Tru64 UNIX

Documentation Overview

Part Number: AA-RH8RA-TE

July 1999

Product Version: Tru64 UNIX Version 5.0

This manual describes the documentation for the Compaq Tru64 UNIX (formerly DIGITAL UNIX) operating system. It also describes the structure of the documentation kits and provides information about ordering them.

Compaq Computer Corporation
Houston, Texas

Contents

Figures

Tables

About This Manual

This manual describes the documentation that comes with the Compaq Tru64™ UNIX® operating system. It also provides information about the structure of the documentation kits and gives you information to help you order printed versions of the documentation.

Included in this manual is a glossary of terms that are commonly used in the UNIX environment.

Audience

This manual is for anyone who needs to access documentation about the Tru64 UNIX operating system. It can help you decide which manuals in the documentation set are most useful to you and how you can use those manuals.

New and Changed Features

This manual has been revised to provide you with information about changes to the Tru64 UNIX documentation that have been made since the last release of the operating system.

For an overview of the major changes to the Version 5.0 documentation set, see Section 1.2.

Organization

This manual is organized as follows:

Chapter 1	Provides general information about the Tru64 UNIX documentation set.
Chapter 2	Describes the Tru64 UNIX documentation set, provides a description of each manual, and describes how the documentation is packaged.
Chapter 3	Describes documentation that supplements the Tru64 UNIX documentation set. Some of these documents are produced by Compaq and others are produced by other organizations and provided by Compaq to help meet specific needs when working with the Tru64 UNIX operating system.

Chapter 4	Describes the Tru64 UNIX reference pages, shows the various ways of displaying the reference pages on line, and describes the contents of the Reference Pages Documentation Kit.
Chapter 5	Describes the Device Driver Kit and the contents of the books in that kit.
Chapter 6	Provides information on how to order Tru64 UNIX printed documentation.
Glossary	Provides definitions for many of the terms you may see while using the Tru64 UNIX documentation. Although the majority of terms deal with the UNIX environment, you will also find other common terms that you will encounter; for example, words related to the Internet.

Reader's Comments

Compaq welcomes any comments and suggestions you have on this and other Tru64 UNIX manuals.

You can send your comments in the following ways:

- Fax: 603-884-0120 Attn: UBPG Publications, ZKO3-3/Y32
- Internet electronic mail: `readers_comment@zk3.dec.com`

  A Reader's Comment form is located on your system in the following location:

  `/usr/doc/readers_comment.txt`

- Mail:

  Compaq Computer Corporation
  UBPG Publications Manager
  ZKO3-3/Y32
  110 Spit Brook Road
  Nashua, NH 03062-9987

  A Reader's Comment form is located in the back of each printed manual. The form is postage paid if you mail it in the United States.

Please include the following information along with your comments:

- The full title of the book and the order number. (The order number is printed on the title page of this book and on its back cover.)
- The section numbers and page numbers of the information on which you are commenting.
- The version of Tru64 UNIX that you are using.
- If known, the type of processor that is running the Tru64 UNIX software.

The Tru64 UNIX Publications group cannot respond to system problems or technical support inquiries. Please address technical questions to your local system vendor or to the appropriate Compaq technical support office. Information provided with the software media explains how to send problem reports to Compaq.

1

Finding the Documentation You Need

You can find most of the Tru64 UNIX documentation you need on the Documentation CD–ROM that comes with the Tru64 UNIX product and also on the World Wide Web. Most of this documentation is provided on line in HTML and PDF formats and is also available in printed books.

Your Web browser gives you access to the HTML format, and the Adobe Acrobat Reader lets you view and print the PDF versions. The printed books are packaged in kits that you can purchase from Compaq.

A smaller subset of documentation is included on the Tru64 UNIX Operating System CD–ROM and on the Associated Products CD–ROMs.

This chapter can help you to find the documentation you need, in the format that is most convenient to you.

Note

If you already know the books you want, you can skip to Chapter 6 for information on how to place an order.

1.1 Defining the Documentation Set

The following list describes some of the terms used to categorize documentation that is provided with the operating system or used in conjunction with utilities and applications that run on the operating system:

- **Core documentation** consists of books that help you to use the Tru64 UNIX operating system, including the components provided when you install the system's optional and mandatory subsets. Most of this documentation was created by Compaq writers who worked closely with the developers of the operating system to document the system's various components. The books that make up the core documentation are described in Chapter 2.

  On line, the Tru64 UNIX **reference pages** are available as optional subsets of the operating system and in HTML form on the Documentation CD–ROM. They are also available in printed format in a 22–volume kit. For information about the reference pages, see Chapter 4.

Documentation to aid program developers in writing device drivers for Tru64 UNIX was produced by Compaq writers working closely with the device-driver engineers. This documentation is packaged in a separately orderable kit, which also includes driver development tools, device driver examples, and X Consortium code. For information about the device driver documentation, see Chapter 5.

- **Supplementary documentation** tends to be less formal in structure than the core documentation. This category includes many different types of documents, such as Software Product Descriptions (SPDs), release notes for some components of the Tru64 UNIX operating system, and industry reference material and white papers that come from Compaq engineers and other sources. Supplementary documentation exists only on line. It is not distributed in a hard copy format, so does not exist in any of the Tru64 UNIX documentation kits. For more information see Chapter 3.
- **Layered products documentation** consists of books and other documents that aid in the use of separately licensed products, such as DEC C++ and the TruCluster Software Products. For more information, see Section 1.6.

1.2 Changes to the Version 5.0 Documentation Set

With this release of the Tru64 UNIX operating system, changes have been made to all of the books in the core documentation set. In some books, the changes were minor improvements, while others have been expanded and substantially reorganized to document changes to the operating system and to improve the usability of the information.

The following sections describe some of the more noteworthy changes to the Version 5.0 documentation set:

1.2.1 Changes to the Installation Guide

Because the knowledge level of users who install the operating system ranges from novice to expert, the installation instructions have been redesigned to accommodate all users. As a result, the *Installation Guide* has been divided into two books:

- The reorganized *Installation Guide* has been reduced in scope in order to help the novice user progress from zero to operational in less time and with less confusion.
- The new *Installation Guide — Advanced Topics* provides expert power installers with the information they need.

Both of these books are packaged in the Startup Documentation kit.

1.2.2 New Documentation

The following books have been added to the core documentation set:

- *Writing Kernel Modules*

  See Section 2.2 for an overview of this book.

- *System Configuration Supplement: OEM Platforms*

  See Section 2.1.3 for an overview of this book.

The following documents have been added to the Supplementary bookshelf of the Documentation CD–ROM. See Chapter 3 for descriptions of these documents.

- *BIND Configuration File Guide* (replaces the *Name Server Operations Guide for BIND*)
- *Object File / Symbol Table Specification*

1.2.3 Retired Documentation

The documentation in the following list is no longer provided on the Documentation CD–ROM and is unavailable in printed form. However, the versions of this documentation that were provided in previous releases of the operating system are still available on the Web in the documentation sets of those releases.

- *ULTRIX to DIGITAL UNIX Migration Guide*

  This manual had been included in the System and Network Management Documentation Kit.

- *DECwindows User's Guide*

  This manual had been included in the General User Documentation Kit.

- The Display PostScript supplemental documentation from Adobe Systems Incorporated. These files had been available on the operating system CD–ROM.

- *DECwindows Companion to the OSF/Motif Style Guide*

  This book had been included on the Supplementary Documents bookshelf of the Documentation CD–ROM.

- *DECwindows Motif Guide to Application Programming*

  This book had been included on the Supplementary Documents bookshelf of the Documentation CD–ROM.

- *DECwindows Extensions to Motif*

  This book had been included on the Supplementary Documents bookshelf of the Documentation CD–ROM.

1.2.4 Changes to the Printed Documentation Set

The following sections describe major changes to the printed Documentation Kit in this release.

1.2.4.1 Books No Longer Available in Printed Form

The following books are no longer available in printed form; they are, however, still available in HTML and PDF formats on the Documentation CD–ROM and on the Web:

- *DECevent Translation and Reporting Utility*
- *Common Desktop Environment: Advanced User's and System Administrator's Guide*
- *Common Desktop Environment: Application Builder User's Guide*
- *Common Desktop Environment: Desktop KornShell User's Guide*
- *Common Desktop Environment: Product Glossary*
- *Common Desktop Environment: Help System Author's and Programmer's Guide*
- *Common Desktop Environment: Internationalization Programmer's Guide*
- *Common Desktop Environment: Programmer's Guide*
- *Common Desktop Environment: Programmer's Overview*
- *Common Desktop Environment: Style Guide and Certification Checklist*
- *Common Desktop Environment: ToolTalk Messaging Overview*
- *Common Desktop Environment: User's Guide*

1.2.4.2 Changes to the Kit Structure

With this release, the Developer's Documentation Kit is no longer separated into general programming and windows programming kits. This one kit now contains all of the printed core documentation you need to program software on and for the Tru64 UNIX operating system.

Additionally, the order numbers for the kits have changed from those you might be used to. See Chapter 6 for information about these changes.

1.3 What You Get

The documentation that you receive from Compaq depends upon the purchase you make:

- If you purchase a Tru64 UNIX media kit, you receive all of the online documentation, as well as printed versions of the Startup documentation described in Chapter 2.
- If you purchase the Tru64 UNIX Documentation Kit, you receive printed versions of the books described in Chapter 2.
- If you purchase an Update Contract for the media, you receive the most current version of the Documentation CD–ROM, and printed versions of the books in the Startup documentation that were revised (which for Version 5.0 is all of the Startup documentation).
- If you purchase an Update Contract for the documentation set, you receive printed versions of the books that have been updated (which for Version 5.0 are all of the books listed in Chapter 2).
- If you purchase a system with the Tru64 UNIX operating system preinstalled (often called a FIS system, for factory installed software), online versions of the following books are installed on the system:
  - *Release Notes*
  - *Installation Guide*

    Installation Guide — Advanced Topics
  - *System Administration*
  - *Network Administration*
  - *Security*
  - *Safety & Comfort Guide*
  - *Documentation Overview* (the book you are reading)

1.4 Online Documentation

The documentation for the Tru64 UNIX operating system, including the reference pages, is available on the Tru64 UNIX Documentation CD–ROM in a format that is readable with a Web browser or with the Adobe Acrobat Reader.

Most of this documentation, as well as documentation for associated applications, is also available on the World Wide Web (see Section 1.4.2). The features that are available on the Documentation CD–ROM, such as the links to other documentation, are also available on the Web.

If you are working on an Alpha system, the operating system provides a Netscape browser for viewing the HTML documentation. If you want to view the documentation on a Windows PC or Macintosh, you will need a

Java-compatible browser. Compaq recommends using Version 4.0 or higher of Netscape Navigator, Netscape Communicator, or Microsoft Internet Explorer.

To view the PDF files you need to install Version 3.0 or higher of the Adobe Acrobat Reader or the Reader plug-in to your Web browser. The Documentation CD–ROM includes Version 3.01 of the Acrobat Reader for Tru64 UNIX systems, Windows PCs, Macintosh, and several other platforms. You can also obtain the latest version directly from the Adobe Systems Inc. Web site, **http://www.adobe.com**. With the Acrobat Reader you can scroll through books, print selected sections or entire books, and copy sections to the clipboard.

1.4.1 The Documentation CD–ROM

The Documentation CD–ROM contains HTML and PDF versions of Tru64 UNIX documents, including books, white papers, and the complete set of operating system reference pages. The CD–ROM's bookshelves provide links to these HTML and PDF documents.

You can use the Tru64 UNIX Documentation CD–ROM on a Tru64 UNIX system, as well as on a Windows PC or Macintosh – laptop or workstation – or on any other system that uses the ISO 9660 Level 1 CD–ROM standard. (For additional information, see the information on the jacket containing the Documentation CD–ROM.)

To help you locate specific information in the documentation, the CD–ROM contains a copy of the AltaVista Search CD–ROM software with a complete index of the HTML documents. The AltaVista software runs on an x86–based PC with Windows 95, Windows 98, or Windows NT Version 4.0 and higher.

The CD's online Documentation Library is arranged into "bookshelves," which generally follow the structure of the printed documentation kits. Each bookshelf is represented by a bulleted list, with the bullet for each book colored to indicate the documentation subkit to which the book belongs:

- A blue bullet indicates General User Documentation.
- A purple bullet indicates Programming Documentation.
- A maroon bullet indicates the System and Network Management Documentation.
- A silver bullet indicates Supplementary Documentation, which is documentation that is not part of any kit and cannot be ordered from Compaq.
- A green bullet indicates the Reference Pages.

Books that meet the needs of different audiences appear on several bookshelves. For example, you will find the *Security* book on the General User, Programming, and System and Network Management bookshelves because it serves each of those audiences. Because it is part of the General User subkit, its bullet is blue in all of the bookshelves in which it appears.

Most cross-references are hot links, so you can follow those references from book to book, from book to reference page, from reference page to book, and from reference page to reference page.

Each book you refer to opens in a separate window so you can easily move from book to book as you gather the information you need. When following links to reference pages, the first reference page opens in a new window, and references to subsequent reference pages will appear in the same reference page window as long as you keep it on your screen.

With this release, the HTML help system has been redesigned to make it easier to use. Most of the help topics are illustrated with animated graphics in order to make the information clearer and easier to understand.

The Documentation CD–ROM also contains the `webman` viewer, a program you can use with your browser to view reference pages. For information about the `webman` viewer, see Chapter 4.

To mount the Documentation CD–ROM on Tru64 UNIX systems, PCs, and Macintosh computers, see the inside panels of the Documentation CD–ROM's jacket. The Tru64 UNIX *Installation Guide* provides an overview of using compact discs on Tru64 UNIX systems.

1.4.2 Documentation on the World Wide Web

You can find most of the Tru64 UNIX manuals in the core documentation, the reference pages, and other useful documentation on the World Wide Web.

To view this documentation, point your Web browser to the following URL:

http://www.unix.digital.com/faqs/publications/pub_page/pubs_page.html

There you will find documentation sets for Tru64 UNIX versions dating back to Version 3.0. Books for the current and most recent versions are available in HTML and PDF formats. The books for earlier versions of the operating system are available only in PostScript format.

The Web site also gives you access to the Tru64 UNIX Reference Pages, as well as to documentation for the following products:

- Device Driver Kit
- TruCluster Server (including the TruCluster reference pages)

- Advanced Server for UNIX
- Open Source Internet Solutions for Tru64 UNIX

The Web also provides updates to documentation between formal product releases. For example, Compaq maintains online Technical Updates, which provide information about restrictions and problems that have been discovered with specific versions since their release. Visit the following Web site to see the Technical Update For Version 5.0:

http://www.unix.digital.com/faqs/publications/updates/V50/TITLE.HTM

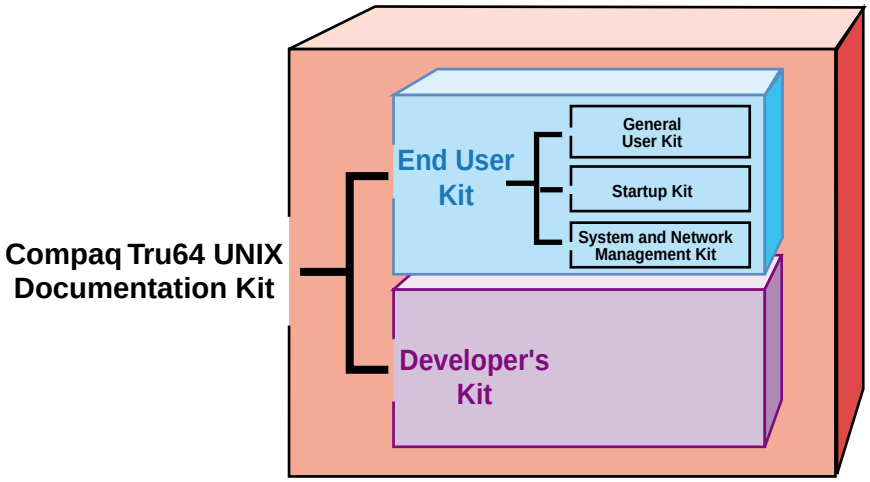
Compaq recommends that you view this site periodically to see if any new information has been added.

1.5 Printed Documentation

When you purchase a Tru64 UNIX media kit, you receive printed versions of a small number of manuals to help you install and begin using the operating system. To receive printed versions of the rest of the core documentation, you must order a Tru64 UNIX Documentation Kit or one or more of its subkits, as described in Chapter 2.

Figure 1–1 shows the structure of the Tru64 UNIX Documentation Kit. A discussion of that structure follows.

Figure 1–1: Structure of the Tru64 UNIX Documentation Kit



- End User Documentation Kit

  This kit contains all the information needed to install and use the Tru64 UNIX system. This kit is divided into three subkits as follows:

  - Startup Documentation Kit

    This kit is primarily for the person who installs the Tru64 UNIX operating system. It contains information on how to configure software components and some information (such as the *Release Notes*) needed by all Tru64 UNIX users.

