

RRRRRRRRRRR		TTTTTTTTTTTT	PPPPPPPPPPP		AAAAAAAAA		DDDDDDDDDDD	
RRRRRRRRRRR		TTTTTTTTTTTT	PPPPPPPPPPP		AAAAAAAAA		DDDDDDDDDDD	
RRRRRRRRRRR		TTTTTTTTTTTT	PPPPPPPPPPP		AAAAAAAAA		DDDDDDDDDDD	
RRR	FRR	TTT	PPP	PPP	AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDD	DDD
RRRRRRRRRRR		TTT	PPPPPPPPPPP		AAA	AAA	DDD	DDD
RRRRRRRRRRR		TTT	PPPPPPPPPPP		AAA	AAA	DDD	DDD
RRRRRRRRRRR		TTT	PPPPPPPPPPP		AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAAAAAAAAAAAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAAAAAAAAAAAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAAAAAAAAAAAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDDDDDDDDDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDDDDDDDDDD	DDD
RRR	RRR	TTT	PPP	PPP	AAA	AAA	DDDDDDDDDDD	DDD

```

RRRRRRR      SSSSSSS  XX      XX  RRRRRRR  TTTTTTTTTT
RRRRRRR      SSSSSSS  XX      XX  RRRRRRR  TTTTTTTTTT
RR      RR  SS      XX      XX  RR      RR  TT
RR      RR  SS      XX      XX  RR      RR  TT
RR      RR  SS      XX      XX  RR      RR  TT
RR      RR  SS      XX      XX  RR      RR  TT
RRRRRRR      SSSSSS   XX      XX  RRRRRRR  TT
RRRRRRR      SSSSSS   XX      XX  RRRRRRR  TT
RR  RR      SS      XX      XX  RR  RR  TT
RR  RR      SS      XX      XX  RR  RR  TT
RR      RR  SS      XX      XX  RR      RR  TT
RR      RR  SS      XX      XX  RR      RR  TT
RR      RR  SSSSSSS  XX      XX  RR      RR  TT
RR      RR  SSSSSSS  XX      XX  RR      RR  TT

```

```

LL      I11111  SSSSSSS
LL      I11111  SSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL  I11111  SSSSSSS
LLLLLLLLLL  I11111  SSSSSSS

```

: f
:

....
....
....
....

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57

```

0001 0 MODULE RSXRT (
0002 0     IDENT = 'V04-000',
0003 0     ADDRESSING_MODE(EXTERNAL=GENERAL)
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 * ALL RIGHTS RESERVED.
0012 1 *
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 * TRANSFERRED.
0019 1 *
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 * CORPORATION.
0023 1 *
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *
0028 1 *****
0029 1
0030 1 ++
0031 1
0032 1 FACILITY: REMOTE TERMINAL SUPPORT
0033 1
0034 1 ABSTRACT:
0035 1     THIS PROGRAM SUPPORTS THE RSX-11M REMOTE TERMINAL PROTOCOL.
0036 1
0037 1
0038 1 ENVIRONMENT:
0039 1
0040 1     VAX/VMS Operating System
0041 1
0042 1 --
0043 1
0044 1
0045 1 AUTHOR: W M CARDOZA, CREATION DATE: 2-JAN-80
0046 1
0047 1 MODIFIED BY:
0048 1
0049 1     V03-003 WMC0002 Wayne Cardoza 28-Feb-1984
0050 1     Fix check for cancel-all.
0051 1
0052 1     V03-002 MHB0081 Mark Bramhall 1-Sep-1982
0053 1     Use IOS_TTYREADALL instead of IOS_READPBLK.
0054 1
0055 1     V03-001 WMC0001 Wayne Cardoza 6-May-1982
0056 1     Check for valid CURRENTIO in CANCEL.
0057 1

```

```

58 0058 1 !**
59 0059 1 LIBRARY 'SYSS$LIBRARY:LIB';
60 0060 1 LIBRARY 'SYSS$LIBRARY:CLIMAC';
61 0061 1
62 0062 1
63 0063 1
64 0064 1 FORWARD ROUTINE
65 0065 1 GETTERMCHAR: NOVALUE,
66 0066 1 GETBUF,
67 0067 1 FREEBUF,
68 0068 1 INDREAD,
69 0069 1 LINKRECV: NOVALUE,
70 0070 1 WRITE: NOVALUE,
71 0071 1 TERMMBXMSG: NOVALUE,
72 0072 1 READ: NOVALUE,
73 0073 1 CNTRLCAST: NOVALUE,
74 0074 1 CNTRLFAST: NOVALUE,
75 0075 1 READSINGLE: NOVALUE,
76 0076 1 ATTACH: NOVALUE,
77 0077 1 RSXRT: NOVALUE,
78 0078 1 LINKMBXMSG: NOVALUE,
79 0079 1 BROADCAST: NOVALUE,
80 0080 1 READPROMPT: NOVALUE,
81 0081 1 QIODONE: NOVALUE,
82 0082 1 CANCEL: NOVALUE,
83 0083 1 TERMINATOR,
84 0084 1 UNSUPPORTED: NOVALUE,
85 0085 1 MAPMODIFIER,
86 0086 1 LINKWRTDONE: NOVALUE,
87 0087 1 NEXTIO: NOVALUE,
88 0088 1 UNSDATENBL: NOVALUE,
89 0089 1 ONECHAR: NOVALUE;
90 0090 1
91 0091 1
92 0092 1 MACRO
93 0093 1 RTP_BUF = BLOCK[32] FIELD(RTP_FIELDS) %,
94 0094 1 QUIT = BEGIN
95 0095 1 $SETAST (ENBFLG = 0); ! STOP EVERYTHING
96 0096 1 WAKEFLAG = 1;
97 0097 1 $WAKE(); ! WAKE UP BASE LEVEL
98 0098 1 RETURN;
99 0099 1 END %,
100 0100 1 QUIT_ON_ERROR = IF (.RETSTATUS AND 1) EQL 0 THEN
101 0101 1 QUIT %;
102 0102 1
103 0103 1 EQUATED SYMBOLS:
104 0104 1
105 0105 1 LITERAL
106 0106 1 ! FUNCTION CODES
107 0107 1 RF_NOP = 0,
108 0108 1 RF_SSD = 1,
109 0109 1 RF_DIS = 2,
110 0110 1 RF_WTD = 3,
111 0111 1 RF_RDD = 4,
112 0112 1 RF_WRD = 5,
113 0113 1 RF_UNSC = 6,
114 0114 1 RF_RSC = 7,
! NOP
! CONFIGURATION
! DISCONNECT
! WRITE DATA
! READ DATA
! READ WITH PROMPT
! UNSOLICITED INPUT DISABLE/ENABLE
! READ SINGLE CHARACTERS

```

```
115 0115 1 RF_KIL = 8. : CANCEL I/O
116 0116 1 RF_ATT = 9. : ATTACH
117 0117 1 RF_GTC = 10. : GET TERMINAL CHARACTERISTICS
118 0118 1 RF_STC = 11. : SET TERMINAL CHARACTERISTICS
119 0119 1 RF_ECR = 12. : EXCEPTION CONDITION
120 0120 1 ! MODIFIERS
121 0121 1 RM_WBN = 1. : WRITE BINARY
122 0122 1 RM_WBT = 2. : BROADCAST
123 0123 1 RM_RBN = 4. : READ BINARY
124 0124 1 RM_RTC = 8. : READ TERMINATES ON CONTROL CHARACTERS
125 0125 1 RM_RNE = 16. : READ NO ECHO
126 0126 1 RM_RTO = 32. : RESET TIME OUT ON EACH CHARACTER
127 0127 1 RM_DET = 128. : DETACH TERMINAL
128 0128 1 RM_NWC = 128. : NO WRITE COMPLETE STATUS
129 0129 1 RM_TUI = 128. : TERMINATE UNSOLICITED INPUT
130 0130 1 RM_TSC = 128. : TERMINATE SINGLE CHARACTER INPUT
131 0131 1 ! FLAGS
132 0132 1 RM_PRI = 2. : PROCESS REQUEST IMMEDIATELY
133 0133 1 RM_CAO = 4. : CANCEL ABORT OUTPUT
134 0134 1 ! STATUS CODES
135 0135 1 RS_SFC = 0. : SUCCESS
136 0136 1 RS_FPE = 1. : FUNCTION PROCESSING ERROR
137 0137 1 RS_UFC = 2. : UNSUPPORTED FUNCTION
138 0138 1 RS_IPF = 3. : ILLEGAL PROTOCOL FUNCTION
139 0139 1 RS_IPD = 4. : ILLEGAL PROTOCOL DATA
140 0140 1 RS_ICF = 5. : ILLEGAL CHARACTERISTICS FUNCTION
141 0141 1 ! TERMINAL CHARACTERISTIC CODES
142 0142 1 RC_HHT = 18. : HARDWARE TABS
143 0143 1 RC_NEC = 19. : NO ECHO
144 0144 1 RC_TTP = 22. : TERMINAL TYPE
145 0145 1 RC_SCP = 23. : CRT
146 0146 1 RC_BIN = 24. : BINARY MODE
147 0147 1 RC_TPL = 28. : PAGE LENGTH
148 0148 1 RC_MAX = 28. : ***** KEEP THIS THE MAXIMUM *****
149 0149 1 ! EXCEPTION CONDITION CODES
150 0150 1 RE_SAR = 0. : SYSTEM ATTENTION REQUEST
151 0151 1 FIELD
152 0152 1 RTP_FIELDS = : REMOTE TERMINAL PROTOCOL
153 0153 1 SET
154 0154 1 RTP_LNK = [0,0,32,0]. : QUEUE LINK WORDS
155 0155 1 RTP_LN2 = [1,0,32,0].
156 0156 1 RTP_IOS = [2,0,16,0]. : IOSB
157 0157 1 RTP_IOC = [2,16,16,0]. : I/O COUNT
158 0158 1 RTP_IO2 = [3,0,32,0].
159 0159 1 RTP_FNC = [4,0,8,0]. : FUNCTION CODE
160 0160 1 RTP_MOD = [4,8,8,0]. : FUNCTION MODIFIER BITS
161 0161 1 RTP_FLG = [4,16,8,0]. : FUNCTION FLAGS
162 0162 1 RTP_STS = [4,24,8,0]. : RETURN STATUS
163 0163 1 RTP_IDN = [5,0,8,0]. : IDENTIFIER
164 0164 1 RTP_RSV = [5,8,8,0]. : RESERVED, MBZ
165 0165 1 RTP_RCT = [5,16,16,0]. : RECEIVE BYTE COUNT
166 0166 1 RTP_TCT = [6,0,16,0]. : TRANSMIT BYTE COUNT
167 0167 1 RTP_DAT = [6,16,32,0]. : DATA
168 0168 1 TES:
169 0169 1 :
170 0170 1 :
171 0171 1 :
```

```

172      0172 1 OWN
173      0173 1
174      0174 1 NAMEIOSB: VECTOR[4,WORD],
175      0175 1 VMSCONFIG: INITIAL(PLIT BYTE(RF_SSD,1,0,0, ! CONFIGURATION MSG
176      0176 1 WORD(4,2), ! PROTOCOL SUPPORTED
177      0177 1 WORD(128), ! BUFFER SIZE
178      0178 1 2,1, ! LINE-FEED PREFIXING
179      0179 1 3,1, ! SINGLE CHAR INPUT
180      0180 1 5,1, ! BROADCAST
181      0181 1 7,1, ! CASE CONVERSION
182      0182 1 8,1, ! NO ECHO
183      0183 1 9,1, ! READ TERMINATORS
184      0184 1 10,1, ! CRT'S
185      0185 1 11,1, ! ^R
186      0186 1 12,1, ! READ/WRITE BINARY
187      0187 1 13,1, ! UNSOLICITED INPUT
188      0188 1 127,1, ! VERSION 1 OF PROTOCOL
189      0188 1 0,0)),
189      0189 1 TERMMBXDATA: VECTOR[4,WORD],
190      0190 1 UNSOLENBLFLG: REF RTP_BUF INITIAL(0),
191      0191 1 ATTACHFLAG: BYTE INITIAL(0),
192      0192 1 SINGLEINPROG: BYTE INITIAL(0),
193      0193 1 UNSOLPEND: BYTE INITIAL(0),
194      0194 1 READINPROG: BYTE INITIAL(0),
195      0195 1 SINGLEFLAG: REF RTP_BUF INITIAL(0),
196      0196 1 CURRENTIO: REF RTP_BUF INITIAL(0),
197      0197 1 INDDATA: REF RTP_BUF INITIAL(0),
198      0198 1 IOQUEUE: VECTOR[2] INITIAL(IOQUEUE,IOQUEUE),
199      0199 1 BUFQUEUE: VECTOR[2] INITIAL(BUFQUEUE,BUFQUEUE),
200      0200 1 CNTRLCMSG: VECTOR[4,BYTE] INITIAL(BYTE(RF_ECR,0,0,RE_SAR)),
201      0201 1 LINKMAIL: VECTOR[40,BYTE],
202      0202 1 STERMMASK: VECTOR[4] INITIAL('FFFFFFFF',0,0,'E0000000'),
203      0203 1 STERMDISC: VECTOR[2] INITIAL(16,STERMMASK),
204      0204 1 NTERMMASK: INITIAL('0C002000'),
205      0205 1 NTERMDISC: VECTOR[2] INITIAL(4,NTERMMASK),
206      0206 1 REQ_DSCNTRLY: $CLIREQDESC (RQTYPE=CLISERV, BITNUM=8);
207      0207 1
208      0208 1 ! THIS TELLS REMOTE TERMINAL MAIN PROGRAM WHAT PROTOCOL WE SUPPORT
209      0209 1
210      0210 1 PSECT OWN = PROTOTBL (ALIGN(0));
211      0211 1 OWN
212      0212 1 PROTOMASK: WORD INITIAL(2), ! RSX-11
213      0213 1 RSXADDR: ALIGN(0) INITIAL(RSXRT);
214      0214 1
215      0215 1 ! EXTERNAL REFERENCES:
216      0216 1
217      0217 1 EXTERNAL ROUTINE
218      0218 1 SYS$CLI : ADDRESSING_MODE(LONG_RELATIVE),
219      0219 1 LIB$GET_VM;
220      0220 1 BUILTIN
221      0221 1 INSQUE,
222      0222 1 REMQUE;
223      0223 1 EXTERNAL
224      0224 1 TTYDESC,
225      0225 1 REM$NEFDIS,
226      0226 1 RDWRTCHAN: WORD,
227      0227 1 CNTRLCHAN: WORD,
228      0228 1 TERMMBXCHAN: WORD,

```

RSXRT  
V04-000

J 2  
16-Sep-1984 02:18:51  
14-Sep-1984 13:04:57

VAX-11 Bliss-32 V4.0-742  
DISK\$VMMASTER:[RTPAD.SRC]RSXRT.B32;1 Page 5 (1)

:	229	0229	1	MAILCHAN: WORD,
:	230	0230	1	LINKCHAN: WORD,
:	231	0231	1	SYSINRAB: \$RAB_DECL,
:	232	0232	1	SYSINFAB: \$FAB_DECL,
:	233	0233	1	INDFLAG: BYTE,
:	234	0234	1	WAKEFLAG: BYTE,
:	235	0235	1	RETSTATUS;

RSX  
V04

.....

```

: 237      0236 1 ROUTINE RSXRT: NOVALUE =
: 238      0237 1 ++
: 239      0238 1
: 240      0239 1 Functional Description:
: 241      0240 1     Performs initialization functions for RSX remote terminals.
: 242      0241 1
: 243      0242 1
: 244      0243 1 Calling Sequence:
: 245      0244 1     standard
: 246      0245 1
: 247      0246 1 Input Parameters:
: 248      0247 1     none
: 249      0248 1
: 250      0249 1 Implicit Inputs:
: 251      0250 1     none
: 252      0251 1
: 253      0252 1 Output Parameters:
: 254      0253 1     none
: 255      0254 1
: 256      0255 1 Implicit Outputs:
: 257      0256 1     none
: 258      0257 1
: 259      0258 1 Routines Called:
: 260      0259 1     GETBUF
: 261      0260 1
: 262      0261 1 Routine Value:
: 263      0262 1     none
: 264      0263 1
: 265      0264 1 Signals:
: 266      0265 1     none
: 267      0266 1
: 268      0267 1 Side Effects:
: 269      0268 1     A configuration message is transmitted.
: 270      0269 1     A prompt is displayed on the screen.
: 271      0270 1     Reads are initiated on the terminal mailbox and on the link.
: 272      0271 1
: 273      0272 1 --
: 274      0273 2 BEGIN
: 275      0274 2 LOCAL
: 276      0275 2     BUFFER: REF RTP_BUF;
: 277      0276 2 RETSTATUS =
: 278      P 0277 2 $QIOW (CHAN = .LINKCHAN,      ! SEND CONFIGURATION MESSAGE
: 279      P 0278 2     FUNC = IOS_WRITEVBLK,
: 280      P 0279 2     P1 = .VMSCONFIG,
: 281      0280 2     P2 = 4 * (.VMSCONFIG-4));
: 282      0281 2 QUIT_ON_ERROR;
: 283      0282 2 RETSTATUS =
: 284      P 0283 2 $QIOW (CHAN = .RDWRCHAN,      ! ENABLE UNSOLICITED INPUT
: 285      0284 2     FUNC = IOS_WRITEVBLK+IOSM_ENABLMBX);
: 286      0285 2 QUIT_ON_ERROR;
: 287      0286 2 RETSTATUS =
: 288      P 0287 2 $QIO (CHAN = .TERMMBXCHAN,      ! UNSOLICITED DATA MBX READ
: 289      P 0288 2     FUNC = IOS_READVBLK,
: 290      P 0289 2     ASTADR = TERMMBXMSG,
: 291      P 0290 2     P1 = TERMMBXDATA,
: 292      0291 2     P2 = 8);
: 293      0292 2 QUIT_ON_ERROR;

