Pro Linux Embedded Systems

Gene Sally

Apress®
For Tammi and our sons, Jonah, Benjamin, and Adam.
Contents at a Glance

Contents at a Glance ................................................................. iv
Contents .................................................................................... v
About the Author ...................................................................... xvi
About the Technical Reviewer ............................................... xvii
Acknowledgments .................................................................. xviii
Introduction ............................................................................... xix
Chapter 1: About Embedded Linux ........................................ 1
Chapter 2: Configuring the Software Environment ............... 25
Chapter 3: Target Emulation and Virtual Machines .............. 55
Chapter 4: Starting Your Project .......................................... 63
Chapter 5: Getting Linux for Your Board .......................... 83
Chapter 6: Creating a Linux Distribution from Scratch ....... 107
Chapter 7: Booting the Board ............................................. 143
Chapter 8: Configuring the Application Development Environment ... 169
Chapter 9: Application Development ................................. 197
Chapter 10: Debugging Applications ................................ 215
Chapter 11: Kernel Configuration and Development ............ 237
Chapter 12: Real Time ......................................................... 257
Chapter 13: Using Open Source Software Projects ......... 273
Chapter 14: BusyBox ........................................................... 293
Chapter 15: System Design .................................................. 309
Chapter 16: System Tuning ................................................... 335
Chapter 17: Deploying Applications .................................. 363
Chapter 18: Handling Field Updates ................................. 383
# Contents

## Contents at a Glance

- iv

## Contents

- v

## About the Author

- xvi

## About the Technical Reviewer

- xvii

## Acknowledgments

- xviii

## Introduction

- xix

### Chapter 1: About Embedded Linux

- 1

#### Why Use Embedded Linux?

- 2

##### Technical Reasons to Use Embedded Linux

- 2

##### Commercial Reasons to Use Embedded Linux

- 7

#### 10,000-Foot Embedded Linux Development Flyover

- 9

##### Target Hardware

- 9

##### Obtaining Linux

- 10

##### Booting Linux

- 10

##### Development Environment

- 10

##### System Design

- 11

#### Anatomy of an Embedded Linux System

- 11

##### Boot Loader

- 12

##### Kernel

- 13

##### Root File System

- 13

##### Your Application

- 14

##### Cross-Compiler

- 14

##### Tools of the Trade

- 15
Where to Get Help ......................................................................................................... 19
  University of Google ........................................................................................................... ..................19
  Mailing Lists and Newsgroups................................................................................................... ..........19
  Vendor-Sponsored Resources..................................................................................................... .........20
  Trade Group and Community Interest Sites ....................................................................................... ..21
  IRC..................................................................................................................................................23

Next Up .........................................................................................................................2 3

Chapter 2: Configuring the Software Environment ................................................. 25
  Host Environment.......................................................................................................... 26
    Linux .........................................................................................................................................26
    Windows .....................................................................................................................................29
  Host Services ................................................................................................................ 43
    Turn Off Your Firewall ......................................................................................................... .................44
    TFTP ..........................................................................................................................................44
    DHCP .........................................................................................................................................45
    NFS...............................................................................................................................................47
    PXE...............................................................................................................................................50
  Cabling .........................................................................................................................................51
    Serial (for Console)........................................................................................................... ....................51
    Network.......................................................................................................................................51
  Avoiding an Angry Visit from IT ............................................................................................. 52
    Dual-Homed Host.................................................................................................................................52

Chapter 3: Target Emulation and Virtual Machines ............................................. 55
  Why Target Emulation? ................................................................................................. 55
  Emulation via QEMU................................................................................................................. ..................56
    Compiling QEMU........................................................................................................... ....................56
    Using QEMU to Emulate a Target ......................................................................................... .................58
    Using QEMU to Compile under Emulation ........................................................................... ..................60
Chapter 8: Configuring the Application Development Environment .... 169

Pick the Right Tool for the Job................................................................. 169
Know Your Application ........................................................................ 169
  Hardware Constraints ........................................................................ 170
What to Use for Development ............................................................ 172
  C........................................................................................................ 172
  C++................................................................................................. 173
  Java................................................................................................. 173
Non-Traditional Embedded Languages................................................. 175
  Python.......................................................................................... 175
  TCL ............................................................................................... 177
  Shell Scripting............................................................................. 178
  PHP............................................................................................... 179
Performance and Profiling Tools.......................................................... 180
  Profiling....................................................................................... 180
  Leak Detection........................................................................... 184
  Static Analysis............................................................................ 187
IDE....................................................................................................... 188
  Your Editor + Make + Shell......................................................... 188
  Eclipse......................................................................................... 189
What’s Next....................................................................................... 196

