Digital UNIX

Documentation Overview, Glossary, and Master Index

Order Number: AA-QTM6A-TE

March 1996

Product Version: Digital UNIX Version 4.0 or higher

This manual describes the documentation kits that are available for your Digital UNIX system. It also provides a glossary and a master index for the Digital UNIX documentation set.

Digital Equipment Corporation
Maynard, Massachusetts

Digital Equipment Corporation makes no representations that the use of its products in the manner described in this publication will not infringe on existing or future patent rights, nor do the descriptions contained in this publication imply the granting of licenses to make, use, or sell equipment or software in accordance with the description.

Possession, use, or copying of the software described in this publication is authorized only pursuant to a valid written license from Digital or an authorized sublicensor.

Contents

About This Manual

Audience v
New and Changed Features v
Organization vi
Reader's Comments vi
Conventions vii

Part I: Digital UNIX Documentation

Overview of the Digital UNIX Documentation Set

1 Release Notes

2 Online Reference Information

3 The Digital UNIX Full Documentation Set

3.1 End User Documentation Kit 5
3.1.1 Startup Documentation Kit 5
3.1.2 System and Network Management Documentation Kit 8
3.1.3 General User Documentation Kit 12
3.2 Developer's Documentation Kit 15
3.2.1 General Programming Kit 15

3.2.2 Windows Programming Kit 20
3.2.3 Writing Device Drivers Documentation Kit 25

4 The Reference Pages Documentation Set

5 Ordering Digital UNIX Documentation

Part II: Glossary of Terms

Glossary

Part III: Master Index for the Documentation Set

Introduction to the Master Index

Index

Figures

1: Organization of the Full Documentation Set 4
2: Startup Documentation Kit 6
3: System and Network Management Documentation Kit 9
4: General User Documentation Kit 13
5: General Programming Documentation Kit 16
6: Windows Programming Documentation Kit 21
7: Writing Device Drivers Documentation Kit 26
8: Reference Pages Documentation Set 28

Tables

1: Digital UNIX Documentation Kit Order Numbers 30
2: Reference Card and Poster Documentation Packages 30

3: Individual Manual Order Numbers 31

About This Manual

This guide describes the documentation kits and documents that are available for your Digital UNIX system. It also provides a glossary of Digital UNIX terms and a a master index for the Digital UNIX documentation set.

Audience

This guide is for anyone using the Digital UNIX documentation. It can help you decide which manuals in the documentation set are most useful to you and how you can use those manuals.

The glossary helps you understand terms that are new to you.

The master index helps you find a specific topic in the documentation set by pointing you to the pages in the appropriate manual where the topic is discussed.

New and Changed Features

The *Documentation Overview, Glossary, and Master Index* has been revised in the following ways for this release:

- *Documentation Overview*

  This part of the manual has been revised to describe accessing the new HTML documents. These new documents replace the Bookreader documents that were included in previous releases.

  The manual has also been revised to add descriptions for new manuals and to modify descriptions of manuals, where necessary. Order numbers for documentation kits and individual manuals have been updated.

- *Glossary*

  New terms have been added for this release. Where needed, definitions of existing terms have been updated.

- *Master Index*

  The *Master Index* has been updated and reflects the new hardcopy Digital UNIX documentation set. Also, an HTML version of this part of the manual is available for the first time at this release.

Organization

This guide is divided into three parts as follows:

- Part I

  Describes the Digital UNIX documentation set. It explains how the documentation is organized into separate kits, provides a brief description of each manual in the set, and provides information on how to order printed documentation.

  This part of the guide is organized as follows:

  – Section 1 provides information about the release notes.

  – Section 2 describes the Digital UNIX online reference information.

  – Section 3 describes the packaging of the Digital UNIX documentation and the contents of each manual.

  – Section 4 describes the contents of the Digital UNIX Reference Pages.

  – Section 5 provides information on how to order Digital UNIX printed documentation.

- Part II

  Contains a glossary of computer terms, particularly those related to the operating system.

- Part III

  Contains the master index for the documentation set.

Reader's Comments

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

You can send your comments in the following ways:

- Fax: 603-881-0120 Attn: UEG Publications, ZK03-3/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:

  Digital Equipment Corporation
  UEG Publications Manager
  ZK03-3/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 Digital UNIX that you are using.
- If known, the type of processor that is running the Digital UNIX software.

The Digital 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 Digital technical support office. Information provided with the software media explains how to send problem reports to Digital.

Conventions

The following conventions are used in this document:

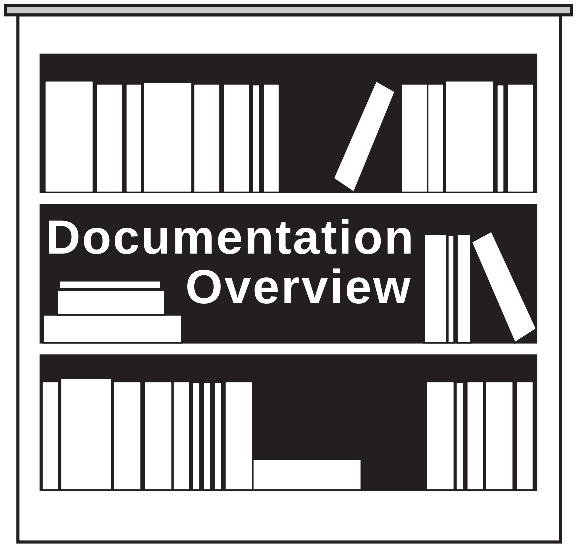
`%`
`$`
A percent sign represents the C shell system prompt. A dollar sign represents the system prompt for the Bourne and Korn shells.

`%` **`cat`**
Boldface type in interactive examples indicates typed user input.

`cat`(1)
A cross-reference to a reference page includes the appropriate section number in parentheses. For example, `cat`(1) indicates that you can find information on the `cat` command in Section 1 of the reference pages.

Part I: Digital UNIX Documentation

This part introduces you to the Digital UNIX documentation set. It explains how the documentation is organized into separate kits, provides a brief description of each manual in the set, and provides information on how to order printed documentation.



Overview of the Digital UNIX Documentation Set

Information in the Digital UNIX documentation set is presented in different formats:

- Most of the Digital UNIX documentation, including the reference pages, is available on the Digital UNIX CD-ROM in a format that is readable with the `netscape` software. To read this HTML documentation, launch the `netscape`, software from the CDE Applications panel. Then choose the documentation link on the Digital UNIX homepage.

  For help using the `netscape` software, pull down the Help menu from the `netscape` menu bar.

  Although most manuals are available in HTML format, a few are available online only in Bookreader format. Those manuals are noted in Section 3.

- In addition to being available through the `netscape` software, the Digital UNIX reference pages are available through the `xman` or `man` command. For information about using the `xman` or `man` command, see Section 2.

- The Digital UNIX product is supplied with a small number of manuals in printed form. These are the manuals in the Startup Documentation Kit, described in Section 3.

  To receive any other manual in the documentation set in printed form, you must order that manual.

  Certain manual are available only in printed form. Those manuals are noted in Section 3.

  For information about ordering printed manuals, see Section 5.

- A few manuals are provided in PostScript format. These manuals are not accessible through the `netscape` software or the `xman` or `man` command. You cannot order printed copies of these manuals from Digital. See the *Release Notes* for information about these manuals.

1 Release Notes

The *Release Notes* are for all users of the Digital UNIX system. They are packaged with the Startup Documentation Kit to ensure that the person who installs the system has access to them during installation.

Before using the Digital UNIX operating system, be sure to read the *Release Notes*.

2 Online Reference Information

The Digital UNIX documentation provides online reference information. This information consists of 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. (For more information about the contents of the reference pages, see Section 4.)

You can read the reference pages on line using the `netscape` command, just as you read the other manuals in the documentation set. You can also print reference pages you view with the `netscape` software by clicking on the Print Toolbar button. The `netscape` software displays a dialog box that allows you to choose the characteristics of the print job.

Alternatively, you can read the reference pages using the `xman` command or the `man` 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.

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, issue the following command:

```
% man man
```

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

In addition to being available online, the reference pages are available in printed manual form. See Section 5 for information about ordering printed reference pages.

3 The Digital UNIX Full Documentation Set

The Digital UNIX Full Documentation Set (shown in Figure 1) contains all the manuals in the Digital UNIX documentation set (except the manuals that are available only in PostScript format and the reference pages).

The Full Documentation Set is divided into two kits, each of which contains subkits that address the needs of different users, as follows:

- End User Documentation Kit

  The End User Documentation kit contains all the information needed to install and use the Digital UNIX system.

  This kit is divided into three subkits as follows:

  – Startup Documentation

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

    When you order the Digital UNIX operating system, the Startup Documentation Kit is packaged with the distribution media.

  – System and Network Management Documentation

    This kit is for people who are responsible for managing the Digital UNIX system or network. The manuals in this kit provide information on how to manage Digital UNIX systems.

  – General User Documentation

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

- Developer’s Documentation Kit

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

  This kit is divided into the following subkits:

  – General Programming

    This kit provides information of interest to most programmers who write applications that run on the Digital UNIX system. Manuals in this kit describe using the programming tools and interfaces.

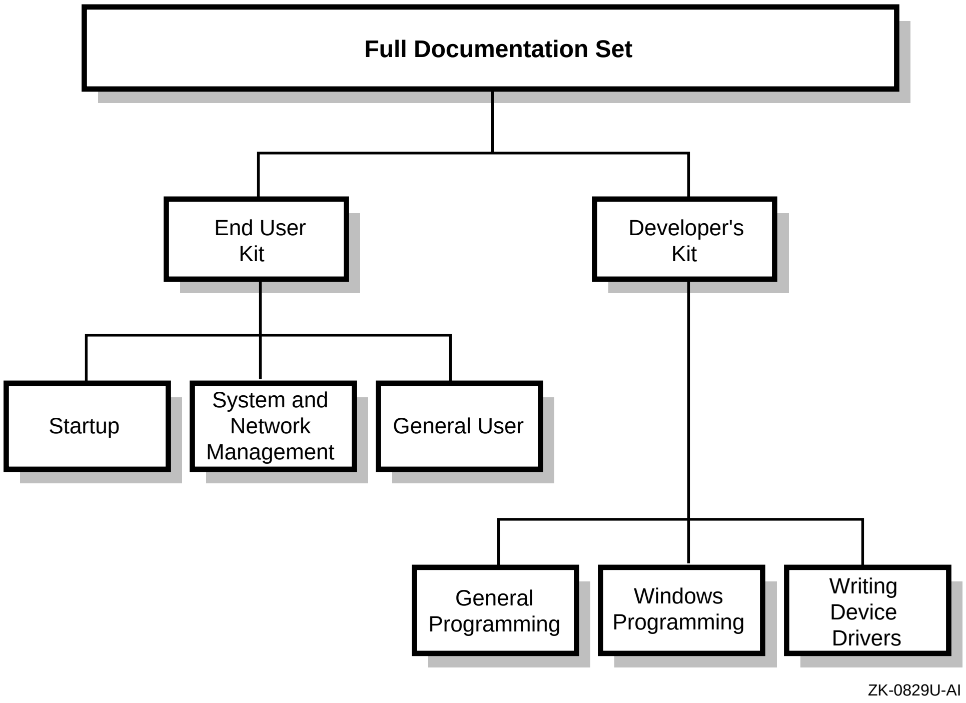
- Windows Programming

  This kit provides information for programmers who are creating a window interface to an application. It provides information about programming for the Common Desktop Environment (CDE), OSF/Motif, and the X Window System.

- Writing Device Drivers

  This kit describes how to write device drivers for hardware that runs the Digital UNIX operating system.

Figure 1: Organization of the Full Documentation Set



The printing on the spines of the Digital UNIX documentation set is color coded to help specific audiences quickly find the books that meet their needs. This color coding is reinforced with the use of an icon on the spines of the books. The following list describes this convention:

Audience	**Icon**	**Color Code**
General users	G	Blue
System and network administrators	S	Red
Programmers	P	Purple
Device driver writers	D	Orange
Reference page users	R	Green

Some books in the documentation set help meet the needs of several audiences. For example, the information in some system books is also used by programmers. Keep this in mind when searching for information on specific topics.

Also, you might occasionally require a book even though you are not a member of its target audience. For example, a general user might want to use the manual *Programming Support Tools* to get advanced information about using certain commands, such as `grep`, `awk`, and `sed`.

3.1 End User Documentation Kit

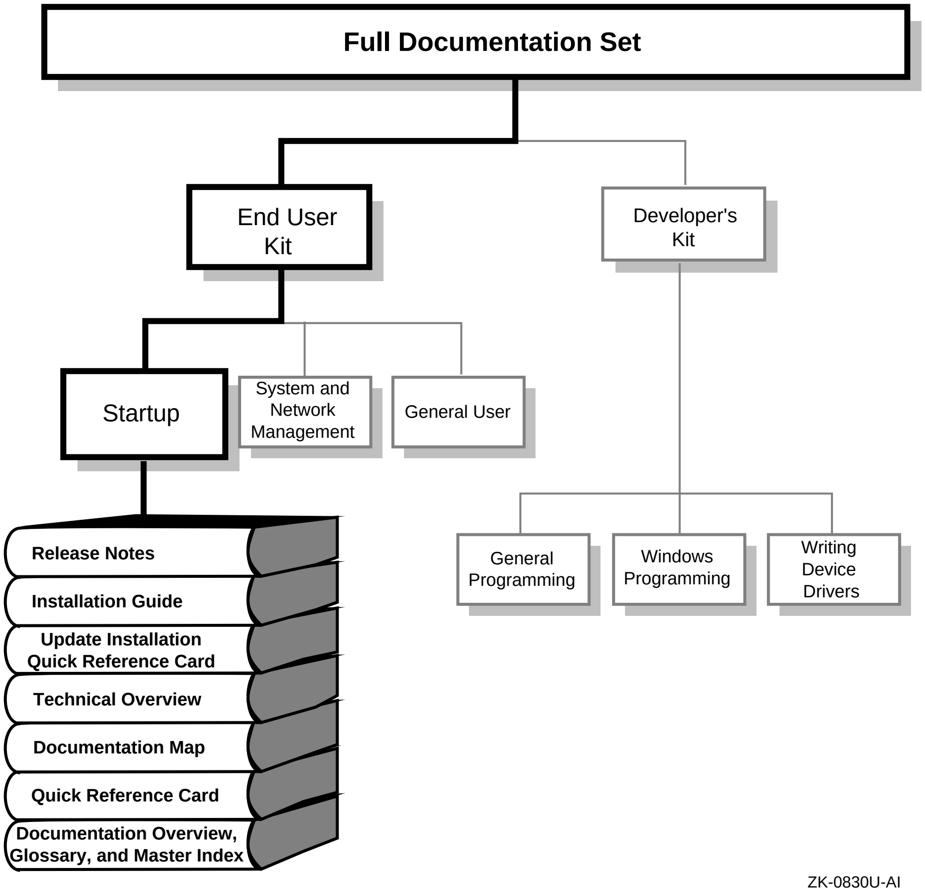
The End User Documentation Kit contains all the information you need to install and configure your Digital UNIX system, to bring it up on a network, and to use the system.

The End User Documentation Kit consists of three subkits. The following sections describe the manuals in each of these kits.

3.1.1 Startup Documentation Kit

The Startup Documentation Kit is packaged in printed form with your Digital UNIX media. As shown in Figure 2, this kit consists of documents you need to install, configure, and start your system. This kit also contains the *Documentation Map,* the *Quick Reference Card,* and the *Update Installation Quick Reference Card*.

Figure 2: Startup Documentation Kit



The following list describes the documents in the Startup Documentation Kit:

- *Release Notes*

  The content of this manual differs in the printed and HTML formats.

  Both formats contain the release notes for the Digital UNIX Version 4.0 operating system. The release notes describe problems you might encounter when working with the Digital UNIX system and possible solutions for those problems.

  The printed format also contains information about new and changed features of the operating system, as well as plans to retire obsolete features of the operating system. Obsolete features are features that have been replaced by new technology or otherwise outdated and are no longer needed.

The HTML bookshelves contain a separate manual, *New and Changed Features*, that describes the new and changed features and plans to retire obsolete interfaces.

The release notes are for the person installing the product and for anyone using the product following installation.

- *Installation Guide*

  This manual describes the procedures to perform an update installation, a basic installation, or an advanced installation of the Digital UNIX product on all supported processors. It explains how to prepare your system for installation, boot the processor, and perform the installation procedure. It also discusses system management procedures in a standalone environment.

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

  This foldout card provides easy access to the information you need to update the Digital UNIX system. You perform an update installation when you want to update the current version of the Digital UNIX system to the next version. Update installations preserve disk partitions, file systems, and file customizations.

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

- *Technical Overview*

  This manual provides a technical overview of the Digital UNIX system, focusing on the networking subsystem, the file system, virtual memory, and the development environment. In addition, the manual lists all system limits.

  This manual does not supersede the Software Product Description (SPD), which is the definitive description of the Digital UNIX system.

- *Documentation Map* (available only in printed form)

  This poster illustrates the Digital UNIX full documentation set to help you determine which books in the documentation set are of interest to you.

- *Quick Reference Card* (available only in printed form)

  This foldout card allows you to quickly look up the format of user commands, such as `cd`, `chmod`, `grep`, `lpr`, and `man`. The card also describes flags that you are likely to use with each command. The card contains command summaries for the `vi`, `emacs`, `MH`, and `mailx` 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.

  This card is for anyone who uses the Digital UNIX operating system.

- *Documentation Overview, Glossary, and Master Index*

  This manual has three parts, as described in the following list:

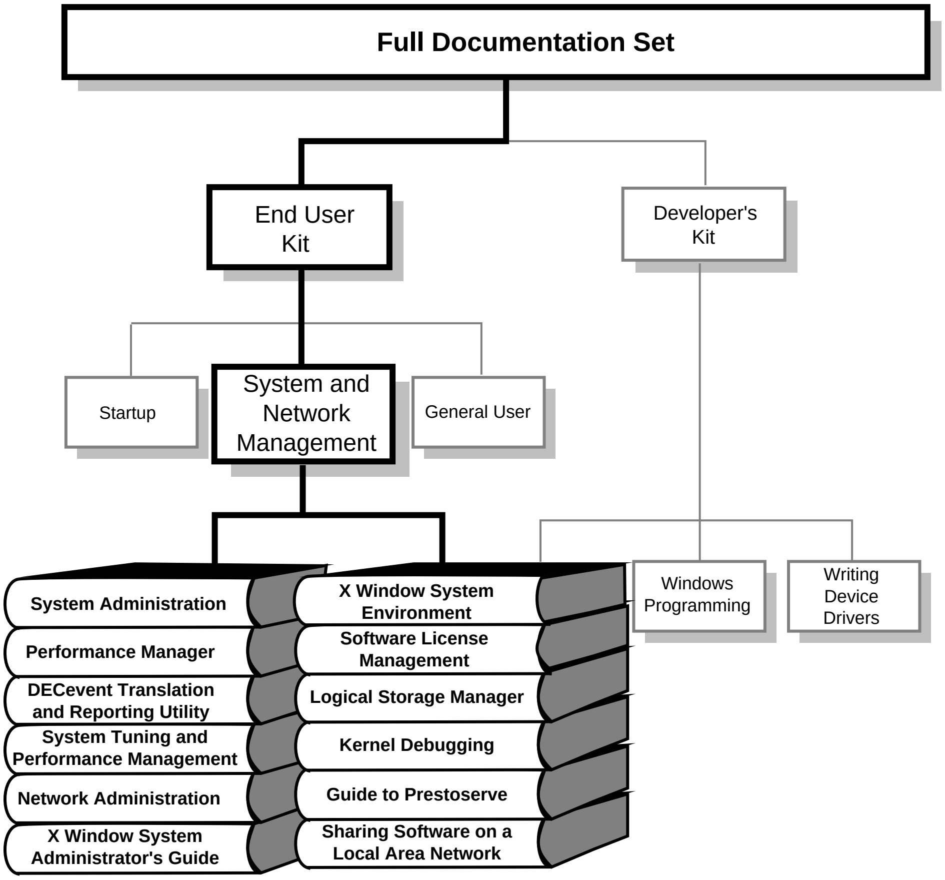
  - Part 1 provides an overview of the Digital UNIX documentation.
  - Part 2 provides a glossary of Digital UNIX terms.
  - Part 3 provides a master index for the Digital UNIX documentation set. The master index contains index entries from the manuals in the Digital UNIX documentation set. These index entries are designed to help you determine which book you should reference for information on a particular topic.

  This manual is for all users of the documentation set.

3.1.2 System and Network Management Documentation Kit

The System and Network Management Documentation Kit, shown in Figure 3, provides information for the person who manages the Digital UNIX system and its networks. This kit contains manuals that describe how to configure systems and networks, maintain disks, and use system administration tools.

Figure 3: System and Network Management Documentation Kit


Full Documentation Set
End User Kit
Developer's Kit
Startup
System and Network Management
General User
Windows Programming
Writing Device Drivers
System Administration
Performance Manager
DECevent Translation and Reporting Utility
System Tuning and Performance Management
Network Administration
X Window System Administrator's Guide
X Window System Environment
Software License Management
Logical Storage Manager
Kernel Debugging
Guide to Prestoserve
Sharing Software on a Local Area Network

ZK-0831U-AI

The System and Network Management Documentation Kit contains the following manuals:

- *System Administration*

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

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

- *Performance Manager*

  This manual explains the concepts of the the POLYCENTER Performance Solution 2.0. The manual describes how to perform the following tasks on your system:

  - Monitoring
  - Thresholding
  - Archiving
  - Distributed command execution

  This manual is intended for system administrators who use the POLYCENTER Performance Solution 2.0.

- *DECevent Translation and Reporting Utility* (Available online in Bookreader format)

  This manual describes all DECevent command features related to the translation and reporting of events on Digital UNIX operating systems. DECevent provides the interface between a system user and the operating system's event logger.

  The manual 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 Digital UNIX operating systems.

  Read this manual if you intend to use the DECevent software to troubleshoot Digital UNIX systems problems.

- *System Tuning and Performance Management*

  This manual describes tools and methodologies for diagnosing system performance problems. It also describes possible resolutions to the problems.

  This manual is for system administrators who want to improve the performance of their system. It is also for systems programmers who are writing applications that affect operating system performance.

- *Network Administration*

  This manual provides information on the tasks you need to complete to establish your system on a network and to configure your network software (such as NFS and BIND). 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.

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

  This manual describes how to customize a wide range of X window 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 Digital UNIX systems. The manual gives information on how to perform system administration tasks for the Digital UNIX X Window System environment. It also describes how to customize X Window System resources and key mappings and provides information about programming with the Digital UNIX X Window System environment.

- *Software License Management*

  This manual describes how to use the License Management Facility (LMF) to manage software licenses from Digital Equipment Corporation. System administrators can use LMF to help them ensure that licenses are used as intended.

  This manual is for system administrators responsible for managing software licenses on Digital UNIX systems. The manual also provides information for anyone who uses licensed software on Digital UNIX systems.

- *Logical Storage Manager*

  This manual provides system administrators with a thorough knowledge of the concepts and procedures involved with disk and volume management using the Logical Storage Manager (LSM). LSM helps you to more effectively manage disk resources, gain high data availability, and increase I/O performance. LSM provides the ability to divide disks into subdisks, concatenate disks, stripe data across disks, and mirror data for duplication of data.

  The manual describes in detail the three LSM interfaces used to perform LSM disk management operations. The LSM interfaces include a graphical-user interface (dxlsm), a character-cell menu interface (voldiskadm), and a command-line interface. You can also use LSM utilities to encapsulate user data currently existing on UNIX-style partitions, Logical Volume Manager (LVM) volume groups, or Advanced File System (AdvFS) storage domains, into LSM volumes.

  This manual is for system administrators who use the LSM software to manage their disk space.

- *Kernel Debugging*

  This manual provides information about debugging kernels. 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.

- *Guide to Prestoserve*

  This manual shows how to use and monitor Prestoserve. Prestoserve speeds up synchronous disk writes, including NFS server access, by reducing the amount of disk I/O.

  This manual is for the person who manages and maintains a Digital UNIX system that includes the optional Prestoserve hardware and software. It assumes that this individual is familiar with Digital UNIX commands, system configuration, and system hardware.

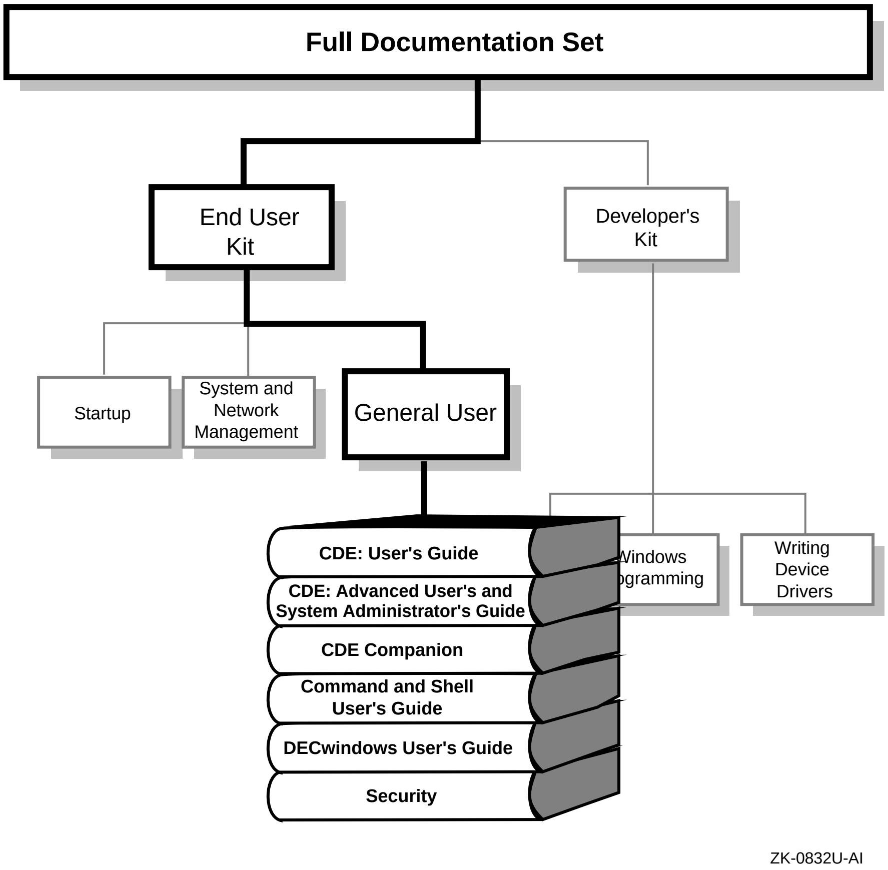
- *Sharing Software on a Local Area Network*

  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 shares the `/usr` and `/var` file systems.

3.1.3 General User Documentation Kit

The General User Documentation Kit, shown in Figure 4, contains important information for all users of the Digital UNIX operating system. This kit contains introductory information for people who are unfamiliar with the Digital UNIX system.

Figure 4: General User Documentation Kit



The following manuals are part of the General User Documentation Kit:

- *Common Desktop Environment: User's Guide*

  This manual describes the basic features of the Common Desktop Enviroment (CDE). It describes how to use the desktop and the desktop applications.

  This manual is intended for anyone who uses CDE.

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

  This manual describes how to customize the appearance and behavior of CDE. It includes chapters on:

  - 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 advanced users who want to perform customizations that cannot be accomplished using the desktop user interface. The desktop provides user-specific locations for many of its configuration files. This book is also intended for system administrators. Many of the tasks in this book require superuser permission.

- *CDE Companion*

  This manual describes Digital's implementation of CDE.

  This manual is intended for anyone who uses CDE on a Digital UNIX system.

- *Command and Shell User's Guide*

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

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

- *DECwindows User's Guide*

  This manual describes how to log on to a Digital UNIX workstation and begin working with the DECwindows Motif interface. It also explains how to customize your windows environment, how to use advanced features of Mail, and how to use AccessX software.

  This manual is primarily for users who have little or no familiarity with computers or those with little knowledge of UNIX-compatible systems. Advanced users might refer to this manual for its description of desktop applications, such as `dxdiff`, and for topics such as how to use DECwindows with a keyboard.

- *Security*

  This manual describes how to use, administer, and write programs that run on the Digital UNIX system with the optional enhanced security subsets installed. When installed, the optional enhanced security subsets help protect your system or data from access by unauthorized users.

Information about the Security Information Architecture (SIA) and about base security can also be found in this manual.

The manual has three audiences, as described in the following list:

- Part 1 describes how to use the Digital UNIX system with enhanced security from the command line.
- Part 2 describes how to administer the enhanced security aspects of the operating system, 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 Digital UNIX system with enhanced security. This part assumes general programming knowledge, including knowing how to use Digital UNIX programming tools.

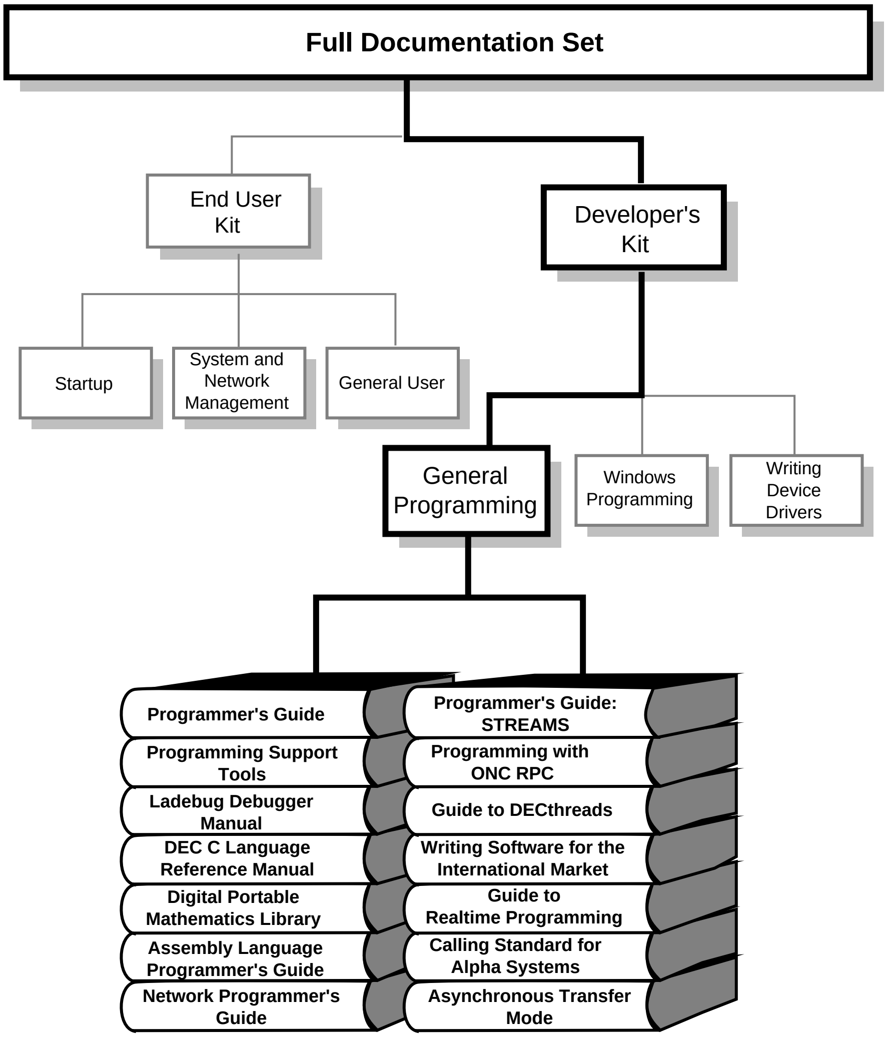
3.2 Developer’s Documentation Kit

The Developer’s Documentation Kit is for software developers who write software applications on or for the Digital UNIX system. The documentation in this kit is divided into three subkits, which are explained in the following sections.

3.2.1 General Programming Kit

The manuals in the General Programming Kit, shown in Figure 6, describe the Digital UNIX programming environment.

Figure 5: General Programming Documentation Kit


Full Documentation Set
End User Kit
Developer's Kit
Startup
System and Network Management
General User
General Programming
Windows Programming
Writing Device Drivers
Programmer's Guide
Programming Support Tools
Ladebug Debugger Manual
DEC C Language Reference Manual
Digital Portable Mathematics Library
Assembly Language Programmer's Guide
Network Programmer's Guide
Programmer's Guide: STREAMS
Programming with ONC RPC
Guide to DECthreads
Writing Software for the International Market
Guide to Realtime Programming
Calling Standard for Alpha Systems
Asynchronous Transfer Mode

ZK-0834U-AI

The General Programming Kit includes the following manuals:

- *Programmer's Guide*

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

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

- *Programming Support Tools*

  This manual describes several commands and utilities in the Digital UNIX system, including facilities for text manipulation, macro and program generation, and source file management.

  The commands and utilities described in this manual are primarily for programmers, but some of them (such as `grep, awk, sed,` and the Source Code Control System (SCCS)) are useful for other users. This manual assumes that you are a moderately experienced user of UNIX systems.

