

Addendum to AOS/VS, AOS/VS II, and AOS/RT32 System Call Dictionary, ?R Through ?Z

086-000196-00

This addendum updates your manual 093-000543-02. Please see "Updating Your Manual." If you are running AOS/VS Revision 7.69, do not insert this addendum, which becomes effective with AOS/VS Revision 7.70.

Notice

DATA GENERAL CORPORATION (DGC) HAS PREPARED THIS DOCUMENT FOR USE BY DGC PERSONNEL, LICENSEES, AND CUSTOMERS. THE INFORMATION CONTAINED HEREIN IS THE PROPERTY OF DGC; AND THE CONTENTS OF THIS MANUAL SHALL NOT BE REPRODUCED IN WHOLE OR IN PART NOR USED OTHER THAN AS ALLOWED IN THE DGC LICENSE AGREEMENT.

DGC reserves the right to make changes in specifications and other information contained in this document without prior notice, and the reader should in all cases consult DGC to determine whether any such changes have been made.

THE TERMS AND CONDITIONS GOVERNING THE SALE OF DGC HARDWARE PRODUCTS AND THE LICENSING OF DGC SOFTWARE CONSIST SOLELY OF THOSE SET FORTH IN THE WRITTEN CONTRACTS BETWEEN DGC AND ITS CUSTOMERS. NO REPRESENTATION OR OTHER AFFIRMATION OF FACT CONTAINED IN THIS DOCUMENT INCLUDING BUT NOT LIMITED TO STATEMENTS REGARDING CAPACITY, RESPONSE-TIME PERFORMANCE, SUITABILITY FOR USE OR PERFORMANCE OF PRODUCTS DESCRIBED HEREIN SHALL BE DEEMED TO BE A WARRANTY BY DGC FOR ANY PURPOSE, OR GIVE RISE TO ANY LIABILITY OF DGC WHATSOEVER.

This software is made available solely pursuant to the terms of a DGC license agreement, which governs its use.

Restricted Rights Legend: Use, duplication, or disclosure by the U. S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at [DFARS] 252.227-7013 (October 1988).

Data General Corporation
4400 Computer Drive
Westboro, MA 01580

AViON, CEO, DASHER, DATAPREP, DESKTOP GENERATION, ECLIPSE, ECLIPSE MV/4000, ECLIPSE MV/6000, ECLIPSE MV/8000, GENAP, INFOS, microNOVA, NOVA, PRESENT, PROXI, SWAT, TRENDVIEW, and WALKABOUT are U.S. registered trademarks of Data General Corporation; and AOSMAGIC, AOS/VSMAGIC, AROSE/PC, ArrayPlus, AV Object Office, AV Office, BaseLink, BusiGEN, BusiPEN, BusiTEXT, CEO Connection, CEO Connection/LAN, CEO Drawing Board, CEO DXA, CEO Light, CEO MAILL, CEO Object Office, CEO PXA, CEO Wordview, CEOwrite, COBOL/SMART, COMPUCALC, CSMAGIC, DASHER/One, DASHER/286, DASHER/286-12c, DASHER/286-12j, DASHER/386, DASHER/386-16c, DASHER/386-25, DASHER/386-25k, DASHER/386SX, DASHER/386SX-16, DASHER/386SX-20, DASHER/486-25, DASHER II/486-33TE, DASHER/LN, DATA GENERAL/One, DESKTOP/UX, DG/500, DG/AROSE, DGConnect, DG/DBUS, DG/Fontstyles, DG/GATE, DG/GEO, DG/HEO, DG/L, DG/LIBRARY, DG/UX, DG/XAP, ECLIPSE MV/1000, ECLIPSE MV/1400, ECLIPSE MV/2000, ECLIPSE MV/2500, ECLIPSE MV/3500, ECLIPSE MV/5000, ECLIPSE MV/5500, ECLIPSE MV/5600, ECLIPSE MV/7800, ECLIPSE MV/9300, ECLIPSE MV/9500, ECLIPSE MV/9600, ECLIPSE MV/10000, ECLIPSE MV/15000, ECLIPSE MV/18000, ECLIPSE MV/20000, ECLIPSE MV/30000, ECLIPSE MV/35000, ECLIPSE MV/40000, ECLIPSE MV/60000, FORMA-TEXT, GATEKEEPER, GDC/1000, GDC/2400, Intellibook, microECLIPSE, microMV, MV/UX, OpenMAC, PC Liaison, RASS, REV-UP, SLATE, SPARE MAIL, SUPPORT MANAGER, TEO, TEO/3D, TEO/Electronics, TURBO/4, UNITE, and XODIAC are trademarks of Data General Corporation.

Addendum to
AOS/VS, AOS/VS II, and AOS/RT32 System Call Dictionary, ?R Through ?Z
086-000196-00

In the margins of replacement pages, a vertical bar indicates substantive technical change from 093-000543-02.

The addendum number appears on all pages in this addendum.

Updating Your Manual

This addendum (086-000196-00) to *AOS/VS*, *AOS/VS II*, and *AOS/RT32 System Call Dictionary, ?R Through ?Z* introduces new information effective with AOS/VS II Release 2.20. It also includes minor corrections.

To update your copy of 093-000543-02, please remove manual pages and insert addendum pages as follows:

Remove

Title/Notice

iii/iv

old Table of Contents

2-595 through 2-604

2-625/2-626

2-629/2-630

2-643/2-644

2-649 through 2-654

2-693/2-694

2-703/2-704

2-709 through 2-712

2-721 through 2-724

2-741 through 2-744

2-759 through 2-768

2-783/2-784

2-861/ 2-862

2-877/2-878

2-881/2-882

B-5/B-6

B-13/B14

old Index

old Document Set

Insert

Title/Notice

iii/iv

new Table of Contents

2-595 through 2-604

2-625/2-626

2-629 through 2-632.2

2-643/2-644

2-649 through 2-654

2-693/2-694

2-703/2-704

2-709 through 2-712

2-721 through 2-724

2-741 through 2-744

2-759 through 2-768

2-783/2-784

2-861/ 2-862

2-877/2-878

2-881/2-882

B-5 through B-6.2

B-13 through B-14.2

new Index

new Document Set

Where new material requires additional pages, the pages have a decimal and number suffix; for example 5-21.1, 5-22.2.

Insert this sheet immediately behind the new Title/Notice page.

AOS/VS, AOS/VS II, and AOS/RT32 System Call Dictionary, ?R Through ?Z

093-000543-02

For the latest enhancements, cautions, documentation changes, and other information on this product, please see the Release Notice (085-series) supplied with the software.

Copyright ©Data General Corporation, 1988, 1990, 1991
All Rights Reserved
Unpublished - all rights reserved under the copyright laws of the United States.
Printed in the United States of America
Rev. 02, December, 1991
Licensed Material – Property of Data General Corporation
Ordering No. 093-000543

Notice

DATA GENERAL CORPORATION (DGC) HAS PREPARED THIS DOCUMENT FOR USE BY DGC PERSONNEL, LICENSEES, AND CUSTOMERS. THE INFORMATION CONTAINED HEREIN IS THE PROPERTY OF DGC; AND THE CONTENTS OF THIS MANUAL SHALL NOT BE REPRODUCED IN WHOLE OR IN PART NOR USED OTHER THAN AS ALLOWED IN THE DGC LICENSE AGREEMENT.

DGC reserves the right to make changes in specifications and other information contained in this document without prior notice, and the reader should in all cases consult DGC to determine whether any such changes have been made.

THE TERMS AND CONDITIONS GOVERNING THE SALE OF DGC HARDWARE PRODUCTS AND THE LICENSING OF DGC SOFTWARE CONSIST SOLELY OF THOSE SET FORTH IN THE WRITTEN CONTRACTS BETWEEN DGC AND ITS CUSTOMERS. NO REPRESENTATION OR OTHER AFFIRMATION OF FACT CONTAINED IN THIS DOCUMENT INCLUDING BUT NOT LIMITED TO STATEMENTS REGARDING CAPACITY, RESPONSE-TIME PERFORMANCE, SUITABILITY FOR USE OR PERFORMANCE OF PRODUCTS DESCRIBED HEREIN SHALL BE DEEMED TO BE A WARRANTY BY DGC FOR ANY PURPOSE, OR GIVE RISE TO ANY LIABILITY OF DGC WHATSOEVER.

This software is made available solely pursuant to the terms of a DGC license agreement, which governs its use.

Restricted Rights Legend: Use, duplication, or disclosure by the U. S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at [DFARS] 252.227-7013 (October 1988).

Data General Corporation
4400 Computer Drive
Westboro, MA 01580

AViiON, CEO, DASHER, DATAPREP, DESKTOP GENERATION, ECLIPSE, ECLIPSE MV/4000, ECLIPSE MV/6000, ECLIPSE MV/8000, GENAP, INFOS, microNOVA, NOVA, PRESENT, PROXI, SWAT, TRENDVIEW, and WALKABOUT are U.S. registered trademarks of Data General Corporation; and **AOSMAGIC, AOS/VSMAGIC, AROSE/PC, ArrayPlus, AV Object Office, AV Office, BaseLink, BusiGEN, BusiPEN, BusiTEXT, CEO Connection, CEO Connection/LAN, CEO Drawing Board, CEO DXA, CEO Light, CEO MAILI, CEO Object Office, CEO PXA, CEO Wordview, CEOwrite, COBOL/SMART, COMPUCALC, CSMAGIC, DASHER/One, DASHER/286, DASHER/286-12c, DASHER/286-12j, DASHER/386, DASHER/386-16c, DASHER/386-25, DASHER/386-25k, DASHER/386SX, DASHER/386SX-16, DASHER/386SX-20, DASHER/486-25, DASHER II/486-33TE, DASHER/LN, DATA GENERAL/One, DESKTOP/UX, DG/500, DG/AROSE, DGConnect, DG/DBUS, DG/Fontstyles, DG/GATE, DG/GEO, DG/HEO, DG/L, DG/LIBRARY, DG/UX, DG/XAP, ECLIPSE MV/1000, ECLIPSE MV/1400, ECLIPSE MV/2000, ECLIPSE MV/2500, ECLIPSE MV/3500, ECLIPSE MV/5000, ECLIPSE MV/5500, ECLIPSE MV/5600, ECLIPSE MV/7800, ECLIPSE MV/9300, ECLIPSE MV/9500, ECLIPSE MV/9600, ECLIPSE MV/10000, ECLIPSE MV/15000, ECLIPSE MV/18000, ECLIPSE MV/20000, ECLIPSE MV/30000, ECLIPSE MV/35000, ECLIPSE MV/40000, ECLIPSE MV/60000, FORMA-TEXT, GATEKEEPER, GDC/1000, GDC/2400, Intellibook, microECLIPSE, microMV, MV/UX, OpenMAC, PC Liaison, RASS, REV-UP, SLATE, SPARE MAIL, SUPPORT MANAGER, TEO, TEO/3D, TEO/Electronics, TURBO/4, UNITE, and XODIAC** are trademarks of Data General Corporation.

AOS/VS, AOS/VS II, and AOS/RT32 System Call Dictionary, ?R Through ?Z
093-000543-02

Revision History:
Original Release – October 1988
First Revision – February 1990
Second Revision – December 1991
Addendum 086-000196 – June 1992

Effective with:

AOS/VS, Rel. 7.70
AOS/VS II, Rel. 2.20
AOS/RT32, Rel. 5.01

A vertical bar in the outer margin of a page indicates substantive change from the previous revision of this manual.

Preface

The System Call Dictionary spans two manuals — one for system calls ?A through ?Q, and the other for system calls ?R through ?Z. Much information appears twice in these two manuals for your ease of use. For example, the table of contents and indexes are identical. Chapter 2 in both books has the same title, but their contents differ. Chapter 2 in manual 093–000543 is a continuation of Chapter 2 in manual 093–000542, and is paginated accordingly. Appendixes A and B follow Chapter 2 in the second manual.

This manual is intended for use by experienced assembly language programmers. Experienced high-level language programmers can also use this manual to create programs that make direct calls to the operating systems.

Organization

This manual is organized as follows:

- Chapter 2 begins with a summary table of all AOS/VS and AOS/RT32 system calls, followed by detailed descriptions of all the system calls whose names begin with ?A through ?Q.
- Appendix A contains 12 program sets that illustrate AOS/VS and AOS/RT32 system calls. We have written 11 of the program sets in assembly language and the twelfth in FORTRAN 77. The Appendix is located at the end of the complementary manual *AOS/VS, AOS/VS II, and AOS/RT32 System Call Dictionary, ?R Through ?Z* (093–000543).
- Appendix B describes the format of the *system log (SYSLOG) file*, into which both AOS/VS and AOS/VS II and privileged processes can write records that log the occurrence of certain events. The Appendix is located at the end of the complementary manual *AOS/VS, AOS/VS II, and AOS/RT32 System Call Dictionary, ?R Through ?Z* (093–000543).

Related Documentation

As mentioned earlier, the complement of this manual is *AOS/VS, AOS/VS II, and AOS/RT32 System Call Dictionary, ?R Through ?Z* (093–000543). The following documents are ancillary to both manuals.

- *AOS/VS System Concepts* (093–000335)
- *Introduction to AOS/RT32* (069–400061)
- *AOS/VS and AOS/VS II Error and Status Messages* (093–000540)
- *AOS/VS and AOS/VS II Glossary* (069–000231)

AOS/VS System Concepts and Introduction to AOS/RT32, listed at the beginning of this section, contain a general description of operating system calls and how to use them. This manual and its companion system call dictionary manual contain detailed descriptions of each AOS/VS and AOS/RT32 call. For your convenience, the system call descriptions in the two dictionaries are in alphabetical order.

The Documentation Set, after the index in each manual, contains a complete annotated list of AOS/VS and AOS/VS II manuals.

If you are not experienced with assembly language, we suggest that you read the following manuals before you read this book:

- *Fundamentals of Small Computer Programming* (093–000090), which provides a general introduction to Data General computers.
- *AOS/VS Macroassembler (MASM) Reference Manual* (093–000242), which gives detailed information about the syntax of AOS/VS assembly language and about the Macroassembler utility.
- *ECLIPSE® MV/Family (32-Bit) Systems Principles of Operation* (014–001371), which explains the processor-independent concepts and functions of ECLIPSE® MV/Family systems to assembly language programmers.
- *ECLIPSE® MV/Family (32-Bit) Systems Instruction Dictionary* (014–001372), which explains each instruction in the ECLIPSE MV/Family instruction set to assembly language programmers. Processor-dependent information, available in machine-specific supplements, complements this and the previous manual. An example of such information is found in the manual *ECLIPSE® MV/20000™ Series Systems Principles of Operation Supplement* (014–001169).
- *ECLIPSE® MV/Family Instruction Reference Booklet* (014–000702), which provides a brief summary of the instruction set and register information. The reference booklet lists each instruction by assembler-recognizable mnemonic with a shorthand description of its function.
- *FORTRAN 77 Environment Manual (AOS/VS)* (093–000288).

Update and Release Notices

Certain features of the operating systems may change from revision to revision. Therefore, please refer to the current Release Notice for the most up-to-date information about functional changes and enhancements. The Release Notice is usually in the utilities directory (:UTIL) on your system. The filename of the AOS/VS Model 3900 Update Notice is 078_000105_**; that of the AOS/VS II Release Notice is 085_000930_**. Suffixes (**) change with each revision. Your system manager should be able to tell you the exact pathname of the Release Notice.

The AOS/VS and AOS/VS II Release and Update Notices contain the latest details about all the system software, including enhancements and changes, notes and warnings. Notices are supplied both as printed listings and as disk files that you can print. The manuals and the Notices comprise the documentation for the system calls for AOS/VS Revision 7.69, and for AOS/VS II Revision 2.10. There are no documentation-changes files for this manual.

You should read the Update and Release Notices. If you want to know the features of a release, or have problems with a release, read the notice for solutions. The notices assume that you know the operating system well — so parts of the notices may be difficult to understand until you *do* know the system.

The Newsletter

Finally, you will find the *AOS/VS Monthly Newsletter* a useful source of information on the latest enhancements to both AOS/VS and AOS/VS II.

Reader Please Note

Throughout this manual we use the following format conventions:

COMMAND required [optional] ...

Where	Means
COMMAND	You must enter the command (or its accepted abbreviation) as shown.
required	You must enter some argument (such as a filename). Sometimes, we use $\left\{ \begin{array}{l} \text{required}_1 \\ \text{required}_2 \end{array} \right\}$ which means you must enter one of the arguments. Do not type the braces; they only set off the choices.
[optional]	You have the option of entering this argument. Do not type the brackets; they only identify the argument as an option.
...	You may repeat the preceding entry.

Standard Symbols

Additionally, we use the following symbols:

Symbol	Means
↵	Press the New Line, Carriage Return (CR), or Enter key on your terminal keyboard.
)	The CLI prompt.
< >	Angle brackets indicate the paraphrase of an argument or statement. (You supply the actual argument or statement.)
*	One asterisk indicates multiplication. For example, 2*3 means 2 multiplied by 3.
**	Two asterisks indicate exponentiation. For example, 2**3 means 2 raised to the third power.
OS	The operating system in the accumulator I/O, figure, and table categories.

Unless the text supplies a specific radix (as it often does), all memory addresses are octal values and all other numbers are decimal values. To explicitly specify a decimal number, we sometimes use a period after the last digit. To explicitly specify an octal number, we sometimes use the phrases *octal value* or *base eight*. For example, the phrase "a baker's dozen cookies" has traditionally meant 13. = 15 base eight cookies.

In this manual, AOS/VS means AOS/VS, AOS/VS II, or both, unless otherwise noted.

Finally, in examples we use

This typeface to show your entry.

This typeface to show system queries and responses.

This typeface to show listings and status displays.

Contacting Data General

Data General wants to assist you in any way it can to help you use its products. Please feel free to contact the company as outlined below.

Manuals

If you require additional manuals, please use the enclosed TIPS order form (United States only) or contact your local Data General sales representative.

Telephone Assistance

If you are unable to solve a problem using any manual you received with your system, free telephone assistance is available with your hardware warranty and with most Data General software service options. If you are within the United States or Canada, contact the Data General Customer Support Center (CSC) by calling 1-800-DG-HELPS. Lines are open from 8:00 a.m. to 5:00 p.m., your time, Monday through Friday. The center will put you in touch with a member of Data General's telephone assistance staff who can answer your questions.

Joining Our Users Group

Please consider joining the largest independent organization of Data General users, the North American Data General Users Group (NADGUG). In addition to making valuable contacts, members receive *FOCUS* monthly magazine, a conference discount, access to the Software Library and Electronic Bulletin Board, an annual Member Directory, Regional and Special Interest Groups, and much more. For more information about membership in the North American Data General Users Group, call 1-800-253-3902 or 1-508-443-3330.

End of Preface

Contents

Chapter 1 — Introducing the System Calls

Standard Format for System Calls	1-2
Parameter Packets	1-3
Parametric Coding	1-3
Other System Call Input	1-4
Reserved Symbols	1-5
A Complete Example — Assembly Language	1-5
General Step	1-5
Program Listing	1-6
Program Construction	1-8
Program Execution	1-8
Error Codes	1-9
High-Level Language Interface	1-10

Chapter 2 — AOS/VS, AOS/VS II, and AOS/RT32 System Calls

Summary Table	2-2
---------------------	-----

?A Through ?Q

?ALLOCATE	Allocates disk blocks.	2-15
?ASSIGN	Assigns a character device to a process.	2-17
?AWIRE	Changes the wiring characteristics of the Agent.	2-18
?BLKIO	Performs (reads/writes) block I/O.	2-19
?BLKPR	Blocks a process.	2-26
?BNAME	Determines whether process name/queue name is on local or remote host.	2-28
?BRKFL	Terminates a process and creates a break file	2-29
?CDAY	Converts a scalar date value.	2-31
?CGNAM	Gets a complete pathname from a channel number.	2-32
?CHAIN	Passes control from a Ring 7 caller to a new program.	2-33
?CKVOL	Checks volume identifier of a labeled magnetic tape.	2-35
?CLASS	Gets or sets class IDs.	2-36
?CLOSE	Closes an open channel.	2-38
?CLR DV	Clears a device.	2-40
?CLSCHED	Enables, disables, or examines class scheduling.	2-42
?CLSTAT	Returns class scheduling statistics.	2-45
?CMATRIX	Gets or sets the class matrix.	2-51
?CON	Becomes a customer of a specified server.	2-56
?CONFIG	Display or reset current MRC routes	2-58
?CONINFO	Request for addressing information on a terminal or console	2-58.6
?CPMAX	Sets maximum size for a control point directory (CPD).	2-58.26
?CREATE	Creates a file or directory.	2-60
?CRUDA	Creates a user data area (UDA).	2-70

?A Through ?Q

?CTERM	Terminates a customer process.	2-72
?CTOD	Converts a scalar time value.	2-74
?CTYPE	Changes a process type.	2-75
?DACL	Sets, clears, or examines a default access control list.	2-77
?DADID	Gets the PID of a process's father.	2-79
?DCON	Breaks a connection (disconnects) in Ring 7.	2-80
?DDIS	Disables access to all devices.	2-81
?DEASSIGN	Cancel a character device.	2-82
?DEBL	Enables access to all devices.	2-83
?DEBUG	Calls the Debugger utility.	2-84
?DELAY	Suspends a 16-bit task for a specified interval (16-bit processes only).	2-85
?DELETE	Deletes a file entry.	2-86
?DFRSCH	Disables task rescheduling and indicates prior state of rescheduling.	2-88
?DIR	Changes the working directory.	2-89
?DQTSK	Removes from the queue one or more previously queued tasks.	2-90
?DRCON	Breaks a connection (disconnects).	2-92
?DRSCH	Disables scheduling.	2-93
?ENBRK	Enables a break file.	2-94
?ENQUE	Sends a message to IPC and spooler files.	2-98
?ERMSG	Reads the error message file.	2-99
?ERSCH	Enables multitask scheduling for the calling process.	2-102
?ESFF	Flushes shared file memory pages to disk.	2-103
?EXEC	Requests a service from EXEC.	2-104
?EXPO	Sets, clears, or examines execute-protection status.	2-141
?FDAY	Converts date to a scalar value.	2-143
?FEDFUNC	Interfaces to File Editor (FED) utility.	2-144
?FEOV	Forces end-of-volume on labeled magnetic tape.	2-148
?FIDEF	Defines a fast user device.	2-149
?FIXMT	Transmits a message from an interrupt service routine in Ring 0.	2-157
?FLOCK	Locks an object.	2-159
?FLUSH	Flushes the contents of a shared page to disk.	2-162
?FSTAT	Gets file status information.	2-163
?FTOD	Converts time of day to a scalar value.	2-171
?FUNLOCK	Unlocks an object.	2-172
?GACL	Gets a file entry's access control list (ACL).	2-174
?GBIAS	Gets the current bias factor values.	2-176
?GCHR	Reads device characteristics of a character device.	2-177
?GCLOSE	Closes a file previously opened for block I/O.	2-183
?GCPN	Gets the terminal port number.	2-185
?GCRB	Gets the base of the current resource (16-bit processes only).	2-186
?GDAY	Gets the current date.	2-187
?GDLM	Gets a delimiter table.	2-188
?GECHR	Get extended characteristics.	2-190
?GHRZ	Gets the frequency of the system clock.	2-201

?A Through ?Q

?GLINK	Gets the contents of a link entry.	2-202
?GLIST	Gets the contents of a search list.	2-203
?GMEM	Returns the number of undedicated memory pages.	2-204
?GNAME	Gets a complete pathname.	2-205
?GNFN	Lists a particular directory's entries.	2-207
?GOPEN	Opens a file for block I/O.	2-210
?GPID	Returns all active PIDs based on a host ID.	2-217
?GPORT	Returns the PID associated with a global port number.	2-219
?GPOS	Gets the current file–pointer position.	2-220
?GPRNM	Gets a program's pathname.	2-222
?GRAPHICS	Manipulates pixel maps.	2-223
?GRNAME	Returns complete pathname of generic file.	2-239
?GROUP	Changes a group access control list of a process.	2-240
?GSHPT	Lists the current shared partition size.	2-243
?GSID	Gets the system identifier.	2-244
?GTACP	Gets access control privileges.	2-245
?GTIME	Gets the time, date, and time zone.	2-247
?GTMES	Gets an initial IPC message.	2-250
?GTNAM	Returns symbol closest in value to specified input value.	2-256
?GTOD	Gets the time of day.	2-258
?GTRUNCATE	Truncates a disk file.	2-259
?GTSVL	Gets the value of a user symbol.	2-261
?GUHPI	Gets unique hardware processor identification.	2-263
?GUNM	Gets the username of a process.	2-265
?GVPID	Gets the virtual PID of a process.	2-266
?HNAME	Gets a hostname or host identifier.	2-267
?IDEF	Defines a user device.	2-269
?IDGOTO	Redirects a task's execution path.	2-278
?IDKIL	Kills a task specified by its TID.	2-279
?IDPRI	Changes the priority of a task specified by its TID.	2-280
?IDRDY	Readies a task specified by its TID.	2-281
?IDSTAT	Returns task status word (16–bit processes only).	2-282
?IDSUS	Suspends a task specified by its TID.	2-283
?IESS	Initializes an extended state save (ESS) area (16–bit processes only).	2-284
?IFPU	Initializes the floating–point unit.	2-285
?IHIST	Starts a histogram for a 16–bit process (16–bit processes only).	2-286
?ILKUP	Returns a global port number.	2-288
?IMERGE	Modifies a ring field within a global port number.	2-289
?IMSG	Receives an interrupt service message.	2-290
?INIT	Initializes a logical disk.	2-291
?INTWT	Defines a terminal interrupt task.	2-293
?IQTSK	Creates a queued task manager.	2-294
?IREC	Receives an IPC message.	2-295
?IRMV	Removes a user device.	2–304.9
?ISEND	Sends an IPC message.	2-305
?ISPLIT	Finds the owner of a port (including its ring number).	2-308

?A Through ?Q

?IS.R	Sends and then receives an IPC message.	2-309
?ITIME	Returns the OS-format internal time.	2-313
?IXIT	Exits from an interrupt service routine.	2-314
?IXMT	Transmits a message from an interrupt service routine.	2-315
?JPINIT	Initializes a job processor.	2-317
?JPMOV	Moves a job processor to a new logical processor.	2-320
?JPREL	Releases a job processor.	2-322
?JPSTAT	Gets the status of a job processor.	2-324
?KCALL	Keeps the calling resource and acquires a new resource (16-bit processes only).	2-328
?KHIST	Kills a histogram.	2-329
?KILAD	Defines a kill-processing routine.	2-330
?KILL	Kills the calling task.	2-331
?KINTR	Simulates keyboard interrupt sequences.	2-332
?KIOFF	Disables control-character terminal interrupts.	2-333
?KION	Re-enables control-character terminal interrupts.	2-334
?KWAIT	Waits for a terminal interrupt.	2-335
?LABEL	Creates a label for a magnetic tape or diskette.	2-336
?LDUINFO	Obtain logical disk information.	2-340
?LEFD	Disables LEF mode.	2-350
?LEFE	Enables LEF mode.	2-351
?LEFS	Returns the current LEF mode status.	2-352
?LMAP	Maps a lower ring.	2-353
?LOCALITY	Changes user locality.	2-354
?LOGCALLS	Logs system calls.	2-357
?LOGEV	Enters an event in the system log file.	2-360
?LPCCLASS	Gets/sets logical processor class assignments.	2-362
?LPCREA	Creates a logical processor.	2-365
?LPDELE	Deletes a logical processor.	2-367
?LPSTAT	Gets the status of a logical processor.	2-369
?MAPDV	Maps a device into logical address space.	2-373
?MBFC	Moves bytes from a customer's buffer.	2-377
?MBTC	Moves bytes to a customer's buffer.	2-379
?MDUMP	Dumps the memory image from a user-specified ring to a file.	2-381
?MEM	Lists the current unshared memory parameters.	2-383
?MEMI	Changes the number of unshared pages in the logical address space.	2-384
?MIRROR	Mirrors and synchronizes LDU images.	2-386
?MPHIST	Starts a histogram on a uni- or multi-processor system (32-bit processes only).	2-394
?MYTID	Gets the priority and TID of the calling task.	2-399
?NTIME	Sets the time, date, and time zone.	2-400
?ODIS	Disables terminal interrupts.	2-403
?OEBL	Enables terminal interrupts.	2-404
?OPEN	Opens a file.	2-405
?OPER	Creates and maintains an operator interface.	2-424
?OPEX	Communicates between the current process and an operator process.	2-437

?A Through ?Q

?OVEX	Releases an overlay and returns (16-bit processes only).	2-509
?OVKIL	Exits from an overlay and kills the calling task (16-bit processes only).	2-510
?OVLOD	Loads and goes to an overlay (16-bit processes only).	2-511
?OVREL	Releases an overlay area (16-bit processes only).	2-513
?PCLASS	Gets a process's class and locality.	2-514
?PCNX	Passes a connection from one server to another in Ring 7.	2-516
?PIDS	Gets information about PIDs.	2-517
?PMTPF	Permits access to a protected file.	2-519
?PNAME	Gets a full process name.	2-521
?PRCNX	Passes a connection from one server to another.	2-523
?PRDB/?PWRB	Performs physical block I/O.	2-525
?PRI	Changes the priority of the calling task.	2-530
?PRIPR	Changes the priority of a process.	2-531
?PRKIL	Kills all tasks of a specified priority.	2-533
?PROC	Creates a process.	2-534
?PROFILE	Performs a profile request.	2-551
?PRRDY	Readies all tasks of a specified priority.	2-558
?PRSUS	Suspends all tasks of a specified priority.	2-559
?PSTAT	Returns status information on a process.	2-560
?PTRDEVICE	Controls input from a pointer device.	2-567
?PWDCRYP	Performs a password data encryption request.	2-590
?PWRB	Performs physical block I/O.	2-592

Index

Document Set

Chapter 2 — AOS/VS, AOS/VS II, and AOS/RT32 System Calls (Continued)

?R Through ?Z

?RCALL	Releases one resource and acquires a new one (16-bit processes only).	2-594
?RCHAIN	Chains to a new procedure (16-bit processes only).	2-595
?RDB/?WRB	Performs (reads/writes) block I/O.	2-596
?RDUDA/?WRUDA	Reads/writes a user data area (UDA).	2-602
?READ/?WRITE	Performs (reads/writes) record I/O.	2-604
?REC	Receives an intertask message.	2-627
?RECNW	Receives an intertask message without waiting.	2-628
?RECREATE	Recreates a file.	2-629
?RELEASE	Releases an initialized logical disk.	2-630
?RENAME	Renames a file.	2-631
?RESCHED	Reschedules current time slice (32-bit processes only).	2-633

?R Through ?Z

?RESIGN	Resigns as a server.	2-634
?RETURN	Terminates the calling process and passes the termination message to the father.	2-635
?RINGLD	Loads a program file into a specified ring.	2-637
?RNAME	Determines whether a pathname contains a reference to a remote host.	2-639
?RNGPR	Returns the .PR filename for a ring.	2-640
?RNGST	Stops ?RINGLD from loading lower rings.	2-642
?RPAGE	Releases a shared page.	2-643
?RTODC	Reads the time-of-day conversion data.	2-645
?RUNTM	Gets runtime statistics on a process.	2-648
?SACL	Sets a new access control list (ACL).	2-650
?SATR	Sets or removes the permanent attribute for a file or directory.	2-653
?SBIAS	Sets the bias factors.	2-655
?SCHR	Sets a character device's characteristics.	2-656
?SCLOSE	Closes a file previously opened for shared access.	2-658
?SDAY	Sets the system calendar.	2-660
?SDBL	Disables a BSC line.	2-661
?SDLM	Sets a delimiter table.	2-662
?SDPOL	Defines a polling list or a poll-address/ select-address pair.	2-664
?SDRT/?SERT	Disables/re-enables a relative terminal.	2-667
?SEBL	Enables a BSC line.	2-669
?SECHR	Sets extended characteristics of a device.	2-679
?SEND	Sends a message to a terminal.	2-681
?SERMSG	Returns text for associated error code (16-bit processes only).	2-683
?SERT	Re-enables a relative terminal.	2-685
?SERVE	Becomes a server.	2-685
?SGES	Gets BSC error statistics.	2-686
?SIGNL	Signals another task.	2-688
?SIGWT	Signals another task and then waits for a signal.	2-690
?SINFO	Gets selected information about the current operating system.	2-692
?SLIST	Sets the search list for the calling process.	2-695
?SONS	Gets a list of son processes for a target PID.	2-696
?SOPEN	Opens a file for shared access.	2-699
?SOPPF	Opens a protected shared file.	2-701
?SPAGE	Performs a shared-page read.	2-704
?SPOS	Sets the position of the file pointer.	2-707
?SRCV	Receives data or a control sequence over a BSC line.	2-709
?SSHPT	Establishes a new shared partition.	2-718
?SSID	Sets the system identifier.	2-719
?SSND	Sends data or a control sequence over a BSC line.	2-720
?STMAP	Sets the data channel map.	2-729
?STOD	Sets the system clock.	2-732
?STOM	Sets the time-out value for a device.	2-733
?SUPROC	Enters, leaves, or examines Superprocess mode.	2-735

?R Through ?Z

?SUS	Suspends the calling task.	2-736
?SUSER	Enters, leaves, or examines Superuser mode.	2-737
?SYLOG	Manipulates the system log files.	2-739
?SYSPRV	Enters, leaves, or examines a privilege state.	2-744
?TASK	Initiates one or more tasks.	2-747
?TERM	Terminates a process.	2-754
?TIDSTAT	Returns status of target task (32-bit processes only).	2-756
?TLOCK	Protects a task from being redirected.	2-757
?TMSG	Defines the termination message format.	2-759
?TPID	Translates a PID.	2-769
?TPORT	Translates a local port number to its global equivalent.	2-770
?TRCON	Reads a task message from the process terminal.	2-771
?TRUNCATE	Truncates a file at the current position.	2-773
?TUNLOCK	Allows a task to be redirected.	2-774
?UBLPR	Unblocks a process.	2-776
?UIDSTAT	Returns the status of a task and an unambiguous identifier.	2-777
?UNWIND	Unwinds the stack and restores the previous environment (16-bit processes only).	2-778
?UNWIRE	Unwires pages previously wired.	2-779
?UPDATE	Flushes file descriptor information.	2-780
?VALAD	Validates a logical address.	2-781
?VALIDATE	Validates an area for Read or Write access.	2-782
?VCUST	Verifies a customer in Ring 7.	2-786
?VMEM	Changes the partition size of a process.	2-787
?VRCUST	Verifies a customer in a specified ring.	2-789
?VTFCREATE	Creates a Virtual Timer Facility timer.	2-790
?VTFKILL	Kills a Virtual Timer Facility timer.	2-797
?VTFMODIFY	Modifies a Virtual Timer Facility timer.	2-799
?VTFSUS	Suspends or Restarts a Virtual Timer Facility timer.	2-804
?VTFXIT	Exits from a Virtual Timer Facility interrupt routine.	2-806
?WALKBACK	Returns information about previous frames in the stack (16-bit processes only).	2-807
?WDELAY	Suspends a task for a specified time (32-bit processes only).	2-809
?WHIST	Starts a histogram (32-bit processes only).	2-810
?WINDOW	Manipulates windows.	2-813
?WIRE	Wires pages to the working set.	2-842
?WRB	Writes block I/O.	2-844
?WRITE	Writes record I/O.	2-844
?WRUDA	Writes a user data area (UDA).	2-844
?WTSIG	Waits for a signal from another task or process.	2-845
?WTVERR	Waits for a Virtual Timer Facility error message.	2-846
?WTVSIG	Waits for a Virtual Timer Facility signal.	2-848
?XCREATE	Creates a file or directory (extended).	2-849
?XFSTAT	Gets file status information (extended).	2-862
?XGTACP	Gets access control privileges (extended).	2-882
?XINIT	Initializes a logical disk (extended).	2-886
?XMT	Transmits an intertask message.	2-898

?R Through ?Z

?XMTW	Transmits an intertask message and waits for it to be received.	2-899
?XPSTAT	Returns extended status information on a process.	2-900

Appendix A — Sample Programs

Program Set 1 — HEAR.SR, SPEAK.SR, SON.SR	A-3
Program Set 2 — RUNTIME.SR	A-11
Program Set 3 — RINGLOAD.SR, INRING.SR, and GATE.ARRAY.SR	A-14
Program Set 4 — FILCREATE.SR	A-19
Program Set 5 — WRITE.SR	A-22
Program Set 6 — DLIST.SR	A-26
Program Set 7 — NEWTASK.SR	A-29
Program Set 8 — BOOMER.SR	A-32
Program Set 9 — TIMEOUT.SR	A-37
Program Set 10 — DIRCREATE.F77 and CHECK.F77	A-39
Program Set 11 — CREATE_WINDOW.SR	A-46
Program Set 12 — GRAPHICS_SAMPLE.SR	A-56

Appendix B — System Log Record Format

Reporting the Contents of the SYSLOG File	B-1
Reading the SYSLOG File	B-1
Record Header Format	B-2
SYSLOG Record Formats	B-3
Anatomy of a System Log File Record	B-20

Index

Document Set

Tables

Table	?A Through ?Q
2-1	Summary of AOS/VS and AOS/RT32 System Calls 2-3
2-2	Contents of ?BLKIO Packet 2-21
2-3	Contents of ?CLASS Packet 2-37
2-4	Contents of ?CLSCHED Packet 2-44
2-5	Contents of ?CLSTAT Main Packet 2-48
2-6	Contents of ?CLSTAT Subpacket 2-50
2-7	Contents of ?CMATRIX Main Packet 2-53
2-8	Contents of ?CMATRIX Subpacket 2-54
2-8.1.	Valid ?CONFIG_FUNCTION Function Codes 2-58.1
2-8.2	?CONFIG GET CURRENT ROUTE Function Subpacket Contents 2-58.3
2-8.3	?CONFIG RESET_MRD_CHANNEL Function Subpacket Contents 2-58.4
2-8.4	?CONFIG RESET_MRC_CONTROLLER Function Subpacket Contents 2-58.5
2-8.5	Contents of the ?CONINFO Packet 2-58.7
2-8.6	Input Values to ?CON_PKTUSER_FLGS Offset 2-58.8
2-8.7	?CON_RET_TYPES Return Buffer Console Types and Definitions 2-58.9
2-8.8	Contents of ?CON_TCP_RET_TYPE Packet 2-58.10
2-8.9	Contents of ?CON_XNS_RET_TYPE Packet 2-58.12
2-8.10	Contents of ?CON_CON_RET_TYPE Packet 2-58.13
2-8.11	Contents of ?CON_TNET_RET_TYPE Return Packet 2-58.14
2-8.12	Contents of ?CON_ITC_MIN_DATA Packet 2-58.15
2-8.13	Contents of ?CON_TSC_MIN_DATA Packet 2-58.16
2-8.14	Contents of ?CON_PVC_RET_TYPE Packet 2-58.17
2-8.15	?CON_PVC_RET_TYPE Subpacket Types and Definitions 2-58.18
2-8.16	Contents of ?CON_PVC_NAME Subpacket 2-58.19
2-8.17	Contents of ?CON_PVC_NAME_PORT Subpacket 2-58.20
2-8.18	Contents of ?CON_PVC_IP Subpacket 2-58.21
2-8.19	Contents of ?CON_PVC_IP_PORT Subpacket 2-58.22
2-8.20	Contents of ?CON_PVC_ETH Subpacket 2-58.23
2-8.21	Contents of ?CON_PVC_PORT Subpacket 2-58.24
2-8.22	Contents of ?CON_PVC_NET Subpacket 2-58.25
2-9	Valid ?CREATE File Types 2-61
2-10	Contents of ?CREATE IPC Packet 2-63
2-11	Contents of ?CREATE Directory Packet 2-65
2-12	Contents of ?CREATE Packet for Other File Types 2-67
2-13	Contents of ?ENBRK Packet 2-96
2-14	Flags for EXEC Functions ?XFXUN and ?XFXML 2-108
2-15	Contents of ?EXEC Packet for Tape Backup 2-111
2-16	Contents of ?EXEC Packet for Queue Requests 2-114
2-17	Contents of ?XFUSR Subpacket of ?EXEC System Call 2-125
2-18	Contents of AOS/VS ?EXEC Packet for Hold, Unhold, or Cancel Queue Requests 2-128
2-19	Contents of ?EXEC Packet for Status Information 2-129
2-20	Contents of ?EXEC Packet for Extended Status Information 2-130
2-21	Contents of Selected Offsets in the ?EXEC Packet for the ?XFMOD Function 2-135

Table

?A Through ?Q

2-22	Queue Status Bit Definitions in Offset ?XQ1FG	2-136
2-23	?EXEC Queue Types in Offset ?XQQT	2-138
2-24	Contents of ?FEDFUNC Packet to Evaluate a FED String	2-147
2-25	Contents of Map Definition Entry	2-154
2-26	Contents of ?FLOCK Packet	2-161
2-27	Flags Returned in Offset ?SSTS	2-169
2-28	Contents of ?FUNLOCK Packet	2-173
2-29	Character Device Characteristics Words	2-178
2-30	Commonly Used Device Characteristics	2-181
2-33	Device Types for Rubout Echo and Cursor Controls	2-200
2-34	Contents of ?GNFN Packet	2-208
2-35	Filename Template Characters	2-208
2-36	Contents of ?GOPEN Packet for IPC Files	2-211
2-37	Contents of Standard ?GOPEN Packet	2-212
2-38	Option Flags for Offset ?ODF1	2-215
2-39	Valid Format Options (DPM disks)	2-216
2-40	Contents of ?GPID Packet	2-218
2-41	?GRAPHICS Function Codes	2-224
2-42	Contents of the ?GRAPHICS Main Packet	2-225
2-43	Contents of the ?GRAPH_OPEN_WINDOW_PIXELMAP Subpacket	2-227
2-44	Contents of the ?GRAPH_CREATE_MEMORY_PIXELMAP Subpacket	2-228
2-45	Contents of the ?GRAPH_PIXELMAP_STATUS Subpacket	2-230
2-46	Contents of the ?GRAPH_SET_CLIPRECTANGLE Subpacket	2-231
2-47	Contents of the ?GRAPH_MAP_PIXELMAP Subpacket	2-233
2-48	Contents of the ?GRAPHICS_SET_DRAW_ORIGIN Subpacket	2-237
2-49	Contents of the ?GRAPHICS_GET_DRAW_ORIGIN Subpacket	2-238
2-50	Contents of ?GROUP Packet	2-242
2-51	Contents of ?GTIME Packet	2-249
2-52	Contents of ?GTMES Packet	2-251
2-53	Input Parameters for Offset ?GREQ (Request Types)	2-252
2-54	Output from ?GTMES Requests	2-254
2-55	Contents of ?GTRUNCATE Packet	2-260
2-56	Contents of ?GUHPI Packet	2-264
2-57	Contents of Map Definition Entry	2-274
2-58	Structure of ?IHIST Array	2-287
2-59	Contents of ?INIT Packet	2-292
2-60	Contents of ?IREC Header	2-297
2-61	Termination Codes for 16-Bit Processes	2-298
2-62	Process Termination Codes in Offset ?IUFL for ?IREC and ?ISEND Headers	2-299
2-63	?TEXT Code Termination Messages Sent on an A-Type 32-Bit Process User Trap ..	2-301
2-63.1	?TRAP Termination Messages for A-Type 16-Bit Processes	2-303
2-63.2	Contents of Termination Message from a 32-bit B- or C-Type Process	2-304.2
2-64	Contents of Termination Message from a 16-bit B- or C-Type Process	2-304.6
2-65	Contents of ?ISEND Header	2-306
2-66	Contents of ?IS.R Header	2-311
2-67	Contents of ?JPINIT Packet	2-319
2-68	Contents of ?JPMOV Packet	2-321
2-69	Contents of ?JPREL Packet	2-323
2-70	Contents of ?JPSTAT Main Packet	2-325