    When you order the Tru64 UNIX operating system, the books in the Startup Documentation Kit are packaged with the distribution media.

  - System and Network Management Documentation Kit

    This kit is for people who are responsible for managing the Tru64 UNIX operating system or network. The manuals in this kit provide information on how to configure, manage, and tune Tru64 UNIX systems.

  - General User Documentation Kit

    This kit provides general information on how to use the Tru64 UNIX operating system. The manuals in this kit are for everyone who uses Tru64 UNIX.

- Developer's Documentation Kit

  This kit is for software developers who write programs on or for the Tru64 UNIX operating system. The books in this kit include information on tools and programming recommendations.

The printed manuals include a letter icon on the spine to help you quickly find the book you are looking for on your bookshelf. The following icons are used:

G Indicates books that are in the General User Documentation Kit

S Indicates books that are in the System and Network Administration Documentation Kit

P Indicates books that are in the Developer's Documentation Kit (the P is for programmers)

Although a book's primary audience determines the documentation kit it is packaged in, some books help meet the needs of several audiences. For example, a general user can read the *Programming Support Tools* manual (packaged in the Developer's Documentation Kit) to get advanced information about using commands such as `grep` or `awk`. The *Security* manual (packaged in the General User Documentation Kit) provides information for system administrators and programmers, as well as for general users. Keep this in mind when searching for information on specific topics.

1.6 Documentation for Layered Applications

Compaq and third-party vendors produce many applications that run on Tru64 UNIX systems. The documentation for most of these products is included with the individual application software. The following sections provide some guidance.

1.6.1 Products Packaged with Tru64 UNIX

The CD–ROM labeled Associated Products Volume 2 included in the Tru64 UNIX media kit contains the software and documentation for several separately licensed products. You do not need a license to view this documentation, which is usually provided in both HTML and PDF (or PostScript) formats.

The CD–ROM contains the following separately licensed products:

- Advanced File System (AdvFS) Utilities
- Advanced Server for UNIX
- Advanced Printing Software

This CD, as well as the one labeled Associated Products Volume 1, also contains optional software that is included with the operating system. See Section 3.3 for information about the documentation for these components.

1.6.2 Tru64 UNIX Alpha Online Documentation Library

Many of the layered applications produced by Compaq, such as DEC C++ and the Distributed Computed Environment (DCE), are included on the Tru64 UNIX Alpha Software Product Library, a collection of CD–ROMs in a kit that is updated every three months. The documentation for these products is included on the companion Tru64 UNIX Alpha Online Documentation Library CD–ROM. These software and documentation kits are separately orderable products; they are not included with the Tru64 UNIX media kit. For information about ordering these kits, contact your Compaq sales representative.

1.6.3 Device Driver Documentation

Documentation for programmers who create device driver software for the Tru64 UNIX operating system is available as a separately orderable kit. See Chapter 5 for more information about the Device Driver Documentation Kit.

2

The Tru64 UNIX Manuals

This chapter provides brief descriptions of the core documentation. For convenience, it is divided into sections that follow the structure of the printed documentation kits. See Section 1.4 for information about the online Documentation Library structure.

2.1 End User Documentation

Manuals for the end user contain the information you need to install and configure a Tru64 UNIX system, bring it up on a network, and use it. The documentation in this category is grouped in three primary areas: startup documentation, general user documentation, and system and network management documentation.

2.1.1 Startup Documentation

The startup documentation is packaged in printed form with your Tru64 UNIX media. It consists of the documents you need to install, configure, and begin using your system.

Documentation Overview

This is the book you are reading.

Installation Guide

This manual describes the procedures for performing an update installation or a basic installation of the Tru64 UNIX product on all supported processors.

Installation Guide — Advanced Topics

This manual describes the procedures for performing an advanced installation of the Tru64 UNIX product on all supported processors. It also discusses system management procedures in a standalone environment.

Quick Reference Card (available only in printed form)

This foldout card provides fingertip access to the format of common user commands, such as `cd`, `chmod`, `lpr`, and `man`. The card also describes options that are commonly used with each command.

Additionally, this card provides command summaries for the `vi`, `emacs`, `Mail`, and `mail` applications, and it summarizes the rules for forming regular expressions. It provides a description of command control symbols (such as `|`, the pipe symbol) and gives definitions of shell environment variables and metacharacters.

Release Notes for Version 5.0

The *Release Notes for Version 5.0* are for all users of the Tru64 UNIX operating system. This book includes the following information:

- An overview of the new and changed features of the Version 5.0 software
- Announcements about features and interfaces that are scheduled for removal in future versions of Tru64 UNIX
- Information about installing the software
- Information that is relevant to specific Compaq Alpha processors
- Information about the base operating system
- Information about the development environment
- Information about the graphical interfaces
- Additions and corrections to the books in the Tru64 UNIX documentation set
- Information about the contents of the media and disk-space requirements
- Information about changes to time zone directories and files

Compaq recommends that you read this information before installing or using the Tru64 UNIX operating system.

Note

Compaq maintains an online *Technical Update* that contains information about restrictions and problems that have been discovered since Version 5.0 began shipping. To view this document on the Web, go to the following URL:

http://www.unix.digital.com/faqs/publications/updates/V50/TITLE.HTM

Compaq recommends that you visit this site periodically to see if any new information has been added.

Technical Overview

This manual describes the major components of the Tru64 UNIX operating system. Among its topics are the following:

- An introduction to the operating system, including discussions of recent enhancements, licensing, and available optional software
- The various features available to system administrators
- The available networking protocols and applications
- A summary of the supported file systems
- Issues regarding the kernel, including system attributes and parameters, symmetric multiprocessing, and virtual memory
- The major features of the Tru64 UNIX development environment
- The requirements of the C2 evaluation class and additional security features
- Issues concerning internationalization

Tru64 UNIX Version 5.0 CD–ROMs

This one-page document provides a brief listing of the components on the CD–ROMs in the media kit. *Version 5.0 CD–ROMs*. An online version of this document is included on the Supplementary bookshelf of the Tru64 UNIX Documentation CD–ROM.

Update Installation Quick Reference Card (available only in printed form)

This foldout card provides easy access to the information needed to upgrade a Tru64 UNIX system to the next version. Update installations preserve disk partitions, file systems, and file customizations.

The information on this card is covered in detail in Chapter 2 of the *Installation Guide*.

2.1.2 General User Documentation

General user documentation contains important information for all users of the Tru64 UNIX operating system. The books in this category provide introductory information for people who are unfamiliar with Tru64 UNIX.

CDE Companion

This manual provides an introduction to Compaq's implementation of the Common Desktop Environment (CDE), an easy method of interacting with the Tru64 UNIX operating system.

For users migrating from DECwindows Motif to CDE, this book serves as an introduction to the new environment, providing information on how to use CDE to complete tasks previously done by using DECwindows.

For users who are new to desktop environments, this book provides quick-start information on topics such as logging into the system, navigating the system, and using and managing the desktop and applications.

This book can be used in conjunction with the *Common Desktop Environment: User's Guide*.

Command and Shell User's Guide

This manual introduces the basic features of the Tru64 UNIX operating system. It describes how to use the command line interface and to perform such tasks as copying files and creating directories. It also describes how to use the shells and their built-in commands.

Although this manual is primarily for users who have little or no familiarity with UNIX-compatible systems, experienced users can find useful shortcuts and tips.

Master Index

This book can help you find the information you need in the core documentation set, as follows:

- In the printed version of this book, the index items advise you in which books to look and direct you to the pages where the topic is discussed.
- In the online version of this book, clicking on an index item in many cases will take you to the appropriate book at the point where the topic is discussed. For the Common Desktop Environment books and various other books, clicking on the index item will take you only to the beginning of the appropriate book, where you can use the table of contents or the book's index to find out where the topic is discussed. (This restriction is caused by the tools that create the online books; it will be corrected in a future release.)

Security

This manual provides information needed by users and system administrators who are working on Tru64 UNIX systems that have the optional enhanced security subsets installed. It also provides information needed by developers who are writing programs that will run on secure systems. The enhanced security features help protect systems or data from access by unauthorized users.

This manual also provides information about security on systems that do not have the enhanced security subsets installed.

The information is organized as follows:

- Part 1 describes how to use the Tru64 UNIX operating system with enhanced security from the command line.
- Part 2 describes how to administer the operating system's enhanced security, which includes enhanced passwords and the audit subsystem. The Security Integration Architecture (SIA) is also discussed. This part of the manual assumes prior knowledge and experience administering secure systems.
- Part 3 describes how to write programs that run on the Tru64 UNIX operating system with enhanced security. This part assumes general programming knowledge, including knowing how to use the Tru64 UNIX programming tools.

2.1.3 System and Network Management Documentation

System and network management documentation provides information on topics such as configuring systems and networks, maintaining disks, and using system administration tools.

AdvFS Administration

AdvFS is the default file system for the Tru64 UNIX operating system. It provides rapid crash recovery and a flexible structure that allows you to manage your file system while it is on line.

The AdvFS Utilities, which are available as a layered product, extend the file system capabilities by including utilities to add volumes, create clones, stripe files, and balance file domains.

This manual provides task-based information and instruction on using, configuring, troubleshooting, tuning, and administering AdvFS base and utilities functions.

Guide to Prestoserve

Prestoserve speeds up synchronous disk writes, including Network File System (NFS) server access, by reducing the amount of disk I/O.

This manual describes how to manage and maintain a Tru64 UNIX system that includes the optional Prestoserve hardware and software.

Kernel Debugging

The manual describes using the `dbx`, `kdbx`, and `kdebug` debuggers to find problems in kernel code. It also describes how to write a `kdbx` utility extension and how to create and analyze a crash dump file.

This manual is for system administrators responsible for modifying, rebuilding, and debugging the kernel configuration. It is also for system programmers who need to debug their kernel-space programs.

Logical Storage Manager

The Logical Storage Manager (LSM) software provides high data availability, better performance, and greater storage management flexibility through online support for disk storage devices on Alpha systems.

This manual explains LSM concepts and how to install and administer the LSM software.

Network Administration

This manual provides information on establishing a Tru64 UNIX system on a network and configuring network software such as NFS and DNS. It also explains how to manage a network and network applications and how to solve problems that might arise.

This manual is for experienced system and network administrators who have knowledge of TCP/IP networking concepts and network configuration. Readers should also have knowledge of operating system concepts, commands, and configuration.

Sharing Software on a Local Area Network

This manual describes Remote Installation Services (RIS) and Dataless Management Services (DMS). The RIS utility is used for installing software across a network, instead of using locally mounted media. DMS allows a server system to maintain the root, `/usr`, and `/var` file systems for client

systems. Each client system has its own root file system on the server, but clients share the /usr and /var file systems.

Software License Management

This manual describes how to use the License Management Facility (LMF) to manage software licenses from Compaq.

Although intended primarily for system administrators responsible for managing software licenses on Tru64 UNIX systems, this manual also provides information for anyone who uses licensed software on Tru64 UNIX systems.

System Administration

This manual describes how to configure, use, and maintain the Tru64 UNIX operating system. It includes information on general day-to-day activities and tasks, changing system configurations, and locating and eliminating sources of trouble.

System administrators are provided with overviews of administrative concepts, as well as step-by-step instructions to perform necessary tasks using the either the SysMan graphical interface or the command-line tools.

This manual is for system administrators responsible for managing the operating system. It assumes a knowledge of operating system concepts, commands, and configurations.

System Configuration and Tuning

This manual describes how to set up and tune high-performance and high-availability systems running the Tru64 UNIX operating system. It can help system administrators to accomplish many system tasks, including the following:

- Determine the needs of the environment
- Configure and tune a system that will meet current and future needs
- Understand how hardware, operating system subsystems, and layered applications interact to affect system performance
- Learn recommended ways to improve performance

This manual replaced the *System Tuning and Performance Management* manual that existed prior to the DIGITAL UNIX Version 4.0D release.

System Configuration Supplement: OEM Platforms

This manual provides information needed to set up OEM platforms running the Tru64 UNIX operating system. It helps system and network

administrators configure Alpha VME single-board computers (SBCs), PCI/ISA modular SBCs, and VMEbus backplane networks in which SBCs operate as Ethernet nodes.

This manual is for experienced system and network administrators who are thoroughly familiar with their platform's I/O bus and with the operating system concepts, commands, and configurations.

X Window System Administrator's Guide (available only in printed form)

This manual, published by O'Reilly & Associates, describes how to customize a wide range of X Window System environments, from an individual workstation to groups of workstations and X terminals connected on a network. Major topics include security, the X display manager (`xdm`), fonts, color, X terminals, and X client applications.

X Window System Environment

This manual describes various aspects of the X Window System environment as it is implemented on Tru64 UNIX. It provides information on how to perform system administration tasks for the Tru64 UNIX X Window System environment, and describes how to customize X Window System resources and key mappings. It also provides information about programming within the Tru64 UNIX X Window System environment.

2.2 Documentation for Program Developers

The programming documentation consists of manuals designed for software developers who write software applications on or for the Tru64 UNIX operating system.

Assembly Language Programmer's Guide

This manual describes the Alpha hardware architecture's assembly language, which is supported by the Tru64 UNIX compiler system. The manual describes the assembly language syntax rules, and how to write assembly language programs.

This manual is for system software developers who are writing assembly language programs on or for Tru64 UNIX.

Asynchronous Transfer Mode

This manual describes the Tru64 UNIX Asynchronous Transfer Mode (ATM) subsystem, how to configure the subsystem, and how to use the ATM kernel interfaces. It is written for experienced UNIX kernel programmers who are responsible for writing ATM device drivers and

kernel modules. After reading this manual, a kernel programmer should be able to do the following:

- Understand the ATM subsystem architecture
- Configure ATM software
- Understand how the different kernel interfaces operate
- Write a kernel module

This manual does not describe the application programming interface (API) that user-level applications would use to access the ATM subsystem. This manual is not an ATM networking tutorial.

Calling Standard for Alpha Systems

This manual defines the requirements, mechanisms, and conventions used in the interface that supports procedure calls on Tru64 UNIX for Alpha systems. The standard defines data structures, constants, algorithms, conventions, methods, and functional interfaces, which enable a native, user-mode procedure to operate correctly in the multilanguage and multithreaded Tru64 UNIX environment on Alpha hardware.

Although this manual primarily defines requirements for compiler and debugger writers, the information applies to procedure calling for all programmers at all levels of programming.

Compaq Portable Mathematics Library

This manual provides reference and exception information for the Compaq Portable Mathematics Library (CPML) software. This manual documents the CPML routines and, in particular, how they behave when given an exception input argument. It also documents operating system entry points and supported floating-point data types.

This manual is for compiler writers and system and application programmers who do not have high-level language support of CPML routines in their language of choice, but instead need to access CPML routines directly from application programs.

DEC C Language Reference Manual

This manual provides reference information for using the DEC C language on Compaq systems. DEC C is an ISO/ANSI-compliant C compiler for Tru64 UNIX and OpenVMS VAX and Alpha systems.

This manual is based on the ISO C Standard (ISO 9899:1990[1992]), formerly the ANSI X3J11 committee's standard for the C programming

language (commonly called ANSI C). All library functions and language extensions to the ANSI C standard are also described.

This manual is intended for programmers who need reference information on the DEC C language. For task-oriented information or platform-specific information see the cc(1) reference page and the *Programmer's Guide*.

Guide to DECthreads

This manual provides usage and reference information on DECthreads routines. It provides information on the three DECthreads interfaces used to perform multithreaded operations: cma, pthread, and pthread exception-returning.

This manual is for programmers writing multithreaded applications. It assumes experience with a high-level programming language (such as C), with UNIX operating systems, and with UNIX software development tools.

Guide to Preparing Product Kits

This manual describes the procedures for creating, maintaining, and installing the collections of files and directories that make up a layered products kit. A kit is the standard mechanism by which layered product modifications are delivered and maintained on a Tru64 UNIX system. Kits are distributed on CD–ROM, diskettes, or tape for installation on customer's systems.

This is the same manual that is included in the Tru64 UNIX Device Drivers Kit (DDK).

Guide to Realtime Programming

This manual is for programmers who are developing realtime applications on Tru64 UNIX systems. It provides information on writing new realtime applications and porting existing realtime applications from other systems.

This manual does not present function syntax or reference information; the online reference pages provide that information.

This manual is for application programmers or system engineers who are already familiar with the C programming language. It assumes experience with UNIX operating systems and with UNIX software development tools.

Ladebug Debugger Manual

The Ladebug Debugger is a tool for debugging executable programs at the source-code and machine-code levels. It can debug programs written in C and C++, Ada, COBOL, and Fortran.