- *Ladebug Debugger Manual* (Available online in Bookreader format)

  This manual describes the Digital Ladebug debugger for Digital UNIX systems. The manual is organized in two parts. Part 1 describes the Ladebug graphical (window-based) user interface (GUI). You can run all major Ladebug features from the GUI. Part 2 describes the command-line interface. You can use the command-line interface from the command prompt within the GUI or from the shell-level prompt.

  You use the Ladebug debugger to debug executable programs at both the source-code and machine-code levels. You can debug programs written in C and C++, Ada, COBOl, and Fortran. Ladebug enables you to debug multiprocess and multithreaded applications, perform kernel debugging, and perform remote client/server debugging. This manual is for developers who use the Ladebug debugger.

- *DEC C Language Reference Manual* (Available online in Bookreader format)

  This manual provides reference information for using the DEC C language on Digital systems. DEC C is an ISO/ANSI-compliant C compiler for Digital 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 (called the ANSI C standard in this manual). 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. The manual contains little task-oriented material or platform-specific material; for that type of information, see the `cc`(1) reference page and the *Programmer’s Guide*.

- *Digital Portable Mathematics Library* (Available online in Bookreader format)

  This manual provides reference and exception information for DPML, Digital’s Portable Mathematics Library software. This manual documents the DPML 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 DPML routines in their language of choice. This audience needs to access DPML routines directly from their application programs.

- *Assembly Language Programmer's Guide*

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

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

- *Network Programmer's Guide*

  This manual describes the Digital 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 Digital 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 and the reader is familiar with the following:

  – C language

  – Programming interfaces for UNIX operating systems

  – Basic data communications concepts, including the Open System Interconnection 7-layer model

  – Typical software interfaces at each layer of the OSI model

  – Requirements for writing communications applications

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

  This manual (developed by AT&T) provides information on the use of the STREAMS mechanism at the user and kernel levels. The manual contains introductory information for those who are unfamiliar with the STREAMS mechanism. It 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. The book also describes the STREAMS multiplexing facility and the STREAMS-based terminal and pseudo-terminal subsystems.

This manual is for programmers developing user-level applications, modules, and drivers using the STREAMS mechanism.

- *Programming with ONC RPC*

  This manual provides an overview of high-level programming with open-network remote procedure calls (ONC RPC). 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.

- *Guide to DECthreads*

  This manual provides usage and reference information on DECthreads routines.

  DECthreads provides the following three interfaces that allow you to perform multithreaded operations:

  – cma

  – pthread

  – 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.

- *Writing Software for the International Market*

  This manual provides an overview of writing international software and details about using the tools provided on the Digital UNIX system.

  Internationalization is the process of designing or adapting programs to meet international requirements, such as those of multiple local languages and the specific character sets associated with them. An international program interacts with users in their own language and reflects the culture of the users’ region.

  This manual is for programmers developing international applications for the Digital UNIX system.

- *Guide to Realtime Programming* (Available online in Bookreader format)

  This manual is for programmers who are developing realtime applications on Digital UNIX systems. Users may be writing new realtime applications or they may be 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. (Available online in Bookreader format)

- *Calling Standard for Alpha Systems*

  This manual defines the requirements, mechanisms, and conventions used in the interface that supports procedure calls on Digital 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 Digital 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.

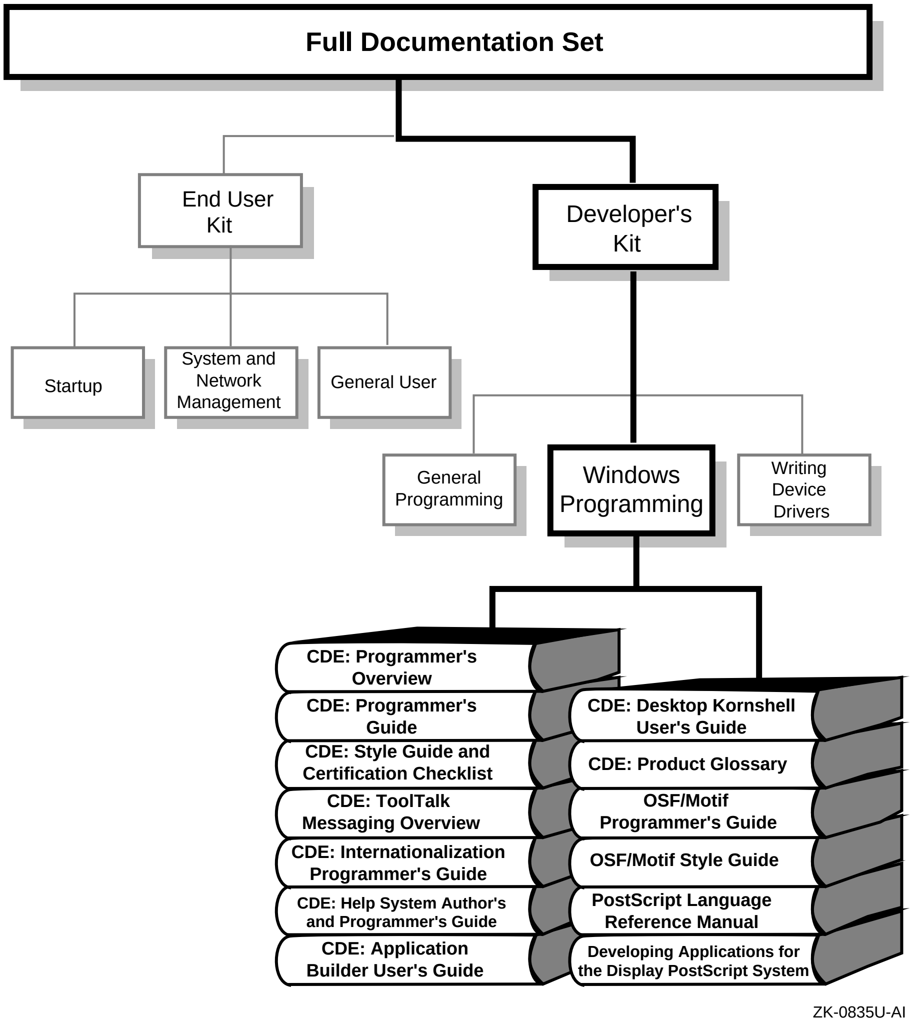
- *Asynchronous Transfer Mode*

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

3.2.2 Windows Programming Kit

The Windows Programming Kit, shown in Figure 7, contains programming information specifically for programmers developing Common Desktop Envrionment (CDE) applications or X window applications on or for the Digital UNIX system.

Figure 6: Windows Programming Documentation Kit


Full Documentation Set
End User Kit
Developer's Kit
Startup
System and Network Management
General User
General Programming
Windows Programming
Writing Device Drivers
CDE: Programmer's Overview
CDE: Programmer's Guide
CDE: Style Guide and Certification Checklist
CDE: ToolTalk Messaging Overview
CDE: Internationalization Programmer's Guide
CDE: Help System Author's and Programmer's Guide
CDE: Application Builder User's Guide
CDE: Desktop Kornshell User's Guide
CDE: Product Glossary
OSF/Motif Programmer's Guide
OSF/Motif Style Guide
PostScript Language Reference Manual
Developing Applications for the Display PostScript System
ZK-0835U-AI

The Windows Programming Kit includes the following manuals:

- *Common Desktop Environment: Programmer's Overview*

  This manual provides a high-level view of the development environment and the developer documentation set for CDE. Read this book first before starting application design and development.

  This manual is intended 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: 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 is for programmers who integrate existing applications into CDE, or develop new applications that use the features and functionality of CDE. This manual assumes that you are familiar with Motif, X, UNIX, or C programming.

- *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.

  This manual is for programmers who develop applications that conform to CDE style.

- *Common Desktop Environment: ToolTalk Messaging Overview*

  This manual describes how the ToolTalk service works and how it uses information that your application supplies to deliver messages. It also describes how applications use the ToolTalk service and ToolTalk components.

  This manual is for developers who create or maintain applications that use the ToolTalk service to interoperate with other CDE applications. This manual assumes familiarity with the ToolTalk service and its functionality, UNIX operating system commands, system administrator commands, and system terminology.

- *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 intended for CDE application programmers whose products are available worldwide.

- *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: Application Builder User's Guide*

  This manual introduces the Application Builder (referred to throughout this document as App Builder) and explains how to use it to build CDE applications.

  This manual is for anyone who wants to build a user interface or prototype interface with App Builder. Because you can easily create and modify user interfaces using App Builder, it is a powerful tool for programmers and non-programmers, including 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 (`kshell`) scripts. It also provides several example scripts of increasing complexity.

  This manual is intended 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: Product Glossary*

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

  This manual is for all CDE users.

- *OSF/Motif Programmer's Guide* (Available online in Bookreader format)

  This manual (developed by the OSF) 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 online in Bookreader format)

  This manual (developed by the OSF) 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.

- *PostScript Language Reference Manual* (available only in printed form)

  This manual provides the definitive documentation for the syntax and semantics of the standard PostScript language, the associated imaging model, and the effects of the graphical operators.

  This manual is for programmers writing applications that generate PostScript page descriptions.

  From time to time, Adobe System, Inc. publishes supplements to their documentation. The latest set of supplements is included on the Digital UNIX distribution kit. See Section 6 for information about printing copies of the supplements.

- *Developing Applications for the Display PostScript System*

  This manual introduces the Display PostScript system extensions of Digital's windowing software. The manual describes specific concepts, tasks, and facts that programmers must know to write Display PostScript applications for windowing software.

  This manual is for experienced UNIX programmers. It assumes a familiarity with the C programming language and the PostScript programming language. In addition, this manual is meant to be used in conjunction with the *PostScript Language Reference Manual*.

- The following manuals are unavailable in printed form. However, they are available online in Bookreader format:

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

    This manual provides supplemental information to the *OSF/Motif Style Guide*. It contains more detailed explanations and illustrations for application developers to help them create consistent user interfaces for their applications.

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

- *DECwindows Motif Guide to Application Programming*

  This manual describes the DECwindows Motif Toolkit and how to use it to design a DECwindows application interface. In particular, it describes the programming interface for widgets provided by Digital in the Toolkit.

  This manual is for programmers who need information about the DECwindows Motif Toolkit.

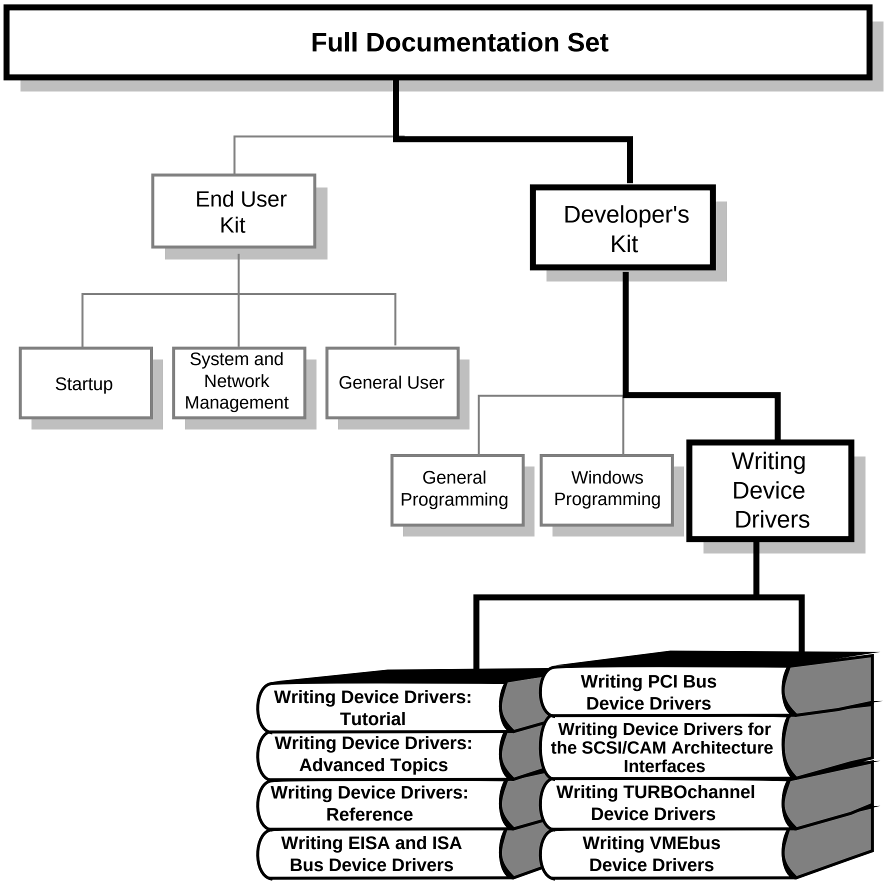
- *DECwindows Extensions to Motif*

  This manual describes the programming extensions that Digital provides to supplement the X Window System, Version 11, Release 5, and OSF/Motif Toolkit components included in systems based on the UNIX environment. This manual supplements the *OSF/Motif Programmer's Guide* and the X Window System manuals and contains reference information for programmers who want to write applications that use the DECwindows Motif Version 1.2 interfaces.

3.2.3 Writing Device Drivers Documentation Kit

The Writing Device Drivers Kit, shown in Figure 8, contains programming information specifically for system engineers developing device drivers for the Digital UNIX operating system.

Figure 7: Writing Device Drivers Documentation Kit


Full Documentation Set
End User Kit
Developer's Kit
Startup
System and Network Management
General User
General Programming
Windows Programming
Writing Device Drivers
Writing Device Drivers: Tutorial
Writing Device Drivers: Advanced Topics
Writing Device Drivers: Reference
Writing EISA and ISA Bus Device Drivers
Writing PCI Bus Device Drivers
Writing Device Drivers for the SCSI/CAM Architecture Interfaces
Writing TURBOchannel Device Drivers
Writing VMEbus Device Drivers
ZK-0836U-AI

The Writing Device Drivers Kit includes the following manuals:

- *Writing Device Drivers: Tutorial*

  This manual provides information for systems engineers who write device drivers for hardware that runs the Digital UNIX operating system. Systems engineers can find information on driver concepts, device driver interfaces, kernel interfaces used by device drivers, kernel data structures, configuration of device drivers, and header files related to device drivers.

- *Writing Device Drivers: Advanced Topics*

  This manual provides information on topics that are beyond the scope of the core tutorial. Systems engineers can find information on such advanced topics as kernel threads and writing device drivers in a symmetric multiprocessing (SMP) environment. The manual also contains information about writing disk drivers.

- *Writing Device Drivers: Reference*

  This manual contains descriptions of the header files, kernel support interfaces, `ioctl` commands, global variables, data structures, device driver interfaces, and bus configuration interfaces associated with device drivers. The descriptions are formatted similarly to the Digital UNIX reference pages.

- *Writing EISA and ISA Bus Device Drivers*

  This manual provides information for systems engineers who write device drivers for the EISA/ISA bus. The manual describes EISA/ISA bus-specific topics, including EISA/ISA bus architecture and data structures that EISA/ISA bus device drivers use.

- *Writing PCI Bus Device Drivers*

  This manual provides information for systems engineers who write device drivers for the PCI bus. The manual describes PCI bus-specific topics, including PCI bus architecture and data structures that PCI bus device drivers use.

- *Writing Device Drivers for the SCSI/CAM Architecture Interfaces*

  This manual provides information for systems engineers who write device drivers for the SCSI/CAM Architecture interfaces.

  The manual provides an overview of the Digital UNIX SCSI/CAM Architecture and describes User Agent routines, data structures, common and generic routines and macros, error handling and debugging routines.

  The manual includes information on configuration and installation. Examples show how programmers can define SCSI/CAM device drivers and write to the SCSI/CAM special I/O interface supplied by Digital to process special SCSI I/O commands.

- *Writing TURBOchannel Device Drivers*

  This manual contains information systems engineers need to write device drivers that operate on the TURBOchannel bus. The manual describes TURBOchannel-specific topics, including TURBOchannel kernel interfaces that TURBOchannel device drivers use.

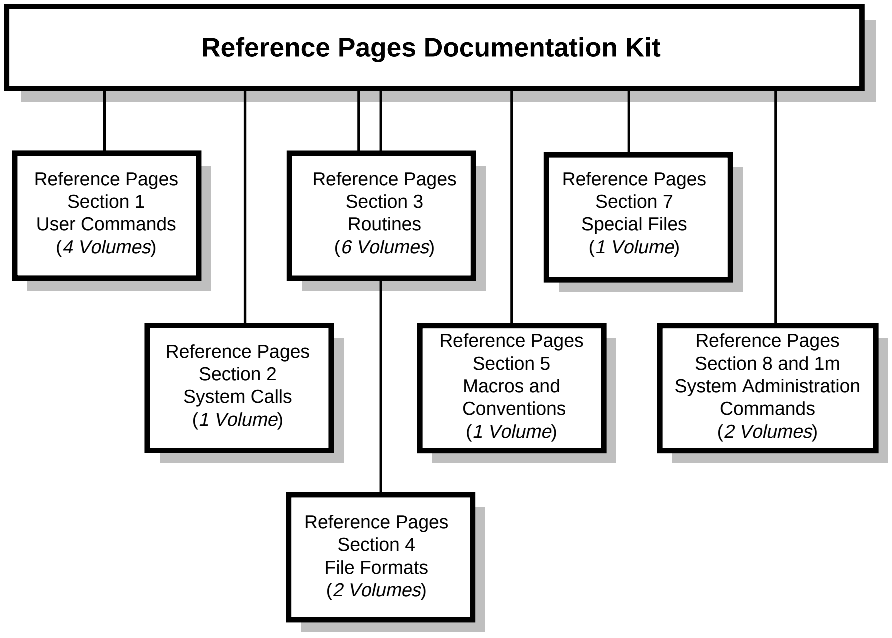
- *Writing VMEbus Device Drivers*

  This manual contains information systems engineers need to write device drivers that operate on the VMEbus. The manual describes VMEbus-specific topics, including VMEbus architecture and kernel interfaces that VMEbus device drivers use. A VMEbus device driver example illustrates the use of these kernel interfaces.

4 The Reference Pages Documentation Set

The Reference Pages Documentation Set, shown in Figure 10, contains the reference information for the Digital UNIX operating system. The Reference Pages Documentation Set is not part of the Full Documentation Set, but is separately orderable as described in Section 5.

Figure 8: Reference Pages Documentation Set


Reference Pages Documentation Kit
Reference Pages Section 1 User Commands (*4 Volumes*)
Reference Pages Section 3 Routines (*6 Volumes*)
Reference Pages Section 7 Special Files (*1 Volume*)
Reference Pages Section 2 System Calls (*1 Volume*)
Reference Pages Section 5 Macros and Conventions (*1 Volume*)
Reference Pages Section 8 and 1m System Administration Commands (*2 Volumes*)
Reference Pages Section 4 File Formats (*2 Volumes*)

ZK-0923U-AI

The spines on the Reference Pages Documentation Set are printed in green to help readers quickly find the Reference Pages on their bookshelf. This color-coding is reinforced with the use of the R icon on the spine of the books. The Reference Pages Documentation Set contains the following manuals:

- *Reference Pages Section 1*

  This section describes user commands that are available to everyone who uses the Digital UNIX system. In printed format, this section is divided into four volumes.

- *Reference Pages Section 2*

  This section defines system calls (entries into the Digital 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. This section is for programmers.

- *Reference Pages Section 3*

  This section describes the routines available in Digital UNIX programming libraries, including the C library, Motif library, and X library. This section is for programmers. In printed format, this section is divided into six volumes.

- *Reference Pages Section 4*

  This section 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 format, this section is divided into two volumes.

- *Reference Pages Section 5*

  This section 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.

- *Reference Pages Section 7*

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

- *Reference Pages Section 8 and 1m*

  This section describes commands for system operation and maintenance. It is for system administrators. In printed format, this section is divided into two volumes.

5 Ordering Digital UNIX Documentation

You can order Digital UNIX documentation electronically, at the Electronic Store, or by telephone or direct mail. Refer to the ‘‘How to Order Additional Documentation’’ page at the back of this manual for information in using these methods to order documentation.

Table 1 details the documentation kits that are available.

Table 1: Digital UNIX Documentation Kit Order Numbers

Documentation Kit	Order Number
Full Documentation Set	QA-MT4AP-GZ
End User Documentation Kit	QA-MT4AR-GZ
Startup Documentation Kit	QA-MT4AS-GZ
General User Documentation Kit	QA-MT4AT-GZ
System and Network Management Documentation Kit	QA-MT4AU-GZ
Developer’s Documentation Kit	QA-MT5AE-GZ
General Programming Documentation Kit	QA-MT5AF-GZ
Windows Programming Documentation Kit	QA-MT5AG-GZ
Writing Device Drivers Documentation Kit	QA-MT5AH-GZ
Reference Pages Documentation Set	QA-MT4AG-GZ

You can also order the quick reference cards, the *Documentation Map,* or individual manuals.

When you order quick reference cards or the *Documentation Map,* you receive a package of 15 copies of the card or poster. Table 2 lists the order numbers for these 15-item packages.

Table 2: Reference Card and Poster Documentation Packages

Package Name	Order Number
Quick Reference Card Package	AI-Q6PBB-TE
Update Installation Quick Reference Card Package	AI-Q6PCC-TE
Documentation Map Package	AI-Q6PDC-TE

Table 3 lists the order number for each manual in the Digital UNIX Version 4.0 documentation set. Individual manuals listed in this table are revised periodically. If you have a version of the Digital UNIX system other than Version 4.0, the manual numbers listed in this table might not be correct for your system. The best way to ensure that you are ordering the correct manuals for your operating system is to telephone Technical Support before you place your order. (The telephone number for Technical Support is listed on the ‘‘How to Order Additional Documentation’’ page.)

Table 3: Individual Manual Order Numbers

Title	Order Number
Assembly Language Programmer's Guide	AA-PS31D-TE
Asynchronous Transfer Mode	AA-QDP5C-TE
CDE Companion	AA-QTLPA-TE
Common Desktop Environment: Advanced User's and System Administrator's Guide	AA-QTLQA-TE
Common Desktop Environment: Application Builder User's Guide	AA-QTM2A-TE
Common Desktop Environment: Desktop KornShell User's Guide	AA-QTL0A-TE
Common Desktop Environment: Help System Author's and Programmer's Guide	AA-QTLXA-TE
Common Desktop Environment: Internationalization Programmer's Guide	AA-QTM2A-TE
Common Desktop Environment: Product Glossary	AA-QTM4A-TE
Common Desktop Environment: Programmer's Guide	AA-QTLXA-TE
Common Desktop Environment: Programmer's Overview	AA-QTLWA-TE
Common Desktop Environment: Style Guide and Certification Checklist	AA-QTM3A-TE
Common Desktop Environment: ToolTalk Messaging Overview	AA-QTLZA-TE
Common Desktop Environment: User's Guide	AA-QTLNA-TE
Command and Shell User's Guide	AA-PS2HD-TE
Calling Standard for Alpha Systems	AA-PY8AC-TE
DEC C Language Reference Manual	AA-QTLTA-TE
DECevent Translation and Reporting Utility	AA-QTLSA-TE
DECwindows User's Guide	AA-Q917B-TE
Developing Applications for the Display PostScript System	AA-Q15WB-TE
Digital Portable Mathematics Library	AA-PUBXC-TK
Documentation Map	AV-QTLFA-TE
Documentation Overview, Glossary, and Master Index	AA-QTM6A-TE
Guide to DECthreads	AA-Q2DPC-TK
Guide to Prestoserve	AA-PQT0D-TE
Guide to Realtime Programming	AA-PS33D-TE
Installation Guide	AA-QTLGA-TE
Kernel Debugging	AA-PS2TE-TE
Ladebug Debugger Manual	AA-PZ7EE-TE

Table 3: (continued)

Title	Order Number
Logical Storage Manager	AA-Q3NCE-TE
Network Administration	AA-PS2SC-TE
Network Programmer’s Guide	AA-PS2WD-TE
OSF/Motif Programmer’s Guide[a]	AA-PY8BB-TE
OSF/Motif Style Guide[a]	AA-PY8CA-TE
Performance Manager	AA-QTLRA-TE
PostScript Language Reference Manual[b]	AA-HL86A-TE
Programmer’s Guide	AA-PS30D-TE
Programmer’s Guide: STREAMS[a]	AA-Q15XA-TE
Programming Support Tools	AA-PS32D-TE
Programming with ONC RPC	AA-Q0R5B-TE
Quick Reference Card	QV-QTLJA-TE
Read This First Letter	AV-QTLHA-TE
Release Notes	AA-QTLMA-TE
Reference Pages Section 1 (Volume 1)	AA-PS2PD-TE
Reference Pages Section 1 (Volume 2)	AA-PS2QD-TE
Reference Pages Section 1 (Volume 3)	AA-PWZQD-TE
Reference Pages Section 1 (Volume 4)	AA-QUPQA-TE
Reference Pages Section 2	AA-PS36D-TE
Reference Pages Section 3 (Volume 1)	AA-PS37D-TE
Reference Pages Section 3 (Volume 2)	AA-PS38D-TE
Reference Pages Section 3 (Volume 3)	AA-PS39D-TE
Reference Pages Section 3 (Volume 4)	AA-PS3AD-TE
Reference Pages Section 3 (Volume 5)	AA-Q2UAC-TE
Reference Pages Section 3 (Volume 6)	AA-QUPRA-TE
Reference Pages Section 4 (Volume 1)	AA-PS2UD-TE
Reference Pages Section 4 (Volume 2)	AA-QUPSA-TE
Reference Pages Section 5	AA-Q7YXD-TE
Reference Pages Section 7	AA-QUPTA-TE
Reference Pages Section 8 (Volume 1)	AA-PS2VD-TE
Reference Pages Section 8 (Volume 2)	AA-Q7YYB-TE
Security	AA-Q0R2D-TE

Sharing Software on a Local Area Network	AA-PS3LE-TE
Software License Management	AA-PS2ED-TE
System Administration	AA-PS2RD-TE
System Tuning and Performance Management	AA-Q0R3E-TE
Technical Overview	AA-QTLLA-TE
Update Installation Quick Reference Card	AV-QTLKA-TE
Writing EISA and ISA Bus Device Drivers	AA-Q0R6C-TE
Writing Device Drivers for the SCSI/CAM Architecture Interfaces	AA-PS3GD-TE
Writing Device Drivers: Advanced Topics	AA-Q7RPB-TE
Writing Device Drivers: Reference	AA-PUBWD-TE
Writing Device Drivers: Tutorial	AA-PUBVD-TE
Writing PCI Bus Device Drivers	AA-Q7RQC-TE
Writing Software for the International Market	AA-Q0R4C-TE
Writing TURBOchannel Device Drivers	AA-PS3HD-TE
Writing VMEbus Device Drivers	AA-Q0R7E-TE
X Window System Environment	AA-Q7RNB-TE
X Window System Administrator’s Guide[c]	AA-PS2NA-TE

Table Notes:

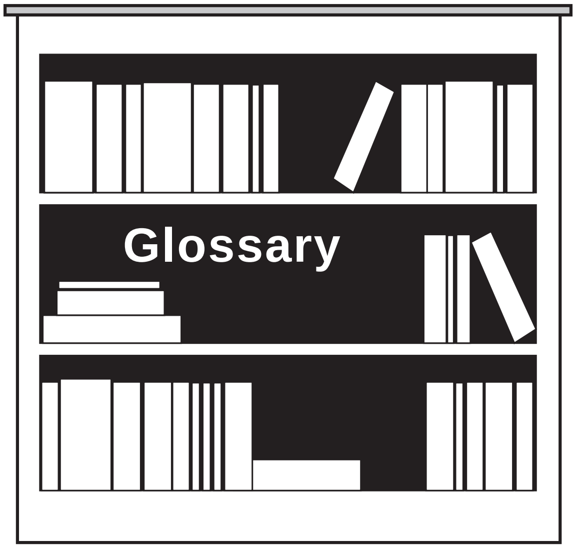
a. Published by Prentice-Hall

b. Published by Addison-Wesley

c. Published by O’Reilly and Associates

Part II: Glossary of Terms

This part contains a glossary of computer terms, particularly those related to the Digital UNIX operating system.



Glossary

/

See **root (2).**

. (dot)

A shorthand expression representing the user’s 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. Contrast with **relative pathname.**

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.

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** and **man.**

archive

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

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.

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 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**

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, a powerful pattern matching utility for textual data manipulation. An awk program is a sequence of patterns and corresponding actions that are carried out when a pattern is read. The `awk` program is a more powerful tool for 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 issued with an ampersand (&) character following it. For example, to run the calculator program in background, a user would issue the command `dxcalc &.` As a result, the calculator would be invoked in one window, while the command line on which the `dxcalc` command was issued would be ready to accept new commands. Contrast with **foreground process.**

Berkeley Internet Name Domain

See **BIND.**

Berkeley Software Distribution

See **BSD.**

Berkeley UNIX

See **BSD.**

/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, path,** and **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. Contrast with **character device.**

block device switch table
The method used by the Digital 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** and **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. Compare to **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)

UNIX software release of the Computer System Research Group of the University of California at Berkeley –- the basis for some features of the Digital 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**.

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. Contrast with **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. Contrast with **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.

CAM Control Block

See **CCB.**

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. Contrast with **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 Digital UNIX operating system are case sensitive. Contrast with **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.**

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).

CDE (Common Desktop Environment)

A graphical user interface for interacting with the Digital 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.

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

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

character device switch table
The method used by the Digital 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. Compare to **STREAMS.**

Command Description Block
See **CDB.**

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 Desktop Environment
See **CDE.**

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 intermediate binary file referred to as an object file).

compile time
Refers to actions that are taken by a compiler during the compilation of a program. Contrast with **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 Digital 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. Contrast with **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. Compare to **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**.

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 defacto 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**.

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 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.

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 File System

See **DFS.**

DMS (Dataless Management Services (DMS)

A service provided by Digital whereby 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**.

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 **communication domain, domain name system, Internet 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. Contrast with **up time.**

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 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.

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 Digital 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. Contrast with **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**.

EPG (External Gateway Protocol)

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

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.

escape

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

/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.

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. See also **out-of-band data**.

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.**

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** and **EOF.**

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 (1).**

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. Contrast with **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. Contrast with **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 command

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. See also **login group**. (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.

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 action 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 Adaptor)

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.

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.**

$home

A process variable containing the absolute pathname of the user’s home directory. See also **$HOME.**

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.** Contrast with **working directory.**

host

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

Host Bus Adapter

See **HBA**.

host name

The name given to a computer on the network.

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**.

