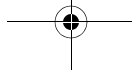
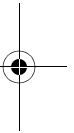
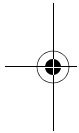

CONTENTS

Acknowledgments	xvii
Chapter 1 Introduction to Software Development	1
1.1 Life Cycle of a Software Development Project	2
1.1.1 Requirement Gathering	2
1.1.2 Writing Functional Specifications	4
1.1.3 Creating Architecture and Design Documents	4
1.1.4 Implementation and Coding	5
1.1.5 Testing	6
1.1.6 Software Releases	8
1.1.7 Documentation	8
1.1.8 Support and New Features	9
1.2 Components of a Development System	10
1.2.1 Hardware Platform	10
1.2.2 Operating System	11
1.2.3 Editors	11
1.2.4 Compilers and Assemblers	12
1.2.5 Debuggers	12
1.2.6 Version Control Systems	12
1.2.7 E-mail and Collaboration	13



1.2.8	X-Windows	13
1.3	Selection Criteria for Hardware Platform	13
1.4	Selection Criteria for Software Development Tools	14
1.5	Managing Development Process	14
1.5.1	Creating Deadlines	14
1.5.2	Managing the Development Team	15
1.5.3	Resolving Dependencies	15
1.6	Linux Development Platform Specifications (LDPS) and Linux Standard Base (LSB)	15
1.6.1	Libraries	15
1.6.2	Current Contributors to LSB	16
1.7	References	16
Chapter 2 Working With Editors		17
2.1	What to Look for in an Editor	17
2.1.1	Extensibility	17
2.1.2	Understanding Syntax	18
2.1.3	Tag Support	18
2.1.4	Folding Code	18
2.2	Emacs	18
2.2.1	Using Emacs	19
2.2.2	Basic Emacs Concepts	20
2.2.3	Using Buffers and Windows	24
2.2.4	Language Modes	26
2.2.5	Using Tags	27
2.2.6	Compiling	30
2.2.7	Xemacs	32
2.3	Jed	32
2.3.1	Configuring Jed	33
2.3.2	Using Jed	34
2.3.3	Folding Code	35
2.4	VIM	37
2.4.1	VIM Concepts	38
2.4.2	Basic Editing	38
2.4.3	Using Tags with VIM	41
2.4.4	Folding Code	42
2.5	References and Resources	42



Chapter 3	Compilers and Assemblers	43
3.1	Introduction to GNU C and C++ Compilers	44
3.1.1	Languages Supported by GCC	44
3.1.2	New Features in GCC 3.x	46
3.2	Installing GNU Compiler	48
3.2.1	Downloading	48
3.2.2	Building and Installing GCC	48
3.2.3	Environment Variables	54
3.2.4	Post-Installation Tasks	56
3.2.5	What Not to Do when Installing Development Tools	58
3.3	Compiling a Program	58
3.3.1	Simple Compilation	59
3.3.2	Default File Types	60
3.3.3	Compiling to Intermediate Levels	61
3.3.4	Compilation with Debug Support	63
3.3.5	Compilation with Optimization	64
3.3.6	Static and Dynamic Linking	65
3.3.7	Compiling Source Code for Other Languages	66
3.3.8	Summary of gcc Options	70
3.4	Linking a program	91
3.5	Assembling a Program	91
3.6	Handling Warning and Error messages	92
3.7	Include files	92
3.8	Creating Libraries	92
3.9	Standard Libraries	93
3.10	Compiling Pascal Programs	94
3.10.1	Using Free Pascal (fpc)	95
3.10.2	Using GNU Pascal	96
3.11	Compiling Fortran Programs	96
3.12	Other Compilers	98
3.12.1	Smalltalk	98
3.12.2	Oberon	98
3.12.3	Ruby	98
3.13	References and Resources	98
Chapter 4	Using GNU make	101
4.1	Introduction to GNU make	102
4.1.1	Basic Terminology	103

