Undergraduate Topics in Computer Science

Undergraduate Topics in Computer Science (UTiCS) delivers high-quality instructional content for undergraduates studying in all areas of computing and information science. From core foundational and theoretical material to final-year topics and applications, UTiCS books take a fresh, concise, and modern approach and are ideal for self-study or for a one- or two-semester course. The texts are all authored by established experts in their fields, reviewed by an international advisory board, and contain numerous examples and problems. Many include fully worked solutions.

More information about this series at http://www.springer.com/series/7592

Alan Holt · Chi-Yu Huang

Embedded Operating Systems

A Practical Approach

Springer

Alan Holt
IP Performance
Bristol
UK

Chi-Yu Huang
Tata Technologies Ltd.
Bristol
UK

ISSN 1863-7310 ISSN 2197-1781 (electronic)
ISBN 978-1-4471-6602-3 ISBN 978-1-4471-6603-0 (eBook)
DOI 10.1007/978-1-4471-6603-0

Library of Congress Control Number: 2014948743

Springer London Heidelberg New York Dordrecht

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

In memory of Marjorie Rose Holt
and Siou-yìn Zheng Huang

Preface

Most people are aware of mainstream operating systems such as Microsoft's Windows or Mac OS X. This is because these are the operating systems that they directly interact with on their personal computers. However, personal computers rarely operate in isolation nowadays.

Many computing applications require *online* access. Access to the Internet relies on many computing systems, for example, Ethernet switches, packet routers, Wireless access-points and even DNS servers. Unlike personal computers, network infrastructure equipment perform dedicated tasks and are often described as embedded systems.

The term embedded system covers a range of computing systems. It derives from computing systems that are *embedded* within some larger device, for example, within a car's engine management system. The term has clearly broadened to cover standalone systems too. Nor is it limited to network infrastructure systems or consumer goods such as satellite navigation devices.

The term embedded system is applied to devices which are small and have limited resources (relative to a personal computer). Mobile devices such as smart phones and tablets are, therefore, described as embedded devices, even though their function closely resembles a personal computer. Nevertheless, an embedded system needs an operating system specific to its needs, for example:

- Small form factor.
- Low processing power and small memory.
- Reduced power consumption for battery longevity.
- Support for real-time applications.

There are many embedded operating systems available, however, in this book we confine our discussion to the GNU/Linux operating system. GNU/Linux is not exclusively an embedded operating system and is used on personal computers and servers as Windows and OS X. However, due to its open source nature it is highly customisable and can be adapted to many environments. For this reason, GNU/Linux systems have gained considerable popularity in the embedded system market.

This is not to say that GNU/Linux is a panacea for embedded systems and there are many excellent alternatives. Nevertheless, there are some good reasons for choosing GNU/Linux:

- GNU/Linux is open source and is freely available.
- Due to its open source nature, GNU/Linux is highly customisable so we can build bespoke systems specific to our needs.
- It is widely used for embedded systems.
- GNU/Linux, like its Unix predecessor, was used extensively in education to teach operating systems.
- While the subject of the book is embedded operating systems, our choice of GNU/Linux means it could be used as a text for a more general course on operating systems.

Operating systems is a diverse subject area and there many books on the subject and many on GNU/Linux alone, covering topics such as the kernel, administration, networking, wireless, high-performance computing and systems programming (to name a few). There are even books on GNU/Linux embedded systems.

However, we felt there was a gap in the literature which described the component parts of an operating system and how they worked together. We address these issues in this book by adopting a practical approach. The procedure for building each component of the operating system, namely the bootloader, kernel, filesystem, shared libraries, start-up scripts, configuration files and system utilities, from its source code, is described in detail. By the end of this book the reader will be able to build a fully functional GNU/Linux embedded operating system.

We take this opportunity to pre-empt any Amazon reviews stating this book is not for beginners. Indeed, this is not an introductory text on operating systems, rather it is aimed at undergraduate/graduate level students and industry professionals.

Acknowledgments

The authors would like to thank the following people for the valuable contribution they made to this book: Paul While and Edwin Chen.

Contents

Abbreviations

API	Application programming interface
Bash	Bourne again shell—a command-line interpreter
Busybox	A collection of Unix utilities designed for embedded systems
Debian	A GNU/Linux distribution
GCC	GNU compiler collection
Glibc	GNU C Library
GNU	GNU is not Unix
Grub	Grand unified bootloader
Ncurses	A library of screen handling functions
NSS	Name switch service
TCP/IP	Transmission control protocol/internet protocol
Tmpfs	Temporary file system
Ubuntu	A GNU/Linux distribution based upon Debian
UML	User mode Linux
Vi	A text editor
VM	Virtual machine