Table**?A Through ?Q**

2-71	Contents of ?JPSTAT General Information Subpacket	2-326
2-72	Contents of ?JPSTAT Specific Information Subpacket	2-327
2-73	Contents of ?LABEL Packet	2-338
2-74	Contents of ?LDUINFO Main Packet	2-341
2-75	Contents of ?LDU_PKT Subpacket	2-344
2-76	?PIECE_PKT Subpacket Contents	2-348
2-77	Contents of ?LOCALITY Packet	2-356
2-78	Contents of ?LPCLASS Packet	2-364
2-79	Contents of ?LPCREA Packet	2-366
2-80	Contents of ?LPDELE Packet	2-368
2-81	Contents of ?LPSTAT Main Packet	2-370
2-82	Contents of ?LPSTAT General Information Subpacket	2-371
2-83	Contents of ?LPSTAT Specific Information Subpacket	2-372
2-84	Contents of ?MAPDV Packet	2-376
2-85	Contents of ?MIRROR Packet	2-388
2-86	Contents of 16-Bit ?MIRROR Packet	2-391
2-87	Contents of ?MPHIST Packet	2-397
2-88	Structure and Contents of ?MPHIST Histogram Array	2-398
2-89	Contents of ?NTIME Packet	2-402
2-90	Contents of ?OPEN Packet	2-408
2-91	File Creation Options for Offset ?ISTI	2-413
2-92	Common File Types You Can Create with ?OPEN	2-414
2-93	Contents of ?OPEN Extension Packet for Pipes	2-417
2-94	Contents of Labeled Magnetic Tape Packet Extension	2-420
2-95	Contents of ?OPER Main Packet	2-426
2-96	Contents of ?OPON Subpacket	2-427
2-97	Contents of ?OPOFF Subpacket	2-429
2-97	Contents of ?OPOFF Subpacket	2-430
2-98	Contents of ?OPSEND Subpacket	2-431
2-99	Contents of ?OPRCV Subpacket	2-433
2-100	Contents of ?OPRESP Subpacket	2-434
2-101	Contents of ?OPINFO Subpacket	2-436
2-102	Contents of ?OPEX Main Packet	2-439
2-103	Contents of Access Command Subpacket	2-442
2-104	Contents of Align Command Subpacket	2-443
2-105	Contents of Batch_List Command Subpacket	2-444
2-106	Contents of Batch_Output Command Subpacket	2-445
2-107	Contents of Binary Command Subpacket	2-446
2-108	Contents of Brief Command Subpacket	2-447
2-109	Contents of Cancel Command Subpacket	2-448
2-110	Contents of Consolestatus Command Subpacket	2-450
2-111	Contents of Continue Command Subpacket	2-451
2-112	Contents of CPL Command Subpacket	2-452
2-113	Contents of Create Command Subpacket	2-453
2-114	?OPEX Queue Types in Offset ?ZCRQ	2-453
2-115	Contents of Defaultforms Command Subpacket	2-454
2-116	Contents of Disable Command Subpacket	2-455
2-117	Contents of Dismounted Command Subpacket	2-456

Table**?A Through ?Q**

2-118	Contents of Elongate Command Subpacket	2-457
2-119	Contents of Enable Command Subpacket	2-459
2-120	Contents of Even Command Subpacket	2-460
2-121	Contents of Flush Command Subpacket	2-461
2-122	Contents of Forms Command Subpacket	2-462
2-123	Contents of Halt Command Subpacket	2-463
2-124	Contents of Headers Command Subpacket	2-464
2-125	Contents of Hold Command Subpacket	2-465
2-126	Contents of Limit Command Subpacket	2-466
2-127	Contents of Logging Command Subpacket	2-468
2-128	Contents of LPP Command Subpacket	2-469
2-129	Contents of Mapper Command Subpacket	2-470
2-130	Contents of Mounted Command Subpacket	2-472
2-131	Contents of Mountstatus Command Subpacket	2-474
2-132	Contents of Operator Command Subpacket	2-476
2-133	Contents of Pause Command Subpacket	2-477
2-134	Contents of Premount Command Subpacket	2-478
2-135	Contents of Priority Command Subpacket	2-479
2-136	Contents of Prompts Command Subpacket	2-480
2-137	Contents of Qpriority Command Subpacket	2-482
2-138	Contents of Refused Command Subpacket	2-483
2-139	Contents of Release Command Subpacket	2-484
2-140	Contents of Restart Command Subpacket	2-485
2-141	Contents of Silence Command Subpacket	2-486
2-142	Contents of Spoolstatus Command Subpacket	2-488
2-143	Contents of Stack Command Subpacket	2-491
2-144	Contents of Start Command Subpacket	2-493
2-145	Contents of Status Command Subpacket	2-496
2-146	Contents of Stop Command Subpacket	2-499
2-147	Contents of Trailers Command Subpacket	2-500
2-148	Contents of Unhold Command Subpacket	2-501
2-149	Contents of Unitstatus Command Subpacket	2-503
2-150	Contents of Unlimit Command Subpacket	2-504
2-151	Contents of Unsilence Command Subpacket	2-505
2-152	Contents of Subpacket User-Command	2-507
2-153	Contents of Verbose Command Subpacket	2-508
2-154	Contents of ?PCLASS Packet	2-515
2-155	Contents of ?PIDS Packet	2-518
2-156	Contents of ?PMTPF Packet	2-520
2-157	Contents of ?PRDB/?PWRB Packet	2-527
2-158	?PRDB/?PWRB Packet: Controller Status Words	2-528
2-159	Error Reports Returned in ?PRDB/?PWRB Offsets	2-529
2-160	Contents of ?PROC Packet	2-538
2-161	Privilege Bits in Offset ?PPRV	2-542
2-162	Contents of ?PROC Parameter Packet Extension for AOS/VS and AOS/RT32	2-547
2-163	Contents of ?PROC Parameter Packet Extension for AOS/VS II	2-548
2-164	Contents of ?PROFILE Parameter Packet	2-553
2-165	Contents of ?PSTAT Parameter Packet	2-563
2-166	?PTRDEVICE Function Codes	2-568

Table**?A Through ?Q**

2-167	Contents of the ?PTRDEVICE Main Packet	2-570
2-168	Flags for Selecting Pointer Events (Flag Word ?PTRDEV_SET_EVTS.EVTS)	2-575
2-169	Flags for Selecting Buttons (Flag Word ?PTRDEV_SET_EVTS.BTNS)	2-576
2-170	Contents of the ?PTRDEV_SET_DELTA Subpacket	2-577
2-171	Contents of the ?PTRDEV_LAST_EVENT Subpacket	2-578
2-172	Contents of the ?PTRDEV_SET_POINTER Subpacket	2-580
2-173	Contents of the ?PTRDEV_GET_PTR_STATUS Subpacket	2-582
2-174	Flags for Each Possible Pointer Device Event (Flag Word ?PTRDEV_GSTATUS.EVTS)	2-584
2-175	Contents of the ?PTRDEV_GENERATE_EVENT Subpacket	2-586
2-176	Contents of the ?PTRDEV_GET_PTR_LOCATION Subpacket	2-588
2-177	Contents of the ?PTRDEV_GET_TABLET_LOCATION Subpacket	2-589
2-178	Contents of ?PWDCRYP Packet	2-591

?R Through ?Z

2-179	Contents of ?RDB/?WRB Packet	2-598
2-180	Contents of the Standard ?READ/?WRITE Packet	2-608
2-181	Contents of Screen-Management Packet Extension	2-613
2-182	Contents of Selected Field Translation Packet Extension	2-618
2-183	Contents of the New Screen Management Packet	2-623
2-185	Supported (Y) and Unsupported (N) New Screen Management Packet Features in AOS/VS	2-626
2-186	Contents of ?RNGPR Packet	2-641
2-187	Contents of ?RTODC Packet	2-647
2-188	Contents of ?SEBL Packet	2-671
2-189	BSC Protocol Data-Link Control Characters (DLCC)	2-674
2-190	Contents of the ?SERMSG Packet	2-684
2-191	Contents of ?SGES Packet	2-687
2-192	Contents of ?SONS Packet	2-698
2-193	Contents of ?SOPPF Packet	2-702
2-194	Contents of ?SPAGE Packet	2-705
2-195	File-Pointer Settings	2-708
2-196	Contents of ?SRCV Packet	2-711
2-197	Masks Returned on ?SRCV System Calls	2-716
2-198	Contents of ?SSND Packet	2-722
2-199	?SSND Call Types	2-726
2-200	Contents of ?SYSPRV Packet	2-746
2-201	Contents of Standard Task Definition Packet	2-749
2-202	Contents of Extended Task Definition Packet	2-752
2-203	Contents of Termination Message a 32-bit Process Receives	2-762
2-204	Contents of Termination Message a 16-bit Process Receives	2-766
2-205	?VALIDATE Functions and Their Codes	2-783
2-206	Contents of ?VMEM Packet	2-788
2-207	Contents of ?VTFCREATE Packet	2-792
2-208	Contents of ?VTFKILL Packet	2-798
2-209	Contents of ?VTFMODIFY Packet	2-801
2-210	Histogram Array Structure	2-812
2-211	?WINDOW Function Codes	2-814
2-212	Contents of the ?WINDOW Main Packet	2-817

Table

?R Through ?Z

2-213 Contents of the ?WIN_CREATE_WINDOW Subpacket 2-819

2-214 Contents of the ?WIN_DEFINE_PORTS Subpacket 2-824

2-215 Flags in Flag Word ?WIN_SINT.FLAGS 2-829

2-216 Border Types (Offset ?WIN_SINT.BORDER_TYPE) 2-829

2-217 Flags in the ?WIN_GET_USER_INTERFACE
Flag Word (Flag Word ?WIN_GINT.FLAGS) 2-830

2-218 Border Types (Offset ?WIN_GINT.BORDER_TYPE) 2-831

2-219 Contents of the ?WIN_GTITLE Subpacket 2-832

2-220 Contents of the ?WIN_WINDOW_STATUS Subpacket 2-835

2-221 Contents of the ?WIN_DEVICE_STATUS Subpacket 2-839

2-222 Contents of ?XCREATE Main Packet 2-850

2-223 Valid ?XCREATE File Types 2-852

2-224 Contents of ?XCREATE Time-Block Subpacket 2-853

2-225 Contents of ?XCREATE Other Subpacket 2-854

2-226 Contents of ?XCREATE Directory Subpacket 2-856

2-227 Contents of ?XCREATE IPC Subpacket 2-857

2-228 Contents of ?XCREATE Link Subpacket 2-857

2-229 Valid ?XFSTAT File Types 2-863

2-230 ?XFSTAT IPC File Status 2-867

2-231 ?XFSTAT Status Flags 2-869

2-232 ?XFSTAT Directory and Other Packet File Status 2-871

2-233 ?XFSTAT Unit Packet File Status 2-876

2-234 Contents of ?XGTACP Packet 2-884

2-235 Contents of ?XINIT Packet 2-889

2-236 Contents of 16-Bit ?XINIT Packet 2-894

2-237 Contents of ?XPSTAT Standard Parameter Packet 2-905

2-238 Contents of ?XPSTAT Extension Packet for AOS/V5 and AOS/V5 II 2-908

A-1 Sample Program Sets and their Descriptions A-1

B-1 SYSLOG Event Codes and Record Lengths B-3

Figures

Figure	?A Through ?Q	
1-1	Parametric Coding Example	1-3
1-2	Listing File of Program DIRCREATE.SR	1-6
2-1	Structure of ?BLKIO Packet	2-20
2-2	Examples of Read Next Allocated Element Option	2-23
2-3	Structure of Physical I/O Controller Status Block	2-25
2-4	Structure of ?CLASS Packet	2-37
2-5	Structure of ?CLOSE Packet	2-39
2-6	Structure of ?CLSCHED Packet	2-44
2-7	Structure of ?CLSTAT Main Packet	2-47
2-8	Structure of ?CLSTAT Subpacket	2-49
2-9	Structure of ?CMATRIX Main Packet	2-52
2-10	Structure of ?CMATRIX Subpacket	2-53
2-11	Addresses of ?CMATRIX Main Packet and Its Subpacket Offsets	2-55
2-11.1	Structure of ?CONFIG Main Packet	2-58.1
2-11.2	Structure of the ?CONFIG_GET_CURRENT_ROUTE Function SubPacket	2-58.2
2-11.3	Structure of the ?CONFIG_RESET_MRC_CHANNEL Function SubPacket	2-58.4
2-11.4	Structure of the ?CONFIG_RESET_MRC_CTRLR Function SubPacket	2-58.5
2-11.5	Structure of ?CONINFO Main Packet	2-58.7
2-11.6	Structure of ?CON_TCP_RET_TYPE Return Packet	2-58.10
2-11.7	Structure of ?CON_XNS_RET_TYPE Return Packet	2-58.11
2-11.8	Structure of ?CON_CON_RET_TYPE Return Packet	2-58.12
2-11.9	Structure of ?CON_TNET_RET_TYPE Return Packet	2-58.13
2-11.10	Structure of ?CON_ITC_MIN_DATA Return Packet	2-58.14
2-11.11	Structure of ?CON_TSC_MIN_DATA Return Packet	2-58.15
2-11.12	Structure of ?CON_PVC_RET_TYPE Return Packet	2-58.16
2-11.13	Structure of ?CON_PVC_NAME Return Packet	2-58.19
2-11.14	Structure of ?CON_PVC_NAME_PORT Return Packet	2-58.20
2-11.15	Structure of ?CON_PVC_IP Return Packet	2-58.21
2-11.16	Structure of ?CON_PVC_IP_PORT Return Packet	2-58.22
2-11.17	Structure of ?CON_PVC_ETH Return Packet	2-58.23
2-11.18	Structure of ?CON_PVC_PORT Return Packet	2-58.24
2-11.19	Structure of ?CON_PVC_NET Return Packet	2-58.25
2-12	Structure of ?CPMAX packet	2-59
2-13	Structure of ?CREATE IPC Packet	2-62
2-14	Structure of ?CREATE Time Block	2-63
2-15	Structure of ?CREATE Directory Packet	2-64
2-16	Structure of ?CREATE Packet for Other File Types	2-66
2-17	Structure of ?CRUDA Packet	2-71
2-18	Structure of ?DELETE Packet	2-87
2-19	Extended Task Definition Packet	2-91
2-20	Structure of ?ENBRK Packet	2-96
2-21	Error Code Structure in ERMES File	2-100
2-22	Structure of ?EXEC Packet for Unlabeled Mount Function ?XFMUN	2-106
2-23	Structure of ?EXEC Extended Packet for Unlabeled Mount Function ?AFXUN	2-107

Figure**?A Through ?Q**

2-24	Structure of ?EXEC Packet for Labeled Mount Function ?XFMLT	2-107
2-25	Structure of ?EXEC Extended Packet for Labeled Mount Function ?XFXML	2-108
2-26	Structure of ?EXEC Packet for Dismounting a Tape, ?XFDUN	2-109
2-27	Structure of ?EXEC Packet for Tape Backup	2-110
2-28	Structure of ?EXEC Packet for Queue Requests	2-113
2-29	Structure of the IPC Print Notification Message from ?EXEC	2-123
2-30	Structure of ?XFUSR Subpacket of ?EXEC System Call	2-124
2-31	Structure of AOS/VS ?EXEC Packet for Hold, Unhold, or Cancel Queue Requests.	2-127
2-32	Structure of ?EXEC Packet for Status Information	2-129
2-33	Structure of ?EXEC Packet for Extended Status Information	2-130
2-34	Structure of ?EXEC Packet for a MOUNT Queue Request	2-131
2-35	Structure of ?EXEC Packet for Dismounting a Unit (extended request)	2-132
2-36	Structure of ?EXEC Packet for Changing Queuing Parameters	2-133
2-37	Structure of ?EXEC Packet for Obtaining Queue Names	2-137
2-38	Structure of ?EXEC Packet for Obtaining QDISPLAY Information	2-139
2-39	Structure of ?FEDFUNC Packet to Change Radix	2-145
2-40	Structure of ?FEDFUNC Packet to Open Symbol Table File	2-145
2-41	Structure of ?FEDFUNC Packet to Evaluate a FED String	2-145
2-42	Structure of ?FEDFUNC Packet to Disassemble an Instruction	2-146
2-43	Structure of ?FEDFUNC Packet to Insert a Temporary Symbol	2-146
2-44	Structure of ?FEDFUNC Packet to Delete a Temporary Symbol	2-147
2-45	Structure of Device Control Table (DCT)	2-152
2-46	Structure of Map Definition Table	2-153
2-47	Structure of ?FLOCK Packet	2-160
2-48	Structure of ?FSTAT Packet for Unit Files	2-165
2-49	Structure of ?FSTAT Packet for IPC Files	2-166
2-50	Structure of ?FSTAT Packet for Directory Files	2-167
2-51	Structure of ?FSTAT Packet for Other File Types	2-168
2-52	Structure of ?SSTS Packet	2-169
2-53	Structure of ?FUNLOCK Packet	2-173
2-54	Structure of ?GACL Packet	2-175
2-55	Structure of ?GNFN Packet	2-208
2-56	Structure of ?GOPEN Packet for IPC Files	2-211
2-57	Structure of Standard ?GOPEN Packet	2-212
2-58	Structure of ?GOPEN Packet Extension	2-215
2-59	Structure of ?GPID Packet	2-218
2-60	Structure of ?GPOS Packet	2-221
2-61	Structure of the ?GRAPHICS Main Packet	2-225
2-62	Structure of the ?GRAPH_OPEN_WINDOW_PIXELMAP Subpacket	2-226
2-63	Structure of the ?GRAPHICS_CREATE_MEMORY_PIXELMAP Subpacket	2-228
2-64	Structure of the ?GRAPH_PIXELMAP_STATUS Subpacket	2-229
2-65	Structure of the ?GRAPH_SET_CLIP_RECTANGLE Subpacket	2-231
2-66	Structure of the ?GRAPH_MAP_PIXELMAP Subpacket	2-233
2-67	Structure of the ?GRAPH_WRITE_PALETTE Subpacket	2-235
2-68	Structure of the ?GRAPH_READ_PALETTE Subpacket	2-236
2-69	Structure of the ?GRAPHICS_SET_DRAW_ORIGIN Subpacket	2-237
2-70	Structure of the ?GRAPHICS_GET_DRAW_ORIGIN Subpacket	2-238
2-71	Structure of ?GROUP Packet	2-241
2-72	Structure of ?GROUP log entry	2-241

Figure**?A Through ?Q**

2-73	Structure of ?GTIME Packet	2-248
2-74	Structure of ?GTMES Packet	2-251
2-75	Structure of ?GTRUNCATE Packet	2-260
2-76	Structure of ?GUHPI Packet	2-264
2-77	Structure of Device Control Table (DCT) for 32-Bit Processes	2-271
2-78	Structure of Device Control Table (DCT) for 16-Bit Processes	2-272
2-78.1	Structure of Extended Packet for 16-Bit Processes	2-272
2-79	Structure of Map Definition Table	2-273
2-80	Structure of ?IHIST Packet	2-287
2-81	Structure of ?IREC Header	2-296
2-82	Structure of Offset ?IUFL	2-298
2-82.1	Structure of Termination Message from a 32-bit B- or C-Type Process	2-304.1
2-82.2	Structure of Termination Message from a 16-bit B- or C-Type Process	2-304.4
2-83	Structure of ?ISEND Header	2-306
2-84	Structure of ?IS.R Header	2-310
2-85	Structure of ?JPINIT Packet	2-318
2-86	Structure of ?JPMOV Packet	2-321
2-87	Structure of ?JPREL Packet	2-323
2-88	Structure of ?JPSTAT Main Packet	2-325
2-89	Structure of ?JPSTAT General Information Subpacket	2-326
2-90	Structure of ?JPSTAT Specific Information Subpacket	2-326
2-91	Structure of ?LABEL Packet	2-337
2-92	Structure of ?LDUINFO Main Packet	2-341
2-93	Structure of ?LDU_PKT Subpacket	2-343
2-94	Structure of ?LDU_PKT Array Record	2-346
2-95	Structure of ?PIECE_PKT Subpacket	2-347
2-96	Structure of ?LOCALITY Packet	2-355
2-97	Structure of ?LOGEV Event Logging Format	2-361
2-97.1	Structure of Termination Message from a 32-bit B- or C-Type Process	2-304.1
2-97.2	Structure of Termination Message from a 16-bit B- or C-Type Process	2-304.4
2-98	Structure of ?LPCLASS Packet	2-363
2-99	Structure of ?LPCREA Packet	2-366
2-100	Structure of ?LPDELE Packet	2-368
2-101	Structure of ?LPSTAT Main Packet	2-370
2-102	Structure of ?LPSTAT General Information Subpacket	2-371
2-103	Structure of ?LPSTAT Specific Information Subpacket	2-371
2-104	Structure of ?MAPDV Packet	2-375
2-105	Structure of ?MBFC Packet	2-378
2-106	Structure of ?MBTC Packet	2-380
2-107	Structure of ?MIRROR Packet	2-387
2-108	Structure of ?MIRROR Subpacket	2-390
2-109	Structure of 16-Bit ?MIRROR Packet	2-390
2-110	Structure of 16-Bit ?MIRROR Subpacket	2-393
2-111	Structure of ?MPHIST Packet	2-396
2-112	Structure of ?NTIME Packet	2-401
2-113	Structure of ?OPEN Packet	2-407
2-114	Sample Delimiter Table	2-416
2-115	Structure of ?OPEN Extension Packet for Pipes	2-416
2-116	Structure of Labeled Magnetic Tape Packet Extension	2-419

Figure

?A Through ?Q

2-117	Structure of ?OPER Main Packet	2-425
2-118	Structure of ?OPON Subpacket	2-427
2-119	Structure of ?OPOFF Subpacket	2-429
2-120	Structure of ?OPSEND Subpacket	2-432
2-121	Structure of ?OPRCV Subpacket	2-432
2-122	Structure of ?OPRESP Subpacket	2-434
2-123	Structure of ?OPINFO Subpacket	2-435
2-124	Structure of ?OPEX Main Packet	2-438
2-125	Structure of Access Command Subpacket	2-442
2-126	Structure of Align Command Subpacket	2-443
2-127	Structure of Batch_List Command Subpacket	2-444
2-128	Structure of Batch_Output Command Subpacket	2-445
2-129	Structure of Binary Command Subpacket	2-446
2-130	Structure of Brief Command Subpacket	2-447
2-131	Structure of Cancel Command Subpacket	2-448
2-132	Structure of Consolestatus Command Subpacket	2-449
2-133	Structure of Continue Command Subpacket	2-451
2-134	Structure of CPL Command Subpacket	2-452
2-135	Structure of Create Command Subpacket	2-453
2-136	Structure of Defaultforms Command Subpacket	2-454
2-137	Structure of Disable Command Subpacket	2-455
2-138	Structure of Dismounted Command Subpacket	2-456
2-139	Structure of Elongate Command Subpacket	2-457
2-140	Structure of Enable Command Subpacket	2-458
2-141	Structure of Even Command Subpacket	2-460
2-142	Structure of Flush Command Subpacket	2-461
2-143	Structure of Forms Command Subpacket	2-462
2-144	Structure of Halt Command Subpacket	2-463
2-145	Structure of Headers Command Subpacket	2-464
2-146	Structure of Hold Command Subpacket	2-465
2-147	Structure of Limit Command Subpacket	2-466
2-148	Structure of Logging Command Subpacket	2-467
2-149	Structure of LPP Command Subpacket	2-469
2-150	Structure of Mapper Command Subpacket	2-470
2-151	Structure of Mounted Command Subpacket	2-472
2-152	Structure of Mountstatus Command Subpacket	2-473
2-153	Structure of Operator Command Subpacket	2-476
2-154	Structure of Pause Command Subpacket	2-477
2-155	Structure of Premount Command Subpacket	2-478
2-156	Structure of Priority Command Subpacket	2-479
2-157	Structure of Prompts Command Subpacket	2-480
2-158	Structure of Qpriority Command Subpacket	2-481
2-159	Structure of Refused Command Subpacket	2-483
2-160	Structure of Release Command Subpacket	2-484
2-161	Structure of Restart Command Subpacket	2-485
2-162	Structure of Silence Command Subpacket	2-486
2-163	Structure of Spoolstatus Command Subpacket	2-487
2-164	Structure of Stack Command Subpacket	2-491
2-165	Structure of Start Command Subpacket	2-492

Figure**?A Through ?Q**

2-166	Structure of Status Command Subpacket	2-495
2-167	Structure of Stop Command Subpacket	2-499
2-168	Structure of Trailers Command Subpacket	2-500
2-169	Structure of Unhold Command Subpacket	2-501
2-170	Structure of Unitstatus Command Subpacket	2-502
2-171	Structure of Unlimit Command Subpacket	2-504
2-172	Structure of Unsilence Command Subpacket	2-505
2-173	Structure of User-Command Subpacket	2-506
2-174	Structure of Verbose Command Subpacket	2-508
2-175	Structure of ?PCLASS Packet	2-515
2-176	Structure of ?PIDS Packet	2-518
2-177	Structure of ?PMTPF Packet	2-520
2-178	Structure of ?PRDB/?PWRB Packet	2-526
2-179	Structure of ?PROC Packet	2-537
2-180	Structure of ?PROC Extension Packet for AOS/VS and AOS/RT32	2-546
2-181	Structure of ?PROC Extension Packet for AOS/VS II	2-546
2-182	Structure of ?PROFILE Parameter Packet	2-552
2-183	Structure of ?PROFILE Field Descriptor Packet	2-553
2-184	Structure of ?PSTAT Memory Descriptor	2-561
2-185	Structure of ?PSTAT Packet	2-562
2-186	Structure of the ?PTRDEVICE Main Packet	2-569
2-187	Tablet States	2-572
2-188	An Example of a Tablet Area Menu	2-573
2-189	Structure of the ?PTRDEV_SET_EVENTS Subpacket	2-573
2-190	Structure of the ?PTRDEV_SET_DELTA Subpacket	2-576
2-191	Structure of the ?PTRDEV_LAST_EVENT Subpacket	2-577
2-192	Subpacket for ?PTRDEV_SET_POINTER Subpacket	2-579
2-193	Structure of the ?PTRDEV_GET_PTR_STATUS Subpacket	2-581
2-194	Structure of the ?PTRDEV_GENERATE_EVENT Subpacket	2-585
2-195	Structure of the ?PTRDEV_GET_PTR_LOCATION Subpacket	2-587
2-196	Structure of the ?PTRDEV_GET_TABLET_LOCATION Subpacket	2-589
2-197	Structure of ?PWDCRYP Packet	2-591

?R Through ?Z

2-198	Structure of ?RDB/?WRB Packet	2-597
2-199	Structure of ?RDUDA/?WRUDA Packet	2-603
2-200	Structure of ?READ/?WRITE Packet	2-605
2-201	Structure of Screen Management Packet Extension	2-606
2-202	Structure of Selected Field Translation Packet Extension	2-606
2-203	Structure of New Screen Management Packet Extension	2-606
2-204	Structure of Screen-Management Packet Extension	2-612
2-205	Structure of Selected Field Translation Packet Extension	2-617
2-206	Selected Field Translation Packet Sample Listing	2-619
2-207	Structure of ?RENAME Packet	2-632
2-208	Structure of ?RNGPR Packet	2-641
2-209	Structure of ?RTODC Packet	2-646
2-210	Structure of ?RUNTM Packet	2-649
2-211	Structure of ?SACL Packet	2-652
2-212	Structure of ?SATR Packet	2-654

Figure**?R Through ?Z**

2-213	Polling List Defined by a Control Station	2-666
2-214	Poll and Select Addresses Defined by a Tributary	2-666
2-215	Structure of ?SEBL Packet	2-670
2-216	Station Identification System Call Sequence	2-673
2-217	Structure of the ?SERMSG Packet	2-684
2-218	Structure of ?SGES Packet	2-686
2-219	Structure of ?SINFO Packet	2-693
2-220	Structure of ?SONS Packet	2-697
2-221	Structure of ?SOPPF Packet	2-702
2-222	Structure of ?SPAGE Packet	2-705
2-223	Structure of ?SRCV Packet	2-710
2-224	ITB Receive Buffer Format	2-717
2-225	Structure of ?SSND Packet	2-721
2-226	Structure of ?SYLOG Exclusion Bit Map Packet	2-743
2-227	Structure of ?SYSPRV Packet	2-745
2-228	Structure of Standard Task Definition Packet	2-748
2-229	Stack Parameters for Initiating One or More Tasks	2-750
2-230	Extended Task Definition Packet	2-751
2-231	Structure of Termination Message a 32-bit Process Receives	2-761
2-232	Structure of Termination Message a 16-bit Process Receives	2-764
2-233	Structure of ?UIDSTAT Packet	2-777
2-234	Structure of ?VMEM Packet	2-788
2-235	Structure of ?VTFCREATE Packet	2-791
2-236	Execution of Two Virtual Timers	2-796
2-237	Structure of ?VTFKILL Packet	2-798
2-238	Structure of ?VTFMODIFY Packet	2-800
2-239	Structure of ?WHIST Packet	2-811
2-240	Sample Histogram Parameters	2-811
2-241	Structure of the ?WINDOW Main Packet	2-816
2-242	Structure of the ?WIN_CREATE_WINDOW Subpacket	2-818
2-243	Structure of the ?WIN_DEFINE_PORTS Subpacket	2-823
2-244	Structure of the ?WIN_SET_USER_INTERFACE Subpacket	2-828
2-245	Structure of the ?WIN_GET_USER_INTERFACE Subpacket	2-830
2-246	Structure of the ?WIN_GET_TITLE Subpacket	2-831
2-247	Structure of the ?WIN_GTITLE Subpacket	2-832
2-248	Structure of the ?WIN_WINDOW_STATUS Subpacket	2-834
2-249	Structure of the ?WIN_DEVICE_STATUS Subpacket	2-838
2-250	Structure of the ?WIN_RETURN_GROUP_WINDOWS Subpacket	2-840
2-251	Structure of the ?WIN_RETURN_DEVICE_WINDOWS Subpacket	2-841
2-252	Structure of ?XCREATE Main Packet	2-850
2-253	Structure of ?XCREATE Time-Block Subpacket	2-853
2-254	Structure of ?XCREATE Other Subpacket	2-854
2-255	Structure of ?XCREATE Directory Subpacket	2-855
2-256	Structure of ?XCREATE IPC Subpacket	2-856
2-257	Structure of ?XCREATE Link Subpacket	2-857
2-258	Example of ?XCREATE Main Packet for TXT File	2-858
2-259	Example of ?XCREATE Other Subpacket for TXT File	2-858
2-260	Example of ?XCREATE Main Packet for DIR File	2-859
2-261	Example of ?XCREATE Time Subpacket for DIR File	2-859

Figure**?R Through ?Z**

2-262	Example of ?XCREATE Directory Subpacket for DIR File	2-860
2-263	Example of ?XCREATE Main Packet for IPC File	2-860
2-264	Example of ?XCREATE IPC Subpacket for IPC File	2-861
2-265	Example of ?XCREATE Main Packet for LNK File	2-861
2-266	Example of ?XCREATE Link Subpacket for LNK File	2-861
2-267	Structure of ?XFSTAT IPC Packet	2-866
2-268	Structure of ?XFSTAT Directory Packet	2-870
2-269	Structure of ?XFSTAT Other Packet	2-873
2-270	Structure of ?XFSTAT Unit Packet	2-875
2-271	Example of Pathname Supplied	2-879
2-272	Example of Directory Type Grouping	2-881
2-273	Structure of ?XGTACP Packet	2-883
2-274	Structure of ?XINIT Packet	2-888
2-275	Structure of ?XINIT Subpacket	2-892
2-276	Structure of 16-Bit ?XINIT Packet	2-893
2-277	Structure of 16-Bit ?XINIT Subpacket	2-897
2-278	Structure of ?XPSTAT Packet	2-902
2-279	Structure of ?XPSTAT Standard Memory Descriptor	2-904
2-280	Structure of ?XPSTAT Extended Memory Descriptor (AOS/VS and AOS/VS II)	2-904
2-281	Structure of ?XPSTAT Extension Packet for AOS/VS and AOS/VS II	2-909
A-1	Listing of Program HEAR.SR	A-3
A-2	Listing of Program SPEAK.SR	A-7
A-3	Listing of Program SON.SR	A-9
A-4	Listing of Program RUNTIME.SR	A-11
A-5	Listing of Program RINGLOAD.SR	A-14
A-6	Listing of Program INRING.SR	A-16
A-7	Listing of Program GATE.ARRAY.SR	A-17
A-8	Listing of Program FILCREATE.SR	A-19
A-9	Listing of Program WRITE.SR	A-22
A-10	Listing of Program DLIST.SR	A-26
A-11	Listing of Program NEWTASK.SR	A-29
A-12	Listing of Program BOOMER.SR	A-32
A-13	Listing of Program TIMEOUT.SR	A-37
A-14	Listing of Program DIRCREATE.F77	A-41
A-15	File DIRCREATE_SYMBOLS	A-41
A-16	File DIRCREATE_SYMBOLS.F77.IN	A-42
A-17	Listing of Subroutine CHECK.F77	A-42
A-18	File CHECK_SYMBOLS	A-43
A-19	File CHECK_SYMBOLS.F77.IN	A-43
A-20	Listing of Program CREATE_WINDOW.SR	A-46
A-21	Code to Turn Permanence On in Program CREATE_WINDOW	A-55
A-22	Code to Turn Permanence Off in Program CREATE_WINDOW	A-55
A-23	Listing of Program GRAPHICS_SAMPLE.SR	A-57
A-24	Possible Results of Executing Program GRAPHICS_SAMPLE	A-76
B-1	Log Record Header	B-2
B-2	Log Record Codes, Events, and Message Lengths, Excluding Header	B-12
B-3	An Octal and Decimal DISPLAY of a System Log File	B-20
B-4	Octal and Decimal Versions of a SYSLOG Record	B-21

?RCHAIN

Chains to a new procedure (16-bit processes only).

?RCHAIN [*procedure entry*]

Input

Input accumulators and carry bit are passed to the new procedure

Output

Accumulators and carry bit are unchanged, unless the new procedure modifies them

Error Codes

The following codes may be passed to the ?BOMB routine:

ERICM	Illegal system command
ERLRF	Overlay load error
ERSEN	Invalid shared library reference (AOS/VS and AOS/RT32 do not support shared libraries.)

Why Use It?

?RCHAIN allows you to chain from a ?KCALLED or ?RCALLED procedure to another procedure. Only procedures that have been ?KCALLED or ?RCALLED can issue ?RCHAIN. Typically, you use ?RCHAIN to connect separate sequential pieces of a large resource.

Who Can Use It?

There are no special process privileges needed to issue this call, and there are no restrictions concerning file access.

What It Does

Like ?RCALL, ?RCHAIN releases the calling procedure, loads a new resource, and transfers control to the procedure entry that you specify within the new resource. You can pass the procedure entry as an argument to ?RCHAIN or pass it on the top of the stack, via the .PTARG pseudo-op.

Unlike ?RCALL, ?RCHAIN is issued from one called procedure to another and, therefore, has no normal return. Although each procedure must begin with an ?RSAVE instruction, only the last procedure on the chain should end with a RTN. When the operating system executes the RTN, it transfers control back to the initial procedure, not to the ?RCHAIN caller.

Notes

- See the descriptions of ?KCALL and ?RCHAIN in this chapter.

?RDB/?WRB

Performs (reads/writes) block I/O.

?RDB [*packet address*]

error return

normal return

?WRB [*packet address*]

error return

normal return

Input

AC0	Reserved (Set to 0.)
AC1	Target file's channel number
AC2	Address of the ?RDB/?WRB packet, unless you specify the address as an argument to the system call

Output

AC0	Undefined
AC1	Byte count of the bytes read or written, unless offset ?PRNH is -1 (See "Special Magnetic Tape Considerations" in this description.)
AC2	Address of the ?RDB/?WRB packet

Error Codes in AC0

ERDIO	Attempt to issue MCA direct I/O with outstanding requests
EREOF	End of file
ERILN	Illegal link number
ERIRV	Illegal retry value
ERPRO	Attempt to issue MCA request with direct I/O in progress
ERSPC	File space exhausted
ERVWP	Invalid word pointer passed as a system call argument
ER_FS_DIRECTORY_NOT_AVAILABLE	Directory not available because the LDU was force released (AOS/VS II only)
ER_FS_TLA_MODIFY_VIOLATION	Attempt to modify an AOS/VS II file with ?ODTL value supplied in ?GOPEN packet (?WRB only)

Why Use It?

?RDB and ?WRB allow you to perform block I/O on a file. (A system call named ?BLKIO also lets you perform block I/O on a file, plus it includes functionality for reading the next allocated element in a file.)

Who Can Use It?

There are no special process privileges needed to issue this call. You must have obtained a channel number via ?GOPEN or ?OPEN before issuing this call. Also, you have Read access to the file before issuing ?RDB or Write access to the file before issuing ?WRB.

What It Does

?RDB and ?WRB respectively read and write blocks of data on magnetic tape, disk, or MCA links. Note that you cannot use ?RDB to read blocks into the write-protected area of your logical address space. However, you can use ?WRB to write from the write-protected area.

Before you issue ?RDB or ?WRB, load AC1 with the channel number that the operating system assigned to the target file when you opened it (?GOPEN), and define the ?RDB or ?WRB packet in your logical address space. You can load the packet address into AC2 before you issue the block I/O system call, or you can specify the packet address as an argument to ?RDB or ?WRB. The value for the length of the packet is ?PBLT. Figure 2-198 shows the structure of the block I/O packet, and Table 2-179 describes the contents of each offset.

	0	7 8	15 16	23 24	31	
?PSTI	Flag word (see Table 2-179)		Block count	Reserved (Set to 0.)	Priority	?PSTO
?PCAD	Address of data buffer in your logical address space					
?PRNH	Block number of first disk block to be transferred, file number for tape, link number for MCA or retry count for MCA					?PRNL
?PRCL	Number of bytes in last disk block transferred (?WRB only), last tape block transferred, or in last MCA transmission		Reserved (Set to 0.)			?PRES
	?PBLT = packet length					

Figure 2-198. Structure of ?RDB/?WRB Packet

Before you can issue ?RDB or ?WRB against a file, you must open it with ?GOPEN.

?RDB/?WRB Continued

Table 2-179. Contents of ?RDB/?WRB Packet*

Offset	Contents
?PSTI	Left byte contains the following flags: ?IMIO--Suspend MCA protocol. ?ENOV--Override logical end-of-tape or enable vertical format unit (VFU). ?SAFM--Safe ?WRB request for magnetic tape. Right byte contains the block count.
?PSTO	Left byte is reserved. (Set to 0.) Right byte contains the priority.
?PCAD (doubleword)	Address of data buffer in your logical address space.
?PRNH	Number of first disk block transferred (high-order bits), file number for tape, or link number for MCA.
?PRNL	Low-order bits of disk file block number (?PRNH), block number or -1 for tape, or retry count for MCA.
?PRCL	Number of bytes in the last disk block transferred (for ?WRB only), number of bytes in the last tape block transferred, or the number of bytes in the last MCA transmission. Also, see the explanation of this offset in the section "Special Magnetic Tape Considerations."
?PRES	Reserved. (Set to 0.)

* There is no default unless otherwise specified.

Disk Blocks

Specify the number of blocks you want to read or write in the right byte of offset ?PSTI. Bit ?IMIO in the left byte of ?PSTI applies only to MCA transfers. Bit ?ENOV in the same offset applies only to magnetic tape transfers. For more information about these options, see "Special Magnetic Tape Considerations" and "Special MCA Considerations."

Offset ?PCAD in the packet should point to the data buffer you reserved in your logical address space for the block I/O transfer. If you read or write disk blocks, use offsets ?PRNH and ?PRNL to indicate the relative block number of the first disk block you want to transfer.

If you write disk blocks (?WRB), use offset ?PRCL to indicate the number of bytes in the last block you want to transfer. As in the physical block I/O packet, this value indicates the last valid byte in the block. If you issue ?WRB to extend the file, the operating system places the end-of-file mark immediately after this byte. For disk block transfers, offset ?PRCL applies only to ?WRB. If you set offset ?PRCL to 0, the operating system sets the bytes in the last block to the default, which is 512 (a full block).

Special Magnetic Tape Considerations

If you perform block I/O on magnetic tapes, you specify the file number at ?GOPEN time (for example, @MTB0:3) or in offset ?PRNH of the ?RDB/?WRB packet. If you used the CLI MOUNT

command to mount the tape, you must ?GOPEN the file using its linkname. When you specify the file number at ?GOPEN time, offset ?PRNH is ignored and offset ?PRNL contains the block number. If you do not specify a file number at ?GOPEN time, then offset ?PRNH contains the file number and ?PRNL the block number. For a 32-bit record block number use the ?B32N offset in ?BLKIO. Figure 2-1 shows the ?B32N offset on page 2-20.

When you ?GCLOSE the file, the operating system follows the last block written on a tape file with an end-of-file mark (two consecutive tape marks). By selecting the appropriate options in the block I/O packet, you can write beyond the end-of-tape mark and perform primitive tape functions, such as file positioning and rewinding.

To write beyond the end-of-tape mark, set offset ?PRNH (file number) to the highest file number plus one. Set the Block Count parameter (offset ?PSTI, right byte) to 0, and select bit ?ENOV (Bit 1) in offset ?PSTI. (Note that you must set ?ENOV to write beyond the end-of-tape mark.) Then, the operating system automatically follows the appended data with a new end-of-tape mark. To rewind a tape, set offsets ?PRNH, ?PRNL, and ?PSTI to 0.

When the operating system encounters an end-of-tape (EOT) mark while it is performing an ?RDB operation, it transfers the data it was able to read to the buffer designated by ?PCAD, and returns the byte count of that data to AC1. The byte count in AC1 is always a true value; that is, it is not rounded. The operating system then takes the error return from ?RDB, and returns error code EREOF to AC0.

Offset ?PRCL of the ?RDB/?WRB packet is the Block Length parameter for block I/O operations on magnetic tapes. If the operating system finds a block larger than this value during an ?RDB, it truncates the block to the length that you specify in ?PRCL.

The operating system always rounds the value of ?PRCL to an even number of bytes to force transfers to the buffer to be word aligned. The operating system discards the byte that exceeds the requested block size. If a block consists of an odd number of bytes on input, the operating system returns an even number of bytes on output. In this case, the last byte is undefined.

To position a tape at the end of a particular file, specify the file number in offset ?PRNH of the block I/O packet, and set offset ?PRNL (block number) to -1. To append data to the file, set the block count parameter (?PSTI, right byte) to a nonzero value. (You can append only on a ?WRB.) If you set the block count parameter to 0, the operating system returns the total number of blocks transferred to AC1.

Select the ?SAFM option in offset ?PSTI to write a logical end-of-tape (LEOT) or to write data followed by LEOT. By selecting this option, you ensure that the tape is always readable, even if you do not issue ?GCLOSE. This means that if your system fails, your tape will still be readable.

Special MCA Considerations

MCA processes in your system must be resident. You begin block I/O operations on MCAs by issuing ?GOPEN to open the MCA as a unit or to open a specific link. To open the MCA as a unit, load AC0 with a byte pointer to one of the following filenames before you issue ?GOPEN:

@MCAT
@MCAR
@MCAT1
@MCAR1

where

@MCAT is an MCA to use for writing (transmitting).

@MCAR is an MCA to use for reading (receiving).

1 is the second MCAT or MCAR device.

?RDB/?WRB Continued

If you open the MCA as a unit, you must name a specific MCA link in offset ?PRNH of the block I/O packet for the ?RDB or ?WRB operation.

To open a specific MCA link, load AC0 with one of the following pathnames before you issue ?GOPEN:

@MCAT:n
@MCAR:n
@MCAT1:n
@MCAR1:n

where

n is the link number (range 0 through 15).

If you open a specific MCA link, then all subsequent ?RDB or ?WRB system calls will be to that link. However, if you open the MCA as a unit, you can change the link number for each ?RDB or ?WRB system call by altering the value of ?PRNH in the ?RDB/?WRB packet. A link number of 0 applies to ?RDB system calls only, and directs the operating system to accept transmissions from any MCAT. The operating system satisfies read requests from link 0 after it performs all other read requests from specific link numbers.

When you close an MCA file previously opened for a specific link (that is, you specified the link number with ?GOPEN), the operating system transmits an end-of-file (EREOF) condition with a word count of 0 to the MCA receiver (after a 2-millisecond time-out). Do not use the link format unless the remote MCA is prepared for this transmission protocol.

The retry count, offset ?PRNL in the block I/O packet, applies only to ?WRB operations from an MCAT. You can set this parameter to any value within the range 0 through 255. Each count represents 20 attempts to establish communications with the target MCAR. A retry value of 0 directs the operating system to perform the maximum number of retries.

You can circumvent the MCA protocol and perform direct MCA I/O by selecting bit ?IMIO in offset ?PSTI. This method results in fewer interrupts. To use the direct mode, however, you must perform your own data and line validations. In addition, the operating system does not honor direct MCA transmission requests unless the MCAT's output queue is empty. If you try to transmit when the queue is not empty, the block I/O system call fails and the operating system returns error code ERPRO to AC0.

Similarly, once the operating system has queued a direct write request to an MCAT link, it does not queue any further direct I/O or protocol I/O requests until it completes the current request. If you try to perform a read or write on an MCA link while direct I/O is in progress, the system call fails, and the operating system returns error code ERDIO in AC0.

Data Channel Line Printers

If you perform block writes to a data channel line-printer controller with a DVFU unit, set bit ?ENOV in offset ?PSTI of the ?WRB packet. The DVFU (direct-access vertical forms control unit) is a hardware device that interfaces with the operating system's Forms Control Utility (FCU) program. By issuing VFU commands through the FCU, you can control the printer's horizontal and vertical formatting. A memory capability within the DVFU eliminates the need for mechanical formatting with a VFU tape.

If you omit ?ENOV, the operating system ignores all VFU definition commands.

For a complete description of the FCU utility, refer to your operating system's Command Line Interpreter (CLI) user's manual.