4.1.2	Input Files	105
4.1.3	Typical Contents of a Makefile	106
4.1.4	Running make	108
4.1.5	Shell to Execute Commands	109
4.1.6	Include Files	109
4.2	The make Rules	110
4.2.1	Anatomy of a Rule	110
4.2.2	A Basic Makefile	111
4.2.3	Another Example of Makefile	113
4.2.4	Explicit Rules	118
4.2.5	Implicit Rules	118
4.3	Using Variables	119
4.3.1	Defining Variables	120
4.3.2	Types of Variables	120
4.3.3	Pre-Defined Variables	121
4.3.4	Automatic Variables	121
4.4	Working with Multiple Makefiles and Directories	122
4.4.1	Makefile in The Top Directory	123
4.4.2	Makefile in common-dir Directory	125
4.4.3	Makefile in the ftp-dir Directory	126
4.4.4	Makefile in the tftp-dir Directory	127
4.4.5	Makefile in the dns-dir Directory	127
4.4.6	Building Everything	128
4.4.7	Cleaning Everything	129
4.4.8	Making Individual Targets	129
4.5	Special Features of make	130
4.5.1	Running Commands in Parallel	130
4.5.2	Non-Stop Execution	130
4.6	Control Structures and Directives	131
4.6.1	The ifeq Directive	132
4.6.2	The ifneq Directive	132
4.6.3	The ifdef Directive	132
4.6.4	The ifndef Directive	133
4.6.5	The for Control Structure	133
4.7	Getting the Latest Version and Installation	133
4.7.1	Compilation	133

4.7.2	Installation	134
4.8	References and Resources	134
Chapter 5	Working with GNU Debugger	135
5.1	Introduction to GDB	136
5.2	Getting Started with GDB	136
5.2.1	Most Commonly Used gdb Commands	137
5.2.2	A Sample Session with gdb	138
5.2.3	Passing Command Line Arguments to the Program Being Debugged	141
5.3	Controlling Execution	144
5.3.1	The step and finish Commands	144
5.4	Working with the Stack	146
5.5	Displaying Variables	151
5.5.1	Displaying Program Variables	151
5.5.2	Automatic Displaying Variables with Each Command	153
5.5.3	Displaying Environment Variables	154
5.5.4	Modifying Variables	155
5.6	Adding Break Points	156
5.6.1	Continuing from Break Point	158
5.6.2	Disabling Break Points	159
5.6.3	Enabling Break Points	159
5.6.4	Deleting Break Points	160
5.7	Debugging Optimized Code	160
5.8	Files and Shared Libraries	163
5.9	Using gdb With GNU Emacs	164
5.10	Debugging Running Processes	165
5.11	Installing GDB	168
5.11.1	Downloading and Building	168
5.11.2	Final Installation	168
5.12	Other Open Source Debuggers	169
5.12.1	The kdbg Debugger	169
5.12.2	The ddd Debugger	172
5.12.3	The xxgdb Debugger	173
5.13	References and Resources	174

Chapter 6 Introduction to CVS	175
6.1 CVS Policies	176
6.2 Project Management and Communication	176
6.3 Installing and Managing CVS	176
6.3.1 Configuring CVS	177
6.3.2 Importing a Project into the Repository	179
6.4 Using the CVS Client	180
6.4.1 Local Repositories	181
6.4.2 Remote Repositories	182
6.4.3 Checking out a Project	182
6.4.4 Finding the Status of a Project	183
6.4.5 Finding Differences	184
6.4.6 Resolving Conflicts	185
6.4.7 Checking the Project Back In	186
6.4.8 Adding Files to a Project	186
6.4.9 Removing Files from a Project	187
6.4.10 Renaming Files within a Project	188
6.4.11 Removing your Working Copy	188
6.4.12 Tags and Releases	189
6.5 Introduction to jCVS	190
6.5.1 System Requirements	190
6.5.2 Installation Instructions	190
6.5.3 Using jCVS	191
6.6 Using Emacs with CVS	196
6.6.1 Installing pcl-cvs	197
6.6.2 Using pcl-cvs	197
6.7 Secure remote access with CVS	199
6.7.1 Secure Shell Access	199
6.8 References and Resources	201
Chapter 7 Miscellaneous Tools	203
7.1 Using indent Utility	204
7.1.1 Getting Started with Indent	205
7.1.2 Selecting Coding Styles	206
7.1.3 Blank Lines and Comments	209
7.1.4 Formatting Braces	210
7.1.5 Formatting Declarations	211