This manual is for developers who need to debug multiprocess and multithreaded applications, perform kernel debugging, and perform remote client/server debugging. It is organized in two parts:

- Part 1 describes the Ladebug graphical (window-based) user interface (GUI), which provides access to all of the major Ladebug features.
- Part 2 describes the command line interface, which can be used from within the GUI or from the shell-level prompt.

Network Programmer's Guide

This manual describes the Tru64 UNIX network programming environment. It describes in depth the X/Open Transport Interface (XTI) and the sockets and STREAMS programming frameworks, including information about system calls, header files, and libraries. Additionally, it provides information about porting sockets-based applications to XTI.

This manual also describes the software bridge `ifnet` (STREAMS module and DLPI STREAMS pseudodevice driver) that the Tru64 UNIX operating system supports. This bridge allows programs that use sockets-based protocol stacks to access STREAMS drivers, and programs that use STREAMS-based protocol stacks to access BSD-based drivers.

This manual is for experienced UNIX programmers.

OSF/Motif Programmer's Guide (available in printed form only)

This manual (developed by the OSF and published by Prentice Hall) provides programming information on how to use the various components of the OSF/Motif environment: the Toolkit, window manager, and user interface language.

This manual is for programmers who want to create applications in the OSF/Motif environment.

OSF/Motif Style Guide (available in printed form only)

This manual (developed by the OSF and published by Prentice Hall) provides a framework of behavior specifications to guide application developers, widget developers, and window manager developers in the design and implementation of products consistent with the Presentation Manager and the OSF/Motif user interface.

This manual establishes consistent behavior among new products by drawing out common elements from a variety of current behavioral models. It anticipates the evolution of graphical user interfaces as new technology becomes available and as the use of the OSF/Motif user interface spreads.

This manual is for programmers and interface designers developing OSF/Motif applications who want to present a uniform and usable software interface consistent with other OSF/Motif applications.

Programmer's Guide

This manual describes the programming environment of the Tru64 UNIX operating system, with an emphasis on the C programming language.

This manual is for all programmers who use the Tru64 UNIX operating system to create or maintain programs in any supported language.

Programmer's Guide: STREAMS (available only in printed form)

This manual (developed by AT&T and published by Prentice Hall) provides information on the use of the STREAMS mechanism at the user and kernel levels. It contains introductory information for those who are unfamiliar with the STREAMS mechanism.

This manual addresses topics such as using STREAMS to monitor, control, and poll STREAMS; designing and implementing STREAMS modules and drivers; and using STREAMS-based pipes and FIFOs. It also describes the STREAMS multiplexing facility and the STREAMS-based terminal and pseudo-terminal subsystems.

Programming Support Tools

This manual describes commands and utilities for text manipulation, macro and program generation, and source file management on Tru64 UNIX.

Although the commands and utilities described in this manual are primarily for programmers, some of them (such as `grep`, `awk`, `sed`, and the Source Code Control System (SCCS)) are useful for general users.

This manual assumes that the reader is a moderately experienced user of UNIX systems.

Programming with ONC RPC

This manual provides an overview of high-level programming with remote procedure calls (RPC) in the Open-Network Computing Environment (ONC). It describes how to use the `rpcgen` protocol compiler to create RPC applications and describes the RPC programming interface.

This manual is for programmers who want to write network applications without knowledge of the underlying network.

Writing Software for the International Market

Internationalization (often called I18N) is the process of designing or adapting programs to interact with users in their own language and to reflect the culture of the user's region.

This manual provides an overview of internationalization concepts. It also provides details about writing internationalized programs, creating and using message catalogs, and developing or customizing locales.

This manual is for programmers developing internationalized applications for the Tru64 UNIX operating system.

Writing Kernel Modules

Kernel modules are binary images containing code and data structures that run in the UNIX kernel. These modules provide a level of common code that improves the efficiency of a system by combining like tasks in a single area and eliminating redundant code.

This manual provides information for developers who want to augment the kernel with modules tailored to their particular environment.

3

Supplementary Documentation

To help meet specific needs when working with the Tru64 UNIX operating system, Compaq makes available documentation that supplements the Tru64 UNIX documentation set. Some of these documents are produced by Compaq and others are produced by other organizations and provided by Compaq.

The Documentation CD–ROM, Operating System CD–ROM, Associated Products CD–ROMs volumes 1 and 2 all contain supplementary documentation. Much of this documentation pertains to individual components.

For example, the Associated Products Volume 1 CD–ROM contains installation guides for the Multimedia Services for Tru64 UNIX run-time environment, DECevent, and the Porting Assistant. Other component-specific information includes release notes, Software Product Descriptions (SPD), and user information.

This *Documentation Overview* does not list all of the supplementary documentation on the CD–ROMs, but it does provide some guidance in the following sections.

3.1 Documentation CD–ROM

You can access the supplementary documentation described in the following sections by clicking on the bookshelf called *Supplementary Documents* on the Documentation CD–ROM. Links to some of this documentation are also provided on other bookshelves.

Much of this documentation is provided in both HTML and PDF formats; some is provided only in one of the two formats. If your system is configured as suggested, you can click on the HTML link to view the document in your Web browser or click on the PDF link to view the document using Adobe Acrobat Reader.

3.1.1 Advanced Printing Software *User Guide*

This manual provides information about using the Advanced Printing Software, which is included as an optional component of the Tru64 UNIX

operating system. The *User Guide* describes how to submit jobs to your printer using the command-line interface. It also shows you how to set up your local print environment and monitor jobs you have submitted.

This manual is also packaged with the software on the Associated Products Volume 2 CD–ROM, as is the following companion documentation:

- Advanced Printing Software *System Administration and Operation Guide*
- Advanced Printing Software *Command Reference Guide*
- Advanced Printing Software *Release Notes*
- Advanced Printing Software *Installation Guide*

3.1.2 CD–ROM Contents Listing

This is an online version of the document called *Tru64 UNIX Version 5.0 CD–ROMs*, which is included in the Tru64 UNIX media kit. It provides brief listings of the components on the CD–ROMs in the media kit.

3.1.3 Common Desktop Environment and TookTalk Documentation

Prior to this release, the Common Desktop Environment documentation was provided in hard copy format and was therefore considered part of the core documentation set. It is now available only on line.

3.1.3.1 Documentation for Users and System Administrators

The following books were previously included in the General User Documentation Kit:

- *Common Desktop Environment: Advanced User's and System Administrator's Guide*

  This manual describes how to customize the appearance and behavior of CDE. It provides information on topics such as the following:

  – Customizing system initialization, login, and session initiation

  – Adding applications and providing interface representations for applications and their data

  – Configuring desktop processes, applications, and data across the network

  – Customizing desktop services such as window management, printing, colors, and fonts

  This book is intended for users who want to perform customizations that cannot be accomplished using the desktop user interface. This book

is also intended for system administrators; many of the tasks in this book require superuser permission.

- *Common Desktop Environment: User's Guide*

  This manual describes the basic features of the CDE and describes how to use the desktop and the desktop applications. It expands upon some of the topics in the *CDE Companion* and provides figures of the graphical interface as it is displayed on workstation screens.

3.1.3.2 Documentation for Windows Developers

The following books were previously included in the Windows Programming Documentation Kit:

- *Common Desktop Environment: Application Builder User's Guide*

  This manual introduces the Application Builder and explains how to use it to build CDE applications. Because the Application Builder helps to easily create and modify user interfaces, it is a powerful tool for programmers, user interface designers, and project managers.

- *Common Desktop Environment: Desktop KornShell User's Guide*

  This manual provides the information needed to create Motif applications with KornShell (`ksh`) scripts. It also provides several example scripts of increasing complexity.

  This manual is for programmers who want to develop Motif applications using KornShell scripts rather than the C programming language. This manual assumes knowledge of KornShell programming, Motif, and the Xt Intrinsics. Familiarity with the X programming library (`Xlib`) is also assumed.

- *Common Desktop Environment: Help System Author's and Programmer's Guide*

  This manual describes how to develop online help for CDE applications. It describes how to create help topics and how to integrate online help into a CDE application.

  This manual is for application programmers who want to do the following:

  – Design, create, and view online help information

  – Create software applications that provide a fully integrated help facility

- *Common Desktop Environment: Internationalization Programmer's Guide*

This manual provides information for internationalizing the desktop and enabling applications to support various languages and cultural conventions in a consistent user interface.

This manual is for CDE application programmers whose products are available worldwide.

- *Common Desktop Environment: Programmer's Guide*

  This manual contains the information needed to integrate an existing application into the CDE desktop. It also describes how to write new CDE applications.

  This manual assumes a familiarity with Motif, X, UNIX, and C programming.

- *Common Desktop Environment: Programmer's Overview*

  This manual provides a high-level view of the development environment and the developer documentation set for CDE. It is for the following audiences:

  – Application developers who develop new CDE applications, or integrate existing OSF/Motif applications into CDE

  – Managers or project leaders interested in designing a project involving applications that will run on CDE

- *Common Desktop Environment: Style Guide and Certification Checklist*

  This manual provides style guidelines for CDE application design and lists the requirements for CDE application-level certification. CDE requirements consist of the OSF/Motif Version 1.2 requirements with CDE-specific additions.

- *Common Desktop Environment: Product Glossary*

  This glossary provides a comprehensive list of terms used in the Common Desktop Environment.

  This manual is for all CDE users.

3.1.3.3 ToolTalk Service Documentation

ToolTalk is an interapplication communication service that is included as part of the Common Desktop Environment. ToolTalk provides a way for applications and desktop components to request services of each other and to announce events. The following books provide information about the ToolTalk service:

- *Common Desktop Environment: ToolTalk Messaging Overview*

  This manual describes how the ToolTalk service works and how it uses information that applications supply to deliver messages. It also

describes how applications use the ToolTalk service and ToolTalk components.

This manual assumes familiarity with the ToolTalk service, UNIX operating system commands, system administrator commands, and system terminology.

This book was previously included in the Windows Programming Documentation Kit:

- *ToolTalk User's Guide*

  This manual is useful to developers who create or maintain applications that use the ToolTalk service to interoperate with other applications; it is also useful to system administrators who set up workstations. This manual assumes the reader is familiar with operating system commands, system administrator commands, and system terminology.

- *ToolTalk Reference Manual*

  This manual describes components of the ToolTalk application programming interface such as enumerated types and functions. It also describes ToolTalk-enhanced operating system shell commands, error messages, and standard ToolTalk messaging sets.

3.1.4 *BIND Configuration File Guide*

The *BIND Configuration File Guide* describes the Berkeley Internet Name Domain (BIND) Version 8 and its implementation as an Internet name server for BSD-derived operating systems.

Version 8 is more configurable than previous releases of BIND; there are entirely new areas of configuration, such as access control lists and categorized logging. Also, you can now selectively use many options that previously applied to all zones. These features, plus a consideration of future configuration needs, led to the creation of a new configuration file format.

3.1.5 *DECevent Translation and Reporting Utility*

DECevent provides an interface between a system user and the operating system's event logger. DECevent can help system administrators to troubleshoot Tru64 UNIX system problems.

This manual describes the DECevent command features related to the translation and reporting of events on Tru64 UNIX operating systems. It contains an overview of the utility, information on how to obtain help for the utility, and information about all the commands necessary to translate event logs on Tru64 UNIX operating systems.

3.1.6 *How to Order Tru64 UNIX Documentation*

The file titled *How to Order Tru64 UNIX Documentation* duplicates the information in Chapter 6. It provides an easy way to move from the bookshelf to the information and the Web links you need to order Tru64 UNIX documentation.

3.1.7 Java Documentation

The Tru64 UNIX operating system includes a Java Development Kit (JDK), which provides tools to develop and run Java applets and programs on the Tru64 UNIX operating system.

Access to the Java documentation depends upon whether the Java Development Kit is installed on your system and how your system administrator has set up your system. Clicking on *Java Overview* on the Supplementary Documentation bookshelf takes you to a page that provides information about viewing the Java documentation. This page also provides a Web link to the JavaSoft JDK documentation at the Sun Microsystems Java site.

3.1.8 *New and Changed Features from Prior Releases*

The *New and Changed Features from Prior Releases* document lists the major features that were introduced and changes that were made to the Tru64 UNIX operating system from Version 4.0 through Version 4.0F. The information in this document was originally presented in the *New and Changed Features* chapter of the operating system *Release Notes* for each of those versions.

3.1.9 *Object File / Symbol Table Specification*

The *Object File / Symbol Table Specification* fills a need for technical information for compiler and debugger writers and other developers who must access or manipulate object files on the Tru64 UNIX operating system. It supplements or replaces information that has been previously available in the Assembly Language Programmer's Guide. A familiarity with basic program development and symbol table concepts is assumed.

3.1.10 *Performance Manager*

Performance Manager is a real-time performance manager that allows system administrators to detect and correct performance problems in Tru64

UNIX systems. The three primary components of the Performance Manager are a graphical user interface, the Performance Manager daemon, and an activity daemon. An additional daemon monitors systems running the TruCluster software.

This manual explains the concepts of the Performance Manager software, and describes tasks such as monitoring, thresholding, archiving, and distributed command execution.

3.1.11 *Safety & Comfort Guide*

The *Safety & Comfort Guide* describes how to best set up your workstation environment to minimize the possibility of injury and provide you with the maximum of comfort. This guide focuses on posture and on other health and work habits for computer users. Additionally, it provides product safety information applicable to all Compaq products.

3.1.12 *sendmail Installation and Operation Guide*

The *sendmail Installation and Operation Guide* describes the configuration file for Version 8.7 of the `sendmail` utility, which implements a general-purpose internetwork mail routing facility under the Tru64 UNIX operating system.

The `sendmail` utility is not tied to any one transport protocol — its function can be likened to a crossbar switch, relaying messages from one domain into another. In the process, it can do a limited amount of message header editing to put the message into a format that is appropriate for the receiving domain. All of this is done under the control of a configuration file.

3.1.13 Technical References for Asian Languages

The following guides provide language-specific information and describe the features of several Asian languages supported on the Tru64 UNIX system:

- *Technical Reference for Using Chinese Features*
- *Technical Reference for Using Japanese Features*
- *Technical Reference for Using Korean Features*
- *Technical Reference for Using Thai Features*

3.1.14 *Tru64 UNIX Year 2000 Readiness*

The *Tru64 UNIX Year 2000 Readiness* document contains information about the Tru64 UNIX Year 2000 program to help you prepare your system

for the turn of the century. It also includes other important Year 2000 information on previous Tru64 UNIX versions and layered products, and describes testing methodologies and guidelines.

Included in this document are links to related Web sites.

3.1.15 X Window System Documentation

The documentation described in this section pertains to X Windows System, Version 11, Release 6. These documents are provided by the X Consortium.

- *Inter-Client Communication Conventions Manual*

  This document proposes suitable conventions for interclient communications with X Version 11 software. The proposed conventions do not attempt to enforce any particular user interface.

- *X Toolkit Intrinsics — C Language Interface*

  Intrinsics are a programming library tailored to the special requirements of user-interface construction within a network window system — specifically the X Window System. The Intrinsics and a widget set make up the X Toolkit. This document describes the X Toolkit Intrinsics.

- *X Logical Font Description Conventions*

  This document provides a standard logical font description and the conventions to be used in the core protocol so that clients can query and access screen type libraries in a consistent manner across all X servers.

- *Xlib — C Language X Interface*

  This document provides reference information for the low-level C language interface to the X Window System protocol. It provides a detailed description of each function in the library, as well as a discussion of the related background information.

- *X Window System Protocol*

  This document describes the X Window System protocol.

- *X Window System: The Complete Reference to Xlib, X Protocol, ICCCM, XFLD*

  This 1000–page manual describes the features of the X Window System Version 11, Release 5. It was written by Robert W. Scheifler and James Gettys, who were on the team that designed X Window System, and was originally published by Digital Press.

3.2 Operating System CD–ROM

The documentation described in the following sections is included on the Tru64 UNIX Operating System CD–ROM. Some of this documentation is available only when you install the software it describes.

3.2.1 Software Product Descriptions

A Software Product Description (SPD) is the legal description of the Tru64 UNIX product. It describes the software and gives information about its capabilities and about the hardware it supports. With this release, the SPD for the operating system provides a listing of system limits.