```



```

: 294
: 295 P 0293 2
: 296 P 0294 2
: 297 P 0295 2
: 298 P 0296 2
: 299 P 0297 2
: 300 P 0298 2
: 301 P 0299 2
: 302 P 0300 2
: 303 P 0301 2
: 304 P 0302 2
: 305 P 0303 2
: 306 P 0304 2
: 307 P 0305 2
: 308 P 0306 2
: 309 P 0307 2
: 310 P 0308 2
: 311 P 0309 2
: 312 P 0310 2
: 313 P 0311 2
: 314 P 0312 2
: 315 P 0313 2
: 316 P 0314 2
: 317 P 0315 2
: 318 P 0316 2
: 319 P 0317 2
: 320 P 0318 2
: 321 P 0319 2
: 322 P 0320 2
: 323 P 0321 2
: 324 P 0322 2
: 325 P 0323 2
: 326 P 0324 2
: 327 P 0325 2
: 328 P 0326 2
: 329 P 0327 2
: 330 P 0328 2
: 331 P 0329 2
: 332 P 0330 2
: 333 P 0331 2
: 334 P 0332 2
: 335 P 0333 2

```

```

RETSTATUS =
$QIO (CHAN = .MAILCHAN, ! LINK MAILBOX READ
      FUNC = IOS_READVBLK,
      ASTADR = LINKMBXMSG,
      P1 = LINKMAIL,
      P2 = 40);
QUIT ON ERROR;
RETSTATUS =
$QIO (CHAN = .CNTRLCHAN, ! HANDLE CONTROL-C
      FUNC = IOS_SETMODE+IOSM_CTRLCAST,
      P1 = CNTRLCAST);
QUIT ON ERROR;
RETSTATUS =
$QIO (CHAN = .CNTRLCHAN, ! HANDLE CONTROL-Y
      FUNC = IOS_SETMODE+IOSM_CTRLCAST,
      P1 = CNTRLCAST);
QUIT ON ERROR;
SYSSCLIT(REQ_DSCNTRLY,0,0); ! DISABLE CLI ^Y
RETSTATUS =
$QIO (CHAN = .RDWRCHAN, ! GIVE AN RSX PROMPT
      FUNC = IOS_WRITEVBLK,
      P1 = UPLIT_BYTE('>'),
      P2 = 1);
QUIT ON ERROR;
IF .INDFLAG NEQ 0 THEN
  BEGIN ! THERE IS AN INDIRECT FILE
    INDDATA = GETBUF(); ! GET BUFFER FOR FILE READ
    SYSINRAB[RAB$L_UBF] = INDDATA[RTP_DAT]; ! BUFFER ADDRESS
    SYSINRAB[RAB$W_USZ] = 100; ! ALLOW 100 CHARACTERS
    INDREAD(); ! READ IT
  END;
BUFFER = GETBUF(); ! REQUEST A BUFFER
RETSTATUS =
$QIO (CHAN = .LINKCHAN, ! WAIT FOR SOMETHING ON LINK
      FUNC = IOS_READVBLK,
      IOSB = BUFFER[RTP_IOS],
      ASTADR = LINKRECV,
      ASTPRM = .BUFFER,
      P1 = BUFFER[RTP_FNC],
      P2 = 128);
QUIT_ON_ERROR;
END;

```

```

.TITLE RSXRT
.IDENT \V04-000\
.PSECT PROTOTBL,NOEXE,0

```

```

0002 00000 PROTOMASK:
00000000' 00002 RSXADDR: .ADDRESS RSXRT
.PSECT $PLITS,NOWRT,NOEXE,2
00 00 00000009 00000 .LONG 9
00 00 01 01 00004 P.AAA: .BYTE 1, 1, 0, 0

```

```

0002 0004 00008      .WORD 4, 2
0080 0000C      .WORD 128
OB 01 0A 01 09 01 08 01 07 01 05 01 03 C1 02 0000E      .BYTE 2, 1, 3, 1, 5, 1, 7, 1, 8, 1, 9, 1, 10, -
00 00 01 7F 01 0D 01 0C 01 0001D      .BYTE 1, 11, 1, 12, 1, 13, 1, 127, 1, 0, 0
00026      .BLKB 2
3E 00028 P.AAB: .ASCII \>\
                                .PSECT $OWNS,NOEXE,2

00000 NAMEIOSB:
                                .BLKB 8
00000000' 00008 VMSCONFIG:
                                .ADDRESS P.AAA
0000C TERMMBXDATA:
                                .BLKB 8
00000000 00014 UNSOLENBLFLG:
                                .LONG 0
00 00018 ATTACHFLAG:
                                .BYTE 0
00 00019 SINGLEINPROG:
                                .BYTE 0
00 0001A UNSOLPEND:
                                .BYTE 0
00 0001B READINPROG:
                                .BYTE 0
00000000 0001C SINGLEFLAG:
                                .LONG 0
00000000 00020 CURRENTIO:
                                .LONG 0
00000000 00024 INDDATA:
                                .LONG 0
00000000' 00000000' 00028 IOQUEUE:
00000000' 00000000' .ADDRESS IOQUEUE, IOQUEUE
00000000' 00000000' 00030 BUFQUEUE:
                                .ADDRESS BUFQUEUE, BUFQUEUE
00 00 00 0C 00038 CNTRLCMSG:
                                .BYTE 12, 0, 0, 0
0003C LINKMAIL:
                                .BLKB 40
E0000000 00000000 00000000 FFFFFFFF 00064 STERMMASK:
                                .LONG -1, 0, 0, -536870912
00000010 00074 STERMDESC:
                                .LONG 16
00000000' 00078 .ADDRESS STERMMASK
0C002000 0007C NTERMMASK:
                                .LONG 201334784
00000004 00080 NTERMDESC:
                                .LONG 4
00000000' 00084 .ADDRESS NTERMMASK
05 00088 REQ_DSCNTRLY:
                                .BYTE 5
00 00089 .BYTE 8
00 0008A .BYTE 0
00 0008B .BYTE 0
00000000 0008C .LONG 0
00000000 00090 .LONG 0, 0
00000000 00098 .LONG 0
00000000 0009C .LONG 0
00000000 000A0 .LONG 0

```

```

.EXTRN SYSSCLI, LIB$GET VM
.EXTRN TTYDESC, REM$ NETDIS
.EXTRN RDWRICHAN, CNTRLCHAN
.EXTRN TERMMBXCHAN, MAILCHAN
.EXTRN LINKCHAN, SYSINRAB
.EXTRN SYSINFAB, INDFLAG
.EXTRN WAKEFLAG, RETSTATUS
.EXTRN SYSSQIOW, SYSSSETAST
.EXTRN SYSSWAKE, SYSSQIO

```

.PSECT \$CODE\$,NOWRT,2

				01FC 00000	RSXRT:	.WORD	Save R2,R3,R4,R5,R6,R7,R8		0236
		58	00000000G	00 9E 00002		MOVAB	CNTRLCHAN, R8		
		57	00000000G	00 9E 00009		MOVAB	RDWRICHAN, R7		
		56	00000000G	00 9E 00010		MOVAB	LINKCHAN, R6		
		55	00000000G	CF 9E 00017		MOVAB	INDDATA, R5		
		54	00000000G	00 9E 0001C		MOVAB	SYSSQIOW, R4		
		53	00000000G	00 9E 00023		MOVAB	SYSSQIO, R3		
		52	00000000G	00 9E 0002A		MOVAB	RETSTATUS, R2		
				7E 7C 00031		CLRQ	-(SP)		0280
				7E 7C 00033		CLRQ	-(SP)		
		50	E4	A5 D0 00035		MOVL	VMSCONFIG, R0		
7E	FC	A0		02 78 00039		ASHL	#2, -4(R0), -(SP)		
				50 DD 0003E		PUSHL	R0		
				7E 7C 00040		CLRQ	-(SP)		
		7E		30 7D 00042		MOVQ	#48, -(SP)		
		7E		66 3C 00045		MOVZWL	LINKCHAN, -(SP)		
				7E D4 00048		CLRL	-(SP)		
		64		0C FB 0004A		CALLS	#12, SYSSQIOW		
		62		50 D0 0004D		MOVL	R0, RETSTATUS		
		61		62 E9 00050		BLBC	RETSTATUS, 1\$		0284
				7E 7C 00053		CLRQ	-(SP)		
				7E 7C 00055		CLRQ	-(SP)		
				7E 7C 00057		CLRQ	-(SP)		
				7E 7C 00059		CLRQ	-(SP)		
				7E D4 0005B		CLRL	-(SP)		
		7E	B0	8F 9A 0005D		MOVZBL	#176, -(SP)		
		7E		67 3C 00061		MOVZWL	RDWRICHAN, -(SP)		
				7E D4 00064		CLRL	-(SP)		
		64		0C FB 00066		CALLS	#12, SYSSQIOW		
		62		50 D0 00069		MOVL	R0, RETSTATUS		
		66		62 E9 0006C		BLBC	RETSTATUS, 2\$		0291
				7E 7C 0006F		CLRQ	-(SP)		
				7E 7C 00071		CLRQ	-(SP)		
				08 DD 00073		PUSHL	#8		
			E8	A5 9F 00075		PUSHAB	TERMMBXDATA		
				7E D4 00078		CLRL	-(SP)		
			0000V	CF 9F 0007A		PUSHAB	TERMMBXMSG		
		7E		31 7D 0007E		MOVQ	#49, -(SP)		
		7E	00000000G	00 3C 00081		MOVZWL	TERMMBXCHAN, -(SP)		
				7E D4 00088		CLRL	-(SP)		
		63		0C FB 0008A		CALLS	#12, SYSSQIO		
		62		50 D0 0008D		MOVL	R0, RETSTATUS		
		62		62 E9 00090		BLBC	RETSTATUS, 3\$		
				7E 7C 00093		CLRQ	-(SP)		0298

		7E	7C	00095	CLRQ	-(SP)		
		28	DD	00097	PUSHL	#40		
	18	A5	9F	00099	PUSHAB	LINKMAIL		
		7E	D4	0009C	CLRL	-(SP)		
	0000V	CF	9F	0009E	PUSHAB	LINKMBXMSG		
7E		31	7D	000A2	MOVQ	#49, -(SP)		
7E	00000000G	00	3C	000A5	MOVZWL	MAILCHAN, -(SP)		
		7E	D4	000AC	CLRL	-(SP)		
63		0C	FB	000AE	CALLS	#12, SYSSQIO		
62		50	D0	000B1	MOVL	R0, RETSTATUS		
67		62	E9	000B4	BLBC	RETSTATUS, 4\$		
		7E	7C	000B7	CLRQ	-(SP)	0303	
		7E	7C	000B9	CLRQ	-(SP)		
		7E	D4	000BB	CLRL	-(SP)		
	0000V	CF	9F	000BD	PUSHAB	CNTRLCAST		
		7E	7C	000C1	CLRQ	-(SP)		
		7E	D4	000C3	CLRL	-(SP)		
7E	0123	8F	3C	000C5	MOVZWL	#291, -(SP)		
7E		68	3C	000CA	MOVZWL	CNTRLCHAN, -(SP)		
		7E	D4	000CD	CLRL	-(SP)		
63		0C	FB	000CF	CALLS	#12, SYSSQIO		
62		50	D0	000D2	MOVL	R0, RETSTATUS		
46		62	E9	000D5	BLBC	RETSTATUS, 4\$		
		7E	7C	000D8	CLRQ	-(SP)	0308	
		7E	7C	000DA	CLRQ	-(SP)		
		7E	D4	000DC	CLRL	-(SP)		
	0000V	CF	9F	000DE	PUSHAB	CNTRLCAST		
		7E	7C	000E2	CLRQ	-(SP)		
		7E	D4	000E4	CLRL	-(SP)		
7E	A3	8F	9A	000E6	MOVZBL	#163, -(SP)		
7E		68	3C	000EA	MOVZWL	CNTRLCHAN, -(SP)		
		7E	D4	000ED	CLRL	-(SP)		
63		0C	FB	000EF	CALLS	#12, SYSSQIO		
62		50	D0	000F2	MOVL	R0, RETSTATUS		
77		62	E9	000F5	BLBC	RETSTATUS, 6\$		
		7E	7C	000F8	CLRQ	-(SP)	0310	
	64	A5	9F	000FA	PUSHAB	REQ_DSCNTRLY		
00000000G	EF	03	FB	000FD	CALLS	#3, SYSSCLI		
		7E	7C	00104	CLRQ	-(SP)	0315	
		7E	7C	00106	CLRQ	-(SP)		
		01	DD	00108	PUSHL	#1		
	0000'	CF	9F	0010A	PUSHAB	P.AAB		
		7E	7C	0010E	CLRQ	-(SP)		
7E		30	7D	00110	MOVQ	#48, -(SP)		
7E		67	3C	00113	MOVZWL	RDWRTCHAN, -(SP)		
		7E	D4	00116	CLRL	-(SP)		
64		0C	FB	00118	CALLS	#12, SYSSQIOW		
62		50	D0	0011B	MOVL	R0, RETSTATUS		
4E		62	E9	0011E	BLBC	RETSTATUS, 6\$		
	00000000G	00	95	00121	TSTB	INDFLAG	0317	
		1D	13	00127	BEQL	5\$		
	0000V	CF	00	FB	00129	CALLS	#0, GETBUF	0319
		65	D0	0012E	MOVL	R0, INDDATA		
00000000G	00	65	1A	C1	00131	ADDL3	#26, INDDATA, SYSINRAB+36	0320
	0000V	00	8F	9B	00139	MOVZBW	#100, SYSINRAB+32	0321
	0000V	CF	00	FB	00141	CALLS	#0, INDREAD	0322
	0000V	CF	00	FB	00146	CALLS	#0, GETBUF	0324
		5\$:						

		7E	7C	0014B		CLRQ	-(SP)		
		7E	7C	0014D		CLRQ	-(SP)		
	7E	80	8F	9A	0014F	MOVZBL	#128, -(SP)		0332
		10	A0	9F	00153	PUSHAB	16(BUFFER)		
			50	DD	00156	PUSHL	BUFFER		
			0000V	CF	9F	00158	PUSHAB	LINKRCV	
			08	A0	9F	0015C	PUSHAB	8(BUFFER)	
				31	DD	0015F	PUSHL	#49	
	7E		66	3C	00161	MOVZWL	LINKCHAN, -(SP)		
			7E	D4	00164	CLRL	-(SP)		
	63		0C	FB	00166	CALLS	#12, SYSSQIO		
	62		50	DC	00169	MOVL	R0, RETSTATUS		
	19		62	E8	0016C	BLBS	RETSTATUS, 7\$		
			7E	D4	0016F	CLRL	-(SP)		
		00000000G	00	01	FB	00171	CALLS	#1, SYSSSETAST	
		00000000G	00	01	90	00178	MOVB	#1, WAKEFLAG	
				7E	7C	0017F	CLRQ	-(SP)	
		00000000G	00	02	FB	00181	CALLS	#2, SYSSWAKE	
				04	00188	7\$: RET			0334

; Routine Size: 393 bytes, Routine Base: \$CODE\$ + 0000

```

0335 1 ROUTINE GETBUF =
0336 1 ++
0337 1
0338 1 Functional Description:
0339 1     Allocate a buffer.
0340 1
0341 1
0342 1 Calling Sequence:
0343 1     standard
0344 1
0345 1 Input Parameters:
0346 1     none
0347 1
0348 1 Implicit Inputs:
0349 1     BUFQUEUE
0350 1
0351 1 Output Parameters:
0352 1     none
0353 1
0354 1 Implicit Outputs:
0355 1     none
0356 1
0357 1 Routines Called:
0358 1     LIB$GET_VM
0359 1
0360 1 Routine Value:
0361 1     buffer address
0362 1
0363 1 Signals:
0364 1     none
0365 1
0366 1 Side Effects:
0367 1     none
0368 1
0369 1 --
0370 2 BEGIN
0371 2 LOCAL
0372 2     BUFADR:;
0373 2     IF REMQUE(.BUFQUEUE,BUFADR) EQL 3 THEN           ! WAS QUEUE EMPTY?
0374 2     LIB$GET_VM(UPLIT(128+16),BUFADR);                 ! GET A BUFFER
0375 2     RETURN .BUFADR;
0376 1 END;

