

Inhaltsverzeichnis

1	Programmieren unter Linux	1
1.1	Das Unix-Betriebssystem	1
1.1.1	Die Unix-Familie	1
1.1.2	Besondere Eigenschaften von Unix	2
1.1.3	Die Werkzeug-Philosophie	3
1.2	Linux	4
1.2.1	Der Kernel	4
1.2.2	Distributionen	5
1.2.3	Kompatibilität und Portabilität	6
1.3	Kommerzielle und freie Software	7
1.3.1	Das GNU-Projekt	8
1.3.2	Die GNU General Public License	9
1.3.3	Andere Open-Source-Ansätze	11
1.3.4	Vorteile von Open Source	12
1.3.5	Motivation für Open Source	13
1.3.6	Fazit	15
1.4	Programmentwicklung in Unix	16
1.4.1	Wichtige Begriffe	17
1.4.2	Systemdateien zur Entwicklung	19
1.5	Übungsfragen	21
2	Grundlagen der objektorientierten Programmierung in C++	23
2.1	Grundideen	24
2.1.1	Beherrschung der Komplexität	24
2.1.2	Rückblick auf strukturiertes Programmieren	25
2.1.3	Objekte	28
2.1.4	Klassen	30
2.1.5	Methoden und Prozessabstraktion	32
2.1.6	Datenabstraktion	33
2.1.7	Zusammenfassung	34

2.1.8	Übungsaufgaben	35
2.2	Die C++-Programmiersprache	35
2.2.1	Historisches	36
2.2.2	C++ und C	37
2.2.3	C++ und Linux	37
2.2.4	Neuerungen im GCC 3.0	39
2.2.5	Das erste C++-Programm	40
2.2.6	Datentypen und Typumwandlung	45
2.2.7	Operatoren	50
2.2.8	Ausdrücke	54
2.2.9	Zusammenfassung	55
2.2.10	Übungsaufgaben	56
2.3	Umgang mit dem GNU-C++-Compiler	57
2.3.1	Installation	57
2.3.2	Aufruf und Optionen	57
2.3.3	Name für die ausführbare Datei	58
2.3.4	Debug-Informationen	59
2.3.5	Fehler und Warnungen	59
2.3.6	Kompilierung zur Objektdatei	61
2.3.7	Pfade zu Header-Dateien	61
2.3.8	Bibliotheken	62
2.3.9	Optimierung	63
2.3.10	Info-Seiten	64
2.3.11	Editieren mit <i>NEdit</i> und anderen	66
2.3.12	Zusammenfassung	68
2.3.13	Übungsaufgaben	69
2.4	Klassen und Objekte	70
2.4.1	Klassendeklaration und -definition	70
2.4.2	Objekte von Klassen	73
2.4.3	Zugriffsbeschränkungen	74
2.4.4	Freunde	76
2.4.5	Zusammenfassung	77
2.4.6	Übungsaufgaben	78
2.5	Funktionen und Methoden	78
2.5.1	Funktionen in C++	78
2.5.2	Der Prototyp	82
2.5.3	Überladen von Funktionen	84
2.5.4	Überladen von <code>main()</code>	86
2.5.5	Vorgabewerte für Parameter	87
2.5.6	Referenzen und Parameterübergabe	89
2.5.7	Zugriffsroutinen	95

2.5.8	Inline-Funktionen	96
2.5.9	Zusammenfassung	101
2.5.10	Übungsaufgaben	102
2.6	Konstruktoren und Destruktoren	103
2.6.1	Überblick über Konstruktoren	104
2.6.2	Standardkonstruktor	106
2.6.3	Allgemeine Konstruktoren	108
2.6.4	Initialisierung mit Listen	111
2.6.5	Kopierkonstruktor	113
2.6.6	Typumwandlungskonstruktor	115
2.6.7	Destruktoren	119
2.6.8	Beispiel: Benutzerinformationen	120
2.6.9	Zusammenfassung	126
2.6.10	Übungsaufgaben	126
2.7	Vererbung und Polymorphismus	127
2.7.1	Basisklassen und abgeleitete Klassen	127
2.7.2	Vererbung in C++	129
2.7.3	Erzeugung von Unterklassenobjekten	135
2.7.4	Zugriffsbeschränkungen	138
2.7.5	Mehrfachvererbung	141
2.7.6	Polymorphismus	143
2.7.7	Rein virtuelle Funktionen und abstrakte Klassen	148
2.7.8	Zusammenfassung	150
2.7.9	Übungsaufgaben	151
3	Programmieren mit C++	153
3.1	Basiselemente	154
3.1.1	Bedingungen	154
3.1.2	Mehrfache Auswahl	161
3.1.3	Schleifen	168
3.1.4	Zusammenfassung	183
3.1.5	Übungsaufgaben	184
3.2	Dateien und Ströme	184
3.2.1	Standardein- und -ausgabe	184
3.2.2	Ein- und Ausgabe mit Dateien	187
3.2.3	Positionierung des Dateizeigers	193
3.2.4	Ausgabeformatierung	195
3.2.5	Beispiel: Umrechnung DM – Euro	196
3.2.6	Zusammenfassung	206
3.2.7	Übungsaufgaben	207