PostScript versions of the SPDs are located on the CD–ROM in the *mount_point*/DOCUMENTATION/POSTSCRIPT directory. The files are named as follows:

```
Tru64_UNIX_Operating_System_SPD.ps
Tru64_UNIX_Developers_Extensions_SPD.ps
Tru64_UNIX_Server_Extensions_SPD.ps
Prestoserve_for_Tru64_UNIX_SPD.ps
Tru64_UNIX_Logical_Storage_Manager_SPD.ps
```

PDF versions of the SPDs are located on the CD–ROM in the *mount_point*/DOCUMENTATION/ACRO directory. You view PDF files using the Adobe Acrobat Reader. The files are named as follows:

```
Tru64_UNIX_Operating_System_SPD.pdf
Tru64_UNIX_Developers_Extensions_SPD.pdf
Tru64_UNIX_Server_Extensions_SPD.pdf
Prestoserve_for_Tru64_UNIX_SPD.pdf
Tru64_UNIX_Logical_Storage_Manager_SPD.pdf
```

3.2.2 Listings of Patches to the Operating System

Listings of fixes (usually called patches) that Compaq has made to each version of the Tru64 UNIX operating system are provided in text files on the CD–ROM in the *mount_point*/DOCUMENTATION/TEXT directory. The files are named as follows:

```
Digital_UNIX_V3_2C_CLD_Fixes.txt
Digital_UNIX_V3_2D_CLD_Fixes.txt
Digital_UNIX_V3_2G_CLD_Fixes.txt
Digital_UNIX_V4_0_CLD_Fixes.txt
Digital_UNIX_V4_0A_CLD_Fixes.txt
Digital_UNIX_V4_0B_CLD_Fixes.txt
DIGITAL_UNIX_V4_0D_CLD_Fixes.txt
DIGITAL_UNIX_V4_0E_CLD_Fixes.txt
Tru64_UNIX_V4_0F_CLD_Fixes.txt
```

```
Tru64_UNIX_V5_0_CLD_Fixes.txt
```

To obtain Tru64 UNIX patch kits from Compaq, visit the following Web site:

http://www.service.digital.com/patches/index.html

3.2.3 X Image Extension Documentation

The X Image Extension (XIE) code (developed by the X Consortium) provides a powerful mechanism for the transfer and display of virtually any image on X-capable hardware. Documentation for XIE is installed in compressed format in the `/usr/share/doclib/xie` directory.

Before you can view or print a copy of one of the XIE documents, you must uncompress it using the `gunzip` command. For information about `gunzip`, see the `gzip`(1) reference page.

The following list describes the XIE documentation. The names of the individual files are listed after the titles.

- *X Image Extension Overview* (`overview.ps.gz`)

  This document provides general information about the X Image Extension code. The topics covered include XIE design goals, XIE historical summary, XIE architecture, element definitions, and subsetting.

- *XIElib Specification* (`xielib.ps.gz`)

  This document contains reference information about the XIElib functions, XIElib events, and XIElib errors. The Functions section covers such functions as startup, LUT, photomap, ROI, photoflo, client data, abort and await, photoflo element, technique, and free.

- *XIE Sample Implementation Architecture* (`xieSIarch.ps.gz`)

  This document provides an architecture overview of XIE, including chapters on extension initialization, memory management, request dispatching, data representation, data structures, protocol requests, DIXIE photoflo management, DDXIE photoflo management, and photo elements.

- *X Image Extension Protocol Reference Manual, Version 5.0* (`XIEProto.ps.gz`)

  This document specifies the X wire protocol for XIE. It defines the syntax, structure, and semantics of the XIE protocol elements. The topics covered include syntax specification, parameter types, resources, pipelined processing, import elements, process elements, export elements, events and errors, techniques, service class, and protocol encodings.

3.3 Associated Products CD–ROMs

The Associated Products CD–ROMs contain documentation for various components of Tru64 UNIX. The document called *Tru64 UNIX Version 5.0 CD–ROMs*, described in Section 3.1.2, provides a list of the applications located on these CDs, as do the `00README.TXT` files located on the CDs.

Both CDs provide a graphical user interface that you can use to access documentation and to install product software. To use this interface, invoke Netscape and go to the following location:

`file://<`*`mount_point`*`>/index.html`

A list of products on the CD–ROM will be displayed. When you click on a product name you will be given the option to install the product or access the documentation, if applicable.

4

The Tru64 UNIX Reference Pages

The Tru64 UNIX operating system provides an extensive set of reference pages (also called man pages or manual pages), each of which describes one topic, such as a command, function, or file. This chapter discusses the reference pages.

4.1 Reading Reference Pages On Line

You can use the `webman` utility to read the reference pages on line using your Web browser, and print them by clicking on the browser's Print menu. Alternatively, you can use the `xman` or `man` utilities from the command line to read the reference pages.

Your system administrator determines at system installation time whether to make the operating system reference pages available on your system. If you receive an error message when you try to read reference pages, the problem might be that they are not installed.

Your system administrator also decides whether to install the reference pages that are included with layered products. This includes products such as theTruCluster Server, which provides a full set of its own reference pages.

4.1.1 Using the Documentation CD–ROM

The Library on the Documentation CD–ROM includes a bookshelf that contains the full set of base operating system reference pages in HTML form. You can view these references pages with your Web browser just as you would view any of the HTML books. These references pages are indexed and hot-linked with the books for easy access.

4.1.2 Using webman

The Documentation CD–ROM contains the `webman` viewer, a UNIX program you can use with your browser to view the reference pages that are installed with the operating system. The `webman` viewer is most useful when the Documentation CD–ROM is not mounted, or when you need to access reference pages that have been installed with layered applications.

For information about installing and using webman, see the following file on the Documentation CD–ROM:

/*mount-point*/DOCS/HTML/webman/Installing_webman.txt

See Section 1.4.1 for information about the Documentation CD–ROM.

4.1.3 Using the xman Command

The xman command starts an X Window System reference-page browsing tool. One of the functions of the tool is to display a list of the reference pages. You display a reference page by double-clicking on its name in the list. For more information about the xman command, start up the tool by entering the following command:

```
% xman &
```

The ampersand (&) runs the command in the background, allowing the command line to be used for other tasks. The application displays a small window that contains three buttons. Click on the Manual Page button to read a reference page about the xman command.

4.1.4 Using the man Command

The man command displays the reference page specified on the man command line. For more information about the man command, read the man(1) reference page. To display this reference page, enter the following command:

```
% man man
```

4.2 Reading Reference Pages in Printed Books

Compaq provides a separately orderable Reference Pages Documentation Set. In this printed format, the Reference Page books are grouped according to their sections, each of which contains one or more volumes. The books contain an R on the spines to help you to quickly identify them as reference page books on your bookshelf.

To order the 22–volume Reference Page Kit, refer to Chapter 6 or contact your Compaq representative.

Note

The Reference Page Kit is usually updated only with major releases of Tru64 UNIX. Therefore, the online reference pages

provide more current information about features that are added or revised in minor releases.

4.3 Overview of the Sections

The reference pages are grouped into sections, according to their function, audience, or both. The section numbers appear as numbers in parentheses next to the name of the reference page. For example:

```
lpr(1)
automount(8)
createlabel(3)
```

The following list describes the section numbers:

Reference Pages Section 1

Section 1 describes user commands that are available to everyone who uses the Tru64 UNIX operating system.

In printed form, this section is divided into five volumes.

Reference Pages Section 2

Section 2 defines system calls (entries into the Tru64 UNIX kernel) that programmers use. The introduction to Section 2, `intro`(2), lists error numbers with brief descriptions of their meanings. The introduction also defines many of the terms used in this section.

In printed form, this section is in one volume.

Reference Pages Section 3

Section 3 describes the routines available in Tru64 UNIX programming libraries, including the C library, Motif library, and X library. This section is for programmers.

In printed form, this section is divided into seven volumes.

Reference Pages Section 4

Section 4 describes the format of system files and how the files are used. The files described include assembler and link editor output, system accounting, and file system formats. This section is for programmers and system administrators.

In printed form, this section is divided into two volumes.

Reference Pages Section 5

Section 5 contains miscellaneous information, including ASCII character codes, mail-addressing formats, text-formatting macros, and a description of the root file system. This section is for programmers and system administrators.

In printed form, this section is divided into two volumes.

Reference Pages Section 7

Section 7 describes special files, related device driver functions, databases, and network support. This section is for programmers and system administrators.

In printed form, this section is in one volume.

Reference Pages Sections 8 and 1m

Sections 8 and 1m describe commands for system operation and maintenance. These sections are for system administrators.

In printed form, these sections are divided into three volumes.

Reference Pages Sections 9

Section 9 describes data structures and routines used in writing kernel modules. This section is for developers.

In printed form, this section is in one volume.

5

Device Driver Documentation

The Tru64 UNIX Device Driver Kit (DDK) provides programming information specifically for system engineers who are developing device drivers for the Tru64 UNIX operating system. Information about the DDK is provided here because of the close relationship between the device driver documentation and the Tru64 UNIX documentation set.

The DDK contains printed versions of the device driver documentation and a CD–ROM that provides driver development tools, device driver examples, X Consortium code, white papers, HTML and PDF versions of the device driver documentation, and the device driver reference pages.

The printed device driver books contain a D on the spines to help you to quickly identify them as device driver books on your bookshelf.

Note

The Device Driver Kit is revised independently from the operating system. The following information is for the Release 2 of the DDK, which is expected to be released soon after the release of Tru64 UNIX Version 5.0. Contact your Compaq representative to check on the current status of the DDK.

The rest of this chapter provides brief descriptions of the DDK documentation.

Writing Device Drivers

This manual provides information for systems engineers who write device drivers for hardware that runs the Tru64 UNIX operating system. Systems engineers can find information on device driver concepts, interfaces, and configurations, as well as summary information about routines, and data structures. (See the Section 9r, 9s, 9u, and 9v reference pages for complete descriptions of these concepts).

This manual also provides a glossary of device driver terms.

Prior to the Release 2 of the DDK, most of the information in this book had been included in the book titled *Writing Device Drivers: Tutorial*.

Writing EISA and ISA Bus Device Drivers

The Extended Industry Standard Architecture (EISA) bus is an industry-standard, high-performance bus; it is a superset of the 8–bit and 16–bit Industry Standard Architecture (ISA). Device driver developers can write drivers that support an ISA device that operates on the EISA bus, the ISA bus, or both buses.

This manual provides information for systems engineers who write device drivers for the EISA or ISA buses. Topics include the EISA/ISA bus architecture, kernel routines, and data structures that EISA/ISA bus device drivers use. Also provided is an example driver that operates on the EISA bus.

Writing Device Drivers for the SCSI/CAM Architecture Interfaces

The Small Computer System Interface (SCSI) is a set of evolving ANSI standard interfaces that allow computers to communicate with peripheral hardware such as disk drives, CD-ROM drives, printers, and scanners. The SCSI/CAM architecture is a reliable, maintainable, and high performance SCSI subsystem based on the industry-standard Common Access Method (CAM) architecture.

This manual provides an overview of the Tru64 UNIX SCSI/CAM architecture for systems engineers who write SCSI/CAM device drivers. Topics include descriptions of user agent routines, data structures, common and generic routines and macros, error handling, and debugging routines. Also provided is a sample generic CAM peripheral driver.

Writing Kernel Modules

Kernel modules are binary images containing code and data structures that run in the UNIX kernel. These modules provide a level of common code that improves the efficiency of a system by combining like tasks in a single area and eliminating redundant code.

This manual provides information for device drivers developers who would benefit from having an intermediate layer of code between the driver software and physical devices. It would also benefit third-party developers who want to augment the kernel with modules tailored to their particular environment.

This is the same manual that is included in the Programmer's Kit of the Tru64 UNIX documentation set. This book, which is new with Release 2 of the DDK, contains most of the information that had been included in the book called *Writing Device Driver: Advanced Topics*.

Writing PCI Bus Device Drivers

The Peripheral Component Interconnect (PCI) local bus is a physical interconnect mechanism that is used between a computer system's highly integrated peripheral controller components and the processor and memory systems.

This manual provides information for engineers who write drivers for devices that operation on the PCI bus in Tru64 UNIX operating system. Topics include information on the PCI bus architecture, kernel routines, and data structures that PCI device drivers use.

Writing TURBOchannel Device Drivers

The TURBOchannel bus is a synchronous, asymmetrical I/O channel that some Alpha CPUs support.

This manual provides information that systems engineers need to write device drivers that operate on the TURBOchannel bus; it does not emphasize any specific types of device drivers. Topics include information on the TURBOchannel kernel routines and data structures that TURBOchannel device drivers use. Also included is an example driver that operates on the TURBOchannel bus.

Although this manual is written for engineers who are familiar with driver concepts, an intimate understanding of the TURBOchannel hardware is not necessary to write TURBOchannel device drivers.

Writing VMEbus Device Drivers

The VMEbus is an industry-standard, high-performance bus that supports 8–bit, 16–bit, and 32–bit transfers over a nonmultiplexed 32–bit data bus. Like other buses, the VMEbus architecture defines a computer data path. Unlike other buses, however, the VMEbus is microprocessor independent, can be easily upgraded from 16–bit to 32–bit processors, and is suitable for a vendor to build compatible products.

This manual contains information systems engineers need to write device drivers that operate on the VMEbus. Topics include information on the VMEbus architecture, kernel routines, and data structures that VMEbus device drivers use. A VMEbus device driver example shows the use of these interfaces.

Writing Network Device Drivers

This manual introduces driver writers to topics that are specific to writing a device driver for a local area network (LAN) device.

Writing a Graphics Device Driver and DDX for the Tru64 UNIX X Server

Device-dependent X (DDX) is a shared library that adds server support for a graphics device. Each type of graphics device needs its own DDX.

This manual describes the components of the Tru64 UNIX graphics hardware architecture; it explains how to add support for a new graphics display and how to package the resulting product. This manual does not describe how to write generic X Window System device-specific code or generic UNIX device drivers.

Guide to Preparing Product Kits

This manual describes the procedures for creating, maintaining, and installing the collections of files and directories that make up a layered product kit. A product kit is the standard mechanism by which layered product modifications are delivered and maintained on a Tru64 UNIX operating system. Kits are distributed on CD–ROM, diskettes, or tape for installation on customer's systems.

This is the same manual that is included in the Programmer's Kit of the Tru64 UNIX documentation set.

Reference Pages, Section 9r, Device Drivers (Volume 1)

This book, which provides printed versions of the Section 9r reference pages, is new with Release 2 of the DDK. Information in this book had previously been included in the book called *Writing Device Drivers: Reference*.

Reference Pages, Section 9s, 9u, and 9v, Device Drivers (Volume 2)

This book, which provides printed versions of the Section 9s, 9u, and 9v reference pages, is new with Release 2 of the DDK. Information in this book had previously been included in the book called *Writing Device Drivers: Reference*.

Installation Instructions and Release Notes

This document provides instructions for installing the DDK components: online documentation, example files and the Driver Development Tool, and the X Developer's Kit. It also provides product release notes.

White Papers

The white papers in the Device Driver Kit are provided only on line. Release 2 contains the following papers:

- *Writing Device Drivers for the SCSI-3/CAM-3 Architecture Interfaces*
- *Writing PCMCIA Bus Device Drivers*
- *Managing Data for Kernel Subsystems*

6

How to Order Tru64 UNIX Documentation

You can order Tru64 UNIX documentation at the following Web site:

http://www.businesslink.digital.com/

If you need further help in deciding which documentation best meets your needs, you can call **800–344–4825** in the United States and Canada. In Puerto Rico, call **787–781–0505**. In other countries, contact your local Compaq subsidiary.

If you have access to Compaq's intranet, you can place an order at the following Web site:

http://asmorder.nqo.dec.com/

Table 6–1 provides the order number for the Tru64 UNIX Documentation CD–ROM and the kits containing the printed documentation. (The indentation indicates which kits are included in those above them. For example, the Tru64 UNIX Documentation Kit contains all of the kits listed below it.)

Table 6–1: Tru64 UNIX Documentation Kits and CD–ROM

Documentation Kit	Order Number
Tru64 UNIX Documentation CD–ROM	QA-6ADAA-G8
Reference Page Kit	QA-6ADAF-GZ
Tru64 UNIX Documentation Kit	QA-6ADAA-GZ
Developer's Documentation Kit	QA-6ADAG-GZ
End User Documentation Kit	QA-6ADAB-GZ
Startup Documentation Kit	QA-6ADAC-GZ
General User Documentation Kit	QA-6ADAD-GZ
System and Network Management Documentation Kit	QA-6ADAE-GZ

Glossary of Common UNIX and General Computer Terms

This glossary provides definitions for many of the terms you may see while using the Tru64 UNIX documentation. Although the majority of terms deal with the UNIX environment, you will also find other common terms you will encounter; for example, words related to the Internet.

You can also refer to the glossaries in the following books for definitions to terms used in those subject areas:

- *AdvFS Administration*
- *Command and Shell User's Guide*
- *Common Desktop Environment: Product Glossary*
- *Guide to DECthreads*
- *Guide to Preparing Product Kits*
- *Logical Storage Manager*
- *Performance Manager*
- *Network Programmer's Guide*
- *Programming Support Tools*
- *Security*
- *Sharing Software on a Local Area Network*
- *Software License Management*
- *System Configuration and Tuning*
- *Writing Kernel Modules*
- *Writing Software for the International Market*

Special Characters

/

See *root*

. (dot)

A shorthand expression representing the user's working directory.