```

```

.PSECT $SPLITS,NOWRT,NOEXE,2
00000090 00029 .BLKB 3
0002C P.AAC: .LONG 144

```

```

.PSECT $CODE$,NOWRT,2
7E 0000' DF OF 0000 GETBUF: .WORD Save nothing : 0335
50 DC 00007 REMQUE @BUFQUEUE, BUFADR : 0373
MOVPSL R0 :

```

RSXRT  
V04-000

E 3  
16-Sep-1984 02:18:51  
14-Sep-1984 13:04:57

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[RTPAD.SRC]RSXRT.B32;1

Page 13  
(3)

50

50

02  
03

01 EF 00009  
50 D1 0000E  
0D 12 00011  
5E DD 00013  
CF 9F 00015  
02 FB 00019  
6E D0 00020 1\$:  
04 00023

EXTZV #1, #2, R0, R0  
CMPL R0, #3  
BNEQ 1\$  
PUSHL SP  
PUSHAB P.AAC  
CALLS #2, LIB\$GET\_VM  
MCVL BUFADR, R0  
RET

00000000G 00  
50

0000'

0374  
0375  
0376

: Routine Size: 36 bytes, Routine Base: \$CODE\$ + 0189

: 379 0377 1

RS  
VO

```

: 381      0378 1 ROUTINE FREEBUF(BUF) =
: 382      0379 1 ++
: 383      0380 1
: 384      0381 1 Functional Description:
: 385      0382 1 Release a buffer.
: 386      0383 1
: 387      0384 1
: 388      0385 1 Calling Sequence:
: 389      0386 1 standard
: 390      0387 1
: 391      0388 1 Input Parameters:
: 392      0389 1 BUF = buffer address
: 393      0390 1
: 394      0391 1 Implicit Inputs:
: 395      0392 1 none
: 396      0393 1
: 397      0394 1 Output Parameters:
: 398      0395 1 none
: 399      0396 1
: 400      0397 1 Implicit Outputs:
: 401      0398 1 BUFQUEUE
: 402      0399 1
: 403      0400 1 Routines Called:
: 404      0401 1 none
: 405      0402 1
: 406      0403 1 Routine Value:
: 407      0404 1 none
: 408      0405 1
: 409      0406 1 Signals:
: 410      0407 1 none
: 411      0408 1
: 412      0409 1 Side Effects:
: 413      0410 1 none
: 414      0411 1
: 415      0412 1 --
: 416      0413 2 BEGIN
: 417      0414 2 INSQUE(.BUF,BUFQUEUE)
: 418      0415 1 END;

```

```

0000' CF      04      0000 0000 FREEBUF: .WORD      Save nothing      : 0378
                    BC 0E 00004      CLR      R0      : 0414
                    02 12 0000A      INSQUE @BUF, BUFQUEUE
                    50 D6 0000C      BNEQ 1$
                    04 0000E 1$:      INCL R0
                                RET

```

; Routine Size: 15 bytes, Routine Base: \$CODE\$ + 01AD



```

: 420 0416 1 ROUTINE LINKREC'V(BUFFER): NOVALUE =
: 421 0417 1 **
: 422 0418 1
: 423 0419 1 Functional Description:
: 424 0420 1 Receive a message on the link and call the correct service routine.
: 425 0421 1
: 426 0422 1
: 427 0423 1 Calling Sequence:
: 428 0424 1 standard
: 429 0425 1
: 430 0426 1 Input Parameters:
: 431 0427 1 BUFFER = input buffer address
: 432 0428 1
: 433 0429 1 Implicit Inputs:
: 434 0430 1 none
: 435 0431 1
: 436 0432 1 Output Parameters:
: 437 0433 1 none
: 438 0434 1
: 439 0435 1 Implicit Outputs:
: 440 0436 1 RETSTATUS
: 441 0437 1
: 442 0438 1 Routines Called:
: 443 0439 1 WRITE
: 444 0440 1 READ
: 445 0441 1 READPROMPT
: 446 0442 1 UNSDATENBL
: 447 0443 1 CANCEL
: 448 0444 1 READSINGLE
: 449 0445 1 ATTACH
: 450 0446 1 UNSUPPORTED
: 451 0447 1 GETBUF
: 452 0448 1 GETTERMCHAR
: 453 0449 1
: 454 0450 1 Routine Value:
: 455 0451 1 none
: 456 0452 1
: 457 0453 1 Signals:
: 458 0454 1 none
: 459 0455 1
: 460 0456 1 Side Effects:
: 461 0457 1 A new read to the link is initiated.
: 462 0458 1 If there is an error on the read, a $WAKE is issued to force the
: 463 0459 1 program to exit.
: 464 0460 1
: 465 0461 1 --
: 466 0462 2 BEGIN
: 467 0463 2 LOCAL
: 468 0464 2 NEWBUF: REF RTP_BUF;
: 469 0465 2 MAP BUFFER: REF RTP_BUF;
: 470 0466 2 RETSTATUS = .BUFFER[RTP_IOS];
: 471 0467 2 IF .RETSTATUS EQL SSS_ABORT THEN
: 472 0468 2 RETURN; ! Link gone - mailbox message will tell why
: 473 0469 2 QUIT_ON ERROR;
: 474 0470 2 CASE .BUFFER[RTP_FNC] FROM 0 TO 12 OF
: 475 0471 2 SET
: 476 0472 2 [RF_WTD]: WRITE(.BUFFER);

```

```

: 477 0473 2
: 478 0474 2
: 479 0475 2
: 480 0476 2
: 481 0477 2
: 482 0478 2
: 483 0479 2
: 484 0480 2
: 485 0481 2
: 486 0482 2
: 487 0483 2
: 488 0484 2
: 489 P 0485 2
: 490 P P 0486 2
: 491 P P 0487 2
: 492 P P 0488 2
: 493 P P 0489 2
: 494 P 0490 2
: 495 0491 2
: 496 0492 2
: 497 0493 2
: 498 0494 2
: 499 0495 1

```

```

[RF_RDD]: READ(.BUFFER);
[RF_WRD]: READPROMPT(.BUFFER);
[RF_JNS]: UNSDATENBL(.BUFFER);
[RF_IL]: CANCEL(.BUFFER);
[RF_RSC]: READSINGLE(.BUFFER);
[RF_ATT]: ATTACH(.BUFFER);
[RF_GTC]: GETTERMCHAR(.BUFFER);
[INRANGE]: UNSUPPORTED(.BUFFER);
[OUTRANGE]: UNSUPPORTED(.BUFFER);
TES:
NEWBUF = GETBUF(); ! GET ANOTHER BUFFER
RETSTATUS =
$QIO (CHAN = .LINKCHAN, ! READ LINK AGAIN
      FUNC = IOS$ READVBLK,
      IOSB = NEWBUF[RTP_IOS],
      ASTADR = LINKRECV,
      ASTPRM = .NEWBUF,
      P1 = NEWBUF[RTP_FNC],
      P2 = 128);
IF .RETSTATUS EQL $$$_ABORT THEN
  RETURN; ! Link gone - mailbox msg will tell why
QUIT_ON_ERROR;
END;

```

		000C 0000 LINKRECV:							
		53	00000000G	00	9E	00002	.WORD	Save R2,R3	: 0416
		52	04	AC	D0	00009	MOVAB	RETSTATUS, R3	: 0466
		63	08	A2	3C	0000D	MOVZWL	8(R2), RETSTATUS	: 0467
		50		63	D0	00011	MOVL	RETSTATUS, R0	
		2C		50	D1	00014	CMPL	R0, #44	
					01	12	BNEQ	1\$	
						04	RET		
		03		50	E8	0001A	1\$: BLBS	R0, 2\$	: 0468
				00A6	31	0001D	BRW	14\$	
001C	0C	00	10	A2	8F	00020	2\$: CASEB	16(R2), #0, #12	: 0470
0049	0064	0064		0064		00025	3\$: .WORD	12\$-3\$,-	
0064	0037	002E		0025		0002D		12\$-3\$,-	
	005B	0052		0040		0C035		12\$-3\$,-	
				0064		0003D		4\$-3\$,-	
								5\$-3\$,-	
								6\$-3\$,-	
								7\$-3\$,-	
								8\$-3\$,-	
								9\$-3\$,-	
								10\$-3\$,-	
								11\$-3\$,-	
								12\$-3\$,-	
								12\$-3\$	
				48	11	0003F	BRB	12\$	: 0481
				52	DD	00041	4\$: PUSHL	R2	: 0472
		0000V	CF	01	FB	00043	CALLS	#1, WRITE	
				46	11	00048	BRB	13\$	

0000V	CF	52	DD	0004A	5\$:	PUSHL	R2		0473
		01	FB	0004C		CALLS	#1	READ	
		3D	11	00051		BRB	13\$		
0000V	CF	52	DD	00053	6\$:	PUSHL	R2		0474
		01	FB	00055		CALLS	#1	READPROMPT	
		34	11	0005A		BRB	13\$		
0000V	CF	52	DD	0005C	7\$:	PUSHL	R2		0475
		01	FB	0005E		CALLS	#1	UNSDATENBL	
		2B	11	00063		BRB	13\$		
0000V	CF	52	DD	00065	8\$:	PUSHL	R2		0476
		01	FB	00067		CALLS	#1	CANCEL	
		22	11	0006C		BRB	13\$		
0000V	CF	52	DD	0006E	9\$:	PUSHL	R2		0477
		01	FB	00070		CALLS	#1	READSINGLE	
		19	11	00075		BRB	13\$		
0000V	CF	52	DD	00077	10\$:	PUSHL	R2		0478
		01	FB	00079		CALLS	#1	ATTACH	
		10	11	0007E		BRB	13\$		
0000V	CF	52	DD	00080	11\$:	PUSHL	R2		0479
		01	FB	00082		CALLS	#1	GETTERMCHAR	
		07	11	00087		BRB	13\$		
0000V	CF	52	DD	00089	12\$:	PUSHL	R2		0480
		01	FB	0008B		CALLS	#1	UNSUPPORTED	
FF38	CF	00	FB	00090	13\$:	CALLS	#0	GETBUF	0483
		7E	7C	00095		CLRQ	-(SP)		0491
		7E	7C	00097		CLRQ	-(SP)		
	7E	80	8F	9A	00099	MOVZBL	#128, -(SP)		
		10	A0	9F	0009D	PUSHAB	16(NEWBUF)		
			50	DD	000A0	PUSHL	NEWBUF		
		FF5A	CF	9F	000A2	PUSHAB	LINKRECV		
		08	A0	9F	000A6	PUSHAB	8(NEWBUF)		
			31	DD	000A9	PUSHL	#49		
	7E	00000000G	00	3C	000AB	MOVZWL	LINKCHAN, -(SP)		
			7E	D4	000B2	CLRL	-(SP)		
00000000G	00		0C	FB	000B4	CALLS	#12, SYSSQIO		
	63		50	D0	000BB	MOVL	R0, RETSTATUS		
	2C		50	D1	000BE	CMPL	R0, #44		0492
			1C	13	000C1	BEQL	15\$		
	19		50	E8	000C3	BLBS	R0, 15\$		0493
			7E	D4	000C6	CLRL	-(SP)		
00000000G	00		01	FB	000C8	CALLS	#1, SYSSSETAST		
00000000G	00		01	90	000CF	MOVB	#1, WAKEFLAG		
			7E	7C	000D6	CLRQ	-(SP)		
00000000G	00		02	FB	000D8	CALLS	#2, SYSSWAKE		
			04	000DF	15\$:	RET			0495

; Routine Size: 224 bytes, Routine Base: \$CODE\$ + 01BC

```

0496 1 ROUTINE WRITE(BUFFER): NOVALUE =
0497 1 ++
0498 1
0499 1 Functional Description:
0500 1 Perform a write QIO function to the terminal.
0501 1
0502 1 Calling Sequence:
0503 1 standard
0504 1
0505 1 Input Parameters:
0506 1 BUFFER = address of buffer from link
0507 1
0508 1 Implicit Inputs:
0509 1 CURRENTIO
0510 1
0511 1 Output Parameters:
0512 1 none
0513 1
0514 1 Implicit Outputs:
0515 1 IOQUEUE
0516 1
0517 1 Routines Called:
0518 1 BROADCAST
0519 1
0520 1 Routine Value:
0521 1 none
0522 1
0523 1 Signals:
0524 1 none
0525 1
0526 1 Side Effects:
0527 1 An I/O may be queued for later action
0528 1
0529 1 --
0530 2 BEGIN
0531 2 MAP BUFFER: REF RTP BUF;
0532 2 IF (.BUFFER[RTP_MOD] AND RM_WBT) NEQ 0 THEN
0533 2 BROADCAST(.BUFFER) ! IT IS A BROADCAST WRITE
0534 2 ELSE
0535 3 BEGIN
0536 3 IF .CURRENTIO EQL 0 THEN
0537 4 BEGIN
0538 4 RETSTATUS =
0539 4 $QIO (CHAN = .RDWRTCHAN, ! WRITE TO THE TERMINAL
0540 4 FUNC = IOS_WRITEVBLK,
0541 4 IOSB = BUFFER[RTP_IOS],
0542 4 ASTADR = QIODONE,
0543 4 ASTPRM = .BUFFER,
0544 4 P1 = BUFFER[RTP_DAT],
0545 4 P2 = .BUFFER[RTP_TCT]);
0546 4 QUIT ON ERROR;
0547 4 CURRENTIO = .BUFFER;
0548 4 END
0549 3 ELSE
0550 3 INSQUE(.BUFFER,.IOQUEUE[1]); ! QUEUE IT FOR LATER
0551 3 END;
0552 2 END;
0553 1
0554 1
0555 1
0556 1
0557 1

```

			000C 00000	WRITE:	.WORD	Save R2,R3		0496
	53	00000000G	00 9E 00002		MOVAB	RETSTATUS, R3		
	52	04	AC D0 00009		MOVL	BUFFER, R2		0532
08	11	A2	01 E1 0000D		BBC	#1, 17(R2), 1\$		
			52 DD 00012		PUSHL	R2		0533
	0000V	CF	01 FB 00014		CALLS	#1, BROADCAST		
			04 00019		RET			
		0000'	CF D5 0001A	1\$:	TSTL	CURRENTIO		0536
			4C 12 C001E		BNEQ	3\$		
			7E 7C 00020		CLRQ	-(SP)		0545
			7E 7C 00022		CLRQ	-(SP)		
	7E	18	A2 3C 00024		MOVZWL	24(R2), -(SP)		
		1A	A2 9F 00028		PUSHAB	26(R2)		
			52 DD 0002B		PUSHL	R2		
		0000V	CF 9F 0002D		PUSHAB	QIODONE		
		08	A2 9F 00031		PUSHAB	8(R2)		
			30 DD 00034		PUSHL	#48		
	7E	00000000G	00 3C 00036		MOVZWL	RDWRTCHAN, -(SP)		
			7E D4 0003D		CLRL	-(SP)		
	00000000G	00	0C FB 0003F		CALLS	#12, SYSSQIO		
		63	50 D0 00046		MOVL	R0, RETSTATUS		
		1A	63 E8 00049		BLBS	RETSTATUS, 2\$		
			7E D4 0004C		CLRL	-(SP)		
	00000000G	00	01 FB 0004E		CALLS	#1, SYSSSETAST		
	00000000G	00	01 90 00055		MOVB	#1, WAKEFLAG		
			7E 7C 0005C		CLRQ	-(SP)		
	00000000G	00	02 FB 0005E		CALLS	#2, SYSSWAKE		
			04 00065		RET			
	0000'	CF	52 D0 00066	2\$:	MOVL	R2, CURRENTIO		0547
			04 0006B		RET			0536
	0000'	DF	62 0E 0006C	3\$:	INSQUE	(R2), @IOQUEUE+4		0550
			04 00071		RET			0552