3.3	Felder, Zeiger und dynamische Speicherverwaltung	208
3.3.1	Felder (Arrays)	208
3.3.2	Zeichenketten	211
3.3.3	Zeiger	213
3.3.4	Dynamische Speicherverwaltung	218
3.3.5	Konstruktoren und Destruktoren	225
3.3.6	Beispiel: CGI-Programmierung	228
3.3.7	Zusammenfassung	244
3.3.8	Übungsaufgaben	245
3.4	Die C-Bibliothek	247
3.4.1	Umfang der C-Bibliothek	247
3.4.2	Das <i>man</i> -Kommando	249
3.4.3	Mathematische Standardfunktionen (cmath)	251
3.4.4	Numerische Limits (climits und cfloat)	253
3.4.5	Auswertung von Fehlern bei Bibliotheksfunktionen (cerrno)	254
3.4.6	Behandlung von Signalen (csignal)	256
3.4.7	Allgemeine Utilities (cstdlib)	261
3.4.8	Ein- und Ausgabefunktionen (cstdio)	266
3.4.9	Zugriff auf und Manipulation von char-Strings (cstring)	266
3.4.10	Zusammenfassung	268
3.4.11	Übungsaufgaben	268
3.5	Eigene Bibliotheken	269
3.5.1	Statische Bibliotheken	270
3.5.2	Beispiel: Zugriff auf Verzeichnisse	274
3.5.3	Dynamische Bibliotheken (shared libraries)	279
3.5.4	Zusammenfassung	284
3.5.5	Übungsaufgaben	285
3.6	Tipps und Konventionen	286
3.6.1	Namenskonventionen	287
3.6.2	Projektorganisation	289
3.6.3	Programmierstil	290
3.6.4	Sicheres Programmieren	291
3.6.5	C++-Programmierstil	292
3.6.6	Zusammenfassung	294
4	Fortgeschrittenes C++	295
4.1	Namensräume	295
4.1.1	Definition	296
4.1.2	Zugriff auf Bezeichner in Namensräumen	298

4.1.3	Zusammenfassung mehrerer Namensräume	301
4.1.4	Verschachtelte Namensräume	301
4.1.5	Zusammenfassung	302
4.1.6	Übungsaufgaben	302
4.2	Templates	303
4.2.1	Funktionstemplates	304
4.2.2	Organisation des Quelltextes	307
4.2.3	Klassentemplates	308
4.2.4	Zusammenfassung	317
4.2.5	Übungsaufgaben	319
4.3	Die STL: die Containerklassen der C++-Standardbibliothek	320
4.3.1	Namenskonventionen	321
4.3.2	Strings	322
4.3.3	Container	324
4.3.4	Iteratoren	329
4.3.5	Algorithmen	331
4.3.6	Zusammenfassung	334
4.3.7	Übungsaufgaben	335
4.4	Operatoren zur Typumwandlung	335
4.4.1	Der static_cast-Operator	336
4.4.2	Der dynamic_cast-Operator	337
4.4.3	Der const_cast-Operator	339
4.4.4	Der reinterpret_cast-Operator	340
4.4.5	Zusammenfassung	342
4.4.6	Übungsaufgaben	343
4.5	Überladen von Operatoren	343
4.5.1	Operatorfunktionen und -methoden	345
4.5.2	Arten von Operatoren	346
4.5.3	Der Indexoperator	347
4.5.4	Der Inkrementoperator	350
4.5.5	Der Zuweisungsoperator	351
4.5.6	Vergleiche und mathematische Operatoren	355
4.5.7	Operatoren als Freunde	357
4.5.8	Ein- und Ausgabeoperator	358
4.5.9	Typumwandlungsoperator	359
4.5.10	Allgemeine Prinzipien	361
4.5.11	Zusammenfassung	364
4.5.12	Übungsaufgaben	365
4.6	Ausnahmebehandlung (Exceptions)	366
4.6.1	Behandlung von Fehlersituationen	367

