

The Linux Internals

(MSIT-)



Contributing Authors:

Sri. M.V. Panduranga Rao

Lecturer

Dept of Information Science & engg.

JNNCE

Shimoga.

Smt. K. Roopa

Lecturer

Dept of Computer Science & engg.

JNNCE

Shimoga.

This book is for Linux enthusiasts who want to know how the Linux kernel works. It is not an internals manual. Rather it describes the principles and mechanisms that Linux uses; how and why the Linux kernel works the way that it does.

Table of Contents

UNIT 1. Linux Overview

- 1.0 Introduction
- 1.1 Objectives
- 1.2 Welcome to LINUX
- 1.3 What is an operating system
 - 1.3.1 Memory management
 - 1.3.2 Processes
 - 1.3.3 Device drivers
 - 1.3.4 File system
- 1.4 Kernel
 - 1.4.1 Kernel Data structures
 - 1.4.1.1 Linked Lists
 - 1.4.1.2 Hash Tables
 - 1.4.1.3 Abstract interfaces
 - 1.4.2 Arrangement of Kernel Sources
- 1.5 Shell
 - 1.5.1 Shell features and benefits
- 1.6 Linux system directories
- 1.7 Login, Logoff, Issue commands & switching to another account
- 1.8 Pros & cons Of Linux
- 1.9 Summary
- 1.10 Check Your Progress

UNIT 2. LINUX Comparisons & vi editor

- 2.0 Introduction
- 2.1 Objectives
- 2.2 DOS-to-LINUX Comparisons
 - 2.2.1 significant differences
 - 2.2.2 DOS commands
 - Attrib,Backup,Msbackup,Restore,Chdir,Cd,Chkdsk,Copy,Xcopy,
Date,Time,Del,Erase,Dir,Doskey>Edit,Edlin,Format,Help,Mem,
Mkdir,Md,More,Path,Prompt,Ren,Rename
- 2.3 VI Editor quick references
 - 2.3.1 vi editor usage
 - 2.3.2 Simple Character Motion in vi
 - 2.3.3 Inserting Text into the File with i, a, o, and O
 - 2.3.4 Quick Reference for vi editor commands
 - 2.3.5 Quick Reference for ex mode commands
 - 2.3.6 The most common erase characters
- 2.4 Summary
- 2.5 Check Your Progress

UNIT 3. Linux Internal Commands & Builtins

- 3.0 Introduction
- 3.1 Objectives
- 3.2 Basic commands
 - ls, cat, tac, rev, cp, mv, rm, rmdir, mkdir, cd, pwd, chmod, ln
- 3.3 Complex commands
 - find, expr, date, zdump
- 3.4 Time / Date commands
 - time, touch, cal, sleep, times
- 3.5 Text Processing commands
 - sort, tsort, diff, patch, cmp, comm, uniq, sort, join, head, tail, grep, sed, awk, wc, tr, pr, lex, yacc
- 3.6 File and Archiving commands
 - tar, locate, slocate, split, make
- 3.7 Communication commands
 - host, traceroute, ping, finger, ftp, uucp, telnet, rlogin, rsh, rcp, ssh, write, banner,
- 3.8 System and Administrative commands
 - chown, chgrp, useradd, userdel, who, su, mesg, wall, uname, du, df, hostname, halt, shutdown, reboot, finger, passwd
- 3.9 Networking commands
 - ifconfig, mount, umount, mkswap, swapon, swapoff, fdisk, fsck, e2fsck, debugfs, mkbootdisk, fdformat, umask, Aliases
- 3.10 Shell Related commands
 - echo, printf, read, let, exit, exec, true, false, help, jobs, disown, fg, bg, wait, suspend, logout,
- 3.11 Process commands
 - kill, ps,
- 3.12 Help commands
 - man, less, info, --help, apropos, whatis
- 3.13 Summary
- 3.14 Check Your Progress

UNIT 4. Processes

- 4.0 Introduction
- 4.1 Objectives
- 4.2 Linux Processes
 - 4.2.1 Creating a process
 - 4.2.2 Running a process in the Background
 - 4.2.3 Checking on a process
 - 4.2.3.1 ps
 - 4.2.4 Cancelling a process
 - 4.2.4.1 kill
 - 4.2.4.2 problem checklist
- 4.3 Identifiers or permissions
- 4.4 Scheduling
 - 4.4.1 Scheduling in Multiprocessor systems
- 4.5 Files
- 4.6 Virtual memory
- 4.7 Times & Timers
- 4.8 Executing programs
 - 4.8.1 ELF