: Routine Size: 114 bytes, Routine Base: \$CODE\$ + 029C

```

559 0553 1 ROUTINE READ(BUFFER): NOVALUE =
560 0554 1 ++
561 0555 1
562 0556 1 Functional Description:
563 0557 1 Perform a read QIO function to the terminal.
564 0558 1
565 0559 1 Calling Sequence:
566 0560 1 standard
567 0561 1
568 0562 1 Input Parameters:
569 0563 1 BUFFER = address of the link buffer
570 0564 1
571 0565 1 Implicit Inputs:
572 0566 1 CURRENTIO
573 0567 1 INDDATA
574 0568 1
575 0569 1 Output Parameters:
576 0570 1 none
577 0571 1
578 0572 1 Implicit Outputs:
579 0573 1 IOQUEUE
580 0574 1 CURRENTIO
581 0575 1 READINPROG
582 0576 1 UNSOLPEND
583 0577 1 Routines Called:
584 0578 1 INDREAD
585 0579 1 QIODONE
586 0580 1
587 0581 1 Routine Value:
588 0582 1 none
589 0583 1
590 0584 1 Signals:
591 0585 1 none
592 0586 1
593 0587 1 Side Effects:
594 0588 1 An I/O may be queued for later action.
595 0589 1
596 0590 1 --
597 0591 2 BEGIN
598 0592 2 MAP BUFFER: REF RTP_BUF;
599 0593 2 LOCAL
600 0594 2 FUNCTION;
601 0595 2 IF .INDDATA NEQ 0 THEN
602 0596 3 BEGIN ! WE ALREADY HAVE INDIRECT COMMAND DATA
603 0597 3 BUFFER[RTP_IOS] = .INDDATA[RTP_IOS]; ! COPY THE IOSB
604 0598 3 BUFFER[RTP_IOC] = .INDDATA[RTP_IOC];
605 0599 3 CH$MOVE(.INDDATA[RTP_IOC]+1,INDDATA[RTP_DAT], ! COPY THE DATA
606 0600 3 BUFFER[RTP_DAT]);
607 0601 3 INDREAD(); ! LOOK FOR MORE DATA
608 0602 3 QIODONE(.BUFFER); ! PASS THIS DATA ON
609 0603 3 RETURN;
610 0604 2 END;
611 0605 2 IF .CURRENTIO EQL 0 THEN
612 0606 3 BEGIN
613 0607 3 IF (.BUFFER[RTP_MOD] AND RM RBN) NEQ 0 THEN
614 0608 3 FUNCTION = IOS_TTYREADALL ! BINARY
615 0609 3 ELSE

```

```

: 616
: 617
: 618
: 619
: 620
: 621
: 622
: 623
: 624
: 625
: 626
: 627
: 628
: 629
: 630
: 631
: 632
: 633

```

```

P 0610
P 0611
P 0612
P 0613
P 0614
P 0615
P 0616
P 0617
P 0618
P 0619
0620
0621
0622
0623
0624
0625
0626
0627
1

```

```

FUNCTION = IOS_READVBLK; ! NORMAL
RETSTATUS =
$QIO (CHAN = .RDWRICHAN, ! READ FROM THE TERMINAL
      FUNC = .FUNCTION+MAPMODIFIER(.BUFFER[RTP_MOD]),
      IOSB = BUFFER[RTP_IOS],
      ASTADR = QIODONE,
      ASTPRM = .BUFFER,
      P1 = BUFFER[RTP_DAT],
      P2 = .BUFFER[RTP_RCT],
      P4 = TERMINATOR(.BUFFER[RTP_MOD]));
QUIT ON ERROR;
CURRENTIO = .BUFFER;
UNSOLPEND = 0; ! NO MORE DATA PENDING
READINPROG = 1;
END
ELSE
INSQUE(.BUFFER,.IOQUEUE[1]); ! QUEUE IT FOR LATER
END;

```

58	00000000G	00	01FC	00000	READ:	.WORD	Save R2,R3,R4,R5,R6,R7,R8	: 0553
57	0000	CF	9E	00002		MOVAB	RETSTATUS, R8	
50	04	A7	D0	0000E		MOVAB	CURRENTIO, R7	
		22	13	00012		MOVL	INDDATA, R0	: 0595
56	04	AC	D0	00014		BEQL	1\$	
08	A6	08	A0	D0 00018		MOVL	BUFFER, R6	: 0597
		51	0A	A0 3C 0001D		MOVL	8(R0), 8(R6)	
			51	D6 00021		MOVZWL	10(R0), R1	: 0599
1A	A6		51	28 00023		INCL	R1	
	0000V	CF	00	FB 00029		MOVZWL	R1, 26(R0), 26(R6)	: 0600
			56	DD 0002E		CALLS	#0, INDREAD	: 0601
	0000V	CF	01	FB 00030		PUSHL	R6	: 0602
				04 00035		CALLS	#1, QIODONE	
			52	04 AC D0 00036	1\$:	RET		: 0596
			67	D5 0003A		MOVL	BUFFER, R2	: 0607
			72	12 0003C		TSTL	CURRENTIO	: 0605
	05	11	A2	02 E1 0003E		BNEQ	5\$	
			53	3A D0 00043		BBC	#2, 17(R2), 2\$	: 0607
				03 11 00046		MOVL	#58, FUNCTION	: 0608
			53	31 D0 00048	2\$:	BRB	3\$	
				7E 7C 0004B	3\$:	MOVL	#49, FUNCTION	: 0610
			7E	11 A2 9A 0004D		CLRQ	-(SP)	: 0619
	0000V	CF		01 FB 00051		MOVZBL	17(R2), -(SP)	
				50 DD 00056		CALLS	#1, TERMINATOR	
				7E D4 00058		PUSHL	R0	
			7E	16 A2 3C 0005A		CLRL	-(SP)	
				1A A2 9F 0005E		MOVZWL	22(R2), -(SP)	
				52 DD 00061		PUSHAB	26(R2)	
				0000V CF 9F 00063		PUSHL	R2	
				08 A2 9F 00067		PUSHAB	QIODONE	
				7E 11 A2 9A 0006A		PUSHAB	8(R2)	
	0000V	CF		01 FB 0006E		MOVZBL	17(R2), -(SP)	
				6043 9F 00073		CALLS	#1, MAPMODIFIER	
						PUSHAB	(R0)[FUNCTION]	

RSXRT  
V04-000

N 3  
16-Sep-1984 02:18:51  
14-Sep-1984 13:04:57

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[RTPAD.SRC]RSXRT.B32;1

Page 27  
(7)

7E 0000000G	00	3C 00076	MOVZWL	RDWRTCHAN, -(SP)	:
		7E D4 0007D	CLRL	-(SP)	:
C0000000G	00	0C FB 0007F	CALLS	#12, SYSSQIO	:
	68	50 D0 00086	MOVL	R0, RETSTATUS	:
	1A	68 E8 00089	BLBS	RETSTATUS, 4\$	:
		7E D4 0008C	CLRL	-(SP)	:
00000000G	00	01 FB 0008E	CALLS	#1, SYSSSETAST	:
00000000G	00	01 90 00095	MOVB	#1, WAKEFLAG	:
		7E 7C 0009C	CLRQ	-(SP)	:
00000000G	00	02 FB 0009E	CALLS	#2, SYSSWAKE	:
		04 000A5	RET		:
	67	52 D0 000A6 4\$:	MOVL	R2, CURRENTIO	0621
FA	A7	0100 8F B0 000A9	MOVW	#256, UNSOLPEND	0622
		04 000AF	RET		0605
OC	B7	62 0E 000B0 5\$:	INSQUE	(R2), @IOQUEUE+4	0626
		04 000B4	RET		0627

; Routine Size: 181 bytes, Routine Base: \$CODE\$ + 030E



```

: 635 0628 1 ROUTINE READPROMPT(BUFFER): NOVALUE =
: 636 0629 1 ++
: 637 0630 1
: 638 0631 1 Functional Description:
: 639 0632 1 Perform a readprompt QIO function to the terminal.
: 640 0633 1
: 641 0634 1 Calling Sequence:
: 642 0635 1 standard
: 643 0636 1
: 644 0637 1 Input Parameters:
: 645 0638 1 BUFFER = address of the link buffer
: 646 0639 1
: 647 0640 1 Implicit Inputs:
: 648 0641 1 CURRENTIO
: 649 0642 1 INDDATA
: 650 0643 1
: 651 0644 1 Output Parameters:
: 652 0645 1 none
: 653 0646 1
: 654 0647 1 Implicit Outputs:
: 655 0648 1 CURRENTIO
: 656 0649 1 UNSOLPEND
: 657 0650 1 READINPROG
: 658 0651 1 IOQUEUE
: 659 0652 1
: 660 0653 1 Routines Called:
: 661 0654 1 READ
: 662 0655 1
: 663 0656 1 Routine Value:
: 664 0657 1 none
: 665 0658 1
: 666 0659 1 Signals:
: 667 0660 1 none
: 668 0661 1
: 669 0662 1 Side Effects:
: 670 0663 1 An I/O may be queued for later action.
: 671 0664 1
: 672 0665 1 --
: 673 0666 2 BEGIN
: 674 0667 2 MAP BUFFER: REF RTP_BUF;
: 675 0668 2 LOCAL
: 676 0669 2 FUNCTION;
: 677 0670 2 IF .INDDATA NEQ 0 THEN
: 678 0671 2 BEGIN ! WE HAVE INDIRECT COMMAND FILE DATA
: 679 0672 3 READ (.BUFFER); ! GET THE DATA
: 680 0673 3 RETURN;
: 681 0674 2 END;
: 682 0675 2 IF .CURRENTIO EQL 0 THEN
: 683 0676 3 BEGIN
: 684 0677 3 IF (.BUFFER[RTP MOD] AND RM RBN) NEQ 0 THEN
: 685 0678 3 FUNCTION = IOS_TTYREADPALL ! BINARY
: 686 0679 3 ELSE
: 687 0680 3 FUNCTION = IOS_READPROMPT; ! NORMAL
: 688 0681 3 RETSTATUS =
: 689 P 0682 3 $QIO (CHAN = .RDWRCHAN, ! READPROMPT TO THE TERMINAL
: 690 P 0683 3 FUNC = .FUNCTION+MAPMODIFIER(.BUFFER[RTP_MOD]),
: 691 P 0684 3 IOSB = BUFFER[RTP_IOS],

```

```

: 692 P 0685
: 693 P 0686
: 694 P 0687
: 695 P 0688
: 696 P 0689
: 697 P 0690
: 698 0691
: 699 0692
: 700 0693
: 701 0694
: 702 0695
: 703 0696
: 704 0697
: 705 0698
: 706 0699

```

```

ASTADR = QIODONE,
ASTPRM = .BUFFER,
P1 = BUFFER[RTP_DAT],
P2 = .BUFFER[RTP_RCT],
P4 = TERMINATOR(.BUFFER[RTP_MOD]),
P5 = BUFFER[RTP_DAT],
P6 = .BUFFER[RTP_TCT]);
QUIT ON ERROR;
CURRENTIO = .BUFFER;
UNSOLPEND = 0; ! NO MORE DATA PENDING
READINPROG = 1;
END
ELSE
INSQUE(.BUFFER,.IOQUEUE[1]); ! QUEUE IT FOR LATER
END;

```

```

                                003C 00000 READPROMPT:
                                .WORD Save R2,R3,R4,R5
55 00000000G 00 9E 00002 MOVAB RETSTATUS, R5 : 0628
54 0000 04 CF 9E 00009 MOVAB CURRENTIO, R4
                                TSTL INDDATA : 0670
                                04 09 13 00011 BEQL 1$
FF30 CF 04 AC DD 00013 PUSHL BUFFER : 0672
                                01 FB 00016 CALLS #1, READ
                                04 0001B RET
52 04 AC D0 0001C 1$: MOVL BUFFER, R2 : 0671
                                64 D5 00020 TSTL CURRENTIO : 0677
                                77 12 00022 BNEQ 5$ : 0675
05 11 A2 02 E1 00024 BBC #2, 17(R2), 2$ : 0677
53 3B D0 00029 MOVL #59, FUNCTION : 0678
                                03 11 0002C BRB 3$
53 37 D0 0002E 2$: MOVL #55, FUNCTION : 0680
7E 18 A2 3C 00031 3$: MOVZWL 24(R2), -(SP) : 0691
                                1A A2 9F 00035 PUSHAB 26(R2)
                                11 A2 9A 00038 MOVZBL 17(R2), -(SP)
0000V CF 01 FB 0003C CALLS #1, TERMINATOR
                                50 DD 00041 PUSHL R0
                                7E D4 00043 CLRL -(SP)
7E 16 A2 3C 00045 MOVZWL 22(R2), -(SP)
                                1A A2 9F 00049 PUSHAB 26(R2)
                                52 DD 0004C PUSHL R2
                                0000V CF 9F 0004E PUSHAB QIODONE
                                08 A2 9F 00052 PUSHAB 8(R2)
0000V 7E 11 A2 9A 00055 MOVZBL 17(R2), -(SP)
CF 01 FB 00059 CALLS #1, MAPMODIFIER
                                6043 9F 0005E PUSHAB (R0)[FUNCTION]
7E 00000000G 00 3C 00061 MOVZWL RDWRTCHAN, -(SP)
                                7E D4 00068 CLRL -(SP)
00000000G 00 0C FB 0006A CALLS #12, SYSSQIO
65 50 D0 00071 MOVL R0, RETSTATUS
1A 65 E8 00074 BLBS RETSTATUS, 4$
                                7E D4 00077 CLRL -(SP)
00000000G 00 01 FB 00079 CALLS #1, SYSSSETAST

```

RSXRT  
V04-000

D 4  
16-Sep-1984 02:18:51  
14-Sep-1984 13:04:57

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[RTPAD.SRC]RSXRT.B32;1

Page 25  
(8)

00000000G	00		01	90	00080	MOVB	#1, WAKEFLAG	:	
			7E	7C	00087	CLRQ	-(SP)	:	
00000000G	00		02	FB	00089	CALLS	#2, SYSSWAKE	:	
				04	00090	RET		:	
	64		52	D0	00091	4\$:	MOVL	R2, CURRENTIO	: 0693
FA	A4	0100	8F	B0	00094		MOVW	#256, UNSOLPEND	: 0694
				04	0009A		RET		: 0675
	OC	B4	62	0E	0009B	5\$:	INSQUE	(R2), @IOQUEUE+4	: 0698
				04	0009F		RET		: 0699

; Routine Size: 160 bytes, Routine Base: \$CODES + 03C3

RS  
VO

```

708 0700 1 ROUTINE QIODONE(BUFFER): NOVALUE =
709 0701 1 ++
710 0702 1
711 0703 1 Functional Description:
712 0704 1     Send a message on the link when a terminal QIO completes.
713 0705 1     Interpret the "EXIT RMT" command to exit this program.
714 0706 1
715 0707 1 Calling Sequence:
716 0708 1     standard
717 0709 1
718 0710 1 Input Parameters:
719 0711 1     BUFFER = address of the link buffer.
720 0712 1
721 0713 1 Implicit Inputs:
722 0714 1     none
723 0715 1
724 0716 1 Output Parameters:
725 0717 1     none
726 0718 1
727 0719 1 Implicit Outputs:
728 0720 1     READINPROG
729 0721 1     CURRENTIO
730 0722 1     RETSTATUS
731 0723 1
732 0724 1 Routines Called:
733 0725 1     NEXTIO
734 0726 1     FREEBUF
735 0727 1
736 0728 1 Routine Value:
737 0729 1     none
738 0730 1
739 0731 1 Signals:
740 0732 1     none
741 0733 1
742 0734 1 Side Effects:
743 0735 1     If there is an error on the write to the link, a $WAKE will be issued
744 0736 1     to cause this program to abort.
745 0737 1
746 0738 1 --
747 0739 2 BEGIN
748 0740 2 MAP BUFFER: REF RTP_BUF;
749 0741 2 LOCAL
750 0742 2     COUNT: ;
751 0743 2 IF .BUFFER[RTP_IOS] AND 1 THEN
752 0744 2     BUFFER[RTP_STS] = RS_SFC           ! GOOD STATUS
753 0745 2 ELSE
754 0746 2     BUFFER[RTP_STS] = RS_FPE;         ! ERROR
755 0747 2 BUFFER[RTP_FLG] = 0;
756 0748 2 BUFFER[RTP_TCT] = 0;
757 0749 2 COUNT = 10;                          ! MINIMUM MESSAGE LENGTH
758 0750 2 IF .BUFFER[RTP_FNC] NEQ RF_WTD THEN
759 0751 3     BEGIN
760 0752 3         ! IT WAS A READ
761 0753 3         COUNT = .COUNT + .BUFFER[RTP_IOC];    ! ADD THE DATA
762 0754 3         IF (.BUFFER[RTP_MOD] AND (RM_RTC+RM_RNE)) EQL 0 THEN
763 0755 4             BEGIN
764 P 0756 4                 ! CHECK FOR A CARRIAGE RETURN
764                 IF (.BUFFER+.COUNT+16)<0,8> EQL 13 THEN
764                     $QIOW (CHAN = .RDWRTCHAN,    ! ECHO CAR-RET

```

```

: 765
: 766
: 767
: 768
: 769
: 770
: 771
: 772
: 773
: 774
: 775
: 776
: 777
: 778
: 779
: 780
: 781
: 782
: 783
: 784
: 785
: 786
: 787
: 788
: 789
: 790
: 791
: 792
: 793
: 794
: 795
: 796
: 797
: 798
: 799
: 800
: 801
: 802
: 803
: 804
: 805
: 806
: 807
: 808
: 809

```

```

P 0757 4
P 0758 4
  0759 4
  0760 3
  0761 3
  0762 4
  0763 4
  0764 4
  0765 4
  0766 4
  0767 3
  0768 3
  0769 3
  0770 3
  0771 3
  0772 4
  0773 4
  0774 4
  0775 3
  0776 3
  0777 3
  0778 3
  0779 3
  0780 4
  0781 4
  0782 4
  0783 3
  0784 3
  0785 3
  0786 2
  0787 2
  0788 2
  0789 2
  0790 2
  0791 2
  0792 2
  0793 2
  0794 2
  0795 2
  0796 2
  0797 2
  0798 2
  0799 2
  0800 2
  0801 1