See also *working directory*

.. (dot-dot)

A shorthand expression representing the immediate parent of the user's working directory.

A

absolute pathname

A pathname that begins at the root directory; a pathname that always begins with a slash (/). For example, `/usr/games` is an absolute pathname. Also called a full pathname.

See also *relative pathname*

active user

In an XTI transport connection, the transport user that initiated the connection.

See also *client process*, *passive user*, *XTI (X/Open Transport Interface)*

adb

A program designed to assist the user in debugging other programs under development.

Address Resolution Protocol

See *ARP (Address Resolution Protocol)*

alias

A name or symbol used in place of another name, symbol, or group of symbols; usually shorter or easier to use than what it represents. For example, if you often access a certain directory, you could set up an alias so that the word `work` would be an alias for "`cd /share/tomb/tools/tools/work`". Thereafter, typing `work` would put you in the `/share/tomb/tools/tools/work` directory. For more information see the `alias`(1) reference page.

append

1. To add data to the end of existing data.

2. In an editing environment, to attach a file to the end of another file.

application

A program or set of programs designed to perform a particular useful function or set of functions; for example, the Source Code Control System (SCCS) is an application for managing program source code.

apropos

A command that displays the reference page names and summary lines that contain a specified word or string of characters. The `apropos` command is the same as the `man -k` command.

See also *reference page*, *man*

archive

1. To store programs, data files, text files, and other types of files for safekeeping.
2. A repository for such files.

argc

A variable containing the number of arguments passed by the shell to a command.

See also *argument count*

argument count

The number of arguments passed by a command interpreter to a command, or from a routine in a program to a subroutine, procedure, or function.

argument list

The actual information (arguments) passed by a command interpreter to a command, or from a routine in a program to a subroutine, procedure, or function.

argv

An array, each of whose elements is one of the arguments passed by the shell to a command.

See also *argument list*

ARP (Address Resolution Protocol)

1. The Internet (TCP/IP) Protocol that can dynamically bind a high-level Internet address to a low-level, physical hardware address. ARP can be used only across a single physical network and in networks that support the hardware broadcast feature.
2. The Internet (TCP/IP) Protocol that dynamically maps between Internet addresses, Baseband Adapter addresses, and Token-Ring Adapter addresses on a local area network (LAN).

array
A collection of data elements (variables) identified by a common name and distinguished from one another by numbers representing their positions in the collection. The distinguishing numbers are called subscripts.

assignment statement
A statement that sets a value for a particular field or parameter. In program source files and scripts, assignment statements often have the form *parameter=value.*

asynchronous event
See *event*

asynchronous execution
1. The execution of processes or threads in which each process or thread does not await the completion of the others before starting.
2. In XTI, a mode of execution that notifies the transport user of an event without forcing it to wait.

Asynchronous Transfer Mode
See *ATM (Asynchronous Transfer Mode)*

ATM (Asynchronous Transfer Mode)
A 25 M/bps to 622 M/bps network standard that uses cell switching. It is connection oriented, providing switched, full-duplex communication circuits between nodes.

attribute-value pair
In the key file of a software product kit, a line specifying the name and value for a single attribute of the kit. Controls how the kit is built by the `kits` command and how it is installed by the `setld` utility.

awk
The command for executing programs written in the `awk` programming language. An `awk` program is a sequence of patterns and corresponding actions that are carried out when a pattern is read. The `awk` utility is a more powerful tool for pattern matching and text manipulation than either `grep` or `sed`.

See also *grep*, *sed*

B

background job
See *background process*

background process
A job that runs without interfering with normal command-line entries. A process runs in the background when the command to begin the process is entered with an ampersand (&) character following it. For example, to run the X Window System clock program in background, a user would enter the command `xclock &`. As a result, the clock would be invoked in one window, while the command line on which the `xclock` command was entered would be ready to accept new commands.

See also *foreground process*

Berkeley Internet Name Domain
See *BIND (Berkeley Internet Name Domain)*

Berkeley Software Distribution
See *BSD (Berkeley Software Distribution)*

Berkeley UNIX
See *BSD (Berkeley Software Distribution)*

/bin directory
A directory that contains executable programs and scripts. For example, the `/usr/bin` directory contains programs that nonprivileged users can run, and the `/sbin` directory contains programs that only privileged users can run.

See also *binary file*, *path*, *script*

binary
1. Referring to the number 2 or the system of binary numeration.
2. Referring to an executable file created by a compilation process.
3. Referring to a situation that can assume one of two possible states.

binary file
A file created by a compilation process. Binary files contain codes that are not part of the ASCII character set and utilize all 256 possible byte values.

binary operator
1. A symbol that represents an operation to be performed on two arrays, data items, or expressions. The four types of binary operators are character, logical, numeric, and relational.
2. An arithmetic operator that has two terms.

BIND (Berkeley Internet Name Domain)
A name service available on internet networks.

bit bucket
A term for any receptacle into which data is placed without the possibility of retrieval. It is often used to refer to the null device `/dev/null`.

block device
A data storage or transfer device that manipulates data in groups of a fixed size; for example, a disk, whose data storage size is usually 512 bytes.

See also *character device*

block device switch table
The method used by the Tru64 UNIX operating system to select the entry points associated with a particular block device.

See also *character device switch table*

blocking mode
See *synchronous execution*

block special file
A device special file that provides access to an input or output device and is capable of supporting a file system.

See also *device special file*

BOM
A sequence of characters written on a magnetic tape to signify the beginning of medium.

See also *EOF (end of file)*, *file mark*

Boolean
1. An algebra (named for George Boole) that is similar in form to ordinary algebra, but in which the values of the variables are restricted to the two possible values true and false. The logic of Boolean algebra works well with the binary logic of computers, where values are represented by the digits 0 and 1.
2. A term sometimes used to refer to Boolean operators, including AND, OR, NOT, EXCEPT, IF, THEN, TRUE, and FALSE.

Bourne shell
The command interpreter and interpreted programming language originally developed by Steve Bourne.

See also *shell*

breakpoint
A place in a source code program that stops the debugger during program execution. Breakpoints aid in the testing and debugging of programs.

See also *tracepoint*

break statement

In a programming language, a statement that causes the program to exit immediately from the current control structure (such as a case statement or a for loop). A break statement is often used to terminate execution of a loop before the programmed number of iterations has been performed.

BSD (Berkeley Software Distribution)

The UNIX software release of the Computer System Research Group of the University of California at Berkeley — the basis for some features of the Tru64 UNIX version of the UNIX system.

BSD socket interface

A transport-layer interface provided for applications to perform interprocess communication between two unrelated processes on a single system or on multiple connected systems. This interprocess communications facility allows programs to use sockets for communications between other programs, protocols, and devices.

built-in

A command that is built into a shell, as opposed to a command that stands alone as a separate executable file and is invoked by a shell.

C

c89

A command that invokes the C compiler and whose use is recommended for portability among systems that conform to the X/Open UNIX CAE specification for commands and utilities.

See also *cc*, *compiler*

call

In a programming language, a statement that invokes a subroutine, function, or procedure.

call by reference

In a programming language, a method of passing an argument to a subroutine, a function, or a procedure by supplying the address of the data rather than its actual value.

See also *call by value*

call by value

In a programming language, a method of passing an argument to a subroutine, a function, or a procedure by supplying the actual value of the data.

See also *call by reference*

CAM (Common Access Method)

The ANSI standard that defines the software interface between device drivers and the Host Bus Adapters, as well as other means by which SCSI peripherals are attached to a host processor.

See also *SCSI (Small Computer System Interface)*

CAM Control Block

See *CCB (CAM Control Block)*

carriage return

A character that forces all following text to the left margin of the next line or that signals the end of user input. The Return key is usually used to produce a carriage return.

case insensitive

Unable to distinguish between uppercase and lowercase letters. A case-insensitive device or program considers `A` and `a` to be the same character.

See also *case sensitive*

case sensitive

Able to distinguish between uppercase and lowercase letters. A case sensitive device or program considers `A` and `a` to be different characters. Devices and programs that are part of the Tru64 UNIX operating system are case sensitive.

See also *case insensitive*

case statement

In a programming language, a control structure that can take any of several possible paths depending on the evaluation of its argument.

cbreak mode

A terminal driver operation mode that allows processes to read input as it is being typed. This mode eliminates the character, mode, and line editing input facilities.

cc

A command commonly used to invoke the C compiler on UNIX systems.

See also *c89*, *compiler*

CCB (CAM Control Block)

The data structure provided by SCSI peripheral drivers to the XPT transport level to control the execution of a function by the SCSI Interface Module (SIM).

CDB (Command Description Block)
A data structure that contains the SCSI operation code, parameters, and control bits for a specific operation.

CDE (Common Desktop Environment)
A graphical user interface for interacting with the Tru64 UNIX operating system. The CDE interface was jointly developed and is based on industry standards, including the X Consortium's X Window System and the Open Software Foundation's Motif interface.

character device
A data storage or transfer device that manipulates data in increments of a single character; for example, a terminal.

See also *block device*

character device switch table
The method used by the Tru64 UNIX operating system to select the entry points associated with a particular character device.

See also *block device switch table*

character special file
A file through which processes can access either a character-stream oriented I/O interface or an unstructured (raw) device, such as a communication line or an unbuffered magnetic tape or disk.

child process
See *parent process*

client
A computer system that uses resources provided by another computer system called a server.

client process
In the client/server model of communication, a process that requests services from a server process.

clist
A data structure used by a BSD-type of terminal driver to store data coming from, or going to, terminals.

See also *STREAMS*

Command Description Block
See *CDB (Command Description Block)*

command history
See *history list*

command mode

A state of a system or device in which the user can enter commands.

command substitution

The ability to capture the output of any command as an argument to another command by placing that command line within grave accents (` `). The shell first executes the command or commands enclosed within the grave accents and then replaces the whole expression, including grave accents, with their output. This feature is often used in assignment statements.

comment out

To selectively disable interpretation of a portion of a program or document source file.

Common Access Method

See *CAM (Common Access Method)*

Common Desktop Environment

See *CDE (Common Desktop Environment)*

common internet address notation

On internet networks, the decimal for the 32-bit internet address. Also called dotted-decimal notation.

communication domain

An abstraction used by the interprocess communication facility of a system to define the properties of a network. Properties include a set of communication protocols, rules for manipulating and interpreting names, and the ability to transmit access rights.

compile

To process one or more program source files in order to produce an executable binary file or an object file.

compiler

A program that translates programs written in a particular source language into executable binary files (or into intermediate binary files referred to as object files). The input can include one or more source-language files together with one or more object files. Compiled programs execute faster than interpreted programs because the compiler has already performed the interpretation. The `cc` program is a C compiler.

See also *binary file*, *compile*, *interpreter*, *object file*

compile time

Refers to actions that are taken by a compiler during the compilation of a program.

See also *run time*

computer virus

See *virus*

computer worm

See *worm*

concatenate

To place together. Data elements such as strings can be concatenated to produce a string that contains all of the characters of the first original string, followed by the characters of the next original string, and so on. Files can be concatenated by combining their contents in a similar manner, either into a new file or into one of the original files.

conditional compilation

During the compilation of a program, a portion of the process (code block) that is enabled or disabled by a variable or condition external to the code block under consideration. For example, a certain program might contain a block that is to be compiled only if the compilation is performed on a Tru64 UNIX system.

conditional execution

During the execution of a program, a portion of the program's behavior or output that is enabled or disabled by a variable or condition. For example, a certain program might contain code that asks the user questions only if the user initiates the program to run in a menu mode.

conditional statement

In a programming language, a statement (for example, the if statement) that evaluates one or more variables or conditions and uses the result to choose one of several possible paths through the subsequent code.

configuration

1. The machines, devices, and programs that make up a data processing system or network.

2. The act of making a subsystem, or a set of subsystems, available for use by a running operating system.

3. The set of configured subsystems in an operating system.

configuration file

A file that specifies the characteristics of a system or subsystem.

connectionless mode

A mode of service supported by a transport endpoint that requires no established connection for transmitting data. Data is delivered in self-contained units, called **datagrams**.

connection-oriented mode
A mode of service supported by a transport endpoint for transmitting data over an established connection.

construct
A data structure used for a particular purpose.

context search
See *global search*

control statement
In a programming language, a statement that can cause different actions to ensue, depending on the results of an evaluation or test.

cooked mode
The condition of a device driver in which the driver interprets the data passing through it. For example, a UNIX terminal driver operating in cooked mode translates a Return character from the terminal into a Line Feed character to be passed to the system.

See also *raw mode*

cron
A daemon that executes commands at specified times and dates, according to instructions in the `crontab` file.

See also *daemon*

crontab file
A file that specifies the dates and times at which specified commands are to be executed. The `cron` daemon examines the `crontab` file at specified intervals, and executes the indicated commands at the specified dates and times.

csh
The command that invokes the C shell.

See also *C shell*, *shell*

C shell
A command interpreter and interpreted programming language developed at the University of California at Berkeley; so named because many of its constructs resemble the equivalent C language constructs.

See also *shell*

current directory
See *working directory*

cursor
For video display screens, a symbol that shows the location of keyboard input. The cursor shows the position at which the next character to be displayed will be placed.

See also *pointer*

cursor movement keys
A set of keys, usually labeled with arrows pointing up, down, left, and right, that position the cursor on a video display screen.

D

daemon
A process that performs a system management function that is transparent to the user. A daemon can perform its task automatically or periodically. For example, the `cron` daemon periodically performs the tasks listed in the `crontab` file. Daemons can be generated by the system and by applications. Some daemons can also be started manually; for example, the `binlogd` command starts a daemon that logs binary event records to specified files. The commands that manually start daemons usually end with a `d`.

data communications
The transmission of information between computers by means of a network such as an Ethernet, a telephone system, or a satellite link.

datagram
A unit of data that is transmitted across a network by the connectionless service of a transport provider. In addition to user data, a datagram includes the information needed for its delivery. It is self-contained, in that it has no relationship to any datagrams previously or subsequently transmitted.

datagram socket
A socket that provides datagrams consisting of individual messages for transmission in connectionless mode.

Dataless Management Services
See *DMS (Dataless Management Services)*

dbxd
The command that invokes the `dbx` program, which is used by developers to help debug other programs under development.

DCE (Distributed Computing Environment)
A de facto standard for distributed computing that defines a uniform set of services that share certain global properties for common naming, security,

time synchronization, system availability, access to data, and system management. DCE enables applications and data on heterogeneous systems to work together.

delta

In an RCS or SCCS file, the set of changes that constitute a specific version of the file.

dependency file

See *dependent*

dependency subset

The condition in which a subset may or may not require the presence of other subsets in order to function properly. Evaluated by a subset's software control program (SCP) under control of the `setld` utility.

See also *SCP (Subset Control Program)*, *subset*

dependent

Also called a **dependency file**. In the `make` utility, an entity on which a file to be built (the target) depends. A source file is a dependent of an object module.

detached job

A job that continues processing after the user has logged out.

device driver

The software that controls a peripheral device such as a disk or a printer.

device special file

A file used by processes to access hardware devices. For example, a printer is accessed through a device special file.

See also *block special file*

DFS (Distributed File System)

A distributed DCE application that provides a unified, globally distributed file system. Under this file system, a DFS file is accessible from any DCE DFS machine using the same name, regardless of the server currently storing the file.

DHCP (Dynamic Host Configuration Protocol)

An Internet (TCP/IP) Protocol that enables the automatic assignment of Internet addresses to clients on the network from a pool of reusable addresses. Address assignment occurs automatically whenever client systems such as portable computers are attached to the network.

directory
A type of file containing the names and controlling information for other files or other directories.

directory hierarchy
The arrangement of directories in a file system. The root directory is at the top of the directory hierarchy and contains pointers to all file systems and all directories on the system.

directory stack
A data structure that stores directories for later recall.

disk label
The disk information, usually located in sector 0 (zero), that includes the disk geometry and partition divisions. This information is used by the system disk driver and the boot program to identify a drive, and to determine how to program a drive and where to find the file systems.

See also *geometry*, *partition*

disk partition
See *partition*

Distributed Computing Environment
See *DCE (Distributed Computing Environment)*

Distributed File System
See *DFS (Distributed File System)*

DMS (Dataless Management Services)
A service provided by Compaqwhereby a server computer system maintains the `root`, `/usr`, and `/var` file systems for client computer systems connected to the server via a local area network (LAN).

See also *LAN (Local Area Network)*

DMS area
A reserved disk area that is physically connected to a DMS server and that contains multiple copies of the DMS root area, one for each DMS client.

DMS client
A computer system whose system disk area is physically connected to a DMS server rather than to the client itself and is accessed across the network by the client.

domain
See *domain name system*

domain name system

A tree-structured system for organizing hosts names for an entire internet.