- 4.8.2 Script files
- 4.9 Summary
- 4.10 Check your progress

UNIT 5. Shell Programming

- 5.0 Introduction
- 5.1 Objectives
- 5.2 Why Shell Programming?
- 5.3 When not to use shell scripts
- 5.4 Starting Off With a Sha-Bang
- 5.5 Redirecting I/O
 - 5.5.1 Standard Input and Standard Output
 - 5.5.1.1 putting text in a file
 - 5.5.1.2 understanding access permissions
 - 5.5.2 pipes and filters
- 5.6 Invoking the script
- 5.7 Special Characters
- 5.8 Introduction to Variables and Parameters
 - 5.8.1 Variable Assignment
 - 5.8.2 Positional Parameters
 - 5.8.3 Special meanings of escaped characters
 - 5.8.4 Examples of using pipelines to connect commands
- 5.9 If Then Else Constructs
 - 5.9.1 File test operators
 - 5.9.2 Comparison operators (binary)
 - 5.9.3 Internal Variables
- 5.10 Loops and Branches
 - 5.10.1 Loops
 - 5.10.1.1 for loop
 - 5.10.1.2 while loop
 - 5.10.1.3 until loop
- 5.11 Nested Loops
- 5.12 Branching Construct
- 5.13 Examples of Bourne shell scripts
 - 5.13.1 To read i/p to a command and process it in some way
 - 5.13.2 To read commands from the terminal and process them
- 5.14 Summary
- 5.15 Check Your Progress

UNIT 6. System Administration

- 6.0 Introduction
- 6.1 Objectives
- 6.2 Root account
- 6.3 Creating user accounts
- 6.4 Changing user passwords
- 6.5 Disabling user accounts
- 6.6 Removing user accounts
- 6.7 Linux password & shadow file formats
- 6.8 System shutdown & restart

- 6.9 Checking storage space
- 6.10 Managing processes
- 6.11 Starting & stopping processes
- 6.12 Summary
- 6.13 Check Your Progress

UNIT 7. The File system

- 7.0 Introduction
- 7.1 Objectives
- 7.2 LINUX File system
 - 7.2.1 Home Directory
 - 7.2.2 Working Directory
 - 7.2.3 The Directory Tree
 - 7.2.4 Absolute Pathnames
 - 7.2.5 Relative Pathnames
 - 7.2.6 Changing Your Working Directory
 - 7.2.6.1 pwd
 - 7.2.6.2 cd
 - 7.2.7 Files in the Directory Tree
 - 7.2.8 Listing Files
 - 7.2.8.1 ls
- 7.3 Looking Files
 - 7.3.1 cat
 - 7.3.2 more
 - 7.3.3 pg
- 7.4 Protecting and Sharing Files
 - 7.4.1 Directory Access Permissions
 - 7.4.2 File Access Permissions
 - 7.4.3 More Protection Under Linux
- 7.5 File Management
 - 7.5.1 Methods of Creating Files
 - 7.5.2 File and Directory Names
 - 7.5.3 File and Directory Wildcards
 - 7.5.4 Managing Your Files
 - 7.5.4.1 Creating Directories
 - 7.5.4.1.1 mkdir
 - 7.5.4.2 Copying Files
 - 7.5.4.2.1 cp
 - 7.5.4.2.2 ftp
 - 7.5.4.3 Finding Files
 - 7.5.4.4 Files on Other Operating Systems
- 7.6 The Second Extended File system (EXT2)
- 7.7 The Virtual File System (VFS)
 - 7.7.1 Mounting a File System
 - 7.7.1.1 /dev
 - 7.7.1.2 /proc
- 7.8 Summary
- 7.9 Check Your Progress

UNIT 8. Interprocess Communication Mechanisms

- 8.0 Introduction
- 8.1 Objectives
 - 8.1.1 System V, IPC Mechanisms
 - 8.1.2 IPC Identifiers
 - 8.1.3 IPC Keys
 - 8.1.4 The `ipcs` Command
 - 8.1.5 The `ipcrm` Command
- 8.2 Message Queues
 - 8.2.1 Basic Concepts
 - 8.2.2 SYSTEM CALL: `msgget()`
 - 8.2.3 SYSTEM CALL: `msgsnd()`
 - 8.2.4 SYSTEM CALL: `msgctl()`
- 8.3 Semaphores
 - 8.3.1 Basic Concepts
 - 8.3.2 SYSTEM CALL: `semget()`
 - 8.3.3 SYSTEM CALL: `semop()`
 - 8.3.4 SYSTEM CALL: `semctl()`
- 8.4 Shared Memory
 - 8.4.1 Basic Concepts
 - 8.4.2 SYSTEM CALL: `shmget()`
 - 8.4.3 SYSTEM CALL: `shmat()`
 - 8.4.4 SYSTEM CALL: `shmctl()`
 - 8.4.5 SYSTEM CALL: `shmdt()`
- 8.5 Summary
- 8.6 Check Your Progress