```

```

FUNC = IOS$ WRITEVBLK,
P1 = UPLIT(13),
P2 = 1);
END;
IF .BUFFER[RTP_IOC] EQL 8 THEN ! COULD BE AN EXIT
BEGIN
IF CH$EQL(8,BUFFER[RTP_DAT],8,UPLIT('EXIT RMT')) THEN
QUIT; ! GET OUT
IF CH$EQL(8,BUFFER[RTP_DAT],8,UPLIT('exit rmt')) THEN
QUIT; ! GET OUT
END;
IF .BUFFER[RTP_RCT] NEQ .BUFFER[RTP_IOC] THEN
COUNT = .COUNT + 1 ! ADD TERMINATOR
ELSE
IF .COUNT NEQ 128 THEN
BEGIN ! THIS IS A KLUGE FOR RSX
COUNT = .COUNT+1;
(.BUFFER+15+.COUNT)<0,8> = 0; ! ADD A NULL
END;
BUFFER[RTP_RCT] = .BUFFER[RTP_IOC]; ! COUNT
READINPROG = 0; ! DONE
END;
IF ((.BUFFER[RTP_MOD] AND RM_WBT) EQL 0) AND
(.BUFFER[RTP_FNC] NEQ RF_RSC) THEN
CURRENTIO = 0; ! CURRENT I/O HAS COMPLETED
IF ((.BUFFER[RTP_FNC] EQL RF_WTD) AND
((.BUFFER[RTP_MOD] AND RM_NWC) NEQ 0))
OR (.BUFFER[RTP_IOS] EQL SSS_ABORT) THEN
FREEBUF(.BUFFER)
ELSE
BEGIN
RETSTATUS =
$QIO (CHAN = .LINKCHAN, ! WRITE TO ' NK
FUNC = IOS$ WRITEVBLK,
IOSB = BUFFER[RTP_IOS],
ASTADR = LINKWRTDONE,
ASTPRM = .BUFFER,
P1 = BUFFER[RTP_FNC],
P2 = .COUNT);
IF .RETSTATUS EQL SSS_ABORT THEN
RETURN; ! Link gone - mailbox msg will tell why
QUIT_ON_ERROR;
END;
NEXTIO(); ! CHECK FOR A PENDING I/O
END;

```

```

.PSECT $SPLITS,NOWRT,NOEXE,2
54 4D 52 20 54 49 58 45 0000000D 00030 P.AAD: .LONG 13
74 6D 72 20 74 69 78 65 00034 P.AAE: .ASCII \EXIT RMT\
0003C P.AAF: .ASCII \exit rmt\
.PSECT $CODE$,NOWRT,2

```

Address	Label	Op Code	Op	Op Data	Comment	Address
54	04	AC	DO	00002	MOVL	0700
56	10	A4	9E	00006	MOVAB	0743
05	08	A4	E9	0000A	BLBC	0744
	03	A6	94	0000E	CLRB	0743
		04	11	00011	BRB	0744
	03	A6	01	90 00013	1\$: MOVB	
		02	A6	94 00017	2\$: CLRB	
	18	A4	B4	0001A	CLRW	0746
55		0A	DO	0001D	MOVL	0747
03		66	91	00020	CMPB	0748
		6E	13	00023	BEQL	0749
50	CA	A4	3C	00025	MOVZWL	0750
55		50	CO	00029	ADDL2	0752
18	01	A6	93	0002C	BITB	0753
		26	12	00030	BNEQ	
0D	10	A544	91	00032	CMPB	0755
		1F	12	00037	BNEQ	
		7E	7C	00039	CLRQ	0759
		7E	7C	0003B	CLRQ	
		01	DD	0003D	PUSHL	
		0000'	CF	9F 0003F	PUSHAB	
		7E	7C	00043	CLRQ	
7E		30	7D	00045	MOVQ	
7E	00000000G	00	3C	00048	MOVZWL	
		7E	D4	0004F	CLRL	
	00000000G	00	0C	FB 00051	CALLS	
		08	0A	A4 B1 00058	3\$: CMPW	0761
		12	12	0005C	BNEQ	
0000'	CF	1A	A4	08 29 0005E	CMPC3	0763
		07	13	00065	BEQL	
0000'	CF	1A	A4	08 29 00067	CMPC3	0765
		0A	A4	7A 13 0006E	4\$: BEQL	
		0A	A4	16 A4 B1 00070	5\$: CMPW	0768
		04	13	00075	BEQL	
		55	D6	00077	INCL	0769
		0F	11	00079	BRB	
	00000080	8F	55	D1 0007B	6\$: CMPL	0771
		06	13	00082	BEQL	
		55	D6	00084	INCL	0773
		0F	A544	94 00086	CLRB	0774
	16	A4	0A	A4 B0 0008A	7\$: MOVW	0776
		0000'	CF	94 0008F	CLRB	0777
09		66	09	E0 00093	8\$: BBS	0779
		07	66	91 00097	CMPB	0780
		04	13	0009A	BEQL	
		0000'	CF	D4 0009C	CLRL	0781
		03	66	91 000A0	9\$: CMPB	0782
		04	12	000A3	BNEQ	
		66	B5	000A5	TSTW	0783
		06	19	000A7	BLSS	
	2C	08	A4	B1 000A9	10\$: CMPW	0784
		09	12	000AD	BNEQ	
		54	DD	000AF	11\$: PUSHL	0785
FC94	CF	01	FB	000B1	CALLS	
		4C	11	000B6	BRB	

Save R2,R3,R4,R5,R6

MOVL BUFFER, R4  
 16(R4), R6  
 8(R4), 1\$  
 3(R6)  
 2\$  
 #1, 3(R6)  
 2(R6)  
 24(R4)  
 #10, COUNT  
 (R6), #3  
 8\$  
 10(R4), R0  
 R0, COUNT  
 1(R6), #24  
 3\$  
 16(COUNT)[R4], #13  
 3\$  
 -(SP)  
 -(SP)  
 #1  
 P.AAD  
 -(SP)  
 #48, -(SP)  
 RDWRTCHAN, -(SP)  
 -(SP)  
 #12, SYSSQIOW  
 10(R4), #8  
 5\$  
 #8, 26(R4), P.AAE  
 4\$  
 #8, 26(R4), P.AAF  
 13\$  
 22(R4), 10(R4)  
 6\$  
 COUNT  
 7\$  
 COUNT, #128  
 7\$  
 COUNT  
 15(COUNT)[R4]  
 10(R4), 22(R4)  
 READINPROG  
 #9, (R6), 9\$  
 (R6), #7  
 9\$  
 CURRENTIO  
 (R6), #3  
 10\$  
 (R6)  
 11\$  
 8(R4), #44  
 12\$  
 R4  
 #1, FREEBUF  
 14\$

		7E 7C 000B8 12\$:	CLRQ	-(SP)	:	0795
		7E 7C 000BA	CLRQ	-(SP)	:	
		S5 DD 000BC	PUSHL	COUNT	:	
	0050	8F BB 000BE	PUSHR	#^M<R4,R6>	:	
	0000V	CF 9F 000C2	PUSHAB	LINKWRIDONE	:	
	08	A4 9F 000C6	PUSHAB	8(R4)	:	
		30 DD 000C9	PUSHL	#48	:	
	7E 00000000G	00 3C 000CB	MOVZWL	LINKCHAN, -(SP)	:	
		7E D4 000D2	CLRL	-(SP)	:	
00000000G	00	0C FB 000D4	CALLS	#12, SYSSQIO	:	
00000000G	00	50 D0 000DB	MOVL	R0, RETSTATUS	:	
	2C	50 D1 000E2	CMLP	R0, #44	:	0796
		22 13 000E5	BEQL	15\$	:	
	1A	50 E8 000E7	BLBS	R0, 14\$	:	0797
		7E D4 000EA 13\$:	CLRL	-(SP)	:	
00000000G	00	01 FB 000EC	CALLS	#1, SYSSSETAST	:	
00000000G	00	01 90 000F3	MOVB	#1, WAKEFLAG	:	
		7E 7C 000FA	CLRQ	-(SP)	:	
00000000G	00	02 FB 000FC	CALLS	#2, SYSSWAKE	:	
		04 00103	RET		:	
0000V	CF	00 FB 00104 14\$:	CALLS	#0, NEXTIO	:	0800
		04 00109 15\$:	RET		:	0801

: Routine Size: 266 bytes, Routine Base: \$CGDES + 0463

: 810 0802 1

.....

```

0812 0803 1 ROUTINE LINKWRTDONE(BUFFER): NOVALUE =
0813 0804 1 ++
0814 0805 1
0815 0806 1 Functional Description:
0816 0807 1 Free the link buffer when a write to the link completes
0817 0808 1
0818 0809 1 Calling Sequence:
0819 0810 1 standard
0820 0811 1
0821 0812 1 Input Parameters:
0822 0813 1 BUFFER = address of the link buffer.
0823 0814 1
0824 0815 1 Implicit Inputs:
0825 0816 1 none
0826 0817 1
0827 0818 1 Output Parameters:
0828 0819 1 RETSTATUS
0829 0820 1
0830 0821 1 Implicit Outputs:
0831 0822 1 none
0832 0823 1
0833 0824 1 Routines Called:
0834 0825 1 FREEBUF
0835 0826 1
0836 0827 1 Routine Value:
0837 0828 1 none
0838 0829 1
0839 0830 1 Signals.
0840 0831 1 none
0841 0832 1
0842 0833 1 Side Effects:
0843 0834 1 If there was an error on the write to the link, a $WAKE is issued to
0844 0835 1 cause the program to abort.
0845 0836 1
0846 0837 1 --
0847 0838 2 BEGIN
0848 0839 2 MAP BUFFER: REF RTP BUF;
0849 0840 2 RETSTATUS = .BUFFER[RTP IOS];
0850 0841 2 IF .RETSTATUS EQL $$$_ABORT THEN
0851 0842 2 RETURN; ! Link gone - mailbox msg will tell why
0852 0843 2 QUIT ON ERROR;
0853 0844 2 FREEBUF(.BUFFER); ! WE NO LONGER NEED THE BUFFER
0854 0845 1 END;

```

```

000C 0000 LINKWRTDONE:
53 00000000G 00 9E 00002 .WORD Save R2,R3 : 0803
52 04 AC D0 00009 MOVAB RETSTATUS, R3 : 0840
63 08 A2 3C 0000D MOVZWL 8(R2), RETSTATUS : 0841
50 63 D0 00011 MOVL RETSTATUS, R0 : 0841
2C 50 D1 00014 CML R0, #44 :
1A 24 13 00017 BEQL 2$ :
50 E8 00019 BLBS R0, 1$ : 0842

```



RSXRT  
V04-000

J 4  
16-Sep-1984 02:18:51  
14-Sep-1984 13:04:57

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[RTPAD.SRC]RSXRT.B32;1

Page 31  
(10)

RS  
V0

00000000G	00	7E	D4	0001C	CLRL	-(SP)	:
00000000G	00	01	FB	0001E	CALLS	#1, SYSS\$SETAST	:
		01	90	00025	MOVB	#1, WAKEFLAG	:
00000000G	00	7E	7C	0002C	CLRQ	-(SP)	:
		02	FB	0002E	CALLS	#2, SYSS\$WAKE	:
			04	00035	RET		:
		52	DD	00036	PUSHL	R2	: 0844
FC03	CF	01	FB	00038	CALLS	#1, FREEBUF	:
		04	0003D	2\$:	RET		: 0845

; Routine Size: 62 bytes, Routine Base: \$CODE\$ + 056D

```

: 856 0846 1 ROUTINE UNSDATENBL(BUFFER): NOVALUE =
: 857 0847 1 ++
: 858 0848 1
: 859 0849 1 Functional Description:
: 860 0850 1 Enable or disable unsolicited data to the RSX system.
: 861 0851 1
: 862 0852 1 Calling Sequence:
: 863 0853 1 standard
: 864 0854 1
: 865 0855 1 Input Parameters:
: 866 0856 1 BUFFER = address of the link buffer
: 867 0857 1
: 868 0858 1 Implicit Inputs:
: 869 0859 1 UNSOLPEND
: 870 0860 1 INDDATA
: 871 0861 1
: 872 0862 1 Output Parameters:
: 873 0863 1 none
: 874 0864 1
: 875 0865 1 Implicit Outputs:
: 876 0866 1 UNSOLENBLFLG
: 877 0867 1
: 878 0868 1 Routines Called:
: 879 0869 1 TERMMBXMSG
: 880 0870 1 FREEBUF
: 881 0871 1 READ
: 882 0872 1
: 883 0873 1 Routine Value:
: 884 0874 1 none
: 885 0875 1
: 886 0876 1 Signals:
: 887 0877 1 none
: 888 0878 1
: 889 0879 1 Side Effects:
: 890 0880 1 If unsolicited input is enabled, any pending data is read.
: 891 0881 1
: 892 0882 1 --
: 893 0883 2 BEGIN
: 894 0884 2 MAP BUFFER: REF RTP_BUF;
: 895 0885 2 LOCAL
: 896 0886 2 NEWBUF: REF VECTOR;
: 897 0887 2 IF .BUFFER[RTP_FLG] NEQ RM_TUI THEN
: 898 0888 3 BEGIN
: 899 0889 3 IF .INDDATA NEQ 0 THEN
: 900 0890 4 BEGIN ! THERE IS INDIRECT FILE DATA
: 901 0891 4 NEWBUF = GETBUF(); ! GET A SUBSTITUTE BUFFER
: 902 0892 4 CH$MOVE(40,.BUFFER,.NEWBUF); ! COPY HEADER + SOME
: 903 0893 4 READ (.NEWBUF); ! GET IT
: 904 0894 3 END;
: 905 0895 3 UNSOLENBLFLG = .BUFFER; ! ENABLE
: 906 0896 3 IF .UNSOLPEND NEQ 0 THEN
: 907 0897 3 TERMMBXMSG(); ! DATA ALREADY PENDING
: 908 0898 3 END
: 909 0899 2 ELSE
: 910 0900 2 BEGIN ! DISABLE
: 911 0901 2 FREEBUF(.BUFFER); ! NO LONGER NEED BUFFER
: 912 0902 3 IF .UNSOLENBLFLG NEQ 0 THEN

```

RSXRT  
V04-000

L 4  
16-Sep-1984 02:18:51  
14-Sep-1984 13:04:57

VAX-11 Bliss-32 V4.0-742 Page 33  
DISK\$VMSMASTER:[RTPAD.SRC]RSXRT.B32;1 (11)

: 913 0903 4  
: 914 0904 4  
: 915 0905 4  
: 916 0906 3  
: 917 0907 2  
: 918 0908 1

BEGIN  
FREEBUF(.UNSOLENBFLG); ' UNSOL DATA BUFFER  
UNSOL+NBFLG = 0;  
END;

END; END;

01FC 0000 UNSDATENBL:

	58	0000'	CF	9E	00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8	: 0846
	56	04	AC	D0	00007	MOVAB	UNSOLENBFLG, R8	: 0887
80	8F	12	A6	91	0000B	MOVL	BUFFER, R6	: 0889
			26	13	00010	CMPB	18(R6), #128	: 0891
		10	A8	D5	00012	BEQL	2\$	: 0892
			13	13	00015	TSTL	INDDATA	: 0893
FBC2	CF		00	FB	00017	BEQL	1\$	: 0895
	57		50	D0	0001C	CALLS	#0, GETBUF	: 0896
67	66		50	D0	0001C	MOVL	R0, NEWBUF	: 0897
			28	28	0001F	MOVC3	#40, (R6), (NEWBUF)	: 0901
FBC3	CF		57	DD	00023	PUSHL	NEWBUF	: 0902
	68		01	FB	00025	CALLS	#1, READ	: 0904
FD39	CF		56	D0	0002A	MOVL	R6, UNSOLENBFLG	: 0905
		06	A8	95	0002D	TSTB	UNSOLPEND	: 0908
			1B	13	00030	BEQL	3\$	: 0909
0000V	CF		00	FB	00032	CALLS	'?', TERMMBXMSG	: 0910
				04	00037	RET		: 0911
			56	DD	00038	PUSHL	Rc	: 0912
FBC3	CF		01	FB	0003A	CALLS	#1, FREEBUF	: 0913
	50		68	D0	0003F	MOVL	UNSOLENBFLG, R0	: 0914
			09	13	00042	BEQL	3\$	: 0915
			50	DD	00044	PUSHL	R0	: 0916
FBB7	CF		01	FB	00046	CALLS	#1, FREEBUF	: 0917
			68	D4	0004B	CLRL	UNSOLENBFLG	: 0918
			04	0004D	3\$:	RET		: 0919

