

# Kurs: Programmierwerkzeuge unter Linux

© T. Birnthaler, OSTC GmbH ([www.ostc.de](http://www.ostc.de))

(Version 1.0 vom 22.11.2004)

**Dauer:** 2-3 Tage

**Motivation:** Sie wollen wissen, wie das Linux-System selbst (der Kernel), ein Hardware-Treiber, die unter Linux verwendeten GNU-Programme oder z.B. der Web-Server 'Apache', der Datenbank-Server 'MySQL' oder andere Linux-Programme wie 'xine', 'mplayer,... aus den Quellcode-Dateien übersetzt und installiert werden? Sie wollen Änderungen am Quellcode eines dieser Programme durchführen? Sie wollen eigene C/C++-Programme auf einem Linux-System erstellen und verwalten? Dann ist dieser Kurs genau richtig für Sie!

**Ziel:** Mittels vieler Übungen lernen Sie die typischen Programmierwerkzeuge von Linux zur Versionskontrolle, zur Programm-Erstellung, zum Debuggen, zum Profilieren und zum Testen kennen, die vor allem im Zusammenhang mit den klassischen Linux-Programmiersprachen C und C++ verwendet werden.

Im Rahmen dieses Kurses werden die Begriffe binäres Linux-Programme, Quellcode-datei, Projekt, Objektdatei, statische/dynamische Bibliothek, Ablauf eines Übersetzungsvorgangs, Revisionsverwaltung, Präprozessor, Compiler, Linker, Archivierer, Make, Makefile, Profiler, Debugger, ... genau erläutert und die damit jeweils verbundenen Linux-Werkzeuge behandelt.

Sie können anschließend unter Linux selbst oder von anderen entwickelte C/C++-Quellcodes sowohl auf der Kommandozeile als auch in einer IDE verwalten, übersetzen, analysieren und debuggen.

**Hinweis:** Der Kurs ist auch für andere UNIX-Systeme geeignet!

## **Inhalt:**

- Übersicht und Grundbegriffe:
- Informationsbeschaffung mit 'strings', 'strace/truss', 'size', 'time', 'ldd', 'nm', 'objdump' und 'readelf'

- Änderungsverwaltung mit 'diff' und 'patch'
- Revisionsverwaltung mit 'rcs' (Revision Control System)
- Parallele Programmentwicklung mit 'cvs' (Concurrent Revision System)
- Compilieren mit 'gcc' und 'g++'
- Linken und Archive bauen mit 'ld', 'ar', 'ranlib' und 'strip'
- Programme bauen mit 'make'
- Software automatisch konfigurieren mit 'autoconf' und 'automake'
- Fehlersuche mit 'gdb' und 'ddd'
- Statische Programmanalyse mit 'lint'
- Laufzeitanalyse mit 'gprof' und 'gcov'
- Einblick in die integrierten Entwicklungsumgebungen (IDEs) 'Kdevelop', 'Kdebug', 'Eclipse' und 'Komodo'

#### **Schulungsunterlagen:**

- Buch
- Lösungsblätter zu allen Übungen

#### **Voraussetzungen:**

- Muss: Allgemeine Computer-Kenntnisse (Tastatur, Maus, Grafische Oberfläche, Editor)
- Muss: Linux/UNIX-Kenntnisse im Rahmen des Kurses "Linux/UNIX-Einführung"
- Muss: Kenntnis einer beliebigen Programmiersprache (vorzugsweise C, C++, Java oder C#)
- Optional: Kenntnis einer Integrierten Entwicklungsumgebung (IDE)

#### **Abgrenzung:**

Aufgrund der dafür notwendigen Zeit werden folgende Punkte in diesem Kurs nicht behandelt, können aber extra gebucht werden:

- Programmierung in C, C++, ... an sich
- Compilerbau (lexikalische Analyse mit 'lex/flex' und syntaktische Analyse mit 'yacc/bison')
- Prototypen erstellen (mit 'Tcl/Tk', 'Perl/Tk' oder 'Python')
- Portables Programmieren (Qt, GTK, GTK+, Windows, Linux, MacOS)
- Portierung von Windows-Programmen nach Linux