UNIT 9. Networks

- 9.0 Introduction
- 9.1 Objectives
- 9.2 An Overview of TCP/IP Networking
- 9.3 The Linux TCP/IP Networking Layers
- 9.4 The BSD Socket Interface
- 9.5 The INET Socket Layer
 - 9.5.1 Creating a BSD Socket
 - 9.5.2 Binding an Address to an INET BSD Socket
 - 9.5.3 Making a Connection on an INET BSD Socket
 - 9.5.4 Listening on an INET BSD Socket
 - 9.5.5 Accepting Connection Requests
- 9.6 The IP Layer
 - 9.6.1 Socket Buffers
 - 9.6.2 Receiving IP Packets
 - 9.6.3 Sending IP Packets
 - 9.6.4 Data Fragmentation
- 9.7 The Address Resolution Protocol (ARP)
- 9.8 Summary
- 9.9 Check Your Progress

UNIT 10. Installation and Hardware Configuration

- 10.0 Introduction
- 10.1 Objectives
- 10.2 Creating an Installation Diskette
- 10.3 Booting Linux Installation Program
- 10.4 Partitioning Hard Drive(s)
 - 10.4.1 Using the Partitioning Program: Fdisk
 - 10.4.1 Setting up Swap Space
 - 10.4.2 Choosing Partitions to Format
- 10.5 Choosing Desired Packages to Install
- 10.6 Hardware Configuration
- 10.7 Booting with LILO
 - 10.7.1 Multi-boot with Other Operating Systems
 - 10.7.2 Dual Booting: Linux and Windows with LiLo
- 10.8 Downloading and Installing Red Hat Updates
 - 10.8.1 Configuration from A-Z with Linuxconf
 - 10.8.2 TAR's
 - 10.8.3 Using the Red Hat Package Manager (RPM)
 - 10.8.4 Installing or Upgrading Without RPM
 - 10.8.5 Red Hat Linux Sound Configuration
- 10.9 Summary
- 10.10 checkyour progress

UNIT 11. GLOSSARY

UNIT 12. APPENDIX

Preface

Linux is a phenomenon of the Internet. Born out of the hobby project of a student it has grown to become more popular than any other freely available operating system. To many Linux is an enigma. How can something that is free be worthwhile? In a world dominated by a handful of large software corporations, how can something that has been written by a bunch of ``hackers" (sic) hope to compete? How can software contributed to by many different people in many different countries around the world have a hope of being stable and effective? Yet stable and effective it is and compete it does. Many Universities and research establishments use it for their everyday computing needs. People are running it on their home PCs and I would wager that most companies are using it somewhere even if they do not always realize that they do. Linux is used to browse the web, host web sites, write theses, send electronic mail and, as always with computers, to play games. Linux is emphatically not a toy; it is a fully developed and professionally written operating system used by enthusiasts all over the world.

Linux was the solution to a simple need. The only software that Linus Torvalds, Linux's author and principle maintainer was able to afford was `Minix`. `Minix` is a simple, Unix™ like, operating system widely used as a teaching aid. Linus was less than impressed with its features, his solution was to write his own software. He took Unix

TM as his model as that was an operating system that he was familiar with in his day to day student life. He started with an Intel 386 based PC and started to write. Progress was rapid and, excited by this, Linus offered his efforts to other students via the emerging world wide computer networks, then mainly used by the academic community. Others saw the software and started contributing. Much of this new software was itself the solution to a problem that one of the contributors had. Before long, Linux had become an operating system. It is important to note that Linux contains no Unix TM code, it is a rewrite based on published POSIX standards. Linux is built with and uses a lot of the GNU (GNU's Not Unix TM) software produced by the Free Software Foundation in Cambridge, Massachusetts.