Sample Packet

```
PKT:  .BLK  ?PBLT          ;Allocate enough space for the packet.
      .LOC  PKT+?PSTI      ;Packet length = ?PBLT.
      .WORD 4              ;Number of blocks to read or write.
      .LOC  PKT+?PSTO      ;Transfer four blocks.
      .WORD 4              ;Reserved.
      .LOC  PKT+?PCAD      ;You must set this value to 0.
      .DWORD 0             ;Word address of data buffer.
      .LOC  PKT+?PRNH      ;Data buffer address is BUFF.
      .DWORD 0             ;Block number.
      .LOC  PKT+?PRCL      ;Actual number of bytes transferred.
      .WORD 0              ;The OS returns this value.
      .LOC  PKT+?PRES      ;Reserved.
      .WORD 0              ;You must set this value to 0.
      .LOC  PKT+?PBLT      ;End of packet.
BUFF: .BLK  4*256         ;Text buffer area.
```

Notes

- See the description of ?GOPEN and ?BLKIO in this chapter.

?RDUDA/?WRUDA

Reads/writes a user data area (UDA).

?RDUDA [*packet address*]

error return

normal return

?WRUDA [*packet address*]

error return

normal return

Input

- AC0 One of the following:
- Byte pointer to the pathname of the file associated with the UDA
 - 0 if a packet address is supplied
- AC1 Reserved (Set to 0.)
- AC2 One of the following:
- Address of a 128-word UDA receive buffer if not using a packet
 - Address of the ?RDUDA/?WRUDA packet, unless you specify the address as an argument to ?RDUDA/?WRUDA

Output

- AC0 Unchanged
- AC1 Undefined
- AC2 Unchanged or address of ?RDUDA/?WRUDA packet

Error Codes in AC0

- ERFAD File access denied
- ERIFT Illegal file type
- ERNUD User data area does not exist
- ERVBP Invalid byte pointer passed as a system call argument
- ERVWP Invalid word pointer passed as a system call argument
- ER_FS_DIRECTORY_NOT_AVAILABLE
Directory not available because the LDU was force released (AOS/VS II only)
- ER_FS_TLA_MODIFY_VIOLATION
Attempt to modify an AOS/VS II file with ?ODTL value supplied in ?GOPEN packet (?WRUDA only)

Why Use It?

?WRUDA allows you to write to a file's UDA. The UDA frequently stores forms control parameters for files you intend to print on a data channel line printer that is controlled by the EXEC utility.

(You can use a UDA for other purposes as well.) Because ?RDUDA returns the contents of a file's UDA, you can issue it to check the UDA contents before you issue ?WRUDA.

Who Can Use It?

There are no special process privileges needed to issue this call. To gain access to the file when you supply a pathname, you need Execute access to the file's parent directory and one or more of the following three access rights:

- Owner access to the file.
- Read access to the file (?RDUDA).
- Write access to the file (?WRUDA).

To gain access to the file when you supply a channel number (i.e., a packet address), you don't need Execute access to the file's parent directory — but you still need one or more of the previous three access rights.

What It Does

?WRUDA writes to the target file's user data area (UDA). ?RDUDA reads the contents of the target file's UDA. The target file or directory can be specified in one of two ways: either by a byte pointer to the entry's pathname in AC0, or by using offset ?PUDCN in the ?RDUDA/?WRUDA packet. The use of the packet is only necessary if you choose to specify the target entry's channel number. AC0 must be set to 0 to indicate the use of a packet.

If not using a packet, load AC2 with a word pointer to a 128-word receive/send buffer from/for the UDA. If using a packet, offsets ?PUDAH and ?PUDAL must instead be used. The UDA must already exist. (You can use the ?CRUDA system call or the CLI Forms Control Utility to create a UDA.)

To find out whether a file has a UDA, issue ?FSTAT, and then examine bit ?FUDA in offset ?SSTS of the returned packet. If ?FUDA is set, the UDA exists.

When you write to a UDA, do not use codes that will set Bit 0 of the first UDA word (word 0). Data General reserves codes with this configuration for its own use.

Figure 2-199 shows the structure of the ?RDUDA/?WRUDA packet.

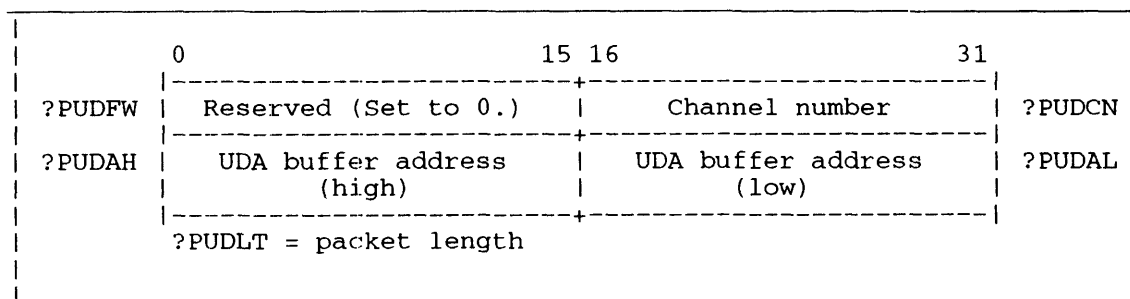


Figure 2-199. Structure of ?RDUDA/?WRUDA Packet

Notes

- See the descriptions of ?CRUDA and ?FSTAT in this chapter.

?READ/?WRITE

Performs (reads/writes) record I/O.

?READ [*packet address*]

error return

normal return

?WRITE [*packet address*]

error return

normal return

Input

AC0	Reserved (Set to 0.)
AC1	Reserved (Set to 0.)
AC2	Address of the I/O packet, unless you specify the address as an argument to the system call

Output

AC0	Undefined
AC1	Undefined
AC2	Address of the I/O packet

Error Codes in AC0

EREOF	End of file
ERFIL	File read error
ERIFL	IAC (Intelligent Asynchronous Controller) failure
ERISO	Illegal screen edit request (from PMGR)
ERITP	Translation error (Only for the selected field translation packet extension)
ERLTL	Line too long
ERSPC	File space exhausted
ERPUF	Physical unit failure. If the device is a terminal, an IAC (Intelligent Asynchronous Controller) failed.
ERRAD	Read access denied (both ?READ and ?WRITE)
ERVBP	Invalid byte pointer passed as a system call argument
ERVWP	Invalid word pointer passed as a system call argument
ERWAD	Write access denied (?WRITE only)
ER_FS_DIRECTORY_NOT_AVAILABLE	Directory not available because the LDU was force released (AOS/VS II only)

Who Can Use It?

There are no special process privileges needed to issue this call.

For ?READ: you must have obtained a channel number to the file via ?OPEN before issuing ?READ. Also, you must have had Read access to the file at the time of the ?OPEN.

For ?WRITE: you must have obtained a channel number to the file via ?OPEN before issuing ?WRITE. Also, you must have had Write access to the file at the time of the ?OPEN.

What It Does

?READ and ?WRITE perform record I/O on an open file.

As Figure 2-200 shows, the basic ?READ/?WRITE packet has the same structure as the ?OPEN and ?CLOSE packets. Some specifications apply exclusively to ?READ and ?WRITE, while others apply to all record I/O system calls. When you issue ?READ or ?WRITE against a file, you can default some specifications to the values that you selected when you created or opened the file.

CAUTION: *If you are using a pixel-mapped system, do not attempt to use device code 10 (TTI or system console input, such as a keyboard) or device code 11 (TTO or system console output, such as a screen or printer), once your operating system is running.*

Table 2-184. Contents of the New Screen Management Status Word Packet.

Offset	Contents
?SCMGT.STATUS (double word)	This status word's bits return information to the caller. Bit positions are numbered. Unused bits are reserved and should be ignored. Some bit positions are reserved to provide easy conversion between the old and new packets. ?SCMGT.STATUS.ESDD = 5. The system sets this flag, if the read ended with a function key delimiter. Same as ?ESDD. ?SCMGT.STATUS.ESBE = 9. The system sets this flag, if a get type-ahead read terminated from an empty input buffer, rather than a delimiter or maximum number of characters. Same as ?ESBE. ?SCMGT.STATUS.ESPE = 10. The system sets this flag, if the read ended with a mouse event delimiter. Same as ?ESPE. ?SCMGT.STATUS.ESII = 11. The system sets this flag, if the read ended in Insert Mode. ?SCMGT.STATUS.ESMI = 12. The system sets this flag if any characters were entered into the user's buffer during a Screen Edit read.
?SCMGT.ESEP_IN (word)	At the start of the read, this offset contains the desired relative cursor position within the Predisplay buffer, as a byte offset from the beginning of the read buffer. Ignored unless both ESSE and ESRD flags are set.
?SCMGT.ESEP_OUT (word)	This offset returns the relative cursor position as a byte offset from the beginning of the read buffer. Ignored unless both ESSE and ESCO flags are set.
?SCMGT.ESCR_IN.COL (word)	This offset contains the initial cursor column position. Ignored unless ESCP flag is set.
?SCMGT.ESCR_IN.ROW (word)	This offset contains the initial cursor row position. Ignored unless ESCP flag is set.
?SCMGT.ESCR_OUT.COL (word)	This offset contains the initial cursor column position. Ignored unless ESRP flag is set.
?SCMGT.ESCR_OUT.ROW (word)	This offset contains the initial cursor row position. Ignored unless ESRP flag is set.

?READ/?WRITE Continued

Dual Asynchronous Receiver–Transmitters (DRTs) and Intelligent Asynchronous Controllers (IACs, supporting Data Channel or Local Bus protocols) control screen management on the MV/Family host and execute on the host or on the IAC controller.

In AOS/VS, several New Screen Management packet features are unsupported on the host or on the IAC controller. Table 2–185 lists supported (Y for yes) and unsupported (N for no) new Screen–Management features for either the host or the IAC controller implementations. All of these new Screen–Management features are fully supported in AOS/VS II.

Table 2–185. Supported (Y) and Unsupported (N) New Screen Management Packet Features in AOS/VS

Feature and offset	Locus of Execution	
	On Host	On IAC
Left autotermination ?SCMGT.FLAGS.ESLT=16	Y	Y
Normal carriage return ?SCMGT.FLAGS.ESNC=17	Y	Y
Return entire buffer on timeout ?SCMGT.FLAGS.ESRT=15	N	Y ¹
Null as a space ?SCMGT.FLAGS.ESNP	Y	Y
Read ended in insert mode ?SCMGT.STATUS.ESII=11	Y	Y
Read buffer modified ?SCMGT.STATUS.ESMI	Y	Y
Relative cursor column returned ?CMGT.ESEP_OUT	N	Y ¹

¹ 68000-based IACs support this feature; IACs prior to 68000-based ones do not support this feature.

Notes

- See the descriptions of ?GECHR, ?OPEN, and ?CLOSE in this chapter.

?RECREATE

error return

normal return

Input

AC0 Byte pointer to the target
file's pathname

AC1 Reserved (Set to 0.)

AC2 Reserved (Set to 0.)

Output

AC0 Unchanged

AC1 Undefined

AC2 Undefined

Error Codes in AC0

ERDID Directory delete error

ERDIU Directory in use

ERFAD File access denied

ERFDE File does not exist

ERMPR System call parameter address error

ERPRM Cannot delete permanent file

ERVBP Invalid byte pointer passed as a system call argument

Why Use It?

?RECREATE combines the functions of ?DELETE and ?CREATE in that it allows you to delete all user data in a file, but maintain its characteristics. ?RECREATE is useful for releasing a file's disk blocks.

Who Can Use It?

There are no special process privileges needed to issue this call. If you don't have Owner access to the file, you must have Execute, Write, and Read access to the file's parent directory. If you do have Owner access to the file, you must have Execute access to the file's parent directory and either Write or Append access to the file's parent directory.

What It Does

?RECREATE deletes the target file or directory, and creates it again with the same ACL, filename, file type, creation date, and user data area (UDA) information, but without data. In short, ?RECREATE gives you a file with the same characteristics as your original file, but with a length of 0.

If the target file is open when you issue ?RECREATE, the operating system maintains all the data until you issue the last ?CLOSE. However, the data is visible only to the processes that opened the file before you issued ?RECREATE.

You cannot issue ?RECREATE against a file or directory with the PERMANENCE attribute or a directory that is currently in use (that is, your working directory or one cited in a search list).

Notes

- See the descriptions of ?CLOSE, ?CREATE, and ?DELETE in this chapter.

?RELEASE

Releases an initialized logical disk.

?RELEASE

error return

normal return

Input

AC0 Byte pointer to the
pathname of the logical
disk unit (LDU)

AC1 One of the following:

- Reserved in AOS/VS (Set to 0.)
- ?RLFRC, in AOS/VS II,
to force the release of the LDU
named by the AC0 byte pointer.

AC2 Reserved (Set to 0.)

Output

AC0 Unchanged

AC1 Undefined

AC2 Undefined

Error Codes in AC0

ERIFT Illegal file type

ERVBP Invalid byte pointer passed as a system call argument

ERVIU LDU in use, cannot release (AOS/VS only)

ERWAD Write access denied

ER_FS_CANNOT_RELEASE_RESTRICTED_LDU

Logical disks containing the ROOT, PAGE, or SWAP directories cannot be force released
(AOS/VS II only)

ER_FS_DIRECTORY_NOT_AVAILABLE

Directory not available, because the LDU was force released (AOS/VS II only)

Why Use It?

Once a logical disk has been initialized, it remains initialized until you release it. Therefore, you must issue ?RELEASE to reuse the disk drives on which an LDU's component volumes are mounted. In AOS/VS you cannot release an LDU if it is in use.

In AOS/VS II, if you use the ?RLFC option, you can release an LDU, even if it is in use. Therefore choose this feature when you need to release an LDU, but cannot identify who has files open in the LDU.

Who Can Use It?

There are no special process privileges needed to issue this call. You must have Execute access to the logical disk's parent directory and Owner access to the logical disk itself.

What It Does

?RELEASE releases an LDU that you previously initialized with ?INIT. In AOS/VS, you cannot release an LD that is currently in use.

In AOS/VS II, you can force-release an LDU by setting the option flag ?RLFRC in AC1. The forced release of an LDU is similar to an emergency shutdown release of an LDU. Any LDU you specify in AC0 is released, except one containing the ROOT, PAGE, or SWAP directories. If you try to release any LDU containing the ROOT, PAGE, or SWAP directories, the operating system will return the error ER_FS_CANNOT_RELEASE_RESTRICTED_LDU. The operating system releases an LDU regardless of the number of open files, and ?RELEASE then logs errors about any attempt to remove a force-released LDU.

Active mirror synchronizations are broken during the LDU forced release. All open files in the target LDU are forced to close, and file attributes are updated to the disk. But any user's channel remains open until closed by the process that opened the channel.

CAUTION: If you force release an LDU, any open file, shared access file, or other user-buffered data from the file system on the LDU will not be written to disk or updated. Such files could be corrupted.

Subsequent requests to force-close files will return the error ER_FS_DIR_NOT_AVAILABLE until the channel is closed, or until the process that opened the channel is terminated.

?RELEASE releases any LDUs subordinate to the force-released LDU. Any subsequent request to use any force-released LDU is aborted and reported to the operator console. Files are accessible again after the force-released LDU is reinitialized.

Notes

- See the description of ?INIT in this chapter.

?RENAME

Renames a file or pathname.

?RENAME [*packet address*]

error return

normal return

Input

Output

AC0	One of the following: <ul style="list-style-type: none">• Byte pointer to the file's original filename• 0 if a packet address is supplied	AC0	Unchanged
AC1	Byte pointer to the new filename (AOS/VS and AOS/VS II) or new pathname (AOS/VS II only)	AC1	Unchanged
AC2	One of the following: <ul style="list-style-type: none">• Reserved (Set to 0 if not supplying a packet)• Address of the ?RENAME packet, unless you specify the address as an argument to ?RENAME	AC2	Undefined or address of ?RENAME packet if not supplying a packet

Error Codes in AC0

ERIFC	Illegal filename character
ERIFT	Illegal file type
ERCRR	Cannot rename the root
ERCPD	Control point directory maximum size exceeded
ERVBP	Invalid byte pointer passed as a system call argument
ERWAD	Write access denied
ER_FS_CANNOT_CROSS_LDU_BOUNDARY	Cannot rename across a logical disk unit (LDU) boundary (AOS/VS II only)
ER_FS_CANNOT_RENAME_UNDER_HIERARCHY	Cannot rename a directory within its own directory hierarchy (AOS/VS II only)
ER_FS_DIRECTORY_NOT_AVAILABLE	Directory not available because the LDU was force released (AOS/VS II only)
ER_FS_WRITE_OR_APPEND_ACCESS_REQ	Write or append access is required (AOS/VS II only)

Why Use It?

You can use ?RENAME to change the name of a file or LDU. You can also use ?RENAME to move a file or directory to a different location within the same logical disk unit's directory hierarchy. ?RENAME is much faster than moving the file because the move copies all the data blocks.

Who Can Use It?

There are no special process privileges needed to issue this call. If you specified the file with a channel number, you must have Owner access to the target file or Write access to the target file's parent directory. If, on the other hand, you specified the file with a pathname, you must also have Execute access to the parent directory.

What It Does

In AOS/VS, and in releases prior to AOS/VS II 2.20, ?RENAME deletes the target file's current pathname (a filename including a ":", "@", "=", or a "^") or name and gives the file the new name you specify in AC1.

In AOS/VS II release 2.20 only, ?RENAME deletes the target file's current pathname (a filename including a ":", "@", "=", or a "^") or name and gives the file the new *pathname* or name you specify in AC1.

Specifying a pathname renames a file across directories, and the following restrictions apply. When you use ?RENAME across directories, the parent directory of the pathname or filename in AC1 must be within the same LDU as the target file, and the new pathname or filename cannot exist in the target directory hierarchy, or you will produce the ERNAE error.

When you use ?RENAME across directories, the calling process must have either write or append access to the target file's new parent directory (the parent directory of the new name specified in AC1).

The target filename can be specified in one of two ways: either by a byte pointer to the original pathname in AC0, or by using offset ?GCPCN in the ?RENAME packet. Only use the packet if you need to specify the target file's channel number. If you use a channel number, you must set AC0 to zero to indicate the use of a channel number. If a byte pointer is specified in AC0, then the current (original) name may be a pathname; the new name can either be a filename or a full pathname depending on your software release.

If you rename an LDU, the new name becomes invalid if you use ?RELEASE to release the LDU, and you then reinitialize it. The LDU resumes its original name.

You can only rename a restricted set of files types if you are renaming across directories. You cannot use ?RENAME to rename the following file types across directories:

Symbol	Restricted File Type
?FLDU	LDU files.
?LIPC through ?HIPC	All console and IPC files.
?LUNT through ?HUNT	All unit and device files.
?FGFN	Generic files, for example @DATA, @LIST, @NULL must reside in :PER.
?FMTF	Magnetic tape files.
?FREM	Remote host – Remote Access (RMA) files.
?FHST	Remote host – X.25 access files.
?FNPN	Network process name files.

If you rename a restricted file, the ERIFT error results.

The rename-across-directories feature requires that the new parent directory (and all directories up to the logical disk root directory) has enough space remaining to accommodate the target file and any subordinate directory hierarchy. Space is limited by the space remaining in any control point directory.

?RENAME Continued

- To use ?RENAME to rename files, the calling process must have Write access to the target file's parent directory, or Owner access to the target file.

Figure 2-207 shows the structure of the ?RENAME packet.

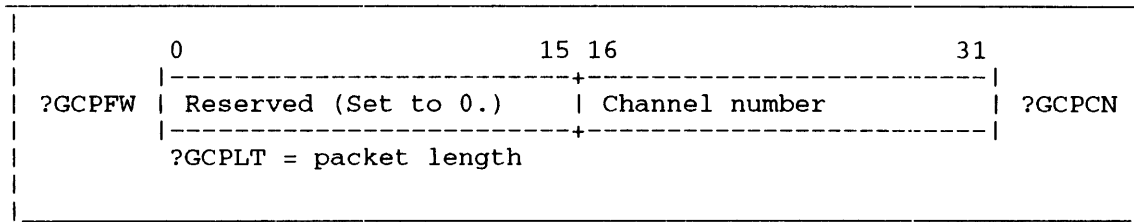


Figure 2-207. Structure of ?RENAME Packet

?RPAGE

Releases a shared page.

?RPAGE

error return

normal return

Input

AC0 Contains the following:

- Bit 0 is a flag bit:

If Bit 0 = 1, flush the contents of the page to disk before releasing it

If Bit 0 = 0, do not flush the contents of the page to disk before releasing it
- Bits 1 through 31 contain any address in the target page (The OS converts this value to the correct logical page number.)

AC1 Reserved (Set to 0.)

AC2 Reserved (Set to 0.)

Output

AC0 Unchanged

AC1 Undefined

AC2 Undefined

Error Codes in AC0

ERNSA Shared I/O request not to shared area

ER_FS_DIRECTORY_NOT_AVAILABLE

Directory not available because the LDU was force released (AOS/VS II only)

Why Use It?

?RPAGE removes a shared page from your logical address space, but does not necessarily remove it from memory. This eliminates the need for disk I/O, if another process requires the same page.

When you set the flag bit (Bit 0) in AC0, the ?RPAGE simulates a ?FLUSH; that is, the operating system retains the page in memory, but flushes its contents to disk if it is modified. This option guarantees that the shared page is updated immediately.

?RPAGE Continued

Who Can Use It?

There are no special process privileges needed to issue this call. The restrictions concerning file access are the same as those for system call ?SOPEN.

What It Does

?RPAGE releases a shared memory page from the caller's logical address space. If no other process is using the target page after the ?RPAGE, the operating system places it on the LRU chain for eventual reuse. (Note that the page remains in the user context; ?RPAGE invalidates the address for the user.)

If you set Bit 0 of AC0 and the page has been modified, the operating system flushes the contents of the page to disk. This variation is functionally the same as issuing ?FLUSH followed immediately by a standard ?RPAGE.

?RUNTM Continued

You must also reserve a packet of ?GRLTH words to receive the ?RUNTM statistics. You can cite the packet address as an argument to ?RUNTM or you can load its address into AC2 before you issue ?RUNTM. Figure 2-210 shows the packet structure.

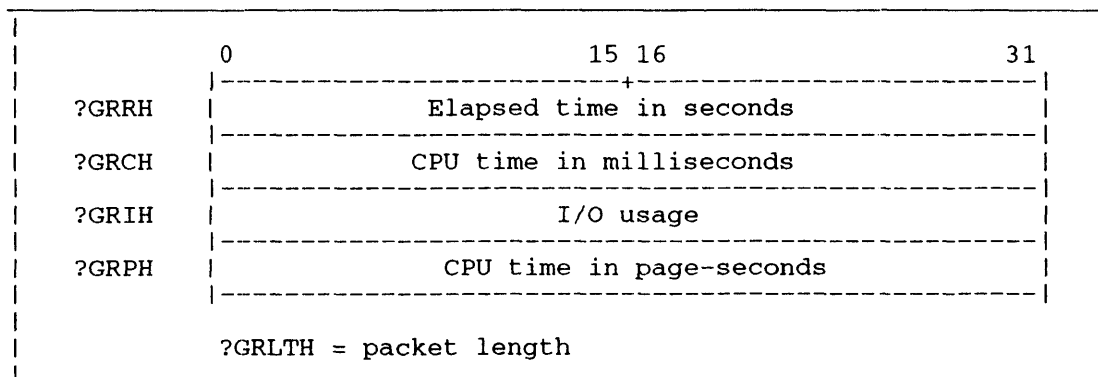


Figure 2-210. Structure of ?RUNTM Packet

As Figure 2-210 shows, the operating system returns the following information to the ?RUNTM packet: the elapsed time of the target process, its CPU time, its page usage over CPU time, and its I/O usage (number of blocks read or written).

Sample Packet

```
PKT:    .BLK    ?GRLTH          ;Allocate enough space for packet.
                                     ;Packet length = ?GRLTH.

        .LOC    PKT+?GRRH      ;The OS will return elapsed time
        .DWORD  0              ;(in seconds).

        .LOC    PKT+?GRCH      ;The OS will return CPU time (in
        .DWORD  0              ;milliseconds).

        .LOC    PKT+?GRIH      ;The OS will return I/O usage
        .DWORD  0              ;information.

        .LOC    PKT+?GRPH      ;The OS will return page usage
        .DWORD  0              ;information.

        .LOC    PKT+?GRLTH     ;End of packet.
```

Notes

- See the descriptions of ?PSTAT and ?WHIST in this chapter.

?SACL

Sets a new access control list (ACL).

?SACL [*packet address*]

error return

normal return

Operating System Differences

Under AOS/RT32, ?SACL does nothing.

Input

- AC0 One of the following:
- Byte pointer to the pathname of the target file or directory
 - 0 if a packet address is supplied

- AC1 One of the following:
- Byte pointer to the new ACL
 - -1 to delete the current ACL

- AC2 One of the following:
- Reserved (Set to 0 if not supplying a packet.)
 - Address of the ?SACL packet, unless you specify the address as an argument to ?SACL

Output

AC0 Unchanged

AC1 Undefined

AC2 Undefined or address of ?SACL packet

Error Codes in AC0

ERACL Illegal ACL

ERVBP Invalid byte pointer passed as a system call argument

ERWAD Write access denied

ER_FS_DIRECTORY_NOT_AVAILABLE

Directory not available because the LDU was force released (AOS/VS II only)

ER_FS_TLA_MODIFY_VIOLATION

Attempt to modify file with ?ODTL value supplied in ?GOPEN packet

Why Use It?

?SACL allows you to alter or delete a file's current access control list (ACL). You can use ?SACL to change the ACL you set when you created the file.

The operating system reverts to the default access privileges when you issue ?DELETE, and then ?CREATE, against a file.

To find out what the ACL is for a particular file or directory before you issue ?SACL, issue ?GACL.

Who Can Use It?

There are no special process privileges needed to issue this call. If you specified the file with a channel number, you must have Owner access to the target file or Write access to the target file's parent directory. If, on the other hand, you specified the file with a pathname, you must also have Execute access to the parent directory.

What It Does

Under AOS/RT32, ?SACL does nothing. Under AOS/VS, ?SACL replaces the target file's or directory's ACL with the ACL that you specify in AC1. If there is no new ACL, ?SACL deletes the existing ACL.

You can specify the target file or directory in one of two ways: either by a byte pointer to the file or directory's pathname in AC0, or by using offset ?GCPCN in the ?SACL packet. The use of the packet is only necessary if you choose to specify the target file or directory's channel number. AC0 must be set to zero to indicate the use of a packet.

Before you issue ?SACL, define the new ACL in your address space and load AC1 with a byte pointer to its address. Structure the ACL specification as follows:

```
username<0><access privs>[username<0><access privs>...]<0>
```

For an example of 9 bytes:

```
OP<0><?FACO+?FACW+?FACA+?FACR+?FACE>CW<0><?FACR+?FACE><0>
```

where

access privs is one or more of the following access types:

```
?FACO = Owner access
?FACW = Write access
?FACA = Append access
?FACR = Read access
?FACE = Execute access
```

The brackets ([]) in the format mean that you can repeat the

```
username<0><access privs>
```

entry. However, be sure to type the ACL specifications on one line. If you use a carriage return or New Line character, the operating system interprets that character as part of the ACL. Do not insert any spaces between the ACL mnemonics, and do terminate the ACL specification with a null byte.

To give a series of access privileges to all users, use the +<0> template before the ACL specification. For example, the specification

```
+<0><?FACR+?FACE><0>
```

gives all users Read and Execute access to the file. The symbol ?MXACL represents the maximum length for an ACL specification.

?SACL Continued

For another example, the CLI command

```
ACL FOO SAM,OAR LYNN,RE
```

sets the ACL of file FOO. You can also do this with the ?SACL system call. One step is to create a buffer containing the following 12 bytes in its leftmost bytes.

```
SAM<0><?FACO+?FACA+?FACR>LYNN<0><?FACR+?FACE><0>
```

Each of the symbols

<?FACO+?FACA+?FACR> and <?FACR+?FACE>

requires 1 byte.

Figure 2-211 shows the structure of the ?SACL packet.

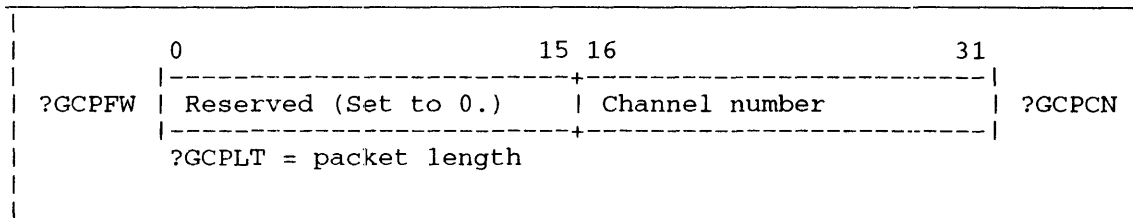


Figure 2-211. Structure of ?SACL Packet

Notes

- See the descriptions of ?CREATE, ?DELETE, and ?GACL in this chapter.

?SATR

Sets or removes the permanent attribute for a file or directory.

?SATR [*packet address*]

error return

normal return

Input

AC0 One of the following:

- Byte pointer to the pathname of the target file or directory
- 0 if a packet address is supplied

AC1 Bit 31 is a flag bit:

- Bit 31 = 1 to set the permanent attribute
- Bit 31 = 0 to remove the PERMANENCE attribute

AC2 One of the following:

- Reserved (Set to 0 if not supplying a packet.)
- Address of the ?SATR packet, unless you specify the address as an argument to ?SATR

Output

AC0 Unchanged

AC1 Unchanged

AC2 Undefined or address of ?SATR packet

Error Codes in AC0

ERIFT Illegal file type

ERVBP Invalid byte pointer passed as a system call argument

ERWAD Write access denied

ER_FS_DIRECTORY_NOT_AVAILABLE

Directory not available because the LDU was force released (AOS/VS II only)

Why Use It?

You can protect a file from being deleted by any process, regardless of the process's access privileges, by assigning that file the permanent attribute. If you assign the PERMANENCE attribute to a file, you should assign that attribute to its parent directory as well. Otherwise, a process can delete the file indirectly, simply by deleting the parent directory.

To find out if a file has the PERMANENCE attribute, issue ?FSTAT and examine the ?SSTS offset in the ?FSTAT return packet.

Who Can Use It?

There are no special process privileges needed to issue this call. You must have Execute access to the file's parent directory and either Owner access to the file or Write access to the file's parent directory.

What It Does

?SATR either assigns or removes the PERMANENCE attribute for a file or directory, depending on your input to AC1. You cannot delete any entries that have the PERMANENCE attribute. (The PERMANENCE attribute is the only file attribute that is currently defined.)

The target file can be specified in one of two ways: either by a byte pointer to the entry's pathname in AC0, or by using offset ?GCPCN in the ?SATR packet. The use of the packet is only necessary if you choose to specify the target entry's channel number. AC0 must be set to zero to indicate the use of a packet.

Figure 2-212 shows the structure of the ?SATR packet.

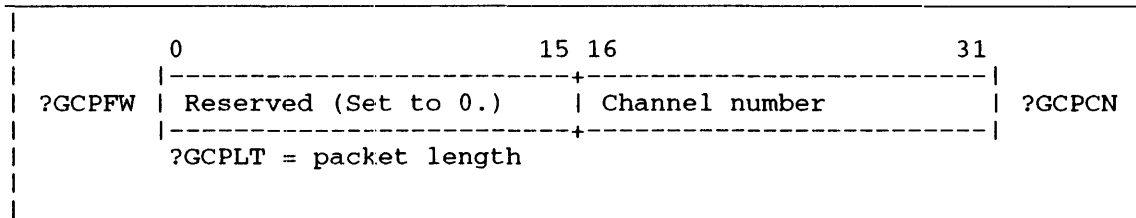


Figure 2-212. Structure of ?SATR Packet

Notes

- See the description of ?FSTAT in this chapter.

offset ?SIRS must be set equal to ?PKR1, and the byte pointer to the buffer to hold the name must be included. You must reserve an area in your logical address space ?SIPL words long for the ?SINFO packet if the pathname of the executing operating system is not wanted, or else ?SIEX words long if the name is wanted.

Figure 2-219 shows the structure of the ?SINFO packet.

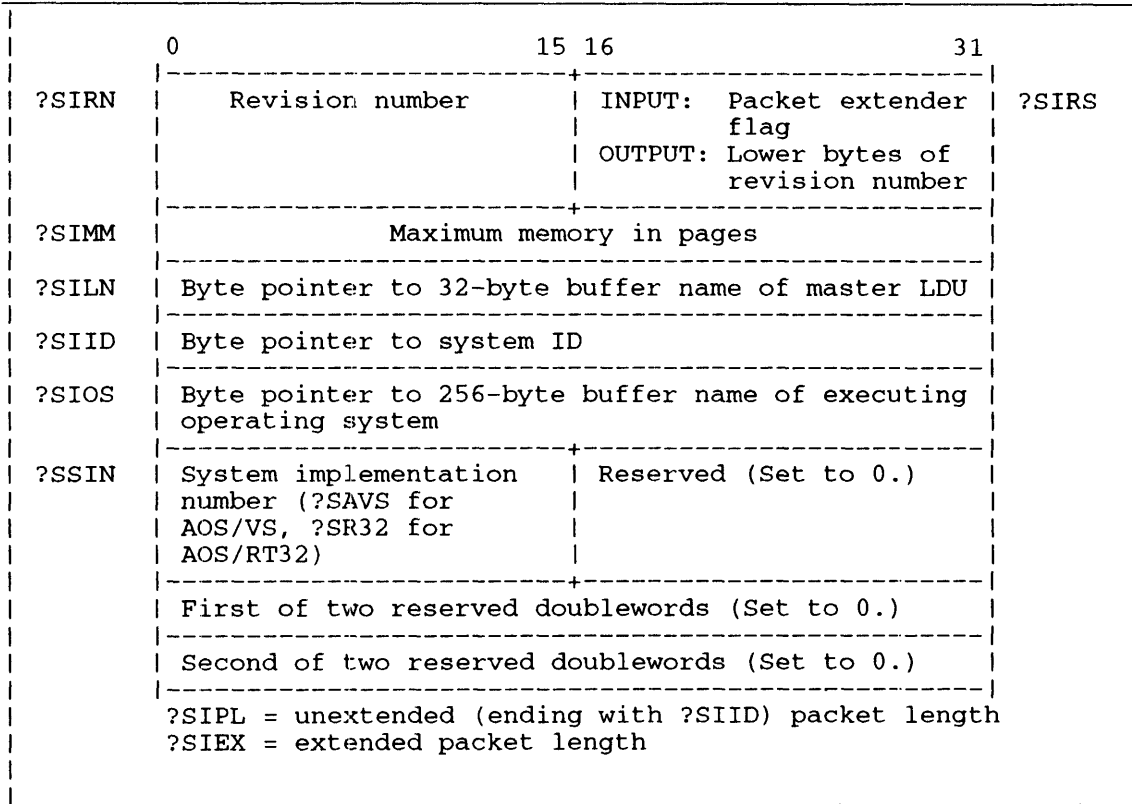


Figure 2-219. Structure of ?SINFO Packet

The operating system returns its major and minor revision numbers as octal values to the left and right byte of offset ?SIRN respectively. If the revision number is 00.00.01.01, for example, ?SIRN contains 00101. The operating system returns 2100 in offset ?SIRN for AOS/VS II Release 1.00.

Offset ?SIMM contains the number of the highest memory page. In a 256-Mbyte system with contiguous memory, for example, the value of ?SIMM is octal 400000; that is decimal 131072. The first memory page is page zero, and the last is the highest memory page known to the operating system. On some systems, not all memory is contiguous and available to the operating system. Thus memory pages may not be contiguous up to the page value returned in offset ?SIMM. Also, on some systems part of system memory is dedicated to firmware scratch space.

If you want the operating system to return the name of the master LDU you are using and the system identifier, set up two 32-byte receiving areas in your logical address space. Use offset ?SILN as a byte pointer to the buffer area for the master LDU name, and use offset ?SIID as a byte pointer to the buffer for the system identifier. If you do not want the name of the master LDU or the name of the system identifier, place 0 in the appropriate offset(s) and omit the corresponding buffer area(s).

?SINFO Continued

If you want the operating system to return the pathname of the executing operating system, then make an extended ?SINFO call. The ?SIRS word in the packet is used to indicate an extended call on INPUT, and will contain the second 2 bytes of the system Revision Number on OUTPUT. If this word is set to ?PKR1 a new doubleword byte pointer has been appended to the packet. This byte pointer is to the ?MXPL—byte buffer which will hold the pathname. If this word is zero, an unextended ?SINFO call is being made, and the packet does not contain the pathname byte pointer.

If either the word pointer in AC2 or the byte pointers in the packet refer to locations that are outside your logical address space, ?SINFO fails and the system returns the error code ERMPR in AC0. If ?SIRS is not set to either 0 or ?PKR1, an “Invalid system call parameter” error will result. If an extended call is made, but the byte pointer is invalid, an “Invalid byte pointer” error will result.

Sample Packet

```
PKT:      .BLK      ?SIEX          ;Allocate enough space for
          ;the extended packet. Packet
          ;length = ?SIEX.

          .LOC      PKT+?SIRN      ;Revision number.
          .WORD     0              ;The OS returns this value.

          .LOC      PKT+?SIRS      ;Supply ?PKR1 on input to specify
          .WORD     ?PKR1          ;an extended ?SINFO call.

          .LOC      PKT+?SIMM      ;Maximum memory.
          .DWORD    0              ;The OS returns this value.

          .LOC      PKT+?SILN      ;Byte pointer to buffer
          ;that contains name of
          ;master LD.
          .DWORD    MLD*2          ;Buffer is MLD.

          .LOC      PKT+?SIID      ;Byte pointer to system
          ;ID buffer.
          .DWORD    SID*2          ;Buffer is SID.

          .LOC      PKT+?SIOS      ;Byte pointer to buffer
          ;that contains name of
          ;executing op sys.
          .DWORD    SOS*2          ;Buffer is SOS.

          .LOC      PKT+?SSIN      ;OS identifying number.
          .WORD     0              ;The OS returns this value.

          .LOC      PKT+?SIEX      ;End of packet.

MLD:      .BLK      16.
SID:      .BLK      16.
SOS:      .BLK      128.
```

If the ?PFFO flag in offset ?PFFLG is not set, ?SOPPF causes the following to occur:

- The caller (user) passes the unique protected file ID as input.
- The operating system returns an error if the caller does not have explicit permission from the first opener of the shared file to open it.
- The operating system returns a channel number for the caller to use.
- The operating system grants the caller those access privileges that the first opener's ?PMTPF specified.

Notes

- See the description of ?PMTPF' in this chapter.

?SPAGE

Performs a shared–page read.

?SPAGE [*packet address*]

error return

normal return

Input

| | |
|-----|---|
| AC0 | Reserved (Set to 0.) |
| AC1 | Target file's channel number (returned to AC1 as output to ?SOPEN) |
| AC2 | Address of the ?SPAGE packet, unless you specify the address as an argument to ?SPAGE |

Output

| | |
|-----|------------------------------|
| AC0 | Undefined |
| AC1 | Unchanged |
| AC2 | Address of the ?SPAGE packet |

Error Codes in AC0

ERVBP Invalid byte pointer passed as a system call argument

ERVWP Invalid word pointer passed as a system call argument

ER_FS_DIRECTORY_NOT_AVAILABLE

Directory not available because the LDU was force released (AOS/VS II only)

Why Use It?

You can use ?SPAGE to move a page or pages into the shared area of your logical address space. Offset ?PSTI in the ?SPAGE packet allows you to restrict the calling process to read access only. This is a way of preventing modifications to shared routines or shared data.

Who Can Use It?

There are no special process privileges needed to issue this call, and there are no restrictions concerning file access. Set Bit 0 of offset ?PSTI to limit access to the file to Read access only.

What It Does

?SPAGE validates a shared page or pages in the caller's logical address space. Note that you must open the target file with ?SOPEN before you can issue ?SPAGEs against it.

If the target page is already on the LRU chain and, therefore, memory resident, the operating system checks the caller's access to it, and adds the page to the working set when the caller references it. If the page is not in memory, the operating system reads it into memory before it adds the page to the working set (at the time the page is referenced).

Specify the number of blocks you want to read or write in the right 15 bits of offset ?PSTI.

Figure 2–222 shows the structure of the ?SPAGE packet, and Table 2–194 describes its contents. Note that the packet is structurally identical to the packet for the block I/O calls ?RDB and ?WRB, which are described in this chapter.

?SRCV

Receives data or a control sequence over a BSC line.

?SRCV

error return

normal return

Input

AC0 (Used by multipoint control stations only) One of the following:

- -1 for a general poll
- Byte pointer to a poll address for a specific poll

AC1 Channel number assigned to the line

AC2 Address of the ?SRCV packet

Output

AC0 Unchanged

AC1 Unchanged

AC2 Unchanged

Error Codes in AC0

| | |
|--------|---|
| ERCRC | CRC check (Indicates a cyclical redundancy block check error.) |
| ERDIS | Disconnect occurred on a switched line |
| ERENQ | ENQ received after time-out (The OS detected an ENQ from the sending station after it reached the limit for retries.) |
| EREOT | EOT character received |
| EREPL | End of polling list reached |
| ERISE | Input status error |
| ERLIS | Line in session (You tried to issue two receive initial system calls in a row.) |
| ERLNA | Attempt to enable nonsynchronous line |
| ERNAK | Transmission failure (The OS received a NAK while it was waiting for a line bid, and it has reached its limit for retries (point-to-point lines only).) |
| ERNPL | Polling list not defined (This error occurs only on multipoint lines.) |
| ERNSL | Device associated with the channel number is not a synchronous line |
| ERSCS | You tried to issue a receive continue without a prior receive initial |
| ERSIM | There is an outstanding system call on this line |
| ERTOFO | Time-out value exceeded |
| ERTRF | Transmitter failure |
| ERUNI | Unknown or inappropriate response received |
| ERVBP | Invalid byte pointer passed as a system call argument |

Why Use It?

You can use ?SRCV whenever you want to receive data over an enabled BSC line.

Who Can Use It?

There are no special process privileges needed to issue this call, and there are no restrictions concerning file access.

?SRCV Continued

What It Does

?SRCV prepared the calling station to receive a block of data or data-link control characters over an enabled BSC line. Before you issue ?SRCV, perform the following steps:

1. Reserve a receive buffer in your logical address space for the incoming data or control sequence.
2. Set up the ?SRCV packet in your logical address space.
3. Load AC1 with the BSC line's channel number.
4. Load AC2 with the packet address.

Figure 2-223 shows the structure of the ?SRCV packet, and Table 2-196 describes each offset.

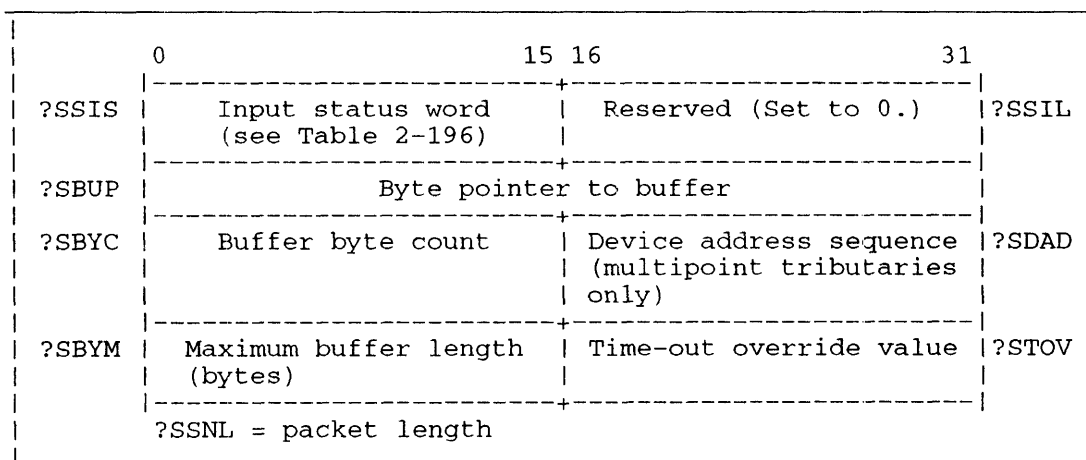


Figure 2-223. Structure of ?SRCV Packet

Table 2-196. Contents of ?SRCV Packet*

| Offset | Contents | Input Value | Output Value |
|--------|---|--|---|
| ?SSIS | Input status word: | | |
| | Text mode. | None. | ?TRAN---If received transparent data. |
| | Block start. | None. | ?STXB---STX received.
?SOHB---SOH received. |
| | Intermediate text block indicator. | None. | ?SITB. |
| | Multipoint information. | None. | ?SPLR---Your system was polled.
?SSLR---Your system was selected. |
| | Station ID indicator | ?SRID--Receive a station's ID or send yours. | |
| | Response type (receive continue only) | | |
| | (Directs OS to respond to the sender with the specified control character.) | ?SRVI--Reverse interrupt.
?SNAK--NAK.
?SACK--ACK0 or ACK1.
?SAK0--Override ACK1, send ACK0.
?SAK1--Override ACK0, send ACK1. | Unchanged. |
| | Call type. | ?CINT--Receive initial.
?CONT--Receive continue. | If ?CINT was set and the bid for the line was succesful, the call type is changed to ?CONT. |

* There is no default unless otherwise specified.

