UNIX System Programming Using C++

Terrence Chan

An imprint of Pearson Education

ISBN 978-81-317-2280-0

First Impression, 2008

This edition is manufactured in India and is authorized for sale only in India, Bangladesh, Bhutan, Pakistan, Nepal, Sri Lanka and the Maldives. Circulation of this edition outside of these territories is UNAUTHORIZED.

Published by Dorling Kindersley (India) Pvt. Ltd., licensees of Pearson Education in South Asia.

Head Office: 482, F.I.E., Patparganj, Delhi 110 092, India.
Registered Office: 14 Local Shopping Centre, Panchsheel Park, New Delhi 110 017, India.

Printed in India by Sheel Print-N-Pack

Table of Contents

Preface

The content of this book is derived from my several years of teaching Advanced UNIX Programming with C and C++ at two University of California Extensions (Berkeley and Santa Cruz). The objectives of the courses were to teach students advanced programming techniques using UNIX system calls and the ANSI C and C++ programming languages. Specifically, students who took the courses learned the following:

- Advanced ANSI C and C++ programming techniques, such as how to use function pointers and create functions that accept variable numbers of arguments
- The ANSI C library functions and C++ standard classes, and how to use them to reduce development time and to maximize portability of their applications
- Familiarity with the UNIX kernel structure and the system calls. These allow users to write sophisticated applications to manipulate system resources (e.g., files, processes, and system information), and to design new operating systems
- How to create network-based, multitasking, client/server applications which run on heterogenous UNIX platforms

The objective of this book is to convey to readers the techniques and concepts stated above. Furthermore, this book provides more detailed explanations and comprehensive examples on each topic than can be done in a course. Thus, readers can gain a better understanding of the subject matter and can learn at their own pace. This book also describes the latest advanced UNIX programming techniques on remote procedure calls and multithreaded programs. These techniques are important for the development of advanced distributed client/server applications in a symmetrical multiprocessing and network-based computing environment.

All the aforementioned information will be described in the C++ language. This is because in the last few years more and more advanced software developers are using C++ in applications development. This is due to the fact that the C++ language provides much stronger type-checking and includes object-oriented programming constructs than other procedural programming languages. These features are very useful in facilitating large-scale, complex UNIX system applications development and management.

This book covers the C++ programming language based on the draft version of the ANSI/ISO C++ standard [1, 2, 3]. Most of the latest C++ compilers provided by various computer vendors (e.g., Sun Microsystems Inc., Microsoft Corporation, Free Software Foundation, etc.) are compliant with this standard.

In addition to the C++ language, some significant C library functions, as defined by the ANSI C standard [4], are also described in this book. These functions are not covered by the C++ standard classes or by the UNIX application program interface. Thus, it is important that users be familiar with these to increase their knowledge base and choices of library functions.

The UNIX operating systems covered in this book include: UNIX System V.3, UNIX System V.4, BSD UNIX 4.3 and 4.4, Sun OS 4.1.3, and Solaris 2.4. The last two operating systems belong to SUN Microsystems, where Sun OS 4.1.3 is based on BSD 4.3 with UNIX System V.3 extensions, and Solaris 2.4 is based on the UNIX System V.4.

Although the primary focus of this book is on UNIX system programming, the IEEE (Institute of Electrical and Electronics Engineering) POSIX.1, POSIX.1b, and POSIX.1c standards are also covered in detail. This is to aid system programmers to develop applications that can be readily ported to different UNIX systems, as well as to POSIX-compliant systems (e.g., VMS and Windows-NT). This is important as most advanced commercial software products must run on heterogenous platforms by various computer vendors. Thus, the POSIX and ANSI standards can help users create highly platform-independent applications.

Target Audience

The book is targeted to benefit experienced software engineers and managers who are working on advanced system applications development in a UNIX environment. The products they develop may include advanced network-based client/server applications, distributed database systems, operating systems, compilers, or computer-aided design tools.

The readers should be familiar with the C++ language based on the AT&T version 3.0 (or the latest) and should have developed some C++ application programs on their own in the past. Moreover, the readers should be familiar with at least one version of UNIX system (e.g.,

UNIX System V). Specifically, the readers should know the UNIX file system architecture, user accounts assignment and management, file access control, and jobs control methods. Readers who need to brush up on UNIX system knowledge may consult any text book covering an introduction to the UNIX system.