7.1.6	Breaking Long Lines	212
7.1.7	Summary of Options	213
7.2	Using sed Utility	215
7.3	Using diff Utility	215
7.3.1	Other Forms of diff Utility	218
7.4	Using cscope and cbrowser	219
7.5	Generating C Function Prototypes from C Source Code	
Using cproto		222
7.6	Using ltrace and strace Utilities	223
7.7	Using GNU Binary Utilities	226
7.7.1	Using the ar Utility	226
7.7.2	Using the ranlib Utility	228
7.7.3	Using the nm Utility	228
7.7.4	Using the strip Utility	231
7.7.5	Using the objcopy Utility	231
7.7.6	Using the objdump Utility	232
7.7.7	Using the size Utility	236
7.7.8	Using the strings Utility	237
7.7.9	Using the addr2line Utility	237
7.8	Using the ldd Utility	238
7.9	References and Resources	238
Chapter 8 Cross-Platform and Embedded Systems Development		239
8.1	Introduction to the Cross-Platform Development Process	240
8.1.1	Host Machine	240
8.1.2	Target Machine	240
8.1.3	Native and Cross Compilers	241
8.1.4	Cross Platform Development Cycle	241
8.2	What are Embedded Systems?	243
8.2.1	Embedded Systems and Moving Parts	244
8.2.2	Embedded Systems and Power Consumption	245
8.2.3	Embedded Operating Systems	245
8.2.4	Software Applications for Embedded Systems	246
8.3	How Development Systems Differ for Embedded Systems	246
8.3.1	Knowledge of Target System Hardware	246
8.3.2	Is the Target System Real-Time?	247

8.3.3	Testing Methodology	247
8.4	Cross Compilations	247
8.4.1	Software Emulators	248
8.4.2	In-circuit emulators	249
8.4.3	Introduction to JTAG and BDM	249
8.5	Connecting to Target	250
8.5.1	Using gdbserver with GNU Debugger	250
8.5.2	Attaching to a Running Process Using gdbserver	255
8.5.3	Using Stubs with GNU Debugger	256
8.5.4	Debugging the Debug Session	256
8.6	Hardware Used for Cross Platform and Embedded Systems Development	258
8.6.1	Arcom SBC-GX1 Board	258
8.6.2	Artesyn PM/PPC Mezzanine Card	260
8.7	References	261
Chapter 9 Platform Independent Development with Java		263
9.1	How Java Applications Work	264
9.1.1	Java Compiler	264
9.1.2	Java Virtual Machine	264
9.2	Kaffe	264
9.3	The Jboss Java Development System	266
9.4	Java 2 SDK	267
9.4.1	Java 2 SDK Standard Edition	267
9.4.2	Getting and Installing Java SDK from Sun	269
9.4.3	Creating jar Files	269
9.5	Building Java Applications	270
9.5.1	Creating Source Code File	270
9.5.2	Compiling Java Code	270
9.5.3	Running Java Applications	271
9.5.4	Using gcj to Build Java Applications	271
9.6	Building Applets	271
9.7	Testing Applets with Netscape	272
9.8	Jikes for Java	272
9.9	Miscellaneous	274
9.9.1	Embedded Java	274
9.9.2	Real Time Java	274



Contents

xv

9.9.3	Wireless Applications	275
9.10	References	275
Appendix A	Typical Hardware Requirements for a Linux Development Workstation	277
	Index	283