#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"
```

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**.

internet

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

Internet

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. The Internet is also known as the TCP/IP Internet.

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 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**.

Internet Protocol

See **IP.**

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.

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.

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.

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.

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 Digital 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.

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 Digital protocol that supports communications between host computer systems and terminal servers with terminals, PCs, printers, modems, and other devices over LANs. See also **LAN**.

layered product

An optional software product designed to be installed as an added feature of the Digital 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.

lexical analyzer

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

Lexical Analyzer Generator

See **lex.**

line editor

An interactive or noninteractive text editor that works on one line of text at a time. Contrast with **full-screen editor.**

link

A directory entry referring to a file. See also **hard link** and **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 transport

See **LAT.**

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 Digital 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 Digital 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.

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

man 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**.

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. Contrast with **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.

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

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 **OSI**.

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 Digital 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.

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.

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

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 environment variable containing the user’s search path for commands. Directory names in the `$PATH` variable are separated with colons. See also **$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.

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.

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, also 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.

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. Compare to **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. Contrast with **SVID.**

POSIX shell

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

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. Contrast with **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 **job, daemon**.

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. Contrast with **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.

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. Often called manual pages or manpages, this online documentation is viewed by using the `man` and `xman` commands. For example, to view documentation on the `mkdir` (make directory) command, a user would type `man mkdir.`

When the reference page subset is installed, the reference pages are located in the `/usr/share/man` and `/usr/dt/share/man` directories. By default, the `man` and `xman` commands search both of these directories for reference pages.

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. Contrast with **absolute pathname.**

restricted shell

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

Revision Control System

See **RCS.**

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 operating system software 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.

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 clients.

roff

A family of text formatting programs designed to prepare output for different types of display devices.

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 (2).**

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 (1).**

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. Contrast with **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.**

SCP (software 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.**

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.**

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 **subsets.**

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 **C shell, Korn shell, POSIX shell,** and **Bourne shell**.

shell variable

See **process variable** and **environment variable.**

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 Network Management Protocol

See **SNMP.**

SLIP (Serial Line Internet Protocol)

A transmission line protocol that encapsulates and transfers IP datagrams over asynchronous serial lines.

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 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.

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. Compare to **clist**. See also **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 Digital 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 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*.

SVID (System V Interface Definition)

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

SVVS (System V Verification Suite)

A program used to test adherence to the System V Interface Definition.

switch

Another name for an option. See **option (1)**.

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.**

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 (SVID)

See **SVID.**

System V Verification Suite

See **SVVS.**

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** and **Internet Protocol**.

$TERM

An environment variable containing the user's terminal type.

termcap database

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

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.

terminal session

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

terminated job

A process that is permanently stopped. Contrast with **suspended job.**

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. Compare to **breakpoint.**

Transmission Control Protocol

See **TCP.**

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.

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, also UID

See **user ID.**

ULTRIX

One of two UNIX operating system products available from Digital Equipment Corporation. The ULTRIX operating system runs on VAX and RISC computers, whereas Digital 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.**

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. Contrast with **down time.**

upward compatible

Pertaining to that which is designed for use on small machines, but capable of running without change on larger machines.

User Datagram Protocol

See **UDP.**

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.

virus

A computer program designed to insinuate itself into other programs or files in a system and then to replicate itself through any available means (disk file, network, and so forth) into other similar computers, from which it can attack yet more systems. Viruses are designed with the object of damaging or destroying the ‘‘infected’’ programs or systems and are often programmed to become destructive at a specific time, such as the birthday of the virus’s programmer. Contrast with **worm.**

visual editor

See **full-screen editor** and **line editor.**

W

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.

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. Contrast with **home directory.**

worm

A computer program designed to insinuate itself into other programs or files in a system and then to replicate itself through any available means (disk file, network, and so forth) into other similar computers, from which it can attack yet more systems. Worms are designed with no serious intent to do damage, but they are harmful because they occupy resources intended for legitimate use. Contrast with **virus.**

WORM

Refers to a write-once, read-many-times device.

X

X/Open Transport Interface (XTI)
: 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.

XPT
: A layer of software that SCSI peripheral drivers use to originate the execution of CAM (Common Access Method) functions.

XTI
: See **X/Open Transport Interface**.

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 **parser**.

younger file
: For the `make` utility, a dependency file that has changed more recently than its target.

Part III: Master Index for the Documentation Set

This part contains the master index for the Digital UNIX documentation set.

Introduction to the Master Index

The master index helps you locate topics in the Digital UNIX documentation set by directing you to books and page numbers. Master index entries follow these conventions:

- Each master index page reference is preceded by an abbreviated manual title. For example, the following entry points to page 3–2 in the guide *Sharing Software on a Local Area Network*:

  Installation
  RIS software subsets, *SharingSW* 3–2

  Table 1 maps the abbreviated title for each manual to its complete book title.

- Page references for some figures, tables, and notes are indicated by a letter that appears at the end of the page reference. For example, an ‘‘f’’ appears at the end of references to figures and an ‘‘n’’ appears at the end of references to notes. The following example shows an index entry for a table:

  character class
  in REs, *ProgSupTools* 1–1t

- The format of page references varies according to the pagination method used in a particular book. For example, some books are numbered sequentially throughout the manual; others are numbered so that the page number includes a chapter number and a page number. The following example shows the types of page references contained in the master index:

  NFS *NetOverview* 3–2, *XWindowAdmin* 172
  and hosts database, *NetworkConfig* 7–4n

The following table lists the abbreviated title for each book:

Table 1: Book Title Abbreviations

Abbreviation	Book Title
AppBuilder	*Common Desktop Environment: Application Builder User's Guide*
AsmLangGde	*Assembly Language Programmer's Guide*
ATMProgGde	*Asynchronous Transfer Mode*
CallStandard	*Calling Standard for Alpha Systems*
CDEAdvUsr	*Common Desktop Environment: Advanced User's and System Administrator's Guide*
CDECompanion	*CDE Companion*
CDEHelp	*Common Desktop Environment: Help System Author's and Programmer's Guide*
CDE_I18N_Prog	*Common Desktop Environment: Internationalization Programmer's Guide*
CDEProgGuide	*Common Desktop Environment: Programmer's Guide*
CDEProgOverview	*Common Desktop Environment: Programmer's Overview*
CDEStyle	*Common Desktop Environment: Style Guide and Certification Checklist*
CDE_User	*Common Desktop Environment: User's Guide*
CommandShellGde	*Command and Shell User's Guide*
DECC	*DEC C Language Reference Manual*
DECevent	*DECevent Translation and Reporting Utility*
DECthreads	*Guide to DECthreads*
Desktopksh	*Common Desktop Environment: Desktop KornShell User's Guide*
DevDriverAdvTopics	*Writing Device Drivers: Advanced Topics*
DevDriverEisa	*Writing EISA and ISA Bus Device Drivers*
DevDriverPci	*Writing PCI Bus Device Drivers*
DevDriverReference	*Writing Device Drivers: Reference*
DevDriverTURBOchannel	*Writing TURBOchannel Device Drivers*
DevDriverTutorial	*Writing Device Drivers: Tutorial*
DevDriverVMEbus	*Writing VMEbus Device Drivers*
displayPS	*Developing Applications for the Display PostScript System*

Table 1: (continued)

Abbreviation	Book Title
DPML	*Digital Portable Mathematics Library*
DWUser	*DECwindows User's Guide*
InstallGuide	*Installation Guide*
International	*Writing Software for the International Market*
KernelDebug	*Kernel Debugging*
Ladebug	*Ladebug Debugger Manual*
LicenseManage	*Software License Management*
MotifProg	*OSF/Motif Programmer's Guide*
MotifStyle	*OSF/Motif Style Guide*
NetAdmin	*Network Administration*
NetProgGde	*Network Programmer's Guide*
OncRpcProgram	*Programming with ONC RPC*
OSInstall	*Installation Guide*
PerformManage	*Performance Manager*
Prestoserve	*Guide to Prestoserve*
ProgGde	*Programmer's Guide*
ProgSupTools	*Programming Support Tools*
PSLangRefMan	*PostScript Language Reference Manual*
Realtime	*Guide to Realtime Programming*
SCSIDevDrivers	*Writing Device Drivers for the SCSI/CAM Architecture Interfaces*
Security	*Security*
SharingSW	*Sharing Software on a Local Area Network*
StorageManager	*Logical Storage Manager*
STREAMSProg	*Programmer's Guide: STREAMS*
SystemAdmin	*System Administration*
SystemTune	*System Tuning and Performance Management*
TechOver	*Technical Overview*
ToolTalkMsg	*Common Desktop Environment: ToolTalk Messaging Overview*
XWinSysEnv	*X Window System Environment*

Index

Special Characters

! subcommand (ftp), *CommandShellGde* 12–8

*** wildcard character**, *CDEAdvUsr* 193

? subcommand (ftp), *CommandShellGde* 12–8

? wildcard character, *CDEAdvUsr* 193

A

abnormal_termination function, *ProgGde* 11–14

abstract for help volume, *CDEHelp* 108

accelerator keys

CDE conventions and standards for, *CDEStyle* 148, 149, 150, 183

accept socket call

contrast to XTI t_accept function, *NetProgGde* 3–46

access control list, *Security* 5–1

See ACL

access to files, *CommandShellGde* 5–1

accessible symbols, *STREAMSProg* C–1

account management, *Security* 9–3

account subcommand (ftp), *CommandShellGde* 12–5

accountability, *Security* 1–3

accounting, *SystemAdmin* 12–1

accounts, *Security* 9–1

acctcms command, *SystemAdmin* 12–28

acctcom command, *SystemAdmin* 12–25

acctdisk command, *SystemAdmin* 12–34

acctdusg command, *SystemAdmin* 12–33

acctmerg command, *SystemAdmin* 12–40

acctprc1 command, *SystemAdmin* 12–29

acctprc2 command, *SystemAdmin* 12–30

ACL, *Security* 5–1

acos routine, *DPML* DPML–5, B–1

acosd routine, *DPML* DPML–5, B–1

acosh routine, *DPML* DPML–6, B–2

action file, *CDEAdvUsr* 142, 161

action icon, *CDE_User* 119, *CDEAdvUsr* 67, 131, 161, 162

action invocation

in CDE program development, *CDEProgOverview* 69, 70

actions

creating, *CDE_User* 126

creating manually, *CDEAdvUsr* 155

creating with Create Action, *CDEAdvUsr* 141

executing, *CDE_User* 96

in CDE application development, *CDEProgGuide* 109 to 111, 113, 115, 124

introduction to, *CDEAdvUsr* 127

active user

typical state transitions, *NetProgGde* 3–21f

activity license, *LicenseManage* 1–5

acucap file, *NetAdmin* 4–22

Ada language debugging support, *Ladebug* 12-1

addgroup utility, *SystemAdmin* 9–18

Address Resolution Protocol

See ARP

addressing in DLPI, *NetProgGde* 2–7

adduser utility, *SystemAdmin* 9–5

administrators, introduction, *Security* 6–1

Advanced File System

See AdvFS

advanced sockets topics, *NetProgGde* 4–41 to 4–55

AdvFS, *SystemAdmin* 7–1, *TechOver* 4–8

choosing during installation, *InstallGuide* 5–19

file system quotas, *SystemAdmin* 7–11

fileset quotas, *SystemAdmin* 7–11

managing quotas, *SystemAdmin* 7–11

setting up, *SystemAdmin* 7–9

tuning, *SystemTune* 3–28 to 3–30

aliases, execution

in CDE help, *CDEHelp* 72

aliases, predefined, *Ladebug* 7-5, REF-17

_align storage class modifier, *ProgGde* 2–8

alignment

defined, *DevDriverAdvTopics* 2–3

implementation on Alpha CPU hardware platforms, *DevDriverAdvTopics* 2–3

placement of data item in memory, *DevDriverAdvTopics* 2–3

allocating resources for DMA data transfers

with dma_map_alloc and dma_map_load, *DevDriverVMEbus* 5–1

allocating system resources for DMA

by calling dma_map_alloc kernel interface, *DevDriverTutorial* 18–29

allocating VMEbus address space

See allocating resources for DMA data transfers

Alpha 64-bit architecture

performance considerations, *SystemTune* 1–1

alt.soft-sys.tooltalk news group, *ToolTalkMsg* xviii

animation, *CDEAdvUsr* 234

ANSI

standards and application development considerations, *ProgGde* 1–2

ANSI C

and CDE compliance, *CDEProgOverview* 25, 37

ANSI C conformance summary, *DECC* B-1

ANSI C Standard, *DECC* XIV

app-defaults file, *CDEAdvUsr* 256, 279

and application font names, *CDEProgGuide* 20

for dtksh, *Desktopksh* 14

Application Builder, *AppBuilder* 1 to 150

introduction to CDE, *CDEProgOverview* 34

application design principles

CDE conventions and standards for, *CDEStyle* 73 to 95

application development

in CDE, *CDEProgOverview* 24, 25, 32 to 35

phases of, *ProgGde* 1–1

application group, *CDE_User* 119 to 120, *CDEAdvUsr* 40, 41, 45, 46, 67 to 73, 77, 82

personal, *CDE_User* 123

application icon, *CDEAdvUsr* 67, 131, 134, 142, 158, 161

application integration

in CDE, *CDEProgOverview* 41 to 76

application integration (cont.)
using ToolTalk, *ToolTalkMsg* 20
Application Manager, *CDEAdvUsr* 28, 39, 40, 41, 43, 48, *CDECompanion* 3–2
application performance, *SystemAdmin* 3–22
application programming interface
sockets, *NetProgGde* 1–1
STREAMS, *NetProgGde* 1–1
WPI, *International* A–1
XTI, *NetProgGde* 1–1
application programming interfaces
DLI, *TechOver* 3–18
DLPI, *TechOver* 3–18
sockets, *TechOver* 3–18
STREAMS, *TechOver* 3–18
XTI, *TechOver* 3–18
application programs
compiling and linking in System V habitat, *ProgGde* B–1 to B–3
porting, *ProgGde* 6–12
porting to XTI, *NetProgGde* 3–41
application registration in CDE, *CDEHelp* 244
application resources
administering, *CDEAdvUsr* 255
application search path, *CDEAdvUsr* 40, 47, 48, 116 to 118, 125
application servers, *CDEAdvUsr* 47, 57, 87, 94, 99 to 101
application window management
CDE conventions and standards for, *CDEStyle* 65, 70, 190, 191
applications, *CDEAdvUsr* 25, 28, 32, 40 to 44, 48, 55, 60, 63, 106
accessing from front panel, *CDECompanion* 3–1
applications (cont.)
adding and administering, *CDEAdvUsr* 39
creating Motif applications with dtksh, *Desktopksh* 11 to 18
integrated into CDE, *CDECompanion* 5–4
registering, *CDEAdvUsr* 53
running from the desktop, *CDE_User* 117
architecture
See VMEbus architectures
of RIS, *SharingSW* 2–1 to 2–4
archiving services, *SystemAdmin* 11–1
arguments for actions, *CDEAdvUsr* 133, 168, 170 to 175
ARP, *TechOver* 3–15
arp command, *NetAdmin* 14–3
as command, *ProgGde* 2–4
ASCII equivalence table, *DECC* C-1
ascii subcommand (ftp), *CommandShellGde* 12–5
asin routine, *DPML* DPML–7, B–2
asind routine, *DPML* DPML–7, B–2
asinh routine, *DPML* DPML–8, B–2
assembler directives, *AsmLangGde* 5–1 to 5–15
assert_wait_mesg kernel interface
explanation of code fragment, *DevDriverAdvTopics* 7–23, 7–9
assigning program numbers, *OncRpcProgram* 1–5
asynchronous I/O, *STREAMSProg* 2–7
realtime, *Realtime* 7–1
Asynchronous Transfer Mode
See ATM
at command, *SystemAdmin* 3–3
atan routine, *DPML* DPML–9, B–3
atan2 routine, *DPML* DPML–10, B–3

atand routine, *DPML* DPML–9, B–3
atand2 routine, *DPML* DPML–10, B–3
atanh routine, *DPML* DPML–11, B–4
ATM, *TechOver* 3–16
  cells, *ATMProgGde* 3–3
  CMM routines, *ATMProgGde* A–1
  configuring the software, *ATMProgGde* 2–1 to 2–5
  connection management, *ATMProgGde* 7–1
  convergence module interface, *ATMProgGde* 6–1
  device driver interface, *ATMProgGde* 4–1
  error codes, *ATMProgGde* 3–2
  flow control, *ATMProgGde* 10–1
  global data structures, *ATMProgGde* 3–14
  header files, *ATMProgGde* 3–1
  Module Management Interface, *ATMProgGde* 8–1
  programming examples, *ATMProgGde* B–1
  queuing guidelines, *ATMProgGde* 9–1
  signaling
    module
      interface, *ATMProgGde* 5–1
  tuning, *ATMProgGde* 11–1
ATM architecture
  overview, *ATMProgGde* 1–1 to 1–5
ATM subsystem
  interfaces, *ATMProgGde* 1–4
atmarp command, *ATMProgGde* 2–4
atmconfig command, *ATMProgGde* 2–5
  and PVCs, *ATMProgGde* 3–7
ATOM, *TechOver* 13–6
Atom tools, *ProgGde* 9–1
atomicity
  defined, *DevDriverAdvTopics* 2–2
  type of serialization, *DevDriverAdvTopics* 2–2
attached process
  debugging, *Ladebug* 5-2
attaching in App Builder, *AppBuilder* 50 to 54, 89, 99
attachments
  adding to dtmail message, *CDE_User* 166
  to documents, *CDEStyle* 33 to 37, 83
audio/visual capabilities
  installing, *InstallGuide* D–14
audit subsystem, *Security* 10–1
authentication
  of client, *OncRpcProgram* 2–20
authorization
  X, *CDEAdvUsr* 93
auto repeat, *DWUser* 7–7
autoconfiguration
  declarations and definitions section, *DevDriverTURBOchannel* 6–7 to 6–8
  defined, *DevDriverTutorial* 17–1
  EISA/ISA bus device drivers, *DevDriverEisa* 6–1
  support section, *DevDriverTURBOchannel* 6–15 to 6–23
Autoconfiguration Support Section
  description of code example for /dev/dmaex driver, *DevDriverVMEbus* 6–8 to 6–14
automount daemon, *NetAdmin* 8–2
availability license, *LicenseManage* 1–4
awakening kernel thread
  See unblocking kernel thread
awk utility, *ProgSupTools* 2–1 to 2–21
AXPvme Single-Board Computers, *DevDriverVMEbus* D–1

B

- **b_resid member**
  - use as argument with copyin kernel interface, *DevDriverTutorial* 18–11
- **backdrops**, *CDEAdvUsr* 242, 245, 246
- **background process**, *CommandShellGde* 6–6
- **backup**
  - performing with NetWorker SingleServer, *InstallGuide* D–10
  - procedures, *SystemAdmin* 11–1
- **backward compatibility**
  - shared libraries, *ProgGde* 4–18
- **BADADDR interface**
  - explanation of code fragment, *DevDriverTutorial* 7–17
- **base operating system software subsets**, *InstallGuide* C–1
- **basic regular expression**, *ProgSupTools* 1–2
- **bcheckrc script**, *SystemAdmin* 3–2, 3–6
- **bcopy kernel interface**
  - explanation of code fragment, *DevDriverTutorial* 18–9, 6–18
- **Berkeley Internet Name Domain**
  - *See* BIND naming service
- **BGP**, *TechOver* 3–6
- **big endian**
  - *See* byte ordering
  - byte ordering, *AsmLangGde* 1–2
- **binary incompatibility**
  - shared libraries, *ProgGde* 4–18
- **binary subcommand (ftp)**, *CommandShellGde* 12–5
- **BIND**
  - configuring, *NetAdmin* 6–7 to 6–11
  - testing, *NetAdmin* 15–1 to 15–10
  - using, *NetAdmin* 11–4
- **bind event**, *NetProgGde* 3–15
- **BIND naming service**
  - defined, *TechOver* 3–28
  - introduction, *TechOver* 3–28
- **bind socket call**
  - contrast to XTI t_bind function, *NetProgGde* 3–46
- **binding**
  - lazy binding, *AsmLangGde* 9–15
- **binlogd daemon**, *SystemAdmin* 13–23, 13–32
- **bit fields**, *ProgGde* 6–12
- **bit gravity**, *displayPS* 4–4
- **bit-to-text translation**, *DECevent* 3–1
- **bitmaps**, *CDEAdvUsr* 202 to 204
- **block**, *DECC* 2-1
- **block device**
  - comparison with character device, *DevDriverTutorial* 16–5
  - defined, *DevDriverTutorial* 2–6
- **block device driver**
  - introductory discussion and examples of, *DevDriverTutorial* 1–2
- **Bookreader**, *DWUser* 5–10
- **bootable tape**, *SystemAdmin* 11–3, *TechOver* 13–8
- **booting**
  - genvmunix, *SystemAdmin* 2–3
  - overview, *SystemAdmin* 2–1
- **booting processors**, *InstallGuide* 4–1
- **booting single-board computers**, *InstallGuide* 4–1
- **bootp**, *SharingSW* 6–1
- **bootpd**, *SharingSW* 6–1, *TechOver* 3–24
- **bootptab**, *SharingSW* 6–1
- **Border Gateway Protocol**
  - *See* BGP

borders around objects
implementing, *AppBuilder* 91, 93, 137
bounds file
description of, *KernelDebug* 4–10
Bourne shell, *CommandShellGde* 7–1, 8–1
breaking remote cu connection (UUCP), *CommandShellGde* 14–8
breaking remote tip connection (UUCP), *CommandShellGde* 14–14
breakpoints, *Ladebug* 4-9
bridging drivers
BSD drivers to STREAMS protocol stacks, *NetProgGde* 7–11
STREAMS drivers to sockets protocol stacks, *NetProgGde* 7–2
broadcast RPC, *OncRpcProgram* 2–20, 3–17
browser help volume, *CDEHelp* 12, 94, 95, 97
browser (Module Browser), *AppBuilder* 26, 128
BSD drivers
bridging to STREAMS protocol stacks, *NetProgGde* 7–11
BSD sockets, *NetProgGde* 4–37
btcreate command, *SystemAdmin* 11–3
btextract command, *SystemAdmin* 11–3
buf structure
associated include file, *DevDriverReference* 3–4
defined, *DevDriverTutorial* 9–3
formal description, *DevDriverTutorial* 9–3
list of member names and data types, *DevDriverReference* 3–4
buffer cache
management of, *DevDriverTutorial* 9–8
buffering, *displayPS* 4–2
building applications with App Builder, *AppBuilder* 107 to 112
building programs
See lex program
See make utility
See yacc program
building the kernel, *InstallGuide* 5–31
built-in functions
pragma counterparts, *ProgGde* 3–5
bus
See also OPENbus
defined, *DevDriverTutorial* 1–9
relationship to device driver, *DevDriverTutorial* 1–9, 16–3
bus physical address
defined, *DevDriverTutorial* 2–17
description of address type that device drivers use, *DevDriverTutorial* 2–17
bus structure
associated include file, *DevDriverReference* 3–8
defined, *DevDriverTutorial* 7–24
list of member names and data types, *DevDriverReference* 3–8
bus support subsystem
accessing device registers of bus adapter, *DevDriverTutorial* 16–8
defined, *DevDriverTutorial* 15–4
buses, supported, *TechOver* 6–3
EISA, *TechOver* 6–3
Futurebus+, *TechOver* 6–3
ISA, *TechOver* 6–3
PCI, *TechOver* 6–3
SCSI, *TechOver* 6–3
TURBOchannel, *TechOver* 6–3
VME, *TechOver* 6–3

buses, supported (cont.)
XMI, *TechOver* 6–3
busy wait time
defined, *DevDriverAdvTopics* 2–13
button binding, *CDEAdvUsr* 249 to 252
buttons, *CDE_User* 19
using push buttons, *DWUser* 3–13
bye subcommand (ftp), *CommandShellGde* 12–5
byte operations
See reading data from a device register
See writing data from a device register
byte ordering, *ProgGde* 2–5
big endian, *AsmLangGde* 1–2
little endian, *AsmLangGde* 1–2
byte string
copying bcopy kernel interface, *DevDriverTutorial* 18–8, 6–18
bzero kernel interface
explanation of code fragment, *DevDriverTutorial* 18–10

C

C compiler, *DECC* XIV, *TechOver* 7–1
C function, *DECC* 5-1
C language syntax summary, *DECC* A-1
C language, program checking
external names, *ProgGde* 6–12
function definitions, *ProgGde* 6–6
initializing variables, *ProgGde* 6–10
migration, *ProgGde* 6–10
portability, *ProgGde* 6–11
structure, union, *ProgGde* 6–5
use of characters, *ProgGde* 6–11
use of uninitialized variables, *ProgGde* 6–10
C lexicon, *DECC* XIV
C naming conventions, *CDEProgGuide* 141
C shell, *CommandShellGde* 7–1, 8–1
C++
in CDE application development, *CDEProgOverview* 25
C++ language debugging support, *Ladebug* 10-1
cabs routine, *DPML* DPML–13, B–4
Calculator, *DWUser* 5–10
Calendar, *DWUser* 5–21, 5–26
calendar
converting to CDE format, *CDECompanion* 10–6 to 10–7
calendar (dtcm)
and CDE application development, *CDEProgOverview* 74
integrating CDE applications with, *CDEProgGuide* 139 to 160
using, *CDE_User* 215
call chain, *CallStandard* 7–3
call graphs
gprof tool, *ProgGde* 8–10
call stack, *Ladebug* 4-27
callback procedure
and RPC, *OncRpcProgram* 3–32
callbacks
and CDE application development, *CDEProgGuide* 77, 85, 91, 107
in CDE help, *CDEHelp* 212, 221, 231, 232
using in dtksh, *Desktopksh* 20 to 22
caller process, *OncRpcProgram* 1–1
CAM
common structures and routines, *SCSIDevDrivers* 1–6
configuration driver structures and routines, *SCSIDevDrivers* 1–7

CAM (cont.)
generic structures and routines, *SCSIDevDrivers* 1–6
SCSI CDROM/AUDIO device structures and commands, *SCSIDevDrivers* 1–6
SCSI disk device structures and routines, *SCSIDevDrivers* 1–6
SCSI tape device structures and routines, *SCSIDevDrivers* 1–6
SCSI/CAM peripheral drivers, *SCSIDevDrivers* 1–5
SCSI/CAM special I/O interface, *SCSIDevDrivers* 1–6
SIM SCSI Interface layer, *SCSIDevDrivers* 1–7
tuning, *SystemTune* 3–30
User Agent driver structures and routines, *SCSIDevDrivers* 1–4
XPT transport layer, *SCSIDevDrivers* 1–7

CAM SIM callback handling
description, *SCSIDevDrivers* 8–1

CAM SIM routines
introduction, *SCSIDevDrivers* 8–2

CAM SIM SCSI I/O CCB priority
description, *SCSIDevDrivers* 8–3

CAM SIM SCSI I/O CCB reordering
description, *SCSIDevDrivers* 8–4

CAM User Agent driver
error handling, *SCSIDevDrivers* 2–2
introduction, *SCSIDevDrivers* 2–1

CAM XPT routines
introduction, *SCSIDevDrivers* 7–1

cam_data.c
converting, *SystemAdmin* 5–1

Cardfiler, *DWUser* 5–27, 5–29

cascading menus
in CDE, *CDEProgGuide* 69, 89

case-sensitive symbolic variable names in debugging, *Ladebug* 12-2

cb device driver
source listing, *DevDriverTURBOchannel* B–1
TURBOchannel device driver example, *DevDriverTURBOchannel* 6–1

cb_configure interface
configuration, *DevDriverTURBOchannel* 6–27 to 6–30
querying, *DevDriverTURBOchannel* 6–37
setting up, *DevDriverTURBOchannel* 6–25 to 6–26
unconfiguration, *DevDriverTURBOchannel* 6–35 to 6–36

cb_ctlr_unattach interface, *DevDriverTURBOchannel* 6–22 to 6–23

cbattach interface, *DevDriverTURBOchannel* 6–20 to 6–21

cbclose interface, *DevDriverTURBOchannel* 6–41

cbincled interface, *DevDriverTURBOchannel* 6–75 to 6–76

cbintr interface, *DevDriverTURBOchannel* 6–77 to 6–78

cbioctl interface
incrementing the lights, *DevDriverTURBOchannel* 6–68
performing an interrupt test, *DevDriverTURBOchannel* 6–70 to 6–72
