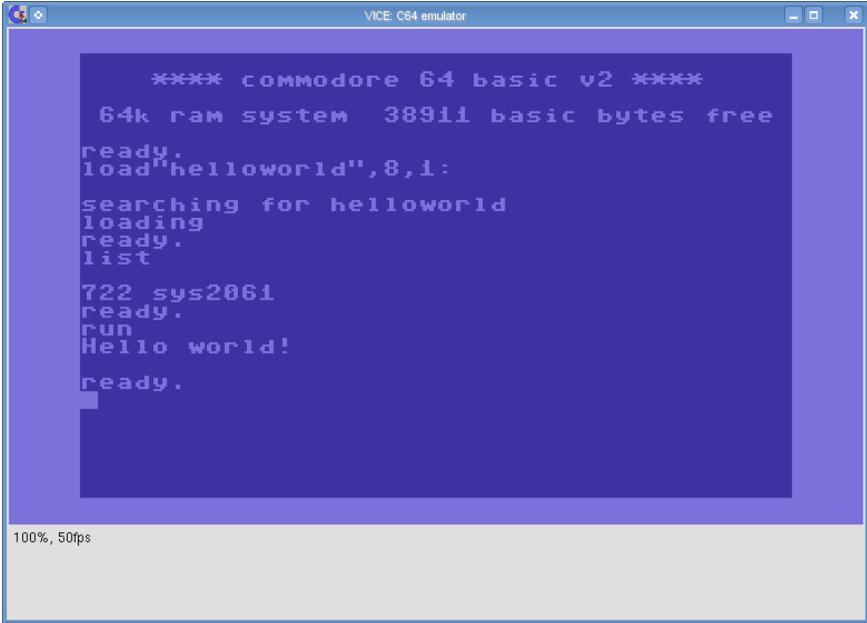
- Allgemeines zu Crosscompilern und Biarch-Compilern.
- Gründe, warum man Crosscompiler einsetzt.
- Compilieren eines Hello-World Programms für Windows und Testen der Anwendung mit Wine.
- Compilieren von umfangreicheren Programmen (mit zusätzlichen Libraries).
- Compiler für ältere Systeme (Bsp: cc65 für den Commodore 64)
- Compilieren eines Hello-World-Programms für den C64 und Test mit dem Emulator Vice.
- Sonstige Ergänzungen (Windows-Installationspakete, Buildsysteme, ...)

3 Zielgruppe

Entwickler und Programmierer, die daran interessiert sind, ihre Programme auch für Windows und andere Systeme zu übersetzen. Elementare C-Kenntnisse sind wünschenswert.

4 Praktische Vorführung

Eine kurze praktische Vorführung (Übersetzen eines Hello-World C-Programms für Windows, Commodore64), ... ist geplant.



The image shows a screenshot of a Commodore 64 emulator window titled "VICE: C64 emulator". The window displays a text-based interface with the following text:

```
**** commodore 64 basic v2 ****
64k ram system 38911 basic bytes free
ready.
load"helloworld",8,1:
searching for helloworld
loading
ready.
list
722 sys2061
ready.
run
Hello world!
ready.
```

At the bottom left of the emulator window, the text "100%, 50fps" is visible.

5 Literatur/Weblinks

- CC65 – the 6502 C Compiler <http://www.cc65.org/>
- Wikipedia-Seite: <http://de.wikipedia.org/wiki/Cross-Compiler>