; Routine Size: 78 bytes, Routine Base: \$CODE\$ + 05AB

```

: 920 0909 1 ROUTINE TERMMBXMSG: NOVALUE =
: 921 0910 1 ++
: 922 0911 1
: 923 0912 1 Functional Description:
: 924 0913 1 Handle messages from the terminal mailbox indicating unsolicited data
: 925 0914 1 or hangup.
: 926 0915 1
: 927 0916 1 Calling Sequence:
: 928 0917 1 standard
: 929 0918 1
: 930 0919 1 Input Parameters:
: 931 0920 1 none
: 932 0921 1
: 933 0922 1 Implicit Inputs:
: 934 0923 1 READINPROG
: 935 0924 1 UNSOLENBLFLG
: 936 0925 1 ATTACHFLAG
: 937 0926 1 SINGLEFLAG
: 938 0927 1
: 939 0928 1 Output Parameters:
: 940 0929 1 none
: 941 0930 1
: 942 0931 1 Implicit Outputs:
: 943 0932 1 UNSOLENBLFLG
: 944 0933 1 SINGLEINPROG
: 945 0934 1 UNSOLPEND
: 946 0935 1
: 947 0936 1 Routines Called:
: 948 0937 1 GETBUF
: 949 0938 1
: 950 0939 1 Routine Value:
: 951 0940 1 none
: 952 0941 1
: 953 0942 1 Signals:
: 954 0943 1 none
: 955 0944 1
: 956 0945 1 Side Effects:
: 957 0946 1 In the case of unsolicited input, a read to the terminal is initiated
: 958 0947 1 if either unsolicited input or single character mode is enabled. A
: 959 0948 1 new read to the terminal mailbox is also initiated.
: 960 0949 1 In the case of a hangup, a $WAKE is issued to cause the program to
: 961 0950 1 abort.
: 962 0951 1
: 963 0952 1 --
: 964 0953 2 BEGIN
: 965 0954 2 MAP UNSOLENBLFLG: REF VECTOR;
: 966 0955 2 LOCAL
: 967 0956 2 NEWBUF: REF VECTOR;
: 968 0957 2 IF .TERMMBXDATA[0] EQL MSG$_TRMUNSOLIC THEN
: 969 0958 3 BEGIN
: 970 0959 3 IF .READINPROG EQL 0 THEN
: 971 0960 4 BEGIN
: 972 0961 4 IF (.UNSOLENBLFLG NEQ 0) AND
: 973 0962 4 ((.ATTACHFLAG OR .SINGLEFLAG) EQL 0) THEN
: 974 0963 5 BEGIN
: 975 0964 5 READ(.UNSOLENBLFLG); ! READ IT
: 976 0965 5 NEWBUF = GETBUF(); ! GET ANOTHER BUFFER

```

```

: 977
: 978
: 979
: 980
: 981
: 982
: 983
: 984
: 985
: 986
: 987
: 988
: 989
: 990
: 991
: 992
: 993
: 994
: 995
: 996
: 997
: 998
: 999
1000
: 1001
: 1002
: 1003
: 1004
: 1005
: 1006
: 1007
: 1008
: 1009

```

```

0966 5 NEWBUF[4] = .UNSOLENBLFLG[4];
0967 5 NEWBUF[5] = .UNSOLENBLFLG[5];
0968 5 UNSOLENBLFLG = .NEWBUF;
0969 5 END
0970 4 ELSE IF .SINGLEFLAG NEQ 0 THEN
0971 5 BEGIN ! READ A SINGLE CHARACTER
0972 5 RETSTATUS =
0973 5 $QIO (CHAN = .RDWRTCHAN,
0974 5 FUNC = IOS$ READVBLK+IOSM BINARY+
0975 5 MAPMODIFIER(.SING[EFLAG[RTP_MOD]]),
0976 5 IOSB = SINGLEFLAG[RTP_IOS],
0977 5 ASTADR = ONECHAR,
0978 5 ASTPRM = .SINGLEFLAG,
0979 5 P1 = SINGLEFLAG[RTP_DAT],
0980 5 P2 = 1);
0981 5 QUIT ON ERROR;
0982 5 SING[EINPROG] = 1;
0983 5 UNSOLPEND = 0; ! NO MORE DATA PENDING
0984 5 END
0985 4 ELSE
0986 4 UNSOLPEND = 1; ! UNSOLICITED DATA PENDING
0987 4 END;
0988 4 RETSTATUS =
0989 4 $QIO (CHAN = .TERMMBXCHAN, ! DO IT AGAIN
0990 4 FUNC = IOS$ READVBLK,
0991 4 ASTADR = TERMMBXMSG,
0992 4 P1 = TERMMBXDATA,
0993 4 P2 = 8);
0994 4 QUIT_ON_ERROR;
0995 4 END
0996 2 ELSE
0997 2 QUIT ! HANGUP - SO QUIT
0998 2 END;

```

```

001C 00000 TERMMBXMSG:
54 00000000G 00 9E 00002 .WORD Save R2,R3,R4 : 0909
53 00000000G 00 9E 00009 MOVAB SYS$QIO, R4
52 0000' CF 9E 00010 MOVAB RETSTATUS, R3
01 F8 A2 B1 00015 MOVAB UNSOLENBLFLG, R2
03 13 00019 BEQL TERMMBXDATA, #1 : 0957
0091 31 0001B BRW 1$
07 A2 95 0001E 1$: TSTB READINPROG : 0959
68 12 00021 BNEQ 4$
51 62 D0 00023 MOVL UNSOLENBLFLG, R1 : 0961
23 13 00026 BEQL 2$
50 04 A2 9A 00028 MOVZBL ATTACHFLAG, R0 : 0962
50 08 A2 C8 0002C B'0'2 SINGLEFLAG, R0
19 12 00030 F'0'2 2$
51 DD 00032 FUSHL R1 : 0964
FCDC CF 01 FB 00034 CALLS #1, READ
FB52 CF 00 FB 00039 CALLS #0, GETBUF : 0965
51 62 D0 0003E MOVL UNSOLENBLFLG, R1 : 0966

```

10	A0	10	A1	7D	00041	MOVQ	16(R1), 16(NEWBUF)	:			
	62		50	D0	00046	MOVL	NEWBUF, UNSOLENBLFLG	:	0968		
			40	11	00049	BRB	4\$	:	0961		
	50	08	A2	D0	0004B	2\$:	MOVL	SINGLEFLAG, R0	:	0970	
			36	13	0004F		BEQL	3\$	:		
			7E	7C	00051		CLRQ	-(SP)	:	0980	
			7E	7C	00053		CLRQ	-(SP)	:		
			01	DD	00055		PUSHL	#1	:		
		1A	A0	9F	00057		PUSHAB	26(R0)	:		
			50	DD	0005A		PUSHL	R0	:		
		0000V	CF	9F	0005C		PUSHAB	ONECHAR	:		
			08	A0	9F	00060	PUSHAB	8(R0)	:		
	7E		11	A0	9A	00063	MOVZBL	17(R0), -(SP)	:		
0000V	CF			01	FB	00067	CALLS	#1, MAPMODIFIER	:		
			71	A0	9F	0006C	PUSHAB	113(R0)	:		
	7E	00000000G		00	3C	0006F	MOVZWL	RDWRTCHAN, -(SP)	:		
				7E	D4	00076	CLRL	-(SP)	:		
	64			0C	FB	00078	CALLS	#12, SYSSQIO	:		
	63			50	D0	0007B	MOVL	R0, RETSTATUS	:		
	2E			63	E9	0007E	BLBC	RETSTATUS, 5\$	:		
05	A2			01	B0	00081	MOVW	#1, SINGLEINPROG	:	0982	
				04	11	00085	BRB	4\$	:	0970	
06	A2			01	90	00087	5\$:	MOVB	#1, UNSOLPEND	:	0986
				7E	7C	0008B	4\$:	CLRQ	-(SP)	:	0993
				7E	7C	0008D		CLRQ	-(SP)	:	
				08	DD	0008F		PUSHL	#8	:	
		F8		A2	9F	00091		PUSHAB	TERMMBXDATA	:	
				7E	D4	00094		CLRL	-(SP)	:	
		FF66		CF	9F	00096		PUSHAB	TERMMBXMSG	:	
	7E			31	7D	0009A		MOVQ	#49, -(SP)	:	
	7E	00000000G		00	3C	0009D		MOVZWL	TERMMBXCHAN, -(SP)	:	
				7E	D4	000A4		CLRL	-(SP)	:	
	64			0C	FB	000A6		CALLS	#12, SYSSQIO	:	
	63			50	D0	000A9		MOVL	R0, RETSTATUS	:	
	19			63	E8	000AC		BLBS	RETSTATUS, 6\$	:	
				7E	D4	000AF	5\$:	CLRL	-(SP)	:	0996
00000000G	00			01	FB	000B1		CALLS	#1, SYSSSETAST	:	
00000000G	00			01	90	000B8		MOVB	#1, WAKEFLAG	:	
				7E	7C	000BF		CLRQ	-(SP)	:	
00000000G	00			02	FB	000C1		CALLS	#2, SYSSWAKE	:	
				04	000C8	6\$:		RET	:	0998	

; Routine Size: 201 bytes, Routine Base: \$CODE\$ + 05F9

```

: 1011 0999 1 ROUTINE BROADCAST(BUFFER): NOVALUE =
: 1012 1000 1 ++
: 1013 1001 1
: 1014 1002 1 Functional Description:
: 1015 1003 1 Issue a broadcast function to the terminal.
: 1016 1004 1
: 1017 1005 1 Calling Sequence:
: 1018 1006 1 standard
: 1019 1007 1
: 1020 1008 1 Input Parameters:
: 1021 1009 1 BUFFER = address of the link buffer
: 1022 1010 1
: 1023 1011 1 Implicit Inputs:
: 1024 1012 1 none
: 1025 1013 1
: 1026 1014 1 Output Parameters:
: 1027 1015 1 none
: 1028 1016 1
: 1029 1017 1 Implicit Outputs:
: 1030 1018 1 none
: 1031 1019 1
: 1032 1020 1 Routines Called:
: 1033 1021 1 QIODONE
: 1034 1022 1
: 1035 1023 1 Routine Value:
: 1036 1024 1 none
: 1037 1025 1
: 1038 1026 1 Signals:
: 1039 1027 1 none
: 1040 1028 1
: 1041 1029 1 Side Effects:
: 1042 1030 1 none
: 1043 1031 1
: 1044 1032 1 --
: 1045 1033 2 BEGIN
: 1046 1034 2 MAP BUFFER: REF RTP_BUF;
: 1047 1035 2 LOCAL
: 1048 1036 2 BRDCSTDESC: VECTOR[2];
: 1049 1037 2 BRDCSTDESC[0] = .BUFFER[RTP_TCT]; ! COUNT
: 1050 1038 2 BRDCSTDESC[1] = BUFFER[RTP_DAT]; ! DATA ADDRESS
: 1051 P 1039 2 BUFFER[RTP_IOS] = $BRDCST (MSGBUF = BRDCSTDESC, ! BROADCAST IT
: 1052 1040 2 DEVNAM = TTYDESC);
: 1053 1041 2 QIODONE(.BUFFER); ! CLEAN UP
: 1054 1042 1 END;

```

.EXTRN SYSSBRDCST

0004 0000 BROADCAST:

	5E		04	C2	00002	.WORD	Save R2		: 0999
	52					SUPL2	#4, SP		: 1037
	7E	04	AC	D0	00005	MCVL	BUFFER, R2		: 1038
		18	A2	3C	00009	MOVZWL	24(R2), BRDCSTDESC		: 1040
04	AE	1A	A2	9E	0000D	MOVAB	26(R2), BRDCSTDESC+4		
				20	DD	PUSHL	#32		
				7E	D4	CLRL	-(SP)		

RSXRT  
V04-000

D 5  
16-Sep-1984 02:18:51  
14-Sep-1984 13:04:57

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[RTPAD.SRC]RSXRT.B32;1  
Page 38  
(13)

		00000000G	00	9F	00016
			OC	AE	9F 0001C
00000000G	00			04	FB 0001F
08	A2			50	B0 00026
				52	DD 0002A
FD70	CF			01	FB 0002C
				04	00031

PUSHAB	TTYDESC	:
PUSHAB	BRDCSTDESC	:
CALLS	#4, SYS\$BRDCST	:
MOVW	R0, 8(R2)	:
PUSHL	R2	: 1041
CALLS	#1, QIDONE	:
RET		: 1042

: Routine Size: 50 bytes, Routine Base: \$CODE\$ + 06C2

: 1055 1043 1



```

: 1057      1044 1 ROUTINE CNTRLCAST: NOVALUE =
: 1058      1045 1 ++
: 1059      1046 1
: 1060      1047 1 Functional Description:
: 1061      1048 1     Handle the AST indicating that a control-C was typed on the terminal.
: 1062      1049 1
: 1063      1050 1 Calling Sequence:
: 1064      1051 1     standard
: 1065      1052 1
: 1066      1053 1 Input Parameters:
: 1067      1054 1     none
: 1068      1055 1
: 1069      1056 1 Implicit Inputs:
: 1070      1057 1     none
: 1071      1058 1
: 1072      1059 1 Output Parameters:
: 1073      1060 1     none
: 1074      1061 1
: 1075      1062 1 Implicit Outputs:
: 1076      1063 1     none
: 1077      1064 1
: 1078      1065 1 Routines Called:
: 1079      1066 1     none
: 1080      1067 1
: 1081      1068 1 Routine Value:
: 1082      1069 1     none
: 1083      1070 1
: 1084      1071 1 Signals:
: 1085      1072 1     none
: 1086      1073 1
: 1087      1074 1 Side Effects:
: 1088      1075 1     A message is sent to the host and the control-C AST is enabled. An
: 1089      1076 1     error will cause a $WAKE to be issued to abort the program.
: 1090      1077 1
: 1091      1078 1 --
: 1092      1079 2 BEGIN
: 1093      1080 2 RETSTATUS =
: 1094      1081 2 $QIO (CHAN = .LINKCHAN, ! TELL HOST
: 1095      1082 2     FUNC = IOS_WRITEVBLK,
: 1096      1083 2     P1 = CNTRLMSG,
: 1097      1084 2     P2 = 4);
: 1098      1085 2 QUIT ON ERROR;
: 1099      1086 2 RETSTATUS =
: 1100      1087 2 $QIO (CHAN = .CNTRLCHAN, ! REENABLE IT
: 1101      1088 2     FUNC = IOS_SETMODE+IOSM_CNTRLCAST,
: 1102      1089 2     P1 = CNTRLCAST);
: 1103      1090 2 QUIT_ON_ERROR;
: 1104      1091 1 END;

```

P  
P  
P  
P  
P  
P  
P  
P

000C 00000 CNTRLCAST:

```

53 00000000G 00 9E 00002 .WORD Save R2,R3
52 00000000G 00 9E 00009 MOVAB SYS$QIO, R3
MOVAB RETSTATUS, R2

```

: 1044  
:  
:

		7E	7C	00010	CLRQ	-(SP)	
		7E	7C	00012	CLRQ	-(SP)	
		04	DD	00014	PUSHL	#4	1084
			CF	9F	PUSHAB	CNTRLMSG	
		7E	7C	0001A	CLRQ	-(SP)	
		7E	30	7D	MOVQ	#48, -(SP)	
		7E	00	3C	MOVZWL	LINKCHAN, -(SP)	
			7E	D4	CLRL	-(SP)	
		63	0C	FB	CALLS	#12, SYSSQIO	
		62	50	D0	MOVL	R0, RETSTATUS	
		24	62	E9	BLBC	RETSTATUS, 1\$	
			7E	7C	CLRQ	-(SP)	1089
			7E	7C	CLRQ	-(SP)	
			7E	D4	CLRL	-(SP)	
			C6	AF	PUSHAB	CNTRLCAST	
			7E	7C	CLRQ	-(SP)	
			7E	D4	CLRL	-(SP)	
		7E	8F	3C	MOVZWL	#291, -(SP)	
		7E	00	3C	MOVZWL	CNTRLCHAN, -(SP)	
			7E	D4	CLRL	-(SP)	
		63	0C	FB	CALLS	#12, SYSSQIO	
		62	50	D0	MOVL	R0, RETSTATUS	
		19	62	E8	BLBS	RETSTATUS, 2\$	
			7E	D4	CLRL	-(SP)	
		00000000G	00	01	CALLS	#1, SYSSSETAST	
		00000000G	00	01	MOVB	#1, WAKEFLAG	
			7E	7C	CLRQ	-(SP)	
		00000000G	00	02	CALLS	#2, SYSSWAKE	
			04	0006E	RET		1091