returning a ROM word, *DevDriverTURBOchannel* 6–73 to 6–74

cbioctl interface (cont.)
setting the I/O mode, *DevDriverTURBOchannel* 6–69
setting up, *DevDriverTURBOchannel* 6–66 to 6–67
stopping incrementing the lights, *DevDriverTURBOchannel* 6–73 to 6–74
updating the CSR, *DevDriverTURBOchannel* 6–73 to 6–74
cbminphys interface, *DevDriverTURBOchannel* 6–51
cbopen interface, *DevDriverTURBOchannel* 6–39 to 6–40
cbprobe interface, *DevDriverTURBOchannel* 6–16 to 6–19
cbread interface, *DevDriverTURBOchannel* 6–43 to 6–46
cbreg.h file, *DevDriverTURBOchannel* 6–3 to 6–5
cbrt routine, *DPML* DPML–14, B–4
cbstart interface, *DevDriverTURBOchannel* 6–63 to 6–65
cbstrategy interface
converting 32-bit physical address, *DevDriverTURBOchannel* 6–58 to 6–59
converting buffer virtual address, *DevDriverTURBOchannel* 6–57
initializing buf structure for transfer, *DevDriverTURBOchannel* 6–54
setting up, *DevDriverTURBOchannel* 6–52 to 6–53
starting I/O and checking for timeouts, *DevDriverTURBOchannel* 6–60 to 6–61

cbstrategy interface (cont.)
testing low-order 2 bits and using internal buffer, *DevDriverTURBOchannel* 6–55 to 6–56
cbwrite interface, *DevDriverTURBOchannel* 6–47 to 6–50
cc command
use by other compiler commands, *ProgGde* 2–19
using in System V habitat, *ProgGde* B–1
ccos routine, *DPML* DPML–15, B–4
cd subcommand (ftp), *CommandShellGde* 12–7
CD-ROM File System
See CDFS
CDA Viewer, *DWUser* 5–30, 5–36
CDE, *TechOver* 8–1
components of, *CDECompanion* 1–1
CDE applications
basic integration, *CDEProgGuide* 3 to 5, 13
integrating into the CDE desktop, *CDEProgOverview* 24, 32, 45 to 61, 63 to 77
recommended integration, *CDEProgGuide* 17 to 56
CDE command line interface, *SystemAdmin* 1–7
CDE (Common Desktop Environment)
introduction to, *CDE_User* 1
CDE development environment, *CDEProgOverview* 23
CDE graphical interface, *SystemAdmin* 1–7
CDE Motif, *CDEProgOverview* 12, 13, 80 to 82
CDE widgets, *CDEProgOverview* 13, 64 to 67
CDFS, *SystemAdmin* 6–2, *TechOver* 4–7

cdiv routine, *DPML* DPML–16, B–5
cdup subcommand (ftp), *CommandShellGde* 12–7
ceil routine, *DPML* DPML–17, B–5
cells
- defined, *ATMProgGde* 3–3
- OAM, *ATMProgGde* 4–2
- receiving, *ATMProgGde* 4–2

central processing unit
- defined, *DevDriverTutorial* 16–2
- interrupts generated on, *DevDriverTutorial* 16–7
- relationship to device driver, *DevDriverTutorial* 16–2

centralized account management, *Security* 9–3
certification checklist for CDE and Motif applications, *CDEStyle* 145 to 255
cexp routine, *DPML* DPML–18, B–5
cfg_attr_t structure
- associated include file, *DevDriverReference* 3–12
- definition of members, *DevDriverReference* 3–12
- description of, *ProgGde* C–5

cfg_subsys_attr_t structure
- associated include file, *DevDriverReference* 3–16
- definition of members, *DevDriverReference* 3–16
- description of, *ProgGde* C–7

cfgmgr daemon
- defined, *DevDriverTutorial* 12–15

changing
- identity, *CommandShellGde* 5–17

changing identity, *CommandShellGde* 5–17
changing permissions, *CommandShellGde* 5–3
CHAP
- use with PPP, *NetAdmin* 4–13

character device
- defined, *DevDriverTutorial* 16–6
- discussion of, *DevDriverTutorial* 16–6

character device driver
- defined, *DevDriverTutorial* 1–2
- introductory discussion and examples of, *DevDriverTutorial* 1–2

character set, *DECC* 1-2
character sets, *International* 1–2
- types of coded, *International* 2–2

character-cell applications
- running, *DWUser* 2–3

characters
- creating and using user-defined, *International* 6–23
- use in a C program, *ProgGde* 6–11

characters, composing, *DWUser* B–1
check boxes, *CDE_User* 19
check buttons
- CDE conventions and standards for, *CDEStyle* 8, 234

CI, *SystemAdmin* C–1, *TechOver* 6–13
- configuration, *SystemAdmin* C–1

circuits
- ATM types, *ATMProgGde* 3–13

client, *CDEAdvUsr* 86, 93, 233
- authentication, *OncRpcProgram* 2–20
- description of for RIS, *SharingSW* 1–2
- programming with rpcgen, *OncRpcProgram* 2–19

client handle used by rpcgen, *OncRpcProgram* 2–7

Client Library
described, *displayPS* 2–2
header files, *displayPS* 5–1
naming conventions
See Naming conventions, Client Library
Client Library routines, *displayPS* 5–4 to 5–13
client system
configuring, *NetAdmin* 7–17, 8–8, 11–13
configuring for dial-in access, *NetAdmin* 4–20
client systems
sharing software from servers, *SharingSW* 1–2
client/server interaction, *NetProgGde* 4–8
client/server relationships, *DWUser* 1–1
clientsync operator, *displayPS* 6–3
clipping path, *displayPS* 4–4
Clock, *DWUser* 5–37, 5–40
clock
realtime, *Realtime* 6–1
clog routine, *DPML* DPML–19, B–5
clone, *STREAMSProg* 2–1, 2–8, 10–1
cloned installations, *InstallGuide* 1–16
close call, *STREAMSProg* 5–2, 5–6 to 5–7, 9–1
close socket call
contrast to XTI t_snddis function, *NetProgGde* 3–47
close subcommand (telnet), *CommandShellGde* 13–4
CLTS
See connectionless service in XTI
cmul routine, *DPML* DPML–20, B–6
cmx exerciser, *SystemAdmin* 13–21
COBOL lanauage debugging support, *Ladebug* 13-1
code generation with App Builder, *AppBuilder* 107
codesets, *CDE_I18N_Prog* 56, 59, 64 to 67, 69, 70, 72, *International* 2–2
conversion between, *CDE_I18N_Prog* 59 to 61
CoEd demo program, *ToolTalkMsg* 95 to 123
coexistence
of STREAMS and sockets, *NetProgGde* 7–1 to 7–12
collation, *CommandShellGde* C–3
color, *CDE_User* 59, 136, 137, 139, 259, 277
administering, *CDEAdvUsr* 255
CDE conventions and standards for, *CDEStyle* 39, 47, 49
Color Chooser in App Builder, *AppBuilder*
color systems, *DWUser* 7–1
colormap
See setrgbXactual operator
colors, choosing, *DWUser* 4–10
combo boxes
CDE conventions and standards for, *CDEStyle* 236
COMMAND action, *CDEAdvUsr* 156, 159, 166
command boxes
CDE conventions and standards for, *CDEStyle* 236
Command Line Login mode, *CDE_User* 37
command tagged queueing, *TechOver* 6–10
commands
continuing on the next line, *CommandShellGde* 11–16
run from remote host (UUCP), *CommandShellGde* 14–18
using, *CommandShellGde* 1–4

Common Access Method
See CAM

Common Access Method I/O Subsystem
CAM, *SystemAdmin* B–1

common C extensions, *DECC* D-1

common data structures
introduction to CAM, *SCSIDevDrivers* 3–1

common files
See header files

common macros
introduction to CAM, *SCSIDevDrivers* 3–6

common routines
introduction to CAM, *SCSIDevDrivers* 3–1, 3–8

common terminate CCB routine (CAM), *SCSIDevDrivers* 3–13, B–75

communicating with remote host (UUCP), *CommandShellGde* 14–1

communication between desktop clients, *CDE_I18N_Prog* 55

communication bridge
DLPI STREAMS pseudodriver, *TechOver* 3–21
ifnet STREAMS module, *NetProgGde* 1–7

communications system
See terminal communications system

Compact Disc File System
See CDFS

comparing files, *CommandShellGde* 3–24

compilation unit, *DECC* 1-2, 2-3

compiler system, *ProgGde* 2–1
driver programs, *ProgGde* 2–2

compiling applications
in CDE, *CDEProgOverview* 25, 35, 79
in System V habitat, *ProgGde* B–1 to B–3

completion handling, *ProgGde* 11–5

complex lock
asserting a read-only complex lock with lock_read, *DevDriverAdvTopics* 4–5
asserting a write complex lock with lock_write, *DevDriverAdvTopics* 4–8
declaring complex lock structure with lock_data_t, *DevDriverAdvTopics* 4–1
defined, *DevDriverAdvTopics* 2–8
description of, *DevDriverAdvTopics* 2–8
initializing a complex lock with lock_init, *DevDriverAdvTopics* 4–2
performing access operations on, *DevDriverAdvTopics* 4–4
releasing previously asserted complex lock with lock_done, *DevDriverAdvTopics* 4–10
terminating a complex lock with lock_terminate, *DevDriverAdvTopics* 4–21
trying to assert with lock_try_read, *DevDriverAdvTopics* 4–13
trying to assert with lock_try_write, *DevDriverAdvTopics* 4–13

complex lock structure
declared in code fragment, *DevDriverAdvTopics* 4–2

component activation
CDE conventions and standards for, *CDEStyle* 8, 182 to 187

composing characters, *DWUser* B–1

compound strings, *CDE_I18N_Prog* 80 to 83

Computer Interconnect
See CI

computer virus
See virus

conditional compilation, *DECC* 8-10

configuration

declarations and definitions section, *DevDriverTURBOchannel* 6–9 to 6–11

EISA utility (ECU), *DevDriverEisa* 9–1, B–1

EISA/ISA bus device driver, *DevDriverEisa* 8–1

of kernel subsystems, *SystemAdmin* 4–1

support section, *DevDriverTURBOchannel* 6–24 to 6–37

TURBOchannel device driver, *DevDriverTURBOchannel* 7–1

configuration attributes

See also sysconfig command, sysconfigdb command, /etc/sysconfigtab file

attribute descriptions, *SystemTune* B–1 to B–30

configuration checklist, *TechOver* 12–3

configuration file

adding new devices, *SystemAdmin* 4–16

callout keywords, *SystemAdmin* 4–43

defining network protocols in, *SystemAdmin* 4–48

definitions

statistics, *SystemAdmin* 4–48

device definition keywords, *SystemAdmin* 4–43

EISA/ISA bus, *DevDriverEisa* B–1

writing, *DevDriverEisa* 9–1

entries, *SystemAdmin* 4–30

for CDE, *CDEAdvUsr* 21, 37, 56, 157, 187, 212, 242

keywords, *SystemAdmin* 4–30

makeoptions keywords, *SystemAdmin* 4–50

options keywords, *SystemAdmin* 4–45

pseudodevice keywords, *SystemAdmin* 4–50

configuration file (cont.)

SMP options, *SystemAdmin* 4–45

system definition keyword, *SystemAdmin* 4–42

workstation definitions, *SystemAdmin* 4–51

configuration files

sample files, *DWUser* D–1

configure interface

defined, *DevDriverTutorial* 6–1

configuring, *STREAMSProg* 6–2, 9–5 to 9–6, E–1 to E–4

kernel, *SystemAdmin* 4–1

conformance

to RFC 1122 and 1123, *TechOver* A–1 to A–13

Conformance to Internet Host Requirements, *TechOver* A–1

connecting objects with App Builder, *AppBuilder* 70 to 76

connecting to an unknown remote host via modem (UUCP), *CommandShellGde* 14–10

connecting to other systems, *Security* 3–1

connection

making in ATM, *ATMProgGde* 7–1

point-to-multipoint, *ATMProgGde* 7–1

point-to-point, *ATMProgGde* 7–1, *NetAdmin* 4–1 to 4–19

reserving resources, *ATMProgGde* 7–2

connection-oriented applications

initializing an endpoint, *NetProgGde* 3–24

writing, *NetProgGde* 3–24 to 3–36

connection-oriented programs, *NetProgGde* B–2 to B–17

connectionless applications

writing, *NetProgGde* 3–37 to 3–39

connectionless programs, *NetProgGde* B–17 to B–30

connectionless service in XTI

state transitions allowed, *NetProgGde* 3–17t

connections

passing to another endpoint, *NetProgGde* 3–16

constant, *DECC* 1-13

constants

floating-point, *AsmLangGde* 2–2

scalar, *AsmLangGde* 2–2

string, *AsmLangGde* 2–3

constants and rpcgen, *OncRpcProgram* 2–23

context

color attributes

obtaining, *displayPS* 6–4

setting, *displayPS* 6–5

creating

execution context, *displayPS* 5–7, 5–9

text context, *displayPS* 5–6

defined, *displayPS* 2–2 to 2–3

pausing, *displayPS* 6–3

sharing, *displayPS* 4–7

unfreezing, *displayPS* 5–13

XID, *displayPS* 5–13

context status events

See also XDPSRegisterStatusProc routine and XDPSSetStatusMask routine

context status mask

See also XDPSSetStatusMask routine

context variables

in dtksh, *Desktopksh* 27

context-sensitive help, *CDEHelp* 216

continuing a command on the next line, *CommandShellGde* 11–16

control message, *STREAMSProg* 4–3 to 4–4, 7–1

control part, *STREAMSProg* 4–3, 4–8 to 4–15

control status register (CSR)

See device register

controller

defined, *DevDriverTutorial* 1–9

relationship to device driver, *DevDriverTutorial* 1–9

controller structure

associated include file, *DevDriverReference* 3–20, *DevDriverVMEbus* A–36

defined, *DevDriverTutorial* 17–7

list of member names and data types, *DevDriverReference* 3–20, *DevDriverVMEbus* A–36

controller_config structure

associated include file, *DevDriverReference* 3–24

definition of members, *DevDriverReference* 3–24

controlling access, *CommandShellGde* 5–1

controlling Stream, *STREAMSProg* 1–3, 4–3 to 4–4

controls, *CDEAdvUsr* 215, 219, 220 to 223, 228 to 238

implementing with App Builder, *AppBuilder* 6, 124, 125

controls, groups, and models

CDE conventions and standards for, *CDEStyle* 8, 9, 234, 236, 237, 242 to 246, 248, 250, 252, 254

conversion of data types, *DECC* 6-27

coordinate systems, *displayPS* 2–5 to 2–7

coordinates, converting, *displayPS* 4–2 to 4–3

copyin kernel interface
explanation of code fragment, *DevDriverTutorial* 18–11

copying a memory block
to another I/O address space with io_copyio kernel interface, *DevDriverTutorial* 18–25
to I/O address space with io_copyout kernel interface, *DevDriverTutorial* 18–23
to system memory with io_copyin kernel interface, *DevDriverTutorial* 18–21

copying files, *CommandShellGde* 3–19
local host control (UUCP), *CommandShellGde* 14–24
using UUCP, *CommandShellGde* 14–22

copyout kernel interface
explanation of code fragment, *DevDriverTutorial* 18–12

copysign routine, *DPML* DPML–21, B–6

core file debugging
Ladebug, 14-1

cos routine, *DPML* DPML–22, B–6

cosd routine, *DPML* DPML–22, B–6

cosh routine, *DPML* DPML–23, B–7

cot routine, *DPML* DPML–24, B–7

cotd routine, *DPML* DPML–24, B–7

cp command, *CommandShellGde* 3–19

cpow routine, *DPML* DPML–25, B–7

CPU
See central processing unit

crash data collection, *KernelDebug* 2–48

crash dump analysis, *Ladebug* 22-7
collecting data with crashdc, *KernelDebug* 2–48
examples of, *KernelDebug* 5–1
guidelines for, *KernelDebug* 5–1

crash dump analysis (cont.)
overview, *KernelDebug* 1–1

crash dump file
invoking dbx debugger to examine, *KernelDebug* 2–2
invoking kdbx debugger to examine, *KernelDebug* 2–11

crash dump file creation, *KernelDebug* 4–1

crash dumps
forcing on a hung system, *KernelDebug* 4–13
planning space, *InstallGuide* 3–12

crash recovery, *SystemAdmin* 13–34

crash-data.n file
explanation of contents, *KernelDebug* A–1

crashdc command
explanation of output from, *KernelDebug* A–1
overview of, *KernelDebug* 2–48

Create Action, *CDEAdvUsr* 141

crontab command, *SystemAdmin* 3–12

cross-references in CDE help, *CDEHelp* 137, 144, 150, 178

.cshrc file, *CommandShellGde* 8–3

.cshrc login script, *CommandShellGde* 7–17
modifying for System V Habitat, *CommandShellGde* 9–2

csin routine, *DPML* DPML–26, B–8

csqrt routine, *DPML* DPML–27, B–8

CSR
See device register

ct command
options, connecting to remote host via modem, *CommandShellGde* 14–14

cu command
connect local to remote, *CommandShellGde* 14–8

cu command (cont.)
using local commands, *CommandShellGde* 14–6
current folder, *CDE_User* 84
current session, *CDE_User* 31
currentXgcdrawable operator, *displayPS* 6–3
currentXgcdrawablecolor operator, *displayPS* 6–4
currentXoffset operator, *displayPS* 6–4
customization
workstation environment, *XWinSysEnv* 3–1
customizing application environments, *DWUser* 9–2
cvt_ftof routine, *DPML* DPML–28, B–8

D

DAC, *Security* 5–1
daemon
inetd, *NetProgGde* 4–54
tftpd, *SharingSW* 6–4
data copying interface
category of kernel interface, *DevDriverTutorial* 18–8
data flow
XTI and a sockets-based transport provider, *TechOver* 3–19
XTI and a STREAMS-based transport provider, *TechOver* 3–19
data integration GUIs
in CDE, *CDEProgOverview* 5
Data Link Interface
See DLI
data link interfaces, *NetProgGde* 1–3, 2–1 to 2–12
DLPI, *NetProgGde* 2–1
Data Link Provider Interface
See DLPI
data structure
ATM, *ATMProgGde* 3–14
CAM control block structures, *SCSIDevDrivers* 5–1
displaying the format of with dbx debugger, *KernelDebug* 2–6
displaying with the dbx debugger, *KernelDebug* 2–6
EISA/ISA bus device driver, *DevDriverEisa* A–9
message, *NetProgGde* 5–18
module, *NetProgGde* 5–17
data transfer phase
of connectionless service, *NetProgGde* 3–37
state transitions allowed for connectionless transport services, *NetProgGde* 3–17t
data type conversion, *DECC* 6-27
data types, *CDE_User* 126, *DECC* 3-1
creating manually, *CDEAdvUsr* 185
creating with Create Action, *CDEAdvUsr* 141
introduction to, *CDEAdvUsr* 127
types supported by assembly language, *AsmLangGde* 2–10
data typing
and CDE application development, *CDEProgGuide* 123 to 136
in CDE application development, *CDEProgOverview* 17, 18, 68, 69
database search path, *CDEAdvUsr* 104, 119 to 121, 157
database servers, *CDEAdvUsr* 94, 101 to 104
databases
guidelines for in CDE application development, *CDEProgOverview* 34

dataless management services, *SharingSW* 8–1
Dataless Management Services, *TechOver* 13–8
See also DMS
date command, *SystemAdmin* 2–15
date values
formatting rules in a locale, *International* 7–20
dbx, *TechOver* 7–3
dbx commands
See also dbx debugger
dbx debugger
See also dbx commands
command-line flags, *ProgGde* 5–7
kernel debugging flag, *KernelDebug* 2–2
operator precedence, *ProgGde* 5–9
using to debug kdbx extensions, *KernelDebug* 3–41
dbxinit file, *Ladebug* 15-1
dd command, *SystemAdmin* 6–28
DDR, *SystemAdmin* 5–1
compiling changes to databases, *SystemAdmin* 5–2
conforming to standards, *SystemAdmin* 5–2
converting cam_data.c file, *SystemAdmin* 5–3
database, *SystemAdmin* 5–2
help option, *SystemAdmin* 5–2
introduction, *SystemAdmin* 5–1
SCSI-2 standard, *SystemAdmin* 5–2
synchronizing on-disk and in-memory databases, *SystemAdmin* 5–3
ddr_config command
help option, *SystemAdmin* 5–2
TagQueueDepth parameter changes, *SystemAdmin* 5–3
ddr_config utility, *SystemAdmin* 5–1
ddr.dbase file, *SystemAdmin* 5–2
deadlock
defined, *DevDriverAdvTopics* 6–4
issue to consider when using kernel threads, *DevDriverAdvTopics* 6–4
deallocating system resources for DMA
by calling dma_map_dealloc kernel interface, *DevDriverTutorial* 18–35
debug macros
CAM described, *SCSIDevDrivers* 10–1
introduction to CAM, *SCSIDevDrivers* 10–1
debug routines
introduction to CAM, *SCSIDevDrivers* 10–1
debugger
See dbx debugger
dbx, *TechOver* 7–3
ladebug, *TechOver* 7–2
debugger commands, *Ladebug* REF-3
debugger features, *Ladebug* 1-2
debugger variables, predefined, *Ladebug* REF-28
debugging, *ProgGde* 1–5
See also dbx debugger, ladebug debugger, lint, Third Degree
general concepts, *ProgGde* 5–3
programs using shared libraries, *ProgGde* 4–16
DEC C compiler, *DECC* XIV
DECevent
binary event-logging reports, *SystemAdmin* 13–23
error reporting, *SystemAdmin* 13–1
DECevent features, *DECevent* 3–1 to 3–5
decl_simple_lock_data kernel interface
declaring simple lock structure, *DevDriverAdvTopics* 3–1

decl_simple_lock_data kernel interface (cont.)
explanation of code fragment, *DevDriverAdvTopics* 3–2
declarations, *DECC* 4-1, *OncRpcProgram* 2–24
declarations section
description, *DevDriverTutorial* 5–10
description of code example for /dev/dmaex driver, *DevDriverVMEbus* 6–5 to 6–6
declaring complex lock structure, *DevDriverAdvTopics* 4–1
explanation of code fragment, *DevDriverAdvTopics* 4–2
declaring simple lock structure, *DevDriverAdvTopics* 3–1
DECtalk software, *InstallGuide* D–8
DECterm, *DWUser* 5–40, 7–1, B–1, 5–41
window, *DWUser* 2–3
DECterm software
escape sequences for ruled lines, *International* 4–16
DECthreads, *DECthreads* 1–1
debugging, *Ladebug* 9-1
DECwindows
security, *Security* 4–1
DECwindows Motif, *DWUser* 1–1
customizing the Session Manager, *DWUser* 7–1
customizing windows, *DWUser* 3–12, 7–4
getting help, *DWUser* 2–5
interacting with, *DWUser* 3–12
introduction, *DWUser* 1–1
managing windows, *DWUser* 3–2
migrating to CDE, *CDECompanion* 6–1 to 6–3
mixing colors, *DWUser* 4–6
running applications, *DWUser* 2–3

DECwindows Motif (cont.)
running applications remotely, *DWUser* 4–10
starting a session, *DWUser* 2–1
starting applications, *DWUser* 5–2
using Bookreader, *DWUser* 5–4
using Calculator, *DWUser* 5–10
using Calendar, *DWUser* 5–21
using Cardfiler, *DWUser* 5–27
using CDA Viewer, *DWUser* 5–30
using Clock, *DWUser* 5–37
using DECterm, *DWUser* 5–40
using Differences, *DWUser* 5–63
using Mail, *DWUser* 5–41
using Notepad, *DWUser* 5–47
using Paint, *DWUser* 5–55
using Print Screen, *DWUser* 5–59
using the keyboard, *DWUser* A–1
using the mouse, *DWUser* 3–1
using with a keyboard, *DWUser* A–1
working with text, *DWUser* 4–1
DECwindows Motif, graphical user interface
enhancements for native languages, *International* 6–86
DECwindows, using
setting language, *International* 6–3
default text backstop
See DPSDefaultTextBackstop routine
default user space origin
See user space origin
deferred mode swapping, *SystemAdmin* 6–5
deinstalling software subsets, *InstallGuide* 8–1
DELAY kernel interface
explanation of code fragment, *DevDriverTutorial* 18–14
delete subcommand (ftp), *CommandShellGde* 12–7

demo programs
of CDE application development, *CDEProgOverview* 27, 35, 47, 50, 51, 54, 67, 69, 70, 72, 73, 76
demostration program, CoEd, *ToolTalkMsg* 95 to 123
dense numbers, *AsmLangGde* 8–3
dense space
See dense space address
dense space address
description of address type that device drivers use, *DevDriverTutorial* 2–16
design guidelines, *STREAMSProg* D–1, D–17
desktop, *CDE_User* 4
desktop environment
customizing, *CDE_User* 133
Desktop KornShell (dtksh), *CDEProgOverview* 14, 76
introduction to, *Desktopksh* 11 to 22
desktop management
in CDE application development, *CDEProgOverview* 5, 8 to 12
/dev/dmaex device driver
dmaexreg.h file, *DevDriverVMEbus* 6–2
Include Files Section, *DevDriverVMEbus* 6–3
introductory discussion, *DevDriverVMEbus* 6–1
/dev/pnvram device driver, *DevDriverPci* 6–1
dev_t data type
defined, *DevDriverTutorial* 8–1
development environment, *TechOver* 7–1
DEVGEOMST structure
associated include file, *DevDriverReference* 3–26
definition of members, *DevDriverReference* 3–26
devget structure
associated include file, *DevDriverReference* 3–28
definition of members, *DevDriverReference* 3–28
device
See peripheral device
device autoconfiguration
See autoconfiguration
device controller
See controller
device database, *SystemAdmin* 5–1
device definition keyword
configuring into the kernel, *SystemAdmin* 4–16
device driver
See also block device driver
See also character device driver
See also network device driver
See also pseudodevice driver
ATM interface, *ATMProgGde* 4–1
ATM structures, *ATMProgGde* 4–2
defined, *DevDriverTutorial* 1–1
disk, *DevDriverAdvTopics* 8–1
EISA/ISA bus
configuration, *DevDriverEisa* 8–1, 1–1
example, *DevDriverEisa* 7–1
structure of, *DevDriverEisa* 3–1
example of reading a character, *DevDriverTutorial* 1–10
implementing a close interface, *DevDriverTutorial* 10–4
implementing a configure interface, *DevDriverTutorial* 6–7
implementing a dump interface, *DevDriverTutorial* 9–1

device driver (cont.)
- implementing a memory map interface, *DevDriverTutorial* 8–15
- implementing a probe interface, *DevDriverTutorial* 7–1
- implementing a psize interface, *DevDriverTutorial* 9–2
- implementing a read interface, *DevDriverTutorial* 8–1
- implementing a reset interface, *DevDriverTutorial* 8–8
- implementing a select interface, *DevDriverTutorial* 8–9
- implementing a slave interface, *DevDriverTutorial* 7–27
- implementing a stop interface, *DevDriverTutorial* 8–15
- implementing a strategy interface, *DevDriverTutorial* 9–2
- implementing a write interface, *DevDriverTutorial* 8–5
- implementing an attach interface, *DevDriverTutorial* 7–23
- implementing an interrupt interface, *DevDriverTutorial* 10–9
- implementing an ioctl interface, *DevDriverTutorial* 10–7
- implementing an open interface, *DevDriverTutorial* 10–1
- implementing an unattach interface, *DevDriverTutorial* 7–24
- Include Files Section for VMEbus, *DevDriverVMEbus* 3–3
- overview of driver integration, *DevDriverTutorial* 1–5
- PCI bus, *DevDriverPci* 1–1

device driver (cont.)
- PCI bus (cont.)
  - device configuration, *DevDriverPci* 5–1
  - example, *DevDriverPci* 6–1
- place in Digital UNIX, *DevDriverTutorial* 1–6
- queuing, *ATMProgGde* 9–1
- registering with CMM, *ATMProgGde* 4–1
- relationship to kernel, *DevDriverTutorial* 1–8
- sections
  - declarations, *DevDriverTutorial* 5–10
  - include files, *DevDriverTutorial* 5–4
- STREAMS processing routines for, *NetProgGde* 5–20
- structure of, *DevDriverPci* 3–1, *DevDriverTutorial* 5–1, *DevDriverVMEbus* 3–1
- to handle requests made by the kernel, *DevDriverTutorial* 1–1
- TURBOchannel bus
  - architecture, *DevDriverTURBOchannel* 2–1
  - Autoconfiguration Support Section, *DevDriverVMEbus* 3–3
  - concepts, *DevDriverTURBOchannel* 1–1
  - configuration, *DevDriverTURBOchannel* 7–1
  - data structures, *DevDriverTURBOchannel* 4–1
  - example, *DevDriverTURBOchannel* 6–1
  - kernel interfaces, *DevDriverTURBOchannel* 5–1
  - reference information, *DevDriverTURBOchannel* A–1
  - structure, *DevDriverTURBOchannel* 3–1
- types of, *DevDriverTutorial* 1–1

device driver (cont.)
VMEbus
kernel interfaces used by, *DevDriverVMEbus* 5–1
xxprobe interface, *DevDriverVMEbus* 3–4
when called by the kernel, *DevDriverTutorial* 1–3
device driver configuration
defined, *DevDriverTutorial* 13–1
device driver delivery process
defined, *DevDriverTutorial* 12–1
device driver development
choosing a locking method, *DevDriverAdvTopics* 2–11
comments
discussion of approaches for writing, *DevDriverTutorial* 2–5
deciding the resources to lock in a device driver, *DevDriverAdvTopics* 2–15
tasks
creating driver development and kitting environment, *DevDriverTutorial* 2–24
describing usage of device, *DevDriverTutorial* 2–8
determining structure allocation technique, *DevDriverTutorial* 2–18
identifying conventions used in writing, *DevDriverTutorial* 2–3, C–5
identifying driver entry points, *DevDriverTutorial* 2–11
identifying support in writing, *DevDriverTutorial* 2–9
providing description of device registers, *DevDriverTutorial* 2–9
specifying characteristics of device, *DevDriverTutorial* 2–6
device driver development (cont.)
tasks (cont.)
specifying information about host system, *DevDriverTutorial* 2–2
specifying locking method for SMP safe drivers, *DevDriverTutorial* 2–13
specifying method for identifying bus, controller, and device information, *DevDriverTutorial* 2–18
specifying method for registering device interrupt handlers, *DevDriverTutorial* 2–17
specifying type of driver, *DevDriverTutorial* 2–11
understanding bus issues affecting driver design, *DevDriverTutorial* 3–13
understanding CPU issues affecting driver design, *DevDriverTutorial* 3–1
understanding system data and address types that device drivers use, *DevDriverTutorial* 2–13
understanding atomicity, *DevDriverAdvTopics* 2–2
understanding the locking methods, *DevDriverAdvTopics* 2–6
understanding the need for locking in an SMP environment, *DevDriverAdvTopics* 2–4
device driver subsystem
accessing device registers of specific devices, *DevDriverTutorial* 16–8
defined, *DevDriverTutorial* 15–4
device mnemonics, *SystemAdmin* A–1
configuration file syntax, *SystemAdmin* A–1
in configuration data base, *SystemAdmin* A–1

device name, specifying with cu command (UUCP), *CommandShellGde* 14–5

device pathname

explanation of, *SystemAdmin* 10–6

representation in printcap, *SystemAdmin* 10–7

device register

accessed by device drivers, *DevDriverTutorial* 16–7

defined, *DevDriverTutorial* 16–5

discussion of, *DevDriverTutorial* 16–5

used by the device, *DevDriverTutorial* 16–7

device register header file

defined, *DevDriverTutorial* 5–6

for the /dev/dmaex device driver, *DevDriverVMEbus* 6–2

device register offset

defined, *DevDriverTutorial* 5–6

device special files, *NetProgGde* 5–29

representation in printcap, *SystemAdmin* 10–7

device structure

associated include file, *DevDriverReference* 3–31

defined, *DevDriverTutorial* 17–13

list of member names and data types, *DevDriverReference* 3–31

device switch structure

defined, *DevDriverTutorial* 5–24

initialized for cb driver, *DevDriverTutorial* 5–24

device switch table

See dsent

device_config structure

associated include file, *DevDriverReference* 3–33

definition of members, *DevDriverReference* 3–33

device_info_t structure

associated include file, *DevDriverReference* 3–36

definition of members, *DevDriverReference* 3–36

devices, *Security* 8–1

DHCP, *NetAdmin* 3–1

dialup connection, *NetAdmin* 4–20

dia command

binary event-logging reports, *SystemAdmin* 13–1

dialog boxes

CDE conventions and standards for, *CDEStyle* 25, 97, 102, 116, 124, 209, 212, 218

for CDE help, *CDEHelp* 207 to 214

Differences, *DWUser* 5–63, 5–66

comparing files, *CDECompanion* 5–5

Digital Storage Architecture disk

See DSA disk

Digital System Communication Architecture, *SystemAdmin* C–1

Digital UNIX system calls

and DLI, *NetProgGde* E–16

direct memory access

See DMA

direct memory access device

See also DMA device

discussion of, *DevDriverTutorial* 16–6

directives

assembler directives, *AsmLangGde* 5–1 to 5–15

directories, *CommandShellGde* 2–1, 2–7, 4–1

directory permissions, *CommandShellGde* 5–10, 5–5

dis command, *ProgGde* 2–26

disabilities and user customization

in CDE application development, *CDEProgOverview* 60, 61

discretionary access control

See DAC

discretionary check, *Security* 5–5

disk device driver, *DevDriverAdvTopics* 8–1

disk drive

adding, *SystemAdmin* 5–6

testing with diskx, *SystemAdmin* 13–6

disk partition, *SystemAdmin* 6–2

disk partition layout, *InstallGuide* 3–18

disk partition size, *InstallGuide* 7–10

disk quotas, *SystemAdmin* 6–25