(continued)

?SRCV Continued

Table 2-196. Contents of ?SRCV Packet*

| Offset | Contents | Input Value | Output Value |
|------------------------|-----------------------|---|--|
| ?SSIS
(cont.) | Call type.
(cont.) | The following apply to receive continues from multipoint tributaries only:

?SWAK--Send WACK; then wait for poll or select.
?SPET--Send EOT; then wait for poll or select.
?SPRV--Send RVI; then wait for poll or select.
?SPNK--Send NAK; then wait for poll or select. | |
| | Component Selection. | None. | ?SDAC--Received device address characters. |
| ?SSIL | Reserved. | Set to 0. | N/A. |
| ?SBUP
(double word) | Buffer byte pointer. | Byte pointer to a receive buffer for the data. | Unchanged. |
| ?SBYC | Buffer byte count. | None. | Number of bytes received. |

* There is no default unless otherwise specified.

(continued)

Why Use It?

?SSND allows you to send a block of data over an enabled BSC line.

Who Can Use It?

There are no special process privileges needed to issue this call, and there are no restrictions concerning file access.

What It Does

?SSND lets the calling station send a block of data or data-link control characters over an enabled BSC line. A process can issue ?SSND to bid for the BSC line (send initial), to bid for the line and then send data (send continue), or to send data (send continue).

Before you issue ?SSND, set up a packet in your logical address space, load AC2 with its address, and load AC1 with the line's channel number. Figure 2-225 shows the structure of the ?SSND packet, and Table 2-198 describes each offset.

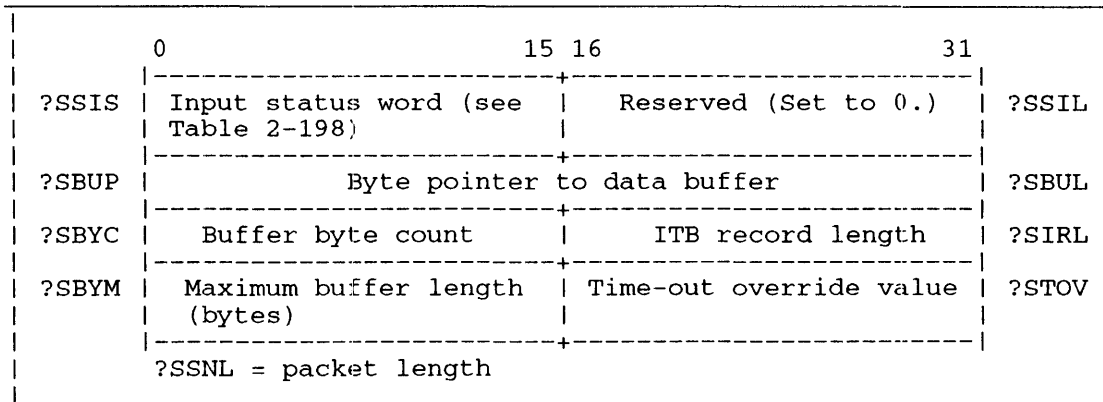


Figure 2-225. Structure of ?SSND Packet

?SSND Continued

Table 2-198. Contents of ?SSND Packet*

| Offset | Contents |
|--------|--|
| ?SSIS | <p>Input status word.</p> <p>Data Type.</p> <p>?NTRN--Nontransparent Text Mode.
?TRAN--Transparent Text Mode.</p> <p>DEFAULT = ?NTRN (Nontransparent Text Mode).</p> <p>Block Start.</p> <p>?STXB--Start of text.
?SOHB--Start of header.</p> <p>DEFAULT = ?STXB (start of text).</p> <p>Block Type.</p> <p>?ETBB--End of block.
?ETXB--End of text.</p> <p>DEFAULT = ?ETBB (end of block).</p> <p>Send Intermediate Text Blocks.</p> <p>?SITB.</p> <p>DEFAULT = No intermediate text blocks.</p> <p>Accept Conversational Replies.</p> <p>?SCON.</p> <p>DEFAULT = Do not accept conversational replies.</p> <p>Send Device Address Characters (for select sequences only).</p> <p>?SDAC.</p> <p>DEFAULT = Do not send device address.</p> |

* There is no default unless otherwise specified.

(continued)

Table 2-198. Contents of ?SSND Packet*

| Offset | Contents |
|--------------------|---|
| ?SSIS (cont.) | Call type.
?CINT--send initial (with data).
?CONT--send continue (with data).
?SWAK--send WACK sequence.
?SEOT--send EOT sequence.
?SDET--send DLE ETB or DLE ETX sequence (to signal end of transparent text mode).
?STTD--send TTD sequence (to signal temporary text delay).
?SINT--send RVI.
?SDIS--perform logical disconnect.
?SNID--send your station's ID. |
| ?SRES | Reserved. (Set to 0.) |
| ?SBUP (doubleword) | Buffer byte pointer.
This is a byte pointer to the buffer in which the data you are sending resides. |
| ?SBYC | Buffer byte count.
This is the number of bytes you wish to send. |
| ?SIRL | ITB record length.
This is the length (in bytes) of the intermediate text blocks, which are records delimited by an ITB character. The ITB record length should not be greater than 255.
DEFAULT = 0 (no ITBs). |
| ?SBYM | Maximum buffer length.
This is the byte length of the data buffer. |
| ?STOV | Time-out override value in seconds.
This is the length of time the OS will wait for a reply before it begins the BSC error-recovery procedures.
DEFAULT = -1 (8 seconds). |

* There is no default unless otherwise specified. (concluded)

Input Status Word

Offset ?SSIS, the input status word, defines certain characteristics of the block you will send.

The data type field specifies whether the data is in Transparent Text Mode (?TRAN), or Nontransparent Text Mode (?NTRN). Under transparent text mode, most data-link control characters are treated only as bit patterns; that is, they have no control significance. The exceptions are DLE STX, which signals the beginning of Transparent Text Mode, and DLE ETB or DLE ETX, which signals the end of Transparent Text Mode.

Although the operating system inserts the necessary data-link characters, including the BCC character, when you issue ?SSND, three fields in offset ?SSIS let you specify the data-link characters you want to use.

?SSND Continued

The block start field specifies whether the operating system should preface the block with an STX (start of text) character or with an SOH (start of header) character. Set mask ?SOHB if you are sending header information in this transmission and set mask ?STXB if you are sending text. The ?STXB mask is the default.

The block type field directs the operating system to append either an ETB (end of text block) or an ETX (end of text) character to the data block. The ETX character terminates the last text block in a message. Thus, you should set mask ?ETXB if this is the last block you are sending. The default for this field is ?ETBB. This mask corresponds to the ETB character, which must terminate every text block except the last one.

The send intermediate text block field directs the operating system to delimit portions of the data block with intermediate transmission (ITB) characters. If you select this option (by setting mask ?SITB), you must set offset ?SIRL to the length of the intermediate text blocks. If you set ?SIRL to 4, for example, the operating system places an ITB character after every fourth byte in the text block. The operating system ignores the contents of ?SIRL if bit ?SITB is not set.

Select the ?SCON mask if you want to accept conversational replies from the receiving station. Conversation mode allows the receiver to respond to the block you send with header information or data, instead of a positive ACK0 or ACK1 sequence. Note that the block you send must be a complete data block (not a data-link sequence) for conversational mode to work. In general, you should not give a conversational reply to a header block or a block that ends with an ETB character.

The call type field defines ?SSND as one of the following types:

- ?CINT, which is a send initial with data. (If the bid for the line is successful, the operating system changes this to ?CONT.)
- ?CONT, which is a send continue with data.
- ?SWAK, which is a wait-before-transmission (positive acknowledgment).
- ?SEOT, which is an end-of-transmission.
- ?SDET, which is a line disconnect call.
- ?STTD, which is a temporary text delay.
- ?SNID, which allows you to send your station's ID. Place the ID in the data buffer, its number of bytes in offset ?SBYC, and the largest possible number of bytes in the expected reply ID in offset ?SBYM. When ?SSND finishes you must check offset ?SBYC to see if the receiving station replied with an ID. If so, it will be in the data buffer. See the section "Station Identification" in the description of system call ?SEBL.

Table 2-199 describes each type of call.

Other ?SSND Offsets

Offsets ?SBUP and ?SBUL point to the buffer you reserved in your logical address space for the data you will send. Note that the buffer byte count you supply in offset ?SBYC must be for the

Why Use It?

Use ?SYLOG to control and monitor the status of ongoing system, Superuser, error, and CON0 (AOS/VS II only) logging. You can use CON0 logging to write to a file, instead of to a hardcopy console. You can also use ?SYLOG to rename the system, the error, and the CON0 log files each day, helping you keep a continuous record of system and CON0 I/O activity.

Who Can Use It?

Only the operator process (PID 2), or a process with System Manager privilege, can issue ?SYLOG. Restrictions concerning file access control are described in the next section.

What It Does

The operator process uses ?SYLOG to find out the current state of each logging function (on or off), and to start and stop system, Superuser, and CON0 logging; you cannot stop error logging with ?SYLOG. System and Superuser logging are written to :SYSLOG including all messages sent by the operating system and/or EXEC, and all messages sent by users via ?LOGEV. Error logging is written to :ERROR_LOG, and CON0 logging to :CON0_LOG. All message logging is buffered in the operating system before it is written to a function's log file.

Stopping and restarting system, Superuser, and CON0 logging flushes each function's message buffer, writing the buffer's content into the function's log file. When the operator process restarts logging, messages are appended to each function's log file. The operator process can use ?SYLOG to create or rename :SYSLOG and :CON0_LOG, but can only rename :ERROR_LOG.

Renaming a log file also flushes its message buffer, and copies the old log file's content plus the buffer's content into the newly named file. You can use renaming to create a continuous record of system activity. Renaming :ERROR_LOG reduces its size to zero bytes, and writes the file plus the buffer content into the newly named file.

When renaming the log file while system logging is enabled, first issue an extended status call to obtain the current level of logging detail. Then issue the rename call with the appropriate detail level set for logging in AC2. When the log file is renamed during a status command, and AC0=-1, AC2 must contain the desired level of logging detail (0 or -1) as well. If you omit the logging detail level, the operating system sets a minimal detail level after renaming the log file, and you could inadvertently change your logging detail level.

You can view :SYSLOG and :ERROR_LOG with the REPORT utility. You can view the ASCII file :CON0_LOG using the CLI TYPE command, or the BROWSE utility. You view a function's most recent messages by renaming the log file with ?SYLOG. For example, to view all the error messages up to the current time, rename :ERROR_LOG. Renaming the file flushes the error log buffer, and writes the most recent error messages and the system time and date stamp into the newly named file. Now use the REPORT utility to view the renamed file's content.

For information about reading :SYSLOG with a program, see Appendix B at the end of AOS/VS, AOS/VS II, and AOS/RT32 System Call Dictionary, ?R Through ?Z (093-000543-02).

The AC0 and AC1 input values determine which function ?SYLOG performs. If you use ?SYLOG to start the system logging function (AC0 = 1) and the system log file does not exist, the operating system creates :SYSLOG. If you use ?SYLOG to start the CON0 logging function (AC0 = 7) and the CON0 log file does not exist, the operating system creates :CON0_LOG.

The access control list for :SYSLOG is NULL, giving only processes with Superuser privilege access to the file and denying access to other users. The :SYSLOG special file type is ?FLOG.

?SYLOG Continued

The access control list for :CON0_LOG is OP,R giving the operator process read access to the file, and denying access to other users. The :CON0_LOG special file type is also ?FLOG. With CON0 logging on, the operating system buffers and then writes all CON0 I/O messages to the file :CON0_LOG.

If you turn off system logging (AC0 = 0), you flush the operating system buffer and close :SYSLOG. Turning system logging off prevents the operating system from logging any further system events or messages to :SYSLOG. Similarly, if you turn off CON0 logging (AC0 = 0), you flush the operating system buffer and close :CON0_LOG. Turning CON0 logging off prevents the operating system from logging any further I/O events or messages to :CON0_LOG.

To rename :SYSLOG, load AC1 with a byte pointer to the new filename before you issue ?SYLOG. The operating system closes the current :SYSLOG file, if it is open, and then renames the file.

If you leave the system logging function on while you rename :SYSLOG, or if you rename the log file and then issue another ?SYLOG to resume system logging, the operating system opens a new system log file called :SYSLOG and writes all subsequent log messages to that file. All previous log files are strictly historical.

You cannot rename :CON0_LOG while CON0 logging is active. To rename :CON0_LOG, turn off CON0 logging (AC0=0), turn on Superuser and load AC1 with a byte pointer to the new filename before you issue ?SYLOG. Now turn off Superuser and restart CON0 logging to :CON0_LOG.

If an error occurs during ?SYLOG, the operating system leaves the current state of the system log file and its name intact.

Exclusion Bit Map

An exclusion packet can be used with ?SYLOG to selectively ignore certain events. Figure 2–243 shows the structure of the exclusion bit map packet. The exclusion packet is a bit map record. The ordinal bit positions are defined in PARU.32.SR from ?LSMI to ?LUMAX (Only ?LSMI to ?LSMA apply to Superuser logging exclusion bit maps.).

To retrieve and view the current bit map settings before changing them, use ?SLRF in AC0. In the first double word (?SYFBM) of the exclusion bit map packet define either ?SYLID or ?SYSUID. ?SYLID indicates a selective bit map, ?SYSUID indicates a Superuser bit map. When setting an exclusion or a Superuser bit map with ?SLSF, if a bit is set in the packet, the system will not log that event.

You can set exclusion bit maps only while system logging is on, and you can set Superuser exclusion bit maps only while Superuser logging is on. Also, Superuser exclusion maps do not affect PMGR system logging events.

The operating system writes the “Set Exclusion Map” message in a SYSLOG entry with code ?LTSF.

NOTE: We recommend that you always include the AOS/VS administration and hardware codes (1–899.).

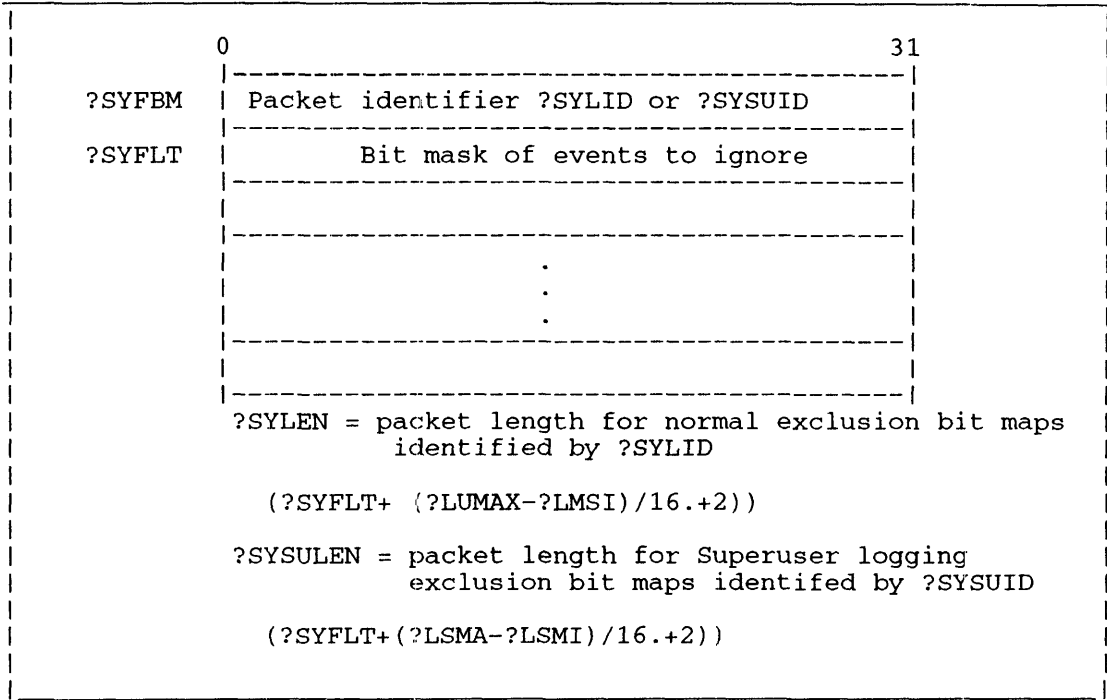


Figure 2-226. Structure of ?SYLOG Exclusion Bit Map Packet

The first two words in the packet contain the system call identifier ?SYLID or ?SYSUID. Place ?SYLID here for ordinary exclusion bit maps, or place ?SYSUID here for bit maps used for superusers during Superuser logging. The remainder of the packet is a bit map record that contains a bit entry for every possible log event. Bit positions correspond to the event codes that are defined in PARU.32.SR. To ignore specific events, set the corresponding bit for that event in the packet.

Notes

- See the description of ?LOGEV in this chapter.
- See the explanation of the CLI command SYSLOG in the manual *Managing AOS/VS and AOS/VS II*.
- Stopping ?SYLOG events or using an exclusion bit map precludes a C2-level secure system.

?SYSPRV

Enters, leaves, or examines a privilege state.

?SYSPRV [*packet address*]

error return

normal return

Operating System Differences

See the section “Who Can Use It” below.

Input

AC0 Reserved (Set to 0.)
AC1 Reserved (Set to 0.)
AC2 Address of the ?SYSPRV packet, unless you specify the address as an argument to ?SYSPRV

Output

AC0 Unchanged
AC1 Unchanged
AC2 Address of the ?SYSPRV packet

Error Codes in AC0

ERPRE Invalid system call parameter
ERPRV Caller not privileged for this action
ERPVM Unknown privilege mode
ERPVO Privilege cannot be held exclusively
ERPVP Other processes using privilege
ERPVX Privilege held exclusively by other process

Why Use It?

Use this system call to inquire about or change certain process privilege states. You need certain privileges before you can execute some system calls. For example, your process must have System Manager privilege turned on before it can issue a ?JPINIT call to initialize a job processor.

Who Can Use It?

Under AOS/VS, there are no restrictions concerning file access. The following list contains the AOS/VS process privileges needed to issue ?SYSPRV.

- To enter (i.e., acquire) a privilege state — if you are allowed to have one of the privileges Superuser, Superprocess, or System Manager and issue ?SYSPRV to enter the respective privilege, AOS/VS lets you do this. (One of the ways you are allowed to have, say, Superuser privilege is for your system manager to use the profile editor utility program, PREDITOR, to allow you this privilege.) If you are not allowed to have one of these privileges and issue ?SYSPRV to enter the privilege, AOS/VS returns error ERPRV.
- To enter (i.e., acquire) the System Manager privilege state exclusively — you must be allowed to have System Manager privilege and no other user has entered System Manager privilege.

**We removed the system call ?TMSG from the System Call Dictionary.
Programs using ?TMSG work as they did before we removed ?TMSG.**

**Use ?IREC for old A-type termination messages, and for new B- and
C-type termination messages.**

**We removed the system call ?TMSG from the System Call Dictionary.
Programs using ?TMSG work as they did before we removed ?TMSG.**

**Use ?IREC for old A-type termination messages, and for new B- and
C-type termination messages.**

**We removed the system call ?TMSG from the System Call Dictionary.
Programs using ?TMSG work as they did before we removed ?TMSG.**

**Use ?IREC for old A-type termination messages, and for new B- and
C-type termination messages.**

**We removed the system call ?TMSG from the System Call Dictionary.
Programs using ?TMSG work as they did before we removed ?TMSG.**

**Use ?IREC for old A-type termination messages, and for new B- and
C-type termination messages.**

We removed the system call ?TMSG from the System Call Dictionary. Programs using ?TMSG work as they did before we removed ?TMSG.

Use ?IREC for old A-type termination messages, and for new B- and C-type termination messages.

We removed the system call ?TMSG from the System Call Dictionary. Programs using ?TMSG work as they did before we removed ?TMSG.

Use ?IREC for old A-type termination messages, and for new B- and C-type termination messages.

**We removed the system call ?TMSG from the System Call Dictionary.
Programs using ?TMSG work as they did before we removed ?TMSG.**

**Use ?IREC for old A-type termination messages, and for new B- and
C-type termination messages.**

**We removed the system call ?TMSG from the System Call Dictionary.
Programs using ?TMSG work as they did before we removed ?TMSG.**

**Use ?IREC for old A-type termination messages, and for new B- and
C-type termination messages.**

We removed the system call ?TMSG from the System Call Dictionary.
Programs using ?TMSG work as they did before we removed ?TMSG.

Use ?IREC for old A-type termination messages, and for new B- and
C-type termination messages.

**We removed the system call ?TMSG from the System Call Dictionary.
Programs using ?TMSG work as they did before we removed ?TMSG.**

**Use ?IREC for old A-type termination messages, and for new B- and
C-type termination messages.**

Table 2-205. ?VALIDATE Functions and Their Codes

| Offset | Description |
|----------------------------|--|
| ?VFUNC | FUNCTION CODES and FLAG

?RVWPL--validate an area defined by word pointer and word length for Read access.
?VWVPL--validate an area defined by word pointer and word length for Write access.
?RVBPL--validate an area defined by byte pointer and byte length for Read access
?VWBPL--validate an area defined by byte pointer and byte length for Write access.
?RVBPX--validate an area defined by byte pointer and maximum byte length (unknown actual length) for Read access, using a delimiter table.
?VRNBR--If this flag is set to 1, the ring in offset ?VRING will be interpreted as a ring number (value 4 - 7). If this flag is set to 0, ?VRING will be interpreted as a ring field in bits 1S1-1S3 (value 0;4 to 0;7). |
| ?VRES | Reserved. (Set to 0.) |
| ?VPOINTER
(double-word) | Starting address of the area to be validated. This will be interpreted as a word address for ?RVWPL and ?VWVPL, and as a byte address for the other functions. |
| ?VRING
(double-word) | Ring to be used for validation purposes. See ?VRNBR flag above for interpretation of this offset. |
| ?VLENGTH
(double-word) | Length of the area to be validated. This will be interpreted as a word length for ?RVWPL and ?VWVPL, and as a byte length for the other functions. |
| ?VDELIM
(double-word) | Word address of a 16-bit delimiter table to be used in validating areas of unknown actual length by the validation function ?RVBPX. You may also specify (-1) to designate the default delimiters (null, New Line, and form feed -- ASCII 0, 12, or 14). This parameter is ignored by the other validation functions. |
| ?VERROR
(double-word) | Returns zero if the area was valid for the type of access specified. If this value is not zero, the area was invalid (see "Other Notes", next page). |
| ?VRES
(double-word) | Reserved for expansion. (Set to 0.) |

The packet length is defined by the parameter ?VPLTH.

?VALIDATE Continued

Error Conditions

The exception return for this call will be taken only if the call is invalid for one of the following reasons:

- It was called by a 16-bit process in Ring 7. ERICM — “Invalid System Command” — will indicate this error.
- The packet address in AC2 is invalid. ERVWP will be returned.
- The function code in ?VFUNC of the packet is unknown. ERUFR will be returned in this case.
- For function ?RVBPX only, if the contents of ?VDELIM is not (-1) or a valid address, error code ERVWP will be returned.
- The ring specified in ?VRING is invalid. ERVPW will be returned.

Note that the normal return indicates that the call was made correctly. However, it does NOT indicate the validity of the area. You must check the error code returned by ?VERROR, which will be zero if the type of access specified is possible. If ?VERROR is nonzero, it will be one of the following codes:

ERVWP This code is returned for functions ?RVWPL or ?WWWPL if the specified area is invalid for the indicated type of access.

ERVBP This code is returned for functions ?RVBPL, ?WVBPL, or ?RVBPX if the specified area is invalid for the indicated access type.

ERPRE will be reported for the following error conditions:

- The area length specified by ?VLEN is invalid. The area length must be a positive number which does not exceed the number of words or bytes in one ring.

Notes

- An area will be considered invalid if it crosses the unshared/shared boundary.

Examples of Use

Suppose an inner ring server for a database system gets a request to return some information from a database. The database caller, a Ring 7 application in this example, passes a byte address and byte length to specify the receiving area.

The database server may attempt to write the information into the customer's space without validation; however, it risks causing a USER TRAP from its ring. By calling ?VALIDATE with Ring 7 in ?VRING, and the user's byte address and length in ?VPTR and ?VLEN, the database server can simply return an error code for the request.

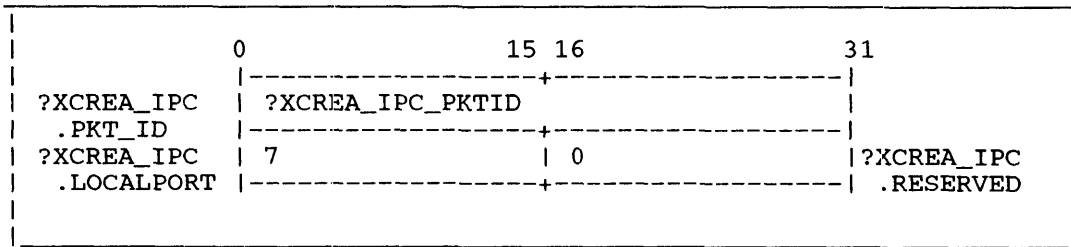


Figure 2-264. Example of ?XCREATE IPC Subpacket for IPC File

Example of Link File Creation

In this example (Figures 2-265 and 2-266), the caller wants to create a link file to, :UDD:BRIANNE:BRIANNE_MEMO, from his working directory, :UDD:DOUGLAS. The caller names the link filename, NOT_MY_MEMO. The caller sets the packets as follows:

Set AC0 = 0

Set AC1 = 0

Set AC2 = Word pointer to main packet

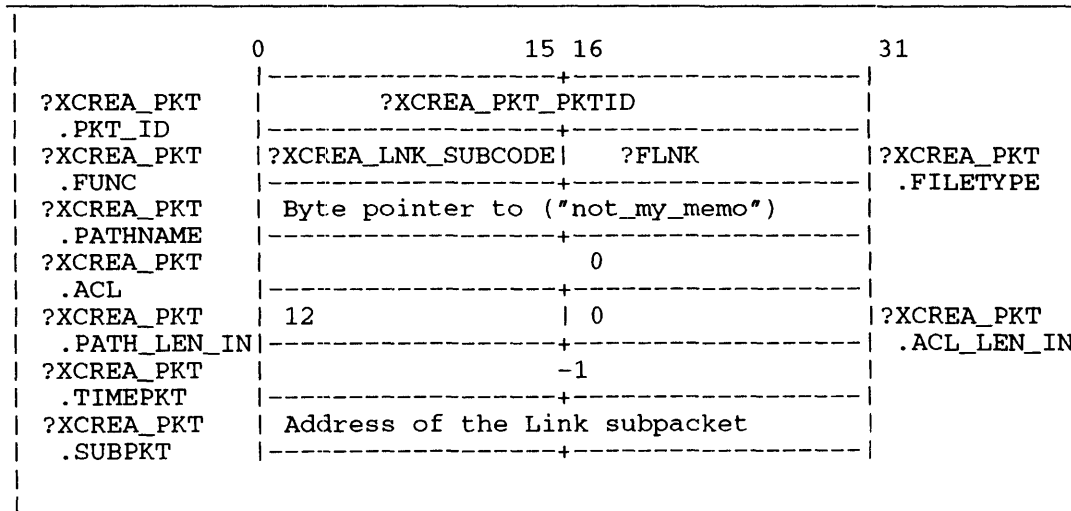


Figure 2-265. Example of ?XCREATE Main Packet for LNK File

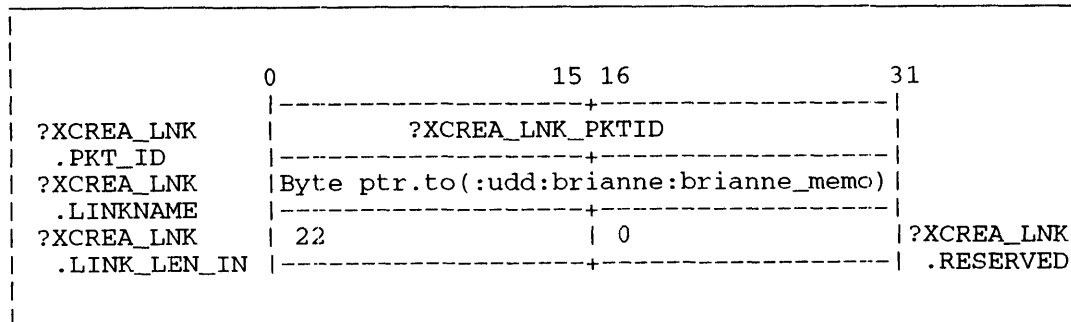


Figure 2-266. Example of ?XCREATE Link Subpacket for LNK File

?XFSTAT

Gets file status information (extended).

AOS/VS II only

?XFSTAT [*packet address*]

error return

normal return

Input

AC0 Reserved (set to 0)
AC1 Reserved (set to 0)
AC2 The system call packet address,
unless you specify the address
as an argument to ?XFSTAT.

Output

AC0 Unchanged or an error code
AC1 Unchanged
AC2 Unchanged

Error Codes in AC0

ERPKT Illegal packet id
ERRVN Reserved value not zero
ERFAD File access denied
ERCIU Channel in use
ERFNO Channel not open
ERICN Illegal channel
ERVWP Invalid word pointer passed as a system call argument
ERVBP Invalid byte pointer passed as system call argument
ERUOL Physical Unit offline
ER_FS_INVALID_XFSTAT_FUNCTION
The ?XFSTAT call does not support subpackets and you specified one. The
?XFSTAT_PKT.FUNC must be set to zero (AOS/VS II only)
ER_FS_DIRECTORY_NOT_AVAILABLE
Directory not available because the LDU was force released (AOS/VS II only)
ER_FS_DIR_DOES_NOT_EXIST
The operating system cannot find the directory
ER_FS_FILE_CLASS_NOT_IMPLEMENTED
You specified an invalid packet ID
ER_FS_FILENAME_DOES_NOT_EXIST
The operating system cannot find the file
ER_FS_ILLEGAL_FILENAME_CHAR
You specified a filename with an illegal character
ER_FS_INVALID_PACKET_REV
The ?XFSTAT packet does not match the current revision of the operating system
ER_FS_INVALID_PATHNAME_BYTE_PTR
You specified an invalid byte pointer to the pathname
ER_FS_INVALID_XFSTAT_OPTIONS
You specified an invalid ?XFSTAT_PKT.OPTIONS flag
ER_FS_SAM_DISK_CURRENTLY_OPEN_EXCLUSIVE
The unit is exclusively opened by another process
ER_FS_ZERO_LENGTH_FILENAME
The specified file contains zero bytes

Table 2-233. ?XFSTAT Unit Packet File Status

| Offset | Contents |
|-------------------------------------|--|
| ?XFSTAT_PKT
.DEF_DENSITY | Returns default density (bytes per inch) of the tape unit. It can be one of the following:

?XFSTAT_DENSITY_800 800 BPI density
?XFSTAT_DENSITY_1600 1600 BPI density
?XFSTAT_DENSITY_6250 6250 BPI density
?XFSTAT_DENSITY_AUTO Automatic density
?XFSTAT_DENSITY_LOW Low density
?XFSTAT_DENSITY_MED Medium density
?XFSTAT_DENSITY_HIGH High density
?XFSTAT_DENSITY_NC No change |
| ?XFSTAT_PKT
.TRANSFER_SIZE | Maximum tape transfer buffer byte size. |
| ?XFSTAT_PKT
.DSK_SMR
_SECTOR | Second most reliable (SMR) sector. The file system uses the SMR for each disk and stores a copy of the physical disk information table in the sector. The SMR sector location varies according to the disk type and size. |
| ?XFSTAT_PKT
.DSK_SECTORS | Total number of unformatted disk blocks available. |
| ?XFSTAT_PKT
.DSK_PATTERN1 | First disk test pattern. |
| ?XFSTAT_PKT
.DSK_PATTERN2 | Second disk test pattern. |
| ?XFSTAT_PKT
.DSK_PATTERN3 | Third disk test pattern. |
| ?XFSTAT_PKT
.DSK_PATTERN4 | Fourth disk test pattern. |
| ?XFSTAT_PKT
.DSK_PATTERN5 | Fifth disk test pattern. |
| ?XFSTAT_PKT
.DSK_PATTERN6 | Sixth disk test pattern. |
| ?XFSTAT_PKT
.DSK_PATTERN7 | Seventh disk test pattern. |
| ?XFSTAT_PKT
.DSK_PATTERN8 | Eighth disk test pattern. |
| ?XFSTAT_PKT
.DSK_PATTERN
_CNT | Number of valid disk test patterns returned in the packet. The patterns test for bad blocks prior to disk formatting. |

(concluded)

Disk test patterns refer to the bad block verification that is usually performed on disks prior to formatting. Error detection programs are run against the patterns written to the disk to find the bad blocks.

?XFSTAT Continued

The intensity of error checking is proportional to the number of patterns used. The number and type of patterns may differ between and within disk classes. There can exist up to eight distinct pattern fields. A pattern count field identifies the number of valid pattern fields, which are sequentially ordered from `patrn` one. Unused fields are set to zero.

16-bit System Call Support

The ?XFSTAT system call supports a 16-bit version of the call. However, 16-bit callers are responsible for supplying the packet address in the 16-bit address format, which includes the ring field. Callers will use the same packet offset symbols as defined for 32-bit call.

Example with Pathname Supplied

The caller wants file information for the entry specified in the target pathname, `:UDD:BRIANNE:MEMO.LS`. In the ?XFSTAT packet, the caller should set the packet as follows:

| | |
|-------------------------|--|
| ?XFSTAT_PKT.PKT_ID | = ?XFSTAT_PKT_PKTID |
| ?XFSTAT_PKT.FUNC | = 0 |
| ?XFSTAT_PKT.CHANNEL | = 0 |
| ?XFSTAT_PKT.PATHNAME | = byte pointer to <code>:UDD:BRIANNE:MEMO.LS</code> |
| ?XFSTAT_PKT.PATH_LEN_IN | = 22 (Caller's pathname buffer is 22 bytes,
including null delimiter) |
| ?XFSTAT_PKT.RESERVED | = 0 |
| ?XFSTAT_PKT.OPTIONS | = ?XFSTAT_OPTIONS_PATHNAME |

Sets `AC2` = word addr. (?XFSTAT_PKT).

After issuing the call, the packet's output fields contain the requested information. The input fields remain unchanged. All packet values are in octal.

The packet information tells the caller that file, `MEMO.LS`, is a user data file (?FUDF); is currently a 1 index level file; has a byte size of 101144; has 100 blocks allocated; its primary element size is 104 while its secondary element size is 4. Determining the values in parentheses are normally found by the symbolic flags defined.

| | 0 | 15 16 | 31 |
|------------------------------|-------------------------|-------|----------------------------|
| ?XFSTAT_PKT
.PKT_ID | ?XFSTAT_PKT_PKTID | | |
| ?XFSTAT_PKT
.FUNC | 0 | | ?XFSTAT_PKT
.CHANNEL |
| ?XFSTAT_PKT
.PATHNAME | 0 | | |
| ?XFSTAT_PKT
.PKT_LEN_IN | 0 | 0 | ?XFSTAT_PKT
.RESERVED |
| ?XFSTAT_PKT
.OPTIONS | ?XFSTAT_OPTIONS_CHANNEL | | |
| ?XFSTAT_PKT
.FILETYPE | ?FDIR | 2003 | ?XFSTAT_PKT
.STATUS |
| | 0 | 0 | |
| ?XFSTAT_PKT
.TCREATE | 16474 | 72333 | |
| ?XFSTAT_PKT
.TLACCESS | 16516 | 65673 | |
| ?XFSTAT_PKT
.TLMODIFY | 16516 | 65673 | |
| ?XFSTAT_PKT
.INDEX_MAX | 3 | 2 | ?XFSTAT_PKT
.INDEX_ELEM |
| ?XFSTAT_PKT
.INDEX_CUR | 1 | 0 | |
| ?XFSTAT_PKT
.FILE_BYTESZ | 1000 | | |
| ?XFSTAT_PKT
.START_LDA | 10751 | | |
| ?XFSTAT_PKT
.PRIMARY_ELEM | 2 | | |
| ?XFSTAT_PKT
.SECNDRY_ELEM | 1 | | |
| ?XFSTAT_PKT
.PRIMARY_CNT | 2 | | |
| ?XFSTAT_PKT
.CPD_MAX_SIZE | 0 | | |
| ?XFSTAT_PKT
.BLOCKS_ALLOC | 1222 | | |
| ?XFSTAT_PKT
.DSK_ACCESSES | 1223 | | |
| | 0 | | |
| ?XFSTAT_PKT
.ACTIVE_OPENS | 1 | 0 | |

Figure 2-272. Example of Directory Type Grouping

Notes

- Should additional information be included in the future, DG will assign a new packet identifier. To access the new information, the caller should specify the most recent packet identifier and packet length constant.
- When you set the ?XFSTAT_OPTIONS_GET_DISK_INFO flag and the file type resolves to a disk unit, the ?XFSTAT call requires additional time to open the disk (if necessary) and to read the information. If the flag remains set for all of the file types, the file system ignores the flag except for a disk unit status.

?XGTACP

Gets access control privileges (extended).

AOS/VS II only

?XGTACP [*packet address*]

error return

normal return

Input

| | |
|-----|---|
| AC0 | Reserved (Set to 0) |
| AC1 | Reserved (Set to 0) |
| AC2 | ?XGTACP Packet Address
unless specified as a
system call parameter. |

Output

| | |
|-----|------------|
| AC0 | Error code |
| AC1 | Unchanged |
| AC2 | Unchanged |

Error Codes in AC0

| | |
|-----------------------------------|---|
| ERFNO | Channel not open |
| ERICN | Illegal channel |
| ERPKT | Invalid packet ID |
| ERPRE | Invalid system call parameter |
| ERPRH | Attempt to access process not in hierarchy |
| ERPRV | Caller not privileged for this action |
| ERRVN | Reserved value not zero |
| ERVBP | Invalid byte pointer passed as a system call argument |
| ER_FS_ILLEGAL_GROUP_LIST_FORMAT | |
| ER_FS_GROUPNAME_TOO_LONG | |
| ER_FS_ILLEGAL_GROUPNAME_CHARACTER | |
| ER_FS_INVALID_GROUP_LIST_BYTE_PTR | |
| ER_FS_INVALID_PATHNAME_BYTE_PTR | |
| ER_FS_INVALID_PKT_PTR | |
| ER_FS_INVALID_USERNAME_BYTE_PTR | |
| ER_FS_INVALID_XGTACP_FUNCTION | |
| ER_FS_TOO_MANY_GROUPS_SPECIFIED | |
| ER_FS_USERNAME_TOO_LONG | |
| ER_FS_ZERO_LENGTH_USERNAME | |
| ER_FS_DIRECTORY_NOT_AVAILABLE | |

Directory not available because the LDU was force released (AOS/VS II only)

Why Use It?

?XGTACP returns the access privileges for a specific file and username/group list combination. You can issue ?XGTACP before you issue ?SACL, which sets the current ACL for a file, or before you issue ?DACL, which sets the default ACL.

Who Can Use It?

You need Execute access to the directory of the file to determine your own access rights to the file. When you set ?XGTACP_USER_OR_PID to -1, the call returns the access rights based on the current username/group list combination of the calling process. Others must have the Superuser privilege.

Table B-1. SYSLOG Event Codes and Record Lengths (cont.)

| Event Code
(decimal) | Meaning | Message
Length
(beyond
headers) |
|---------------------------------|---|--|
| 81 | Host-request-to-SCP error — Error log. | 0 |
| 96 | SCP logging enabled — Error log. | 0 |
| 97 | SCP logging disabled — Error log. | 0 |
| 98 | Main processor halt — Error log. | 0 |
| 99 | BOOT issued (MV/8000) — Error log. | 0 |
| 100 | Power failure — Error log. | 0 |
| 101 | Power restore — Error log. | 0 |
| 102 | Air flow fault — Error log. | 0 |
| 103 | Overtemp fault (not recorded). | 0 |
| 104 | Transfer to battery backup — Error log. | 0 |
| 105 | Reserved. | — |
| 106 | ERCC error, MV/8000 and MV/6000 —
Error log. | 4 |
| 107 | Microsequencer parity error — Error log. | 0 |
| 108 | System cache parity error — Error log. | 0 |
| 109 | Cache to Bank controller parity error — Error
log. | 0 |
| 110 | IOC bus parity error — Error log. | 0 |
| 111 | S-bus timeout — Error log. | 0 |
| 112 | S-bus parity error — Error log. | 0 |
| 113 | Operating system error (unused). | 0 |
| 114 | Diskette log error (MV/8000). | 0 |
| 115 | Infinite protection fault — Error log. | 0 |
| 116 | Infinite page fault — Error log. | 0 |
| 117 | Instruction cache enabled — Error log. | 0 |

(continued)

Table B-1. SYSLOG Event Codes and Record Lengths (cont.)

| Event Code
(decimal) | Meaning | Message
Length
(beyond
headers) |
|---------------------------------|--|--|
| 118 | Instruction cache disabled — Error log. | 0 |
| 119 | Reserved. | — |
| 120 | Reserved. | — |
| 121 | System reset (not recorded). | 0 |
| 122 | ATU accelerator enabled — Error log. | 0 |
| 123 | ATU accelerator disabled — Error log. | 0 |
| 125 | XEQ DTOS command (MV/8000). | 0 |
| 126 | Bad return from DTOS (MV/8000). | 0 |
| 127 | HALT command (MV/8000) — Error log. | 0 |
| 128 | CONTINUE command (MV/8000) — Error log. | 0 |
| 129 | START command (MV/8000) — Error log. | 0 |
| 130 | INIT command (MV/8000) — Error log. | 0 |
| 131 | Bank controller ERCC report disable — Error log. | 0 |
| 132 | Good return from DTOS (MV/8000). | 0 |
| 133 | Hard interrupt (not recorded). | 0 |
| 141 | Unsolicited error from peripheral (H.A.D.A./MV) | 3 |
| 142 | Soft tape error | 17 |
| 901 | Reserved. | 0 |
| 910 | Process created. | Varies |
| 911 | Reserved. | 0 |
| 912 | Process terminated by superior process. | 4 |

(continued)

Table B-1. SYSLOG Event Codes and Record Lengths (cont.)

| Event Code
(decimal) | Meaning | Message
Length
(beyond
headers) |
|---------------------------------|--|--|
| 913 | Superuser turned on or off. | Varies |
| 914 | Superprocess turned on or off. | Varies |
| 915 | Access devices turned on or off (?IDEF turns on; ?IRMV turns off). | 4 |
| 916 | Process loaded program into ring (used ?RINGLD system call). | Varies |

(continued)

| | |
|------|---------------------------------------|
| Word | Code 43, 44, or 45 |
| 8 | Code (43, 44, or 45) |
| 9 | Cause (CPU, I/O access, sniff, other) |
| 10 | Physical page #1 |
| 11 | Double word on module |
| 12 | Syndrome bits |

| | |
|------|-----------|
| Word | Code 106 |
| 8 | ERCC code |
| 9 | Module |
| 10 | Plane # |
| 11 | Bit # |

| | |
|------|-------------|
| Word | Code 141 |
| 8 | DIA Status |
| 9 | DIB Status |
| 10 | Device Code |

| | |
|------|---|
| Word | Code 142 |
| 8 | Device code |
| 9 | Unit number |
| 10 | Suppress Soft error flag (0=suppress, 1=report) |
| 11 | Total number of re-writes |
| 12 | Total number of write errors corrected |
| 14 | Total number of bytes written to the media |

(continued)

| | |
|------|--|
| Word | Code 142 |
| 16 | Total number of re-reads |
| 17 | Total number of read errors corrected |
| 19 | Total number of bytes read from the media |
| 21 | Acceptable Error/Bytes ratio limit |
| 22 | Marginal Error/Bytes ratio limit |
| 23 | Error/Bytes transferred ratio |
| 24 | Acceptance levels:
Good = 01
Marginal = 02
Bad = 03 |

| | |
|------|---|
| Word | Code 910 (process created) |
| 8 | Error code if any (Note 1) |
| 9 | |
| 10 | 0 |
| 11 | Privilege bits (?PROC format) |
| 12 | 0 |
| 13 | New PID |
| 14 | Username, in ASCII, 16 characters, 8 words |
| 21 | |
| 22 | Program (.PR) file pathname (ASCII) or terminated by null |
| n | |

| | |
|------|-------------------------------|
| Word | Code 912 (process terminated) |
| 8 | Error code if any (Note 1) |
| 9 | |
| 10 | Target PID |
| 11 | |

| | |
|------|--|
| Word | Code 913, 914, 964 |
| 8 | Error code (if any) |
| 9 | |
| 10 | n (0 means no change, 1 means off, 2 means on) |
| 11 | |
| 12 | |
| 19 | ASCII USERNAME (Note 4) |

| | |
|------|--|
| Word | Code 915 |
| 8 | Error code if any (Note 1) |
| 9 | |
| 10 | n (0 means no change, 1 means off, 2 means on) |
| 11 | |

| | |
|------|--|
| Word | Code 916 (?RINGLD) |
| 8 | Error code if any (Note 1) |
| 9 | |
| 10 | 0 |
| 11 | Ring number |
| 12 | Program (.PR) file pathname loaded, in ASCII, terminated by null |
| n | |

| | |
|------|---|
| Word | Code 917 (chain) |
| 8 | Error code if any (Note 1) |
| 9 | |
| 10 | Program (.PR) file pathname, in ASCII, terminated by null |
| n | |

Figure B-2. Log Record Codes, Events, and Message Lengths, Excluding Header (continued)

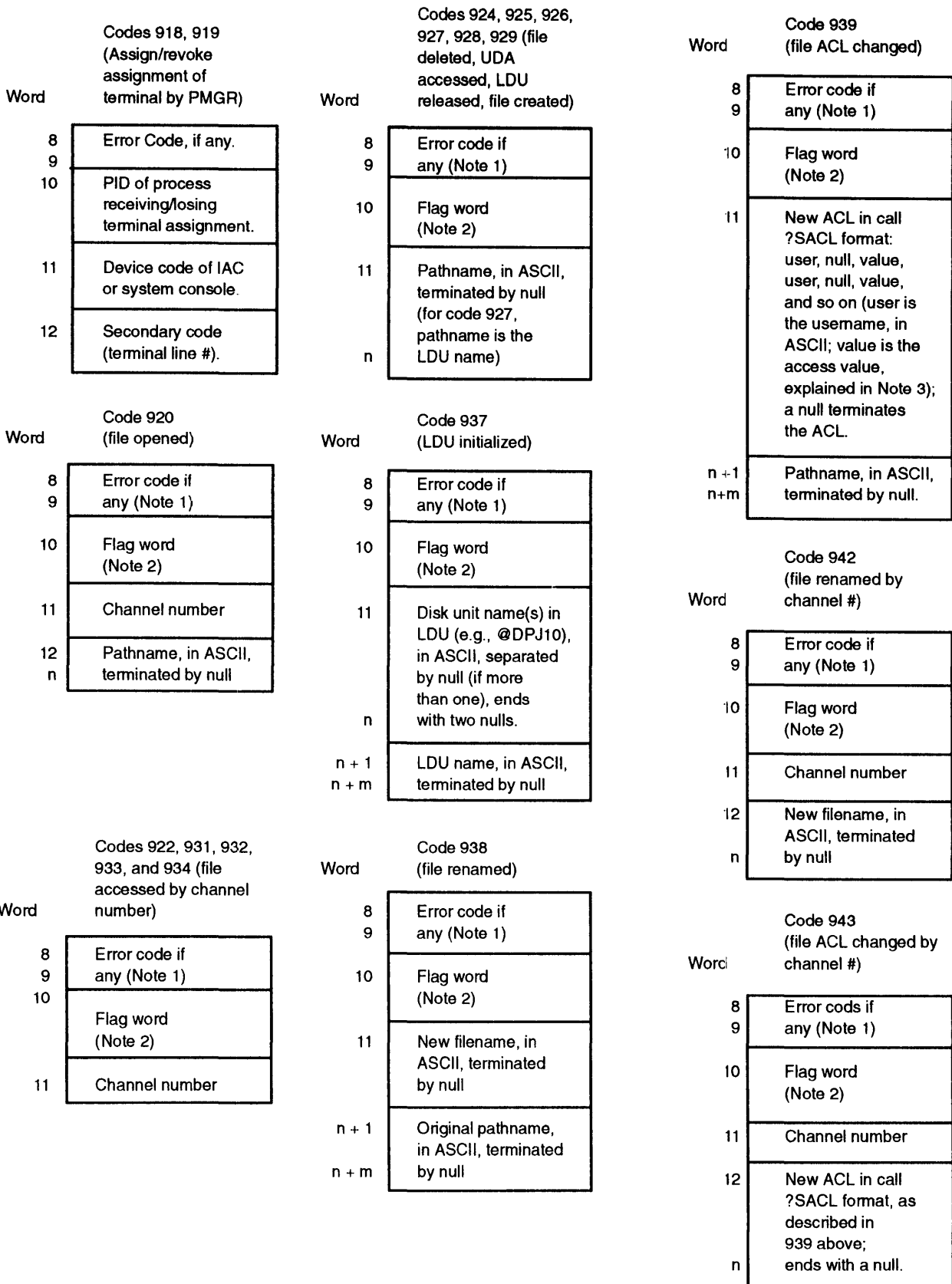


Figure B-2. Log Record Codes, Events, and Message Lengths, Excluding Header (continued)

| Word | Code 945 (shared file opened, first open) |
|------|---|
| 8 | Error code if any (Note 1) |
| 9 | |
| 10 | Flag word (Note 2) |
| 11 | Channel number |
| 12 | File identifier |
| 13 | |
| 14 | Caller's ring |
| 15 | Pathname, in ASCII, terminated by null |
| n | |

| Word | Code 946 (shared file opened, subsequent open) |
|------|--|
| 8 | Error code if any (Note 1) |
| 9 | |
| 10 | Flag word (Note 2) |
| 11 | Channel number |
| 12 | File identifier |
| 13 | |
| 14 | Caller's ring |

| Word | Code 947 (permitted access to, protected word file) |
|------|---|
| 8 | Error code if any (Note 1) |
| 9 | |
| 10 | Flag Word (Note 2) |
| 11 | ACL value (Note 3) |
| 12 | File identifier |
| 13 | |
| 14 | Caller's ring |
| 15 | Target ring |
| 16 | Target PID |

Figure B-2. Log Record Codes, Events, and Message Lengths, Excluding Header (continued)

Index

Within the index, a bold page number indicates a primary reference. A range of page numbers indicates the reference spans those pages. A reference such as ?RTODC_PKT... indicates that all references beginning with those characters are found on the referenced pages.