Chapter 9: Application Development .................................................. 197
Coding for Portability......................................................................... 198
System Differences............................................................................ 199
  FIFO............................................................................................. 199
Getting the Tools.............................................................................. 201
Making Make Work........................................................................... 201
Running the Code on the Target....................................................... 205
More Complex Projects..................................................................... 206
## Chapter 10: Debugging Applications ................................. 215
- Getting Started on Your Application ................................................................. 215
- Types of Debugging ................................................................................................ 215
- Remote Debugging Overview .............................................................................. 216
- Debugging C and C++ .......................................................................................... 217
  - Building GDB ........................................................................................................ 217
  - GDB Front Ends ..................................................................................................... 218
  - Compiling for Debugging ..................................................................................... 219
- Debugging Java ....................................................................................................... 229
- Instrumentation ....................................................................................................... 233
- Java Instrumentation .............................................................................................. 235
- Instrumentation in Scripting Languages ............................................................... 236
- What’s Next ............................................................................................................. 236

## Chapter 11: Kernel Configuration and Development ........... 237
- Kernel Project Layout ............................................................................................. 237
  - Downloading the Kernel ..................................................................................... 239
- Building the Kernel ................................................................................................. 241
  - How Kernel Configuration Works ....................................................................... 244
  - Default Configurations ........................................................................................ 245
  - Editing .config By Hand ..................................................................................... 247
  - Building the Kernel ............................................................................................. 247
  - Building Modules .................................................................................................. 250
  - Cleaning Up .......................................................................................................... 251
- Open Source Community ......................................................................................... 252
  - The Kernel Development Process ....................................................................... 252
  - Contributing to the Linux Kernel ....................................................................... 252
  - Applying Patches .................................................................................................. 254
  - What’s Next ........................................................................................................... 255
Chapter 17: Deploying Applications ................................................................. 363

Deployment for Embedded Devices ........................................................................ 363
  Requirements .................................................................................................................. 364
  Industrial Design ............................................................................................................ 365
  Mechanical Design ....................................................................................................... 365
  Electrical Engineering ....................................................................................................... 366
  Manufacturing Engineering .............................................................................................. 367
  Software Design ................................................................................................................ 367
  Software Engineering ....................................................................................................... 368
  Manufacturing .................................................................................................................. 369

Deployment Strategies and Tactics ........................................................................ 371

Boot Loaders ................................................................................................................. 372
  In General ......................................................................................................................... 372
  UBOOT: Configuring Initial Parameters ............................................................................. 373
  Expect ................................................................................................................................. 374
  Boot Loaders Are Just Programs ...................................................................................... 377

Deployment Root File Systems .................................................................................... 378
  Application Files and Libraries ........................................................................................ 379
  First Field Update at the Factory .................................................................................... 381

What's Next ...................................................................................................................... 381

Chapter 18: Handling Field Updates ........................................................................ 383

Root File System Updates ............................................................................................ 383
  Basic Strategies ............................................................................................................... 384
  Forklift Upgrade ............................................................................................................... 384
  Parallel Systems ............................................................................................................... 388
  Do It Yourself .................................................................................................................... 389
  Using Package Managers ................................................................................................. 390
  Initramfs Root File Systems .......................................................................................... 401

Kernel Updates .............................................................................................................. 401
  Basic Strategies ................................................................................................................. 402
Modules ........................................................................................................................ .....................403
Forklift ................................................................................................................................................405
Field Update Failures ................................................................................................................ 406
  Report Failure, Stop ..................................................................................................................406
  Failsafe Root File System .....................................................................................................406
  Failsafe Kernel .......................................................................................................................406
In Summary ......................................................................................................................................407
Index ................................................................................................................................................409
About the Author

Gene Sally got mixed-up with computers at a young age, his fascination sparked by an Apple II, with the Lemon cooling system add-on, no less. As a software professional, Gene got his first job writing make files and then moved on to more exciting (for certain values of exciting) things like accounting, insurance processing, and social services systems. He first used Linux to set up a shared Internet connection and later used it when working on software that tested telecommunication management software; Gene was happy that a decent Unix-like environment could be had for free, and Linux became his axe of choice. Gene next found himself at a start-up that specialized in tools and distributions for embedded Linux, working in a variety of roles including engineer, trainer, technical support phone-answerer-guy, podcaster, and marketer. Presently, Gene is working at a company that creates safety products for retirement and assisted-living homes.

