

Softwareentwicklung

Tools

Wichtige Hilfsmittel für die Softwareentwicklung:

Editoren, z.B:

- vi(m): textbasiert, fast auf jedem Linux system installiert
- nano: textbasiert
- (x)emacs: graphisch, sehr mächtig

Compiler+Linker bzw. Interpreter:

- g++, f77, python, java, ...

Make, Debugger, Profiler:

- make, gdb, gprof, ...

Codeverwaltung:

- svn, git, ...

Integrierte Entwicklungsumgebungen (IDE):

- KDevelop, Eclipse, (emacs),
kommerzielle Produkte, ...

g++ (gcc)

- GNU compiler collection
- Compiler und Linker für C++ (und andere Sprachen)
- Verarbeitungsschritte:
 - Quellcode
 - Kompilieren: Maschinencode (Objektdatei)
 - Linken: Zusammenfügen von Objektdateien
 - ausführbares Programm oder statische oder dynamische Bibliothek
- Wichtige Optionen:
 - c: kompilieren
 - o: Ausgabedatei
 - I: Pfad für eingebundene Dateien (include)
 - L, -l: Pfad und Name von Bibliotheken
 - shared, -g, -O, -Wall, -fPIC: dynamische Bibliothek, Debugsymbole, Optimierung, Warnungen, positionsunabhängiger Code

make

- Ausführen von Befehlen in Abhängigkeit von Änderungen an Dateien
- Insbesondere verwendet für Kompiler/Linker
- Dateiname: GNUmakefile, makefile, Makefile (oder angegeben mit -f)

- Aufbau aus Regeln:

```
target: dependency  
(tab)    command
```

- Implizite Abhängigkeiten:

```
.cc.o
```

- Definition von Variablen:

```
VAR = value  
SOURCES = $(wildcard *.cc)  
DIR = $(shell /bin/pwd)
```

- Automatische Ermittlung von Abhängigkeiten:

```
makedepend
```

Weitere Build-Tools:

- cmake, scons, ...

- Debugger zum Auffinden von Laufzeitfehlern
- Ausgabe der Position in Programm: **backtrace**, **bt**
- Ausgabe von Variablenwerten: **print**, **p**
- Setzen von Haltepunkten: **break**
- Schrittweises Ausführen: **next**, **n**, **step**, **s**
- Programm muss mit Option **-g** kompiliert sein, um Debugsymbole zu enthalten
- Kann **coredump**-Dateien lesen

- Profiler zum Messen der Ausführungszeit von Programmteilen
- Programm muss mit Option `-pg` kompiliert und gebunden werden
- nach Ausführen des Programms steht die Profilerinformation in der Datei `gmon.out`

- Viele weitere Tools verfügbar,
z.B. zur Überwachung der Speicherverwendung (memory leaks)

- „Subversion“
- Verwaltung von Quellcode (und anderen Dateien)
- Code wird in zentralen „Repository“ gespeichert
- Mehrere Benutzer können gleichzeitig
 - eine lokale Kopie anlegen
 - Dateien ändern
 - die Änderungen zurückschreiben
- Konflikte werden automatisch erkannt und wenn möglich gelöst
- Alle Änderungen werden protokolliert, alte Versionen können wiederhergestellt werden
- Versionen des Codes können als Zweige verwaltet werden
- Verteilte Verwaltungssysteme: git, mercurial, ...