4.6.2	Exception Handling	368
4.6.3	Allgemeine Syntax	369
4.6.4	Auffangen der Ausnahmen	370
4.6.5	Beispiel: Vektor	374
4.6.6	Exceptions und die Standardbibliothek	377
4.6.7	Tipps und Hinweise	378
4.6.8	Zusammenfassung	379
4.6.9	Übungsaufgaben	380
5	Editoren für die Programmierung	381
5.1	vi improved	382
5.1.1	Starten und Beenden	383
5.1.2	Die Bearbeitungsmodi	384
5.1.3	Bewegen des Cursors	385
5.1.4	Text schreiben und löschen	386
5.1.5	Suchen und Ersetzen	388
5.1.6	Speichern und Laden	389
5.1.7	Kopieren und Verschieben	389
5.1.8	Weitere Befehle	390
5.1.9	Zusammenfassung	391
5.2	Der XEmacs-Editor	391
5.2.1	Grundlegende Befehle	392
5.2.2	Suchen und Ersetzen	399
5.2.3	Ausschneiden, Kopieren und Einfügen	401
5.2.4	Modi	403
5.2.5	Fazit	405
5.3	Weitere Editoren	405
5.3.1	Glimmer	405
5.3.2	KWrite	409
5.3.3	XCoral	411
5.3.4	Und was sonst?	415
6	Werkzeuge für die Softwareentwicklung	417
6.1	Steuerung der Übersetzung mit Make-Dateien	418
6.1.1	Aufbau von Makefiles	419
6.1.2	Arbeiten mit <i>make</i>	423
6.1.3	Makros	424
6.1.4	Eingebaute Regeln	425
6.1.5	<i>make</i> für Fortgeschrittene	428
6.1.6	Zusammenfassung	432

6.2	Fehlersuche mit dem Debugger	433
6.2.1	Theoretische Fehlerquellen	433
6.2.2	Statusausgaben im Code	437
6.2.3	Ein fehlerhaftes Programm	438
6.2.4	Fehlersuche mit <i>gdb</i>	440
6.2.5	Der grafische Debugger <i>DDD</i>	453
6.2.6	Fehlervermeidung durch die Überprüfung von Vorbedin- gungen	463
6.2.7	Zusammenfassung	466
6.3	Versionskontrolle mit <i>RCS</i> und <i>CVS</i>	467
6.3.1	Versionskontrolle mit Linux	468
6.3.2	Vorgehensweise	469
6.3.3	Versionsverwaltung mit <i>RCS</i>	470
6.3.4	Versionsverwaltung mit <i>CVS</i>	478
6.3.5	Zusammenfassung	496
7	Integrierte Entwicklungsumgebungen	499
7.1	XEmacs als IDE	500
7.1.1	Der Editor	500
7.1.2	Start des Compilers	501
7.1.3	Start des Programms und des Debuggers	502
7.1.4	Versionsverwaltung mit XEmacs	503
7.1.5	Dateivergleich mit Ediff	504
7.1.6	Zusammenfassung	506
7.2	KDevelop	506
7.2.1	Überblick	506
7.2.2	Installation und Konfiguration	507
7.2.3	Die <i>KDevelop</i> -Entwicklungsumgebung	509
7.2.4	Mit Konsolanwendungen arbeiten	515
7.2.5	KDE-Anwendungen mit KDevelop entwickeln	519
7.2.6	Zusammenfassung	535
7.3	Source Code Engineering mit <i>SNiFF+</i>	536
7.3.1	Projekte in <i>SNiFF+</i>	536
7.3.2	Überblick über die <i>SNiFF</i> -Umgebung	538
7.3.3	Editieren, Kompilieren und Debuggen	540
7.3.4	Zusammenfassung	544
7.4	Weitere Entwicklungsumgebungen	545
7.4.1	Source Navigator	546
7.4.2	CodeWarrior	547
7.4.3	Eclipse	549
7.4.4	Zusammenfassung	554

Literaturverzeichnis	555
Verzeichnis der Abkürzungen	559
Index	561