

<<UNIX初级教程>>

图书基本信息

书名：<<UNIX初级教程>>

13位ISBN编号：9787121068027

10位ISBN编号：7121068028

出版时间：2008-6

出版时间：电子工业出版社

作者：埃弗扎

页数：553

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<UNIX初级教程>>

前言

21世纪初的5至10年是我国国民经济和社会发展的关键时期，也是信息产业快速发展的关键时期。在我国加入WTO后的今天，培养一支适应国际化竞争的一流IT人才队伍是我国高等教育的重要任务之一。

信息科学和技术方面人才的优劣与多寡，是我国面对国际竞争时成败的关键因素。

当前，正值我国高等教育特别是信息科学领域的教育调整、变革的重大时期，为使我国教育体制与国际化接轨，有条件的高等院校正在为某些信息学科和技术课程使用国外优秀教材和优秀原版教材，以使我国在计算机教学上尽快赶上国际先进水平。

电子工业出版社秉承多年来引进国外优秀图书的经验，翻译出版了“国外计算机科学教材系列”丛书，这套教材覆盖学科范围广、领域宽、层次多，既有本科专业课程教材，也有研究生课程教材，以适应不同院系、不同专业、不同层次的师生对教材的需求，广大师生可自由选择 and 自由组合使用。

这些教材涉及的学科方向包括网络与通信、操作系统、计算机组织与结构、算法与数据结构、数据库与信息处理、编程语言、图形图像与多媒体、软件工程等。

同时，我们也适当引进了一些优秀英文原版教材，本着翻译版本和英文原版并重的原则，对重点图书既提供英文原版又提供相应的翻译版本。

<<UNIX初级教程>>

内容概要

UNIX是一类功能强大的主流操作系统。

本书从初学者的角度介绍了UNIX的系统概念及其命令的使用，阐述的内容都是针对初学者完成日常工作所必需的各个方面，涉及UNIX系统的常用命令、UNIX文件系统、vi编辑器和Emacs编辑器、UNIX通信工具、shell命令和程序开发，以及一些更为深入的UNIX命令。

书中还介绍了Linux操作系统以及Bourne Again Shell命令等。

本书帮助读者由浅入深、循序渐进地学习UNIX，形成清晰的概念，并且避免了直接罗列复杂的命令格式。

本书可作为UNIX课程的教学用书或参考书，也可供使用UNIX的科技工作进阅读和参考

<<UNIX初级教程>>

作者简介

作者：（美国）埃弗扎（Amir Afzal）

<<UNIX初级教程>>

书籍目录

1 First Things First 1.1 INTRODUCTION 1.2 COMPUTERS: AN OVERVIEW 1.3 COMPUTER HARDWARE 1.3.1 Input Devices 1.3.2 Processor Unit 1.3.3 Internal Memory 1.3.4 External Storage 1.3.5 Output Devices 1.4 PROCESS OPERATION 1.4.1 Performance Measurement 1.5 WHAT IS SOFTWARE?. 1.5.1 System Software 1.5.2 Application Software REVIEW EXERCISES2 The UNIX Operating System 2.1 UNIX OPERATING SYSTEM: A BRIEF HISTORY 2.1.1 UNIX System V 2.1.2 Berkeley UNIX 2.1.3 UNIX Standards 2.2 OTHER UNIX SYSTEMS 2.2.1 Linux 2.2.2 Solaris 2.2.3 Unix Ware 2.2.4 Which UNIX to Learn? 2.2.5 X Window System 2.3 OVERVIEW OF THE UNIX OPERATING SYSTEM 2.4 UNIX FEATURES 2.4.1 Portability 2.4.2 Multiuser Capability 2.4.3 Multitasking Capability 2.4.4 Hierarchical File System 2.4.5 Device-Independent Input and Output Operations 2.4.6 User Interface: Shell 2.4.7 Utilities 2.4.8 System Services REVIEW EXERCISES3 Getting Started 3.1 ESTABLISHING CONTACT WITH UNIX 3.1.1 Logging In 3.1.2 Changing Your Password: The passwd Command 3.1.3 General Rules for Choosing Passwords 3.1.4 Logging Off 3.2 USING SOME SIMPLE COMMANDS 3.2.1 The Command Line 3.2.2 Basic Command Line Structure 3.2.3 Date and Time Display: The date Command 3.2.4 Information on Users: The who Command 3.2.5 Displaying a Calendar: The Pal Command 3.3 GETTING HELP 3.3.1 Using the learn Command 3.3.2 Using the help Command 3.3.3 Getting More Information: The UNIX Manual 3.3.4 Using the Electronic Manual: The man Command 3.4 CORRECTING TYPING MISTAKES 3.5 USING SHELLS AND UTILITIES 3.5.1 Kinds of Shells 3.5.2 Changing Your Shell 3.5.3 The Shells in This Book 3.6 MORE ABOUT THE LOGGING-IN PROCESS COMMAND SUMMARY REVIEW EXERCISES Terminal Session4 The vi Editor: First Look 4.1 WHAT IS AN EDITOR? 4.1.1 UNIX-Supported Editors5 Introduction to the UNIX File System6 The vi Editor:Last Look7 The Emacs Editor8 The UNIX File System Continued 9 Exploring the Shell10 UNIX Communication11 Program Development13 Shell Programming13 Shell Scripts:Writing Applications14 FAREWELL TO UNIXAppendixIndex

章节摘录

插图：The shell is the parent' process, and when created, the ls program becomes the child process. The parent process (shell) calls fork. The fork routine duplicates the parent (shell) process, and if the creation of the child process is successful, assigns the child process a PID and adds it to the system process table. Next, the parent receives the child PID, the child receives code zero, and control is returned to the parent. The shell calls the wait routine and goes to the wait state (goes to sleep). Meanwhile, the child gets control and calls exec to overlay the child process area with the new program—in this case ls, the command you typed. Now ls carries out the command. It lists your current directory filenames, and when it is finished processing, it calls exit. Thus the child dies. The death of the child generates an event signal. The parent process (shell) is waiting for this event. It is awakened and gets control. The shell program continues, starting execution from the same address it was at before going to sleep (recall that this address was stored in the process table system data area as return address), and the prompt is displayed. What happens if the child is a background process? In that case, the parent (shell) does not call the wait routine; it continues in the foreground, and you see the prompt right away. What creates the first parent and child processes? When UNIX is booted, the init process is activated. Next, init creates one system process for each terminal. Thus, init is the original ancestor to all the processes in the system. For example, if your system supports 64 concurrent terminals, then init creates 64 processes. When you log in to one of these processes, the login process executes the shell. Later, when you log out (when the shell dies), init creates a new login process.

<<UNIX初级教程>>

编辑推荐

《UNIX初级教程(第5版)(英文版)》：国外计算机科学教材系列。

<<UNIX初级教程>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>