See also file system quotas

disk space

planning, *InstallGuide* 3–1

reallocating, *SystemAdmin* 6–25

total required for installation, *InstallGuide* F–1

disk storage hardware, *SystemTune* 1–14

disk storage management, *StorageManager* 1–1

disklabel command, *SystemAdmin* 6–27

disklabel structure

associated include file, *DevDriverReference* 3–38

definition of members, *DevDriverReference* 3–39

disks

cloning, *SystemAdmin* 6–28

copying, *SystemAdmin* 6–28

encapsulating, *StorageManager* 4–1

label, *SystemAdmin* 6–3

load balancing with LSM, *StorageManager* 15–1

disks (cont.)

LSM, *StorageManager* 1–11

mirroring, *StorageManager* 2–6, 5–1

monitoring usage, *SystemAdmin* 6–20

operations with LSM, *SystemAdmin* 8–7

partition sizes, *InstallGuide* G–1

striping, *StorageManager* 2–4

supported, *InstallGuide* 3–5

diskusg command, *SystemAdmin* 12–32

diskx exerciser, *SystemAdmin* 13–6

dispatch tables, *OncRpcProgram* 2–18

display access, *Security* 4–1

display log of UUCP utilities, *CommandShellGde* 14–31

Display Manager, *TechOver* 8–7

display subcommand (telnet), *CommandShellGde* 13–4

displaying applications

using DECwindows Motif, *DWUser* 4–11

displaying differences between files, *CommandShellGde* 3–24

displaying files, *CommandShellGde* 3–5

displaying online reference pages, *CommandShellGde* 1–9

distributed naming services

See BIND naming service

See NIS

distributed system services

naming services, *TechOver* 3–27 to 3–28

time services, *TechOver* 3–28 to 3–30

distribution media

See also CD-ROM

DLI, *TechOver* 3–21

and accessing the local area network, *NetProgGde* E–3

concepts, *NetProgGde* E–1 to E–2

DLI (cont.)
- definition of, *NetProgGde* E–1
- services, *NetProgGde* E–3
- using Digital UNIX system calls, *NetProgGde* E–16

DLI client program
- using 802.3 format packets
  - example, *NetProgGde* E–31
- using Ethernet format packets
  - example, *NetProgGde* E–24

DLI program
- including higher-level services, *NetProgGde* E–4
- transferring data, *NetProgGde* E–22
- using getsockopt and setsockopt
  - example, *NetProgGde* E–41
- writing, *NetProgGde* E–15

DLI server program
- using 802.3 format packets
  - example, *NetProgGde* E–36
- using Ethernet packets
  - example, *NetProgGde* E–27

dli_802_3_conn subroutine
- example, *NetProgGde* E–41

dli_econn subroutine
- example, *NetProgGde* E–27

DLPI, *TechOver* 3–21
- addressing, *NetProgGde* 2–7
- defined, *NetProgGde* 2–2
- modes of communication, *NetProgGde* 2–3
- types of service, *NetProgGde* 2–4

DMA
- allocating system resources by calling dma_map_alloc kernel interface, *DevDriverTutorial* 18–29
- defined, *DevDriverTutorial* 16–4

DMA (cont.)
- device-to-device, *DevDriverVMEbus* 2–7
- for multiple VMEbus adapters, *DevDriverVMEbus* 2–7
- kernel interfaces related to, *DevDriverTutorial* 18–28
- loading and setting allocated system resources by calling dma_map_load kernel interface, *DevDriverTutorial* 18–31
- releasing and deallocating system resources by calling dma_map_dealloc kernel interface, *DevDriverTutorial* 18–35
- returning a kernel segment address by calling dma_kmap_buffer kernel interface, *DevDriverTutorial* 18–40
- unloading system resources by calling dma_map_unload kernel interface, *DevDriverTutorial* 18–33
- VMEbus to and from host, *DevDriverVMEbus* 2–6

DMA data transfer resources
- allocating with dma_map_alloc and dma_map_load, *DevDriverVMEbus* 5–1

DMA device
- defined, *DevDriverTutorial* 16–6

DMA handle
- defined, *DevDriverTutorial* 18–35
- description, *DevDriverTutorial* 18–28

dma_get_curr_sgentry kernel interface
- explanation of code fragment, *DevDriverTutorial* 18–36

dma_get_next_sgentry kernel interface
- explanation of code fragment, *DevDriverTutorial* 18–36

dma_kmap_buffer kernel interface
explanation of code fragment, *DevDriverTutorial* 18–41

dma_map_alloc kernel interface
explanation of code fragment, *DevDriverTutorial* 18–30

dma_map_load kernel interface
explanation of code fragment, *DevDriverTutorial* 18–32

dma_map_unload kernel interface
explanation of code fragment, *DevDriverTutorial* 18–34

dma_put_curr_sgentry kernel interface
explanation of code fragment, *DevDriverTutorial* 18–39

dma_put_prev_sgentry kernel interface
explanation of code fragment, *DevDriverTutorial* 18–39

dmaexclose interface
description of code example for /dev/dmaex driver, *DevDriverVMEbus* 6–17 to 6–19

dmaexminphys interface
description of code example for /dev/dmaex driver, *DevDriverVMEbus* 6–23

dmaexopen interface
description of code example for /dev/dmaex driver, *DevDriverVMEbus* 6–15 to 6–16

dmaexread interface
description of code example for /dev/dmaex driver, *DevDriverVMEbus* 6–20 to 6–21

dmaexreg.h file
description of code example for /dev/dmaex driver, *DevDriverVMEbus* 6–2

dmaexreg.h file (cont.)
device register header file for /dev/dmaex device driver, *DevDriverVMEbus* 6–2

dmaexwrite interface
description of code example for /dev/dmaex driver, *DevDriverVMEbus* 6–22

DMS, *SharingSW* 8–1

DMS area
defined, *SharingSW* 1–1

DMS client
defined, *SharingSW* 1–1

DMS server
defined, *SharingSW* 1–1

dmu utility, *SharingSW* 8–1

doconfig program, *SystemAdmin* 4–16, 4–20, 4–22

dodisk shell script, *SystemAdmin* 12–31

DOMAIN, *TechOver* 3–5

Domain Name Protocol
See DOMAIN

DPSChangeEncoding routine, *displayPS* 5–5

DPSContext handle
defined, *displayPS* 2–4
finding, *displayPS* 5–11, 5–5, 5–6, 5–7

DPSContextFromContextID routine, *displayPS* 5–5

DPSCreateTextContext routine, *displayPS* 5–6

DPSDefaultTextBackstop routine, *displayPS* 5–6

DPSNewUserObjectIndex routine, *displayPS* 5–6

DPSSpace handle
finding, *displayPS* 5–12

dpsXclient.h file
See system-specific header file

drag and drop
CDE conventions and standards for, *CDEStyle* 11 to 37, 178
implementing with App Builder, *AppBuilder* 2, 6, 81, 82
in CDE application development, *CDEProgOverview* 34, 52, 53
integrating into CDE applications, *CDEProgGuide* 33 to 56

drem routine, *DPML* DPML–33, B–8

driver, *STREAMSProg* 9–1 to 9–13

driver close, *STREAMSProg* 5–2, 5–27, 6–6

driver declarations, *STREAMSProg* 6–1, 9–4, 10–2

driver flow control, *STREAMSProg* 5–7, 9–1, 9–3

driver programs, *ProgGde* 2–3

driver structure
associated include file, *DevDriverReference* 3–44, *DevDriverVMEbus* A–38
definition of members, *DevDriverReference* 3–44
list of member names and data types, *DevDriverVMEbus* A–38

DSA disk, *SystemAdmin* 6–20

dsent structure
associated include file, *DevDriverReference* 3–47
definition of members, *DevDriverReference* 3–47

DtComboBox
text field and list box widget, *CDEProgGuide* 85

DTD (document type definition)
CDE HelpTag, *CDEHelp* 195 to 204

DtEditor
text editor widget, *CDEProgGuide* 94

dthelpgen command, *CDEHelp* 193

dthelptag command, *CDEHelp* 90, 190

dthelpview command, *CDEHelp* 93, 192

dtksh app-defaults file, *Desktopksh* 14

dtksh commands, *Desktopksh* 51 to 85

dtksh (Desktop KornShell), *Desktopksh* 1

dtmail, *CDECompanion* 10–1

DtMenuButton
menu button widget, *CDEProgGuide* 89

dtpad text editor
and CDE application development, *CDEProgOverview* 7, 73, 74
using, *CDE_User* 191

.dtprofile file, *CDEAdvUsr* 25, 26, 33, 277

DtSpinBox
text field and arrow button widget, *CDEProgGuide* 73

dump file
guidelines for examining, *KernelDebug* 5–1

dxaccounts program, *Security* 6–7

dxaudit program, *Security* 6–7

dxcalendar
converting to CDE format, *CDECompanion* 10–6 to 10–7

dxdevices program, *Security* 6–7

dxlsm interface (LSM), *StorageManager* 10–1, 11–1, 12–1

dxmail
converting to CDE mail format, *CDECompanion* 10–1 to 10–6
using under DECwindows Motif, *DWUser* 5–41

dxpower command, *SystemAdmin* 3–23

dxpresto command

monitoring Prestoserve with, *Prestoserve* 3–9

Dynamic Device Recognition, *TechOver* 13–8

See DDR

dynamic linking, *AsmLangGde* 9–4

dynamic loader

See also shared libraries

use of, *AsmLangGde* 9–4

dynamic relocation section

See .rel.dyn section

.dynamic section

contents, *AsmLangGde* 9–5

ordering for quickstart, *AsmLangGde* 9–24

dynamic string section

See .dynstr section

dynamic symbol section

See .dynsym section

dynamically configurable subsystem

creating, *ProgGde* C–1 to C–29

.dynstr section, *AsmLangGde* 9–22

.dynsym section, *AsmLangGde* 9–16

relationship with .got section, *AsmLangGde* 9–18

E

/e character

in CDE help, *CDEHelp* 28, 29, 129

ed editor, *CommandShellGde* B–1

editor, *CommandShellGde* 2–1

ed, *CommandShellGde* B–1

vi, *CommandShellGde* A–1

EGP, *TechOver* 3–6

egrep

See grep

EISA bus, *TechOver* 6–5

EISA Configuration Utility (ECU), *DevDriverEisa* 9–1, B–1

EISA/ISA bus

architecture, *DevDriverEisa* 2–1

autonconfiguration, *DevDriverEisa* 6–1

configuration file format, *DevDriverEisa* 9–1, B–1

data structures, *DevDriverEisa* 4–1

device driver configuration, *DevDriverEisa* 8–1

device driver example, *DevDriverEisa* 7–1

kernel interface, *DevDriverEisa* 5–1

structure of device driver, *DevDriverEisa* 3–1

emacs and debugging, *Ladebug* A-1

enabling realtime preemption, *InstallGuide* 6–7

encapsulating disks

LSM, *StorageManager* 4–1

encoding, PostScript language, *displayPS* 4–1, 5–5

encodings and codesets, *CDE_I18N_Prog* 56, 59, 64 to 67, 69, 70, 72

end of message/conversation (local communications), *CommandShellGde* 11–22, 11–23

end system identifier

ESI, *ATMProgGde* 3–5

ending a local message, *CommandShellGde* 11–22

enhanced security, *Security* xxi

entities

in CDE help, *CDEHelp* 46

enum clnt_stat in RPC programming, *OncRpcProgram* 3–3

enumerated data type, *ProgGde* 6–6
enumerations and rpcgen, *OncRpcProgram* 2–22
environment, *STREAMSProg* 2–8, 9–1, C–1
  See also DMS
environment variables
  for CDE help, *CDEHelp* 191, 248, 259
  in CDE, *CDEAdvUsr* 19, 25, 26, 28, 32, 33, 98, 115, 117, 120, 122, 124, 179
  PROFFLAGS, *ProgGde* 8–24
environments area
  See DMS
erf routine, *DPML* DPML–31, B–9
erfc routine, *DPML* DPML–31, B–9
error codes, *ATMProgGde* 3–2
error handling
  CAM_ERROR macro, *SCSIDevDrivers* 9–1
  introduction to CAM, *SCSIDevDrivers* 9–1
error logging, *SystemAdmin* 13–1
  in STREAMS, *NetProgGde* 5–31
error messages, *InstallGuide* H–1
  and CDE application development, *CDEProgOverview* 57, 58
  LSM, *StorageManager* B–1
error recovery routines
  in CDE, *CDEProgGuide* 27
errors
  displaying from CDE applications, *CDEProgGuide* 25 to 28
  in XTI, *NetProgGde* 3–64 to 3–65
ESI
  definition, *ATMProgGde* 3–5
eSNMP, *NetProgGde* 1–7
eSNMP application programming interface
  Digital UNIX support for, *NetProgGde* 6–1
/etc/atm.conf file, *ATMProgGde* 2–2
/etc/rc.config file, *SystemAdmin* 3–3
/etc/sysconfigtab file, *SystemAdmin* 4–10
  EISA/ISA options, *DevDriverEisa* 8–1
ethernet, *NetAdmin* 2–11, B–1 to B–4, *TechOver* 3–17
  configuration definition, *SystemAdmin* 4–53
Ethernet
  setting up a client on, *SharingSW* 3–3
event
  in XTI
    outgoing, *NetProgGde* 3–15t
  management
    and TLI compatibility, *NetProgGde* 3–43
  tracking of outgoing events
    in XTI, *NetProgGde* 3–14
  used by connectionless transport services, *NetProgGde* 3–37
event handling in CDE help dialogs, *CDEHelp* 229 to 233
event logger
  interface to, *DECevent* 1–1
event logging
  crash recovery, *SystemAdmin* 13–34
  creating daily files, *SystemAdmin* 13–27
  in STREAMS, *NetProgGde* 5–31
example applications
  See sample applications
exception code, *ProgGde* 11–6
exception filter, *ProgGde* 11–5
exception frame
  examining with the dbx debugger, *KernelDebug* 2–7
exception handler, *ProgGde* 11–6
exception handling
  application development considerations, *ProgGde* 11–1

exception handling (cont.)
header files, *ProgGde* 11–3
exception_code function, *ProgGde* 11–6
exception_info function, *ProgGde* 11–6
exceptions
defined, *ProgGde* 11–1
frame-based, *ProgGde* 11–5
structured, *ProgGde* 11–5
executable image
creating, *ProgGde* 2–20
execution aliases in CDE help, *CDEHelp* 72
execution link control in CDE help, *CDEHelp* 72
execution string, *CDEAdvUsr* 166 to 170
exp routine, *DPML* DPML–32, B–9
expedited data
and connectionless transport services, *NetProgGde* 3–37
expm1 routine, *DPML* DPML–32, B–9
expression, *DECC* 6-1
Extended Industry Standard Architecture
See EISA bus
See ISA bus
extended regular expression, *ProgSupTools* 1–3
extensible SNMP
See eSNMP
Exterior Gateway Protocol
See EGP
external names, *ProgGde* 6–12
external string table, *AsmLangGde* 8–5
external symbol table, *AsmLangGde* 8–22

F

fabs routine, *DPML* DPML–36, B–10
factory installed software, *InstallGuide* 1–19
fast ethernet, *TechOver* 3–17
fd variable
and outgoing events, *NetProgGde* 3–14
FDDI, *NetAdmin* 2–11 to 2–19, B–5 to B–15, *TechOver* 3–17
setting up a client on, *SharingSW* 3–3
FDFS, *TechOver* 4–8
feedback files, *ProgGde* 8–21
FFM, *TechOver* 4–8
fgrep
See grep
file access, *CommandShellGde* 5–1
file command, *ProgGde* 2–25
File Descriptor File System
See FDFS
file descriptor table, *AsmLangGde* 8–6
file header
flags (s_flags), *AsmLangGde* 7–8
file inclusion, *DECC* 8-11
File Manager, *CDE_User* 81, *CDECompanion* 3–1
and CDE application development, *CDEProgOverview* 6, 7, 10, 11, 17
file names, *CommandShellGde* 2–5
file ownership and security, *CDE_User* 99
file permissions, *CommandShellGde* 5–5
file selection boxes, *CDEStyle*
file system, *CommandShellGde* 2–5, *TechOver* 4–1
accelerating writes to, *Prestoserve* 3–4
checking, *SystemAdmin* 6–14
corrupted, *SystemAdmin* 6–14
creating, *SystemAdmin* 6–14

file system (cont.)
exercising with fsx, *SystemAdmin* 13–3
exporting, *NetAdmin* 8–10
monitoring usage, *SystemAdmin* 6–20
mounting, *SystemAdmin* 6–15
tuning, *SystemAdmin* 6–19
types of, *SystemAdmin* 6–1
file system quotas, *SystemAdmin* 6–23
for groups, *SystemAdmin* 6–23
for user accounts, *SystemAdmin* 6–23
File Transfer Protocol
See FTP
file type
determining, *CommandShellGde* 3–30
File-on-File Mounting File System
See FFM
files, *CommandShellGde* 2–1, 3–1
See header files; object files; source files
copying (UUCP), *CommandShellGde* 14–22
definition of, *CommandShellGde* 2–5
receiving (UUCP), *CommandShellGde* 14–21
sending (UUCP), *CommandShellGde* 14–21
sorting contents of, *CommandShellGde* 3–26
files file
defined, *DevDriverTutorial* 13–6
files file fragment
defined, *DevDriverTutorial* 13–6
use by config program, *DevDriverTutorial* 13–6
FileShare application, *NetAdmin* 8–10
filters, *CommandShellGde* 7–8
FINGER, *TechOver* 3–10
finger command, *CommandShellGde* 10–2
finite routine, *DPML* DPML–37, B–10
FIS
See factory installed software
floating-point constants, *AsmLangGde* 2–2
floating-point range and processing, *ProgGde* 2–5
floor routine, *DPML* DPML–38, B–10
flow control, *ATMProgGde* 10–1 to 10–5, *STREAMSProg* 1–3, 8–1 to 8–7, 9–1 to 9–6, 13–4
flush handling, *STREAMSProg* 9–4, 9–9, 10–7, 11–14, 11–20, 12–7
fmod routine, *DPML* DPML–39, B–10
focus management, *CDE_I18N_Prog* 19, 90, 92, 117
folder command in mail, *CommandShellGde* 11–13
font cache
tuning size of, *International* 6–86
font lists, *CDE_I18N_Prog* 10, 75, 77, 80, 83, 87, 93
font management, *CDE_I18N_Prog* 23, 27
font renderer, *TechOver* 8–8
font server, *TechOver* 8–8
font sets, *CDE_I18N_Prog* 9, 11, 78, 93, 109, 110
fonts
administering, *CDEAdvUsr* 255
and CDE application development, *CDEProgOverview* 55 to 59
for non-English languages, *International* 6–92
in CDE help, *CDEHelp* 63, 174, 260, 263
integrating into CDE applications, *CDEProgGuide* 17 to 21
foreground processes, *CommandShellGde* 6–6
Fortran language debugging support, *Ladebug* 11-1

forwarding files (UUCP), *CommandShellGde* 14–14

fp_class routine, *DPML* DPML–40, B–11

frame-based exception handling, *ProgGde* 11–5

framework

STREAMS, *NetProgGde* 5–1 to 5–32

frexp routine, *DPML* DPML–41, B–11

front panel, *CDE_User* 61

accessing applications, *CDECompanion* 3–2

advanced customizations, *CDEAdvUsr* 211

and CDE application development, *CDEProgOverview* 6, 7, 11

fsck program, *SystemAdmin* 6–15, 6–16

checking file system, *SystemAdmin* 6–14

fstab file, *SystemAdmin* 6–15, 6–16

fsx exerciser, *SystemAdmin* 13–3

FTP, *TechOver* 3–9

ftp subcommands, *CommandShellGde* 12–5, 12–7, 12–8

function, *DECC* 5-1

function prototype, *DECC* 5-8

functions

STREAMS, *NetProgGde* 5–6 to 5–17

funnel

defined, *DevDriverAdvTopics* 2–10

description of, *DevDriverAdvTopics* 2–10

ensuring no locks are required, *DevDriverAdvTopics* 5–1

Futurebus+, *TechOver* 6–7

G

gamma routine, *DPML* DPML–46

gated daemon, *NetAdmin* 2–14, *TechOver* 3–25

gateway

configuration definition, *SystemAdmin* 4–53

gauges

CDE conventions and standards for, *CDEStyle* 9, 254

gawk

See awk

GC

See X Graphic Context

genvmunix

See booting

geometry management, *CDE_I18N_Prog* 89

get subcommand (ftp), *CommandShellGde* 12–5

getty command, *SystemAdmin* 2–11, 2–13, 3–7, 3–8

gettydefs file, *SystemAdmin* 3–3

global optimizer (uopt), *ProgGde* D–1

glossaries in CDE help, *CDEHelp* 27, 83

.got section, *AsmLangGde* 9–13

relationship with .dynsym section, *AsmLangGde* 9–18

gprof

profiling tool, *ProgGde* 8–10

granularity of data access

defined, *DevDriverAdvTopics* 2–3

implementation on Alpha CPU hardware platforms, *DevDriverAdvTopics* 2–3

with reference to size of neighboring units of memory, *DevDriverAdvTopics* 2–3

Graphic Context

See X Graphic Context

graphic file formats for CDE help, *CDEHelp* 76

graphical user interface engine

and CDE application development, *CDEProgOverview* 5, 12 to 15

graphics in CDE help, *CDEHelp* 76 to 78
graphics state, *displayPS* 2–3, 6–2, 6–4
grep utility, *ProgSupTools* 1–9 to 1–11
group
adding automatically, *SystemAdmin* 9–18
adding manually, *SystemAdmin* 9–18
group file, *CommandShellGde* 5–1
adding a group to, *SystemAdmin* 9–18, 9–8
deleting a group from, *SystemAdmin* 9–19
deleting users from, *SystemAdmin* 9–16
groups and grouping in App Builder, *AppBuilder* 90 to 92, 145
grpck command, *SystemAdmin* 9–11

H

halt command, *SystemAdmin* 2–19
halting systems, *SystemAdmin* 2–17
handler_add kernel interface
explanation of code fragment, *DevDriverTutorial* 7–15
handler_enable kernel interface
explanation of code fragment, *DevDriverTutorial* 7–15
handler_intr_info structure
associated include file, *DevDriverReference* 3–51
defined, *DevDriverTutorial* 7–8
list of member names and data types, *DevDriverReference* 3–51
hardware
preparing for hardware installation, *InstallGuide* 4–2
hardware activities
related to device drivers, *DevDriverTutorial* 16–6
hardware components
central processing unit, *DevDriverTutorial* 16–2
hardware device
See also peripheral device
adding support of to the kernel, *SystemAdmin* 4–16
hardware exception
example of debugging, *KernelDebug* 5–4
hardware issues related to synchronization, *DevDriverAdvTopics* 2–1
hardware-dependent subsystem
defined, *DevDriverTutorial* 15–3
hardware-independent model
defined, *DevDriverTutorial* 15–1
hardware-independent subsystem
defined, *DevDriverTutorial* 15–2
hardware-related interface
category of kernel interface, *DevDriverTutorial* 18–14
.hash section, *AsmLangGde* 9–21
hash table section
See .hash section
header file
See also include files
cb device driver, *DevDriverTURBOchannel* 6–3 to 6–5
EISA/ISA bus device driver, *DevDriverEisa* A–1
header files, *DECC* 1-20, 9-1
See also Application, building
including, *ProgGde* 2–10
sockets, *NetProgGde* 4–15 to 4–16
used by device drivers, *SCSIDevDrivers* A–1
XTI and TLI, *NetProgGde* 3–6

help, *CDEAdvUsr* 66, 110, 146, 162, 190, 235, *CommandShellGde* 1–9
  CDE, *CDE_User* 32, 41
  CDE conventions and standards for, *CDEStyle* 82, 130, 185, 203
  writing for CDE applications, *CDEHelp* 1
help and help connections
  creating with App Builder, *AppBuilder* 68 to 71, 140, 150
help dialog boxes
  creating and managing, *CDEHelp* 207 to 214
help key (F1)
  enabling, *CDEHelp* 220
Help Manager, *CDECompanion* 1–7
help menu
  providing in CDE, *CDEHelp* 224
help on help in CDE, *CDEHelp* 235 to 241
help on items in CDE, *CDEHelp* 225
help online, for DECwindows applications, *DWUser* 2–5
help requests
  responding to CDE, *CDEHelp* 215 to 226
help search path, *CDEAdvUsr* 123
help servers, *CDEAdvUsr* 89
help subcommand (ftp), *CommandShellGde* 12–8
help system
  and CDE application development, *CDEProgOverview* 7, 34, 46, 47
help topic organization in CDE, *CDEHelp* 11 to 13, 25 to 48
help user interface in CDE, *CDEHelp* 5 to 10
help volumes, *CDEAdvUsr* 66
  processing and displaying, *CDEHelp* 87 to 100
  structure of CDE, *CDEHelp* 29
HelpTag, *CDEHelp* 88
helptag.opt file, *CDEHelp* 36
Hierarchical Storage Controllers
  See HSC
high-level services
  providing, *NetProgGde* E–4, E–16
hiprof (Atom tool), *ProgGde* 9–2
home session, *CDE_User* 38, 145
hometopic in CDE help, *CDEHelp* 7, 8, 26, 39, 45, 132, 240
host, *configuring* 11–19, *testing* 14–1
host CPU
  See central processing unit
HSC
  configuration, *SystemAdmin* C–1
  controller failures, *SystemAdmin* C–2
  host sharing, *SystemAdmin* C–3
  restrictions, *SystemAdmin* C–3
hung system
  forcing a crash dump on, *KernelDebug* 4–13
hyperlinks in CDE help files, *CDEHelp* 64 to 75
hypot routine, *DPML* DPML–42, B–11

I

I18N installation, *InstallGuide* 9–1
I18N software subset descriptions, *InstallGuide* E–1
I18N update installation, *InstallGuide* 10–1
I/O, *CommandShellGde* 6–1
  access to by applications, *DevDriverVMEbus* 2–8
I/O address
  defined, *DevDriverTutorial* 2–17
  description of address type that device drivers use, *DevDriverTutorial* 2–17

I/O handle
See also io_handle_t data type
defined, *DevDriverTutorial* 2–14
description, *DevDriverTutorial* 18–20
I/O subsystem, *TechOver* 6–1
ICCC compliance, *CDE_I18N_Prog* 122, 124
ICMP, *TechOver* 3–15
Icon Editor (dticon)
using, *CDE_User* 269
icons
CDE conventions and standards for, *CDEStyle* 39, 41, 43, 44, 46, 48, 49, 51, 52, 54, 56, 57, 58, 60, 70
creating for the desktop, *CDEAdvUsr* 201
displaying, *CDE_User* 145
in File Manager, *CDE_User* 108, 112
selecting, *CDE_User* 86, 87
iconv, *CDE_I18N_Prog* 56, 60
ID names
rules for in CDE help, *CDEHelp* 44
identifier, *DECC* 1-6
identifiers, *displayPS* 2–4
IEEE floating-point
See floating-point range and processing
ihandler_t structure
associated include file, *DevDriverReference* 3–53
list of member names and data types, *DevDriverReference* 3–53
ilogb routine, *DPML* DPML–44, B–12
Image Viewer
viewing file, *CDECompanion* 5–7
immediate mode swapping, *SystemAdmin* 6–5
include files
See header files
description of code example for /dev/dmaex driver, *DevDriverVMEbus* 6–3 to 6–4
include files (cont.)
section, *DevDriverTURBOchannel* 6–6 to 6–7
Include Files Section
for /dev/dmaex device driver, *DevDriverVMEbus* 6–3
including files, *DECC* 8-11
incoming events
for tracking by programs, *NetProgGde* 3–16
increment LED section, *DevDriverTURBOchannel* 6–75 to 6–76
index
creating in CDE help, *CDEHelp* 82
Industry Standard Architecture
See ISA bus
inetd daemon, *NetProgGde* 4–54
rpcgen support for, *OncRpcProgram* 2–17
using to start an RPC server, *OncRpcProgram* 3–25
init command, *SystemAdmin* 2–10, 2–17, 2–2
init directory structure, *SystemAdmin* 3–8
init.d directory, *SystemAdmin* 3–2
initial open, *STREAMSProg* 5–5, 9–1, 10–3, 10–4, 11–2
initialization file, *Ladebug* 15-1
initialization of applications
in CDE, *CDEProgOverview* 34
initialization phase
state transitions allowed, *NetProgGde* 3–17t
initialization scripts, *SystemAdmin* 2–2
initialization tasks, *SystemAdmin* 2–2
inittab file, *SystemAdmin* 2–10, 2–2, 3–2, 3–4, 3–8
initdefault entry, *SystemAdmin* 2–11
inkblk, *STREAMSProg* 11–3, 11–11, 11–13

input method, *International* 6–7
handling in X application, *International* 5–21
starting server for, *International* 6–9
X resource for, *International* 6–8
input methods, *CDE_I18N_Prog* 13, 49, 87, 88, 90, 117, *CDECompanion* 5–8
input models
CDE conventions and standards for, *CDEStyle* 3, 147
input/output
See I/O
install icon, *CDE_User* 62, 71, 72, 75
install update
See update installation
installation, *Security* 7–1
disk space required, *InstallGuide* F–13
full, *InstallGuide* 5–1
worldwide, *InstallGuide* 9–1
installation cloning, *InstallGuide* 1–16
installation of applications
CDE conventions and standards for, *CDEStyle* 95, 223
installation package in CDE
preparing, *CDEHelp* 243 to 249
installing software
from a server, *SharingSW* 1–2
installing software subsets after the installation, *InstallGuide* 8–1
installupdate program, *InstallGuide* 2–5
instructions
disassembling using kdbx, *KernelDebug* 2–20
interclient communications conventions for localized text, *CDE_I18N_Prog* 122
intermediate hosts used in file transfers (UUCP), *CommandShellGde* 14–14
international text drawing, *CDE_I18N_Prog* 78, 79
international text input, *CDE_I18N_Prog* 88 to 90
internationalization, *CDE_I18N_Prog* 1 to 5, 11, 1-1, 111, 113, 114, 122, *CDEAdvUsr* 276, 278, 279, 282, *SystemAdmin* 3–18, *TechOver* 10–1
See also I18N
asort, *TechOver* 10–7
creating locales, *TechOver* 10–7
DECwindows X clients, *TechOver* 10–10
handling cultural data in programs, *International* 2–13
handling text presentation and input, *International* 2–17
implications for software development, *International* 1–1
mail, *TechOver* 10–8
Motif widgets, *TechOver* 10–9
of CDE applications, *CDEProgOverview* 34, 54
of CDE help, *CDEHelp* 253 to 265
printing, *TechOver* 10–7
supported languages, *TechOver* 10–2
user-defined characters, *TechOver* 10–8
X/Open WPI interfaces, *International* A–1
internationalization features
using, *CommandShellGde* C–1
internationalization standards, *CDE_I18N_Prog* 4
internationalized User Interface Language (UIL), *CDE_I18N_Prog* 92
internationalizing and localizing dtksh scripts, *Desktopksh* 35

Internet, *NetAdmin* 8–15
  See also TCP/IP
  host requirements, *TechOver* A–1
Internet Control Message Protocol
  See ICMP
Internet Protocol
  See IP
interoperability problems
  solvable with ToolTalk, *ToolTalkMsg* 3 to 4
interrupt, *STREAMSProg* 9–1 to 9–11, 11–7, 12–8, 13–1 to 13–4, C–4, C–15
interrupt handler
  defined, *DevDriverTutorial* 2–17
Interrupt Section, *DevDriverTURBOchannel* 6–77 to 6–78
  description of code example for /dev/dmaex driver, *DevDriverVMEbus* 6–32 to 6–33
interrupt vectors, *DevDriverPci* 2–3
invocation procedures, *CallStandard* 7–1
I/O subsystems
  components
    file systems, *SystemTune* 1–15
  description, *SystemTune* 1–14
io_copyin kernel interface
  explanation of code fragment, *DevDriverTutorial* 18–22
io_copyio kernel interface
  explanation of code fragment, *DevDriverTutorial* 18–26
io_copyout kernel interface
  explanation of code fragment, *DevDriverTutorial* 18–24
io_handle_t data type
  description of data type that device drivers use, *DevDriverTutorial* 2–14
iocblk, *STREAMSProg* 9–13
ioctl call, *STREAMSProg* 9–12, 9–14
ioctl commands, *ATMProgGde* 8–3
ioctl section, *DevDriverTURBOchannel* 6–65 to 6–74
ioctls
  general processing, *STREAMSProg* 9–16, 9–17
  transparent processing, *STREAMSProg* 9–19, 9–36
iodone kernel interface
  explanation of code fragment, *DevDriverTutorial* 18–42
IOPORT, *DevDriverEisa* B–5
IP address, *NetAdmin* 2–4
IP multicasting, *NetProgGde* 4–46, *TechOver* 3–13
IP router, *NetAdmin* 2–8 to 2–15
IPC, *ProgGde* 1–6
  description, *SystemTune* 1–13
ISA bus, *TechOver* 6–4
isnan routine, *DPML* DPML–40, B–11
ISO
  standards and application development considerations, *ProgGde* 1–2
ISO C Standard, *DECC* XIV
ISO EUC, ISO646-IRV, ISO8859, ISO8859-1 codesets, *CDE_I18N_Prog* 66, 67
item_list structure
  associated include file, *DevDriverReference* 3–55
  list of member names and data types, *DevDriverReference* 3–55

J

j0 routine, *DPML* DPML–12, B–12

j1 routine, *DPML* DPML–12, B–12

jn routine, *DPML* DPML–12, B–12

K

kdbx debugger

command aliases, *KernelDebug* 2–14

command syntax, *KernelDebug* 2–11

compiling custom extension for, *KernelDebug* 3–40

debugging extensions of, *KernelDebug* 3–41

description of, *KernelDebug* 2–11

extensions, *KernelDebug* 2–15

library functions, *KernelDebug* 3–2

writing extensions for, *KernelDebug* 3–1

kdebug debugger, *Ladebug* 22-14

description of, *KernelDebug* 2–39

invoking, *KernelDebug* 2–43

problems with setup of, *KernelDebug* 2–45

requirements for, *KernelDebug* 2–40

setting up, *KernelDebug* 2–41

KDM controller, *TechOver* 6–14

kernel

building, *InstallGuide* 5–31

configuration file entries, *SystemAdmin* 4–30

configuring with options menu, *SystemAdmin* 4–20

dynamic configuration, *SystemAdmin* 4–4

enabling realtime preemption, *InstallGuide* 6–7

relationship to device driver, *DevDriverTutorial* 1–8

static configuration, *SystemAdmin* 4–15

kernel address space

copying from with copyout kernel interface, *DevDriverTutorial* 18–12

kernel debugging, *Ladebug* 22-1

debugging customized tools, *KernelDebug* 3–41

developing customized tools, *KernelDebug* 3–1

overview, *KernelDebug* 1–1

using kdbx debugger, *KernelDebug* 2–11

using kdebug debugger, *KernelDebug* 2–39

using the dbx debugger, *KernelDebug* 2–2