; Routine Size: 111 bytes, Routine Base: \$CODE\$ + 06F4

```

: 1106      1092 1 ROUTINE CNTRL YAST: NOVALUE =
: 1107      1093 1 ++
: 1108      1094 1
: 1109      1095 1 Functional Description:
: 1110      1096 1   Handle the AST indicating that a control-Y was typed on the terminal.
: 1111      1097 1
: 1112      1098 1 Calling Sequence:
: 1113      1099 1   standard
: 1114      1100 1
: 1115      1101 1 Input Parameters:
: 1116      1102 1   none
: 1117      1103 1
: 1118      1104 1 Implicit Inputs:
: 1119      1105 1   none
: 1120      1106 1
: 1121      1107 1 Output Parameters:
: 1122      1108 1   none
: 1123      1109 1
: 1124      1110 1 Implicit Outputs:
: 1125      1111 1   none
: 1126      1112 1
: 1127      1113 1 Routines Called:
: 1128      1114 1   none
: 1129      1115 1
: 1130      1116 1 Routine Value:
: 1131      1117 1   none
: 1132      1118 1
: 1133      1119 1 Signals:
: 1134      1120 1   none
: 1135      1121 1
: 1136      1122 1 Side Effects:
: 1137      1123 1   A SWAKE will be issued to abort the program.
: 1138      1124 1
: 1139      1125 1 --
: 1140      1126 2 BEGIN
: 1141      1127 2 QUIT;
: 1142      1128 1 END;

```

```

                                0000 00000 CNTRL YAST:
                                .WORD   Save nothing
                                CLR      -(SP)
                                CALLS   #1, SYSSSETAST
                                MOV      #1, WAKEFLAG
                                CLR      -(SP)
                                CALLS   #2, SYSSWAKE
                                RET
                                : 1092
                                : 1126
                                :
                                :
                                :
                                : 1128

```

: Routine Size: 28 bytes, Routine Base: \$CODE\$ + 0763

: 1143 1129 1

```

1145 1130 1 ROUTINE CANCEL(BUFFER): NOVALUE =
1146 1131 1 ++
1147 1132 1
1148 1133 1 Functional Description:
1149 1134 1 Cancel I/O's as requested by RSX.
1150 1135 1
1151 1136 1 Calling Sequence:
1152 1137 1 standard
1153 1138 1
1154 1139 1 Input Parameters:
1155 1140 1 BUFFER = address of the link buffere
1156 1141 1
1157 1142 1 Implicit Inputs:
1158 1143 1 IOQUEUE
1159 1144 1 CURRENTIO
1160 1145 1
1161 1146 1 Output Parameters:
1162 1147 1 none
1163 1148 1
1164 1149 1 Implicit Outputs:
1165 1150 1 none
1166 1151 1
1167 1152 1 Routines Called:
1168 1153 1 FREEBUF
1169 1154 1
1170 1155 1 Routine Value:
1171 1156 1 none
1172 1157 1
1173 1158 1 Signals:
1174 1159 1 none
1175 1160 1
1176 1161 1 Side Effects:
1177 1162 1 A completion message is sent to the host.
1178 1163 1
1179 1164 1 --
1180 1165 2 BEGIN
1181 1166 2 MAP BUFFER: REF RTP_BUF;
1182 1167 2 LOCAL
1183 1168 2 IOBUF: REF RTP_BUF;
1184 1169 2 IF .BUFFER[RTP_IDN] EQC 255 THEN
1185 1170 3 BEGIN ! KILL ALL I/O
1186 1171 3 $CANCEL (CHAN = .RDWRTCHAN); ! CANCEL CURRENT I/O
1187 1172 3 WHILE .IOQUEUE[0] NEQ IOQUEUE DO
1188 1173 4 BEGIN
1189 1174 4 REMQUE(.IOQUEUE,IOBUF); ! GET NEXT I/O
1190 1175 4 FREEBUF(.IOBUF);
1191 1176 3 END;
1192 1177 3 END
1193 1178 2 ELSE
1194 1179 3 BEGIN ! KILL ONLY ONE I/O
1195 1180 3 IF .CURRENTIO NEQ 0
1196 1181 3 AND .CURRENTIO[RTP_IDN] EQL .BUFFER[RTP_IDN] THEN
1197 1182 4 BEGIN
1198 1183 4 CURRENTIO = 0;
1199 1184 4 $CANCEL (CHAN = .RDWRTCHAN);
1200 1185 4 END
1201 1186 3 ELSE

```

```

: 1202      1187      4
: 1203      1188      4
: 1204      1189      5
: 1205      1190      5
: 1206      1191      5
: 1207      1192      6
: 1208      1193      6
: 1209      1194      6
: 1210      1195      5
: 1211      1196      4
: 1212      1197      3
: 1213      1198      2
: 1214      1199      2
: 1215      P 1200      2
: 1216      P 1201      2
: 1217      P 1202      2
: 1218      P 1203      2
: 1219      P 1204      2
: 1220      P 1205      2
: 1221      1206      2
: 1222      1207      1

```

```

BEGIN
WHILE .IOQUEUE NEQ IOQUEUE DO
  BEGIN
  IOBUF = .IOQUEUE;
  IF .IOBUF[RTP_IDN] EQL .BUFFER[RTP_IDN] THEN
    BEGIN
    REMQUE(.IOBUF,IOBUF);
    FREEBUF(.IOBUF);
    END;
  END;
END;

END;
BUFFER[RTP_FLG] = 0;
SQIO (CRAN = .LINKCHAN, ! WRITE TO LINK
      FUNC = IOS WRITEVBLK,
      IOSB = BUFFER[RTP_IOS],
      ASTADR = LINKWRTDNE,
      ASTPRM = BUFFER,
      P1 = BUFFER[RTP_FNC],
      P2 = 5);
END;

```

```

                                .EXTRN SYSS$CANCEL
                                007C 00000 CANCEL: .WORD Save R2,R3,R4,R5,R6
56 00000000G 00 9E 00002 MOVAB RDWRTCHAN, R6
55 00000000G 00 9E 00009 MOVAB SYSS$CANCEL, R5
54 0000' CF 9E 00010 MOVAB IOQUEUE, R4
52 04 AC D0 00015 MOVL BUFFER, R2
FF 8F 14 A2 91 00019 CMPB 20(R2), #255
7E 66 3C 00020 BNEQ 2$
65 01 FB 00023 MOVZWL RDWRTCHAN, -(SP)
50 64 9E 00026 1$: CALLS #1, SYSS$CANCEL
50 64 D1 00029 MOVAB IOQUEUE, R0
47 13 0002C CMPL IOQUEUE, R0
53 00 B4 0F 0002E BEQL 4$
F9F5 CF 01 FB 00034 REMQUE @IOQUEUE, IOBUF
EB 11 00039 PUSHL IOBUF
50 F8 A4 D0 0003B 2$: CALLS #1, FREEBUF
14 A2 14 A0 91 00041 BRB 1$
0B 12 00046 MOVL CURRENTIO, R0
F8 A4 D4 00048 BEQL 3$
7E 66 3C 0004B BNEQ 3$
65 01 FB 0004E CLRL CURRENTIO
22 11 00051 MOVZWL RDWRTCHAN, -(SP)
50 64 9E 00053 3$: CALLS #1, SYSS$CANCEL
50 64 D1 00056 BRB 4$
1A 13 00059 MOVAB IOQUEUE, R0
53 64 D0 0005B CMPL IOQUEUE, R0
50 04 AC D0 0005E BEQL 4$
14 A0 14 A3 91 00062 MOVL IOQUEUE, IOBUF
EA 12 00067 MOVL BUFFER, R0
CMPB 20(IOBUF), 20(R0)
BNEQ 3$

```

```

: 1130
:
:
: 1169
:
: 1171
:
: 1172
:
: 1174
: 1175
:
: 1172
: 1180
:
: 1181
:
: 1183
: 1184
:
: 1180
: 1188
:
: 1190
: 1191
:

```

RSXRT  
V04-000

J 5  
16-Sep-1984 02:18:51  
14-Sep-1984 13:04:57

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[RTPAD.SRC]RSXRT.B32;1

Page 44  
(16)

	53		63	0F	00069	REMQUE	(IOBUF), IOBUF	:	1193	
			53	DD	0006C	PUSHL	IOBUF	:	1194	
F98B	CF		01	FB	0006E	CALLS	#1, FREEBUF	:		
			DE	11	00073	BRB	3\$	:	1188	
	50	04	AC	DD	00075	4\$:	MOVL	BUFFER, R0	:	1199
		12	A0	94	00079		CLRB	18(R0)	:	
			7E	7C	0007C		CLRQ	-(SP)	:	1206
			7E	7C	0007E		CLRQ	-(SP)	:	
			05	DD	00080		PUSHL	#5	:	
		10	A0	9F	00082		PUSHAB	16(R0)	:	
			50	DD	00085		PUSHL	R0	:	
		FD63	CF	9F	00087		PUSHAB	LINKWRTDONE	:	
		08	A0	9F	0008B		PUSHAB	8(R0)	:	
			30	DD	0008E		PUSHL	#48	:	
	7E	00000000G	00	3C	00090		MOVZWL	LINKCHAN, -(SP)	:	
			7E	D4	00097		CLRL	-(SP)	:	
00000000G	00		0C	FB	00099		CALLS	#12, SYSSQIO	:	
			04	00	00A0		RET	:	1207	

; Routine Size: 161 bytes, Routine Base: \$CODE\$ + 077F

RS  
VC

```

: 1224      1208 1 ROUTINE MAPMODIFIER(RSXMOD) =
: 1225      1209 1 ++
: 1226      1210 1
: 1227      1211 1 Functional Description:
: 1228      1212 1 Convert RSX function code modifiers to VMS format.
: 1229      1213 1
: 1230      1214 1 Calling Sequence:
: 1231      1215 1 standard
: 1232      1216 1
: 1233      1217 1 Input Parameters:
: 1234      1218 1 RSXMOD = RSX modifiers
: 1235      1219 1
: 1236      1220 1 Implicit Inputs:
: 1237      1221 1 none
: 1238      1222 1
: 1239      1223 1 Output Parameters:
: 1240      1224 1 none
: 1241      1225 1
: 1242      1226 1 Implicit Outputs:
: 1243      1227 1 none
: 1244      1228 1
: 1245      1229 1 Routines Called:
: 1246      1230 1 none
: 1247      1231 1
: 1248      1232 1 Routine Value:
: 1249      1233 1 VMS function code modifier
: 1250      1234 1
: 1251      1235 1 Signals:
: 1252      1236 1 none
: 1253      1237 1
: 1254      1238 1 Side Effects:
: 1255      1239 1 none
: 1256      1240 1
: 1257      1241 1 --
: 1258      1242 2 BEGIN
: 1259      1243 2 LOCAL
: 1260      1244 2 VMSMOD: ;
: 1261      1245 2 VMSMOD = IOSM TRMNOECHO;
: 1262      1246 2 IF (.RSXMOD AND RM RNE) NEQ 0 THEN
: 1263      1247 2 VMSMOD = .VMSMOD+IOSM_NOECHO;
: 1264      1248 2 RETURN(.VMSMOD);
: 1265      1249 1 END;

```

```

                                0000 00000 MAPMODIFIER:
                                .WORD Save nothing
                                MOVZWL #4096, VMSMOD : 1208
04                               50 1000 8F 3C 00002 BBC #36, RSXMOD, 1$ : 1245
                                6C                               24 E1 00007 : 1246
                                50 40 A0 9E 0000B MOVAB 64(R0), VMSMOD : 1247
                                04 0000F 1$: RET : 1249

```

: Routine Size: 16 bytes, Routine Base: \$CODE\$ + 0820

```

: 1267 1250 1 ROUTINE ATTACH(BUFFER): NOVALUE =
: 1268 1251 1 ++
: 1269 1252 1
: 1270 1253 1 Functional Description:
: 1271 1254 1 Handle the RSX attach and detach functions.
: 1272 1255 1
: 1273 1256 1 Calling Sequence:
: 1274 1257 1 standard
: 1275 1258 1
: 1276 1259 1 Input Parameters:
: 1277 1260 1 BUFFER = address of the link buffer
: 1278 1261 1
: 1279 1262 1 Implicit Inputs:
: 1280 1263 1 CURRENTIO
: 1281 1264 1 UNSOLPEND
: 1282 1265 1
: 1283 1266 1 Output Parameters:
: 1284 1267 1 none
: 1285 1268 1
: 1286 1269 1 Implicit Outputs:
: 1287 1270 1 ATTACHFLAG
: 1288 1271 1
: 1289 1272 1 Routines Called:
: 1290 1273 1 TERMMBXMSG
: 1291 1274 1 FREEBUF
: 1292 1275 1 NEXTIO
: 1293 1276 1
: 1294 1277 1 Routine Value:
: 1295 1278 1 none
: 1296 1279 1
: 1297 1280 1 Signals:
: 1298 1281 1 none
: 1299 1282 1
: 1300 1283 1 Side Effects:
: 1301 1284 1 The request may be queued for later action.
: 1302 1285 1 If the detach reenables unsolicited input, pending data may be read.
: 1303 1286 1
: 1304 1287 1 --
: 1305 1288 2 BEGIN
: 1306 1289 2 MAP BUFFER: REF RTP_BUF;
: 1307 1290 2 IF .CURRENTIO EQL 0 THEN
: 1308 1291 3 BEGIN
: 1309 1292 3 IF .BUFFER[RTP MOD] NEQ RM_DET THEN
: 1310 1293 3 ATTACHFLAG = 1
: 1311 1294 3 ELSE
: 1312 1295 4 BEGIN
: 1313 1296 4 ATTACHFLAG = 0;
: 1314 1297 4 IF (.UNSOLPEND NEQ 0) OR (.INDDATA NEQ 0) THEN
: 1315 1298 5 BEGIN
: 1316 1299 5 ! DATA ALREADY PENDING
: 1317 1300 5 TERMMBXDATA[0] = MSG$_TRMUNSOLIC;
: 1318 1301 4 TERMMBXMSG();
: 1319 1302 3 END;
: 1320 1303 3 FREEBUF(.BUFFER);
: 1321 1304 3 NEXTIO(); ! CHECK FOR A PENDING I/O
: 1322 1305 3 END
: 1323 1306 2 ELSE

```



RSXRT  
V04-000

M 5  
16-Sep-1984 02:18:51  
14-Sep-1984 13:04:57

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[RTPAD.SRC]RSXRT.B32;1 (18)  
Page 47

: 1324 1307 2  
: 1325 1308 1

END: INSQUE(.BUFFER,.IOQUEUE[1]); ! QUEUE IT FOR LATER

				0004 00000	ATTACH: .WORD	Save R2	: 1250
	52	0000'	CF	9E 00002	MOVAB	ATTACHFLAG, R2	
		08	A2	D5 00007	TSTL	CURRENTIO	: 1290
			33	12 0000A	BNEQ	4\$	
	50	04	AL	D0 0000C	MOVL	BUFFER, R0	: 1292
80	8F	11	A0	91 00010	CMPB	17(R0), #128	
			05	13 00015	BEQL	1\$	
	62		01	90 00017	MOVB	#1, ATTACHFLAG	: 1293
			15	11 0001A	BRB	3\$	
			62	94 0001C	1\$: CLRB	ATTACHFLAG	: 1296
		02	A2	95 0001E	TSTB	UNSOLPEND	: 1297
			05	12 00021	BNEQ	2\$	
		0C	A2	D5 00023	TSTL	INDDATA	
			09	13 00026	BEQL	3\$	
F4	A2		01	B0 00028	2\$: MOVW	#1, TERMMBXDATA	: 1299
FD98	CF		00	FB 0002C	CALLS	#0, TERMMBXMSG	: 1300
		04	AC	DD 00031	3\$: PUSHL	BUFFER	: 1303
F944	CF		01	FB 00034	CALLS	#1, FREEBUF	
0000V	CF		00	FB 00039	CALLS	#0, NEXTIO	: 1304
			04	0003E	RET		: 1290
14	B2	04	BC	0E 0003F	4\$: INSQUE	@BUFFER, @IOQUEUE+4	: 1307
			04	00044	PET		: 1308