Symbols

! operator, 1–4
? CLI macro, 2–217
?CID_PKT.PKT_ID offset, 2–58.7–2–58.25
?ID7 value, 2–410
* and ** symbols, v, xi
< and > symbols, v, xi

Numbers

16-bit process system calls (names of), 2–14

A

AC0–AC3, 1–2–1–4
access
 control list, 2–77, 2–174, 2–240, 2–245, 2–650, 2–882
 shared, 2–699
 to a protected file, permitting, 2–519
 to all devices
 disabling, 2–81
 enabling, 2–83
 to memory, read/write, 2–782
accumulators, 1–2, 1–5
ACK0 and ACK1 characters, 2–674
ACL
 changing a file's group, 2–240
 getting a file's group, 2–882
 getting a file's, 2–77, 2–174, 2–245, 2–650
acquiring
 a new resource, 2–328
 resource, 2–594
active
 group of windows, 2–574
 PIDs, returning, 2–217
Ada language, 1–10
address
 logical, 2–781

 request
 for consoles, 2–58.6
 for terminals, 2–58.6
ring base, 2–565
space
 logical, mapping a device into, 2–373
 remapping a process's, 2–353
?AEPR value, 2–618
Agent, changing the wiring characteristics, 2–18
?ALAU value, 2–618
?ALLOCATE system call, 2–15–2–20, 2–213
allocated blocks, reading, 2–23
allocating, disk blocks, 2–15
ANSI-standard terminal, 2–182
?AOPR value, 2–618
AOS/RT32
 and old file system, 2–205
 system calls names of, –3
AOS/VS
 and old file system, 2–205
 system resources system calls (names of), 2–12
AOS/VS II, and new file system, 2–205
?APND value, 2–408, 2–412
?ASEB value, 2–618
assembly language, iv, 1–1, x
 programming, 1–5–1–9
 sample program sets, iv, x
?ASSIGN system call, 2–17
attribute, permanent file, 2–653
?AUAL value, 2–618
autobaud matching, 2–198
auxiliary clock, 2–401
?AWENT value, 2–18
?AWIRE system call, 2–18
?AWUDS value, 2–18

B

- B operator, 1–4
- ?B32N offset, 2–20
- ?BADL and ?BADR offsets, 2–20
- base
 - address, ring, 2–565
 - current resource, 2–186
- BASIC language, 1–10
- baud, autobaud matching, 2–198
- baud rate, split baud, 2–197
- ?BBAC offset, 2–20
- ?BBAL offset, 2–20
- ?BBAN offset, 2–20
- ?BBLC offset, 2–20
- ?BLL offset, 2–20
- ?BLLN offset, 2–20
- BCC character, 2–674
- ?BCHN offset, 2–20
- ?BERR offset, 2–20
- bias factor values
 - getting, 2–176
 - setting, 2–655
- binary
 - mode, 2–40
 - synchronous communications line, opening a, 2–405
- bisecond, 2–74
- bit
 - setting a, 1–4
 - significant, 1–5
- ?BITM value, 2–376
- ?BLBB offset, 2–20
- ?BLKIO system call, 2–19–2–20, 2–213, 2–216, 2–596
- ?BLKPR system call, 2–26
- bloc, disk identification, 2–291
- block
 - output to a data channel line printer, 2–600–2–601
 - reading allocated, 2–23
- block I/O, 2–19, 2–183, 2–210
 - disk, 2–598
 - MCA, 2–599
 - opening a file for, 2–210, 2–405
 - performing, 2–19, 2–596
 - physical, 2–24–2–25, 2–210, 2–525, 2–592
 - reading, 2–596
 - tape, 2–599
 - writing, 2–596, 2–844
- blocking a process, 2–26
- ?BLTH value, 2–20
- ?BM32R value, 2–21
- ?BMAFE value, 2–21
- ?BMBI value, 2–115–2–116, 2–117
- BMC device, 2–151, 2–271
- ?BMDA value, 2–116
- ?BMDEV offset, 2–198
- ?BMDEV value, 2–191
- ?BMDIO value, 2–21
- ?BMEOR value, 2–21
- ?BMEP value, 2–115
- ?BMFO value, 2–117
- ?BMIO value, 2–21
- ?BMNAB value, 2–21
- ?BMNH and ?BM8B values, 2–115
- ?BMNMB value, 2–21
- ?BMNO value, 2–117
- ?BMNR value, 2–116
- ?BMOP value, 2–117
- ?BMPE value, 2–116
- ?BMPIO value, 2–21
- ?BMRA value, 2–117–2–118
- ?BMSH value, 2–116
- ?BMTB value, 2–115
- ?BMTI value, 2–117–2–118
- ?BMUC value, 2–117
- ?BNAME system call, 2–28
- BOOMER.SR sample program, A–2, A–32–A–36
- boot clock, SCP, 2–401
- ?BPEL offset, 2–20
- ?BPER offset, 2–20
- ?BPVB offset, 2–20
- ?BROBIT–?BR4BIT, baud rate offsets, 2–196
- BRAC0–BRAC3 status words, 2–30
- BRAN utility program, 2–382

- break
 - connection, 2-516, 2-523
 - customer/server relationship, 2-80, 2-92
 - file
 - creating a, 2-29
 - disabling a, 2-95
 - enabling a, 2-94
 - line, 2-40
 - sequences, **2-198**
- ?BRFCT offset, 2-197
- BRFP status word, 2-30
- ?BRKFL system call, 1-2, 2-29-2-30
- BRPC status word, 2-30
- BRSB status word, 2-30
- BRSL status word, 2-30
- BRSP status word, 2-30
- BRTID status word, 2-30
- BSC
 - error statistics, 2-686
 - line
 - device name for, 2-669
 - enabling a, 2-669
 - receiving information over a, 2-709
 - sending information over a, 2-720
 - protocol data-link control characters, 2-674
- ?BSTS offset, 2-20
- ?BTBC offset, 2-20
- ?BTBL offset, 2-20
- buffer
 - mode of tape I/O, 2-214, 2-412
 - moving bytes from a customer, 2-377
 - moving bytes to a customer, 2-379
- byte pointer, 1-5

C

- C language, 1-10
- ?C16B offset, 2-198
- ?C8BT value, 2-178, 2-181
- ?CABD offset, 2-198
- ?CACC offset, 2-198
- ?CACP offset, 2-62-2-68
 - example of, 1-7
- calendar, system, 2-660
- calling resource, 2-328
- ?CALLOUT offset, 2-198

- calls, system, 1-1
- canceling, character device, 2-82
- carriage return character, 2-663
- cascaded virtual timer, 2-794
- ?CBK0 offset, 2-198
- ?CBK1 offset, 2-198
- ?CBK2 offset, 2-198
- ?CCPS offset, 2-66-2-68, 2-168
- ?CCTD offset, 2-198
- ?CCTYPE offset, 2-199
- ?CDAY system call, 2-31
- ?CDEH offset, 2-66-2-68
- ?CDEL offset, 2-66-2-68
- ?CDRXON value, 2-40-2-41
- ?CDSBRK value, 2-40, 2-41
- ?CDT0-?CDT3 values, 2-179, **2-181**, 2-195
- ?CEB0 and ?CEB1 values, 2-179
- ?CEOC value, 2-179, 2-195
- ?CEOL value, 2-178
- ?CEOS value, 2-179, 2-194
- ?CEPI value, 2-178
- ?CESC value, 2-179
- ?CFF value, 2-178
- ?CFKT value, 2-180, 2-614
- ?CFTYP offset, 2-61-2-68
 - example of, 1-6
- ?CGNAM system call, 2-32
- ?CH4 offset, 2-180
- CHAIN command, 2-255
- ?CHAIN system call, 2-33-2-34
- chaining
 - new procedure, 2-595
 - programs, 2-33
- changing
 - agent wiring characteristics, 2-18
 - priority of a process, 2-531
 - priority of a task, 2-280, 2-530
 - process type, 2-75
 - unshared memory pages, 2-384
 - user locality, 2-354
 - working directory, 2-89
- channel
 - closing a, 2-38
 - graphics output, 2-225
 - map, data, 2-729
 - number, 2-226, 2-405, 2-569
 - getting a file's complete pathname from, 2-32

- character device, 2-177, 2-190
 - assigning to a process, 2-17
 - canceling or deassigning a, 2-82
 - deassigning a, 2-82
- characteristics
 - another's, 2-191
 - changing the Agent's wiring, 2-18
 - current, 2-191
 - current and default, 2-190, 2-679
 - default, 2-191
 - device, 2-177
 - extended, 2-190
 - extended device, 2-679
 - owned, 2-191
 - packet parameters, 2-193
 - shared console, ownership, 2-198
 - special keys and device types, 2-200
- CHECK.F77 sample program, A-2, A-42
- ?CHFS offset, 2-64-2-65
 - example of, 1-6
- ?CID_PKT.CHAN_NUM offset, 2-58.7-2-58.25
- ?CID_PKT.CON_LEN offset, 2-58.7
- ?CID_PKT.CON_PTR offset, 2-58.7
- ?CID_PKT.RBUF_PTR offset, 2-58.7
- ?CID_PKT.RBUF_LEN offset, 2-58.7
- ?CID_PKT.RDATA_LEN offset, 2-58.7
- ?CID_PKT.USER_FLGS offset, 2-58.7
- ?CID_PKT_LEN offset, 2-58.7
- ?CID_RET_TYPES offset, session connection types, 2-58.9
- ?CINT value, 2-711, 2-723
- ?CKVOL system call, 2-35
- ?CL... offsets and values, 2-37
- class
 - assignments logical processor, 2-362
 - IDs, 2-36
 - matrix, 2-51
 - process, 2-514
 - scheduling
 - statistics
 - accumulating, 2-42
 - returning, 2-45
 - system calls (names of), 2-12
- ?CLASS system call, 2-36-2-56
- clear to send modem option, 2-180
- Clearing, LBUS interrupts, 2-272
- clearing
 - default access control list, 2-77
 - device, 2-40
 - execute-protection status, 2-141-2-142
- ?CLFP offset, 2-199
- ? CLI, macro, 2-217
- CLI
 - message, getting a, 2-250
 - message format, 2-255
 - program, 1-1
- CLI-format
 - command line, sending via ?PROC, 2-543
 - IPC message, 2-307
- CLI.PR, summary of, A-39
- clip rectangle, 2-223
- ?CLMAX value, 2-180, 2-190
- ?CLMIN value, 2-190
- ?CLMSK data field length mask, 2-197
- ?CLN5 offset, 2-197
- ?CLN6 offset, 2-197
- ?CLN7 offset, 2-197
- ?CLN8 offset, 2-197
- clock
 - auxiliary, 2-401
 - real-time, 2-313
 - SCP boot, 2-401
 - system, 2-187, 2-201, 2-258, 2-645, 2-732
- ?CLOSE system call, 2-38-2-56, 2-183
 - example of, A-23
- closing
 - channel, 2-38
 - file, 2-38, 2-183
 - shared-access file, 2-658
- ?CLRVD system call, 2-40-2-41
- ?CLS_ACC value, 2-44
- ?CLS_GET value, 2-44
- ?CLS_PKT... offsets and values, 2-43-2-44
- ?CLS_SCHED value, 2-44
- ?CLS_SET value, 2-44
- ?CLSCHEM system call, 2-42-2-43
- ?CLST... offsets and values, 2-47-2-50
- ?CLSTAT system call, 2-45-2-50
- ?CLT0 offset, 2-197
- ?CLT1 offset, 2-197
- ?CLTH value, 2-62, 2-64, 2-66
 - example of, 1-7

?CMAT_... offsets and values, 2-53-2-55

?CMATRIX system call, 2-51

?CMDOP offset, 2-198

?CMIL offset, 2-64-2-68
example of, 1-7

?CMOD value, 2-179, 2-181, 2-657, 2-680

?CMPLT value, 2-59

?CMRI value, 2-178, 2-181, 2-657, 2-680

?CMRS offset, 2-64-2-66
example of, 1-7

?CMSH offset, 2-64-2-65
example of, 1-7

?CNAS value, 2-178, 2-181-2-182

?CNLX function, 2-192

?CNNL value, 2-180, 2-181, 2-680

?CNRM value, 2-179, 2-682

COBOL language, 1-10

CODE, MASM macro, 2-101

codes
and text error, returning, 2-683
error, 1-9
exception, 1-2
parametric, 1-3

command line (CLI-format), sending via
?PROC, 2-543

Commands, format conventions, xi

communication, operator/current process, 2-437

complete pathname
getting a, 2-32, 2-205
returning generic file, 2-239

COMSWITCH device, 2-40

?CON system call, 2-56, 2-160

?CON_PKT.USER_FLGS offset, input values,
2-58.8

conditional I/O, 2-220, 2-707

?CONFIG system call, 2-58-2-58.5

?CONFIG_... function codes, 2-58.1-2-58.5

?CONFIG_... offsets and values, for current
device route, 2-58.3

?CONFIG_RESET_... contents and values,
channel rerouting, 2-58.4

?CONINFO
and connection errors, 2-58.9
for TCONS, 2-58.9
and Permanent Virtual Circuits (PVCs),
2-58.9
and teletype connections, 2-58.9
and Telnet connections, 2-58.9
and Xerox Network Services (XNS),
connections, 2-58.9
?CON_CON_... offsets, 2-58.12-2-58.13
?CON_ITC_... offsets, 2-58.14-2-58.15
?CON_PVC_... offsets, 2-58.16-2-58.17
subsockets, 2-58.18-2-58.25
?CON_TCP_... offsets, 2-58.10
?CON_TNET_... offsets, 2-58.13-2-58.14
?CON_TSC_... offsets, 2-58.15-2-58.16
?CON_XNS_... offsets, 2-58.11-2-58.12
console line numbers, 2-58.8
console types, 2-58.9
return packet types, 2-58.8
TermServer console, 2-58.9

?CONFINFO packet contents, 2-58.7

?CONINFO system call, 2-58.6-2-58.25

connect time-out, 2-677

connection
breaking, passing, and re-establishing a,
2-516, 2-523
management, 2-57
message, 2-295
system calls (names of), 2-11, 2-13
session types, 2-58.9

connections, Xerox Network Services (XNS),
2-58.9

console
address request, 2-58.6
line numbers, 2-58.8
types, 2-199

construction, program, 1-8

?CONT value, 2-711, 2-716, 2-723

Contacting Data General, xii

contacting Data General, vi

continue/receive call, 2-714

control
characters (data-link), BSC protocol, 2-674
passing from one program to another, 2-33
point directory
?CREATE packet for, 2-64
maximum size of, 2-58.26
station, multipoint, 2-715

control characters, effect
erase line, 2-200
move left, 2-200
move right, 2-200
rubout echo, 2-200

CONTROL @EXEC family of commands, 2-438

control keys, characteristics, 2-200

control-character terminal interrupt
disabling, 2-333

- re-enabling, 2-334
- controller
 - intelligent, 2-190
 - status word, 2-528-2-529
- converting
 - date to a scalar value, 2-143
 - scalar date value, 2-31
 - scalar time value, 2-74
 - time of day to a scalar value, 2-171
- ?COTT value, 2-178
- ?CPBN value, 2-179, 2-181
- ?CPEN offset, 2-197
- CPI/24 device, 2-41
- ?CPM value, 2-179, 2-181
- ?CPMAX system call, **2-58.26**, 2-213
- ?CPMCN offset, 2-59
- ?CPMFW offset, 2-59
- ?CPMHS offset, 2-59
- ?CPMLS offset, 2-59
- ?CPMSK parity mask, 2-197
- ?CPOR offset, 2-62-2-63
- ?CPR1 offset, 2-197
- ?CPR2 offset, 2-197
- ?CPR0 offset, 2-197
- ?CPTY offset, 2-197
- CPU device, 2-151, 2-271
- CR character, 2-663
- ?CR110 offset, 2-196
- ?CR12H offset, 2-196
- ?CR134 offset, 2-196
- ?CR150 offset, 2-196
- ?CR18H offset, 2-196
- ?CR19K offset, 2-196
- ?CR20H offset, 2-196
- ?CR24H offset, 2-196
- ?CR300 offset, 2-196
- ?CR36H offset, 2-196
- ?CR38K offset, 2-196
- ?CR45 offset, 2-196
- ?CR48H offset, 2-196
- ?CR50 offset, 2-196
- ?CR600 offset, 2-196
- ?CR72H offset, 2-196
- ?CR75 offset, 2-196
- ?CR96H offset, 2-196
- ?CRAC value, 2-178
- ?CRAF value, 2-178
- ?CRAT value, 2-178
- ?CREATE system call, 1-1, 1-5, **2-60-2-68**, 2-168
 - example of, 1-6
- CREATE_WINDOW.SR sample program, A-2, **A-46-A-55**
- creating
 - break file, 2-29
 - directory or file, 2-60, 2-405, 2-849
 - dump file, 2-381
 - label for diskette or magnetic tape, 2-336
 - logical processor, 2-365
 - operator interface, 2-424
 - pipe file, 2-68
 - process, 2-534
 - queued task manager, 2-294
 - user data area, 2-70
- ?CRT1-?CRT15 values, 2-181
- ?CRUDA system call, **2-70**, 2-213
- ?CS10 offset, 2-197
- ?CS15 offset, 2-197
- ?CS20 offset, 2-197
- ?CSBDS value, 2-680
- ?CSBEN value, 2-680
- ?CSFF value, 2-178
- ?CSMK stop bit mask, 2-197
- ?CSPO value, 2-178
- ?CSRDS offset, 2-198
- ?CST value, 2-178
- ?CTCC offset, 2-199
- ?CTCD offset, 2-199
- ?CTDW offset, 2-199
- ?CTERM system call, 2-72-2-73
- ?CTHC offset, 2-199
- ?CTIM offset, 2-62-2-68
 - example of, 1-7
- ?CTLT offset, 2-199
- ?CTO value, 2-179, 2-734
- ?CTOD system call, 2-74
- ?CTSP value, 2-179, 2-181
- ?CTYPE system call, 2-75-2-76

- ?CUCO value, 2-178
- ?CULC value, 2-179, 2-181
- current
 - characteristics, 2-190, 2-679
 - date, 2-187, 2-313
 - process/operator process communication, 2-437
 - resource, base of the, 2-186
 - time, 2-313
- cursor hotspot, 2-573
- customer, 2-56
 - buffer
 - moving bytes from a, 2-377
 - moving bytes to a, 2-379
 - process, terminating a, 2-72
 - verifying a, 2-786, 2-789
- customer/server relationship, 2-34, 2-57
 - breaking a, 2-80, 2-92
 - terminating a, 2-72
- ?CWIN offset, 2-198
- ?CWIN value, 2-191
- ?CWRP value, 2-179
- ?CXLT offset, 2-198
- ?CXLT sets DG ANSI mode, 2-191

D

- ?DAC2 offset, 2-91, 2-748-2-749
- ?DACL system call, 1-2, 2-77-2-78
- ?DADID system call, 2-79
- data, password encryption, 2-590
- data channel
 - line printer, 2-70, 2-600-2-602
 - map, 2-729
- data compression
 - and native mode, 2-215
 - magnetic tape, 2-215
- Data Encryption Standard, 2-590
- data field length, mask, 2-197
- Data General, contacting, vi, xii
- data-link control characters, BSC protocol, 2-674
- data-sensitive files, default delimiters for, 2-663
- date
 - converting to a scalar value, 2-143
 - current, 2-187, 2-313

- last accessed value, 2-214
 - setting the, 2-400
 - value, converting a scalar, 2-31
- day
 - current, 2-187
 - time of, 2-258
- ?DCC offset, 2-91, 2-751-2-752
- DCH device, 2-151, 2-271
- ?DCI offset, 2-91, 2-751-2-752
- ?DCL2 offset, 2-748
- ?DCON system call, 2-80
- ?DDIS system call, 2-81
- dead space on a tablet, 2-572
- ?DEASSIGN system call, 2-82
- deassigning, character device, 2-82
- ?DEBL system call, 2-83
- ?DEBUG system call, 2-84
 - example of, A-16
- debugger, symbolic utility program, 1-3, 2-84
- debugging
 - program, 2-29-2-30
 - system calls (names of), 2-7
- decimal number specification, v, xi
- default
 - access control list, 2-77
 - characteristics, 2-190, 2-679
- defining
 - kill-processing routine, 2-330
 - poll-address pair, 2-664
 - polling list, 2-664
 - select-address pair, 2-664
 - terminal interrupt task, 2-293
 - user device, 2-269
 - fast, 2-149
- definition table, map, 2-153, 2-154, 2-272, 2-731
- ?DELAY system call, 2-85
- delaying, task, 2-85, 2-809
- ?DELETE system call, 2-86, 2-213
 - example of, A-19
- deleting
 - directory or file, 2-86
 - logical processor, 2-367
- delimiter table, 2-188, 2-416, 2-662
 - getting a, 2-188
 - setting a, 2-662
- delimiters for data-sensitive files, default, 2-663

density values, tape, 2-337

DES, 2-590

descriptor information, file, 2-780

device

- assigning to a process, 2-17
- character, 2-177, 2-190
- characteristics
 - commonly used, 2-180-2-202
 - complete, **2-194-2-199**
 - extended, 2-679
 - reading, 2-177
- clearing a, 2-40
- control table, 2-151, 2-271
- driver routine, 2-150, 2-270
- fast user, 2-149
- interrupt handler service routine, 2-156, 2-275
- mapping into logical address space, 2-373
- powerfail/restart routine, user, 2-277
- termination routine, 2-272
 - user, 2-276
- time-out value, 2-733-2-735
- user, 2-18, 2-269, 2-304.9

device reset, pending status, 2-58

devices

- disabling access to all, 2-81
- enabling access to all, 2-83

?DFLGS offset, **2-91**, 2-748-2-749

?DFLRC value, 2-749

?DFRSCH system call, 2-88

DG/VIEW windowing user interface, 2-828

DIB, 2-291

?DID offset, **2-91**, 2-748, 2-749

digitize option for a tablet, 2-572, 2-574, 2-579

?DIR system call, 1-5, 2-89

- examples of, 1-6, A-42

DIRCREATE.F77 sample program, A-2, **A-41-A-45**

DIRCREATE.SR sample program, 1-5-1-9

direct-access vertical forms control unit, 2-600-2-601

directory

- changing the working, 2-89
- creating a, 2-60
- deleting a, 2-86
- entries, 2-207
- file
 - ?CREATE packet for, 2-64
 - getting status of, 2-168-2-169
- filenames, 2-207

disabling

- access to all devices, 2-81
- break file, 2-95
- BSC line, 2-661
- class scheduling, 2-42
- control-character terminal interrupt, 2-333
- LEF mode, 2-350
- relative terminal, 2-667
- task rescheduling, 2-88
- task scheduling, 2-93
- terminal interrupt, 2-403

disconnecting, customer/server relationship, 2-80, 2-92

disk

- block I/O, 2-598
- blocks, allocating, 2-15
- identification bloc, 2-291
- initialized logical, 2-630
- logical, 2-630
- logical, initializing an extended, 2-886
- unit image, synchronized logical, 2-887

Disk Jockey utility, 2-213

diskette label, 2-336

display MRC routes, current, 2-58

DLCC, 2-674

DLE character, 2-674

DLE EOT characters, 2-674

DLIST.SR sample program, A-2, **A-26-A-27**

?DLNK offset, **2-91**, 2-748, 2-749

?DLNKB offset, **2-91**, 2-748, 2-749

?DLNKBL offset, 2-748

?DLNKL offset, 2-91

?DLNL offset, 2-748

?DNUM offset, **2-91**, 2-748, 2-749

Document sets, ix

documentation, related, iv, x

?DPC offset, **2-91**, 2-748, 2-749

?DPCL offset, 2-748

?DPRI offset, **2-91**, 2-748, 2-749

?DQTSK system call, 2-90

?DRCON system call, 2-92

?DRES offset, **2-91**, 2-748-2-749

?DRSCH system call, 2-93

DRT device, 2-41, 2-151, 2-271

?DSCH value, 2-88

?DSFLT offset, **2-91**, 2-748-2-749

?DSH offset, **2-91**, 2-751-2-752

?DSLTH value, 2-748
 ?DSMS offset, 2-91, 2-751-2-752
 DSR value, 2-677
 ?DSSL offset, 2-748-2-749
 ?DSSZ offset, 2-91, 2-748-2-749
 ?DSTB offset, 2-91, 2-748-2-749
 ?DSTL offset, 2-748
 DUART device, 2-151, 2-271
 DUMP CLI command, 2-110
 dump file, creating a, 2-381
 Dump Tool utility program, 2-382
 DUMP_II CLI command, 2-110
 dumping, a memory image, 2-381
 duplex, half, 2-198
 DVFU, 2-600
 .DWORD assembly language statement, 1-4
 ?DXLTH value, 2-91, 2-751

E

?EBAS value, 2-618
 ?EFPF offset, 2-606, 2-617
 ?EFLN offset, 2-606, 2-617
 ?EFMAX value, 2-606, 2-617
 ?EFNF offset, 2-606, 2-617
 ?EFTL offset, 2-606, 2-617
 ?EFTY offset, 2-606, 2-617
 ?ELAC offset, 2-419, 2-421
 ?ELCR offset, 2-419-2-420
 ?ELCT offset, 2-419-2-420
 ?ELFS offset, 2-419, 2-421
 ?ELGN offset, 2-419-2-420
 ?ELL1-?ELL3 values, 2-421
 ?ELLN value, 2-419
 ?ELRE offset, 2-419-2-420
 ?ELUH offset, 2-419, 2-421
 ?ELUT offset, 2-419, 2-421
 ?ELVL offset, 2-419-2-420
 ?ELVR offset, 2-419-2-420
 .ENABLE assembly language statement, 1-4

enabling
 access to all devices, 2-83
 break file, 2-94
 BSC line, 2-669
 class scheduling, 2-42
 LEF mode, 2-351
 multitask scheduling, 2-102
 terminal interrupt, 2-404
 ?ENBFL offset, 2-95-2-96
 ?ENBLN value, 2-96
 ?ENBRK system call, 2-94-2-96
 encryption, password, 2-554, 2-590
 ?ENCST value, 2-96
 end-of-volume, forcing on labeled tape, 2-148
 ?ENDIR value, 2-96
 ?ENESH offset, 2-96
 ?ENET offset, 2-406-2-407, 2-411,
 2-606-2-607
 ?ENEUS offset, 2-96
 ?ENFNP offset, 2-96
 ?ENID value, 2-672
 ?ENOV offset, 2-527
 ?ENOV value, 2-598
 ?ENPRE value, 2-96
 ENQ character, 2-675
 ?ENQUE system call, 2-98
 ?ENR4-?ENR7 values, 2-95-2-96
 ?ENSH value, 2-96
 ?ENSSH offset, 2-96
 ?ENSUS offset, 2-96
 entering
 event in the system log file, 2-360
 privilege state, 2-744
 Superprocess mode, 2-735
 Superuser mode, 2-737
 entry
 directory, 2-207
 link, 2-202
 ?ENUS value, 2-96
 environment, restoring the previous, 2-778
 EOT character, 2-675
 ?EPIP offset, 2-406-2-407, 2-411
 ?ERBA offset, 2-684
 ?ERCH offset, 2-684
 ?ERCS offset, 2-684
 ?ERLTH value, 2-684

ERMES file, 2-99-2-102

?ERMSG system call, 2-99-2-102

ERPFL pipe is full, 2-417

error

- codes, and text, 2-683
- message file, 2-99
- reporting, 1-9
- return, 1-2
- statistics, BSC, 2-686

error codes, 1-9

?ERSCH system call, 2-102

?ESBB value, 2-614

?ESBE value, 2-614

escape key, characteristics, and device types, 2-200

?ESCP value, 2-613

?ESCR offset, 2-606, 2-612

?ESDD value, 2-613

?ESED value, 2-613

?ESEP offset, 2-606, 2-612

?ESFC offset, 2-606, 2-612

?ESFF system call, 2-103, 2-213

?ESGT value, 2-614

?ESLN value, 2-606, 2-612

?ESNE value, 2-614

?ESNR value, 2-613

?ESPE value, 2-614

?ESRD value, 2-613

?ESRP value, 2-614

ESS, 2-284

?ESSE value, 2-613

ETB character, 2-675

?ETBB value, 2-716, 2-722

?ETER offset, 2-406-2-407, 2-411

?ETFL offset, 2-606

?ETFT offset, 2-407, 2-606

?ETLL offset, 2-606

?ETLT offset, 2-407, 2-411, 2-606

?ETMX value, 2-406-2-408

?ETSL offset, 2-606

?ETSN offset, 2-406-2-407, 2-411

?ETSP offset, 2-407, 2-605

ETX character, 2-675

?ETXB value, 2-716, 2-722

event codes

- in system log file, B-1
- special, B-1
- standard, B-1

?EXAC value, 2-781

examining

- class scheduling, 2-42
- default access control list, 2-77
- execute-protection status, 2-141-2-142
- privilege state, 2-744
- Superprocess mode, 2-735
- Superuser mode, 2-737

exception code, 1-2

exclusion bit map packet, 2-743

EXEC (CONTROL @EXEC) family of commands, 2-438

?EXEC functions

- backing up your files, 2-110
- batch processing, 2-112
- changing queuing parameters, 2-133
- dismounting a unit (extended request), 2-132
- dismounting unlabeled and labeled tapes, 2-109
- holding, unholding, canceling queue requests (AOS/VS), 2-127
- IPC print notification, 2-122
- mounting unlabeled and labeled tapes, 2-106-2-112
- obtaining
 - EXEC status information, 2-129
 - extended status information, 2-130
 - QDISPLAY information, 2-138
 - queue names, 2-136
- queuing a file entry, 2-112
- spooling output, 2-112
- submitting a job to a MOUNT queue, 2-131
- summary of, 2-104

?EXEC system call, 2-104, 2-438

EXEC utility program, 2-104

EXECUTE command, 2-255

execute-protection status, 2-141-2-142

execution

- path, task, 2-278
- program, 1-8

exiting

- from an interrupt service routine, 2-314
- from an overlay, 2-510

?EXPO system call, 2-141

extended

- characteristics, 2-190

device characteristics, 2-679
state save area, 2-284
status information about a process, 2-900
?EXTG pseudo-operation, A-17

F

F77BUILD_SYM program, A-40
?FAAB value, 2-245-2-246
?FACA value, 2-78, 2-210, 2-246, 2-520
?FACE value, 2-78, 2-210, 2-246, 2-520
?FACO value, 2-78, 2-210, 2-246, 2-520
?FACR value, 2-78, 2-210, 2-246, 2-520
factors, bias, 2-176, 2-655
?FACW value, 2-78, 2-210, 2-246, 2-520
?FAEA value, 2-169
?FAEB value, 2-245-2-246
?FAOB value, 2-245-2-246
?FARA value, 2-169
?FARB value, 2-245-2-246
fast user device, 2-149
father process, getting the PID of a, 2-79
?FAWB value, 2-245-2-246
?FBEX offset, 2-145, 2-147
?FBSTF offset, 2-145
?FCPC offset, 2-145, 2-146, 2-147
?FCPD file type, 2-61, 2-64-2-66, 2-415, 2-852
?FCPD value, 2-167
FCU, 2-600-2-601
?FDAY system call, 2-143
?FDBA offset, 2-145, 2-147
?FDBFZ offset, 2-146
?FDBL offset, 2-145, 2-147
?FDIR file type, 2-61, 2-64-2-66, 2-415, 2-852
?FDIR value, example of, 1-6-1-7
?FDLE value, 2-169
?FEDFUNC system call, 2-144-2-158
?FEOV system call, 2-148
?FEXPR value, 2-147
FF, 2-663
?FFCC file type, 2-61, 2-852

?FFLAG offset, 2-145-2-147
?FFLPT value, 2-147
?FGLT file type, 2-61, 2-852
?FIDEF system call, 2-149-2-156, 2-794, 2-803
 warnings about, 2-156
field translation, 2-616
FILCREATE.SR sample program, A-2, A-19-A-21
file
 attribute, permanent, 2-653
 block I/O, opening a, 2-210
 changing group ACL, 2-240
 closing a, 2-38, 2-183
 complete pathname of generic, 2-239
 creating a, 2-60, 2-405
 creation and management system calls (names of), 1-6
 creation options, 2-413
 deleting a, 2-86
 descriptor information, 2-780
 directory, getting status of, 2-168-2-169
 dump, 2-381
 error message, 2-99
 flushing to disk, 2-103
 generic, 2-206
 getting ACL, 2-77, 2-174, 2-245, 2-650
 getting group ACL, 2-882
 input/output system calls (names of), 2-7-2-8
 IPC, 2-211, 2-213, 2-288
 IPC, getting status of, 2-166
 opening a, 2-405
 opening for shared-access, 2-699
 other types, getting status of, 2-168
 pointer
 getting the position, 2-220
 positioning the, 2-610, 2-707
 protected, 2-519
 protected shared, 2-701
 recreating a, 2-629
 renaming a, 2-632
 shared, 2-659, 2-699
 access, opening a, 2-405
 flushing to disk, 2-103
 specifications word, 2-412
 status information, getting, 2-163
 symbol table, 2-261
file (continued)
 system log, 2-739
 truncating a, 2-259, 2-773
 unit, getting status of, 2-165
File Editor functions
 change radix, 2-144
 delete a temporary symbol, 2-147
 disassemble an instruction, 2-146

- evaluate a FED string, 2-145
- examining dump file, 2-381
- insert a temporary symbol, 2-146
- interfacing to, 2-144
- open symbol table file, 2-145
- filename
 - directory, 2-207
 - program (.PR), returning, 2-640
 - templates, 2-208
- FILESTATUS command, 2-214
- ?FINA offset, 2-146
- ?FINST value, 2-147
- ?FIPC file type, 2-61, 2-63, 2-66-2-67, 2-415, 2-852
- ?FIPC value, 2-211
- ?FIXMT system call, 2-157-2-158
- ?FLCC file type, 2-61, 2-852
- ?FLCHN offset, 2-161, 2-173
- ?FLCR value, 2-145
- ?FLDIS value, 2-146
- ?FLDU value, 2-167
- ?FLEFS value, 2-145
- ?FLEX offset, 2-145, 2-147
- ?FLLEN value, 2-161, 2-173
- ?FLNK file type, 2-61, 2-852
- floating-point status register, 2-285
- floating-point unit, initializing the, 2-285
- ?FLOCK system call, 2-159-2-160
- ?FLOG file type, 2-741
- ?FLOST value, 2-145
- flow control, hardware, 2-197
 - output, 2-197
- ?FLPID offset, 2-161, 2-173
- ?FLREV offset, 2-161, 2-173
- ?FLRSW offset, 2-161, 2-173
- ?FLSEL offset, 2-161, 2-173
- ?FLSYM value, 2-146, 2-147
- ?FLTY offset, 2-161, 2-173
- ?FLUSH system call, 2-162
- flushing
 - file descriptor information, 2-780
 - shared file memory pages to disk, 2-103
 - shared page to disk, 2-162
- ?FMDB value, 2-169
- ?FMEFS value, 2-147
- ?FNCC file type, 2-61, 2-852
- ?FNIR offset, 2-145
- ?FOCC file type, 2-61, 2-852
- forcing, end-of-volume on labeled tape, 2-148
- form feed character, 2-663
- Format conventions, xi
- Format conventions, v
- Forms Control Utility program, 2-600-2-601
- FORTRAN 77
 - language, 1-10-1-11
 - operating system interface sample program set, A-2-A-3, A-39
 - sample program set, iv, x
- ?FPIP file type, 2-61, 2-852
- ?FPRG file type, 2-61, 2-852
- ?FPRM value, 2-169
- ?FPRV file type, 2-61, 2-415, 2-852
- ?FQUE file type, 2-61, 2-852
- frame information, stack, 2-807
- frame pointer, 1-2
- ?FRCR value, 2-144
- ?FRDIS value, 2-144, 2-146
- ?FRDTS value, 2-144, 2-147
- ?FREFS value, 2-144, 2-145, 2-147
- frequency of the system clock, 2-201
 - getting the, 2-201
- ?FRESA offset, 2-145, 2-147
- ?FRESS offset, 2-145
- ?FRFNC offset, 2-145-2-147
- ?FRITS value, 2-144, 2-146
- ?FROST value, 2-144-2-145
- ?FRRR offset, 2-146
- ?FSDF file type, 2-61, 2-852
- ?FSHB value, 2-169
- ?FSNL offset, 2-146, 2-147
- ?FSNM offset, 2-146, 2-147
- ?FSPR file type, 2-61, 2-63, 2-66-2-67, 2-852
- ?FSTAT system call, 2-163-2-164, 2-214
- ?FSTF file type, 2-61, 2-852
- ?FSVAL offset, 2-146, 2-147
- ?FSVLL offset, 2-146, 2-147
- ?FTCK value, 2-161

?FTER value, 2-161
 ?FTEX value, 2-161
 ?FTOD system call, 2-171
 ?FTPN value, 2-161
 ?FTSH value, 2-161
 ?FTXT file type, 2-61, 2-415, 2-852
 ?FUDA value, 2-169
 ?FUDF file type, 2-61, 2-410, 2-415, 2-852
 ?FULA value, 2-173
 full process name, 2-265, 2-521
 ?FUNLOCK system call, 2-172-2-173
 ?FUNX file type, 2-61, 2-852
 ?FUPF file type, 2-61, 2-852
 ?FWFI offset, 2-146
 ?FWFL value, 2-161

G

?GACL system call, 2-174
 ?GARG value, 2-252, 2-254
 GATE.ARRAY.SR sample program, A-2, A-17
 ?GBIAS system call, 2-176
 ?GCFC value, 2-255
 ?GCHR system call, 2-335
See also The ?GECHR system call
 ?GCLOSE system call, 2-183-2-184, 2-214, 2-216, 2-412
 ?GCMD value, 2-252, 2-254, 2-255
 ?GCNT value, 2-252, 2-254
 ?GCPCN offset, 2-71, 2-87, 2-175, 2-632.2, 2-652, 2-654
 ?GCPFW offset, 2-71, 2-87, 2-175, 2-632.2, 2-652, 2-654
 ?GCPLT value, 2-71, 2-87, 2-175, 2-632.2, 2-652, 2-654
 ?GCPN system call, 2-185
 ?GCRB system call, 2-186
 ?GDAY system call, 2-187
 ?GDLC value, 2-253-2-255
 ?GDLM system call, 2-188-2-189
 ?GECHR system call, 2-190-2-192
 generic file, 2-206
 complete pathname of, 2-239

get/set class ID code, 2-36-2-56
 ?GFCF value, 2-252
 ?GHRZ system call, 2-201
 ?GLINK system call, 2-202
 ?GLIST system call, 2-203
 global port number
 and PID association, 2-219
 local port number with, 2-308
 modifying a ring field within a, 2-289
 returning a, 2-288
 ring field with, 2-308
 translate local to global equivalent, 2-770
 ?GMEM system call, 2-204
 ?GMES value, 2-252, 2-254, 2-255
 GMT, 2-247, 2-400
 ?GNAME system call, 2-205, 2-239
 ?GNFN system call, 1-1, 2-207-2-209
 example of, A-26
 ?GNUM offset, 2-251, 2-253, 2-255
 ?GOPEN system call, 2-168, 2-210-2-216
 example of, A-26
 ?GPID system call, 2-217
 ?GPORT system call, 2-219
 ?GPOS system call, 2-220-2-221
 ?GPRNM system call, 2-222
 ?GRAPH_CLOSE_PIXELMAP function, 2-224, 2-229
 ?GRAPH_CRE... offsets and values, 2-228
 ?GRAPH_CREATE_MEMORY_PIXELMAP function, 2-224-2-225, 2-227
 ?GRAPH_GET_DRAW_ORIGIN function, 2-224
 ?GRAPH_MAP... offsets and values, 2-233
 ?GRAPH_MAP_PIXELMAP function, 2-224, 2-233
 ?GRAPH_OPEN... offsets and values, 2-226-2-258
 ?GRAPH_OPEN_WINDOW_PIXELMAP function, 2-224, 2-226
 ?GRAPH_PIXELMAP_STATUS function, 2-224, 2-229
 ?GRAPH_PIXSTAT... offsets and values, 2-229-2-258
 ?GRAPH_PKT... offsets and values, 2-223, 2-225, 2-228-2-229, 2-234
 ?GRAPH_RDPAL... offsets and values, 2-236
 ?GRAPH_READ_PALETTE function, 2-224, 2-235-2-236