See also *communication domain*, *Internet domain name system*

down time

The period during which a machine is unavailable for use.

See also *up time*

Dynamic Host Configuration Protocol

See *DHCP (Dynamic Host Configuration Protocol)*

E

ed editor

A line-oriented program for modifying the contents of text files. The program operates by accepting commands from the user; for example, issuing the command `s/Unix/UNIX/g` would cause the editor to replace each instance of the string "Unix" on the current line with "UNIX."

editor

A program for modifying the contents of text files. Full-screen editors, such as `vi`, use video display terminals to display several lines of the file being manipulated; they allow the user to move the cursor to a specific location and change the text there. Line editors, such as `ed`, work on a line-by-line basis. Stream editors, such as `sed`, work by applying commands from a previously prepared list (called a script) instead of by accepting commands from the user.

effective root directory

The point where a system starts when searching for a file. Its pathname begins with a slash (/).

effective user ID

The current user ID, but not necessarily the user's ID. For example, a user logged in under a login ID may change to another user's ID. The ID to which the user changes becomes the effective user ID until the user switches back to the original login ID.

EGP (External Gateway Protocol)

A type of routing protocol that allows individual networks to communicate with the Internet backbone.

See also *Internet*

Emacs

A text editor developed by the Free Software Foundation that is available for all UNIX systems, although it is not a standard part of Berkeley UNIX or System V. It is included with the Tru64 UNIX operating system.

environment

The set of conditions under which a user is working on the computer. The environment includes such information as the name of the working directory, the name of the command interpreter, the identity of the user's terminal, and so on.

environment variable

A symbol containing information that can be used by shells or commands. Environment variables are available to all processes in a given process group; they are propagated by the creation of a child process.

See also *process variable*

EOF (end of file)

1. A condition indicating that the end of a data file has been reached by a program reading the file.

2. A specific sequence of characters written on a magnetic tape.

See also *BOM*, *file mark*

equivalence class

A grouping of characters or character strings that are considered equal for purposes of collation. For example, many languages place an uppercase character in the same equivalence class as its lowercase form, but some languages distinguish between accented and unaccented character forms for the purpose of collation.

error

Any condition in which the expected results of an operation are not achieved. In XTI, an indicator that is returned by a function when it encounters a system or library error in the process of executing. The object is to allow applications to take an action based on the returned error code.

escape

1. To protect a character from interpretation by a program by preceding it with a backslash (\).

2. An ASCII character that is usually interpreted as a command to cease a certain activity or as the initial character of a sequence that performs a special function. Cursor control sequences for many terminals and workstations use the escape character.

See also *quote*

/etc

A catchall directory, which usually contains miscellaneous system data files (such as `termcap`, the terminal capabilities database).

Ethernet

A communications concept for local communication networks that interconnects different kinds of computers, information processing products, and office equipment. It is a 10-megabit-per-second baseband local area network (LAN) using carrier sense multiple access with collision detection (CSMA/CD). The network allows multiple stations to access the medium at will without prior coordination, and avoids contention by using carrier sense and deference, and detection and transmission.

event

An occurrence, or happening, that is significant to a transport user. Events are asynchronous, in that they do not happen as a result of an action taken by the user.

event executable image

An executable image located in physical memory.

event management

A mechanism by which transport providers notify transport users of the occurrence of significant events.

executable file

A data file created by a compiler that contains program information a computer can read, interpret, and execute. Also called an image or a binary file.

ex editor

A line-oriented program for modifying the contents of text files. The `ex` editor is an extended version of the `ed` editor.

expedited data

Data that is considered urgent. The semantics of this data are defined by the transport provider.

expression

1. A representation of a value; for example, variables and constants appearing alone or in combination with operators.

2. In programming languages, a language construct for computing a value from one or more operands, such as literals, identifiers, array references, and function calls.

3. A configuration of signs.

extended character

A character other than a 7-bit ASCII character. An extended character can be a 1-byte code point with the eighth bit set (ordinal 128-255).

External Gateway Protocol

See *EGP (External Gateway Protocol)*

F

field

1. The basic unit of information in a record.

2. In awk, one element of an input record.

See also *record*

field separator

One or more characters used to separate fields in a record.

file descriptor

A small unsigned integer that a UNIX system uses to identify a file. A file descriptor is created by a process through issuing an open system call for the file name. A file descriptor ceases to exist when it is no longer held by any process.

file mark

A sequence of characters written on a magnetic tape to signify the end of a data file.

See also *BOM, EOF (end of file)*

file name expansion

See *globbing*

file pointer

An identifier that indicates a structure containing the file name.

file system

The collection of files and file management structures on a physical or logical mass storage device.

filter

1. A command that reads standard input data, modifies the data, and sends it to standard output.

2. A device or program that separates data, signals, or materials in accordance with specific criteria.

flag
See *option*

foreground job
See *foreground process*

foreground process
A job that must be completed or interrupted before the shell will accept more commands; a job receiving input from a workstation or terminal.

See also *background process*

fork
1. The command used to create and start a child process.
2. The result of using the `fork` command.

See also *parent process*

full pathname
See *absolute pathname*

full-screen editor
An editor that displays an entire screen at a time. Also called a visual editor.

See also *line editor*

G

geometry
The sizes (in bytes) of cylinders, tracks, and sectors for a particular disk device.

See also *disk label*

gid, GID
See *group ID*

global
In programming languages, pertaining to information defined in one subdivision of a program and used in at least one other subdivision of the program; pertaining to information available to more than one program or subroutine.

global character
See *wildcard character*

global search

In an editing environment, the process of having the system look through a document for specific characters, words, or groups of characters.

globbing

A UNIX term for the shell's process of wildcard file name expansion to develop a list of literal file names that the shell then passes to a command. The C shell permits the user to disable globbing by default; the Bourne, Korn, and POSIX shells require the user to quote or escape metacharacters in file names if globbing is not desired.

grep

The command that invokes the `grep` program, which is used to search specified files for lines containing characters that match specified patterns, and then writes those matching lines to standard output. The name means Global Regular Expression Printer.

See also *regular expression*

group

1. A collection of users who can share access authorities for protected resources.

2. A list of names that are known together by a single name.

3. A set of related records that have the same value for a particular field in all records.

4. A series of records logically joined together.

See also *login group*

group ID

A unique number assigned to a group of related users. The group number can often be substituted in commands that take a group name as an argument.

H

hard link

1. A mechanism that allows the `ln` command to assign more than one name to a file. Both the new name and the file being linked must be in the same file system.

2. The default result of using the `ln` command.

See also *symbolic link*

hashed passwd database

An indexed database containing the contents of the `passwd` file. The indexed database minimizes the search time needed to retrieve information.

hashing
A method of transforming a search key into an address for the purpose of storing and retrieving items of data.

HBA (Host Bus Adapter)
The hardware and microcode that provides the interface between system memory and a Small Computer System Interface (SCSI) bus.

head
A command that displays a user-specifiable number of lines from the beginning of a text file.

See also *tail*

header file
See *include file*

hidden character
A character in the ASCII character set that is not printable; for example, the DEL and ESC characters.

hidden file
A file whose name begins with a period. by default, the `ls` command omits such files from its listings.

history
In the C shell and the Korn shell, a command that displays the user's history list.

history list
In the C shell and the Korn shell, a listing of the most recent commands entered by the user. Commands in the history list are available for recall, modification, and reexecution.

$HOME
An environment variable containing the absolute pathname of the user's home directory.

See also *$home, environment variable*

$home
A process variable containing the absolute pathname of the user's home directory.

See also *$HOME, process variable*

home directory
A directory that is owned by a specific user and from which that user's other directories descend in a hierarchy. Also known as a **login directory.**

See also *working directory*

host

1. The primary or controlling computer in a communications network.
2. A computer attached to a network.

Host Bus Adapter

See *HBA (Host Bus Adapter)*

host name

The name given to a computer on the network.

HTML (HyperText Markup Language)

The coding (markup) inserted in a file intended for display on a World Wide Web browser that tells the browser how to display a Web page's words. The markup is done with *tags*, which are command words enclosed in angle brackets. For example, the tag **<P>** creates a new paragraph; the tag **<TABLE>** begins the formatting of a table. Although the World Wide Web Consortium (W3C) promotes the standardization of HTML, both Netscape and Microsoft browsers currently implement some features differently and provide nonstandard extensions.

HyperText Markup Language

See *HTML (HyperText Markup Language)*

I

ICMP (Internet Control Message Protocol)

A host-to-host protocol from the Internet Protocol suite that controls errors and the operations of the Internet Protocol (IP).

See also *IP (Internet Protocol)*

#include

A C language precompiler directive specifying interpolation of a named file into the file being compiled. The interpolated file is a standard header file (indicated by placing its name in angle brackets) or any other file containing C language code (indicated by placing its name in double quotation marks). For example:

```
#include <header_file.h>
#include "myfile.c"
```

The absolute pathname of header files whose names are placed in angle brackets (<>) is `/usr/include/`*`file.h`*.

See also *include file*

include file
A text file that contains declarations used by a group of functions, programs, or users. Also known as a header file.

See also *#include*

incremental backup
The process of copying files that have been opened for reasons other than read-only access since the last backup was created and that meet the backup frequency criteria.

infinite loop
A source code error that causes the program to continually repeat the same set of instructions. For example, Instruction A sends the program execution to Instruction B, which in turn sends the program execution back to instruction A. Such a loop can only be interrupted by intervention from outside the program.

init
The command given by a UNIX system as the final step in the boot procedure.

init process
A process created by the system that performs system administration tasks, such as spawning login processes and handling the orderly shutdown from multiuser to single-user mode.

inline editing
A feature of some shells that allows users to edit a current or previously entered command line.

inode
The internal structure that describes the individual files in the operating system. There is one inode for each file. An inode contains the node, type, owner, and location of a file. A table of inodes is stored near the beginning of a file system.

inode number
A number specifying a particular inode file in the file system.

input
Data to be processed.

input redirection
The specification of an input source other than standard input.

instruction

The part of a computer program that tells the computer what function to perform at that stage.

International Standards Organization

See *ISO (International Standards Organization)*

Internet

1. A collection of computing networks consisting of participants from major research institutions, universities, and government labs, including the National Science Foundation (NSF) and the NFSnet regional organizations. The Internet is not a commercial product, but rather a large project in support of research.

2. **internet** A collection of connected networks using the Internet Protocol (IP).

internet address

A unique 32-bit number that identifies a host's connection to an internet network. An internet address consists of a network number and a host number.

Internet Control Message Protocol

See *ICMP (Internet Control Message Protocol)*

Internet domain name system

The domain name system of the Internet, which consists of the following categories of hosts: COM, EDU, GOV, MIL, NET, ORG, and ARPA.

See also *communication domain*, *domain name system*, *Internet*

Internet Protocol

See *IP (Internet Protocol)*

interpreter

A program that interprets and executes programs written in a particular source language. Interpreted programs execute more slowly than compiled programs because the interpreter is performing two operations at once. The UNIX shells are interpreters.

See also *compiler*

interrupt

1. An event that causes a computer to digress from its normal processing stream in order to respond to the condition that triggered the digression. Upon completion of the digression, the normal processing stream is resumed at the point of interruption. Interrupts can be caused either by software instructions or by hardware events such as the completion of an I/O operation.

2. To trigger an interrupt.

interrupt handler

Code in a program or operating system that performs actions in response to an interrupt.

IP (Internet Protocol)

The network layer protocol for the Internet protocol suite that provides the basis for the connectionless, best-effort packet delivery service. IP includes the Internet Control Message Protocol (ICMP) as an integral part. The Internet protocol suite is referred to as TCP/IP because IP is one of the two most fundamental protocols.

IP gateway

See *IP router*

IP router

A host that connects two or more internet networks. The IP router knows how to reach all the hosts on the networks to which it is attached. Also known as an IP gateway.

ISO (International Standards Organization)

An international body composed of the national standards organizations of 89 countries. ISO issues standards on a vast number of goods and services, including networking software.

iterate

To perform the same function repeatedly on different data, often with the object of arriving at a result by successively closer approximation.

J

job

1. A unit of work defined by a user to be done by a system. The term *job* sometimes refers to a representation of the job, such as a set of programs, files, and control statements to the operating system.

2. One or more related procedures or programs grouped into a procedure, identified by appropriate job control statements.

job control

Facilities for monitoring and accessing background processes.

job number

A number assigned to a job as it enters the system to distinguish the job from other jobs.

job queue
A list of the jobs that are waiting to be processed by the system.

job state
The status of the work being done by a system.

K

kdbx
The command that invokes the `kdbx` program, an interactive crash analysis and kernel debugging tool. The `kdbx` program serves as a front end to the `dbx` debugger.

kdebug program
A program that lets programmers control the execution of a running kernel.

kernel
The integral part of the operating system that controls processes, system scheduling, memory management, input and output services, device management, network communications, and the organization of the file systems.

See also *shell*

keyword
1. A word that must be matched when retrieving information.
2. A reserved word whose presence is required in a file.

kill
1. To stop the operation of a process. In most cases, a user can kill a foreground process by pressing Ctrl/c.
2. The Tru64 UNIX command that a user can issue to stop a background or suspended process. A superuser can use this command to stop any process on the system.

Korn shell
A command interpreter and interpreted programming language developed by David Korn. The Korn shell (`ksh`) is semantically an extended version of the Bourne shell, with constructs and commands to implement enhanced features, including job control and command history recall. The POSIX shell is a superset of the Korn shell.

See also *shell*

ksh
The command that invokes the Korn shell; the name of the executable file that is the shell.

See also *Korn shell, shell*

L

label
See *disk label*

LAN (Local Area Network)
A device communications system that operates over a limited physical distance, offering high-speed communications channels optimized for connecting information-processing equipment.

LAT (Local Area Transport)
A protocol that supports communications between host computer systems and terminal servers with terminals, PCs, printers, modems, and other devices over LANs.

See also *LAN (Local Area Network)*

layered product
An optional software product designed to be installed as an added feature of the Tru64 UNIX system.

lex
The command that invokes the Lexical Analyzer Generator, a program for generating other programs that can organize input into units of meaning (symbols) called lexemes.

See also *lexical analyzer*, *parser*, *yacc (Yet Another Compiler-Compiler)*

lexical analyzer
A program or program fragment for analyzing input and assigning elements of it to categories to assist in parsing the input. The `lex` program assists in the creation of lexical analyzers.

See also *parser*

Lexical Analyzer Generator
See *lex*

line editor
An interactive or noninteractive text editor that works on one line of text at a time.

See also *full-screen editor*

link
A directory entry referring to a file.

See also *hard link*, *symbolic link*

linking loader

A single program that loads, relocates, and links compiled and assembled programs, routines, and subroutines to create an executable file. Also known as link loader and linker loader.

lint

A program that checks C code for bugs, portability problems, and errors, such as mismatched argument types and uninitialized variables.

literal

1. A value expression representing a constant.

2. A specific symbol that cannot be modified during the translation of a program.

local area network

See *LAN (Local Area Network)*

local area transport

See *LAT (Local Area Transport)*

local host

The computer system to which a user's terminal is directly connected.

lock file

A file that indicates that operations on one or more other files are restricted or prohibited. The presence of the lock file can be used as the indication, or the lock file can contain information describing the nature of the restrictions. For example, the Tru64 UNIX `setld` utility creates a lock file for each product kit subset that it installs. If a given product includes subsets that require the presence of a previously installed subset, `setld` places in the earlier subset's lock file the names of the later subsets to prevent inadvertent deletion of the earlier subset.

locking

1. In software installation by the `setld` utility, the act of inserting a new subset's name in the lock file of an existing subset so that an attempt to remove the latter subset will flag the user with a dependency warning.

2. In a version control system, the creation and use of information flagging a version control file as being checked out for editing.

locking mechanism

In a version control system, a way to prevent overlapping and concurrent changes to a file. SCCS uses p-files to indicate which files are currently out for editing; RCS creates locks by editing the RCS file to insert lock information.

log in

To begin using a computer system, usually by entering one's login name and a secret password; to gain access to and communicate with the operating system as an authorized user.

login directory

See *home directory*

login group

The primary classification that establishes the access permission for the files created by the user.

See also *group*

login name

The name that identifies a user to a computer system and to other users of the system. When logging into the system, the user enters this name and (usually) a secret password. Also known as user name.

login shell

The shell that a user uses by default upon logging into the system. It is specified by the user's entry in the `passwd` file.

log out, log off

To stop using a computer system, usually by entering a command that tells the operating system that the user is ending the current session.

loop

1. A sequence of instructions that is executed repeatedly until a specified condition is satisfied.

2. In the UNIX virtual memory system, the page clusters in main memory that are repeatedly scanned for replacement.

See also *infinite loop*

M

macro

A shortened form of macro instruction.

macro instruction

An instruction written as part of a source language, which when compiled into machine code will generate several machine code instructions.