kernel interface

commonly used by device drivers, *DevDriverTutorial* 18–1

EISA/ISA bus device driver, *DevDriverEisa* A–4

used by EISA/ISA bus device driver, *DevDriverEisa* 5–1

kernel level functions

STREAMS, *NetProgGde* 5–17 to 5–25

kernel physical address

defined, *DevDriverTutorial* 2–15

description of address type that device drivers use, *DevDriverTutorial* 2–15

kernel subsystems

configuring STREAMS drivers, *NetProgGde* 5–25

configuring STREAMS modules, *NetProgGde* 5–25

setting configuration variables, *SystemAdmin* 3–3

kernel thread

advantages of using in device drivers, *DevDriverAdvTopics* 6–1

asserting a block by calling assert_wait_mesg kernel interface

kernel thread (cont.)

asserting a block by calling assert_wait_mesg kernel interface (cont.)

DevDriverAdvTopics (cont.)

7–7 (cont.)

(cont.) , *DevDriverAdvTopics* 7–7

blocking by calling mpsleep kernel interface, *DevDriverAdvTopics* 7–11

defined, *DevDriverAdvTopics* 2–4

execution issues, *DevDriverAdvTopics* 6–3

issues related to device driver design, *DevDriverAdvTopics* 6–1

issues related to using, *DevDriverAdvTopics* 6–4

operations to be performed by, *DevDriverAdvTopics* 6–5

setting a timer on by calling thread_set_timeout kernel interface, *DevDriverAdvTopics* 7–22

starting a fixed priority kernel thread, *DevDriverAdvTopics* 7–5

starting by calling kernel_thread_w_arg kernel interface, *DevDriverAdvTopics* 7–2

terminating by calling thread_halt_self kernel interface, *DevDriverAdvTopics* 7–16

terminating by calling thread_terminate kernel interface, *DevDriverAdvTopics* 7–16

unblocking by calling thread_wakeup kernel interface, *DevDriverAdvTopics* 7–16

unblocking by calling thread_wakeup_one kernel interface, *DevDriverAdvTopics* 7–16

kernel_isrthread kernel interface

explanation of code fragment, *DevDriverAdvTopics* 7–5

kernel-related interface

category of kernel interface, *DevDriverTutorial* 18–17

kernel_thread_w_arg kernel interface

explanation of code fragment, *DevDriverAdvTopics* 7–3

kernel-unmapped virtual address

defined, *DevDriverTutorial* 2–14

description of address type that device drivers use, *DevDriverTutorial* 2–14

key bindings, *CDEAdvUsr* 252 to 254

keyboard, *DWUser* A–1

basic skills, *CDE_User* 3

DECterm window’s emulation of, *International* 6–5

LK-style with PCXAL, *CDECompanion* 8–5

navigating with, *DWUser* A–1

PC-style with LK201/LK401, *CDECompanion* 8–5

using accelerators, *DWUser* 3–14

keyboard focus model

CDE conventions and standards for, *CDEStyle* 147

keyboard functions and key engravings

CDE conventions and standards for, *CDEStyle* 257 to 265

keyboard input

customizing in localization, *CDE_I18N_Prog* 121

keyboard mappings for space bar and enter key

CDE conventions and standards for, *CDEStyle* 267

keyboard mnemonics, *DWUser* 3–14
keyboard options, *CDECompanion* 5–9
keyboard shortcuts, *CDE_User* 283
keycaps, *CDECompanion* 5–11
keysyms, *CDE_I18N_Prog* 121
keyword, *DECC* 1-8
killall command, *SystemAdmin* 2–17
kit
  See subsets
kmknod
  defined, *DevDriverTutorial* 12–15
Korn shell, *CommandShellGde* 7–1, 8–1
kreg
  defined, *DevDriverTutorial* 12–15
kseg address
  See kernel-unmapped virtual address
.kshrc login script, *CommandShellGde* 7–17

L

labels
  implementing with App Builder, *AppBuilder* 138, 141, 144
ladebug command line, *Ladebug* REF-1
ladebug debugger, *TechOver* 7–2
LAN
  sharing software on, *SharingSW* 1–2
language announcement in programs, *International* 1–5
languages supported by Digital UNIX, *ProgGde* 1–4
lastcomm command, *SystemAdmin* 12–30
LAT, *NetAdmin* 5–1 to 5–17
layered product installation, *InstallGuide* 8–3
lazy binding, *AsmLangGde* 9–15
LBN, *SystemAdmin* 6–2
LBN (Logical Block Number), *SystemAdmin* 6–2
lcd subcommand (ftp), *CommandShellGde* 12–7
ld command
  using in System V habitat, *ProgGde* B–1
ld linker
  linking with shared libraries, *ProgGde* 4–7
ldexp routine, *DPML* DPML–45, B–12
leave statement, *ProgGde* 11–13
lex program, *ProgSupTools* 4–1 to 4–17
Lexicon, *DECC* 1-1
lgamma routine, *DPML* DPML–46, B–13
lgamma_max_float boundary value, *DPML* A–4
libexc
  exception library, *ProgGde* 11–1
.liblist section, *AsmLangGde* 9–23
libpthread.so, *ProgGde* 12–2
libraries
  shared, *ProgGde* 4–1
  specifying, *ProgGde* 2–20
  ucode, *ProgGde* D–6
  XTI and TLI, *NetProgGde* 3–6
library, *DECC* 9-1
library calls
  sockets, *NetProgGde* 4–10 to 4–15
library description files (lint), *ProgGde* 6–14
library routines
  for writing kdbx extensions, *KernelDebug* 3–2
libtli.a library, *NetProgGde* 3–6
libxti.a library, *NetProgGde* 3–6
license database
  for lmf, *LicenseManage* 1–2

License Management Facility, *LicenseManage* 1–1
License Management Facility commands, *LicenseManage* 2–1
license terms and conditions, *LicenseManage* 1–3
license units, *LicenseManage* 1–2
license-checking function, *LicenseManage* 1–2
limiting search paths, *ProgGde* 4–7
linkage, *DECC* 2-11
linkage conventions
  general, *AsmLangGde* 6–3
  memory allocation, *AsmLangGde* 6–17
linker
  See ld linker
linking
  dynamic linking, *AsmLangGde* 9–4
linking applications
  by using ld command, *ProgGde* 2–20
  in System V habitat, *ProgGde* B–1 to B–3
linking files, *CommandShellGde* 3–13
lint, *ProgGde* 6–1
  command syntax, *ProgGde* 6–1
  data type checking, *ProgGde* 6–4
  error messages, *ProgGde* 6–16
  increasing table size, *ProgGde* 6–11
  options, *ProgGde* 6–1
  warning classes, *ProgGde* 6–22
list boxes, *CDEStyle*
listen event, *NetProgGde* 3–16t
listing directory contents, *CommandShellGde* 3–2
.lita section, *AsmLangGde* 6–5
little endian
  See byte ordering
  byte ordering, *AsmLangGde* 1–2
lmf reference page, *LicenseManage* 1–2
lmf utility, *LicenseManage* 1–2
lmfsetup script, *LicenseManage* 1–2
ln command, *CommandShellGde* 3–13
ln routine, *DPML* DPML–47 B–13
loadable drivers, *Ladebug* 22-20
loader
  search path of, *ProgGde* 4–4
  use of dynamic loader, *AsmLangGde* 9–4
loading allocated system resources for DMA
  by calling dma_map_load kernel interface, *DevDriverTutorial* 18–31
loading programs, *AsmLangGde* 9–3
local area network
  See LAN
Local Area Transport (LAT), *Security* 3–5
local area transport support
  See LAT
local communication facility, messages
  sending, write command, *CommandShellGde* 11–21
local host control of file access
  See UUCP utility
local string table, *AsmLangGde* 8–5
local structure and variable definitions section, *DevDriverTURBOchannel* 6–12 to 6–15
local symbol table, *AsmLangGde* 8–4
locales, *International* 1–3, *SystemAdmin* 3–13
  and locale management, *CDE_I18N_Prog* 2, 21 to 23, 101, 102, 104
  binding to program's run-time environment, *International* 2–21
  building and testing, *International* 7–59
  creating, *International* 7–1
  introduction to, *CommandShellGde* C–1

locales (cont.)
multibyte and single byte, *CDEProgOverview* 54
setting, *International* 6–2

localization
implications for software development, *International* 1–3

localized catalog and resource file, *CDE_I18N_Prog* 126

localized desktop sessions
configuring, *CDEAdvUsr* 275
running, *CDE_User* 295

localized resources, *CDE_I18N_Prog* 45, 47 to 49

localized text, *CDE_I18N_Prog* 30, 31, 33, 34, 40, 41, 78

lock structure
associated include file, *DevDriverReference* 3–58
declared in code fragment, *DevDriverAdvTopics* 4–2

lock_data_t data type
to declare complex lock data structure, *DevDriverAdvTopics* 4–1

lock_done kernel interface
explanation of code fragment, *DevDriverAdvTopics* 4–12, 7–23
releasing previously asserted complex lock, *DevDriverAdvTopics* 4–10

lock_init kernel interface
explanation of code fragment, *DevDriverAdvTopics* 4–3
initializing a complex lock, *DevDriverAdvTopics* 4–2

lock_read kernel interface
explanation of code fragment, *DevDriverAdvTopics* 4–6

lock_terminate kernel interface
explanation of code fragment, *DevDriverAdvTopics* 4–22
terminating a complex lock, *DevDriverAdvTopics* 4–21

lock_try_read kernel interface
explanation of code fragment, *DevDriverAdvTopics* 4–15
trying to assert complex lock, *DevDriverAdvTopics* 4–13

lock_try_write kernel interface
explanation of code fragment, *DevDriverAdvTopics* 4–19
trying to assert complex lock, *DevDriverAdvTopics* 4–13

lock_write kernel interface
explanation of code fragment, *DevDriverAdvTopics* 4–9, 7–23

locking considerations
accessing a read-only resource, *DevDriverAdvTopics* 2–16
description of choices, *DevDriverAdvTopics* 2–6
description of complex locks, *DevDriverAdvTopics* 2–8
description of funnels, *DevDriverAdvTopics* 2–10
description of simple locks, *DevDriverAdvTopics* 2–6
device CSR addresses, *DevDriverAdvTopics* 2–16
driver-specific global resources, *DevDriverAdvTopics* 2–17
example analysis of device driver to determine resources to lock, *DevDriverAdvTopics* 2–19

locking considerations (cont.)
method choices, *DevDriverAdvTopics* 2–11
need for locks in device drivers operating in an SMP environment, *DevDriverAdvTopics* 2–4
resources to lock in a device driver, *DevDriverAdvTopics* 2–15
summary of locking methods, *DevDriverAdvTopics* 2–13
system-specific global resources, *DevDriverAdvTopics* 2–18
log routine, *DPML* DPML–47, B–13
log2 routine, *DPML* DPML–47, B–13
log10 routine, *DPML* DPML–47, B–14
log1p routine, *DPML* DPML–47, B–14
logarithm
ilogb, compute unbiased component, *DPML* DPML–43
logb, convert to doubple-precision floating-point, *DPML* DPML–48
logb routine, *DPML* DPML–48, B–14
logging in
See login
logging in to CDE session, *CDE_User* 36
logging out, *CommandShellGde* 1–4
logging out of CDE session, *CDE_User* 31, 36
logical block number
See LBN
logical data boundaries
and protocol independence, *NetProgGde* 3–42
Logical IP Subnet, *ATMProgGde* 2–2
Logical Storage Manager
See LSM
Logical Volume Manager
See LVM
login, *CommandShellGde* 1–1
.login file, *CDEAdvUsr* 25, 28, 33
.login login script
modifying for System V Habitat, *CommandShellGde* 9–2
Login Manager
and CDE session management, *CDEProgOverview* 8
configuring, *CDEAdvUsr* 1
login screen, *CDEAdvUsr* 6, 11 to 16
login script
.cshrc script, *CommandShellGde* 7–17
.kshrc script, *CommandShellGde* 7–17
.login script, *CommandShellGde* 7–17
.profile script, *CommandShellGde* 7–17
login servers, *CDEAdvUsr* 1 to 3, 5, 6, 8, 10, 19, 20, 94
login shell, *SystemAdmin* 9–9
longword operations
See performing read longword operations
See performing write longword operations
lower stream, *STREAMSProg* 11–2, 11–4, 11–7, 11–9
lpr command, *SystemAdmin* 10–19
lprsetup
running, *SystemAdmin* 10–9
lptest command, *SystemAdmin* 10–13
ls (list directory contents) command, *CommandShellGde* 3–2
ls subcommand (ftp), *CommandShellGde* 12–7
LSM, *SystemAdmin* 6–1, *TechOver* 4–9
commands, *StorageManager* 6–1, 7–1, 8–1, C–1
components, *StorageManager* 1–6
configuration definition, *SystemAdmin* 4–53
configuration limits, *StorageManager* 1–6

LSM (cont.)
configurations, *StorageManager* 2–1
disk group operations, *StorageManager* 12–1, 6–1, 9–17
disk operations, *StorageManager* 12–1, 6–1, 9–3, *SystemAdmin* 8–7
encapsulation process, *StorageManager* 4–1
error messages, *StorageManager* B–1
examples, *StorageManager* C–1
features, *StorageManager* 1–4, *SystemAdmin* 8–1
file system operations, *StorageManager* 12–17, 7–1
graphical interface (dxlsm), *StorageManager* 10–1, 11–1, 12–1
installation requirements, *StorageManager* 1–5
licensing requirements, *StorageManager* 1–5
menu interface (voldiskadm), *StorageManager* 9–1
mirrored volume, *StorageManager* 2–6
mirroring root, *StorageManager* 5–1
mirroring swap, *StorageManager* 5–1
monitoring, *SystemTune* 2–16
overview, *StorageManager* 1–1
performance tuning, *StorageManager* 15–1
plex operations, *StorageManager* 12–12, 8–1
setup procedures, *StorageManager* 3–1
striped volume, *StorageManager* 2–4
subdisk operations, *StorageManager* 12–9, 8–1
troubleshooting, *StorageManager* B–1
vold daemon, *StorageManager* 1–8
voliod daemon, *StorageManager* 1–8
volume device driver, *StorageManager* 1–6
volume operations, *StorageManager* 12–7, 7–1, 9–20

LURT, *LicenseManage* 1–2
lvalue, *DECC* 2-18
LVM, *SystemAdmin* 6–1
configuration definition, *SystemAdmin* 4–52

M

m4 macro preprocessor, *ProgSupTools* 5–1 to 5–13
machine code debugging, *Ladebug* 18-1
macro
See m4 macro preprocessor
macro definition, *DECC* 8-2
macros
predefined, *ProgGde* 2–9
magic number, *ProgGde* 2–25
magnetic tape drive
adding, *SystemAdmin* 5–6
testing with tapex, *SystemAdmin* 13–11
Mail, *DWUser* 5–41, 5–46
mail, *NetAdmin* 11–1 to 11–20
converting to CDE format, *CDECompanion* 10–1 to 10–6
differences between dtmail and dxmail, *CDECompanion* A–1 to A–4
enhancements for Asian languages, *International* 6–76
folders, *CommandShellGde* 11–12
the MH program, *CommandShellGde* 11–17, 11–21
using Mail, *CommandShellGde* 11–1 to 11–17
using under DECwindows Motif, *DWUser* 5–41
variables, *CommandShellGde* 11–16 to 11–17

mail (dtmail) basic interchange, *CDE_I18N_Prog* 62
mailboxes
  in CDE, *CDE_User* 161 to 163
mailer (dtmail)
  and CDE application development, *CDEProgOverview* 6, 7, 10, 14, 16
  using, *CDE_User* 147
mailing calendar appointments, reminders, *CDE_User* 232, 233, 238
mailx, *CommandShellGde* 11–1 to 11–17
main panel, *CDE_User* 61, 66, 69, 74, 129
MainWindow layout
  CDE conventions and standards for, *CDEStyle* 73, 193
major kernel interface
  explanation of code fragment, *DevDriverTutorial* 8–2
make utility, *ProgSupTools* 7–1 to 7–22
man command, *CommandShellGde* 1–9
man page
  See reference pages
Management Information Base (MIB), *ATMProgGde* 5–5
mandatory software subsets, *InstallGuide* C–1
markup of CDE help files, *CDEHelp* 28, 101 to 178
max_float boundary value, *DPML* A–1
mbox file, *CommandShellGde* 11–10
memory
  allocating for ATM, *ATMProgGde* 3–8
  detecting leaks, *ProgGde* 7–1
  displaying contents in dbx, *ProgGde* 5–47
  exercising with memx, *SystemAdmin* 13–4
  relationship to device driver, *DevDriverTutorial* 16–2
memory (cont.)
  shared memory
    testing with shmx, *SystemAdmin* 13–5
  system memory, *SystemAdmin* 13–4
  zeroing with bzero kernel interface, *DevDriverTutorial* 18–10
memory access
  detecting uninitialized or invalid, *ProgGde* 7–1
memory address
  defined, *DevDriverTutorial* 2–17
  description of address type that device drivers use, *DevDriverTutorial* 2–17
memory block
  copying a memory block to I/O space with io_copyin kernel interface, *DevDriverTutorial* 18–21
  copying with io_copyio kernel interface, *DevDriverTutorial* 18–25
  copying with io_copyout kernel interface, *DevDriverTutorial* 18–23
Memory File System
  See MFS
memory locking
  realtime, *Realtime* 4–1
memory management
  components
    paging, *SystemTune* 1–9
    swapping, *SystemTune* 1–9
    UBC, *SystemTune* 1–12
    virtual memory, *SystemTune* 1–3
  overview, *SystemTune* 1–2
  types of caches, *SystemTune* 1–3
memory mapped file support
  See mmap

memx exerciser, *SystemAdmin* 13–4

menu bar layout

CDE conventions and standards for, *CDEStyle* 195

menus

basic skills, *CDE_User* 25, 26

CDE conventions and standards for, *CDEStyle* 87, 158, 209

manipulating with App Builder, *AppBuilder* 45 to 50, 51 to 52, 104, 138, 148 to 149

message

allocation, *STREAMSProg* 7–8

filter module declarations, *STREAMSProg* 7–5, 7–7

put procedure, *STREAMSProg* 7–9, 7–10

message block, *STREAMSProg* 4–11, 4–15, 7–1 to 2, 8–1 to 8–2

message catalogs, *SystemAdmin* 3–13

and CDE help, *CDEHelp* 258, 265

creating source files for, *International* 3–2

design and maintenance considerations, *International* 3–22

editing and translating, *International* 3–18

generating from message sources, *International* 3–18

message data structures, *NetProgGde* 5–18

message priority, *STREAMSProg* 4–4, 4–9, 4–11, 7–7, 8–1 to 8–2, B–1

message queue, *STREAMSProg* 4–11, 7–1 to 7–2, 8–1 to 8–2, B–1

realtime, *Realtime* 9–1

message queues and service procedures, *STREAMSProg* 8–1, 8–7

message sets, *ToolTalkMsg* 62, 64

message storage, *STREAMSProg* 7–1 to 7–2

message type, *STREAMSProg* 4–8 to 4–9, 7–1 to 7–8, 8–4 to 8–6, B–1

messages, *CDE_I18N_Prog* 41, 42, 44, 125 to 129, 131, 133 to 135

See messages (local communications)

CDE conventions and standards for, *CDEStyle* 127 to 134, 229

implementing with App Builder, *AppBuilder* 58 to 60

realtime, *Realtime* 9–1

messages (local communications)

ending, end-of-file symbol (EOF), *CommandShellGde* 11–22, 11–23

long, in files, *CommandShellGde* 12–9

sending, write command, *CommandShellGde* 11–21

messages, IPC, *ProgGde* 1–6

messaging in ToolTalk

how to use, *ToolTalkMsg* 15 to 28

messaging toolkit, *ToolTalkMsg* 47 to 93

meta information in CDE help, *CDEHelp* 27, 41, 53, 71, 172

MFS file system, *SystemAdmin* 6–1, *TechOver* 4–7

mget subcommand (ftp), *CommandShellGde* 12–5

MH message handling program, *CommandShellGde* 11–17, 11–21

min_float boundary value, *DPML* A–2

mirrored disks

LSM, *StorageManager* 2–6, 5–1

miscellaneous interface

category of kernel interface, *DevDriverTutorial* 18–42

mixed-language program debugging, *Ladebug* 11-14, 12-2, 13-10

mkdir subcommand (ftp), *CommandShellGde* 12–7

mknod command, *SystemAdmin* 6–13

mmap, *TechOver* 7–13

mmap system call

shared libraries, *ProgGde* 4–17

mnemonic assignments for menu choices

CDE conventions and standards for, *CDEStyle* 265 to 267

mnemonics

CDE conventions and standards for, *CDEStyle* 8, 172, 183, 265

modem

configuring, *NetAdmin* 4–3 to 4–22

modf routine, *DPML* DPML–49, B–14

module

defined, *DevDriverTutorial* 13–6

Module Browser, *AppBuilder* 26, 128

module data structures, *NetProgGde* 5–17

module declarations, *STREAMSProg* 6–1

module flow control, *STREAMSProg* 5–7, 9–1, 9–3

module id, *STREAMSProg* 6–2 to 6–3

Module Management Interface, *ATMProgGde* 6–2

module name, *STREAMSProg* 5–6, 6–2 to 6–3, 9–5

modules

STREAMS processing routines for, *NetProgGde* 5–20

modules in an App Builder project

AppBuilder, *AppBuilder* 13 to 19, 58

moncontrol routine, *ProgGde* 8–25

monetary values

formatting rules in a locale, *International* 7–17

monitor routines

for controlling profiling, *ProgGde* 8–25

monitor_signal routine, *ProgGde* 8–25

monitoring performance history, *TechOver* 13–8

monitoring UUCP, *CommandShellGde* 14–32

monstartup routine, *ProgGde* 8–25

more command, *CommandShellGde* 3–6

Motif, *TechOver* 8–9, *XWinSysEnv* 1–1

and CDE application development, *CDEProgOverview* xiii, xv, 4, 13, 26, 33, 35, 37, 54, 65, 66, 71, 79 to 82

Motif applications

creating with dtksh, *Desktopksh* 11 to 18

Motif custom widgets

and CDE application development, *CDEProgGuide* 69 to 94

Motif dependencies in CDE internationalization, *CDE_I18N_Prog* 73

Motif Window Manager, *DWUser* 6–2

default Digital UNIX window manager, *XWinSysEnv* 1–1

mount command, *NetAdmin* 8–18, *SystemAdmin* 6–15

mouse

basic skills, *CDE_User* 2

CDE conventions and standards for, *CDEStyle* 6, 7, 29, 162, 163, 166, 269, 271 to 276

mouse dialog box, *CDE_User* 140

mouse operations and functions

CDE conventions and standards for, *CDEStyle* 269 to 276

moving files, *CommandShellGde* 3–22

mpsleep kernel interface

explanation of code fragment, *DevDriverAdvTopics* 7–13

mput subcommand (ftp), *CommandShellGde* 12–5

.msym section, *AsmLangGde* 9–20

multicasting, *NetProgGde* 4–46

multihead graphic support, *TechOver* 8–5

multilanguage programs

compiling, *ProgGde* 2–18

multimedia services, *InstallGuide* D–14

multiplex, *STREAMSProg* 3–1

multiplexing

configurations, *STREAMSProg* 11–1, 11–4

driver, *STREAMSProg* 11–8, 11–16

multiplexor

building, *STREAMSProg* 3–4, 3–13

configurations, *STREAMSProg* 3–1, 3–3

dismantling, *STREAMSProg* 3–14, 3–15

routing data through, *STREAMSProg* 3–16

multiprocess debugging, *Ladebug* 20-1

multithreaded applications

debugging, *Ladebug* 9-1

developing, *ProgGde* 12–1

profiling, *ProgGde* 8–29

multithreaded programming, *DECthreads* 1–1

multiuser boot, *SystemAdmin* 2–2

multiuser collaboration

in CDE application development, *CDEProgOverview* 6

N

name resolution

semantics, *ProgGde* 4–5

name services configuration file

See svc.conf file

name space, *DECC* 2-19

Name/Finger Protocol

See FINGER

name.kit file

defined, *DevDriverTutorial* 13–12

NAME.list file

defined, *DevDriverTutorial* 13–2

naming conventions

CDE desktop, *CDEProgOverview* 32

shared libraries, *ProgGde* 4–2

naming conventions, Client Library, *displayPS* 5–4

naming services, *TechOver* 3–27

national language support in CDE, *CDE_I18N_Prog* 8, 11, 13, 19, 20, 92, 104, 111

National Language Support (NLS), *SystemAdmin* 3–13

navigating with keyboard, *DWUser* A–1

navigation

CDE conventions and standards for, *CDEStyle* 4, 5, 150, 154, 158, 161

netconfig application, *NetAdmin* 2–10

Netscape Navigator

invoking to read documentation, *InstallGuide* 6–8

netstat command, *NetAdmin* 2–18, B–1

network

and LAT configuration, *NetAdmin* 5–10

configuring, *NetAdmin* 2–11 to 2–17

configuring the desktop in, *CDEAdvUsr* 85

network (cont.)
- groups in, *NetAdmin* 8–5
- loopback configuration file definition, *SystemAdmin* 4–54
- overview, *NetAdmin* 1–1

network adapters, *TechOver* 3–16

network device
- defined, *DevDriverTutorial* 16–6
- discussion of, *DevDriverTutorial* 16–6

network device driver
- defined, *DevDriverTutorial* 16–6
- introductory discussion of, *DevDriverTutorial* 1–2

Network File System
- *See* NFS

Network Information Service, *TechOver* 3–28
- *See* NIS

network problems, *NetAdmin* 13–3, 16–1

network programming environment, *TechOver* 3–18t
- application programming interfaces, *TechOver* 3–18t
- communication bridges, *TechOver* 3–18t
- components, *TechOver* 3–18t
- data link interface, *TechOver* 3–18t
- data link interfaces, *NetProgGde* 1–3, *TechOver* 3–21

network protocols
- configuration file definition, *SystemAdmin* 4–48

network security concerns, *Security* 4–2

network services access point
- *See* NSAP

network subsystems
- description, *SystemTune* 1–17
- tuning, *SystemTune* 3–30

Network Time Protocol
- *See* NTP

network-transparent events
- using ToolTalk to implement, *ToolTalkMsg* 4

NetWorker SingleServer, *InstallGuide* D–10, *SystemAdmin* 11–2

networking, *TechOver* 3–1

networks and internationalization, *CDE_I18N_Prog* 55

.new.. files, *InstallGuide* 2–3

newfs command, *SystemAdmin* 6–14

nextafter routine, *DPML* DPML–50, B–14

NFS, *SystemAdmin* 6–1, 6–2, *TechOver* 4–3
- managing, *NetAdmin* 8–1 to 8–24
- TCP, *TechOver* 3–9
- tuning, *SystemTune* 3–31 to 3–33
- UDP, *TechOver* 3–9

nfsconfig application, *NetAdmin* 8–7

nint routine, *DPML* DPML–51, B–15

NIS
- account management, *Security* 9–3
- managing, *NetAdmin* 7–1 to 7–22

nm command, *ProgGde* 2–24

NMAGIC files, *AsmLangGde* 7–6
- segment access permissions, *AsmLangGde* 9–2

nonblocking mode, *NetProgGde* 3–5

noninteractive editing
- *See* sed editor

Notepad, *DWUser* 5–47, 5–54

NSAP, *ATMProgGde* 3–5

NTP, *TechOver* 3–29
- managing, *NetAdmin* 10–2 to 1-–11

ntp command, *NetAdmin* 10–2

ntpd daemon, *NetAdmin* 10–11

ntpdate command, *NetAdmin* 10–11

ntpdc command, *NetAdmin* 10–11

ntpq command, *NetAdmin* 10–9

ntpsetup command, *NetAdmin* 10–6

null-terminated character string

comparison of with strcmp kernel interface, *DevDriverTutorial* 18–1

copying of with strncpy kernel interface, *DevDriverTutorial* 18–4, 18–5

returning with strlen kernel interface, *DevDriverTutorial* 18–7

numeric values

formatting rules in a locale, *International* 7–20

O

O_NDELAY value

support in TLI, *NetProgGde* 3–43

object file format, *AsmLangGde* 7–1

object file tools, *ProgGde* 2–23

object file types

demand paged (ZMAGIC) files, *AsmLangGde* 7–24

impure format (OMAGIC) files, *AsmLangGde* 7–21

shared text (NMAGIC) files, *AsmLangGde* 7–22

obtaining VMEbus address

with vba_get_vmeaddr, *DevDriverVMEbus* 5–7

ocnt variable

and incoming events, *NetProgGde* 3–16

and outgoing events, *NetProgGde* 3–14

odump (object file utility), *ProgGde* 2–24

OffTheSpot mode preedit area, *CDE_I18N_Prog* 15

-O* flags (cc -oldc), *ProgGde* D–3

OMAGIC files, *AsmLangGde* 7–21

segment access permissions, *AsmLangGde* 9–2

online help

writing for CDE applications, *CDEHelp* 1

open and close device section, *DevDriverTURBOchannel* 6–38 to 6–41

open call, *STREAMSProg* 5–1, 5–5 to 5–6, 6–2

open procedure, *STREAMSProg* 5–5 to 5–6, 6–1 to 6–2, 6–5 to 6–6

Open Shortest Path Routing

See OSPF

open subcommand (ftp), *CommandShellGde* 12–5

open systems

discussion of goals, *DevDriverTutorial* 15–1

OPENbus

defined, *DevDriverTutorial* 1–9

opened event, *NetProgGde* 3–15

operating system overview, *CommandShellGde* 1–1

operating system processes, *CommandShellGde* 6–1

operator, *DECC* 1-9, 6-1

operators, *displayPS* 6–1 to 6–5

optimization, *ProgGde* 2–19

effect on debugging, *Ladebug* 11-16, 12-11

post linking, *ProgGde* 10–7

optimization (cc -oldc)

compiler options for, *ProgGde* D–2

optimization symbol table, *AsmLangGde* 8–5

optimizer, global

See global optimizer (uopt)

option buttons

CDE conventions and standards for, *CDEStyle* 8, 243

optional software subsets, *InstallGuide* C–7

origin

See coordinate systems

See user space origin

OSF/Motif, *XWinSysEnv* 1–1

OSF_USR license, *LicenseManage* 2–2

OSPF, *TechOver* 3–8

out-of-band data

receiving, *NetProgGde* 4–45

sending, *NetProgGde* 4–45

outgoing events

in XTI, *NetProgGde* 3–15t

over-commitment mode swapping

See deferred mode swapping

overlapping disk partitions

checking for, *SystemAdmin* 6–30

newfs command, *SystemAdmin* 6–14

OverTheSpot mode preedit area, *CDE_I18N_Prog* 16

P

PAAM, *LicenseManage* 1–7

current business policy, *LicenseManage* 2–12

pac command, *SystemAdmin* 12–35

packet size, *STREAMSProg* 6–2 to 6–3

paging and swapping

methods for reducing, *SystemTune* 3–14 to 3–16

overview, *SystemTune* 1–9

PAK

sample PAK license fields, *LicenseManage* 1–8e

PALcode

instruction summaries, *AsmLangGde* D–1

palette

color, *CDE_User* 136 to 138

palette controls, panes, and windows, *AppBuilder* 118, 121, 124, 125

palette objects

dragging and dropping, *AppBuilder* 22

paned windows, *CDEStyle*

panels

CDE conventions and standards for, *CDEStyle* 244

panes

manipulating with App Builder, *AppBuilder* 6, 44, 138

partition

See disk partition

partition sizes, *InstallGuide* G–1

pass_conn event, *NetProgGde* 3–16

passive user

typical state transitions, *NetProgGde* 3–21f

passwd file

deleting entries from, *SystemAdmin* 9–17

distributing for mail, *NetAdmin* 11–5

password, *CommandShellGde* 1–6

assigning, *SystemAdmin* 9–10

file, *CommandShellGde* 5–1

secure, *Security* 2–3

PATH environment variable

setting for System V habitat, *CommandShellGde* 9–2

pathname, *CommandShellGde* 2–8

pathname conventions (UUCP), *CommandShellGde* 14–1

PC-style keyboard, *CDECompanion* 8–5

PCI bus, *DevDriverPci* 2–1, *TechOver* 6–3

pci_config_hdr structure

associated include file, *DevDriverPci* A–8

list of member names and data types, *DevDriverPci* A–8

pci_config_hdr structure (cont.)
member names and data types
PPB device, *DevDriverPci* 5–4t

pci_option structure
associated include file, *DevDriverPci* A–14
list of member names and data types, *DevDriverPci* A–14

pci_rom_data structure
associated include file, *DevDriverPci* A–21
list of member names and data types, *DevDriverPci* A–21

pci_rom_exp_header structure
associated include file, *DevDriverPci* A–19
list of member names and data types, *DevDriverPci* A–19

Performance Manager, *PerformManage* 1–1

performance monitors, *SystemAdmin* 3–22
real-time performance monitor, *SystemAdmin* 3–23

performance tuning
See also system tuning
using commands and scripts, *SystemAdmin* 3–23

performing access operations on complex lock, *DevDriverAdvTopics* 4–4

performing read longword operations
with read_io_port, *DevDriverVMEbus* 5–8

performing read word operations
with read_io_port, *DevDriverVMEbus* 5–8

performing write longword operations
with write_io_port, *DevDriverVMEbus* 5–11

performing write word operations
with write_io_port, *DevDriverVMEbus* 5–11

peripheral device
defined, *DevDriverTutorial* 1–9
discussion of distinctions, *DevDriverTutorial* 16–4

peripheral device (cont.)
relationship to device driver, *DevDriverTutorial* 1–9

Peripheral Device Interconnect
See PCI bus

permanent virtual circuit
See PVC

permissions, *CommandShellGde* 5–1
changing, *CommandShellGde* 5–3

PFS, *TechOver* 4–7

physical point of attachment
PPA, *ATMProgGde* 3–5

physio kernel interface
explanation of code fragment, *DevDriverTutorial* 18–42

pipe character, *CommandShellGde* 7–8

pixie (Atom tool), *ProgGde* 9–2

pixmaps
localizing, *CDE_I18N_Prog* 49

pixstats, *ProgGde* 8–14

platform-independent applications
using ToolTalk with, *ToolTalkMsg* 4

plug-and-play
implementing with ToolTalk, *ToolTalkMsg* 3

point-to-multipoint connection
See connection

point-to-point connection
See connection

point-to-point protocol, *InstallGuide* 5–35