Book Content

Although this book covers the ANSI C++ and C library functions and UNIX APIs extensively, the primary focus in describing these functions is to convey the following information to readers:

- Purposes of these functions
- Conformance of these functions to standard(s)
- How to use these functions
- Examples of their uses
- Where appropriate, how these functions are implemented in a UNIX system
- Any special considerations (e.g., conflict between the UNIX and POSIX standards) in using these functions

It is not the intention of the author to make this book a UNIX system programmer's reference manual. Thus, the function prototypes and header files required to use the ANSI library and UNIX API functions are described, but the detailed error codes that may be returned by these functions and the archive or shared libraries needed by users' programs will not be depicted. This type of information may be obtained via either the man pages of the functions or the programmer's reference manuals from the users' computer vendors.

The general organization of this book is:

- Chapter 1 describes the history of the C++ programming language and various UNIX systems. It also describes the ANSI/ISO C, ANSI/ISO C++, IEEE POSIX.1, POSIX.1b, and POSIX.1c standards
- Chapters 2 and 3 review the draft ANSI/ISO C++ programming language and object-oriented programming techniques. The C++ I/O stream classes, template functions, and exception handlings are also depicted in detail
- Chapter 4 describes the ANSI C library functions
- Chapter 5 gives an overview of the UNIX and POSIX APIs. Special header files and compile time options, as required by various standards, are depicted.
- Chapters 6 and 7 describe UNIX and POSIX.1 file APIs. These depict APIs that can be used to control various types of files in a system. They also describe file-locking techniques used to synchronize files in a multiprocessing environment

- Chapter 8 describes UNIX and POSIX.1 process creation and control methods. After reading this chapter, readers can write their own multiprocessing applications, such as a UNIX shell
- Chapter 9 describes UNIX and POSIX.1 signal handling methods
- Chapter 10 describes UNIX and POSIX.1b interprocess communication methods. These techniques are important in creating distributed client/server applications.
- Chapter 11 describes advanced network programming techniques using UNIX sockets and TLI
- Chapter 12 describes remote procedure call. This is important for development of network transport protocol-independent client/server application development on heterogeneous UNIX platforms
- Chapter 13 describes multithreaded programming techniques. These techniques allow applications to make efficient use of multiprocessor resources available on any machines on which they run

Note that although this book is based on C++, the focus on this book is not object-oriented programming techniques. This is because some readers are expected to be new to UNIX system programming and/or C++ language, thus it may be difficult for these readers to learn both object-oriented and system programming techniques at the same time. However, this book includes many useful C++ classes for interprocess communication, sockets, TLI, remote procedure call, and multithreaded programming. These classes encapsulate the low-level programming interface to these advanced system functions, and can be easily extended and incorporated into user applications to reduce their development efforts, time, and costs.

Example Programs

Throughout the book extensive example programs are shown to illustrate uses of the C++ classes, library functions, and system APIs. All the examples have been compiled by a Sun Microsystems C++ (version 4.0) compiler and tested on a Sun SPARC-20 workstation running Solaris 2.4. These examples are also compiled and tested using the Free Software Foundation GNU g++ compiler (version 2.6.3) on a Sun SPARC-20 workstation. Since the GNU g++ compilers can be ported to various hardware platforms, the examples presented in this book should run on different platforms (e.g., Hewlett Packadd's HP-UX and International Business Machines's AIX) also.

Readers are encouraged to try out the example programs on their own systems to get more in-depth familiarity of this subject matter. Users may download an electronic copy of the example programs via anonymous ftp to *ftp.prenhall.com*. The directory that stores the example tar file is */pub/ptr/professional_computer_science.w-022/chan/unixsys*. There are README files in the tar file that describe the programs and their cross references to chapters in the book. Finally, readers are welcome to send Emails to the author at *twc@tjsystems.com*.

Acknowledgments

I would like to thank Peter Collinson, Jeff Gitlin, Chi Khuong, Frank Mitchell, and my wife Jessica Chan for their careful reviewing of the book manuscript. Much of their valuable input has been incorporated in the final version of this book. Furthermore, I would like to extend my appreciation to Greg Doench, Nick Radhuber, and Brat Bartow for their valuable assistance in helping me through the preparation and publication process of this book.

Finally, I am grateful to my former students at the University of California Santa Cruz Extension who took my Advanced UNIX System Calls course and gave me valuable feedback in the refinement of the course material, much of which is used throughout this book.

References

[1]. Margaret A. Ellis and Bjarne Stroustrup, *The Annotated C++ Reference Manual,* Addison-Wesley, 1990.

[2]. Andrew Koenig, *Working Paper for Draft Proposed International Standard for Information Systems -- Programming Language C++ (Committees: WG21/N0414, X3J16/94-0025)*, January 1994.

[3]. Bjarne Stroustrup, *Standardizing C++. The C++ Report. Vol. 1. No. 1*, 1989.

[4]. American National Standard Institute, *American National Standard for Information Systems - Programming Language C, X3.159 - 1989*, 1989.