See also *instruction*

mail
A system that allows the exchange of written messages with other users. Also known as e-mail (for electronic mail).

mailbox
A file that contains new and unread mail messages. The mailbox file is usually in the `/usr/spool/mail` directory.

make
A tool that builds programs and applications by testing to see whether the source files that produce a given application are newer than the target files produced from them. If any source or intermediate file is newer than its target, `make` performs the actions necessary to rebuild the target file by following a set of rules. The rules can be standard (specified by default) or they can be explicit descriptions of the steps required.

MAKDEV
A script that creates device special files for the devices on a Tru64 UNIX system. This script resides in the `/dev` directory.

makefile
The specification file used by the `make` tool. The makefile specifies the names of target programs and describes rules for their creation.

See also *make*

man
The command that displays reference pages on line; the name is a short form of *manual*.

See also *apropos*, *reference page*

manpage, manual page
See *reference page*

MANPATH
An environment variable whose value provides the default directory search path use buy the `man`, `catman`, and `xman` commands.

See also *search path*

-man
An option for the `roff` family of text formatters. It specifies that the formatter is to use the `man` formatting macro package.

-me
An option for the `roff` family of text formatters. It specifies that the formatter is to use the `me` formatting macro package.

metacharacter
A character that is interpreted by a computer system to mean something other than its obvious meaning. For example, the asterisk is often used to allow wildcard matching in file names.

mode
The set of permissions for a file.

Motif
See *OSF/Motif*

mount
A command used to make a file system available.

See also *unmount*

mount point
A directory file that is the name of a mounted file system.

multiprocessor
A system with two or more processors sharing common physical memory.

N

name service
The service provided to client processes for identifying peer processes for communications purposes.

native software
Software that is written in a language that compiles either to assembly language or directly to the computer's standard machine representation (object files). Native software is more efficient and runs much faster than translated or interpreted software; in addition, it can be tailored to make the most effective use of the machine's resources.

neqn
The command for invoking the `neqn` program, which is used with the `nroff` program to format mathematical expressions.

See also *nroff*

network
Two or more computing systems that are linked for the purpose of exchanging information and sharing resources.

Network File System
See *NFS (Network File System)*

NFS (Network File System)

A service that allows a system (the server) to make file systems available across a network for mounting on other systems (clients). When a client mounts an NFS file system, the client's users see the file system as if it were local to the client.

NFS-mounted

Refers to a file system that is mounted over a network via NFS rather than being physically connected (local) to the system on which it is mounted.

See also *NFS (Network File System)*

nonblocking mode

See *asynchronous execution*

nroff

The command that calls the `nroff` program, a member of the *roff* family of text formatters. The `nroff` program produces ASCII output suitable for display or printing on character-cell devices such as terminals and printers.

O

object file

A nonexecutable intermediate binary file created by a compiler. Object files are frequently used as libraries, to provide precompiled program elements for use in compiling a complete executable binary.

See also *binary file*, *compiler*

octal

A number system that uses 8 as a base (radix). The octal system uses the digits 0 through 7, and each digit position represents a power of 8.

open system

A system that supports the International Organization for Standardization (ISO) Reference Model for Open System Interconnection (OSI).

Open Systems Interconnection

See also *OSI (Open Systems Interconnection)*

operator

In regular expressions, a character that is interpreted to mean something other than its literal meaning. For example, a pair of brackets (`[]`) form an operator that enables a single-character match on any one of the characters enclosed by the brackets.

optimization

The process of selecting the specific method by which a program is to perform a given task such that the most effective use is made of time, I/O, or other resources.

option

1. An argument that controls how the shell executes a command. Options are usually preceded by a hyphen and appear with the command name on a command line; for example, `ls -a`. An option is often referred to as a flag

2. An indicator or parameter that shows the setting of a switch.

3. A character that signals the occurrence of some condition, such as the end of a word.

4. An internal indicator that describes a condition to the CPU.

OSF (Open Software Foundation)

A consortium of software vendors formed for the purpose of developing and marketing widely compatible UNIX systems based on a common set of features.

OSF/Motif

A graphical user interface developed and licensed by the Open Software Foundation, Inc. OSF/Motif is based on the X Window System. Also called *Motif*.

OSI (Open Systems Interconnection)

A set of international standards developed by the International Organization for Standardization. The goal of the OSI is that different vendors' computer systems can interconnect.

owner

Usually, the user who creates a file. The owner has the right to change the list of users or groups who are permitted access to the file and the ways in which those users or groups may access the file. Ownership of a file can be reassigned by the system manager or superuser.

P

package

For the Tru64 UNIX operating system loader, a collection of object entities that share a common name space. Symbol names are unique within a package. Symbols from different packages may bear identical symbol names because they are distinguished by their package names.

page

A fixed-size unit of physical memory.

PALcode (Privileged Architecture Library)

A set of subroutines that are specific to a particular Alpha operating system implementation. These subroutines provide operating-system primitives for context switching interrupts, exceptions, and memory management.

parent directory

The directory in which another directory resides. The directory that is contained in the parent is called a subdirectory.

parent process

A process that has created other processes, called its children. In the UNIX system, every command that is not a shell built-in command creates a child process.

See also *fork*

parser

A program or program fragment for interpreting input and determining how to act upon it. The `yacc` program assists in the creation of parsers.

See also *lexical analyzer*

parsing order

The sequence in which a program interprets information that is input to it. For example, a program using left-to-right parsing order interprets input reading "create a number; write the number" so that the number created by the first step is written. A program with right-to-left parsing order interprets the same input to mean that the program is to write a number that it created in some previous step and then to create a new number.

partition

A physical portion of a disk. Disks are divided into partitions that are then assigned to hold various file systems. For example, the root file system is usually on the first partition, named `a`. The `/usr` file system is on a different partition, often the `g` partition. The use of partitions provides flexibility and control of disk usage, but it is restricted in that it denies unlimited use of all the available space on a given disk for a given file.

passive user

In an XTI transport connection, the transport user that did not initiate the connection.

See also *client process, active user, XTI (X/Open Transport Interface)*

passwd

1. The command by which users change their login password.

2. The UNIX file in which user passwords and associated information are stored; the file's pathname is `/etc/passwd`.

$PATH
An environment variable containing the user's search path for commands. Directory names in the `$PATH` variable are separated with colons.

See also *$path*

$path
A process variable containing the user's search path for commands. Directory names in the `$path` variable are separated with spaces.

See also *$PATH*

path
An ordered list of the directories in which the shell searches for the executable files named by commands that are not entered with a pathname and are not shell built-in commands.

See also *$PATH*, *$path*

pathname
The name of a file, concatenated onto a list of the directories through which access to that file is achieved; hence, the complete name of the file. Absolute pathnames begin at the root directory and are written with an initial slash (for example, `/usr/users/rolf/myfile.txt`). Relative pathnames begin at the user's working directory and are written without the initial slash (for example, `rolf/myfile.txt`).

pathname qualifier
See *variable modifier*

pattern matching
The process of comparing input information (usually text) against a specified set of symbols (usually regular expressions) to find correspondences.

See also *regular expression*

pattern space
In the `sed` editor, the range of lines currently being edited; the pattern space is selected by an address or pair of addresses.

permission code
See *permissions*

permission field
See *permissions*

permissions

The constraints a user places on a file to control what other users or groups may read, write, or execute the file. There are three sets of permissions: those applied to the user, those applied to the user's group, and those applied to everyone else, called "other."

pid, PID

See *process ID*

pipe

The construct that couples the output of one program directory to the input of another. Pipes are created by the use of a vertical bar (|) between commands on the command line. For example:

```
% nroff inputfile -ms | lpr
```

This pipeline processes the input file (with the `nroff` command) and sends the processed file directly to the printer (the `lpr` command).

See also *pipeline*

pipeline

A series of commands connected by pipes. The process of coupling the output of one command directly to the input of another with a pipe is called *pipelining* or *piping*.

piping

See *pipeline*

pixel (picture element)

The smallest element of a display in a graphics application. On a video screen, pixels are the dots that produce the visual image. The number of pixels usually determines the resolution of the image; the more pixels, the better the resolution.

Point-to-Point Protocol

See *PPP (Point-to-Point Protocol)*

pointer

A symbol that specifies position by reflecting the motion of the mouse. The pointer can change shape to indicate the function of the area in which the pointer is position.

See also *cursor*

POSIX (Portable Operating System Interface for Computer Environments)

A collection of standards proposed by the POSIX working groups of the Institute of Electrical and Electronics Engineers (IEEE). POSIX standards define system interfaces to support the source portability of applications.

See also *SVID (System V Interface Definition)*

POSIX shell
The shell that conforms to the POSIX standard. The POSIX shell (`sh`) is a subset of the Korn shell.

See also *Korn shell*, *shell*

PostScript
The registered trademark for a language developed by Adobe Systems, Inc., for specifying the formatting of typeset documents or displays. An encapsulated PostScript file is a file that follows a standard for embedding PostScript files into other PostScript files.

PPP (Point-to-Point Protocol)
A transmission line protocol that encapsulates and transfers IP datagrams over asynchronous serial lines. PPP is more efficient than SLIP.

See also *SLIP (Serial Line Internet Protocol)*

predefined variable
A shell variable defined and maintained by the C shell.

preprocessor
A program that translates some portion of its information in a file into a form understandable to another program. For example, the `tbl` program is a preprocessor for the `nroff` text formatter.

printcap database
A file (`/etc/printcap`) containing descriptions of all the printers known to the system.

process ID
A unique number assigned to a process that is running.

process identification
See *process ID*

process table
A kernel data structure that contains relevant information about all processes in the system.

process variable
A symbol containing information that can be used by the current process only. Process variables are not automatically propagated to child processes.

See also *environment variable*

profile data
Information about how a program is spending its execution time.

See also *profiling*

profiling
The monitoring of how system resources are utilized in a given program. Profiling helps programmers improve the efficiency of their program code. Different versions of the UNIX operating system provide different profiling utilities that work in different ways.

pseudodevice
A device that consists of a software simulation, rather than hardware; for example, a pty (pseudo-tty) device.

pseudoterminal
A special file that effectively functions as a keyboard and display device.

See also *pseudodevice*

pseudo-tty
See *pseudoterminal*

pty
See *pseudoterminal*

pwd
The command that causes the system to display the absolute pathname of the user's working directory.

See also *working directory*

Q

query
1. The action of searching data for desired information.
2. In data communications, the process by which a master station asks a slave station to identify itself and to give its status.
3. In interactive systems, an operation at a terminal or workstation that elicits a response from the system.
4. A request for information from a file based on specific conditions.

queue
A line of items waiting to be processed. For example, a print queue consists of jobs waiting to be printed.

queue daemon

The process that maintains a list of outstanding jobs and sends them to the specified device at the appropriate time.

See also *daemon*, *job*

queued message

A system message that is added to a list of messages stored in a file for user viewing at a later time. Background processes usually produce queued messages. Programs interacting directly with users typically send messages to the screen for immediate user viewing.

queue element

An item in a queue.

quote

To protect a character from interpretation by a program by enclosing it in quotation marks or by preceding it with a backslash character; to mask the special meaning of certain characters, causing them to be taken literally.

See also *escape*

R

raw mode

The condition of a device driver in which the driver does not interpret the data passing through it. For example, a UNIX terminal driver operating in raw mode passes a Return character from the terminal directly to the system.

See also *cooked mode*

raw socket

A socket that provides privileged users access to internal network protocols and interfaces. These socket types can be used to take advantage of protocol features not available through more normal interfaces or to communicate with hardware interfaces.

rc

An element of the name applied to files containing command scripts that control the process of booting a computer. The `rc` characters are also used in the names of files that contain user-customized startup information, such as the BSD mail utility `.mailrc` and the Motif window manager `.mwmrc`.

RCS (Revision Control System)

A set of programs for managing program and documentation source files so that any revision of a given file can be retrieved. Revisions to a file are

stored as a series of incremental changes (deltas) applied to the original version instead of as complete copies of all the versions. The system provides locking mechanisms so that only a single user can apply changes to a given file at any one time.

See also *SCCS (Source Code Control System)*

RCS file

A file stored in the Revision Control System (RCS) library containing the text of the original file and the list of deltas that have been applied to it.

RCS library

The directory in which Revision Control System (RCS) files are stored.

record

1. A collection of related data items treated as a unit. A record contains one or more fields.

2. In `awk`, the information between two consecutive occurrences of the record separator. For most purposes, a record in `awk` can be thought of as a line from the input file.

recursive

In programming, pertaining to a procedure or function that accomplishes its task by repeatedly calling itself until a specified condition is reached. The process of using a recursive procedure or function is called *recursion*.

redirection

The specifying of one or more of the devices with which the standard input, standard output, and standard error virtual files are to be associated during the execution of a given command.

reference page

One of a collection of files containing documentation on all commands, system calls, library routines, and so forth. Reference pages are often called manual pages or man pages. For information about reference pages, see Chapter 4.

regular expression

A pattern of one or more characters used to find text information and formed according to a set of rules that define how the characters are to be interpreted. For example, a period is interpreted as a valid match for any character in the input. The regular expression `a.c` matches any string containing the letter `a` and the letter `c` separated by a single intervening character, such as `abc`, `a?c`, `a9c`, and so on.

See also *pattern matching*

relative pathname
A pathname that begins at the user's working directory; they are written without the initial slash. For example, `docs/myfile.txt` is a relative pathname.

See also *absolute pathname*

restricted shell
A security feature that provides a controlled shell environment with limited features.

Revision Control System
See *RCS (Revision Control System)*

RIS (Remote Installation Services)
A utility for installing software kits across a network instead of by using locally mounted distribution media.

RIS area
A reserved disk area physically connected to a RIS server, containing one or more product environments in which are stored installable software kits.

RIS client
A computer system that has permission to install software across the network by accessing kits stored in the server's RIS area.

RIS server
A computer system that serves other computers by providing software kits for them to install. The software is stored on disks belonging to the server and is accessed across the network by the RIS clients.

RISC (Reduced Instruction Set Computing)
A computer architecture that is based on a limited set of simple instructions instead of a larger and more varied set of more complex instructions.

root
1. The login name for the superuser (system administrator).
2. The name applied to the topmost directory in the UNIX system's tree-like file structure; hence, the beginning of an absolute pathname. The root directory is represented in pathnames by an initial slash (/); a reference to the root directory itself consists of a single slash.

See also *pathname*

root directory
See *root*

root file system

The basic file system, onto which all other file systems can be mounted. The root file system contains the operating system files that get the rest of the system running.

root login

See *root*

routing daemon

A program that provides a routing-management service. The routing daemon, `routed`, is invoked when the system is booted to manage the network routing tables.

See also *daemon*

run time

Pertaining to actions that are taken by a program or system during execution.

See also *compile time*

S

SCCS library

The directory in which Source Code Control System (SCCS) s-files and p-files are stored.

SCCS (Source Code Control System)

A set of programs for managing program and documentation source files so that any revision of a given file can be retrieved. Revisions to a file are stored as a series of incremental changes (deltas) applied to the original version instead of as complete copies of all the versions. The system provides locking mechanisms so that only a single user can apply changes to a given file at any one time.

See also *RCS (Revision Control System)*

SCP (Subset Control Program)

A program that contains path specifications for all of the files related to a product kit. The SCP is written by the kit's developer and is invoked by the `setld` utility during the installation of the kit.

script

1.A nonbinary program that is interpreted and executed by a specified shell.

2. In the `sed` editor, a list of editing commands to be applied to the input file.

SCSI (Small Computer System Interface)
An industry-standard bus for small systems such as personal computers, small multiuser systems, or workstations. SCSI-based devices can be configured in a series, with multiple devices on the same bus. SCSI is pronounced *scuzzy*.

SCSI Interface Module
See *SIM (SCSI Interface Module)*

search path
A list of full pathnames (usually separated by colons) of directories to be searched for executable files and other kinds of files. Users can create search paths by defining variables, such as `path`, `$PATH`, and `MANPATH`.

security
The protection of data, system operations, and devices from accidental or intentional ruin, damage, or exposure.

sed
The command that invokes the `sed` utility, the standard stream editor. The `sed` editor reads one or more text files, makes editing changes according to a script of editing commands, and writes the results to standard output.

Serial Line Internet Protocol
See *SLIP (Serial Line Internet Protocol)*

server
A computer system that serves one or more other computers, called clients, by providing a resource to them.

server process
In the client/server model of communication, a process that provides services to client processes.

See also *passive user*

session
See *terminal session*

setld
A utility for installing, managing, updating, and removing software subsets.

See also *subset*

sh
The command that invokes either the Bourne shell or the POSIX shell, depending on the user setup in the `passwd` file.

shell
A program that interprets commands entered by the user, invoking programs and calling for system resources as needed.