?GRAPH_RECT_STATE_DISABLE value, 2-230, **2-232**

?GRAPH_RECT_STATE_ENABLE value, 2-230, **2-232**

?GRAPH_SET_CLIP... offsets and values, 2-231-2-258

?GRAPH_SET_CLIP_RECTANGLE function, 2-224, 2-230

?GRAPH_SET_DRAW_ORIGIN function, 2-224

?GRAPH_UNMAP_PIXELMAP function, 2-224, 2-232, 2-234

?GRAPH_WRITE_PALETTE function, 2-224, **2-234**

?GRAPH_WRPAL... offsets and values, 2-235

graphics channel, output, 2-225

?GRAPHICS functions

- closing a pixel map, 2-229
- creating a pixel map in memory, 2-227
- getting the coordinates of the draw origin, 2-237-2-238
- getting the status of a pixel map, 2-229
- mapping a pixel map into a program's address space, 2-233
- opening a graphics window's pixel map, 2-225-2-226
- reading from a palette, 2-235-2-236
- setting the clip rectangle, 2-230
- setting the draw origin, 2-236
- transferring data between pixel maps and files, 2-232-2-233
- unmapping a pixel map from a program's address space, 2-234-2-235
- writing to a palette, 2-234

graphics output channel, 2-225

?GRAPHICS system call, 2-191, **2-223-2-238**
example of, A-58

graphics window, 2-226

?GRAPHICS_GET_DRAW_ORIGIN function, 2-237-2-238

?GRAPHICS_GET_ORG... offsets and values, 2-238

GRAPHICS_SAMPLE.SR sample program, A-2, **A-56-A-60**

?GRAPHICS_SET_DRAW_ORIGIN function, 2-236

?GRAPHICS_SET_ORG... offsets and values, 2-237

?GRCH offset, 2-649

Greenwich Mean Time, 2-400

?GREQ offset, 2-251, 2-252

?GRES offset, 2-251, 2-252, 2-254, 2-255

?GRIH offset, 2-649

?GRLTH value, 2-649

?GRNAME system call, 2-239

group

- access control list, 2-548, 2-883-2-885
- buffer, 2-241-2-242
- list, 2-884
- name defined, 2-548

?GROUP system call, 2-240, 2-548

?GROUP_... offsets and values, 2-241-2-242

GRP MASM macro, 2-101

?GRPH offset, 2-649

?GRRH offset, 2-649

?GSHPT system call, 2-243

?GSID system call, 2-244

?GSW offset, 2-251, 2-252, 2-255

?GSWS value, 2-253-2-255

?GTACP system call, 2-245-2-246

?GTIME system call, 2-247-2-249

?GTMES system call, **2-250**, 2-543
examples of, A-26, A-33, A-37

?GTNAM system call, 2-256-2-257

?GTOD system call, 2-258

?GTRUNCATE system call, 2-213, **2-259**

?GTSVL system call, 2-261-2-262

?GTSW value, 2-252, 2-253-2-255

?GUHBP offset, 2-264

?GUHFL offset, 2-264

?GUHFN offset, 2-264

?GUHID offset, 2-264

?GUHLN offset, 2-264

?GUHLR offset, 2-264

?GUHP0 value, 2-264

?GUHPI system call, 2-263

?GUID value, 2-264

?GUNM system call, 2-265

?GVPID system call, 2-266

H

handler service routine
device interrupt, 2-275

fast device interrupt, 2-156

?HAPH array offset, 2-287, 2-812

?HAPL array offset, 2-287, 2-812

?HARAY array offset, 2-287, 2-812

hardware processor identification, unique, 2-263

HEAR.SR sample program, A-1, A-3-A-4

hertz, definition of, 2-645

?HIBUF offset, 2-287

?HIEND offset, 2-287

high-level language interface, 1-1, 1-10-1-11

high-level language sample program set, iv, x

high-order bits, 1-5

?HIST offset, 2-287

histogram

- killing a, 2-329
- multiprocessor, 2-393
- starting a, 2-286, 2-393, 2-810
- uniprocessor, 2-393

?HIWDS offset, 2-287

?HNAME system call, 2-267-2-268

/HOFC switch, 2-180

host

- ID, 2-168, 2-217, 2-267
- local, 2-28
- remote, 2-28, 2-639

hostname, 2-267

hotspot, cursor, 2-573

?HPRH array offset, 2-287, 2-812

?HPRL array offset, 2-287, 2-812

?HRDFLC hardware flow control offset, 2-197

?HRDFLC value, 2-180

?HSBH array offset, 2-287, 2-812

?HSBL array offset, 2-287, 2-812

?HSIH array offset, 2-287, 2-812

?HSIL array offset, 2-287, 2-812

?HTTH array offset, 2-287, 2-812

?HTTL array offset, 2-287, 2-812

?HWBUF offset, 2-811

?HWEND offset, 2-811

?HWLTH value, 2-287, 2-811

?HWST offset, 2-811

?HWWDS offset, 2-811

I

I/O

- and MCA protocol, 2-527
- physical block, 2-525, 2-592

I/O

- and new file system, 2-210
- block, 2-19, 2-596
- conditional, 2-220, 2-707
- disk block, 2-598
- file, 2-220, 2-707
- MCA block, 2-599
- modified sector, 2-216
- physical block, 2-210
- reading and writing record, 2-604
- tape block, 2-599
- writing block or record, 2-844

?IBAD offset, 2-39, 2-221, 2-407, 2-410, 2-605, 2-609

?IBIN value, 2-408, 2-608, 2-611

?IBLT value, 2-406, 2-407, 2-606, 2-608

?ICH offset, 2-39, 2-221, 2-406-2-408, 2-605-2-606

?ICRF value, 2-408, 2-412, 2-608

ID

- host, 2-168, 2-217, 2-267
- pixel map, 2-226

?ID8 value, 2-410, 2-417

?ID16 value, 2-410, 2-417

?ID5 value, 2-410, 2-417

?ID6 value, 2-410, 2-417

?ID62 value, 2-410, 2-417

?ID7 value, 2-417

?IDAM value, 2-410, 2-417

?IDEF system call, 2-269-2-272

?IDEL offset, 2-39, 2-221, 2-406, 2-407, 2-411, 2-416, 2-605, 2-608, 2-610

identification, unique processor hardware, 2-263

identification bloc, disk, 2-291

identifier

- checking volume, 2-35
- host, 2-267
- system, 2-244, 2-719
- unique task, 2-688, 2-690, 2-777

?IDGOTO system call, 2-278

- ?IDKIL system call, 2-279
 - example of, A-29
- ?IDPH offset, 2-306, 2-310, 2-311
- ?IDPN offset, 2-296
- ?IDPRI system call, 2-280
- ?IDRDY system call, 2-281
- ?IDSTAT system call, 2-282
- ?IDSUS system call, 2-283
- ?IESS system call, 2-284
- ?IEXO value, 2-408, 2-412
- ?IFNBK value, 2-296, 2-311
- ?IFNP offset, 2-39, 2-221, 2-407, 2-411, 2-605, 2-610
- ?IFNSP value, 2-306, 2-311
- ?IFOP value, 2-609
- ?IFPR value, 2-296, 2-311
- ?IFPU system call, 2-285
- ?IFRFM value, 2-296, 2-311
- ?IFRING value, 2-296, 2-311
- ?IFSOV value, 2-296, 2-311
- ?IFSTM value, 2-306, 2-311
- ?IHIST system call, 2-286
- ?IIPC value, **2-409**, 2-611
- ?ILKUP system call, 2-288
 - examples of, A-4, A-7
- ?ILTH offset, 2-296, 2-306, 2-310, 2-311
- ?IMERGE system call, 2-289
- ?IMFF value, 2-409
- ?IMHN value, 2-609
- ?IMIO offset, 2-527
- ?IMIO value, 2-598
- ?IMNH value, 2-409
- ?IMP2 value, 2-406, 2-409, 2-609
- implicit system call, A-39
- ?IMRS offset, 2-39, 2-68, 2-221, 2-407, 2-410, 2-605, 2-609
 - for pipe size, 2-416
- ?IMSG system call, 2-290
- index levels, 2-64
- indicating, prior rescheduling state, 2-88
- ?INID value, 2-436
- ?INIT system call, **2-291**, 2-887
- initial IPC message, 2-250
- INITIALIZE CLI command, 2-292
- initializing
 - extended state save area, 2-284
 - floating-point unit, 2-285
 - job processor, 2-317
 - logical disk, 2-291, 2-630
 - logical disk (extended), 2-886
- initiating, a task, 2-747
- initiation queue, task, 2-294
- INRING.SR sample program, A-2, **A-16-A-18**
- intelligent
 - asynchronous controller, 2-17, 2-41, 2-271, 2-405, 2-534, 2-604
 - controller, 2-190
- ?INTEO value, 2-411
- interface
 - assembly language, 1-5-1-9
 - high-level language, 1-1
 - operator, 2-424
- internal time, returning the OS-format, 2-313
- interprocess communications system calls
 - (names of), 2-10
- interprocess signaling mechanism, 2-688, 2-690, 2-691, 2-845
- interrupt
 - control-character terminal, 2-333, 2-334
 - disabling terminal, 2-403
 - enabling terminal, 2-404
 - handler service routine
 - device, 2-275
 - fast device, 2-156
 - sequences, keyboard, 2-332
 - service message, 2-290
 - service routine, 2-151, 2-271
 - exiting from an, 2-314
 - transmitting a message from an, 2-157, 2-315
 - task, 2-278
 - terminal, 2-293, 2-335
- intertask message
 - receiving an, 2-627
 - receiving without waiting, 2-628
 - transmitting an, 2-898, 2-899
- ?INTWT system call, 2-293
- IOC device, 2-151, 2-271
- ?IOPH offset, 2-296
- ?IOPN offset, 2-306, 2-310, 2-311
- ?IOSZ value, 2-39, 2-221, 2-406, 2-407, 2-606, 2-608
- IPC
 - file, 2-211, 2-213, 2-288
 - getting status of, 2-166

- message, 2-219, 2-289
 - CLI-format, 2-307
 - receiving an, 2-295
 - sending an, 2-305
 - sending and then receiving an, 2-309
 - sending via ?PROC, 2-543
- ?IPKL value, 2-406, 2-408, 2-608
- ?IPLTH value, 2-296, 2-306
- ?IPRLTH value, 2-310
- ?IPST value, 2-608, 2-610, 2-707-2-708
- ?IPTR offset, 2-296, 2-300, 2-306, 2-310, 2-311
- ?IQTSK system call, 2-294
- ?IRCL offset, 2-39, 2-221, 2-407, 2-411, 2-605, 2-609, 2-707-2-708
- ?IREC system call, 1-2, **2-295-2-297**
 - example of, A-4
- ?IRES offset, 2-39, 2-221, 2-407, 2-410, 2-605, 2-609
- ?IRLR offset, 2-39, 2-221, 2-407, 2-411, 2-605, 2-610
- ?IRLT offset, 2-310, 2-311
- ?IRMV system call, 2-151, 2-276, **2-304.9**
- ?IRNH offset, 2-39, 2-221, 2-407, 2-411, 2-605, 2-610, 2-707-2-708
- ?IRNW offset, 2-39, 2-221, 2-407, 2-411, 2-605, 2-610
- ?IRPT offset, 2-310, 2-311
- ?IRSV offset, 2-310, 2-311
- ?IS.R system call, 2-309-2-312
- ISC device, 2-271
- ?ISEND system call, 2-305-2-306
 - example of, A-7
- ?ISFL offset, 2-296, 2-306, 2-310, 2-311, 2-543
- ?ISPLIT system call, 2-297, **2-308**

- ?ISTI offset, 2-39, 2-221, 2-406-2-410, 2-412, 2-413, 2-605, 2-608, 2-609, 2-707
- ?ISTO offset, 2-39, 2-221, 2-406, 2-407, 2-409, 2-410, 2-605, 2-609
- ISYS FORTRAN 77 function, 1-10
- ITB character, 2-676
- ?ITIME system call, 2-313
- ?IUFL offset, 2-73, 2-296, 2-297-2-304.8, 2-306, 2-310, 2-311, 2-543
 - example of, 2-297
- ?IXIT system call, 2-151, 2-276, 2-277, **2-314**
- ?IXMT system call, 1-2, 2-151, **2-315**

J

- job processor
 - getting the status of a, 2-324
 - initializing a, 2-317
 - moving to a new logical processor, 2-320
 - releasing a, 2-322
- ?JPI_PKT... offsets and values, 2-318-2-319
- ?JPID_MAX value, 2-319, 2-321, 2-323
- ?JPID_MIN value, 2-319, 2-321, 2-323
- ?JPINIT system call, 2-317-2-336
- JPLCS instruction, 2-319
- ?JPM_PKT... offsets and values, 2-321
- ?JPMOV system call, 2-320
- ?JPR_PKT... offsets and values, 2-323
- ?JPREL system call, 2-322-2-323
- ?JPS_GEN value, 2-325
- ?JPS_GEN... offsets and values, 2-326
- ?JPS_PKT... offsets and values, 2-325
- ?JPS_SPEC value, 2-325
- ?JPS_SPEC... offsets and values, 2-327
- ?JPSTAT system call, 2-324
- JPSTATUS instruction, 2-324

K

- Kanji character sets, 2-192
 - and VT100, 2-192
 - Japanese, 2-192
 - Taiwanese, 2-192
- ?KCALL system call, 2-328
- keyboard interrupt sequences, 2-332
- ?KHIST system call, 2-329
- ?KILAD system call, 2-330
- kill-processing routine, 2-279, 2-330
- ?KILL system call, 2-331
 - example of, A-33
- killing
 - histogram, 2-329
 - task, 2-279, 2-331, 2-510
 - tasks of a specified priority, 2-533
- ?KINTR system call, 2-332
- ?KIOFF system call, 2-333
- ?KION system call, 2-334
- ?KWAIT system call, 2-335

L

- ?LABEL system call, 2-336
- labeled
 - diskette, 2-336
 - magnetic tape, 2-35, 2-38, 2-336, 2-418
 - forcing end-of-volume, 2-148
 - trailer, 2-38
- LAC device, 2-41
- language
 - assembly, 1-1
 - interface, high-level, 1-1, 1-10-1-11
- Language Front-end Processor, options, 2-199
- ?LB8 value, 2-337, 2-338
- ?LB16 value, 2-338
- ?LB5 value, 2-338
- ?LB6 value, 2-338
- ?LB62 value, 2-338
- ?LB7 value, 2-338
- ?LBAC offset, 2-337-2-338
- ?LBAM value, 2-338
- ?LBDV offset, 2-337-2-338
- ?LBFG offset, 2-337-2-338

- ?LBIM value, 2-338
- ?LBLN value, 2-337
- ?LBMF value, 2-338
- ?LBMP value, 2-338
- ?LBMR value, 2-338
- ?LBMS value, 2-338
- ?LBOI offset, 2-337-2-338
- ?LBSC value, 2-338
- ?LBST offset, 2-337-2-338
- ?LBUV offset, 2-337-2-338
- ?LBVD offset, 2-337-2-338
- LCALL instruction, A-14
- ?LDMA event code, 2-361
- LDU images
 - initializing, 2-340
 - mirroring and synchronizing, 2-385
- ?LDU_IMAGE_HARDWARE_MIRRORED value, 2-345
- ?LDU_IMAGE_REMOVED value, 2-344
- ?LDU_MIRROR_BEING_SYNCHRONIZED value, 2-344
- ?LDU_MIRRORED value, 2-344
- ?LDU_PKT... offsets and values, 2-343-2-347
- ?LDU_PRIMARY_IMAGE value, 2-345
- ?LDUINFO system call, 2-340-2-350
- ?LDUINFO... offsets and values, 2-341, 2-345-2-346
- least significant bit, 1-5
- leaving
 - privilege state, 2-744
 - Superprocess mode, 2-735
 - Superuser mode, 2-737
- LEF mode, 2-81, 2-83, 2-350
 - disabling, 2-350
 - enabling, 2-351
 - status, returning, 2-352
- ?LEFD system call, 2-350
- ?LEFE system call, 2-351
- ?LEFS system call, 2-352
- ?LFOP value, 2-357
- line
 - break, 2-40
 - BSC
 - disabling a, 2-661
 - receiving information over a, 2-709
 - sending information over a, 2-720
 - printer, data channel, 2-70, 2-600-2-602

- link entry, 2–202
- Link utility program, 1–8
- listing
 - directory entries, 2–207
 - shared partition size, 2–243
 - unshared memory parameters, 2–383
- ?LMAP system call, 2–353
- ?LMAX event code, 2–361
- LOAD CLI command, 2–110
- LOAD_II CLI command, 2–110
- loading
 - overlay, 2–511
 - program file, 2–637
- ?LOC_... offsets and values, 2–355–2–356
- local
 - host, process or queue name, 2–28
 - port number, 2–219, 2–770
- locality
 - process, 2–514
 - scheduling matrix, class, 2–51
 - user, changing, 2–354
- ?LOCALITY system call, 2–354
- locating, process name, 2–28
- locking, an object, 2–159
- log file
 - ?GROUP entry, 2–241
 - system, 2–360, 2–739
 - system call, 2–357
- ?LOGCALLS system call, 2–357–2–359
- LOGCALLS utility program, 2–358
- ?LOGDREC value, 2–359
- ?LOGEV system call, 2–360
- ?LOGF16U value, 2–358
- logging, system calls, 2–357
- ?LOGHREC value, 2–358
- logical
 - address, 2–781
 - address space, mapping a device into, 2–373
 - disk
 - information, returning, 2–340
 - initialized, 2–340, 2–630
 - initializing a, 2–291
 - initializing a (extended), 2–886
 - unit image, synchronized, 2–887
 - processor
 - class assignments, 2–362
 - creating a, 2–365
 - deleting a, 2–367

- getting the status of a, 2–369
 - moving a job processor to, 2–320
 - shared memory, 2–243
- low-order bits, 1–5
- lower ring
 - loading and stopping, 2–642
 - mapping, 2–353
- ?LPC_PKT... offsets and values, 2–366
- ?LPCL_PKT... offsets and values, 2–363–2–364
- ?LPCLASS system call, 2–362–2–364
- ?LPCREA system call, 2–365–2–366
- ?LPD_PKT... offsets and values, 2–368
- ?LPDELE system call, 2–367–2–368
- ?LPID_MAX value, 2–319, 2–321, 2–365
- ?LPID_MIN value, 2–319, 2–321, 2–365
- ?LPS_FUNC_MAX value, 2–370
- ?LPS_FUNC_MIN value, 2–370
- ?LPS_GEN... offsets and values, 2–370
- ?LPS_PKT... offsets and values, 2–370
- ?LPS_SPEC... offsets and values, 2–370–2–372
- ?LPSTAT system call, 2–369–2–370
- ?LSMI event code, 2–361, 2–742
- ?LSTART value, 2–357
- ?LTSF event code, 2–742
- ?LUMAX value, 2–742
- ?LUMI event code, 2–361

M

- Macroassembler program, iv, 1–8, x
- magnetic tape, 2–417
 - data compression, 2–215
 - densities
 - absolute, 2–212
 - relative, 2–212
 - labeled, 2–35, 2–336, 2–418
 - unit, Model 6352, 2–212, 2–214, 2–338, 2–412
- magnetic tape drives, and native mode, 2–215
- maintaining, and creating an operator interface, 2–424
- manager, queued task, 2–294
- manipulating
 - pixel maps, 2–223
 - the system log file, 2–739
 - windows, 2–813
- map
 - data channel, 2–729

definition table, 2-153-2-154, 2-272, 2-731
 pixel, 2-223
 ?MAPDV system call, 2-373-2-375
 ?MAPDV_PKT_PKTID value, 2-374, 2-376
 mapping
 device into logical address space, 2-373
 lower ring, 2-353
 mask, bit, 1-4
 MASM, 1-5-1-9
 MASM.PR program, 1-8
 MASM.PS file, 1-8
 MASM_32CHAR.PS file, 1-9, A-54, A-74
 ?MAXIMAGES value, 2-895
 ?MBAH offset, 2-378, 2-380
 ?MBBC offset, 2-378, 2-380
 ?MBCH offset, 2-378, 2-380
 ?MBFC system call, 2-377-2-378
 ?MBID offset, 2-378, 2-380
 ?MBLTH value, 2-378, 2-380
 ?MBNHR value, 2-390
 ?MBNLD value, 2-390, 2-391
 ?MBOOP value, 2-390
 ?MBTC system call, 2-379-2-380
 ?MBTRP value, 2-390
 ?MBWAIT value, 2-390
 MCA
 block I/O, 2-599
 unit, 2-210
 MCA protocol, with I/O, 2-527
 ?MCOBIT value, 2-56
 MCP1 device, 2-41
 ?MCPID value, 2-56
 ?MCRNG value, 2-56
 ?MDAC offset, 2-375, 2-376
 ?MDAL offset, 2-375, 2-376
 ?MDCL offset, 2-374, 2-375, 2-376
 ?MDDL offset, 2-374, 2-375, 2-376
 ?MDDT offset, 2-374, 2-375, 2-376
 ?MDID offset, 2-375, 2-376
 ?MDIL offset, 2-374, 2-375, 2-376
 ?MDLA offset, 2-374, 2-375, 2-376
 ?MDLL offset, 2-375, 2-376
 ?MDN0-?MDN1 offsets, 2-374-2-376
 ?MDNL offset, 2-375, 2-376
 ?MDNP offset, 2-375, 2-376
 ?MDOP offset, 2-374, 2-375, 2-376
 ?MDOX offset, 2-375, 2-376
 ?MDP0 value, 2-375
 ?MDPC offset, 2-374, 2-375, 2-376
 ?MDPK offset, 2-374, 2-375, 2-376
 ?MDPL offset, 2-374, 2-375, 2-376
 ?MDPV value, 2-373
 ?MDRE offset, 2-375, 2-376
 ?MDRL offset, 2-374, 2-375, 2-376
 ?MDRP offset, 2-374, 2-375, 2-376
 ?MDRT offset, 2-374, 2-375, 2-376
 ?MDUMP system call, 2-381-2-382, 2-471
 ?MEM system call, 2-383
 ?MEMI system call, 2-384
 memory
 address, 1-1
 dump, 2-471
 image, dumping a, 2-381
 logical shared, 2-243
 management system calls (names of), 2-3
 mapped device, 2-373-2-374
 pages
 changing (unshared), 2-384
 flushing (shared file) to disk, 2-103
 undedicated, 2-204
 parameters (unshared), listing, 2-383
 read/write access to, 2-782
 meridian, prime, 2-247, 2-400
 message
 CLI, 2-250
 error file, 2-99
 initial IPC, 2-250
 IPC, 2-219, 2-295, 2-305
 task, 2-771
 terminal, 2-681
 ?MFBRK value, 2-390
 ?MFSYM value, 2-390
 microcode for a job processor, 2-318
 ?MIFUN offset, 2-389-2-390
 ?MII1 and ?MII2 values, 2-390
 ?MIID offset, 2-389, 2-391
 ?MILD offset, 2-389, 2-391
 ?MIOP offset, 2-389-2-390
 ?MIPHI offset, 2-389-2-390

?MIPLO offset, 2-389-2-390
 ?MIPUL offset, **2-389**, 2-391
 ?MIR1-?MIR4 offsets, 2-389-2-391
 ?MIREs offset, 2-389-2-390
 ?MIRROR system call, 2-385-2-389
 ?MIRROR_... offsets and values, 2-386-2-388
 mirroring, LDU images, 2-385, 2-886-2-897
 ?MMAP value, 2-374, 2-376
 mode, binary, 2-40
 Model 6236-6240 disks, 2-25
 Model 6352 magnetic tape unit, 2-212, 2-214,
 2-338, **2-412**
 modem
 carrier detect, 2-198
 connection, timing, 2-199
 hardware input flow control, 2-198
 options, **2-198**
 user access, 2-198
 modem support, 2-180
 modified sector I/O, 2-20, 2-216
 modifying, ring field within a global port
 number, 2-289
 month, current, 2-187
 most significant bit, 1-5
 MOUNT command, 2-210
 mouse movement, 2-191
 moving
 bytes from a customer buffer, 2-377
 bytes to a customer buffer, 2-379
 job processor to a new logical processor, 2-320
 ?MPH_... offsets and values, 2-393-2-398
 ?MPHIST system call, 2-393-2-395
 ?MPHIST_... offsets, 2-396-2-398
 MRC device routes
 current, 2-58
 diverted, 2-58
 primary, 2-58
 secondary, 2-58
 ?MRDO value, 2-374, 2-376
 multipoint control station, 2-715
 multipoint tributary station, 2-716
 multiprocessor histogram, 2-393
 multiprocessor management system calls
 (names of), 2-11, 2-13

multitask scheduling, enabling, 2-102
 multitasking system calls (names of), 2-9-2-10
 ?MXFN value, 2-345, 2-895
 ?MXHN value, 2-267
 ?MXLPN value, 2-63
 ?MXPL value, 2-202, 2-203, 2-641, 2-694,
 2-695
 ?MXUN value, 2-265, 2-908
 ?MYTID system call, 2-399
 example of, A-29

N

NAK character, 2-676
 name, full process, 2-265, 2-521
 National Bureau of Standards, 2-590
 NBS, 2-590
 new file system, 2-205
 New Line character, 2-663
 NEWTASK.SR sample program, A-2,
A-29-A-31
 ?NFKY offset, 2-208
 ?NFLN value, 2-208
 ?NFSM offset, 2-208
 ?NFRS offset, 2-208
 ?NFTP offset, 2-208
 NL, 2-663
 normal return, 1-2
 ?NPAL offset, 2-218
 ?NPAP offset, 2-218
 ?NPFW offset, 2-218
 ?NPKEY offset, 2-218
 ?NPLTH value, 2-218
 ?NPEN offset, 2-218
 ?NPNUM offset, 2-218
 ?NPPR offset, 2-218
 ?NPRS1 offset, 2-218
 ?NTIME system call, 2-400
 ?NTRN value, 2-722
 null character, 1-10, 2-663
 number
 channel, 2-226, 2-569
 global port, 2-219, 2-288, 2-289, 2-770
 local port, 2-219, 2-770
 window ID, 2-226, 2-569

number specification
decimal, v, xi
octal, v, xi

O

?OBBQ value, 2-426
?OBCE value, 2-426
?OBCE value, 2-409, 2-410, 2-413
?OBCO value, 2-426
?OBGM value, 2-426
?OBHD value, 2-426
obituary message, 2-298
object
locking an, 2-159
unlocking an, 2-172
?OBLD value, 2-426
?OBLT value, 2-426
?OBMI value, 2-426
?OBPR value, 2-426
?OBQU value, 2-426
?OBUD value, 2-426
?OBUT value, 2-426
?OCIL offset, 2-432-2-433
?OCOL offset, 2-432-2-433
?OCR1-?OCR9 offsets, 2-432-2-433
?OCRD offset, 2-432-2-433
octal number specification, v, xi
?ODBS value, 2-215
?ODBY offset, 2-215, 2-216
?ODF1 offset, 2-211, 2-215, 2-216
?ODHD offset, 2-215, 2-216
?ODHS value, 2-215
?ODIS system call, 2-403
?ODMB value, 2-215
?ODND value, 2-215
?ODP0 value, 2-215
?ODSEC offset, 2-215, 2-216
?ODST value, 2-215
?ODTEO value, 2-215
?ODTL value, 2-210, 2-211, 2-213, **2-214**,
2-215, 2-259
?ODTP value, 2-215
?ODTRK offset, 2-215, 2-216
?OEBL system call, 2-404
?OFCE value, 2-409, 2-410, 2-413
?OFCE value, 2-409, 2-410, 2-413
?OFE2 offset, 2-428-2-429
?OFEI offset, 2-428-2-429
?OFEO offset, 2-428-2-429
?OFER offset, 2-428-2-429
offset, 1-3
?OFID value, 2-429, 2-430
?OFIN value, 2-409, 2-414
?OFIO value, 2-409, 2-414
?OFOT value, 2-409, 2-414
?OIGB value, 2-436
?OIGN value, 2-436
?OIIL offset, 2-435-2-436
?OIN2 offset, 2-435-2-436
?OIND offset, 2-435-2-436
?OINL value, 2-435
?OINP offset, 2-435-2-436
?OINR offset, 2-435-2-436
?OINT offset, 2-435-2-436
?OIOL offset, 2-435-2-436
old file system, 2-205
?OMBFM value, 2-410
?OMSTR value, 2-410
?ONE2 offset, 2-427
?ONEI offset, **2-427**, 2-428
?ONEO offset, **2-427**, 2-428
?ONER offset, 2-427
?ONID value, 2-427
?OOF2 and ?OOF3 offsets, 2-428-2-429
?OOFE offset, 2-428-2-429
?OOFL value, 2-428
?OOFN offset, 2-428-2-429
?OOFN offset, 2-428-2-429
?OOFN offset, 2-428-2-429
?OOG2 offset, 2-427
?OOGT offset, 2-427
?OON2 and ?OON3 offsets, 2-427
?OONE offset, 2-427
?OONL value, 2-427

- ?OONN offset, 2-427
- ?OONT offset, 2-427
- ?OPAM value, 2-35, 2-212-2-214
- ?OPCH offset, 2-211, 2-212
- ?OPD0-?OPD2 values, 2-108
- ?OPDH value, 2-35, 2-212, 2-213
- ?OPDL value, 2-35, 2-212, 2-213
- ?OPDM value, 2-35, 2-212, 2-213
- ?OPEH offset, 2-213
- ?OPEN system call, 1-3, 2-160, **2-405**
 - examples of, A-3, A-11, A-14, A-16, A-19, A-22, A-26, A-29, A-33, A-58
- opening
 - file, 2-405
 - file for block I/O, 2-210
 - file for shared access, 2-699
 - protected shared file, 2-701
- ?OPER functions
 - ?OPINFO, 2-425, 2-435-2-436
 - ?OPOFF, 2-425, 2-428-2-429
 - ?OPON, 2-425, 2-427
 - ?OPRCV, 2-425, 2-432-2-433
 - ?OPRESP, 2-425, 2-434
 - ?OPSEND, 2-425, 2-432
- ?OPER system call, ~~2-424~~ **2-425**, 2-438
- operating system, getting information, 2-692-2-692a
- operator
 - !, B, and S, 1-4
 - interface, 2-424
 - process/current process communication, 2-437
- ?OPEW offset, 2-213
- ?OPEX commands, 2-476
 - access, 2-442
 - align, 2-443
 - allocate, 2-443
 - batch_list, 2-444
 - batch_output, 2-445
 - binary, 2-446
 - brief, 2-447
 - cancel, 2-448
 - close, 2-448
 - consolestatus, 2-449-2-450
 - continue, 2-451
 - CPL, 2-452
 - create, 2-453
 - defaultforms, 2-454
 - delete, 2-454
 - disable, 2-455-2-456
 - dismounted, 2-456
 - elongate, 2-457
 - enable, 2-458-2-459
 - even, 2-460
 - flush, 2-461
 - font, 2-461
 - forms, 2-462
 - halt, 2-463
 - headers, 2-464
 - hold, 2-465
 - limit, 2-466
 - logging, 2-467
 - lpp, 2-469
 - mapper, 2-470
 - mdump, 2-471
 - message, 2-471
 - modify, 2-471
 - mounted, 2-472
 - mountstatus, 2-473
 - operator, 2-476
 - pause, 2-477
 - premount, 2-478
 - priority, 2-479-2-480
 - prompts, 2-480
 - purge, 2-480
 - qpriority, 2-481-2-482
 - refused, 2-483
 - release, 2-484
 - restart, 2-485
 - silence, 2-486
 - spoolstatus, 2-487-2-490
 - stack, 2-490-2-492
 - start, 2-492-2-494
 - status, 2-494-2-498
 - stop, 2-499
 - terminate, 2-499
 - trailers, 2-500
 - unhold, 2-501
 - unitstatus, 2-502-2-504
 - unlimit, 2-504
 - unsilence, 2-505
 - verbose, 2-508
 - xbias, 2-508
- ?OPEX system call, 2-437-2-472
- ?OPFC offset, 2-168, 2-213
- ?OPFL offset, 2-212, 2-213, 2-214
- ?OPIL offset, 2-434
- ?OPINFO function, 2-425
- ?OPK2 offset, 2-425-2-426
- ?OPKT offset, 2-425-2-426
- ?OPLT value, 2-211, 2-213
- ?OPMBF value, 2-212
- ?OPMD value, 2-212
- ?OPME value, 2-212, 2-213

- ?OPMST value, 2–212
- ?OPNL value, 2–425
- ?OPOFF function, 2–425
- ?OPON function, 2–425
- ?OPPH offset, 2–211
- ?OPRCV function, 2–425
- ?OPRESP function, 2–425
- ?OPSEND function, 2–425
- ?OPSP offset, 2–425–2–426
- ?OPTY offset, 2–211, 2–213
- ?OPXL value, 2–215, 2–216
- ?OPXP bit, 2–214
- ?OPXP value, 2–212
- ?OPXS offset, 2–215, 2–216
- ?ORC2–?ORC4 offsets, 2–432–2–433
- ?ORCL value, 2–432
- ?ORCN offset, 2–432–2–433
- ?ORCP offset, 2–432–2–433
- ?ORCQ offset, 2–432–2–433
- ?ORCS offset, 2–432–2–433
- ?ORCT offset, 2–432–2–433
- ?ORDS value, 2–66–2–67
- ?ORDY value, 2–66–2–67
- ?ORE2 offset, 2–425–2–426
- ?ORES offset, 2–425–2–426
- ?OREV offset, 2–425–2–426
- ?ORFX value, 2–66–2–67
- ?ORLC value, 2–431
- ?ORLO value, 2–431
- ?ORMNV value, 2–431
- ?ORMU value, 2–431
- ?ORP2–?ORP4 offsets, 2–434
- ?ORPE offset, 2–434
- ?ORPL value, 2–434
- ?ORPN offset, 2–434
- ?ORPP offset, 2–434
- ?ORPS offset, 2–434
- ?ORPT offset, 2–434
- ?ORSC offset, 2–432–2–433
- ?ORVR value, 2–66–2–67

- OS abbreviation, 2–2
- ?OSID offset, 2–432
- ?OSIL offset, 2–432
- ?OSLN value, 2–432
- ?OSN2 and ?OSN3 offsets, 2–432–2–433
- ?OSNF offset, 2–432
- ?OSNG value, 2–431
- ?OSNL value, 2–431
- ?OSNN offset, 2–432
- ?OSNO value, 2–431
- ?OSNP offset, 2–432
- ?OSNQ offset, 2–432
- ?OSNR offset, 2–432
- ?OSNT offset, 2–432
- ?OSOL offset, 2–432
- ?OSPI offset, 2–425–2–426
- other file types, getting status of, 2–168
- output, restarting, 2–40
- overhead, pipe, 2–416
- overlay
 - exiting from an, 2–510
 - loading an, 2–511
 - releasing an, 2–509, 2–513
- overrun, timing, 2–790
- ?OVEX system call, 2–509
- ?OVKIL system call, 2–510
- ?OVL0D system call, 2–511–2–514
- ?OVREL system call, 2–513
- owner of a port, finding the, 2–308

P

- ?PACDEV value, 2–555, 2–556
- packet address and parameters, 1–3
- pages
 - shared, flushing to disk, 2–103, 2–162, 2–643
 - undedicated memory, 2–204
 - unwiring, 2–779
 - wiring, 2–76
- palette, 2–223
- ?PALW value, 2–417
- parity setting, field mask, 2–197
- partition, shared, 2–718
- partition size, changing a process's, 2–787

PARU file, A-39, A-40
 PARU.16.SR file, 1-3
 PARU.32.SR file, 1-3, 1-8, 1-9
 PARU_LONG.SR file, 1-3, 1-8
 Pascal language, 1-10
 passing
 connection, 2-516, 2-523
 control from one program to another, 2-33
 PASSTHRU mode, 2-115
 password
 data encryption, 2-590
 encrypting a, 2-554
 length, 2-590
 path, task execution, 2-278
 pathname
 complete, 2-205
 complete, of generic file, 2-239
 getting a file's complete, 2-32
 process or program, 2-222
 remote host, 2-639
 window, 2-226, 2-569
 ?PBATCHP value, 2-555, 2-556
 ?PBCHPRV value, 2-555, 2-556
 ?PBLKS offset, 2-561, 2-904
 ?PBLMEM value, 2-555, 2-556
 ?PBLT value, 2-597, 2-705
 ?PBRK offset, 2-30
 ?PBRK value, 2-538
 ?PBWSS value, 2-555, 2-556
 PBX support, callout, 2-198
 ?PCAD offset, 2-526-2-527, 2-597, 2-598,
 2-705
 ?PCAL offset, 2-537, 2-540, 2-543
 ?PCHPRI value, 2-555, 2-556
 ?PCHTYP value, 2-555, 2-556
 ?PCHUSER value, 2-555, 2-556
 ?PCHWSSL value, 2-556
 ?PCL... offsets and values, 2-515
 ?PCLASS system call, 2-514
 ?PCNSPRV value, 2-555, 2-556
 ?PCNX system call, 2-516
 ?PCOMMNT value, 2-555
 ?PCON offset, 2-537, 2-540
 ?PCS1-?PCS8 offsets, 2-526-2-527
 ?PDBLOCY value, 2-557
 ?PDEL value, 2-409, 2-412
 ?PDESLN value, 2-561, 2-565, 2-904, 2-907,
 2-909
 ?PDFP offset, 2-537, 2-541
 ?PDIR offset, 2-537, 2-539
 ?PDISKLM value, 2-554
 ?PDLOCY value, 2-557
 ?PDMP value, 2-539
 PED, 2-217
 :PER directory, 2-60
 peripheral directory, 2-60
 permanent file attribute, 2-653
 permitting, access to a protected file, 2-519
 ?PFADW offset, 2-520, 2-702
 ?PFAL offset, 2-557
 ?PFBI value, 2-554
 ?PFBS value, 2-538
 ?PFBY value, 2-557
 ?PFCRE value, 2-553
 ?PFDA value, 2-538
 ?PFDB value, 2-84, 2-538
 ?PFDEL value, 2-553
 ?PFDL offset, 2-554, 2-557
 ?PFDLL value, 2-548
 ?PFDP offset, 2-554, 2-557
 ?PFER offset, 2-557
 ?PFEX value, 2-538
 ?PFFC offset, 2-552, 2-553
 ?PFFD offset, 2-554
 ?PFFLG offset, 2-520, 2-701-2-702
 ?PFFO value, 2-520, 2-701-2-702
 ?PFIAC value, 2-552, **2-553**
 ?PFIH offset, 2-520, 2-702
 ?PFLB value, 2-552
 ?PFLE value, 2-553
 ?PFLG offset, 2-84, 2-537-2-540, 2-544
 ?PFLNG value, 2-520, 2-702
 ?PFNF offset, 2-552, 2-553
 ?PFPID offset, 2-520, 2-702
 ?PFPM value, 2-538, 2-540
 ?PFPP value, 2-538

?PFPR offset, 2-552, 2-553
 ?PFR1 offset, 2-552, 2-553
 ?PFR3 offset, 2-557
 ?PFRDF value, 2-553
 ?PFREN value, 2-553
 ?PFRNG offset, 2-520, 2-702
 ?PFRP value, 2-75, 2-539
 ?PFRS value, 2-75, 2-539
 ?PFRV offset, 2-552, 2-553
 ?PFRW value, 2-520, 2-702
 ?PFSE value, 2-554
 ?PFTAC value, 2-552, **2-553**
 ?PFUFD value, 2-553
 ?PFUN offset, 2-552, 2-553
 ?PFVER value, 2-557
 ?PFW value, 2-554
 ?PFXP value, 2-538, 2-545
 ?PHRDPRV value, 2-556
 physical block I/O, 2-210, 2-405, 2-525, 2-592
 ?PICCFN value, 2-554
 ?PICROG value, 2-554
PID
 and global port number association, 2-219
 getting information about, 2-517
 of a process's father, getting the, 2-79
 returning active, 2-217
 translating a, 2-769
 virtual, 2-266
 ?PIDS system call, 2-517-2-518
 ?PIECE_PKT... offsets and values,
 2-347-2-348
 ?PIFG offset, 2-416-2-417
 ?PIFP offset, 2-537, 2-541
 ?PILN value, 2-416
 ?PILRP offset, 2-518
 ?PILTH value, 2-518
 ?PIMXP offset, 2-518
 ?PINTDIR value, 2-556
 PIO instructions, 2-151, 2-271
 ?PIPC offset, 2-537, 2-539, 2-543
 ?PIPD offset, 2-416-2-417
 pipe extension packet, 2-406, **2-416**
 pipe file, creating a, 2-68
 pipe length, maximum, 2-410
 pipe size, offset ?IMRS, 2-416
 pipes, overhead, 2-416
 ?PIPI offset, 2-416-2-417
 ?PIPR offset, 2-518
 ?PIRS offset, 2-416-2-417
 ?PIRV offset, 2-416-2-417
 PIT device, 2-151, 2-271
 ?PITI offset, 2-416-2-417
 ?PITOT offset, 2-518
 pixel map
 changing colors, 2-234
 deleting, 2-228
 ID, 2-226
 manipulating, 2-223
 related palette, 2-234
 ?PKR0 value, **2-106-2-112**, 2-553
 ?PKR1 value, 2-121, 2-122, 2-694
 PL/I language, 1-10
 ?PLFP offset, 2-537, 2-541
 ?PLOGON value, 2-554
 ?PLTH value, 2-537, 2-546-2-548
 ?PMCTS value, 2-556
 ?PMDIS offset, 2-562, 2-565
 ?PMDSN offset, 2-562, 2-565
 ?PMEM offset, 2-537, 2-539, 2-543
 ?PMGSYS value, 2-555, 2-556
 ?PMODPRV value, 2-555, 2-556
 ?PMTPF system call, 2-160, **2-519-2-520**
 ?PMXSONS value, 2-554, 2-556
 ?PMYSONS value, 2-555, 2-556
 ?PN1FLG value, 2-557
 ?PN2FLG value, 2-557
 ?PNAME system call, 2-521-2-524b
 ?PNBLMEM value, 2-555, 2-556
 ?PNBWSS value, 2-555, 2-556
 ?PNCRYPT value, 2-554
 ?PNM offset, 2-537, 2-539
 ?PNVR value, 2-417
 ?POBLOCY value, 2-557
 ?POFP offset, 2-537, 2-541
 point-to-point station, 2-715
 pointer
 byte, 1-5