pointer size
conversion, *ProgGde* A–1

poll, *STREAMSProg* 2–1 to 2–5

POLYCENTER Advanced File System
See also AdvFS
choosing during installations, *InstallGuide* 5–19

POLYCENTER NetWorker Save and Restore, *SystemAdmin* 11–2
pop-up menu, *CDE_User* 26, 30, 64, 66, 89, 100
CDE conventions and standards for, *CDEStyle* 205
PORT, *DevDriverEisa* B–18
port structure
associated include file, *DevDriverReference* 3–59
defined, *DevDriverTutorial* 17–15
list of member names and data types, *DevDriverReference* 3–59
portability
bit fields, *ProgGde* 6–12
external names, *ProgGde* 6–12
portability and maintenance of CDE applications, *CDEProgOverview* 37
Portable Character Set, *International* 1–6
porting applications to XTI, *NetProgGde* 3–41 to 3–47
porting Digital UNIX device drivers, *DevDriverTutorial* 4–6
porting ULTRIX device drivers, *DevDriverTutorial* 4–1
POSIX
and CDE application development, *CDEProgOverview* xv, 38, 54
standards and application development considerations, *ProgGde* 1–2
POSIX standards
realtime, *Realtime* 1–17
PostScript interpreter, *displayPS* 2–1
PostScript language encoding
See encoding, PostScript language
PostScript language imaging, *displayPS* 1–1
pow routine, *DPML* DPML–52, B–15
power management, *SystemAdmin* 3–23 to 3–26
PPA, *ATMProgGde* 3–5
permanent virtual circuit, *ATMProgGde* 3–7
switched virtual circuit, *ATMProgGde* 3–7
PPP, *NetAdmin* 4–17 to 4–19
See point-to-point protocol
pragma
pointer_size, *ProgGde* A–2
pragma directive, *DECC* 8-12
pragma preprocessor directives, *ProgGde* 3–1
prdaily shell script, *SystemAdmin* 12–41
predefined macros, *DECC* 8-13
preediting in CDE internationalization, *CDE_I18N_Prog* 15, 16, 18, 90
preparing for installation
hardware, *InstallGuide* 4–2
preprocessing, *DECC* 2-20, 8-2
preprocessor directive, *DECC* 2-20, 8-2
preserved message buffer
examining with the dbx debugger, *KernelDebug* 2–8
presto command
description of, *Prestoserve* 3–3
prestoctl_svc daemon
description of, *Prestoserve* 2–12
Prestoserve, *TechOver* 4–11
adding kernel support, *Prestoserve* 2–4
creating generic control device, *Prestoserve* 2–10
improving NFS server performance, *Prestoserve* 1–6
pseudodevice driver, *Prestoserve* 1–2, 3–1
recovering from failures, *Prestoserve* 4–1

Prestoserve (cont.)
speeding up synchronous writes, *Prestoserve* 1–1
prestosetup command
setting up Prestoserve, *Prestoserve* 2–6
print integration
in CDE applications, *CDEProgGuide* 5 to 7, 10 to 12
Print Manager, *CDECompanion* 3–2
print screen, *CDECompanion* 5–13
Print Screen, *DWUser* 5–59, 5–63
print services
See printer
printer, *SystemAdmin* 10–1
device special file name, *SystemAdmin* 10–6
lpc command, *SystemAdmin* 10–16
using with LAT, *NetAdmin* 5–10
printf kernel interface
explanation of code fragment, *DevDriverTutorial* 18–17
printing, *CDEHelp* 10, 98, 240, 241
and Print Manager, *CDE_User* 175
configuring and administrating from the desktop, *CDEAdvUsr* 107
files, *CommandShellGde* 3–10
reference pages, *CommandShellGde* 1–9
subsystem enhancements for Asian languages, *International* 6–60
priority, *STREAMSProg* 4–4, 4–9, 4–11, 4–15
realtime, *Realtime* 2–1
priority inversion
defined, *DevDriverAdvTopics* 6–4
issue to consider when using kernel threads, *DevDriverAdvTopics* 6–4
/proc File System
See PFS
procedure
invoking, *CallStandard* 7–1
procedure descriptor table, *AsmLangGde* 8–13, 8–3
process information, *CommandShellGde* 6–14
process management, *SystemTune* 1–2
processes, *CommandShellGde* 6–1
initialization, *SystemAdmin* 2–2
processing text files
See awk
See m4 macro preprocessor
See sed editor
Product Authorization Key
See PAK
.product.list file
defined, *DevDriverTutorial* 13–2
PROFDIR
profiling environment variable, *ProgGde* 8–23
.profile file, *CDEAdvUsr* 25, 28, 33
.profile login script, *CommandShellGde* 7–17
modifying for System V Habitat, *CommandShellGde* 9–2
profiling, *ProgGde* 8–1
using Atom tools, *ProgGde* 9–2
program checking
C programs, *ProgGde* 6–1
program loading, *AsmLangGde* 9–3
program number assignment, *OncRpcProgram* 1–5
programmer-defined routines
introduction to CAM, *SCSIDevDrivers* 11–1
programmer-defined structures
introduction to CAM, *SCSIDevDrivers* 11–1
samples of, *SCSIDevDrivers* 11–13

programming CDE help
tasks in, *CDEHelp* 19 to 21
programming examples
for DLI, *NetProgGde* E–23
programming for international use, *CDE_I18N_Prog* 92 to 94, 104, 111, 125
project
in App Builder, *AppBuilder* 8 to 13, 19, 102
property editor in App Builder, *AppBuilder* 32, 91, 92, 140 to 148, 150 to 154
.proto.. files, *InstallGuide* 2–4
protocol independence
for XTI applications, *NetProgGde* 3–41
protocol-specific options
and protocol independence, *NetProgGde* 3–42
prototype, *DECC* 2-5, 2-10, 5-5, 5-8, 5-10
prtacct shell script, *SystemAdmin* 12–41
pseudo device driver
defined, *DevDriverTutorial* 2–3
pseudodevice driver
introductory discussion and examples of, *DevDriverTutorial* 1–3
Prestoserve, *Prestoserve* 3–1
pseudoterminals, *NetProgGde* 4–59
BSD STREAMS-based, *SystemAdmin* 5–4
clist-based, *SystemAdmin* 5–4
creating device special files, *SystemAdmin* 5–5
pswrap translation program, *displayPS* 2–2
pt_tbl structure
associated include file, *DevDriverReference* 3–60
definition of members, *DevDriverReference* 3–60
public and private interfaces
and CDE application development, *CDEProgOverview* 33
public directory (UUCP), *CommandShellGde* 14–1
punctuator, *DECC* 1-10
push, *STREAMSProg* 5–6, 6–2, 6–5, 8–11, A–2, D–1
push buttons
CDE conventions and standards for, *CDEStyle* 244
pushable modules, *STREAMSProg* 5–3, 5–5 to 5–6, 8–2, E–4, E–7
put procedure, *STREAMSProg* 6–2, 9–3, 9–6, 9–8 to 9
put subcommand (ftp), *CommandShellGde* 12–5
putting a new bus address/byte count pair in the list
by calling dma_put_curr_sgentry kernel interface, *DevDriverTutorial* 18–38
by calling dma_put_prev_sgentry kernel interface, *DevDriverTutorial* 18–38
PVC
configuring, *ATMProgGde* 2–1
creating, *ATMProgGde* 7–9
PPAs, *ATMProgGde* 3–7
pwck command, *SystemAdmin* 9–11
pwd subcommand (ftp), *CommandShellGde* 12–7

Q

queue structure, *STREAMSProg* A to 2
queue_t structure, *STREAMSProg* 8–1
quick help buttons, *CDEHelp* 230

quick help dialog
creating, *CDEHelp* 211
quickstart, *AsmLangGde* 9–22
See shared libraries
section ordering constraints, *AsmLangGde* 9–24
troubleshooting
fixso, *ProgGde* 4–14
manually, *ProgGde* 4–12
using, *ProgGde* 4–10
quit subcommand (ftp), *CommandShellGde* 12–5
quit subcommand (telnet), *CommandShellGde* 13–4
quotas
AdvFS, *SystemAdmin* 7–2
setting UFS quotas, *SystemAdmin* 9–13
UFS, *SystemAdmin* 6–23

R

race condition
defined, *DevDriverAdvTopics* 6–4
issue to consider when using kernel threads, *DevDriverAdvTopics* 6–4
radio buttons
CDE conventions and standards for, *CDEStyle* 244
RAID subsystems, *TechOver* 6–11, 6–7
LSM support for, *StorageManager* 1–5
random routine, *DPML* DPML–53, B–15
rc directory structure, *SystemAdmin* 3–8
rc0 script, *SystemAdmin* 3–2
rc2 script, *SystemAdmin* 3–2
rc3 script, *SystemAdmin* 3–2
rcmgr command, *SystemAdmin* 3–3
rcn.d directory, *SystemAdmin* 3–2
RCS, *InstallGuide* D–10, *ProgSupTools* 6–1 to 6–34
rcv event, *NetProgGde* 3–16t
rcvconnect event, *NetProgGde* 3–16t
rcvdis1 event, *NetProgGde* 3–16t
rcvdis3 event, *NetProgGde* 3–16t
rcvrel event, *NetProgGde* 3–16t
rcvudata event, *NetProgGde* 3–16t
rcvuderr event, *NetProgGde* 3–16t
read and write device section, *DevDriverTURBOchannel* 6–42 to 6–50
read byte operation
See reading data from a device register
read call, *STREAMSProg* 1–1, 1–3, 4–3, 6–5, 7–3
read-only access
support in TLI, *NetProgGde* 3–43
reading data from a device register
with read_io_port, *DevDriverVMEbus* 5–8
realtime
asynchronous I/O, *Realtime* 7–1
clocks, *Realtime* 6–1
memory locking, *Realtime* 4–1
message queues, *Realtime* 9–1
messages, *Realtime* 9–1
overview, *Realtime* 1–2
POSIX standards, *Realtime* 1–17
priorities, *Realtime* 2–1
scheduling, *Realtime* 2–1
semaphores, *Realtime* 8–1
shared memory, *Realtime* 3–1
timers, *Realtime* 6–1
realtime kernel, *InstallGuide* 6–7
rebuilding the kernel, *InstallGuide* 8–9

receiving files (UUCP), *CommandShellGde* 14–21

recompiling TLI programs, *NetProgGde* 3–43

recv subcommand (ftp), *CommandShellGde* 12–5

redefining keys, *DWUser* 9–8

redirecting input/output, *CommandShellGde* 6–1

Redundant Array of Independent Disks
- *See* RAID

reference pages, *CommandShellGde* 1–9
- displaying in CDE, *CDE_User* 56, 57
- linking to in CDE help, *CDEHelp* 64, 69, 220
- processing translated, *International* 6–80

registering applications, *CDE_User* 127, *CDEAdvUsr* 43, 54, 56 to 60, 62, 66 to 68, 75, 77 to 83
- help considerations, *CDEHelp* 247

registration package, *CDEAdvUsr* 43, 54, 47, 60, 72 to 74, 77

regular expression, *ProgSupTools* 1–1 to 1–9

.rel.dyn section, *AsmLangGde* 9–19
- ordering for quickstart, *AsmLangGde* 9–24

relative file descriptor table, *AsmLangGde* 8–7

releasing system resources for DMA
- by calling dma_map_dealloc kernel interface, *DevDriverTutorial* 18–35

releasing VMEbus address space
- *See* unloading DMA resources

remainder routine, *DPML* DPML–54, B–15

remote boot
- flow, *SharingSW* 6–4

remote commands
- running in UUCP, *CommandShellGde* 14–14

remote debugger server
- writing, *Ladebug* B-1

remote debugging, *Ladebug* 12-1

remote execution, *CDEAdvUsr* 100, 104, 177, 282

remote file transfers (UUCP), *CommandShellGde* 14–14

remote host
- running commands from (UUCP), *CommandShellGde* 14–18

Remote Installation Service (RIS), *InstallGuide* 1–3

Remote Installation Services
- *See* RIS

remote kernel debugging, *KernelDebug* 2–39, *Ladebug* 22-14

remote login, *CommandShellGde* 13–1, 13–3

remote procedure call, *OncRpcProgram* 1–1

removing files, *CommandShellGde* 3–27

removing software subsets, *InstallGuide* 8–1

rename subcommand (ftp), *CommandShellGde* 12–7

renaming files and directories, *CommandShellGde* 3–22

repainting screen with xrefresh, *DWUser* C–2

Request for Comments
- *See* RFC

resfd variable
- and outgoing events, *NetProgGde* 3–14

resolution of symbols
- shared libraries, *ProgGde* 4–3

resource, *CDEAdvUsr* 25, 29, 33, 256, 261, 271 to 273, 279
- defined, *DevDriverAdvTopics* 2–4
- for CDE help, *CDEHelp* 75, 208, 217, 220, 237, 261

resource (cont.)
Motif widget variables
supported by dtksh, *Desktopksh* 12 to 13
resource files, *DWUser* 9–1 to 9–8
restoration
of AdvFS file systems, *InstallGuide* 7–8
of file systems in UNIX shell, *InstallGuide* 7–6
of file systems with NetWorker SingleServer, *InstallGuide* D–10
of UFS file systems, *InstallGuide* 7–6
procedures, *SystemAdmin* 11–1
Restricted Bourne shell, *CommandShellGde* 7–1
return values of dtksh functions, *Desktopksh* 16
returning a kernel segment address
by calling dma_kmap_buffer kernel interface, *DevDriverTutorial* 18–40
returning a pointer to sg_entry structure
by calling dma_get_curr_sgentry kernel interface, *DevDriverTutorial* 18–36
by calling dma_get_next_sgentry kernel interface, *DevDriverTutorial* 18–36
returning to local host during remote connection (UUCP), *CommandShellGde* 14–14, 14–8
Revision Control System
See RCS
RFC
1122 and 1123
configuring to comply with, *TechOver* A–11 to A–13
Digital UNIX conformance, *TechOver* A–1 to A–13
defined, *TechOver* A–1
rint routine, *DPML* DPML–55, B–16
RIP, *TechOver* 3–7
RIS, *SharingSW* 1–2
architecture of, *SharingSW* 2–1 to 2–4
remote booting, *SharingSW* 6–1, 6–4
system components, *SharingSW* 1–2
RIS area
contents of, *SharingSW* 2–2
ris utility, *SharingSW* 1–2
rlogin command, *CDE_User* 131, *CommandShellGde* 13–1
rmdir subcommand (ftp), *CommandShellGde* 12–7
root file system
restoring damaged, *InstallGuide* 7–6
root mode preedit area, *CDE_I18N_Prog* 18
rounding mode
floating-point rounding modes, *AsmLangGde* 4–5 to 4–7
routing, *NetAdmin* 2–6
Routing Information Protocol
See RIP
routing protocols, *TechOver* 3–5
RPC, *OncRpcProgram* 1–1, 3–1
See also rpcgen
administration, *OncRpcProgram* 1–6
language, *OncRpcProgram* 2–22
RPC language
overview, *OncRpcProgram* 1–1
RPC programming, *OncRpcProgram* 3–1
rpcgen, *OncRpcProgram* 2–1
See also RPC
declarations, *OncRpcProgram* 2–24
programming with, *OncRpcProgram* 2–27
ruled lines
drawing on a DECterm window, *International* 4–16

run command scripts, *SystemAdmin* 2–10, 2–2
run levels
changing, *SystemAdmin* 2–11
defaults, *SystemAdmin* 3–4
identifying, *SystemAdmin* 2–10
multiuser, *SystemAdmin* 2–10
single-user, *SystemAdmin* 2–10
run-time errors
using dbx to isolate, *ProgGde* 5–4
run-time libraries, *TechOver* 7–11
runacct shell script, *SystemAdmin* 12–36
runique subcommand (ftp), *CommandShellGde* 12–5
running commands on a remote host (UUCP), *CommandShellGde* 14–18
ruptime command, *CommandShellGde* 10–5
rvalue, *DECC* 2-18

S

-S compilation option, *AsmLangGde* 6–13
.s files, *AsmLangGde* 6–13
sa command, *SystemAdmin* 12–26
sashes
CDE conventions and standards for, *CDEStyle* 245
savecore command, *KernelDebug* 4–9
SBC
See AXPvme Single-Board Computers
scalar constants, *AsmLangGde* 2–2
scalb routine, *DPML* DPML–56, B–16
scales
CDE conventions and standards for, *CDEStyle* 246
SCCS, *ProgSupTools* 6–1 to 6–34
sccs command, *ProgSupTools* 6–18
scheduler, *STREAMSProg* 8–2, 8–4, C–12
scheduling
realtime, *Realtime* 2–1
scope, *DECC* 2-4
screen blanker
settings for, *CDE_User* 141
screen locks, *CDE_User* 143
screen saver
customizing, *CDE_User* 141, 143
screend, *TechOver* 3–2
scripts, *Ladebug* 15-1
scroll bars
CDE conventions and standards for, *CDEStyle* 248
SCSI, *SystemAdmin* B–1
device recognition tasks, *SystemAdmin* 5–1
SCSI bus, *TechOver* 6–8
SCSI device recognition, *SystemAdmin* 5–1
SCSI disk, *SystemAdmin* 6–20
SCSI/CAM special command table (CAM)
entries, *SCSIDevDrivers* 12–6
SCSI/CAM special I/O interface (CAM)
command table entries, *SCSIDevDrivers* 12–6
command tables, *SCSIDevDrivers* 12–5
device driver access, *SCSIDevDrivers* 12–3
SCU command, *SystemAdmin* B–1
device and bus maintenance, *SystemAdmin* B–10
device and bus managment, *SystemAdmin* B–7
format, *SystemAdmin* B–1
general purpose commands, *SystemAdmin* B–3
maintaining SCSI disks, *SystemAdmin* B–1
online help, *SystemAdmin* B–3

SCU command (cont.)
syntax conventions, *SystemAdmin* B–2
SDL (Semantic Delivery Language) in CDE help, *CDEHelp* 19, 87
search path
limiting, *ProgGde* 4–7
loader, *ProgGde* 4–4
shared libraries, *ProgGde* 4–4
search paths, *CDEAdvUsr* 27, 41, 114 to 116, 123, 125, 157, 204, 212
desktop, *CDEAdvUsr* 113
searching for text with grep, *ProgSupTools* 1–9
section data, *AsmLangGde* 7–10, 7–7
section headers
flags (s_flags), *AsmLangGde* 7–8
section name (s_name), *AsmLangGde* 7–7
section relocation information
assembler and linker processing, *AsmLangGde* 7–15
relocation entry, *AsmLangGde* 7–12
relocation table entry, *AsmLangGde* 7–15
secure devices, *Security* 8–1
security, *NetAdmin* 3–3, 4–13, 8–12, 11–18, *Security* xxi, *TechOver* 11–1
audit, *TechOver* 11–1
C2 functionality and TCSEC, *TechOver* 11–1
considerations, *CommandShellGde* 5–20
establishing, *SystemAdmin* 1–2
performance, *TechOver* 11–7
policy, *SystemAdmin* 3–21
sed editor, *ProgSupTools* 3–1 to 3–12
sel_queue structure
associated include file, *DevDriverReference* 3–61
list of member names and data types, *DevDriverReference* 3–61
select socket call
contrast to XTI t_look function, *NetProgGde* 3–46
select system call
using on the server side, *OncRpcProgram* 3–16 to 3–17
selection boxes
CDE conventions and standards for, *CDEStyle* 250
selection models
CDE conventions and standards for, *CDEStyle* 6, 7, 162, 163, 166 to 172
semantics
name resolution, *ProgGde* 4–5
semaphores, *ProgGde* 1–6
realtime, *Realtime* 8–1
send subcommand (ftp), *CommandShellGde* 12–5
sending
files to remote hosts (UUCP), *CommandShellGde* 14–14
files via UUCP, *CommandShellGde* 14–21
local messages, *CommandShellGde* 11–21
long messages, *CommandShellGde* 12–9
server, *CDEAdvUsr* 86, 87, 89, 93, 94, 101, *NetAdmin* 7–10, 15–1, 10–2, 8–8
process, *OncRpcProgram* 1–1
programming of, *OncRpcProgram* 2–20
sharing software with clients, *SharingSW* 1–2
server/client interaction, *NetProgGde* 4–8
service interface, *STREAMSProg* 4–1 to 4–6, 4–8, 4–15, 12–1 to 12–5
service procedures, *STREAMSProg* 8–2, 8–3
service provider, *STREAMSProg* 12–1 to 12–2
service user, *STREAMSProg* 4–1 to 4–4, 4–6 to 4–8, 4–12, 12–1 to 12–2

session, *CDEAdvUsr* 18, 23 to 25, 34 to 37
customizing in DECwindows Motif, *DWUser* 7–1
starting a desktop, *CDE_User* 35
Session Manager
and CDE application development, *CDEProgOverview* 8, 9, 34, 50, 51, 71
configuring, *CDEAdvUsr* 23
customizing in CDE, *CDECompanion* 8–1 to 8–10
customizing in DECwindows Motif, *DWUser* 7–1
integrating CDE applications with, *CDEProgGuide* 29 to 32
using under DECwindows Motif, *DWUser* 1–2
Session Manager save state notice
responding to with dtksh, *Desktopksh* 31
setenv command (dbx)
effect on debugger, *ProgGde* 5–12
setld command, *InstallGuide* 8–1
displaying software subset inventory, *InstallGuide* 8–2
setrgbXactual operator, *displayPS* 6–4
setting a timer on kernel thread
by calling thread_set_timeout kernel interface, *DevDriverAdvTopics* 7–22
setting allocated system resources for DMA
by calling dma_map_load kernel interface, *DevDriverTutorial* 18–31
setting up the system, *InstallGuide* 6–1
setup utilities, *TechOver* 12–3
setXgcdrawable operator, *displayPS* 6–4
setXgcdrawablecolor operator, *displayPS* 6–5
setXoffset operator, *displayPS* 6–5
sg_entry data structure
description, *DevDriverTutorial* 18–29
sg_entry structure
associated include file, *DevDriverReference* 3–62
list of member names and data types, *DevDriverReference* 3–62
shared executable files
dependencies, *AsmLangGde* 9–12
dynamic section, *AsmLangGde* 9–5
loading considerations, *AsmLangGde* 9–3
shared interrupt handler
defined, *DevDriverTutorial* 7–12
shared interrupt line
defined, *DevDriverTutorial* 7–12
shared libraries, *TechOver* 7–5
advantages, *ProgGde* 4–1
applications that cannot use, *ProgGde* 4–8
backwards compatibility, *ProgGde* 4–18
binary incompatibility, *ProgGde* 4–18
creating, *ProgGde* 4–8
debugging, *Ladebug* 16-1
debugging programs using, *ProgGde* 4–16
displaying in dbx, *ProgGde* 5–24
dynamic linking, *AsmLangGde* 9–4
linking with a C program, *ProgGde* 4–7
major version, *ProgGde* 4–21
minor version, *ProgGde* 4–21
mmap system call, *ProgGde* 4–17
multiple version dependencies, *ProgGde* 4–23
naming convention, *ProgGde* 4–2
overview, *ProgGde* 4–2
partial version, *ProgGde* 4–22
search path, *ProgGde* 4–4
support in XTI, *NetProgGde* 3–6

shared libraries (cont.)
symbol resolution, *ProgGde* 4–3
turning off, *ProgGde* 4–7
version identifier, *ProgGde* 4–19
versioning, *ProgGde* 4–18
shared library files
dependencies, *AsmLangGde* 9–12
dynamic section, *AsmLangGde* 9–5
loading considerations, *AsmLangGde* 9–3
quickstart, *AsmLangGde* 9–22
shared library versioning
defined, *ProgGde* 4–18
shared memory, *ProgGde* 1–6
realtime, *Realtime* 3–1
testing with shmx, *SystemAdmin* 13–5
shared object, *ProgGde* 4–10
shared object files
dependencies, *AsmLangGde* 9–12
initialization and termination functions, *AsmLangGde* 9–22
shared object list section
See .liblist section
sharing usr file system, *SharingSW* 8–1
shell, *CDEAdvUsr* 26, 28, 167, 170
shell scripts, *SystemAdmin* 9–9
compatibility with System V, *CommandShellGde* 9–4
internationalizing, *International* 3–26
shells, *CommandShellGde* 8–1
shmx exerciser, *SystemAdmin* 13–5
shutdown command, *SystemAdmin* 2–17
using halt flag, *SystemAdmin* 2–19
signal, *STREAMSProg* 13–5 to 13–6
signaling module interface, *ATMProgGde* 5–1
simple lock
asserting read and write access with simple_lock, *DevDriverAdvTopics* 3–4

simple lock (cont.)
declaring simple lock structure with decl_simple_lock_data, *DevDriverAdvTopics* 3–1
defined, *DevDriverAdvTopics* 2–6
description of, *DevDriverAdvTopics* 2–6
initializing a simple lock structure with simple_lock_init, *DevDriverAdvTopics* 3–3
releasing previously asserted simple lock with simple_unlock, *DevDriverAdvTopics* 3–6
terminating a simple lock with simple_lock_terminate, *DevDriverAdvTopics* 3–12
trying to obtain simple lock with simple_lock_try, *DevDriverAdvTopics* 3–9
using spl interfaces, *DevDriverAdvTopics* 3–16
Simple Mail Transfer Protocol
See SMTP
Simple Network Management Protocol
See SNMP
simple_lock kernel interface
asserting read and write access, *DevDriverAdvTopics* 3–4
explanation of code fragment, *DevDriverAdvTopics* 3–5
simple_lock_init kernel interface
explanation of code fragment, *DevDriverAdvTopics* 3–3
initializing a simple lock structure, *DevDriverAdvTopics* 3–3
simple_lock_terminate kernel interface
explanation of code fragment, *DevDriverAdvTopics* 3–14

simple_lock_terminate kernel interface (cont.)
terminating a simple lock, *DevDriverAdvTopics* 3–12
simple_lock_try kernel interface
explanation of code fragment, *DevDriverAdvTopics* 3–11
trying to obtain simple lock, *DevDriverAdvTopics* 3–9
simple_unlock kernel interface
explanation of code fragment, *DevDriverAdvTopics* 3–8
releasing previously asserted simple lock, *DevDriverAdvTopics* 3–6
sin routine, *DPML* DPML–57, B–16
sincos routine, *DPML* DPML–58, B–16
sincosd routine, *DPML* DPML–58, B–17
sind routine, *DPML* DPML–57, B–17
single binary module
defined, *DevDriverTutorial* 1–3
single-board computers (SBC), *InstallGuide* 4–30
single-user boot, *SystemAdmin* 2–2
singleops, *displayPS* 5–2 to 5–4
SingleServer, *InstallGuide* D–10
sinh routine, *DPML* DPML–59, B–17
sinhcosh routine, *DPML* DPML–60, B–17
size command, *ProgGde* 2–25
slattach option, *TechOver* 3–14
sleep kernel interface
explanation of code fragment, *DevDriverTutorial* 18–18
SLIP, *NetAdmin* 4–3 to 4–10, *TechOver* 3–15
slock structure
associated include file, *DevDriverReference* 3–64
Small Computer Systems Interface
See SCSI bus
SMP, *SystemAdmin* 2–13, 2–9, *TechOver* 2–1
asserting complex lock, *DevDriverAdvTopics* 4–5
asserting read and write access with simple_lock, *DevDriverAdvTopics* 3–4
configuration file options, *SystemAdmin* 4–45
declaring complex lock structure with lock_data_t, *DevDriverAdvTopics* 4–1
declaring simple lock structure with decl_simple_lock_data, *DevDriverAdvTopics* 3–1
defined, *DevDriverAdvTopics* 2–1
description of complex locks, *DevDriverAdvTopics* 2–8
description of funnels, *DevDriverAdvTopics* 2–10
description of locking methods, *DevDriverAdvTopics* 2–6
description of simple locks, *DevDriverAdvTopics* 2–6
ensuring no locks are required with funnels, *DevDriverAdvTopics* 5–1
example analysis of device driver to determine resources to lock, *DevDriverAdvTopics* 2–19
initializing a complex lock with lock_init, *DevDriverAdvTopics* 4–2
initializing a simple lock structure with simple_lock_init, *DevDriverAdvTopics* 3–3
locking methods, *DevDriverAdvTopics* 2–11
need for locking device drivers, *DevDriverAdvTopics* 2–4
rebooting failed processor, *SystemAdmin* 2–14

SMP (cont.)
releasing previously asserted complex lock with lock_done, *DevDriverAdvTopics* 4–10
releasing previously asserted simple lock with simple_unlock, *DevDriverAdvTopics* 3–6
resources to lock in a device driver, *DevDriverAdvTopics* 2–15
summary of locking methods, *DevDriverAdvTopics* 2–13
terminating a complex lock with lock_terminate, *DevDriverAdvTopics* 4–21
terminating a simple lock with simple_lock_terminate, *DevDriverAdvTopics* 3–12
trying to assert complex lock with lock_try_read, *DevDriverAdvTopics* 4–13
trying to assert complex lock with lock_try_write, *DevDriverAdvTopics* 4–13
trying to obtain simple lock with simple_lock_try, *DevDriverAdvTopics* 3–9
unattended reboots, *SystemAdmin* 2–14
using spl interfaces, *DevDriverAdvTopics* 3–16

SMP system
debugging on, *KernelDebug* 2–9

SMTP, *TechOver* 3–10

SNMP, *NetAdmin* 12–1, *TechOver* 3–11

SNMP application programming interface
Digital UNIX support for, *NetProgGde* 6–1

sockaddr_dl structure, *NetProgGde* E–6
and the 802.2 substructure, *NetProgGde* E–11
and the ethernet substructure, *NetProgGde* E–8

socket interface
Digital UNIX support for, *NetProgGde* 4–1

sockets
advanced topics, *NetProgGde* 4–41 to 4–55
binding names to, *NetProgGde* 4–22
BSD, *NetProgGde* 4–37
4.4BSD features
receipt of protocol data, *NetProgGde* 4–38
variable-length network addresses, *NetProgGde* 4–38
coexistence with STREAMS, *NetProgGde* 7–1 to 7–12
common errors, *NetProgGde* 4–40
communication properties, *NetProgGde* 4–3
connection-oriented programs, *NetProgGde* B–2 to B–17
connectionless programs, *NetProgGde* B–17 to B–30
creating, *NetProgGde* 4–19
data structures, *NetProgGde* 4–16 to 4–18
establishing connections
clients, *NetProgGde* 4–23
servers, *NetProgGde* 4–25
getting socket options, *NetProgGde* 4–27
header files, *NetProgGde* 4–15 to 4–16
library calls, *NetProgGde* 4–10 to 4–15
modes of communication
connection-oriented, *NetProgGde* 4–7, 4–7
rewriting applications for XTI, *NetProgGde* 3–45

sockets (cont.)
setting socket options, *NetProgGde* 4–27
states
comparison between sockets and XTI, *NetProgGde* 3–47
system calls, *NetProgGde* 4–9 to 4–10
TCP specific programming information, *NetProgGde* C–1 to C–3
transferring data, *NetProgGde* 4–29
types, *NetProgGde* 4–5

sockets and STREAMS frameworks
communication between, *NetProgGde* 1–7

sockets framework, *NetProgGde* 1–3
components, *NetProgGde* 4–2
relationship to XTI, *TechOver* 3–18f

software architecture
EISA/ISA bus, *DevDriverEisa* 2–1

software sharing via servers, *SharingSW* 1–2

software subset sizes, *InstallGuide* F–1

software subsets
associated products, *InstallGuide* D–1
descriptions of, *InstallGuide* C–1
installing and removing after an installation, *InstallGuide* 8–1
mandatory, *InstallGuide* C–1
selecting, *SharingSW* 10–12
sizes, *InstallGuide* F–1
worldwide software descriptions, *InstallGuide* E–1

software synchronization
defined, *DevDriverAdvTopics* 2–2

sorting file contents, *CommandShellGde* 3–26

source code
checking with lint, *ProgGde* 6–1

source code compatibility
in System V habitat, *ProgGde* B–1 to B–3

Source Code Control System, *ProgSupTools* 6–1 to 6–34

source files
controlling access to, *ProgGde* 1–5

sparse space
See sparse space address

sparse space address
defined, *DevDriverTutorial* 2–15, 2–16
description of address type that device drivers use, *DevDriverTutorial* 2–15

special characters
in CDE help, *CDEHelp* 80, 181
in dtpad, *CDE_User* 199
in dtterm, *CDE_User* 267

spelling
correcting in dtpad, *CDE_User* 206

spin boxes
CDE conventions and standards for, *CDEStyle* 250

spin lock
See also simple lock
using spl interfaces, *DevDriverAdvTopics* 3–16

splbio kernel interface
explanation of code fragment, *DevDriverTutorial* 18–16

spltty kernel interface
explanation of code fragment, *DevDriverAdvTopics* 3–17

splx kernel interface
explanation of code fragment, *DevDriverAdvTopics* 3–17, *DevDriverTutorial* 18–16

sqrt routine, *DPML* DPML–61, B–18

stack frame, *AsmLangGde* 6–7

stack limit, *CallStandard* 6–1
stack trace, *Ladebug* 7-17, 8-4, 10-8
standard error, *CommandShellGde* 6–1
standard input, *CommandShellGde* 6–1
standard library, *DECC* 9-1
standard output, *CommandShellGde* 6–1
standards
programming considerations, *ProgGde* 1–2
start section, *DevDriverTURBOchannel* 6–62 to 6–65
starting a fixed priority kernel thread
by calling kernel_thread_w_arg kernel interface, *DevDriverAdvTopics* 7–5
starting kernel thread
by calling kernel_thread_w_arg kernel interface, *DevDriverAdvTopics* 7–2
startup files
creating, *SystemAdmin* 9–9
state transitions
allowed for data transfer
connectionless transport services, *NetProgGde* 3–17t
allowed for initialization phase, *NetProgGde* 3–17
stateful and stateless encodings
conversion of, *CDE_I18N_Prog* 59
states
comparison of XTI and sockets, *NetProgGde* 3–47
managing in XTI, *NetProgGde* 3–23
status information (UUCP), *CommandShellGde* 14–27
status subcommand (ftp), *CommandShellGde* 12–8
status subcommand (telnet), *CommandShellGde* 13–4
stopping commands, *CommandShellGde* 1–6
storage class, *DECC* 2-13
strategy section, *DevDriverTURBOchannel* 6–50 to 6–61
description of code example for /dev/dmaex driver, *DevDriverVMEbus* 6–25 to 6–31
strcmp kernel interface
explanation of code fragment, *DevDriverTutorial* 18–2
strcpy kernel interface
explanation of code fragment, *DevDriverTutorial* 18–5
stream editor
See sed editor
STREAMS
coexistence with sockets, *NetProgGde* 7–1 to 7–12
components, *NetProgGde* 5–2
configuring drivers, *NetProgGde* 5–25
configuring modules, *NetProgGde* 5–25
device special files, *NetProgGde* 5–29
error logging, *NetProgGde* 5–31
event logging, *NetProgGde* 5–31
framework, *NetProgGde* 5–1 to 5–32
functions, *NetProgGde* 5–6 to 5–17
header files, *NetProgGde* 5–5
kernel-level functions, *NetProgGde* 5–17 to 5–25
processing routines
for drivers and modules, *NetProgGde* 5–20
STREAMS concepts, *NetProgGde* 5–23
STREAMS drivers
bridging STREAMS drivers to sockets
protocol stacks, *NetProgGde* 7–2

STREAMS drivers (cont.)