See also *Bourne shell, C shell, Korn shell, POSIX shell*

sign-extend
To increase the data size of an operand smaller than the computer's data path by appending high-order bits to the operand. If the sign bit of the operand is a one, the added bits are ones; if a zero, they are zeroes. This operation preserves the twos-complement numerical value of the operand.

silent character
See *hidden character*

SIM (SCSI Interface Module)
A subprogram designed to accept CAM Control Blocks routed through the XPT transport layer in order to execute SCSI commands.

Simple Mail Transfer Protocol
See *SMTP (Simple Mail Transfer Protocol)*

Simple Network Management Protocol
See *SNMP (Simple Network Management Protocol)*

SLIP (Serial Line Internet Protocol)
A transmission line protocol that encapsulates and transfers IP datagrams over asynchronous serial lines. SLIP is less efficient than PPP.

See also *PPP (Point-to-Point Protocol)*

SMTP (Simple Mail Transfer Protocol)
The Internet standard protocol for exchanging electronic mail.

SNMP (Simple Network Management Protocol)
The Internet standard protocol for exchanging network management information.

socket
In interprocess communications, an endpoint of communication. Also, the system call that creates a socket and the associated data structure.

socketpair
A pair of sockets that can be created in the UNIX domain for two-way communication. Like pipes, socketpairs require communicating processes to be related.

See also *pipe*

soft link

See *symbolic link*

sort

To organize the information in a file into the desired order based on specifiable criteria.

Source Code Control System

See *SCCS (Source Code Control System)*

source hierarchy

For building software kits, the directory tree and files that are to be compiled by the `kits` command into subsets for a kit.

special file

See *device special file*

spooling

The process of copying files into a reserved disk area and then delivering the temporary copies to a serially accessed device as the device becomes ready to receive each new file. The temporary copies are delivered to the device in the order of their creation and are deleted as their delivery is completed; hence, spooling is a form of FIFO (first in, first out) buffering. The most common use of spooling is for printing. Rather than require a user to wait until the printer becomes available, the system spools the file to be printed. The user can then edit or delete the original copy.

standard error

The file to which programs write error messages. The standard error file (commonly called `stderr`) is a virtual file that is by default assigned to the user's screen but can be reassigned (redirected) to any device or file available to the user.

standard input

The file from which most programs receive input data or commands. The standard input file (commonly called `stdin`) is a virtual file that is by default assigned to the user's keyboard but can be reassigned (redirected) to any device or file available to the user.

standard output

The file to which programs write output data. The standard output file (commonly called `stdout`) is a virtual file that is by default assigned to the user's screen but can be reassigned (redirected) to any device or file available to the user.

statement

An instruction in a source language, shell script, command language, and the like.

status
The state in which a program exists.

stderr
See *standard error*

stdin
See *standard input*

stdout
See *standard output*

store-and-forward
A type of network connection in which a complete transmission is passed to one intermediate host before transmission to the next intermediate host begins.

stream
The TCP/IP definition developed for System V systems, and now in wide use across UNIX systems.

stream editor
A program that manipulates the data in a text file by applying commands from a previously prepared list called a script instead of by accepting commands from the user. Powerful stream editors, such as the UNIX system's `sed`, can perform any operation available to a full-function interactive line editor.

STREAMS
A kernel mechanism developed by AT&T that supports the implementation of device drivers and networking protocol stacks.

See also *clist*, *STREAMS framework*

STREAMS framework
STREAMS components that define the interface standards for character I/O within the kernel and between the kernel and user levels. These components include functions, utility routines, kernel facilities, and data structures.

stream socket
A socket that provides two-way byte streams across a transport connection.

stty
A command that sets or reports certain characteristics of the user's terminal.

su

A command that substitutes another user's login for that of the user who invoked the command, logging the invoking user in under the substituted login. The invoking user must know the login password for the user whose login is being substituted. If no other user's login is specified, the command substitutes the root login.

subdirectory

A directory that is contained (nested) in another directory. The containing directory is called the parent directory.

subset

A software kit module that is installed or removed with the Tru64 UNIX `setld` utility. A subset usually consists of a collection of related files, such as an application and its support files.

subset control program

See *SCP (Subset Control Program)*

subset dependency

The condition in which a given subset requires the presence, or lack thereof, of other subsets in order to function properly. Evaluated by a subset's subset control program (SCP) under control of the `setld` utility.

superuser

A user possessing privileges to override the normal restrictions on file access, process control, and so forth. A user who possesses these privileges becomes a superuser by issuing the `su` command, or by logging into the system as root.

suspended

The condition of a process that is stopped but not killed. C shell, Korn shell, and POSIX shell users have the ability to suspend and reactivate processes by using the `fg` and `bg` commands, or by pressing Ctrl/z. A process that is suspended is called a *suspended job*.

See also *terminated*

SVID (System V Interface Definition)

The specification that defines subroutine calls, system calls, commands, utilities, and services under System V.

See also *POSIX (Portable Operating System Interface for Computer Environments)*

SVVS (System V Verification Suite)

A set of programs used to test adherence to the System V Interface Definition.

switch
Another name for an option.

See also *option*

symbolic link
A file that contains the pathname of another file or directory and acts as a pointer to that file or directory. The symbolic link can occur within the same file system or across file systems; also called a soft link.

See also *hard link*

sync
A command that forces all cached disk write operations to be completed before the system is halted.

synchronous execution
A mode of execution that forces transport primitives to wait for specific events before returning control to the transport user.

system call
Functions that access the file system and communication facilities of the kernel.

system load
The demand that all processes place on the computer. System load is usually expressed as a number, with 1.0 representing 100 percent utilization and 0.1 representing 10 percent utilization of system resources.

System V
A version of the UNIX system developed by AT&T.

System V Interface Definition
See *SVID (System V Interface Definition)*

System V Verification Suite
See *SVVS (System V Verification Suite)*

T

tail
A command that displays a user-specifiable number of lines at the end of a text file.

See also *head*

tar program

A program that makes portable copies of files for archiving or transfer to another system. By default, the `tar` program writes its archive files on the system's primary magnetic tape unit.

target

In the `make` utility, an entity to be built from its dependents. An executable program is a target that is built from one or more object modules. Also called a target file.

target hierarchy

For building software kits, the directory tree into which a software kit is placed by the `kits` command.

task

1. A defined activity; a unit of work to be performed, for example, a user task, a server task, and a processor task.

2. A process and the procedures that run the process.

TCP (Transmission Control Protocol)

The Internet transport-layer protocol that provides a reliable, full-duplex, connection-oriented service for applications. TCP uses the IP protocol to transmit information through the network.

TCP/IP

The two fundamental protocols of the Internet Protocol suite, and an acronym that is frequently used to refer to the Internet Protocol suite. TCP provides for the reliable transfer of data, while IP transmits the data through the network in the form of datagrams.

See also *TCP (Transmission Control Protocol)*, *IP (Internet Protocol)*

$TERM

An environment variable containing the user's terminal type.

See also *environment variable*

termcap database

A file containing descriptions of terminal types and capabilities; used by the `tset` command and BSD curses library routines to determine how a given physical terminal is to be controlled.

See also *terminfo database*

terminal session

A user's interaction with a computer between the time the user logs in and logs out.

terminated
The condition of a process that has been permanently killed. A process that is terminated is called a *terminated job*.

See also *suspended*

terminfo database
A file containing descriptions of terminal types and capabilities; used by the system and X/Open curses library routines to determine how a given terminal is to be controlled.

tilde substitution
In the POSIX, Korn, and C shells, use of a tilde (~)as the first character in a pathname. By default, the shell interprets the tilde as the pathname of the user's home directory; for example, if a user whose login name is `rolf` enters `~/docs/figure_1` as a pathname, the system might expand the entry to be `/usr/users/rolf/docs/figure_1`. If the tilde is followed immediately by a user's login name, the shell interprets the combination as a reference to the named user's home directory; for example, `~willy` represents the path to willy's home directory when entered by any user on the system.

tool
A command or utility designed to help get a job done; for example `make` or `dbx`.

tracepoint
A specific place in a source code program in which the value of a variable is printed, without pausing the program's execution. Used to test and debug a program.

See also *breakpoint*

Transmission Control Protocol
See *TCP (Transmission Control Protocol)*

transport endpoint
A communication path over which a transport user can exchange data with a transport provider.

transport provider
A transport protocol that offers transport layer services in a network.

transport services
The support given by the transport layer in a network to the session layer for the transfer of data between user processes. The two types of services provided are connection-oriented and connectionless.

transport user

A program needing the services of a transport protocol to send data to or receive data from another program or point in a network.

trap

1. In data communications, an unprogrammed, hardware-initiated, conditional jump to a specific address. Similar to an interrupt, but triggered by direct action of an executing program, rather than by an external event.

2. In programming languages, the process of branching or jumping to a subroutine that provides the desirable operation when a specific condition occurs.

3. In the UNIX system, a special statement used to catch signals in a shell script and transfer control to a handler routine within the script.

trap handler

A system-defined routine used when an abnormal situation arises during a program's execution.

tree structure

1. The organization of disk directories in most operating systems. Any given directory can contain files or other directories (called subdirectories), or both. By extension, any subdirectory can contain subdirectories of its own; when diagrammed, the resulting structure resembles the branching of a tree.

2. The organization of data in a manner similar to that described for disk directories. Common tree structures in files are the binary tree, in which each data element has zero, one, or two elements beneath it (called children); and the B+ tree, in which each data element can have more than two children, with the distribution of elements in the tree being balanced so that all of the elements at a given level have the same or similar numbers of children.

Trojan Horse

A computer program that appears to do something useful, but is also designed to damage or destroy other files or programs, or the system itself, without the user's knowledge. An example of a Trojan Horse would be a game program that secretly erased disk files while the game was being played.

See also *virus*, *worm*

trusted host

A computer within a network that permits access without the need to supply password information.

tty

A shorthand term for a terminal.

U

UDP (User Datagram Protocol)
The Internet Protocol that allows application programs on remote machines to send datagrams to one another. UDP uses IP to deliver the datagrams.

uid, UID
See *user ID*

ULTRIX
A forerunner to the Tru64 UNIX operating system produced by Digital Equipment Corporation. The ULTRIX operating system runs on VAX and RISC computers, whereas Compaq Tru64 UNIX runs on Alpha systems.

umask
A three-digit octal number that specifies the default permissions given to a file when it is created. The `umask` command sets or changes this number.

UNIX
A trademark of X-Open Company, Ltd., that can be used in names of operating systems that conform to X/Open UNIX CAE specifications and meet other X/Open UNIX branding requirements. The UNIX operating system was originally developed at the Bell Laboratories of AT&T in the late 1960s and early 1970s and subsequently enhanced by the University of California at Berkeley, AT&T, the Open Software Foundation (OSF), and others.

UNIX-to-UNIX Copy Program
See *UUCP (UNIX-to-UNIX Copy Program)*

unlink
The system call used to sever the connection between files that had been created with the `link` system call.

unmount
To announce to the system that a file system previously mounted on a specified directory is to be removed. Only the person who mounted the particular file system or a superuser can unmount it. A file system is unmounted with the `umount` command.

up time
The period during which a machine is available for use.

See also *down time*

upward compatible
Pertaining to that which is designed for use on small machines, but capable of running without change on larger machines.

URL (Uniform Resource Locator)

The address of a file or other resource accessible on the Internet. The type of file or resource depends on the Internet application protocol. For example, using the HyperText Transfer Protocol (HTTP) , the file can be an HTML page, an image file, or a program such as a CGI application or Java applet. Such an address would look like this: `http://www.unix.digital.com`, which is the URL for the Compaq Tru64 UNIX Web site.

User Datagram Protocol

See *UDP (User Datagram Protocol)*

user ID

The number associated with each login name. This number is stored in the `/etc/passwd` file.

user name

See *login name*

/usr

A read-only file system in which some components of the operating system and of applications are stored. Users' home directories are sometimes also located in a subdirectory of `/usr`.

UUCP (UNIX-to-UNIX Copy Program)

A set of programs and protocols developed at the Bell Laboratories of AT&T for the purpose of connecting computers by means of dial-up lines. The programs include facilities for copying files, logging in to remote computers, and encoding binary files for transmission of 7-bit ASCII data lines. The ease of connection and low cost have made UUCP one of the most popular information networks in the world.

UUCP network

A term applied to any grouping of computers connected by means of the UUCP programs.

V

variable

In programming languages, shell scripts, command procedures, and the like, a symbol whose value is allowed to change.

variable expansion

The replacement of the variable identifier with its associated strings in a shell command line.

variable modifier
A symbol referring to part of a variable, usually under the assumption that its value is a pathname.

version control file
In a version control system, a file that consists of original text and a set of revisions (deltas) that have been made to it. In RCS, this file is called an *RCS file*; in SCCS, an *s-file*.

version control library
A directory that contains files that are organized and maintained under a version control system, such as RCS or SCCS.

version control system
A software tool that aids in the organization and maintenance of file revisions and configurations. In particular, it automates the storing, logging, retrieval, and identification of revisions to source programs, documentation, and data files.

See also *version control library*

vi editor
A full-screen text editor. The `vi` editor is a modal editor. In command mode, it accepts commands for cursor movement, text deletion, and so forth. To insert text into the file, the user gives the editor a command that places the editor in input mode, and all keystrokes thereafter are interpreted as input data until the Escape key is pressed.

See also *full-screen editor*

virus
A piece of software designed to attach itself to other computer programs or files in a system and then to replicate itself indefinitely through any available means (disk file, network, and so forth) into other computers. Viruses are usually designed to damage or destroy "infected" programs or systems and are often programmed to become destructive at a specific time, such as the birthday of the virus's programmer.

See also *Trojan Horse*, *worm*

visual editor
See *full-screen editor*

W

wildcard character
A metacharacter that is used to allow wildcard matching in file names or regular expressions.

See also *metacharacter*, *regular expression*

word identifier

A piece of a command line delimited by blanks and recognized as a unique entity by the shell. Used to save keystrokes. By using word identifiers, a user can select part of a previous command line for use in the current command line.

wordlist

A C shell variable consisting of more than one word.

working directory

1. The directory from which a file is read or into which a file is written when a program does not include a directory path in the name of the file when operating on it.

2. The user's current directory.

See also *home directory*

worm

A computer program designed to replicate itself and spread through a network into other computers. Worms are not attached to other programs or files. Worms are usually designed to damage or destroy "infected" systems and are often programmed to become destructive at a specific time, such as the birthday of the worm's programmer. Some worms are not designed to cause damage, but they are still harmful because they occupy resources intended for legitimate use.

See also *Trojan Horse*, *virus*

WORM

Refers to storage media that can be written once and read many times, such as a CD-R recordable compact disc.

X

X/Open Transport Interface

See *XTI (X/Open Transport Interface)*

XPT

A layer of software that SCSI peripheral drivers use to originate the execution of CAM (Common Access Method) functions.

XTI (X/Open Transport Interface)

Protocol-independent, transport-layer interface for applications. XTI consists of a series of C language functions based on the Transport Layer

Interface (TLI), which in turn was based on the transport service definition for the OSI model.

X Window System

A network-based windowing interface developed by the Massachusetts Institute of Technology (MIT). The X Window System has been adopted by many major computer manufacturers.

Y

yacc (Yet Another Compiler-Compiler)

A program for generating parsers (programs that can interpret their input in a rational manner). The output from `yacc` is a C language program. The `yacc` program is usually used to generate parsers for interpreting the output of a `lex`-generated front end.

See also *lex*, *parser*

younger file

For the `make` utility, a dependency file that has changed more recently than its target.

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

W

Reader's Comments

Tru64 UNIX
Documentation Overview
AA-RH8RA-TE

Compaq welcomes your comments and suggestions on this manual. Your input will help us to write documentation that meets your needs. Please send your suggestions using one of the following methods:

- This postage-paid form
- Internet electronic mail: `readers_comment@zk3.dec.com`
- Fax: (603) 884-0120, Attn: UBPG Publications, ZKO3-3/Y32

If you are not using this form, please be sure you include the name of the document, the page number, and the product name and version.

Please rate this manual:

	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Usability (ability to access information quickly)	☐	☐	☐	☐

Please list errors you have found in this manual:

Page	Description
_____	_____
_____	_____
_____	_____
_____	_____

Additional comments or suggestions to improve this manual:

_____
_____
_____
_____

What version of the software described by this manual are you using? _____

Name, title, department _____
Mailing address _____
Electronic mail _____
Telephone _____
Date _____

--------- Do Not Cut or Tear - Fold Here and Tape ----------

COMPAQ

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 33 MAYNARD MA

POSTAGE WILL BE PAID BY ADDRESSEE

COMPAQ COMPUTER CORPORATION
UBPG PUBLICATIONS MANAGER
ZKO3-3/Y32
110 SPIT BROOK RD
NASHUA NH 03062-9987

--------- Do Not Cut or Tear - Fold Here ----------

Cut on Dotted Line