- device, controlling input from a, 2-567
- event, 2-191
- file, 2-220, 2-610, 2-707
- frame, 1-2
- poll, address and list, 2-664-2-667
- ?POLOCY value, 2-557
- port number
 - global, 2-219, 2-288, 2-289, 2-308, 2-770
 - local, 2-219, 2-770
 - terminal, 2-185
- position, bit, 1-4
- powerfail/restart routine, user device, 2-277
- ?PPASSWD value, 2-554
- ?PPBLT value, 2-526
- ?PPCR offset, 2-537, 2-540
- ?PPDPMGR value, 2-555, 2-556
- ?PPRCINF value, 2-556
- ?PPRI offset, 2-537, 2-539, 2-543
- ?PPRNBLK value, 2-555, 2-556
- ?PPRV offset, 2-537, 2-540
- ?PPSUPP value, 2-555, 2-556
- ?PPWDPRV value, 2-555, 2-556
- ?PQBLOCY value, 2-557
- ?PQLOCY value, 2-557
- ?PRBB offset, 2-526-2-529
- ?PRCL offset, 2-260, 2-526-2-527, 2-597, 2-598, 2-705
- ?PRCNX system call, 2-523-2-524b
- ?PRCRYPT value, 2-557
- ?PRDB system call, 2-525-2-526
- PREDITOR utility program, 2-552, 2-744
- ?PRES offset, 2-597, 2-598, 2-705
- ?PRHRDPR value, 2-557
- ?PRI system call, 2-530
- prime meridian, 2-247, 2-400
- priority
 - changing a process, 2-531
 - changing a task, 2-530
 - getting calling task, 2-399
- ?PRIPR system call, 2-531-2-534b
- privilege state, 2-744
- privileges, access control, 2-245
- ?PRKIL system call, 2-533
- ?PRNH offset, 2-16, 2-260, 2-526-2-527, 2-596, 2-597, 2-598, 2-705
- ?PRNL offset, 2-597, 2-598
- ?PROC extension packet, 2-545-2-548
- ?PROC functions
 - creating offspring, 2-544
 - sending a CLI-like command line, 2-543
 - setting maximum CPU time, 2-544
 - setting the working set size, 2-545
- ?PROC system call, 2-240, 2-534-2-534h
 - examples of, A-9, A-48
- procedure, chaining to a new, 2-595
- process
 - address space, remapping a, 2-353
 - blocking a, 2-26
 - changing priority of a, 2-531
 - class and locality, 2-514
 - communication, operator/current, 2-437
 - creating a, 2-534
 - getting the PID of a father, 2-79
 - getting the virtual PID of a, 2-266
 - location, 2-28
 - management system calls (names of), 2-4-2-5
 - name
 - full, 2-265, 2-521
 - locating a, 2-28
 - partition size, changing, 2-787
 - pathname, getting a, 2-222
 - priority values, 2-532
 - returning status information, 2-560
 - runtime statistics, 2-648
 - son, 2-696
 - status information, extended, 2-900
 - synchronizing, 2-305
 - terminal, 2-771
 - terminating a, 2-29, 2-754
 - termination
 - code, 2-299
 - message, **2-295-2-297**, 2-297, 2-635
 - termination message, 16-bit B-type or C-type, 2-304.6
 - termination messages
 - B-type, 2-304-2-304.8
 - C-type, 2-304-2-304.8
 - type, changing a, 2-75
 - unblocking a, 2-776
 - username, 2-265
 - waiting for another, 2-845
- PROCESS command, 2-255
- Process Environment Display utility program, 2-217
- process types, 2-297, 2-304

processor
 class assignments, logical, 2-362
 identification, unique hardware, 2-263

profile requests and functions, 2-551-2-552

?PROFILE system call, 2-547, 2-548,
 2-551-2-552

program
 assembly language example, 1-5-1-9
 chaining to, 2-33
 construction and execution, 1-8
 file, loading a, 2-637
 getting a pathname, 2-222
 returning (.PR) filename, 2-640
 sample sets, iv, 1-5-1-9, x

protected file, 2-519

protected shared file, 2-701
 opening a, 2-701

protecting, a task from being redirected, 2-757

protocol data-link control characters, BSC,
 2-674

?PRPSSWD value, 2-557

?PRRAPRV value, 2-517, 2-555, 2-556

?PRRDY system call, 2-558

?PRSF'TPR value, 2-557

?PRSUS system call, 2-559

?PSAL value, 2-562, 2-563

?PSCH offset, 2-562, 2-564

?PSCPL offset, 2-562, 2-564

?PSCW offset, 2-562, 2-564

?PSEN offset, 2-562, 2-563

?PSEX offset, 2-562, 2-564, 2-565

?PSF2-?PSF5 offsets, 2-562-2-564

?PSFA offset, 2-562, 2-565

?PSFL offset, 2-562, 2-563

?PSFP offset, 2-562, 2-563

?PSF'TPRV value, 2-556

?PSHRP value, 2-564, 2-906

?PSHSH offset, 2-561, 2-904

?PSHST offset, 2-561, 2-904

?PSHSZ offset, 2-561, 2-904

?PSIH offset, 2-562, 2-565

?PSLFA offset, 2-562, 2-565

?PSLTH value, 2-562

?PSMX offset, 2-562, 2-565

?PSNM offset, 2-537, 2-539

?PSNR offset, 2-562, 2-563

?PSNS offset, 2-562, 2-563

?PSOPIO value, 2-554, 2-556

?PSPD offset, 2-562, 2-564

?PSPH offset, 2-562, 2-564

?PSPP value, 2-563, 2-905

?PSPR offset, 2-562, 2-564

?PSPRST offset, 2-561, 2-904

?PSPV offset, 2-562, 2-564

?PSQF offset, 2-562, 2-563

?PSRH offset, 2-562, 2-564

?PSSL offset, 2-562, 2-565

?PSSN offset, 2-562, 2-563

?PSSP value, 2-563, 2-905

?PSST offset, 2-562, 2-563

?PSTAT system call, 2-560-2-566

?PSTI offset, 2-16, 2-526-2-529, 2-597, 2-598,
 2-704, 2-705

?PSTO offset, 2-526-2-527, 2-597, 2-598,
 2-705

?PSUSER value, 2-555, 2-556

?PSWM offset, 2-562, 2-565

?PSWS offset, 2-562, 2-565

?PSXPT value, 2-564, 2-906

PTE abbreviation, 2-781

?PTRDEV_EVENTS... values, 2-578, 2-586

?PTRDEV_GEN_EVENT... offsets and values,
 2-585-2-586

?PTRDEV_GENERATE_EVENT function,
 2-568, **2-584-2-586**

?PTRDEV_GENERATE_EVENT_PKTID value,
 2-586

?PTRDEV_GET_LOC... offsets, 2-587-2-588

?PTRDEV_GET_PTR_LOCATION function,
 2-568, 2-571, 2-587

?PTRDEV_GET_PTR_STATUS function, 2-568,
 2-581

?PTRDEV_GET_TABLET_LOCATION
 function, 2-568, 2-571, 2-588-2-589

?PTRDEV_GET_TABLOC... offsets and values,
 2-588

?PTRDEV_GSTATUS... offsets and values,
 2-581-2-584

?PTRDEV_LAST_EVENT function, 2-568,
 2-571, **2-577**

?PTRDEV_LEVENT... offsets and values, 2-577-2-578

?PTRDEV_PKT... offsets and values, 2-569-2-571

?PTRDEV_PTR_SHAPE... values, 2-580-2-581, 2-582

?PTRDEV_PTR_STATE... values, 2-580-2-581

?PTRDEV_SET_DELTA function, 2-568, 2-573, **2-576-2-577**

?PTRDEV_SET_DELTA... offsets and values, 2-576-2-577

?PTRDEV_SET_DELTA_LEN value, 2-576

?PTRDEV_SET_EVENTS function, 2-568, 2-573

?PTRDEV_SET_EVTS... offsets and values, 2-573, 2-574-2-576

?PTRDEV_SET_POINTER function, 2-568, **2-579-2-580**

?PTRDEV_SET_PTR... offsets, 2-579-2-580

?PTRDEVICE functions
controlling the operation of the pointer, 2-579-2-580
dead tablet space, 2-574
generating a pointer event, 2-584-2-586
getting
information about the last pointer event, 2-577
pointer status, 2-581
status of the pointer device, 2-587
getting the tablet status, 2-588
moving the pointer, 2-584-2-586
selecting pointer events, 2-573-2-575
specifying a pointer delta, 2-576

?PTRDEVICE system call, 2-191, **2-567-2-568**
example of, A-61

?PTWO value, 2-417

?PUDAH offset, 2-603

?PUDAL offset, 2-603

?PUDCN offset, 2-603

?PUDFW offset, 2-603

?PUDLT value, 2-603

?PUIPCS value, 2-555, 2-556

?PUL_MAX_NAMES value, 2-388, 2-892

?PUL_PKT... offsets and values, 2-388, 2-892

?PUNM offset, 2-537, 2-540

?PUSPR offset, 2-561, 2-904

PVC circuit connections, see ?CONINFO, 2-58.9

?PVCNPRV value, 2-555, 2-556

?PVDV value, 2-542

?PVEX value, 2-542

?PVIP value, 2-542

?PVPC value, 2-542

?PVPP value, 2-542

?PVPR value, 2-542

?PVSP value, 2-542

?PVSU value, 2-542

?PVTY value, 2-542

?PVUI value, 2-542

?PVWM value, 2-542

?PVWS value, 2-542, 2-545

?PWBP offset, 2-591

?PWDCRYP system call, 2-554, **2-590-2-591**

?PFWF offset, 2-591

?PWLO offset, 2-591

?PWMI offset, 2-537, 2-541, 2-545

?PWOW value, 2-591

?PWRB system call, 2-213, **2-525-2-526**, 2-592

?PWRV offset, 2-591

?PWSO value, 2-556

?PWSS offset, 2-537, 2-540, 2-545

?PWSZ value, 2-591

?PXCPU offset, 2-546-2-548

?PXFLG offset, 2-546-2-548

?PXLE value, 2-545, 2-546-2-548

?PXLLOC offset, 2-546-2-548

?PXPAG offset, 2-546-2-548

?PXPGI offset, 2-546-2-548

?PXPGN offset, 2-546-2-548

?PXPUN offset, 2-546-2-548

?PXRES offset, 2-546-2-548

?PXRS0 and ?PXRS1 offsets, 2-546-2-548

?PXSID value, 2-548

?PXSPI offset, 2-546-2-548

?PXULOC offset, 2-546-2-548

?PXUPID offset, 2-546-2-547

Q

QSYM.F77.IN file, A-40

- queue
 - name, locating a, 2-28
 - removing tasks from a, 2-90
 - task manager, 2-294
- queues
 - batch, document names, 2-122
 - print, document names, 2-122

R

- radix, v, xi
- ?RCALL system call, -594
- ?RCHAIN system call, 2-595
- ?RCID value, 2-433
- ?RDAC value, 2-781
- ?RDB system call, 2-596-2-601
- ?RDUDA system call, 2-602-2-603
- re-enabling
 - control-character terminal interrupt, 2-334
 - relative terminal, 2-667, 2-685
- re-establishing, connection, 2-516, 2-523
- ?READ system call, 2-191, ~~2-604~~
 - examples of, A-12, A-19, A-22-A-23, A-33, A-67
- read/write access to memory, 2-782
- reading
 - allocated blocks, 2-23
 - block I/O, 2-596
 - device characteristics, 2-177
 - error message file, 2-99
 - record I/O, 2-604
 - shared-page, 2-704
 - task message from the process terminal, 2-771
 - time-of-day conversion data, 2-645
 - user data area, 2-602-2-604
- reading
 - task, 2-281
 - task status word, 2-282
 - tasks of a specified priority, 2-558
- real-time clock, 2-313
- ?REC system call, 2-627
 - example of, A-34
- receive/continue call, 2-714
- receiving
 - after sending an IPC message, 2-309
 - information over a BSC line, 2-709
 - interrupt service message, 2-290

- intertask message, 2-627
- intertask message without waiting, 2-628
- IPC message, 2-295
- ?RECNEW system call, 2-628
- record I/O, 2-604
 - performing, 2-604
 - writing, 2-604, 2-844
- ?RECREATE system call, 2-629
- recreating, a file, 2-629
- rectangle, clip, 2-223
- redirecting, task, 2-774
 - execution path, 2-278
 - protection, 2-757
- register
 - floating-point status, 2-285
 - stack, 2-314
- Related manuals, ix
- relative terminal, 2-667, 2-685
 - disabling a, 2-667
 - re-enabling a, 2-667, 2-685
- ?RELEASE system call, 2-630
- releasing
 - initialized logical disk, 2-630
 - job processor, 2-322
 - overlay, 2-509, 2-513
 - resource, -594
 - shared page, 2-643
- ?REM value, 2-168
- remapping, a process's address space, 2-353
- remote host, 2-639
 - process and queue name on, 2-28
- removing
 - permanent file attribute, 2-653
 - tasks from a queue, 2-90
 - user device, 2-304.9
- ?RENAME system call, 2-632-2-632.2
- renaming, a file, 2-632
- reporting, index, 1-9
- request, profile, 2-551
- ?RESCHED system call, 2-633
- rescheduling
 - disabling task, 2-88
 - task, 2-280
 - time slice, 2-633
- reserved symbols, 1-5
- reset MRC routes, current, 2-58
- ?RESIGN system call, 2-634

resource
 acquiring a, -594
 acquiring a new, 2-328
 base of the current, 2-186
 calling, 2-328
 releasing a, -594

resources, system calls, 2-12

restoring, the previous environment, 2-778

return
 error and normal, 1-2
 normal, 1-2

?RETURN system call, 1-2, 2-635
 examples of, 1-6-1-7, A-4, A-6, A-8-A-9,
 A-12, A-15, A-19, A-23, A-27, A-29,
 A-33-A-34, A-37-A-38, A-43, A-48,
 A-59

returning
 active PIDs, 2-217
 class scheduling statistics, 2-45
 code and text (error), 2-683
 complete pathname of generic file, 2-239
 error code and text, 2-683
 extended status information on a process,
 2-900
 from a process, 2-635
 global port number, 2-288
 LEF mode status, 2-352
 logical disk information, 2-340
 number of undedicated memory pages, 2-204
 OS-format internal time, 2-313
 PID associated with a global port number,
 2-219
 process statistics, 2-560
 program (.PR) filename, 2-640
 stack frame information, 2-807
 status information on a process, 2-560
 status of a task, 2-756, 2-777
 text and code (error), 2-683
 unique task identifier, 2-688, 2-690, 2-777

?RFAB value, 2-298, 2-635

?RFCF value, 2-298, 2-635
 example of, 1-7

?RFEC value, 2-298, 2-635
 example of, 1-7

?RFER value, 2-298, 2-635
 example of, 1-7

?RFWA value, 2-298, 2-635

ring
 base address, 2-565
 field, 2-289
 loading, stopping, 2-642
 lower, 2-353

?RINGLD system call, 2-160, 2-637-2-641
 example of, A-14

**RINGLOAD.SR sample program, A-2,
 A-14-A-18**

?RLFRC offset, 2-630

RMA access, 2-517

?RNAME system call, 2-639

?RNGBP offset, 2-641

?RNGLB offset, 2-641

?RNGNM offset, 2-641

?RNGPL value, 2-641

?RNGPR system call, 2-640-2-641

?RNGST system call, 2-642

routine, power, fail/restart, 2-272

?RPAGE system call, 2-643-2-644

?RSID value, 2-435

RTC device, 2-151, 2-271

?RTDS value, 2-409, 2-412, 2-608, 2-609

?RTDY value, 2-409, 2-608, 2-609

?RTFX value, 2-409, 2-608, 2-609

?RTODC system call, 2-645-2-647

**?RTODC_PKT... offsets and values,
 2-646-2-647**

?RTUN value, 2-409, 2-608, 2-609

?RTVB value, 2-409, 2-608, 2-609

?RTVR value, 2-409, 2-608, 2-609

runtime process statistics, 2-648
 getting, 2-648

**RUNTIME.SR sample program, A-1,
 A-11-A-13**

?RUNTM system call, 2-648
 example of, A-11

?RVBPL value, 2-783

?RVBPX value, 2-783

RVI character, 2-676

?RVWPL value, 2-783

S

- S operator, 1–4
- ?SACK value, 2–711
- ?SACL system call, 2–213, **2–650–2–652**
- ?SACP offset, 2–165–2–170
- ?SAFM offset, 2–527
- ?SAFM value, 2–598
- ?SAK0 and ?SAK1 values, 2–711
- sample programs, 1–5–1–9
- ?SASC value, 2–671, 2–677
- ?SATR system call, 2–653–2–654
- ?SAVS value, 2–693
- ?SBER offset, 2–686, 2–687
- ?SBIAS system call, 2–655
- ?SBSC value, 2–672
- ?SBUL offset, 2–721
- ?SBUP offset, 2–710, 2–712, 2–721, 2–723
- ?SBYC offset, 2–710, 2–712, 2–721, 2–723
- ?SBYM offset, 2–710, 2–713, 2–721, 2–723
- scalar date value, converting a, 2–31
- scalar time value, converting a, 2–74
- scaled space on a tablet, 2–572
- scheduler, system, 2–27
- scheduling
 - disabling task, 2–93
 - enabling multitask, 2–102
- ?SCHN offset, 2–670, 2–671
- ?SCHR system call, 2–656–2–657
- ?SCIT value, 2–671
- ?SCLOSE system call, 2–658–2–659
- ?SCON value, 2–722
- SCP boot clock, 2–401
- SCP device, 2–151, 2–271
- ?SCPS offset, 2–168
- ?SCRC value, 2–671
- screen management extension, 2–612
- ?SCSH offset, 2–167, 2–168
- ?SCSL offset, 2–167, 2–168
- ?SDAC value, 2–712, 2–716, 2–722
- ?SDAD offset, 2–710, 2–713
- ?SDAY system call, 2–660
- ?SDBL system call, 2–661
- ?SDCN value, 2–188, 2–662
- ?SDCU offset, 2–165
- ?SDDN value, 2–188, 2–662
- ?SDEH offset, 2–168
- ?SDEL offset, 2–168
- ?SDET value, 2–723
- ?SDIS value, 2–723
- ?SDLM system call, 2–662–2–663
- ?SDMD value, 2–671
- ?SDPOL system call, 2–664–2–666
- ?SDPP value, 2–671
- ?SDPR value, 2–671
- ?SDRT system call, 2–667–2–668
- ?SDSC value, 2–671
- ?SDTI value, 2–188, 2–662
- ?SDTO value, 2–188, 2–662
- ?SDTP value, 2–188, 2–662
- search list
 - getting contents of a, 2–203
 - setting the, 2–695
- ?SEBC value, 2–671
- ?SEBL system call, 2–669–2–678
- ?SECHR system call, 2–41, **2–679–2–680**
- sector I/O, modified, 2–20, 2–25, 2–216
- ?SEFH offset, 2–167, 2–168
- ?SEFL offset, 2–167, 2–168
- ?SEFM offset, 2–167, 2–168
- ?SEFW offset, 2–167, 2–168
- ?SEID value, 2–431
- select address, 2–665
- select–address pair, 2–664
- ?SELN value, 2–670
- ?SEND system call, 2–681–2–682
 - example of, A–27
- sending
 - information over a BSC line, 2–720
 - IPC message, 2–305, 2–309
 - terminal message, 2–681
- ?SEOT value, 2–723
- ?SEPR value, 2–671, 2–677
- sequences, keyboard interrupt, 2–332

- ?SERMSG system call, 2-683-2-684
- ?SERT system call, **2-667**, 2-685
- ?SERVE system call, 2-685
- server
 - becoming a, 2-685
 - becoming a customer of, 2-56
 - resigning as a, 2-634
- server/customer relationship, 2-34, 2-57
 - disconnecting a, 2-80, 2-92
 - terminating a, 2-72
- service
 - message, interrupt, 2-290
 - routine
 - device interrupt handler, 2-275
 - fast device interrupt handler, 2-156
- session, connection types, 2-58.9
- set/get class ID code, 2-36-2-56
- setting
 - access control list, 2-650
 - bias factor values, 2-655
 - binary I/O on a pipe, 2-611
 - bit, 1-4
 - class IDs, 2-36
 - class matrix, 2-51
 - data channel map, 2-729
 - default access control list, 2-77
 - delimiter table, 2-662
 - device time-out value, 2-733
 - execute-protection status, 2-141-2-142
 - extended device characteristics, 2-679
 - file-pointer position, 2-707
 - IPC no wait, **2-409**, 2-611
 - logical processor class assignments, 2-362
 - maximum size for a control point directory, 2-58.26
 - permanent file attribute, 2-653
 - search list, 2-695
 - system
 - calendar, 2-660
 - clock, 2-732
 - identifier, 2-719
 - time of day, 2-400, 2-732
- ?SFAH offset, 2-167, 2-168
- ?SFAL offset, 2-167, 2-168
- ?SGES system call, 2-686-2-687
- ?SGLN value, 2-686
- shared
 - access, 2-699
 - access file
 - closing a, 2-658
 - opening a, 2-405
 - file, 2-659, 2-699
 - protected, 2-701
 - file memory pages, flushing to disk, 2-103
 - page, 2-643
 - flushing a disk, 2-162
 - read, 2-704
 - partition, 2-243, 2-718
- ?SHC0 offset, 2-198
- ?SHC0 value, 2-335
- ?SHCO value, 2-180
- ?SHFS offset, 2-167
- ?SHOP value, 2-409, **2-414**
- /SHR switch, 2-180
- ?SHSP value, 2-672
- ?SIDX offset, 2-167, 2-168
- ?SIEX value, 2-693
- signaling
 - another task, 2-688, 2-690
 - mechanism, interprocess, 2-688, 2-690
- signaling mechanism, interprocess, 2-691, 2-845
- significant bits, least and most, 1-5
- ?SIGNL system call, 2-688-2-689, 2-691, 2-845, 2-848
- ?SIGWT system call, 2-690-2-691
- ?SIID offset, 2-693
- ?SILN offset, 2-693
- ?SIMM offset, 2-693
- ?SINFO system call, 2-692-2-692b
- ?SINT value, 2-723
- ?SIOS offset, 2-693
- ?SIPL value, 2-693
- ?SIRL offset, 2-721, 2-723
- ?SIRN offset, 2-693
- ?SIRS offset, 2-693
- ?SITB value, 2-711, 2-716, 2-722
- size, shared partition, 2-243
- ?SLAU offset, 2-167, 2-168
- ?SLBC value, 2-739-2-740
- ?SLCON value, 2-739
- ?SLCSU value, 2-739-2-740
- ?SLDS value, 2-739
- ?SLEC value, 2-739-2-740
- ?SLES value, 2-739

?SLEX value, 2-739
 ?SLFL value, 2-739
 ?SLIST system call, 2-695
 @SLNx device, 2-669
 ?SLON value, 2-739
 ?SLRC value, 2-671, 2-677
 ?SLRE value, 2-739
 ?SLRF value, 2-739-2-740
 ?SLRS value, 2-739
 ?SLSEX value, 2-739
 ?SLSF value, 2-739-2-740
 ?SLSP value, 2-739
 ?SLST value, 2-739-2-740
 ?SLSU value, 2-739
 ?SLTE value, 2-739
 ?SLTH value, 2-165-2-170
 ?SMCH offset, 2-537, 2-541, 2-544
 ?SMDI offset, 2-670, 2-672
 ?SMIL offset, 2-167, 2-168
 ?SMSH offset, 2-167
 ?SMSL offset, 2-167
 ?SNAK value, 2-711
 ?SNID value, 2-673, 2-723
 ?SNKC offset, 2-686, 2-687
 ?SNPR value, 2-671, 2-677
 ?SOAL offset, 2-697-2-698
 ?SOFD offset, 2-697-2-698
 ?SOFW offset, 2-697-2-698
 SOH character, 2-676
 ?SOHB value, 2-711, 2-716, 2-722
 ?SOKEY offset, 2-697-2-698
 ?SOLTH value, 2-697
 son process, 2-696
 SON.SR sample program, A-1, **A-9-A-10**
 ?SONEN offset, 2-697-2-698
 ?SONS system call, 2-696-2-698
 ?SOPEN system call, 2-699-2-700
 ?SOPN offset, 2-165, 2-166, 2-167, 2-168
 ?SOPPF system call, 2-160, **2-701-2-702**
 ?SOPR value, 2-671, 2-677
 ?SORP offset, 2-697-2-698
 ?SOSON offset, 2-697-2-698
 ?SOSP offset, 2-697-2-698
 ?SPAGE system call, 2-704
 ?SPAR value, 2-618
 SPEAK.SR sample program, A-1, A-7
 special key characteristics, 2-191
 ?SPET value, 2-712
 ?SPLR value, 2-711, 2-716
 ?SPNH offset, 2-166
 ?SPNK value, 2-712
 ?SPNL offset, 2-166
 ?SPOS system call, 2-707-2-708
 example of, A-22
 ?SPRO value, 2-705
 ?SPRV value, 2-712
 ?SPTM offset, 2-73
 ?SR32 value, 2-693
 ?SRCV output values, 2-716
 ?SRCV system call, 2-673, **2-709-2-710**
 ?SRES offset, 2-721, 2-723
 ?SRID value, 2-673, 2-711, 2-713
 ?SRVI value, 2-711
 ?SSHPT system call, 2-718
 ?SSID system call, 2-719
 ?SSIL offset, 2-710, 2-712
 ?SSIN offset, 2-693
 ?SSIS offset, 2-710-2-712, 2-721-2-723
 ?SSLR value, 2-711, 2-716
 ?SSND system call, 2-673, **2-720**
 ?SSNL value, 2-710, 2-721
 ?SSTI offset, 2-669-2-672
 ?SSTO offset, 2-687
 ?SSTS offset, 2-165-2-170
 stack
 frame information, 2-807
 register, 2-314
 unwinding the, 2-778
 ?STAH offset, 2-165-2-170, 2-214
 ?STAL offset, 2-165-2-170, 2-214
 standard format for system calls, 1-2
 starting a histogram, 2-286, 2-393, 2-810
 state save area, 2-284

- station
 - identification, **2-672-2-673**, 2-713, 2-725
 - multipoint and point-to-point control, 2-715
 - multipoint tributary, 2-716
- statistics
 - BSC error, 2-686
 - returning class scheduling, 2-45
 - runtime process, 2-648
- status
 - information about a process, extended, 2-900
 - information on a process, returning, 2-560
 - register, floating-point, 2-285
 - word
 - controller, 2-528-2-529
 - task, 2-282
- ?STCH offset, 2-165-2-168
- ?STCL offset, 2-165-2-170
- ?STIM offset, 2-165-2-170
- ?STMAP system call, 2-729-2-731
- ?STMH offset, 2-165-2-170
- ?STML offset, 2-165-2-170
- ?STOC offset, 2-670, 2-672, 2-677
- ?STOD system call, 2-732
- ?STOM system call, 2-733-2-734
- stop bits, stop bit mask, 2-197
- ?STOV offset, 2-710, 2-713, 2-721, 2-723
- streaming mode of tape I/O, 2-215, 2-412
- ?STTD value, 2-723
- ?STTO offset, 2-686
- STX character, 2-676
- ?STXB value, 2-711, 2-716, 2-722
- ?STYP offset, 2-165-2-167, 2-168
- Superprocess
 - mode, 2-735
 - privilege, 2-745
- Superuser
 - mode, 2-737
 - privilege, 2-745
- ?SUPROC system call, 2-735-2-740
- ?SUS system call, 2-736
- ?SUSER system call, 2-737-2-740
- suspending
 - task, 2-27, 2-85, 2-283, 2-736, 2-809
 - tasks of a specified priority, 2-559
- ?SWAK value, 2-712, 2-723
- ?SYFBM offset, 2-743
- ?SYFLT offset, 2-743
- ?SYLEN value, 2-740, 2-743
- ?SYLID value, 2-743
- ?SYLOG, viewing output, 2-741
- ?SYLOG system call, 2-360, **2-739-2-740**
- symbol table file, 2-256, 2-261
- symbolic debugger utility program, 1-3
- symbols, reserved, 1-5
- SYN character, 2-677
- synchronized logical disk unit image, 2-887
- synchronizing
 - LDU images, 2-385
 - processes, 2-305
- SYSID file, A-39, A-40
- SYSID.32.SR file, 1-8
- :SYSLOG file, **2-360**, 2-739-2-740
- SYSLOG record formats, iii, ix, B-1
- ?SYSPRV system call, 2-744
- ?SYSPRV_... offsets and values, 2-745-2-746
- system
 - area, 2-213
 - calendar, 2-660
 - call
 - implicit, A-39
 - log file, 2-357
 - clock, 2-187, 2-258, 2-645
 - frequency of the, 2-201
 - setting the, 2-732
 - identifier
 - getting the, 2-244
 - setting the, 2-719
 - log file
 - entering an event in the, 2-360
 - event codes, B-1
 - manipulating the, 2-739
 - record formats, B-1
 - scheduler, 2-27
- system calls, 1-1
 - 16-bit process (names of), 2-14
 - AOS/RT32 (names of), 2-3
 - AOS/V5 system resources (names of), 2-12
 - class scheduling (names of), 2-12
 - connection management (names of), 2-11, 2-13
 - debugging (names of), 2-7
 - file creation and management (names of), 2-6
 - file input/output (names of), 2-7-2-8
 - interprocess communications (names of), 2-10
 - logging, 2-357
- system calls, 1-1 (continued)
 - memory management (names of), 2-3

- multiprocessor management (names of), 2-11, 2-13
 - multitasking (names of), -9-10
 - process management (names of), 2-4-2-5
 - standard format for, 1-2
 - windowing (names of), 1-9
- system log, record formats, iii, ix
- system log, writing records, B-1
- system log file, iii, ix
- system log file, B-1
- System Manager privilege, 2-745
- system resources, system calls, 2-12
- SYSTEM_CALL_SAMPLES subdirectory of :UTIL, A-1
- ?SYSULEN value, 2-743

T

- ?T32T value, 2-300, 2-301
- table
 - delimiter, 2-188, 2-416, 2-662
 - file, symbol, 2-261
 - map definition, 2-153, 2-154, 2-272, 2-731
- tablet
 - dead space on, 2-572
 - digitize option for, 2-572, 2-574, 2-579
 - digitizing, 2-571-2-572
 - location example, 2-573
 - scaled space on, 2-572
- ?TABR value, 2-300
- ?TALOCK value, 2-757, 2-774
- ?TAOS value, 2-300
- tape
 - block I/O, 2-599
 - density values, 2-337
 - I/O
 - buffered and streaming modes, 2-215
 - buffered mode of, 2-214
 - labeled magnetic, 2-35, 2-38, 2-418
 - forcing end-of-volume, 2-148
 - magnetic, 2-212, 2-417
- task
 - changing priority of a, 2-530
 - changing the priority of a, 2-280
 - control block, 2-282
 - delaying a, 2-85, 2-809
 - execution path, 2-278
 - extended definition, 2-751
 - identifier, 2-399
 - unique, 2-688, 2-690, 2-777
 - initiating a, 2-747
 - initiation queue, 2-294
 - interrupt, 2-278
 - killing a, 2-279, 2-331, 2-510
 - killing a group, 2-533
 - manager, queued, 2-294
 - message, reading from the process terminal, 2-771
 - multi-scheduling, 2-102
 - readying a, 2-281
 - redirecting a, 2-278, 2-774
 - redirection protection, 2-757
 - rescheduling a, 2-88, 2-280-2-281
 - returning status of a, 2-756, 2-777
 - scheduling, disabling, 2-93
 - signaling another, 2-688, 2-690
 - status, returning, 2-756, 2-777
 - status word, 2-282
 - suspending a, 2-27, 2-85, 2-283, 2-736, 2-809
 - suspension, 2-281
 - terminal interrupt, 2-293
 - waiting for another, 2-845
- ?TASK system call, 1-2, **2-747-2-748**
 - examples of, A-29, A-33
- tasks
 - readying, 2-558
 - removing from a queue, 2-90
 - suspending, 2-559
- ?TATH offset, 2-63
- ?TBCX value, 2-73, 2-300
- ?TBLT value, 2-63
- TCB, 2-282
- ?TCCX value, 2-300
- ?TCIN value, 2-299-2-301
- ?TCTH offset, 2-63
- ?TEFH offset, 2-260
- ?TEFM offset, 2-260
- ?TEFW offset, 2-260
- template filename characters, 2-208
- ?TERM system call, 2-754-2-755
 - example of, A-7
- terminal
 - ANSI-standard, 2-182
 - interrupt
 - control-character, 2-333-2-334
 - disabling, 2-403
 - enabling, 2-404
 - task, 2-293
 - waiting for a, 2-335
 - message, 2-681
 - port number, 2-185

- process, 2-771
- relative, 2-667, 2-685
- terminal services, and console line numbers, 2-58.8
- terminating
 - customer/server relationship, 2-72
 - process, 2-29, 2-635, 2-754
- termination
 - code, process, 2-299
 - message
 - 16-bit process, 2-303
 - 32-bit process, 2-302
 - process, 2-297, 2-635
 - routine, user device, 2-276
- termination message, packet structure, B- or C-type, 2-304.5
- text and code error, returning, 2-683
- ?TEXT value, 2-298-2-304.8
- TID
 - and priority of the calling task, getting, 2-399
 - unique, 2-688, 2-690, 2-777
- ?TIDSTAT system call, 2-756
- time
 - current, 2-313
 - date and time zone
 - getting the, 2-247
 - setting the, 2-400
 - Greenwich Mean and Universal, 2-247, 2-400
 - last accessed value, 2-214
 - of day
 - conversion data, reading, 2-645
 - converting to a scalar value, 2-171
 - getting the, 2-258
 - setting the, 2-732
 - overflow, 2-790
 - returning the OS-format internal, 2-313
 - slice, rescheduling a, 2-633
 - value
 - converting a scalar, 2-74
 - format, 2-46
- time-out
 - connect, 2-677
 - value for a device, 2-733-2-735
- ?TIME_PKT... offsets and values, 2-248-2-249, 2-401-2-402
- TIMEOUT.SR sample program, A-2, **A-37-A-38**
- Timer Facility, Virtual, 2-790
- /TLA FILESTATUS command switch, 2-214

- ?TLOCK system call, 2-757-2-758
- ?TLTH value, 2-260
- ?TM6 value, 2-304.2, 2-762
- ?TMPID offset, 2-304.2, 2-762
- ?TMPRV offset, 2-304.2, 2-762
- ?TMRS offset, 2-304.2, 2-762
- ?TMTH offset, 2-63
- ?TMUPD offset, 2-304.2, 2-762
- ?TMYRING value, 2-757, 2-774
- ?TPID system call, 2-769
- ?TPLN value, 2-301
- ?TPORT system call, 2-770
- ?TR32 value, 2-300
- trailer label, 2-38
- ?TRAN value, 2-711, 2-716, 2-722
- transfer modes, magnetic tape, 2-410
- translating
 - local port number to global equivalent, 2-770
 - PID, 2-769
- translation, field, 2-616
- transmitting
 - intertask message, 2-898, 2-899
 - message from an interrupt service routine, 2-157, 2-315
- ?TRAP value, 2-300
- ?TRCON system call, 2-771-2-772
- tributary station, 2-665
 - multipoint, 2-716
- Trojan pointer, 2-781
- ?TRPE offset, 2-198
- ?TRPE value, 2-191, 2-614, 2-680
- ?TRUNCATE system call, **2-773**, A-39
- truncating, a file, 2-259, 2-773
- ?TSELF value, 2-300
- ?TSSP value, 2-756
- ?TSTAT word, 2-282
- ?TSUP value, 2-300, 2-303
- TTD character, 2-677
- TTI device, 2-151, 2-271
- TTO device, 2-151, 2-271
- ?TTY value, 2-181
- ?TUNLOCK system call, 2-774-2-775

U

- ?UBLPR system call, 2-776

- ?UDBM value, 2-153-2-154, 2-273-2-274
- ?UDDA-?UDDP values, 2-153-2-154, 2-273-2-274
- ?UDDRS offset, 2-152, 2-271, 2-272, 2-277
- ?UDDTR offset, 2-152, 2-271, 2-272, 2-276
- ?UDELTH value, 2-153, 2-272
- ?UDFE value, 2-152
- ?UDFX value, 2-152
- ?UDID offset, 2-153-2-154, 2-273-2-274
- ?UDIRL offset, 2-152
- ?UDLE value, 2-271
- ?UDLN value, 2-152, 2-271-2-272
- ?UDLTH value, 2-153, 2-272
- ?UDNS offset, 2-154, 2-274
- UDPR, 2-277
- ?UDRS offset, 2-152, 2-271, 2-272
- UDTR, 2-276
- ?UDVBX offset, 2-152, 2-272
- ?UDVIS offset, 2-152, 2-272
- ?UDVMS offset, 2-152, 2-272
- ?UDVXM offset, 2-152, 2-272
- ?UIDSTAT system call, 2-777
- ?ULI1-?ULI3 values, 2-392, 2-897
- ?ULLN value, 2-392, 2-897
- ?ULN1-?ULN8 offsets, 2-392, 2-897
- ?ULNC offset, 2-392, 2-897
- ?ULPHI offset, 2-392, 2-897
- ?ULPLO offset, 2-392, 2-897
- ?ULR0-?ULR8 offsets, 2-392, 2-897
- unblocking, a process, 2-776
- undedicated memory pages, 2-204
- uniprocessor histogram, 2-393
- unique
 - hardware processor identification, 2-263
 - task identifier, 2-688, 2-690, 2-777
- unit file, getting status of, 2-165
- Universal Time, 2-247, 2-400
- unlocking
 - an object, 2-172
 - whole files, 2-173
- ?UNMR value, 2-374, 2-376

- unshared
 - memory pages, changing, 2-384
 - memory parameters, listing, 2-383
- ?UNWIND system call, 2-778
- unwinding, the stack, 2-778
- ?UNWIRE system call, 2-779
- unwiring, pages, 2-779
- ?UPDATE system call, 2-213, 2-780
- upfront window, 2-574
- UPSC device, 2-151, 2-271
- ?URTB offset, 2-156
- user
 - data area
 - creating a, 2-70
 - reading a, 2-602-2-604
 - writing a, 2-602-2-604, 2-844
 - device, 2-18
 - defining a, 2-269
 - defining a fast, 2-149
 - powerfail/restart routine, 2-277
 - removing a, 2-304.9
 - termination routine, 2-276
 - locality, changing, 2-354
 - symbol, 2-261
- User Runtime Library, 2-156
- username of a process, 2-265
- UTC, 2-247, 2-400
- ?UTID offset, 2-777
- :UTIL:SYSTEM_CALL_SAMPLES
 - subdirectory, A-1
- ?UTLEN value, 2-777
- ?UTPRI offset, 2-777
- ?UTSK offset, 2-156
- ?UTSTAT offset, 2-777
- ?UUID offset, 2-777

V

- ?VALAD system call, 2-781
- ?VALIDATE system call, 2-782
- validating
 - area for Read or Write access, 2-782
 - logical address, 2-781
- ?VBITM value, 2-376
- ?VCUST system call, 2-786
- ?VDELIM offset, 2-783
- verifying a customer, 2-786, 2-789

?VERRIGN value, 2-793, 2-802
 ?VERRNTR value, 2-793, 2-802
 ?VERROR offset, 2-783
 ?VERRTRM value, 2-793, 2-802
 Vertical Form Unit, loading in the VFU printer,
 2-527
 vertical format unit (VFU), 2-600-2-601
 ?VFUNC offset, 2-783
 Viewing, recent messages, in :SYSLOG and
 :CON0_LOG, 2-741
 ?VINIREV value, 2-792
 virtual PID, 2-266
 Virtual Timer Facility
 attaching, 2-149, 2-151
 cascading and setting, 2-794
 creating, 2-790
 exiting from, 2-806
 killing, 2-797
 modifying, 2-799
 restarting and suspending, 2-804
 synchronizing, 2-795
 waiting for a signal from, 2-848
 waiting for an error message from, 2-846
 ?VLAC value, 2-781
 ?VLENGTH offset, 2-783
 ?VMEM system call, 2-787-2-788
 ?VMLTH value, 2-788
 ?VMODILV value, 2-793, 2-802
 ?VMODNUL value, 2-793, 2-802
 ?VMODSIG value, 2-793, 2-802
 ?VMSTAT offset, 2-788
 volume identifier, checking, 2-35
 ?VPLTH value, 2-783
 ?VPOINTER offset, 2-783
 ?VRCUST system call, 2-789
 ?VRES offset, 2-783
 ?VRESD offset, 2-783
 ?VRING offset, 2-783
 ?VRNBR value, 2-783
 ?VSPIDS value, 2-518
 ?VTERIOVR value, 2-847
 ?VTERSOVR value, 2-847
 ?VTFCAS offset, 2-791-2-792, 2-800-2-801
 ?VTFCREATE system call, **2-790-2-796**,
 2-797, 2-799, 2-805, 2-847
 ?VTFENTR offset, 2-791-2-792, 2-800-2-801
 ?VTFHUNG value, 2-847
 ?VTFID offset, 2-791-2-792, 2-798,
 2-800-2-801
 ?VTFIMSK offset, 2-791, 2-794, 2-800, 2-803
 ?VTFINISUS value, 2-793
 ?VTFKILL system call, 2-797-2-798
 ?VTFKLEN value, 2-798
 ?VTFKREV value, 2-798
 ?VTFMODE offset, 2-791, 2-793, 2-800-2-803,
 2-805
 ?VTFMODIFY system call, 2-790, **2-799-2-803**
 ?VTFMODSUS value, 2-800-2-803
 ?VTFMPLN value, 2-791, 2-800
 ?VTFMREV value, 2-801
 ?VTFONESHOT value, 2-793, 2-802
 ?VTFPID offset, 2-791-2-792, 2-800-2-801
 ?VTFPRI offset, 2-791-2-792, 2-800-2-801
 ?VTFREV offset, 2-791-2-792, 2-798,
 2-800-2-801
 ?VTFRST value, 2-805
 ?VTFSAVE value, 2-793, 2-802
 ?VTFSEC value, 2-794, 2-803
 ?VTFSET offset, 2-791-2-792, 2-800-2-801
 ?VTFSKEW offset, 2-791-2-792, 2-800-2-801
 ?VTFSSUD value, 2-805
 ?VTFSUS system call, 2-793, 2-803,
 2-804-2-805
 ?VTFSYNC offset, 2-791-2-792, 2-800-2-801
 ?VFTTICK value, 2-794, 2-803
 ?VFTTID offset, 2-791-2-792, 2-800-2-801
 ?VTFUNIT offset, 2-791, 2-794, 2-800, 2-803
 ?VTFUSEC value, 2-794, 2-803
 ?VTFXIT system call, 2-806
 ?VWSMAX offset, 2-788

W

WACK character, 2-677
 waiting
 task or process, 2-845
 terminal interrupt, 2-335

?WALKBACK system call, 1-2, **2-807-2-808**

walking back through stacks, 2-807

?WDELAY system call, 2-809
examples of, A-4, A-7, A-37

?WHIST system call, 2-810-2-812

?WIN_BORDER_TYPE... values, 2-822, 2-829,
2-831

?WIN_CRE... offsets and values, 2-818-2-860

?WIN_CREATE_WINDOW function, 2-814

?WIN_DEFINE_PORTS function, 2-814, 2-823

?WIN_DELETE_WINDOW function, 2-229,
2-814, **2-822**

?WIN_DEVICE_STATUS function, 2-815,
2-837-2-838

?WIN_DEVSTAT... offsets and values,
2-838-2-839

?WIN_DISABLE_KEYBOARD function, 2-815,
2-827

?WIN_DPORT... offsets and values,
2-823-2-860

?WIN_ENABLE_KEYBOARD function, 2-815

?WIN_GET_TITLE function, 2-815, **2-832**

?WIN_GET_USER_INTERFACE function,
2-815, **2-830**

?WIN_GET_WINDOW_ID function, 2-815,
2-837

?WIN_GET_WINDOW_NAME function, 2-815,
2-832

?WIN_GINT... offsets and values, 2-830-2-860

?WIN_GTITLE... offsets and values, 2-832

?WIN_HIDE_GROUP function, 2-814, **2-826**

?WIN_HIDE_WINDOW function, 2-814, **2-826**

?WIN_LANG_ID... values, 2-839

?WIN_OUTBACK_GROUP function, 2-814,
2-825

?WIN_OUTBACK_WINDOW function, 2-814,
2-825

?WIN_PALETTE_TYPE... values, 2-821-2-822,
2-837

?WIN_PERMANENCE_OFF function, 2-815,
2-827

?WIN_PERMANENCE_ON function, 2-815,
2-827

?WIN_PKT.CHAN_NUM offset, 2-816-2-817

?WIN_PKT.FLAGS... offset and values,
2-816-2-817

?WIN_PKT.FUNC offset, 2-813, 2-816

?WIN_PKT.LEN value, 2-816

?WIN_PKT.PATH... offsets, 2-816, 2-832

?WIN_PKT.PKT_ID offset, 2-816

?WIN_PKT.SUBPKT offset, **2-816-2-817**,
2-822, 2-825-2-833, 2-837, 2-838

?WIN_PKT.WIND_ID offset, 2-816, 2-838

?WIN_PTR_TYPE... values, 2-839

?WIN_RETURN_DEVICE_WINDOWS
function, 2-815, **2-840**

?WIN_RETURN_GROUP_WINDOWS function,
2-815
preface, 2-840

?WIN_RTN_DEVWINDS... offsets and values,
2-841

?WIN_RTN_GRPWINDS... offsets and values,
2-840

?WIN_SET_TITLE function, 2-815, **2-831**

?WIN_SET_USER_INTERFACE function,
2-815, **2-828-2-860**

?WIN_SINT... offsets and values, 2-828-2-829

?WIN_STATUS... offsets and values,
2-834-2-835

?WIN_STITLE... offsets and values, 2-831

?WIN_SUSPEND_GROUP function, 2-814,
2-827

?WIN_SUSPEND_WINDOW function, 2-814,
2-826

?WIN_UNHIDE_GROUP function, 2-814,
2-826

?WIN_UNHIDE_WINDOW function, 2-814,
2-826

?WIN_UNsuspend_GROUP function, 2-815,
2-826-2-827

?WIN_UNsuspend_WINDOW function,
2-814, **2-827**

?WIN_UPFRONT_GROUP function, 2-814,
2-825

?WIN_UPFRONT_WINDOW function, 2-814,
2-825

?WIN_VT_TYPE... values, 2-836

?WIN_WINDOW_STATUS function, 2-815,
2-833-2-834

window
active group of, 2-574
graphics, 2-226
ID number, 2-226, 2-569
manipulating, 2-813

- pathname, 2-226, 2-569
- upfront, 2-574
- ?WINDOW functions
 - bringing a window or group to the front, 2-825
 - changing the view and scan ports, 2-823-2-860
 - controlling
 - keyboard input to a window, 2-827
 - window output, priority, and visibility, 2-825-2-826
 - creating new, 2-818
 - deleting, 2-229, **2-822**
 - getting
 - current user interface settings, 2-829-2-830
 - status, 2-833-2-860
 - status of a physical device, 2-837-2-860
 - title, 2-831-2-860
 - window IDs, 2-837-2-860
 - making hidden window or group visible, 2-826-2-860
 - resuming output to a suspended window, 2-827
 - sending a window or group to the back, 2-825
 - setting window
 - permanence, 2-827
 - title, 2-831-2-860
 - user interface, 2-828-2-860
 - suspending window output, 2-826
- ?WINDOW system call, 2-191, **2-813-2-841**
 - examples of, A-46-A-57
- windowing, system calls (names of), 2-9
- ?WIRE system call, 2-842-2-843
- wiring
 - Agent, 2-18
 - pages, 2-76
 - pages to the working set, 2-842
- word, task status, 2-282
- .WORD assembly language statement, 1-4
- working directory, changing the, 2-89
- working set, wiring pages to the, 2-842
- ?WRAC value, 2-781
- ?WRB system call, 2-213, **2-596-2-597**, 2-844
- ?WRITE system call, **2-604**, 2-844
 - examples of, A-3-A-4, A-11, A-14-A-16, A-19, A-22-A-23, A-26, A-29, A-34
- WRITE.SR sample program, A-2, **A-22-A-23**
- write/read access to memory, 2-782
- writing
 - block I/O, 2-596, 2-844