Most people use Linux as a simple tool, often just installing one of the many good CD ROM-based distributions. A lot of Linux users use it to write applications or to run applications written by others. Many Linux users read the HOWTOs¹ avidly and feel both the thrill of success when some part of the system has been correctly configured and the frustration of failure when it has not. A minority are bold enough to write device drivers and offer kernel patches to Linus Torvalds, the creator and maintainer of the Linux kernel. Linus accepts additions and modifications to the kernel sources from anyone, anywhere. This might sound like a recipe for anarchy but Linus exercises strict quality control and merges all new code into the kernel himself. At any one time though, there are only a handful of people contributing sources to the Linux kernel.

The majority of Linux users do not look at how the operating system works, how it fits together. This is a shame because looking at Linux is a very good way to learn more about how an operating system functions. Not only is it well written, all the sources are freely available for you to look at. This is because although the authors retain the copyrights to their software, they allow the sources to be freely redistributable under the Free Software Foundation's GNU Public License. At first glance though, the sources can be confusing; you will see directories called `kernel`, `mm` and `net` but what do they contain and how does that code work? What is needed is a broader understanding of the overall structure and aims of Linux. This, in short, is the aim of this book: to promote a clear understanding of how Linux, the operating system, works. To provide a mind model that allows you to picture what is happening within the system as you copy a file from one place to another or read electronic mail.

It must be noted that around 95% of the Linux kernel sources are common to all of the hardware platforms that it runs on.

The roots of Linux can be traced back to the origins of Unix TM. In 1969, Ken Thompson of the Research Group at Bell Laboratories began experimenting on a multi-user, multi-tasking operating system using an otherwise idle PDP-7. He was soon joined by Dennis Richie and the two of them, along with other members of the Research Group produced the early versions of Unix TM. Richie was strongly influenced by an earlier project, MULTICS and the name Unix TM is itself a pun on the name MULTICS. Early versions were written in assembly code, but the third version was rewritten in a new programming language, C. C was designed and written by Richie expressly as a programming language for writing operating systems. This rewrite allowed Unix TM to move onto the more powerful PDP-11/45 and 11/70 computers then being produced by DIGITAL. The rest, as they say, is history. Unix TM moved out of the laboratory and into mainstream computing and soon most major computer manufacturers were producing their own versions.

This book does not make any assumptions about the knowledge or experience of the reader. I believe that interest in the subject matter will encourage a process of self education where necessary. That said, a degree of familiarity with computers, preferably the PC will help the reader derive real benefit from the material, as will some knowledge of the C programming language.

Organisation of this Book

This book is *not* intended to be used as an internals manual for Linux. Instead it is an introduction to operating systems in general and to Linux in particular. The units each follow my rule of “working from the general to the particular”.

Each unit is fairly independent, like the Linux kernel subsystem that they each describe. Sometimes, though, there are linkages; for example you cannot describe a process without understanding how virtual memory works.

The Ist UNIT Linux Overview gives brief introduction regarding the operating system , Kernel , Shell.

The IInd UNIT Linux Comparisons & vi editor describes the basic differences and similarities between the Linux and the well-known Dos operating system also, gives brief introduction to vi editor.

The IIIrd UNIT Linux Internal Commands & Builtins deals with most useful commands because standard LINUX commands make shell scripts more versatile.

The IVth UNIT Processes describes what a process is and how the Linux kernel creates, manages and deletes the processes in the system.

The Vth UNIT Shell Programming describes how to write effective shell scripts for usage.

The VIth UNIT System Administration explains the management of users,disk space and other concepts relevant to system administration.

The VIIth UNIT The File system describes how the Linux kernel maintains the files in the file systems that it supports. It describes the Virtual File System (VFS) and explains how the Linux kernel's real file systems are supported.

The VIIIth UNIT Interprocess Communication Mechanisms explains the different Interprocess communication mechanisms present in Linux.

The IXth UNIT Networks describes the concept of networking with different protocols such as TCP, IP, UDP, ARP etc.

The Xth UNIT Installation and Hardware Configuration will detail the procedures needed to install Red Hat linux onto an Intel system; the procedures are similar whether you choose to install using either GUI- or text-based installation. Updation and installation of necessary package and mutiboot concepts are also discussed.

The XIth UNIT describes the glossary of terms.

The XIIth UNIT describes the necessary appendix for the material.

Acknowledgements

We must thank the many people who have been kind enough to take the time to e-mail us with comments about this book. We are more happy to receive comments, however please note our e-mail addresses.

Reference e-mails: raomvp@yahoo.com roopasindhe@lycos.com

URL / Web Site: <http://www.raomvp.bravepages.com>