Gene resides outside of Pittsburgh, Pennsylvania in the rolling hills north of the city with his wife and three sons. When not working, writing, playing with kids, or tending to the never-ending repair list at home, Gene cycles, skis, and hikes throughout western Pennsylvania and eastern Ohio.
William von Hagen (Bill) has been a UNIX system administrator for over 20 years and a Linux fanatic since the early 1990s. He has worked as a systems programmer, system administrator, writer, application developer, drummer, and documentation manager. Bill has written or co-written books on such topics as Ubuntu Linux, GCC, Linux Server Hacks, Linux Filesystems, SUSE Linux, Red Hat Linux, SGML, and Mac OS X.
First, I thank the good Lord for providing me with time, prosperity, curiosity, vigor, intellect, and health; I hope this work puts these gifts to proper use. During the time I spent writing this book, I received tremendous support from my family. I couldn’t have done it without my wife and kids: you helped more than you know.

Writing a book is a team effort, and I am fortunate that the team working on this book was as excellent as any author could have ever expected. Thank you Bill von Hagen for being my technical reviewer, and Michelle Lowman and Frank Pohlmann for being my editors. All of the editors actually read what was written and provided feedback that made this book much better than it would have been without their efforts. I would also like to thank the production team at Apress, who created the cover, laid out the pages, and amazingly turned a collection of documents into the book you’re holding now. Special thanks goes to James Markham, who somehow managed to tolerate my inability to meet a deadline with more patience than I deserved. Last, but not least, the open source community has provided me with an excellent education about Linux, both in mail and newsgroup traffic and in the source code itself.
Introduction

When I got started in embedded Linux nearly a decade ago, the question was, “Should I even use an operating system?” Going with Linux frequently meant porting the operating system to run on your target hardware and building to the tools to do so. Much has changed over the years, to the point that Linux is selected by default for many projects, and the decisions revolve around what features of the operating system can be used on the project. The question today is, “How should I configure my Linux distribution?” In technology terms, this is a seismic shift in developer attitudes in a very short time frame.

Linux is so pervasive in the embedded space that embedded processors and boards ship with Linux by default. Buyers simply expect that the board will boot Linux and they’ll have the tools they need for embedded development provided along with the hardware. Unlike in the early days of Linux, as a developer, you won’t be porting Linux to your board but rather configuring an already-running Linux kernel and root file system so that they suit your application.

With this background in mind, I wrote this book from the perspective of a user who is starting their project with a Linux distribution already running on the board. It may not be the distribution that eventually ships with the product, but it will likely serve as a starting point. Tasks like building the cross-compiler from scratch are documented in the book so you understand the process, but you will probably use a cross-compiler that has been provided with the board so you can concentrate on the application. However, learning how to build and configure the tools for a Linux system isn’t a wasted effort, because when you need to squeeze every bit of memory out of your system, this is an essential skill.

Furthermore, with new System on a Chip (SOC) designs, the Linux distribution that comes on the board has all the drivers necessary for the devices on the chip. Only in the rarest events is driver development necessary. This means that most engineers spend time customizing the kernel rather than building new kernel components, and the total time spent doing any sort of kernel configuration or development is a fraction of what it was in years past.

As the processors in embedded devices become more powerful, developers are finding they can use languages other than C for development. It’s common for C++ to be used for development as well as other higher-level language like Python, TCL, Java, and even Perl or PHP. To somebody who started doing embedded in assembly, using a language like Perl on an embedded target is nearly heresy, if not outright apostasy; however, these higher-level languages greatly increase the productivity of embedded development. In an industry where time-to-market is paramount, higher-level languages will become more mainstream.

Embedded projects have a development process unlike other software projects. First there is the design process, which usually involves creating a special enclosure and a user interface that’s probably a small LCD display and some buttons; but more important are the deployment and update of the software. The code for the project isn’t put on a CD or a web site but must be downloaded on the board along with the Linux distribution. After the initial installation of the software, you’ll likely want to update the software with new versions that contain bug fixes and other goodies. Depending on how your system is configured, updating your system may be a project in itself.
If you’re starting an embedded project for your work or have purchased one of the many, now very inexpensive board targeted to hobbyists, I wish you the best of luck in your endeavors. Using Linux in an embedded project is easier, and more fun, than ever.