- record I/O, 2-604
- user data area, 2-602-2-604, 2-844
- ?WRUDA system call, 2-213, **2-602**, 2-844
- WSB, WSL, and WSP stack registers, 2-314
- ?WTSIG system call, 2-689, 2-691, **2-845**, 2-848
- ?WTVERR system call, 2-793, 2-802, **2-846**
- ?WTVSIG system call, 2-791, **2-848**
- ?WVBPL value, 2-783
- ?VWVPL value, 2-783
- ?XWM1 offsets, **2-135**

X

- ?X0FC offset, 2-198
- ?X1FC offset, 2-198
- ?X1LTH value, 2-113
- ?X2AP value, 2-121
- ?X2CD value, 2-120
- ?X2CM value, 2-121
- ?X2CN value, 2-120
- ?X2CP value, 2-121
- ?X2DD value, 2-121
- ?X2EB value, 2-120
- ?X2EX value, 2-120
- ?X2HO value, 2-120
- ?X2LN value, 2-120
- ?X2RC value, 2-121
- ?X2RM value, 2-121
- ?X2SD value, 2-121
- ?X2SE value, 2-120
- ?X2TO value, 2-121
- ?X3CO value, 2-121
- ?X3TR value, 2-121
- ?XAFD offset, 2-113, 2-118, 2-131, 2-133, 2-135
- ?XAFT offset, 2-113, 2-118, 2-131, 2-133, 2-135
- ?XBFXR value, So String, 2-894
- ?XBIDS value, 2-894
- ?XBNHR value, 2-894
- ?XBORD value, 2-894
- ?XBTRP value, 2-894

?XCREA_DIR... offsets, 2-855-2-856
 ?XCREA_IPC... offsets, 2-856-2-857
 ?XCREA_LNK... offsets, 2-857
 ?XCREA_OTH... offsets, 2-854-2-855
 ?XCREA_PKT... offsets, 2-850-2-851
 ?XCREA_TIME... offsets, 2-853
 ?XCREATE system call, 2-849-2-860
 ?XD2FG offset, 2-138-2-158
 ?XD3FG offset, 2-138-2-158
 ?XDAD offset, 2-139-2-140
 ?XDAT offset, 2-113, 2-114, 2-131, 2-133
 ?XDDBP offset, 2-139-2-140
 ?XD COP offset, 2-139-2-140
 ?XDEP offset, 2-139-2-140
 ?XDEV offset, 2-110-2-111
 ?XDJN offset, 2-139-2-140
 ?XDLMT offset, 2-139-2-140
 ?XDPN offset, 2-139-2-140
 ?XDQP offset, 2-139-2-140
 ?XDRSV offset, 2-139-2-140
 ?XDSD offset, 2-139-2-140
 ?XD SFG offset, 2-138-2-158
 ?XD SQN offset, 2-139-2-140
 ?XDST offset, 2-139-2-140
 ?XDTA offset, 2-139-2-140
 ?XDUL offset, 2-109
 ?XDUN offset, 2-139-2-140
 ?XDUT offset, 2-109
 XEQ DEBUG command, 2-255
 ?XFBAT queue type, 2-114
 ?XFBI value, 2-117, 2-125
 ?XFBP offset, 2-113, 2-118, 2-131, 2-133,
 2-135
 ?XFDA value, 2-116, 2-125
 ?XFDUN function, 2-109
 ?XFEP value, 2-115
 ?XFFO value, 2-117, 2-126
 ?XFFTA queue type, 2-114
 ?XFG2 offset, 2-113, 2-115, 2-131, 2-133,
 2-138, 2-139-2-140
 ?XFGRT value, 2-894
 ?XFGS offset, 2-113, 2-116, 2-117, 2-122,
 2-123, 2-131, 2-133, 2-138-2-158
 ?XFGWD value, 2-894
 ?XFHAM queue type, 2-114
 ?XFHK offset, 2-136, 2-138, 2-139
 ?XFLC value, 2-110-2-111
 ?XFLK offset, 2-136, 2-138, 2-139
 ?XFLO value, 2-110-2-111
 ?XFLPT queue type, 2-114
 ?XFME value, 2-110-2-111
 ?XFMLT function, 2-107-2-108, 2-131
 ?XFMNT function, 2-131
 ?XFMOD function, 2-133, 2-134, 2-471
 ?XFMUN function, 2-106, 2-131
 ?XFNH and ?XF8B values, 2-115
 ?XFNO value, 2-117, 2-125
 ?XFNQN function, 2-136-2-137
 ?XFNR value, 2-116, 2-125
 ?XFNV value, 2-110-2-111
 ?XFOP value, 2-117, 2-125
 ?XFOTH queue type, 2-114
 ?XFP1 and ?XFP2 offsets, 2-127-2-131
 ?XFP2L offset, 2-127-2-131
 ?XFP3 and ?XFP4 offsets, 2-130-2-158
 ?XFPE value, 2-116, 2-126, 2-131
 ?XFPLT queue type, 2-114
 ?XFQDS function, 2-138
 ?XFQN offset, 2-138, 2-139
 ?XFRA value, 2-117, 2-126
 ?XFSH value, 2-116, 2-117, 2-125
 ?XFSNA queue type, 2-114
 ?XFSTAT file types, 2-863
 ?XFSTAT system call, 2-862
 ?XFSTAT_PKT... offsets, 2-866-2-876
 ?XFSTS function, 2-129-2-130
 ?XFSUB queue type, 2-112, 2-114, 2-131
 ?XFTB value, 2-115
 ?XFTI value, 2-117-2-118, 2-126
 ?XFUC value, 2-117
 ?XFWP offset, 2-136, 2-139
 ?XFXDU function, 2-132
 ?XFXML function, 2-108-2-109

?XFXTS function, 2-130
 ?XFXUN function, 2-107, 2-108
 ?XGTACP system call, 2-882
 ?XGTACP_... offsets, 2-883-2-884
 ?XHBP offset, 2-113, 2-122, 2-133
 ?XICNT offset, 2-893, 2-895
 ?XID1-?XID3 offsets, 2-893, 2-895
 ?XIDIR offset, 2-893, 2-895
 ?XIFUN offset, 2-893-2-894
 ?XIGTD value, 2-894
 ?XII1 and ?XII2 values, 2-894
 ?XILDN offset, 2-893, 2-895
 ?XILN value, 2-893
 ?XINIT system call, 2-886
 ?XINIT_CACHE value, 2-889
 ?XINIT_FIXUP_REC value, 2-890
 ?XINIT_LDID_SPECIFIED value, 2-889
 ?XINIT_NO_HARDWARE value, 2-889
 ?XINIT_OVERRIDE value, 2-889
 ?XINIT_PKT... offsets, 2-888-2-910
 ?XINIT_PUL_PKTID value, 2-892
 ?XINIT_ROOT value, 2-889
 ?XINIT_TARGET_DIR value, 2-889
 ?XINIT_TRESPASS value, 2-889
 ?XINIT_WORKING_DIR value, 2-889
 ?XIOP offset, 2-893-2-894
 ?XIP1-?XIP3 offsets, 2-893-2-910
 ?XIPHI offset, 2-893-2-894
 ?XIPL0 offset, 2-893-2-894
 ?XIR1-?XIR9 offsets, 2-893-2-910
 ?XIRA offset, 2-893, 2-896
 ?XIRES offset, 2-893-2-894
 ?XLCAN value, 2-127-2-131
 ?XLDUN value, 2-109
 ?XLFWP value, 2-139-2-140
 ?XLHOL value, 2-127-2-131
 ?XLLC value, 2-110
 ?XLLO value, 2-110
 ?XLME value, 2-110
 ?XLMLT value, 2-107-2-108
 ?XLMNT value, 2-131
 ?XLMOD value, 2-133
 ?XLMT offset, 2-113, 2-115, 2-116, 2-131, 2-133, 2-135
 ?XLMUN value, 2-106
 ?XLNQN value, 2-137
 ?XLNV value, 2-110
 ?XLPN offset, 2-113, 2-115
 ?XLQDS value, 2-139
 ?XLQNB value, 2-137
 ?XLSTS value, 2-129
 ?XLTH value, 2-113
 ?XLUNH value, 2-127-2-131
 ?XLXTS value, 2-130
 ?XMBP offset, 2-113, 2-122, 2-133
 ?XMDF offset, 2-110-2-111
 ?XMEL value, 2-108
 ?XMFC value, 2-108
 ?XMFG offset, 2-110-2-111
 ?XMF1 value, 2-108
 ?XMFN offset, 2-110-2-111
 ?XMFR value, 2-108
 ?XMFS value, 2-111
 ?XMIBM value, 2-35
 ?XMLE offset, 2-108
 ?XMLF offset, 2-108
 ?XMLL offset, 2-107-2-108
 ?XMLR offset, 2-108
 ?XMLS offset, 2-108
 ?XMLT offset, 2-107-2-108
 ?XMLV offset, 2-107-2-108
 ?XMLX value, 2-108
 ?XMSQ offset, 2-110-2-111
 ?XMSQB value, 2-111
 ?XMT system call, 2-898
 ?XMTW system call, 2-899
 example of, A-33
 ?XMUE offset, 2-107
 ?XMUF offset, 2-107
 ?XMUL offset, 2-106
 ?XMUQ offset, 2-107
 ?XMUR offset, 2-107
 ?XMUS offset, 2-107

?XMUT offset, 2-106, 2-109
 ?XMUX value, 2-107
 ?XNRQ offset, 2-139
 ?XNRT offset, 2-139
 XON value, 2-41
 ?XPBP offset, 2-113, 2-118, 2-131
 ?XPCH offset, 2-902, 2-906
 ?XPCID offset, 2-903, 2-907
 ?XPCPL offset, 2-902, 2-906
 ?XPCPU offset, 2-903, 2-907
 ?XPCW offset, 2-902, 2-906
 ?XPDBS offset, 2-902, 2-907
 ?XPDESLN offset, 2-904, 2-907
 ?XPDIS offset, 2-902, 2-907, 2-909
 ?XPDRS offset, 2-902, 2-907
 ?XPEX offset, 2-902, 2-906
 ?XPF2-?XPF5 offsets, 2-902, 2-905-2-906
 ?XPFA offset, 2-902, 2-907
 ?XPFL offset, 2-902, 2-905
 ?XPFP offset, 2-902, 2-905
 ?XPGAB offset, 2-909
 ?XPGBS offset, 2-903, 2-907
 ?XPGLT offset, 2-909
 ?XPGRB offset, 2-909
 ?XPGRS offset, 2-903, 2-907
 ?XPIH offset, 2-902, 2-907
 ?XPLFA offset, 2-903, 2-907
 ?XPLL offset, 2-903, 2-907
 ?XPLTH offset, 2-909
 ?XPLTH value, 2-903
 ?XPMX offset, 2-902, 2-906
 ?XPNBS offset, 2-903, 2-908
 ?XPNR offset, 2-902, 2-905
 ?XPNRS offset, 2-903, 2-908
 ?XPPD offset, 2-902, 2-906
 ?XPPG offset, 2-903, 2-907
 ?XPPH offset, 2-902, 2-906
 ?XPPL offset, 2-903, 2-907
 ?XPPR offset, 2-902, 2-906
 ?XPPU offset, 2-903, 2-908
 ?XPPV offset, 2-902, 2-906
 ?XPR1 offset, 2-902, 2-906
 ?XPRH offset, 2-902, 2-906
 ?XPRI offset, 2-113, 2-131, 2-133, 2-135
 ?XPRI value, 2-116
 ?XPRV offset, 2-105-2-112, 2-113, 2-114
 ?XPSF offset, 2-902, 2-905
 ?XPSID value, 2-905
 ?XPSID1 and ?XPSID2 values, 2-905, 2-909
 ?XPSL offset, 2-903, 2-907
 ?XPSNS offset, 2-902, 2-905
 ?XPSP offset, 2-902, 2-905, 2-908
 ?XPSQF offset, 2-902, 2-905
 ?XPSTAT system call, 2-900-2-909
 ?XPSW offset, 2-902, 2-905
 ?XPSWS offset, 2-904
 ?XPUBS offset, 2-903, 2-908
 ?XPULC offset, 2-903, 2-907
 ?XPUN offset, 2-903, 2-908
 ?XPUPD offset, 2-903, 2-908
 ?XPUQSH offset, 2-909
 ?XPURS offset, 2-903, 2-908
 ?XPUWS offset, 2-904
 ?XPWD offset, 2-113, 2-122, 2-133
 ?XPWM offset, 2-902, 2-906
 ?XPWS offset, 2-902, 2-906
 ?XQ1FG offset, 2-137
 ?XQNJ offset, 2-137
 ?XQQN offset, 2-137
 ?XQQT offset, 2-137
 ?XQRSV offset, 2-137
 ?XRES1-?XRES4 offsets, 2-110-2-112, 2-122,
 2-124, 2-131, 2-133
 ?XRFNC function, 2-132
 ?XRFNC offset, 2-105-2-112
 ?XSCV value, 2-111
 ?XSD1 and ?XSD2 values, 2-111
 ?XSDEN value, 2-111
 ?XSEL value, 2-111
 ?XSEQ offset, 2-113, 2-118, 2-131, 2-133,
 2-134
 ?XSIBM value, 2-111
 ?XSMT value, 2-111

?XSRO value, 2-111
 ?XSVU value, 2-111
 ?XT16T value, 2-304.2, 2-762
 ?XT32T value, 2-304.2, 2-762
 ?XTABR value, 2-304.2, 2-762
 ?XTAOS value, 2-304.2, 2-762
 ?XTBCX value, 2-304.2, 2-762
 ?XTCCX value, 2-304.2, 2-762
 ?XTCIN value, 2-304.2, 2-762
 ?XTIM offset, 2-113, 2-115, 2-131, 2-133
 ?XTR16 value, 2-304.2, 2-762
 ?XTR32 value, 2-304.2, 2-762
 ?XTSUP value, 2-304.2, 2-762
 ?XTYP offset, 2-113, 2-114, 2-131, 2-133,
 2-134
 ?XUSR offset, 2-113, 2-122, 2-133
 XVCT instruction, 2-155
 ?XVOL offset, 2-110-2-111
 XW0-XW3 values, 2-133
 ?XWM0-?XWM3 offsets, 2-133, **2-135**
 ?XWW0-?XWW3 offsets, 2-131, 2-133, **2-135**
 ?XWW0L-?XWW3L offsets, 2-131, 2-133
 ?XWW1 offset, 2-135
 ?XWW2 offset, 2-135
 ?XWW3 offset, 2-135
 ?XXDL offset, 2-132
 ?XXDP offset, 2-132
 ?XXDQ offset, 2-132
 ?XXDT offset, 2-132
 ?XXW0-?XXW3 offsets, 2-113, 2-119-2-120
 ?XXW0L-?XXW3L offsets, 2-113, 2-119-2-120

Y

year, current, 2-187

Z

?ZAC function, 2-439
 ?ZAC3 and ?ZAC4 offsets, 2-442
 ?ZACA offset, 2-442
 ?ZACF offset, 2-442

?ZACI offset, 2-442
 ?ZACR offset, 2-442
 ?ZACZ value, 2-442
 ?ZAG function, 2-439
 ?ZAG value, 2-443
 ?ZAGF offset, 2-443
 ?ZAGI offset, 2-443
 ?ZAGL value, 2-443
 ?ZAGN offset, 2-443
 ?ZAGZ value, 2-443
 ?ZAK value, 2-441
 ?ZAL function, 2-439
 ?ZBI function, 2-439
 ?ZBIA offset, 2-446
 ?ZBIB offset, 2-446
 ?ZBIF offset, 2-446
 ?ZBII offset, 2-446
 ?ZBIL value, 2-446
 ?ZBIZ value, 2-446
 ?ZBL function, 2-439
 ?ZBLA offset, 2-444
 ?ZBLB offset, 2-444
 ?ZBLF offset, 2-444
 ?ZBLI offset, 2-444
 ?ZBLL value, 2-444
 ?ZBLZ value, 2-444
 ?ZBO function, 2-439
 ?ZBOA offset, 2-445
 ?ZBOB offset, 2-445
 ?ZBOF offset, 2-445
 ?ZBOI offset, 2-445
 ?ZBOL value, 2-445
 ?ZBOZ value, 2-445
 ?ZBR function, 2-439
 ?ZBRF offset, 2-447
 ?ZBRI offset, 2-447
 ?ZBRL value, 2-447
 ?ZBRS offset, 2-447
 ?ZBRZ value, 2-447
 ?ZCA function, 2-439
 ?ZCAI offset, 2-448

?ZCAL value, 2-448
 ?ZCAR offset, 2-448
 ?ZCAS offset, 2-448
 ?ZCAZ value, 2-448
 ?ZCL function, 2-439
 ?ZCO function, 2-439
 ?ZCOF offset, 2-451
 ?ZCOI offset, 2-451
 ?ZCOL value, 2-451
 ?ZCOS offset, 2-451
 ?ZCP function, 2-439
 ?ZCPI offset, 2-452
 ?ZCPL value, 2-452
 ?ZCPN offset, 2-452
 ?ZCPR offset, 2-452
 ?ZCPZ value, 2-452
 ?ZCR function, 2-439, 2-451, 2-453, 2-477
 ?ZCRF offset, 2-453
 ?ZCRI offset, 2-453
 ?ZCRL value, 2-453
 ?ZCRN offset, 2-453
 ?ZCRQ offset, 2-453
 ?ZCRR offset, 2-453
 ?ZCRZ value, 2-453
 ?ZCS function, 2-439
 ?ZCS3 offset, 2-449-2-450
 ?ZCSA offset, 2-449-2-450
 ?ZCSB offset, 2-449-2-450
 ?ZCSF offset, 2-449-2-450
 ?ZCSI offset, 2-449-2-450
 ?ZCSK offset, 2-449-2-450
 ?ZCSL value, 2-449-2-450
 ?ZCSN offset, 2-449-2-450
 ?ZCSP offset, 2-449-2-450
 ?ZCSR offset, 2-449-2-450
 ?ZCSU offset, 2-449-2-450
 ?ZCSZ value, 2-450
 ?ZDE function, 2-439
 ?ZDF function, 2-439
 ?ZDFA offset, 2-454
 ?ZDFB offset, 2-454
 ?ZDFE offset, 2-454
 ?ZDFI offset, 2-454
 ?ZDFL value, 2-454
 ?ZDFZ value, 2-454
 ?ZDI function, 2-439
 ?ZDIF offset, 2-455
 ?ZDII offset, 2-455
 ?ZDIL value, 2-455
 ?ZDIN offset, 2-455
 ?ZDIZ value, 2-455
 ?ZDM function, 2-439
 ?ZDM value, 2-471
 ?ZDS function, 2-439
 ?ZDSF offset, 2-456
 ?ZDSI offset, 2-456
 ?ZDSL value, 2-456
 ?ZDSM offset, 2-456
 ?ZDSZ value, 2-456
 ?ZEL function, 2-439
 ?ZELF offset, 2-457
 ?ZELI offset, 2-457
 ?ZELN value, 2-457
 ?ZELR offset, 2-457
 ?ZELZ value, 2-457
 ?ZEN function, 2-439
 ?ZEN3 offset, 2-458-2-459
 ?ZENC offset, 2-458-2-459
 ?ZENF offset, 2-458-2-459
 ?ZENI offset, 2-458-2-459
 ?ZENK offset, 2-458-2-459
 ?ZENL value, 2-458
 ?ZENR offset, 2-458-2-459
 ?ZENT offset, 2-458-2-459
 ?ZENZ value, 2-459
 ?ZEV function, 2-439
 ?ZEVF offset, 2-460
 ?ZEVI offset, 2-460
 ?ZEVL value, 2-460
 ?ZEVN offset, 2-460
 ?ZEVZ value, 2-460
 ?ZFL function, 2-439

?ZFLF offset, 2–461
 ?ZFLI offset, 2–461
 ?ZFLL value, 2–461
 ?ZFLS offset, 2–461
 ?ZFLZ value, 2–461
 ?ZFN function, 2–439
 ?ZFO function, 2–439
 ?ZFOA offset, 2–462
 ?ZFOB offset, 2–462
 ?ZFOF offset, 2–462
 ?ZFOI offset, 2–462
 ?ZFOL value, 2–462
 ?ZFOZ value, 2–462
 ?ZHA function, 2–439
 ?ZHA value, 2–441
 ?ZHAF offset, 2–463
 ?ZHAI offset, 2–463
 ?ZHAL value, 2–463
 ?ZHAR offset, 2–463
 ?ZHAZ value, 2–463
 ?ZHE function, 2–439
 ?ZHEH offset, 2–464
 ?ZHEI offset, 2–464
 ?ZHEL value, 2–464
 ?ZHER offset, 2–464
 ?ZHEZ value, 2–464
 ?ZHO function, 2–439
 ?ZHOI offset, 2–465
 ?ZHOL value, 2–465
 ?ZHOR offset, 2–465
 ?ZHOS offset, 2–465
 ?ZHOZ value, 2–465
 ?ZJ30 value, 2–476
 ?ZJ90 value, 2–480
 ?ZJF1 value, 2–446
 ?ZJH0 value, 2–457
 ?ZJI0 value, 2–460
 ?ZJS1 value, 2–482
 ?ZJV1–?ZJV9 values, 2–496
 ?ZJVD–?ZJVE values, 2–496
 ?ZLI function, 2–439
 ?ZLIA offset, 2–466
 ?ZLIB offset, 2–466
 ?ZLII offset, 2–466
 ?ZLIL value, 2–466
 ?ZLIZ value, 2–466
 ?ZLO function, 2–439
 ?ZLO3–?ZLO6 offsets, 2–467–2–468
 ?ZLOC offset, 2–467–2–468
 ?ZLOF offset, 2–467–2–468
 ?ZLOI offset, 2–467–2–468
 ?ZLOL value, 2–467
 ?ZLOP offset, 2–467–2–468
 ?ZLOZ value, 2–468
 ?ZLP function, 2–439
 ?ZLPI offset, 2–469
 ?ZLPL value, 2–469
 ?ZLPN offset, 2–469
 ?ZLPR offset, 2–469
 ?ZLPZ value, 2–469
 ?ZMD function, 2–439
 ?ZMD value, 2–441
 ?ZME function, 2–439
 ?ZMO function, 2–439
 ?ZMOA offset, 2–472
 ?ZMOB offset, 2–472
 ?ZMOF offset, 2–472
 ?ZMOI offset, 2–472
 ?ZMOL value, 2–472
 ?ZMOM offset, 2–472
 ?ZMOR offset, 2–472
 ?ZMOZ value, 2–472
 ?ZMP function, 2–439
 ?ZMPA offset, 2–470
 ?ZMPB offset, 2–470
 ?ZMPF offset, 2–470
 ?ZMPI offset, 2–470
 ?ZMPL value, 2–470
 ?ZMPZ value, 2–470
 ?ZMS function, 2–439
 ?ZMS0 offset, 2–473
 ?ZMS3–?ZMS9 offsets, 2–473–2–474

?ZMSA offset, 2-473
 ?ZMSD offset, 2-473
 ?ZMSE offset, 2-473-2-474
 ?ZMSF offset, 2-473-2-474
 ?ZMSG offset, 2-473-2-474
 ?ZMSI offset, 2-473-2-474
 ?ZMSK offset, 2-473-2-474
 ?ZMSL value, 2-473
 ?ZMSM offset, 2-473
 ?ZMSP offset, 2-473-2-474
 ?ZMSQ value, 2-474
 ?ZMSR offset, 2-473
 ?ZMSS offset, 2-473
 ?ZMST offset, 2-473
 ?ZMSU offset, 2-473
 ?ZMSV offset, 2-473
 ?ZMSX offset, 2-473
 ?ZMSZ offset, 2-473
 ?ZON function, 2-439
 ?ZOP function, 2-439
 ?ZOPF offset, 2-476
 ?ZOPI offset, 2-476
 ?ZOPL value, 2-476
 ?ZOPR offset, 2-476
 ?ZOPZ value, 2-476
 ?ZPA function, 2-439
 ?ZPAF offset, 2-477
 ?ZPAI offset, 2-477
 ?ZPAL value, 2-477
 ?ZPAS offset, 2-477
 ?ZPAZ value, 2-477
 ?ZPE function, 2-440
 ?ZPE3 and ?ZPE4 offsets, 2-478
 ?ZPEF offset, 2-478
 ?ZPEI offset, 2-478
 ?ZPEL value, 2-478
 ?ZPER offset, 2-478
 ?ZPEU offset, 2-478
 ?ZPEV offset, 2-478
 ?ZPEZ value, 2-478
 ?ZPI function, 2-440
 ?ZPIA offset, 2-479
 ?ZPIB offset, 2-479
 ?ZPIF offset, 2-479
 ?ZPII offset, 2-479
 ?ZPIL value, 2-479
 ?ZPIR offset, 2-479
 ?ZPIZ value, 2-479
 ?ZPR function, 2-440
 ?ZPR value, 2-441
 ?ZPRF offset, 2-480
 ?ZPRI offset, 2-480
 ?ZPRL value, 2-480
 ?ZPRR offset, 2-480
 ?ZPRZ value, 2-480
 ?ZPU function, 2-440
 ?ZQP function, 2-440
 ?ZQPA offset, 2-481-2-482
 ?ZQPB offset, 2-481-2-482
 ?ZQPC offset, 2-481-2-482
 ?ZQPF offset, 2-481-2-482
 ?ZQPH offset, 2-481
 ?ZQPI offset, 2-481-2-482
 ?ZQPL value, 2-481
 ?ZQPN offset, 2-481
 ?ZQPO offset, 2-481
 ?ZQPP offset, 2-481
 ?ZQPS offset, 2-481
 ?ZQPU offset, 2-481
 ?ZQPX value, 2-481
 ?ZQPZ value, 2-482
 ?ZRE function, 2-440
 ?ZREF offset, 2-484
 ?ZREI offset, 2-484
 ?ZREL value, 2-484
 ?ZRER offset, 2-484
 ?ZREZ value, 2-484
 ?ZRF function, 2-440
 ?ZRF value, 2-441
 ?ZRFF offset, 2-483
 ?ZRFI offset, 2-483
 ?ZRFL value, 2-483

?ZRFM offset, 2-483
 ?ZRFZ value, 2-483
 ?ZRT function, 2-440
 ?ZRTA offset, 2-485
 ?ZRTB offset, 2-485
 ?ZRTI offset, 2-485
 ?ZRTL value, 2-485
 ?ZRTZ value, 2-485
 ?ZS1B offset, 2-487-2-490
 ?ZS1C offset, 2-487-2-490
 ?ZS1F offset, 2-487-2-490
 ?ZS1L value, 2-487
 ?ZS1M offset, 2-487-2-490
 ?ZS1P offset, 2-487-2-490
 ?ZS1S offset, 2-487-2-490
 ?ZSI function, 2-440
 ?ZSIF offset, 2-486
 ?ZSII offset, 2-486
 ?ZSIL value, 2-486
 ?ZSIS offset, 2-486
 ?ZSIZ value, 2-486
 ?ZSK function, 2-440
 ?ZSK3 offset, 2-490
 ?ZSKF offset, 2-490
 ?ZSKI offset, 2-490
 ?ZSKL value, 2-490
 ?ZSKR offset, 2-490
 ?ZSKZ value, 2-491
 ?ZSP function, 2-440
 ?ZSP3-?ZSP9 offsets, 2-487-2-488
 ?ZSPB offset, 2-487-2-488
 ?ZSPC offset, 2-487-2-488
 ?ZSPD offset, 2-487-2-488
 ?ZSPE offset, 2-487-2-490
 ?ZSPF offset, 2-487-2-488
 ?ZSPH offset, 2-487-2-488
 ?ZSPI offset, 2-487-2-488
 ?ZSPK offset, 2-487-2-488
 ?ZSPN offset, 2-487-2-488
 ?ZSPP offset, 2-487-2-488
 ?ZSPQ offset, 2-487-2-488
 ?ZSPR offset, 2-487-2-490
 ?ZSPS offset, 2-487-2-488
 ?ZSPT offset, 2-487-2-488
 ?ZSPV offset, 2-487-2-488
 ?ZSPZ value, 2-488
 ?ZSR function, 2-440, **2-492**
 ?ZSR3-?ZSR7 offsets, 2-492
 ?ZSRB offset, 2-492
 ?ZSRC offset, 2-492
 ?ZSRF offset, 2-492
 ?ZSRI offset, 2-492
 ?ZSRL value, 2-492
 ?ZSRM offset, 2-492
 ?ZSRQ offset, 2-492
 ?ZSRR offset, 2-492
 ?ZSRS offset, 2-492
 ?ZSRX offset, 2-492
 ?ZSRZ value, 2-493
 ?ZSS function, 2-440
 ?ZSS0-?ZSS9 offsets, 2-495, 2-497-2-498
 ?ZSSA-?ZSSZ offsets, 2-495-2-498
 ?ZST function, 2-440
 ?ZSTA offset, 2-499
 ?ZSTB offset, 2-499
 ?ZSTF offset, 2-499
 ?ZSTI offset, 2-499
 ?ZSTL value, 2-499
 ?ZSTZ value, 2-499
 ?ZSX0-?ZSX9 offsets, 2-495-2-498
 ?ZTE function, 2-440
 ?ZTR function, 2-440
 ?ZTRI offset, 2-500
 ?ZTRL value, 2-500
 ?ZTRR offset, 2-500
 ?ZTRT offset, 2-500
 ?ZTRZ value, 2-500
 ?ZUC function, 2-440
 ?ZUH function, 2-440
 ?ZUHI offset, 2-501
 ?ZUHL value, 2-501
 ?ZUHR offset, 2-501

?ZUHS offset, 2-501
 ?ZUHZ value, 2-501
 ?ZUL function, 2-440
 ?ZULF offset, 2-504
 ?ZULI offset, 2-504
 ?ZULL value, 2-504
 ?ZULS offset, 2-504
 ?ZULZ value, 2-504
 ?ZUN function, 2-440
 ?ZUNF offset, 2-505
 ?ZUNI offset, 2-505
 ?ZUNL value, 2-505
 ?ZUNS offset, 2-505
 ?ZUNZ value, 2-505
 ?ZUS function, 2-440
 ?ZUS3-?ZUS7 offsets, 2-502-2-503
 ?ZUSF offset, 2-502-2-503
 ?ZUSI offset, 2-502-2-503
 ?ZUSK offset, 2-502-2-503
 ?ZUSL value, 2-502
 ?ZUSM offset, 2-502-2-503
 ?ZUSP offset, 2-502-2-503
 ?ZUSR offset, 2-502-2-503
 ?ZUSU offset, 2-502-2-503
 ?ZUSV offset, 2-502-2-503
 ?ZUSZ value, 2-503
 ?ZVE function, 2-440
 ?ZVEF offset, 2-508
 ?ZVEI offset, 2-508
 ?ZVEL value, 2-508
 ?ZVES offset, 2-508
 ?ZVEZ value, 2-508
 ?ZXB function, 2-440
 ?ZXFG offset, 2-438, 2-441
 ?ZXFU offset, 2-438-2-440
 ?ZXID offset, 2-438, 2-439
 ?ZXIZ value, 2-439
 ?ZXLN value, 2-438
 ?ZXNA offset, 2-438, 2-441
 ?ZXNB offset, 2-438, 2-441
 ?ZXNL offset, 2-438, 2-441
 ?ZXRS offset, 2-438, 2-441
 ?ZXSP offset, 2-438, 2-441
 ?ZXTP offset, 2-438, 2-441
 ?ZY00 value, 2-456
 ?ZY10 value, 2-472
 ?ZY20-?ZY29 values, 2-474
 ?ZY2G-?ZY2J values, 2-474
 ?ZY30 value, 2-476
 ?ZY40 value, 2-478
 ?ZY50 value, 2-483
 ?ZY60 value, 2-484
 ?ZY70-?ZY74 values, 2-503
 ?ZY80-?ZY87 values, 2-468
 ?ZY90 value, 2-480
 ?ZY9G-?ZY9I values, 2-442
 ?ZYG0-?ZYG9 values, 2-450
 ?ZYC0-?ZYC2 values, 2-455
 ?ZYD0-?ZYD9 values, 2-459
 ?ZYDA and ?ZYDB values, 2-459
 ?ZYE0 value, 2-443
 ?ZYF0-?ZYF2 values, 2-446
 ?ZYG0 value, 2-454
 ?ZYH0 value, 2-457
 ?ZYI0 value, 2-460
 ?ZYJ0 value, 2-462
 ?ZYJ2 value, 2-446
 ?ZYK0-?ZYK9 values, 2-488
 ?ZYKA and ?ZYKB values, 2-488
 ?ZYL0 value, 2-444
 ?ZYM0 value, 2-445
 ?ZYMF value, 2-470
 ?ZYN0 value, 2-447
 ?ZY00 value, 2-451
 ?ZYP0 value, 2-453
 ?ZYQ0 value, 2-477
 ?ZYR0-?ZYR2 values, 2-479
 ?ZYS0 and ?ZYS1 values, 2-482
 ?ZYT0 value, 2-486
 ?ZYU0-?ZYU8 values, 2-493
 ?ZYV0-?ZYV9 values, 2-496
 ?ZYVA-?ZYVE values, 2-496

?ZYW0 value, 2-499
?ZYX0 value, 2-504
?ZYY0 value, 2-505
?ZYZ0 value, 2-508
?ZZ00 value, 2-456
?ZZ10 value, 2-472
?ZZ20-?ZZ29 values, 2-474
?ZZ2G-?ZZ2J values, 2-474
?ZZ30 value, 2-476
?ZZ40 value, 2-478
?ZZ50 value, 2-483
?ZZ60 value, 2-484
?ZZ70-?ZZ74 values, 2-503
?ZZ80-?ZZ87 values, 2-468
?ZZ90 value, 2-480
?ZZ9G-?ZZ9I values, 2-442
?ZZB0-?ZZB9 values, 2-450
?ZZC0-?ZZC2 values, 2-455
?ZZD0-?ZZD9 values, 2-459
?ZZDA and ?ZZDB values, 2-459
?ZZE0 value, 2-443
?ZZF0-?ZZF2 values, 2-446
?ZZG0 value, 2-454

?ZZH0 value, 2-457
?ZZI0 value, 2-460
?ZZJ0 value, 2-462
?ZZK0-?ZZK9 values, 2-488
?ZZKA-?ZZKC values, 2-488
?ZZL0 value, 2-444
?ZZM0 value, 2-445
?ZZMF value, 2-470
?ZZN0 value, 2-447
?ZZO0 value, 2-451
?ZZP0 value, 2-453
?ZZQ0 value, 2-477
?ZZR0-?ZZR2 values, 2-479
?ZZS0 and ?ZZS1 values, 2-482
?ZZT0 value, 2-486
?ZZU0-?ZZU8 values, 2-493
?ZZV0-?ZZV9 values, 2-496
?ZZVA-?ZZVE values, 2-496
?ZZW0 value, 2-499
?ZZX0 value, 2-504
?ZZY0 value, 2-505
?ZZZ0 value, 2-508

Document Set

For Users

AOS/VS and AOS/VS II Glossary (069-000231)

For all users, this manual defines important terms used in AOS/VS and AOS/VS II manuals, both regular and preinstalled.

Learning to Use Your AOS/VS System (069-000031)

A primer for all users, this manual introduces AOS/VS (but the material applies to AOS/VS II) through interactive sessions with the CLI, the SED and SPEED text editors, programming languages, Assembler, and the Sort/Merge utility.

Using the CLI (AOS and AOS/VS) is a good follow-up.

SED Text Editor User's Manual (AOS and AOS/VS) (093-000249)

For all users, this manual explains how to use SED, an easy-to-use screen-oriented text editor that lets you program function keys to make repetitive tasks easier. The *SED Text Editor* template (093-000361) accompanies this manual.

Using the AOS/VS System Management Interface (SMI) (069-000203)

Using the AOS/VS II System Management Interface (SMI) (069-000311)

For those working with preinstalled systems and those on regular systems who want an alternative to the CLI, the SMI is an easy-to-use, menu-driven program that helps with system management functions and some file maintenance tasks.

Using the CLI (AOS/VS and AOS/VS II) (093-000646)

For all users, this manual explains the AOS/VS and AOS/VS II file and directory structure and how to use the CLI, a command line interpreter, as the interface to the operating system. This manual explains how to use the CLI macro facility, and includes a dictionary of CLI commands and pseudomacros.

For System Managers and Operators

AOS/VS and AOS/VS II Error and Status Messages (093-000540)

For all users, but especially for system managers and operators of regular systems, this manual lists error and status messages, their source and meaning, and appropriate responses. This manual complements *Installing, Starting, and Stopping AOS/VS*; *Installing, Starting, and Stopping AOS/VS II*; and *Managing AOS/VS and AOS/VS II*.

AOS/VS and AOS/VS II Menu-Based Utilities (093-000650)

A keyboard template to identify function keys. A number of system management programs—such as Disk Jockey, VSGEN, and the SMI—and the BROWSE utility use the function keys identified on this template.

Information Update: Starting Your ECLIPSE MV/1000 DC (014–001728)

Updates Starting and Updating Preinstalled AOS/VS and Starting and Updating Preinstalled AOS/VS II.

Installing, Starting, and Stopping AOS/VS (093–000675)

Installing, Starting, and Stopping AOS/VS II (093–000539)

For system managers and operators of regular (as opposed to preinstalled) systems, these manuals explain the steps necessary to format disks, install a tailored operating system, create the multiuser environment, update the system or microcode, and routinely start up and shut down the system. *AOS/VS and AOS/VS II Error and Status Messages* and *Managing AOS/VS and AOS/VS II* are companions to these manuals.

Managing AOS/VS and AOS/VS II (093–000541)

For system managers and operators, this manual explains managing an AOS/VS or AOS/VS II system. Managing tasks include such topics as editing user profiles, managing the multiuser environment with the EXEC program, backing up and restoring files, using runtime tools, and so forth. This manual complements the “Installing” manuals, whether for regular or preinstalled systems.

Starting and Updating Preinstalled AOS/VS (069–000293)

Starting and Updating Preinstalled AOS/VS II (069–000294)

For those working with preinstalled (as opposed to regular) operating systems on all computers except ECLIPSE MV/3500™ DC and MV/5000 Series systems, these manuals explain how to start, update, and change certain system parameters. The manuals also help you interpret error messages and codes. Companion manuals are *Using the AOS/VS System Management Interface* and *Using the AOS/VS II System Management Interface*.

Starting and Updating Preinstalled AOS/VS on ECLIPSE MV/3500™ DC and MV/5000™ DC Series Systems (069–000481)

Starting and Updating Preinstalled AOS/VS II on ECLIPSE MV/3500™ DC and MV/5000™ DC Series Systems (069–000480)

For those working with preinstalled (as opposed to regular) operating systems on ECLIPSE MV/3500™ DC and MV/5000™ DC Series computers, these manuals explain how to start, update, and change certain system parameters. The manuals also help you interpret error messages and codes. Companion manuals are *Using the AOS/VS System Management Interface* and *Using the AOS/VS II System Management Interface*.

If you have one of these computer systems, use the pertinent manual above; discard any other *Starting and Updating Preinstalled* manuals you receive.

Using the AOS/VS System Management Interface (SMI) (069–000203)

Using the AOS/VS II System Management Interface (SMI) (069–000311)

For those working with preinstalled systems and those on regular systems who want an alternative to the CLI, the SMI is an easy-to-use, menu-driven program that helps with system management functions and some file maintenance tasks.

For Programmers

AOS/VS, AOS/VS II, and AOS/RT32 System Call Dictionary, ?A through ?Q (093–000542)

AOS/VS, AOS/VS II, and AOS/RT32 System Call Dictionary, ?R through ?Z (093–000543)

For system programmers and application programmers who use system calls, this two-volume manual provides detailed information about system calls, including their use, syntax, accumulator input and output values, parameter packets, and error codes. *AOS/VS System Concepts* is a companion manual.

AOS/VS Debugger and File Editor User's Manual (093–000246)

For assembly language programmers, this manual describes using the AOS/VS and AOS/VS II debugger for examining program files, and the file editor FED for examining and modifying locations in any kind of disk file, including program and text files. The *AOS/VS Debug/FED* template (093–000396) accompanies this manual.

AOS/VS Link and Library File Editor (LFE) User's Manual (093–000245)

For AOS/VS and AOS/VS II programmers, this manual describes the Link utility, which builds executable program files from object modules and library files, and which can also be used to create programs to run under the AOS, MP/AOS, RDOS, RTOS, or DG/UX™ operating systems. This manual also describes the Library File Editor utility, LFE, for creating, editing, and analyzing library files; and the utilities CONVERT and MKABS, for manipulating RDOS and RTOS files.

AOS/VS Macroassembler (MASM) Reference Manual (093–000242)

For assembly language programmers, this reference manual describes the use and operation of the MASM utility, which works under AOS/VS and AOS/VS II.

AOS/VS System Concepts (093–000335)

For system programmers and application programmers who write assembly-language subroutines, this manual explains basic AOS/VS system concepts, most of which apply to AOS/VS II as well. This manual complements both volumes of the *AOS/VS, AOS/VS II, and AOS/RT32 System Call Dictionary*.

SPEED Text Editor (AOS and AOS/VS) User's Manual (093–000197)

For programmers, this manual explains how to use SPEED, a powerful (but unforgiving) character-oriented text editor.

Other Related Documents

AOS/VS and AOS/VS II Performance Package User's Manual (093–000364)

For system managers, this manual explains how to use the AOS/VS and AOS/VS II Performance Package (Model 30718), a separate product that is useful for analyzing and perhaps improving the performance of AOS/VS and AOS/VS II systems.

Backing Up and Restoring Files With DUMP_3/LOAD_3 (093–000561)

For system managers, operators, and experienced users, this manual explains the DUMP_3/LOAD_3 product, separately available, which provides backup and enhanced restoration functions, including precise indexing of files on a backup tape set.

Configuring and Managing the High-Availability Disk-Array/MV (H.A.D.A./MV) Subsystem
(014-002160)

For system managers of the H.A.D.A./MV subsystem (a separate product), this manual explains how to configure, operate, and replace subsystem controllers, disk modules, and tape modules. This manual also explains how to replace fans, power supplies, and other subsystem hardware.

Configuring Your Network with XTS (093-00689)

For network administrators, managers, or operators responsible for designing, configuring, or maintaining a network management system, this manual describes how to manage and operate Data General's XODIAC™ Transport Service (XTS and XTS II) under AOS/VS and AOS/VS II.

Installing and Administering DG TCP/IP (093-701027)

For network managers and operators, this manual explains how to install and manage a TCP/IP network under AOS/VS.

Managing AOS/VS II ONC™ /NFS® Services (093-000667)

For network managers and operators, this manual explains how to install and manage an ONC Network File server software under AOS/VS II.

Managing AOS/VS II TCP/IP (093-000704)

For network managers and operators, this manual explains how to install and manage a TCP/IP network under AOS/VS II.

Managing and Operating the XODIAC™ Network Management System (093-000260)

For network managers and operators, this manual describes how to install and manage the Data General proprietary network software.

Managing XTS II with DG/OpenNMS (093-000698)

For network managers and operators, this manual explains how to use DG/OpenNMS to manage the XTS II transport service for large communications networks. It also identifies the XTS II components and explains how to use the NMI menus and screens to manage the XTS II subsystems and the Message Transport Agent (MTA).

*Managing Your DG/PC*Integration Network with DG/ONMS* (093-000624)

For network managers, this manual explains how to manage XTS II and DG/PC*Integration components with DG/OpenNMS.

Managing Your Network with DG/OpenNMS (093-000486)

For network managers, administrators, and operators, this manual describes how to use the DG/OpenNMS software. It also explains how to load the software, create the DG/OpenNMS environment, and use the Network Management Interface (NMI) to manage the network.

Managing Your XODIAC™ Network with DG/ONMS (093-000625)

For network managers, this manual explains how to manage XTS II, MTA, and the XODIAC agents (FTA, RMA, and SVTA) with DG/OpenNMS.

Programming with the Remote Procedure Call (RPC) on AOS/VS II (093-000770)

For experienced network programmers, this manual provides information necessary to write the Remote Procedure Call for the AOS/VS II UDP/IP and TCP/IP networks.

Using CLASP (Class Assignment and Scheduling Package) (093-000422)

For system managers, this manual explains how to use the AOS/VS and AOS/VS II Class Assignment and Scheduling Package (Model 31134), a separate product that is useful for tailoring process scheduling to the needs of a specific site.

Using the MV Data Center Manager (093-000769)

For system managers, this manual explains how to use the MV Data Center Manager software, a separate product that manages multiple ECLIPSE MV/Family computers from an AViiON workstation.

End of Document Set