bridging to sockets protocol stacks, *NetProgGde* 7–2
STREAMS framework, *NetProgGde* 1–3, 5–1 to 5–32
relationship to XTI, *TechOver* 3–18f
streams mechanism
overview, *STREAMSProg* 5–1, 5–2
STREAMS protocol stacks
bridging to BSD drivers, *NetProgGde* 7–11
streamtab, *STREAMSProg* 5–5 to 5–6, 9–1
string constants, *AsmLangGde* 2–3
string interface
category of kernel interface, *DevDriverTutorial* 18–1
string literal, *DECC* 1-12
string literals in CDE internationalization, *CDE_I18N_Prog* 92, 96, 99
string operation
comparing null-terminated character string using strcmp kernel interface, *DevDriverTutorial* 18–1
comparing two strings using strncmp kernel interface, *DevDriverTutorial* 18–3
copying null-terminated character string using strcpy kernel interface, *DevDriverTutorial* 18–4
copying null-terminated character string using strncpy kernel interface, *DevDriverTutorial* 18–5
returning number of characters using strlen kernel interface, *DevDriverTutorial* 18–7
striped disks
LSM, *StorageManager* 2–4
strlen kernel interface
explanation of code fragment, *DevDriverTutorial* 18–7
strncmp kernel interface
explanation of code fragment, *DevDriverTutorial* 18–3
strncpy kernel interface
explanation of code fragment, *DevDriverTutorial* 18–6
structure
used by autoconfiguration software, *DevDriverTutorial* 17–1
structured exception handling, *ProgGde* 11–5
structures
See also unions
XDR, *OncRpcProgram* 2–25
Style Manager, *CDE_User* 3, 6, 133, *CDEAdvUsr* 59, 267, *CDECompanion* 3–2
subcommands (ftp), *CommandShellGde* 12–5, 12–7, 12–8
subdirectories, *CommandShellGde* 2–7
subnet, *NetAdmin* 3–14
subpanel, *CDE_User* 62, 63, 67, 75, *CDEAdvUsr* 215, 218, 223 to 225, 227
subsystem
configuring, *SystemAdmin* 4–1
suffixes, file name
for programming language files, *ProgGde* 2–4
sunique subcommand (ftp), *CommandShellGde* 12–5
superuser, *CommandShellGde* 5–17
SVC
configuring, *ATMProgGde* 2–1
PPAs, *ATMProgGde* 3–7

svc.conf file
defined, *TechOver* 3–27
swap space
displaying with kdbx, *KernelDebug* 2–32
estimating the size of, *SystemAdmin* 6–5
planning, *InstallGuide* 3–9
switched virtual circuit
See SVC
symbol table, *AsmLangGde* 8–1
format of entries, *AsmLangGde* 8–8
listing, *ProgGde* 2–24
symbolic information
debugging programs with limited, *Ladebug* 17-1
symbols
binding, *ProgGde* 4–31
name resolution semantics, *ProgGde* 4–5
options for handling unresolved symbols, *ProgGde* 4–6
resolution, *ProgGde* 4–5
resolving in shared libraries, *ProgGde* 4–3
search path, *ProgGde* 4–4
symmetric multiprocessing
See SMP
symmetrical multiprocessing
See SMP
sync command, *SystemAdmin* 2–17
synchronization
client and context, *displayPS* 4–7
Display PostScript System and X, *displayPS* 4–6
hardware issues related to, *DevDriverAdvTopics* 2–1
synchronous writes
speeding up with Prestoserve, *Prestoserve* 1–1
sysconfig
defined, *DevDriverTutorial* 12–15
sysconfig command, *SystemAdmin* 4–4
sysconfigdb
defined, *DevDriverTutorial* 12–15
sysconfigdb command, *SystemAdmin* 4–11
sysconfigtab file
See /etc/sysconfigtab file
sysconfigtab file fragment
defined, *DevDriverTutorial* 13–8
use by sysconfigdb program, *DevDriverTutorial* 13–9
syslogd daemon, *SystemAdmin* 13–22, 13–31
SysMan applications, *CDECompanion* 5–17 to 5–19, 5–17
SysMan tools, *TechOver* 13–2
system
displaying information about with kdbx, *KernelDebug* 2–31
system accounting
space for, *InstallGuide* 3–13
system address
description of bus physical address, *DevDriverTutorial* 2–17
description of dense space address, *DevDriverTutorial* 2–16
description of I/O address, *DevDriverTutorial* 2–17
description of kernel physical address, *DevDriverTutorial* 2–15
description of kernel-unmapped virtual address, *DevDriverTutorial* 2–14
description of memory address, *DevDriverTutorial* 2–17
description of sparse space address, *DevDriverTutorial* 2–15

system administration, *TechOver* 13–1
graphical-user interface, *TechOver* 13–2
text-based interface, *TechOver* 13–4
system calls
and DLI, *NetProgGde* E–16
differences in System V habitat, *ProgGde* B–1
sockets, *NetProgGde* 4–9 to 4–10
system clock
setting, *SystemAdmin* 2–15
system configuration
dynamic, *SystemAdmin* 4–4
static, *SystemAdmin* 4–15
system configuration file
pseudodevice entry, *SystemAdmin* 5–4
system crash
process of, *KernelDebug* 1–6
system data type
description of io_handle_t, *DevDriverTutorial* 2–14
system environment
customizing, *SystemAdmin* 3–1
system event
analyzing, *DECevent* 1–1
system event reporting, *SystemAdmin* 13–1
system events and errors, *SystemAdmin* 13–1
system exercisers, *SystemAdmin* 13–1 to 13–22, 13–1
system initialization, *SystemAdmin* 2–11
system initialization files, *SystemAdmin* 3–1
system libraries, *ProgGde* 4–1
system parameters
See also configuration attributes
methods of tuning, *SystemTune* 3–5 to 3–7
system performance, *SystemAdmin* 3–22
managing, *PerformManage* 1–1
system reboot
crash dump information created during, *KernelDebug* 4–9
system security, *SystemAdmin* 3–21
system shutdown, *SystemAdmin* 2–1
system startup files, *SystemAdmin* 3–1
system tuning
CPU tuning, *SystemTune* 3–3
I/O tuning, *SystemTune* 3–17 to 3–30
IPC tuning, *SystemTune* 3–16 to 3–17
network subsystem tuning, *SystemTune* 3–30 to 3–33
tuning memory usage, *SystemTune* 3–7 to 3–16
where to start, *SystemTune* 3–1
System V
IPC parameters, *SystemAdmin* 4–37
messages, *SystemAdmin* 4–37
semaphores, *SystemAdmin* 4–37
shared memory, *SystemAdmin* 4–37
System V Compatibility habitat, *TechOver* 9–1
System V Environment, *TechOver* 9–2
SVID compliance, *TechOver* 9–2f
System V functionality, *TechOver* 9–1
System V habitat, *CommandShellGde* 9–1 to 9–5, *ProgGde* B–1
accessing, *CommandShellGde* 9–2
command summary, *CommandShellGde* 9–4 to 9–5
compiling and linking applications in, *ProgGde* B–1
location of, *CommandShellGde* 9–3
modifying .cshrc file for, *CommandShellGde* 9–2
modifying .login file for, *CommandShellGde* 9–2

System V habitat (cont.)
modifying .profile file for, *CommandShellGde* 9–2
setting PATH for, *CommandShellGde* 9–2
summary of system calls, *ProgGde* B–4
using cc command, *ProgGde* B–1
using ld command, *ProgGde* B–1
System V STREAMS
See streams mechanism
system-specific header file, *displayPS* 5–1

T

t_accept function, *NetProgGde* 3–29
contrast to accept socket call, *NetProgGde* 3–46
t_alloc function, *NetProgGde* 3–41
t_bind function, *NetProgGde* 3–26
contrast to bind socket call, *NetProgGde* 3–46
t_close function, *NetProgGde* 3–36
t_connect function, *NetProgGde* 3–28
t_errno variable, *NetProgGde* 3–64
T_ERROR event
support in TLI, *NetProgGde* 3–43
t_free function, *NetProgGde* 3–41
t_listen function, *NetProgGde* 3–28
t_look function, *NetProgGde* 3–41
contrast to select socket call, *NetProgGde* 3–46
T_MORE flag
and protocol independence, *NetProgGde* 3–42
t_open function, *NetProgGde* 3–24
t_optmgmt function, *NetProgGde* 3–63
t_rcv function, *NetProgGde* 3–32
t_rcvdis function, *NetProgGde* 3–34
and protocol independence, *NetProgGde* 3–42
t_rcvrel function, *NetProgGde* 3–35
t_rcvudata function, *NetProgGde* 3–38
t_rcvuderr function, *NetProgGde* 3–39
and protocol independence, *NetProgGde* 3–42
t_snd function, *NetProgGde* 3–31
t_snddis function, *NetProgGde* 3–34
contrast to close socket call, *NetProgGde* 3–47
t_sndrel function, *NetProgGde* 3–35
t_sndudata function, *NetProgGde* 3–37
t_unbind function, *NetProgGde* 3–36
T_UNINIT state
purpose of, *NetProgGde* 3–23
tan routine, *DPML* DPML–62, B–18
tand routine, *DPML* DPML–62, B–18
tanh routine, *DPML* DPML–63, B–18
tape
bootable, *SystemAdmin* 11–3
tape drive
See magnetic tape drive
tapex exerciser, *SystemAdmin* 13–11
task structure, *DevDriverAdvTopics* 7–1
associated include file, *DevDriverReference* 3–65
TCB, *Security* 6–1, 6–2
TCP, *TechOver* 3–11, 3–4
programming information, *NetProgGde* C–1 to C–3
Tearoff button, *AppBuilder* 139
Tearoff menu property, *AppBuilder* 45, 148
telephone number, specifying with cu command (UUCP), *CommandShellGde* 14–5

TELNET, *TechOver* 3–10

telnet command, *CommandShellGde* 13–3

how to use, *CommandShellGde* 13–3

Telnet Protocol

See TELNET

terminal communications system

testing with cmx, *SystemAdmin* 13–21

terminal device

defined, *DevDriverTutorial* 16–6

discussion of, *DevDriverTutorial* 16–6

terminal device driver

defined, *DevDriverTutorial* 1–2

introductory discussion and examples of, *DevDriverTutorial* 1–2

terminal emulator (dtterm), *CDEAdvUsr* 49, 146, 173, 174

using, *CDE_User* 247

terminal emulators

and CDE application development, *CDEProgOverview* 72, 73

terminal interface

enhancements for Asian languages, *International* 6–13

terminating a connection (local communications), *CommandShellGde* 11–22, 11–23

terminating a kernel thread

by calling thread_halt_self kernel interface, *DevDriverAdvTopics* 7–16

by calling thread_terminate kernel interface, *DevDriverAdvTopics* 7–16

terminating a UUCP job with the uustat command, *CommandShellGde* 14–27

terminating remote cu connection (UUCP), *CommandShellGde* 14–8

terminating remote tip connection (UUCP), *CommandShellGde* 14–14

termination handler, *ProgGde* 11–13

TERMINFO environment variable, *SystemAdmin* 3–8

testing help volumes in CDE, *CDEHelp* 99

testing kernels, *KernelDebug* 2–39

testing menus, help, and connections with App Builder, *AppBuilder* 101 to 105

text editor, *CommandShellGde* 2–1

See also dtpad text editor

vi, *CommandShellGde* A–1

text input components

CDE conventions and standards for, *CDEStyle* 252

text input in CDE internationalization, *CDE_I18N_Prog* 34 to 36, 115

text-to-speech conversion, *InstallGuide* D–8

tftp, *SharingSW* 6–1

TFTP, *TechOver* 3–10

tftpd, *SharingSW* 6–1

Third Degree (Atom tool), *ProgGde* 7–1, 9–2

thread

See also kernel thread

defined, *DevDriverAdvTopics* 6–1

Thread Independent Services, *TechOver* 7–12

thread structure, *DevDriverAdvTopics* 7–1

associated include file, *DevDriverReference* 3–66

list of member names and data types, *DevDriverReference* 3–66

thread_block kernel interface

explanation of code fragment, *DevDriverAdvTopics* 7–23, 7–9

thread_set_timeout kernel interface

explanation of code fragment, *DevDriverAdvTopics* 7–23

threads, *DECthreads* 1–1, *Ladebug* 19-1, *TechOver* 7–12
- profiling multithreaded applications, *ProgGde* 8–29

tic command, *SystemAdmin* 3–7

time server, *NetAdmin* 10–5

time services
- NTP, *TechOver* 3–29
- TSP, *TechOver* 3–30

Time Synchronization Protocol
- *See* TSP

time values
- formatting rules in a locale, *International* 7–20

time zone, *SystemAdmin* 3–18
- SVID, *SystemAdmin* 3–18

timeout changes, *OncRpcProgram* 2–19

timeout kernel interface
- explanation of code fragment, *DevDriverTutorial* 18–19

Timeout Section
- description of code example for /dev/dmaex driver, *DevDriverVMEbus* 6–34

timers
- realtime, *Realtime* 6–1

tip command
- options, connecting to a remote computer, *CommandShellGde* 14–8
- options, connecting to a remote host, *CommandShellGde* 14–9
- using local commands, *CommandShellGde* 14–12, 14–9

TLI
- compatibility with XTI, *NetProgGde* 3–43
- contrast with XTI, *NetProgGde* 3–43

token ring, *NetAdmin* 2–6 to 2–24, *TechOver* 3–17
- setting up a client on, *SharingSW* 3–3

toolbars
- CDE conventions and standards for, *CDEStyle* 45, 89, 91, 227

ToolTalk, *CDEAdvUsr* 25, 29, 182
- features, *ToolTalkMsg* 10
- how applications use, *ToolTalkMsg* 9
- introduction to, *ToolTalkMsg* 1

ToolTalk message distribution, *ToolTalkMsg* 11

ToolTalk message handling, *AppBuilder* 87

ToolTalk messaging
- how to use, *ToolTalkMsg* 15 to 27

ToolTalk Messaging Service, *CDEProgOverview* 5, 16, 17, 33, 34, 47, 48, 50

ToolTalk tracing facility (tttrace), *ToolTalkMsg* 33 to 45

topic hierarchy in CDE help, *CDEHelp* 37

topics and subtopics in CDE help, *CDEHelp* 26

tracepoints, *Ladebug* 7-20, 9-13

tracing in ToolTalk (tttrace), *ToolTalkMsg* 33 to 45

transfer models (clipboard, primary, quick, drag)
- CDE conventions and standards for, *CDEStyle* 173 to 178

transfer-status information (UUCP), *CommandShellGde* 14–27

transitions
- between XTI states, *NetProgGde* 3–17

translation
- of message text, *International* 3–18

Transmission Control Protocol
See TCP
transmission control protocol
See TCP
Transmission Control Protocol
TCP/IP, *SystemTune* 1–18
Transport Layer Interface
See TLI
transport provider
and state management, *NetProgGde* 3–23
Transport Service Data Unit
See TSDU
Trash Can, *CDE_User* 98
Trivial File Transfer Protocol
See TFTP
troubleshooting, *InstallGuide* H–1
trunc routine, *DPML* DPML–64, B–19
trusted computing base
See TCB
try body, *ProgGde* 11–13
try...except statement, *ProgGde* 11–5
try...finally statement, *ProgGde* 11–13
TSDU
and protocol independence, *NetProgGde* 3–42
TSP, *TechOver* 3–30
TTSnoop debugger, *ToolTalkMsg* 29 to 32
tunable Parameters, *STREAMSProg* 13–8, E–4
tunefs command, *SystemAdmin* 6–19
tuning, *ATMProgGde* 11–1
See system tuning
TURBOchannel bus, *TechOver* 6–11
architecture, *DevDriverTURBOchannel* 2–1
device driver
concepts, *DevDriverTURBOchannel* 1–1
configuration, *DevDriverTURBOchannel* 7–1
TURBOchannel bus (cont.)
device driver (cont.)
data structures, *DevDriverTURBOchannel* 4–1
example, *DevDriverTURBOchannel* 6–1
kernel interfaces, *DevDriverTURBOchannel* 5–1
reference information, *DevDriverTURBOchannel* A–1
structure, *DevDriverTURBOchannel* 3–1
TURBOchannel test board, *DevDriverTURBOchannel* 6–2
type conversion, *DECC* 6-27
typedef
using with rpcgen, *OncRpcProgram* 2–23
types of service
in DLPI, *NetProgGde* 2–4

U

uagt_close (CAM), *SCSIDevDrivers* 2–5, B–121
UBC, *TechOver* 5–2
description, *SystemTune* 1–12
UDP, *TechOver* 3–11, 3–4
uerf command, *NetAdmin* 14–9, *SystemAdmin* D–1
using with system exercisers, *SystemAdmin* 13–3
UFS, *SystemAdmin* 6–1, 6–2, *TechOver* 4–3
restoring file systems, *InstallGuide* 7–6
setting file system quotas, *SystemAdmin* 6–23
tuning, *SystemTune* 3–24 to 3–28
UID file search, *CDE_I18N_Prog* 95
UIL modules
in App Builder, *AppBuilder* 14, 17

UIL (User Interface Language)
internationalized, *CDE_I18N_Prog* 92
sample Japanese and English program, *CDE_I18N_Prog* 96
uio structure
associated include file, *DevDriverReference* 3–68
formal description, *DevDriverTutorial* 8–3
list of member names and data types, *DevDriverReference* 3–68
uiomove kernel interface
explanation of code fragment, *DevDriverTutorial* 18–13
umount command, *SystemAdmin* 6–16
unblocking kernel thread
by calling thread_wakeup kernel interface, *DevDriverAdvTopics* 7–16
by calling thread_wakeup_one kernel interface, *DevDriverAdvTopics* 7–16
understanding system data types and addresses
description of bus physical address, *DevDriverTutorial* 2–17
description of dense space address, *DevDriverTutorial* 2–16
description of I/O address, *DevDriverTutorial* 2–17
description of io_handle_t data type, *DevDriverTutorial* 2–14
description of kernel physical address, *DevDriverTutorial* 2–15
description of kernel-unmapped virtual address, *DevDriverTutorial* 2–14
description of memory address, *DevDriverTutorial* 2–17
description of sparse space address, *DevDriverTutorial* 2–15
unified buffer cache
See UBC
unions
XDR, *OncRpcProgram* 2–26
UNIX File System
See UFS
UNIX shell
restoring AdvFS file system, *InstallGuide* 7–8
UNIX Shell option, *InstallGuide* 1–17
UNIX-to-UNIX Copy Program
See UUCP utility
unlimited license
registering after installation or upgrade, *LicenseManage* 2–2
unloading DMA resources
with dma_map_unload, *DevDriverVMEbus* 5–6
unloading system resources for DMA
by calling dma_map_unload kernel interface, *DevDriverTutorial* 18–33
unordered routine, *DPML* DPML–65, B–19
unresolved symbols
options to ld command, *ProgGde* 4–6
shared libraries, *ProgGde* 4–3
untimeout kernel interface
explanation of code fragment, *DevDriverTutorial* 18–20
update administration utility, *InstallGuide* 2–31
update installation, *InstallGuide* 2–1, 2–3
worldwide, *InstallGuide* 10–1
user accounts
adding automatically, *SystemAdmin* 9–5
adding manually, *SystemAdmin* 9–18
deleting, *SystemAdmin* 9–14

user address space
copying from, with copyin kernel interface, *DevDriverTutorial* 18–10
User Agent driver (CAM)
error handling, *SCSIDevDrivers* 2–2
introduction, *SCSIDevDrivers* 2–1
user customization issues
and CDE application development, *CDEProgOverview* 59 to 61
User Datagram Protocol
See UDP
user interface
laying out with App Builder, *AppBuilder* 21 to 29
user mask, *CommandShellGde* 5–13
user object index, new
See DPSNewUserObjectIndex routine
user program
relationship to device driver, *DevDriverTutorial* 1–7
user space coordinate system
See coordinate systems
user space origin
defined, *displayPS* 2–5
obtaining, *displayPS* 6–3, 6–4
setting, *displayPS* 6–4, 6–5
user-defined character UDC databases, *International* 6–23
using commands, *CommandShellGde* 1–4
using the symmetric multiprocessor sleep interface
by calling mpsleep kernel interface, *DevDriverAdvTopics* 7–11
/usr file system, *InstallGuide* 3–17
planning, *InstallGuide* 3–15
/usr/shlib directory
shared libraries, *ProgGde* 4–2
utilities
definition of, *STREAMSProg* C–1, C–17
UUCP utility, *CommandShellGde* 14–1, *NetAdmin* 9–3 to 9–17, *Security* 3–5
local commands, *CommandShellGde* 14–6
local host control of file access, *CommandShellGde* 14–24
system, *TechOver* 3–26
uulog command, *CommandShellGde* 14–31
uumonitor command, *CommandShellGde* 14–32
uustat command, *CommandShellGde* 14–27
uuto command
copying files, local host control, options, *CommandShellGde* 14–24
uux command, *CommandShellGde* 14–18
options, used to run remote commands, *CommandShellGde* 14–14

V

v1_bustype_info_t structure
associated include file, *DevDriverReference* 3–70
definition of members, *DevDriverReference* 3–70
v1_device_info_t structure
associated include file, *DevDriverReference* 3–72
definition of members, *DevDriverReference* 3–72
v1_devtype_info_t structure
associated include file, *DevDriverReference* 3–74
definition of members, *DevDriverReference* 3–74

v1_disk_dev_info_t structure
associated include file, *DevDriverReference* 3–75
definition of members, *DevDriverReference* 3–75
v1_tape_dev_info_t structure
associated include file, *DevDriverReference* 3–77
definition of members, *DevDriverReference* 3–77
var area, *InstallGuide* 3–17
planning space for, *InstallGuide* 3–11
variables
See also environment variables
mail, *CommandShellGde* 11–16 to 11–17
variables in dtksh app-defaults file, *Desktopksh* 15
VAX C extensions, *DECC* E-1
vballoc
See dma_map_alloc kernel interface
vbasetup
See dma_map_alloc kernel interface
VendorShell widget, *CDE_I18N_Prog* 15, 18, 19, 87, 88, 90
verbose mode debugging, *Ladebug* 10-20
verbose subcommand (ftp), *CommandShellGde* 12–8
versioning
See shared libraries
shared libraries, *ProgGde* 4–18
VFS, *SystemAdmin* 6–1, 6–2, *TechOver* 4–1
tuning information, *SystemTune* 3–21 to 3–24
vi text editor, *CommandShellGde* 2–2, A–1
virtual circuit
See also PVC
virtual circuit (cont.)
See also SVC
data structure, *ATMProgGde* 3–14
Virtual File System
See VFS
virtual keys
CDE conventions and standards for, *CDEStyle* 3
virtual memory
See VM
virtual memory subsystem
description, *SystemTune* 1–3
virtual space
moving data between user and system with uiomove kernel interface, *DevDriverTutorial* 18–13
virus, *CommandShellGde* 5–21
visibility, *DECC* 2-6
visual design of applications
CDE conventions and standards for, *CDEStyle* 39 to 63
VM, *displayPS* 2–4
eager reservation policy, *TechOver* 5–2
external pager, *TechOver* 5–4
kernel memory allocator, *TechOver* 5–4
lazy allocation policy, *TechOver* 5–1
Mach mmap, *TechOver* 5–3
memory reclamation policy, *TechOver* 5–5
memory-mapped device interface, *TechOver* 5–3
page coloring, *TechOver* 5–4
page in and page out clustering, *TechOver* 5–3
round-robin swapping, *TechOver* 5–3
shared memory segments, secure, *TechOver* 5–3

VM (cont.)
shared text segments, *TechOver* 5–4
unified buffer cache, *TechOver* 5–2
vmcore.n file, *KernelDebug* 4–10
VME 2100 Systems, *DevDriverVMEbus* D–1
vme_handler_info structure
associated include file, *DevDriverVMEbus* A–40
list of member names and data types, *DevDriverVMEbus* A–40
VMEbus
byte ordering, *DevDriverVMEbus* 2–2
data size support, *DevDriverVMEbus* 2–2
porting information, *DevDriverVMEbus* 1–5
support for DMA, *DevDriverVMEbus* 2–5
VMEbus address
obtaining with vba_get_vmeaddr, *DevDriverVMEbus* 5–7
VMEbus address space
See DMA data transfer resources
unloading DMA resources, *DevDriverVMEbus* 5–6
VMEbus architectures
address spaces, *DevDriverVMEbus* 2–1, 2–3
vmunix.n file, *KernelDebug* 4–10
vnode tuning, *SystemTune* 3–21 to 3–24
vold daemon (LSM), *StorageManager* 1–8
voldiskadm interface (LSM), *StorageManager* 9–1
voliod daemon (LSM), *StorageManager* 1–8

W

wakeup kernel interface
explanation of code fragment, *DevDriverTutorial* 18–19
waking up kernel thread
See unblocking kernel thread
wall command, *SystemAdmin* 2–11, 2–17
welcome message, *CDEAdvUsr* 13, 25, 27
what command, *ProgSupTools* 6–22
whatis database
updating, *InstallGuide* 2–38, D–3
wide characters and wide-character strings, *International* 1–5
widget translations
setting with dtskh, *Desktopksh* 37
widgets
and CDE application development, *CDEProgOverview* 40, 47, 64 to 67, 72
widgets, Motif
cutomized for CDE applications development, *CDEProgGuide* 69 to 94
window customization, *Ladebug* 2-5
window management
CDE conventions and standards for, *CDEStyle* 66, 79, 187 to 189
window manager
customizing in DECwindows Motif, *DWUser* 6–2 to 6–7
in CDE internationalization, *CDE_I18N_Prog* 20, 125
MWM, *XWinSysEnv* 1–1
window manager close notice
responding to with dtksh, *Desktopksh* 31
Window menu, *CDE_User* 26, 290, *CDEAdvUsr* 247, 249
windowing environment, *Security* 4–1, *TechOver* 8–1
windows
CDE conventions and standards for, *CDEStyle* 65 to 70, 92

windows (cont.)
implementing with App Builder, *AppBuilder* 6, 22, 35, 118, 138, 139, 143, 147
word operations
See performing write word operations
workspace backdrop, *CDE_User* 103, 139
Workspace Manager
customizing the, *CDEAdvUsr* 241
using to manage application windows, *CDEProgGuide* 63 to 67
using with dtksh, *Desktopksh* 35
Workspace menu, *CDE_User* 28, *CDEAdvUsr* 246 to 248
workspace switch, *CDEAdvUsr* 216, 218, 236
workspaces, *CDE_User* 15 to 18, 66, 77, 104, 136, 290, 291, *CDEAdvUsr* 236, 244, 245
workstation
configuration file definitions, *SystemAdmin* 4–51
worldwide installation, *InstallGuide* 9–1
Worldwide Portability Interfaces WPI, *International* A–1
worldwide software subset descriptions, *InstallGuide* E–1
worldwide support, *SystemAdmin* 3–18
worldwide update installation, *InstallGuide* 10–1
write command (local communications), *CommandShellGde* 11–21
write-only access
support in TLI, *NetProgGde* 3–43
writing CDE help
tasks in, *CDEHelp* 14 to 19, 32 to 48, 51 to 85
writing data from a device register
with write_io_port, *DevDriverVMEbus* 5–8

X

X client libraries, *TechOver* 8–4
X clients, *TechOver* 8–9
introduction, *DWUser* 1–1
X coordinate system
See coordinate systems
X drawing functions
accessing with dtksh, *Desktopksh* 36
X Graphic Context
defined, *displayPS* 2–2
setting, *displayPS* 6–4, 6–5
X libraries
using internationalization features in, *International* 5–1
X Logical Font Description (XLFD), *CDE_I18N_Prog* 11, 23
X server, *CDEAdvUsr* 15, 16, *TechOver* 8–4
defined, *DWUser* 1–1
font-cache tuning, *International* 6–86
X server extensions, *TechOver* 8–5, *XWinSysEnv* 4–1
X terminal, *CDEAdvUsr* 3, 7 to 9, 88, 94
X window
configuration, *XWinSysEnv* 2–1
managing the environment, *XWinSysEnv* 2–1
X Window System, *TechOver* 8–4, *XWinSysEnv* 1–1
documentation about, *XWinSysEnv* 1–2
X11R6
internationalization features in, *International* 5–1
X/Open
standards and application development considerations, *ProgGde* 1–2
standards and internationalization, *International* 1–1

X/Open Transport Interface
See XTI
X-specific operators
See operators
Xconfig file, *CDEAdvUsr* 3, 14, 276
.Xdefaults file, *CDEAdvUsr* 30, 256, *DWUser* 9–1 to 9–10
xdm utility, *XWinSysEnv* 2–8
XDM-AUTHORIZATION-1
Display Manager, *TechOver* 8–8
XDMCP, *CDEAdvUsr* 2, 6 to 9, 21
XDPSContextFromSharedID routine, *displayPS* 5–7
XDPSContextFromXID routine, *displayPS* 5–7
XDPSCreateContext routine, *displayPS* 5–7
XDPSCreateSimpleContext routine, *displayPS* 5–9
XDPSFindContext routine, *displayPS* 5–11
XDPSRegisterStatusProc routine, *displayPS* 5–11
XDPSSetStatusMask routine, *displayPS* 5–11
XDPSSpaceFromSharedID routine, *displayPS* 5–12
XDPSSpaceFromXID routine, *displayPS* 5–13
XDPSUnfreezeContext routine, *displayPS* 5–13
XDPSXIDFromContext routine, *displayPS* 5–13
XDPSXIDFromSpace routine, *displayPS* 5–13
XDR, *OncRpcProgram* 2–1, A–1
enumerations, *OncRpcProgram* 2–22
system routines, *OncRpcProgram* A–1
XDR library, *OncRpcProgram* A–8
XEvent structure
accessing with dtksh, *Desktopksh* 30
XIM, *CDE_I18N_Prog* 34, 35, 36, 38 to 40
xjoin comand, *NetAdmin* 3–12
xlsfonts command, *CDEAdvUsr* 263, 279
XMI bus, *TechOver* 6–13
XMODIFIERS environment variable, *CDE_User* 300
xmodmap, *CDE_User* 300, *DWUser* 9–13
xmodmap keymap format
Display Manager, *TechOver* 8–7
xntpd daemon, *NetAdmin* 10–9
xntpdc command, *NetAdmin* 10–10
XPG3 compliance
and Digital UNIX’s XTI, *NetProgGde* 3–1
XPT routines
xpt_ccb_alloc, *SCSIDevDrivers* 7–1, B–126
xpt_ccb_alloc, *SCSIDevDrivers* 7–1, B–126
xrefresh, *DWUser* C–2
Xservers file, *CDEAdvUsr* 4, 5, 20
Xsession file, *CDEAdvUsr* 17, 19, 25, 26
Xt and Xlib dependencies in internationalization, *CDE_I18N_Prog* 101 to 125
-xtaso flag (cc), *ProgGde* A–1
-xtaso_short flag (cc), *ProgGde* A–1
XTI
and TLI, *NetProgGde* 3–1
and XPG3 compliance, *NetProgGde* 3–1
connection-oriented programs, *NetProgGde* B–2 to B–17
connectionless programs, *NetProgGde* B–17 to B–30
contrast with TLI, *NetProgGde* 3–43
data flow with a sockets-based transport provider, *TechOver* 3–19
data flow with a STREAMS-based transport provider, *TechOver* 3–19

XTI (cont.)
defined, *NetProgGde* 3–1, *TechOver* 3–18
errors
t_errno variable, *NetProgGde* 3–64
events, *NetProgGde* 3–10
used by connectionless transport services, *NetProgGde* 3–37
handling errors, *NetProgGde* 3–64
map of functions, events, and states, *NetProgGde* 3–17
outgoing events, *NetProgGde* 3–15t
passing connections to other endpoints, *NetProgGde* 3–16
porting applications to, *NetProgGde* 3–41 to 3–47
relationship to STREAMS and sockets frameworks, *TechOver* 3–18f
rewriting socket applications for, *NetProgGde* 3–45
state management by transport providers, *NetProgGde* 3–23
states
comparison between XTI and sockets, *NetProgGde* 3–47, 3–10
synchronization of multiple processes, *NetProgGde* 3–20
writing connection-oriented applications, *NetProgGde* 3–24
xti.h header file
and t_errno variable, *NetProgGde* 3–65

Y

y0 routine, *DPML* DPML–12, B–19
y1 routine, *DPML* DPML–12, B–19
yacc program, *ProgSupTools* 4–15 to 4–38
yn routine, *DPML* DPML–12, B–20
ypcat command, *NetAdmin* 7–31
ypmatch command, *NetAdmin* 7–31
yppasswd command, *NetAdmin* 7–31
ypwhich command, *NetAdmin* 7–31

Z

z subcommand (telnet), *CommandShellGde* 13–4
ZMAGIC files, *AsmLangGde* 7–24
segment access permissions, *AsmLangGde* 9–2

How to Order Additional Documentation

Technical Support

If you need help deciding which documentation best meets your needs, call 800-DIGITAL (800-344-4825) before placing your electronic, telephone, or direct mail order.

Electronic Orders

To place an order at the Electronic Store, dial 800-234-1998 using a 1200- or 2400-bps modem from anywhere in the USA, Canada, or Puerto Rico. If you need assistance using the Electronic Store, call 800-DIGITAL (800-344-4825).

Telephone and Direct Mail Orders

Your Location	**Call**	**Contact**
Continental USA, Alaska, or Hawaii	800-DIGITAL	Digital Equipment Corporation P.O. Box CS2008 Nashua, New Hampshire 03061
Puerto Rico	809-754-7575	Local Digital subsidiary
Canada	800-267-6215	Digital Equipment of Canada Attn: DECdirect Operations KAO2/2 P.O. Box 13000 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6
International	———————	Local Digital subsidiary or approved distributor
Internal[a]	———————	SSB Order Processing – NQO/V19 *or* U. S. Software Supply Business Digital Equipment Corporation 10 Cotton Road Nashua, NH 03063-1260

[a] For internal orders, you must submit an Internal Software Order Form (EN-01740-07).

Reader's Comments

Digital UNIX
Documentation Overview, Glossary,
and Master Index
AA-QTM6A-TE

Digital 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) 881-0120, Attn: UEG 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)				
Completeness (enough information)				
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 Dept.
Company Date
Mailing Address
Email Phone

Do Not Cut or Tear − Fold Here and Tape

digital™

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

DIGITAL EQUIPMENT CORPORATION
UEG PUBLICATIONS MANAGER
ZKO3−3/Y32
110 SPIT BROOK RD
NASHUA NH 03062−9987

Do Not Cut or Tear − Fold Here

Cut on
Dotted
Line