: Routine Size: 69 bytes, Routine Base: \$CODE\$ + 0830

: 1326 1309 1

```

: 1328 1310 1 ROUTINE READSINGLE(BUFFER): NOVALUE =
: 1329 1311 1 ++
: 1330 1312 1
: 1331 1313 1 Functional Description:
: 1332 1314 1 Enable and disable RSX single character mode.
: 1333 1315 1
: 1334 1316 1 Calling Sequence:
: 1335 1317 1 \ standard
: 1336 1318 1
: 1337 1319 1 Input Parameters:
: 1338 1320 1 BUFFER = address of the link buffer
: 1339 1321 1
: 1340 1322 1 Implicit Inputs:
: 1341 1323 1 CURRENTIO
: 1342 1324 1 UNSOLPEND
: 1343 1325 1 SINGLEINPROG
: 1344 1326 1
: 1345 1327 1 Output Parameters:
: 1346 1328 1 none
: 1347 1329 1
: 1348 1330 1 Implicit Outputs:
: 1349 1331 1 SINGLEFLAG
: 1350 1332 1 UNSOLPEND
: 1351 1333 1
: 1352 1334 1 Routines Called:
: 1353 1335 1 TERMMBXMSG
: 1354 1336 1 FREEBUF
: 1355 1337 1
: 1356 1338 1 Routine Value:
: 1357 1339 1 none
: 1358 1340 1
: 1359 1341 1 Signals:
: 1360 1342 1 none
: 1361 1343 1
: 1362 1344 1 Side Effects:
: 1363 1345 1 The request may be queued for later action.
: 1364 1346 1 If data is pending when the mode is enabled, it is read.
: 1365 1347 1
: 1366 1348 1 --
: 1367 1349 2 BEGIN
: 1368 1350 2 MAP BUFFER: REF RTP_BUF;
: 1369 1351 2 IF .CURRENTIO EQL 0 THEN
: 1370 1352 3 BEGIN
: 1371 1353 3 IF (.BUFFER[RTP_MOD] AND RM_TSC) EQL 0 THEN
: 1372 1354 4 BEGIN
: 1373 1355 4 SINGLEFLAG = .BUFFER; ! ENABLE SINGLE CHARACTERS
: 1374 1356 4 IF .UNSOLPEND NEQ 0 THEN
: 1375 1357 4 TERMMBXMSG(); ! DATA ALREADY PENDING
: 1376 1358 4 UNSOLPEND = 0;
: 1377 1359 4 END
: 1378 1360 3 ELSE
: 1379 1361 4 BEGIN ! DISABLE SINGLE CHARACTER MODE
: 1380 1362 4 FREEBUF(.BUFFER); ! OF NO USE
: 1381 1363 4 IF .SINGLEINPROG EQL 0 THEN
: 1382 1364 4 FREEBUF(.SINGLEFLAG); ! NOT CURRENTLY IN USE
: 1383 1365 4 SINGLEFLAG = 0;
: 1384 1366 3 END;

```

: 1385 1367 3  
: 1386 1368 3  
: 1387 1369 2  
: 1388 1370 2  
: 1389 1371 1

```
NEXTIO();           ! IS ANYTHING ELSE QUEUED
END
ELSE
INSQUE(.BUFFER,.IOQUEUE[1]); ! QUEUE IT FOR LATER
END;
```

		0004 0000 READSINGLE:				
52	0000'	CF	9E	00002	.WORD	Save R2 : 1310
	04	A2	D5	00007	MOVAB	SINGLEFLAG, R2
		38	12	0000A	TSTL	CURRENTIO : 1351
50	04	AC	D0	0000C	BNEQ	5\$
	11	A0	95	00010	MOVL	BUFFER, R0 : 1353
		13	19	00013	TSTB	17(R0)
62	04	AC	D0	00015	BLSS	2\$
	FE	A2	95	00019	MOVL	BUFFER, SINGLEFLAG : 1355
		05	13	0001C	TSTB	UNSOLPEND : 1356
FD61	CF	00	FB	0001E	BEQL	1\$
		FE	A2	94	CALLS	#0, TERMMBXMSG : 1357
		14	11	00026	CLRB	UNSOLPEND : 1358
		04	AC	DD	BRB	4\$ : 1353
F908	CF	01	FB	0002B	PUSHL	BUFFER : 1362
		FD	A2	95	CALLS	#1, FREEBUF
		07	12	00033	TSTB	SINGLEINPROG : 1363
		62	DD	00035	BNEQ	3\$
F8FC	CF	01	FB	00037	PUSHL	SINGLEFLAG : 1364
		62	D4	0003C	CALLS	#1, FREEBUF
0000V	CF	00	FB	0003E	CLRL	SINGLEFLAG : 1365
		04	00043	CALLS	#0, NEXTIO : 1367	
10	B2	04	BC	0E	RET	: 1351
		04	00049	INSQUE	@BUFFER, @IOQUEUE+4 : 1370	
				RET		: 1371

: Routine Size: 74 bytes, Routine Base: \$CODE\$ + 0875

: 1390 1372 1

```

: 1392      1373 1 ROUTINE ONECHAR(BUFFER): NOVALUE =
: 1393      1374 1 ++
: 1394      1375 1
: 1395      1376 1 Functional Description:
: 1396      1377 1     Handle the completion of a single character mode read.
: 1397      1378 1
: 1398      1379 1 Calling Sequence:
: 1399      1380 1     standard
: 1400      1381 1
: 1401      1382 1 Input Parameters:
: 1402      1383 1     BUFFER = address of the link buffer
: 1403      1384 1
: 1404      1385 1 Implicit Inputs:
: 1405      1386 1     SINGLEFLAG
: 1406      1387 1
: 1407      1388 1 Output Parameters:
: 1408      1389 1     none
: 1409      1390 1
: 1410      1391 1 Implicit Outputs:
: 1411      1392 1     SINGLEINPROG
: 1412      1393 1
: 1413      1394 1 Routines Called:
: 1414      1395 1     QIODONE
: 1415      1396 1     FREEBUF
: 1416      1397 1
: 1417      1398 1 Routine Value:
: 1418      1399 1     none
: 1419      1400 1
: 1420      1401 1 Signals:
: 1421      1402 1     none
: 1422      1403 1
: 1423      1404 1 Side Effects:
: 1424      1405 1     none
: 1425      1406 1
: 1426      1407 1 --
: 1427      1408 2 BEGIN
: 1428      1409 2 LOCAL
: 1429      1410 2     NEWBUF: REF VECTOR;
: 1430      1411 2     MAP BUFFER: REF VECTOR;
: 1431      1412 2     SINGLEINPROG = 0;
: 1432      1413 2     NEWBUF = GETBUF();           ! GET A NEW BUFFER
: 1433      1414 2     NEWBUF[4] = .BUFFER[4];
: 1434      1415 2     NEWBUF[5] = .BUFFER[5];
: 1435      1416 2     NEWBUF[6] = .BUFFER[6];
: 1436      1417 2     QIODONE(.NEWBUF);
: 1437      1418 2     IF .SINGLEFLAG EQL 0 THEN
: 1438      1419 2         FREEBUF(.BUFFER);           ! SINGLE CHAR MODE WAS DISABLED
: 1439      1420 1     END;

```

```

                                0004 0000 ONECHAR: .WORD    Save R2
F8BF    CF    0000'    CF    94 0000                CLRB    SINGLEINPROG
                                00    FB 00006         CALLS  #0, GETBUF
                                04    AC  DO 0000B      MOVL   BUFFER, R2

```

```

: 1373
: 1412
: 1413
: 1414

```

RSXRT  
V04-000

D 6  
16-Sep-1984 02:18:51  
14-Sep-1984 13:04:57

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[RTPAD.SRC]RSXRT.B32;1 Page 51  
(20)

10	A0	10	A2	7D	0000F	MOVQ	16(R2), 16(NEWBUF)	:	
18	A0	18	A2	DD	00014	MOVL	24(R2), 24(NEWBUF)	:	1416
			50	DD	00019	PUSHL	NEWBUF	:	1417
FB84	CF		01	FB	0001B	CALLS	#1, @IDONE	:	
		0000'	CF	D5	00020	TSTL	SINGLEFLAG	:	1418
			07	12	00024	BNEQ	1\$	:	
			52	DD	00026	PUSHL	R2	:	1419
F8C1	CF		C1	FB	00028	CALLS	#1, FREEBUF	:	
			04	0002D	1\$:	RET		:	1420

; Routine Size: 46 bytes, Routine Base: \$CODE\$ + 08BF

RS  
VO

.....

```

: 1441 1421 1 ROUTINE TERMINATOR(RSXMOD) =
: 1442 1422 1 ++
: 1443 1423 1
: 1444 1424 1 Functional Description:
: 1445 1425 1 Provide the correct terminator mask for an RSX read operation.
: 1446 1426 1
: 1447 1427 1 Calling Sequence:
: 1448 1428 1 standard
: 1449 1429 1
: 1450 1430 1 Input Parameters:
: 1451 1431 1 RSXMOD = RSX function modifiers
: 1452 1432 1
: 1453 1433 1 Implicit Inputs:
: 1454 1434 1 none
: 1455 1435 1
: 1456 1436 1 Output Parameters:
: 1457 1437 1 none
: 1458 1438 1
: 1459 1439 1 Implicit Outputs:
: 1460 1440 1 none
: 1461 1441 1
: 1462 1442 1 Routines Called:
: 1463 1443 1 none
: 1464 1444 1
: 1465 1445 1 Routine Value:
: 1466 1446 1 address of the descriptor for the terminator mask
: 1467 1447 1
: 1468 1448 1 Signals:
: 1469 1449 1 none
: 1470 1450 1
: 1471 1451 1 Side Effects:
: 1472 1452 1 none
: 1473 1453 1
: 1474 1454 1 --
: 1475 1455 2 BEGIN
: 1476 1456 2 IF (.RSXMOD AND RM RTC) NEQ 0 THEN
: 1477 1457 3 RETURN(STERMDESC) ! TERMINATE ON CONTROL CHARACTERS
: 1478 1458 2 ELSE
: 1479 1459 2 RETURN(NTERMDESC); ! NORMAL TERMINATORS
: 1480 1460 1 END;

```

```

                                0000 0000 TERMINATOR:
                                .WORD Save nothing
06          6C          23 E1 00002          BBC #35, RSXMOD, 1$
          50          0000' CF 9E 00006          MOVAB STERMDESC, R0
          50          0000' CF 9E 0000C 1$:    RET
          04 00011          MOVAB NTERMDESC, R0
          04 00011          RET
: 1421
: 1456
: 1457
: 1459
: 1460

```

; Routine Size: 18 bytes, Routine Base: \$CODE\$ + 08ED

```

: 1482 1461 1 ROUTINE UNSUPPORTED(BUFFER): NOVALUE =
: 1483 1462 1 ++
: 1484 1463 1
: 1485 1464 1 Functional Description:
: 1486 1465 1 Return an error message to the host for unsupported functions.
: 1487 1466 1
: 1488 1467 1 Calling Sequence:
: 1489 1468 1 standard
: 1490 1469 1
: 1491 1470 1 Input Parameters:
: 1492 1471 1 BUFFER = address of the link buffer
: 1493 1472 1
: 1494 1473 1 Implicit Inputs:
: 1495 1474 1 none
: 1496 1475 1
: 1497 1476 1 Output Parameters:
: 1498 1477 1 none
: 1499 1478 1
: 1500 1479 1 Implicit Outputs:
: 1501 1480 1 RETSTATUS
: 1502 1481 1
: 1503 1482 1 Routines Called:
: 1504 1483 1 none
: 1505 1484 1
: 1506 1485 1 Routine Value:
: 1507 1486 1 none
: 1508 1487 1
: 1509 1488 1 Signals:
: 1510 1489 1 none
: 1511 1490 1
: 1512 1491 1 Side Effects:
: 1513 1492 1 If there is an error on the write to the link, a $WAKE is issued to
: 1514 1493 1 abort the program.
: 1515 1494 1
: 1516 1495 1 --
: 1517 1496 2 BEGIN
: 1518 1497 2 MAP BUFFER: REF RTP_BUF;
: 1519 1498 2 RETSTATUS =
: 1520 P 1499 2 $QIO (CHAN = .LINKCHAN, ! WRITE TO LINK
: 1521 P 1500 2 FUNC = IOS$ WRITEVBLK,
: 1522 P 1501 2 IOSB = BUFFER[RTP IOS],
: 1523 P 1502 2 ASTADR = LINKWRTDONE,
: 1524 P 1503 2 ASTPRM = .BUFFER,
: 1525 P 1504 2 P1 = BUFFER[RTP_FNC],
: 1526 1505 2 P2 = 128);
: 1527 1506 2 IF .RETSTATUS EQL $$$_ABORT THEN
: 1528 1507 2 RETURN; ! Link gone - mailbox msg will tell why
: 1529 1508 2 QUIT_ON_ERROR;
: 1530 1509 1 END;

```

```

0000 0000 UNSUPPORTED:
7E 7C 0002 .WORD Save nothing
(CLRQ -(SP)

```

```

: 1461
: 1505

```

7E	04	7E	80	7E	7C	00004	CLRQ	-(SP)	
		AC		8F	9A	00006	MOVZBL	#128, -(SP)	
			04	10	C1	0000A	ADDL3	#16, BUFFER, -(SP)	
			FC58	AC	DD	0000F	PUSHL	BUFFER	
7E	04	AC		CF	9F	00012	PUSHAB	LINKWRTDONE	
				08	C1	00016	ADDL3	#8, BUFFER, -(SP)	
				30	DD	0001B	PUSHL	#48	
		7E	00000000G	00	3C	0001D	MOVZWL	LINKCHAN, -(SP)	
				7E	D4	00024	CLRL	-(SP)	
00000000G		00		0C	FB	00026	CALLS	#12, SYSSQIO	
00000000G		00		50	D0	0002D	MOVL	R0, RETSTATUS	
		2C		50	D1	00034	CMPL	R0, #44	1506
				1C	13	00037	BEQL	1\$	
		19		50	E8	00C39	BLBS	R0, 1\$	1507
				7E	D4	0003C	CLRL	-(SP)	
00000000G		00		01	FB	0003E	CALLS	#1, SYSSSETAST	
00000000G		00		01	90	00045	MOVB	#1, WAKEFLAG	
				7E	7C	0C04C	CLRQ	-(SP)	
00000000G		00		02	FB	0004E	CALLS	#2, SYSSWAKE	
				04	00055	1\$:	RET		1509

; Routine Size: 86 bytes, Routine Base: \$CODE\$ + 08FF



```

1532 1510 1 ROUTINE NEXTIO: NOVALUE =
1533 1511 1 +-
1534 1512 1
1535 1513 1 Functional Description:
1536 1514 1 Perform the next I/O on the queue.
1537 1515 1
1538 1516 1 Calling Sequence:
1539 1517 1 standard
1540 1518 1
1541 1519 1 Input Parameters:
1542 1520 1 none
1543 1521 1
1544 1522 1 Implicit Inputs:
1545 1523 1 IOQUEUE
1546 1524 1 CURRENTIO
1547 1525 1
1548 1526 1 Output Parameters:
1549 1527 1 none
1550 1528 1
1551 1529 1 Implicit Outputs:
1552 1530 1 none
1553 1531 1
1554 1532 1 Routines Called:
1555 1533 1 WRITE
1556 1534 1 READ
1557 1535 1 READPROMPT
1558 1536 1 ATTACH
1559 1537 1 READSINGLE
1560 1538 1 FREEBUF
1561 1539 1
1562 1540 1 Routine Value:
1563 1541 1 none
1564 1542 1
1565 1543 1 Signals:
1566 1544 1 none
1567 1545 1
1568 1546 1 Side Effects:
1569 1547 1 none
1570 1548 1
1571 1549 1 --
1572 1550 2 BEGIN
1573 1551 2 LOCAL
1574 1552 2 NEWIO: REF RTP BUF;
1575 1553 2 IF (.IOQUEUE[0] NEQ IOQUEUE) AND (.CURRENTIO EQL 0) THEN
1576 1554 2 BEGIN ! TAKE AN I/O OFF THE QUEUE
1577 1555 2 REMQUE(.IOQUEUE,NEWIO);
1578 1556 2 CASE .NEWIO[RTP_FNC] FROM 3 TO 9 OF
1579 1557 2 SET
1580 1558 2 [RF_WTD]: WRITE(.NEWIO);
1581 1559 2 [RF_RDD]: READ(.NEWIO);
1582 1560 2 [RF_WRD]: READPROMPT(.NEWIO);
1583 1561 2 [RF_ATT]: ATTACH(.NEWIO);
1584 1562 2 [RF_RSC]: READSINGLE(.NEWIO);
1585 1563 2 [INRANGE]: FREEBUF(.NEWIO);
1586 1564 2 TES;
1587 1565 2 END;
1588 1566 1 END;

```

		52		0004	00000	NEXTIO:	.WORD	Save R2	:	1510		
		50	0000	CF	9E	00002	MOVAB	IOQUEUE, R2	:			
		50		62	9E	00007	MOVAB	IOQUEUE, R0	:	1553		
				62	D1	0000A	CMPL	IOQUEUE, R0	:			
				4B	13	0000D	BEQL	8\$	:			
			F8	A2	D5	0000F	TSTL	CURRENTIO	:			
				46	12	00012	BNEQ	8\$	:			
		50	00	B2	0F	00014	REMQUE	@IOQUEUE, NEWIO	:	1555		
0036	06	03	10	A0	8F	00018	CASEB	16(NEWIO), #3, #6	:	1556		
	001E	0016		000E		0001D	.WORD	2\$-1\$,-	:			
	0026	0036		002E		00025		3\$-1\$,-	:			
								4\$-1\$,-	:			
								7\$-1\$,-	:			
								6\$-1\$,-	:			
								7\$-1\$,-	:			
								5\$-1\$	:			
				50	DD	0002B	2\$:	PUSHL	NEWIO	:	1558	
			F915	CF	01	FB	0002D	CALLS	#1, WRITE	:		
						04	00032	RET		:		
					50	DD	00033	3\$:	PUSHL	NEWIO	:	1559
			F97F	CF	01	FB	00035	CALLS	#1, READ	:		
						04	0003A	RET		:		
					50	DD	0003B	4\$:	PUSHL	NEWIO	:	1560
			FA2C	CF	01	FB	0003D	CALLS	#1, READPROMPT	:		
						04	00042	RET		:		
					50	DD	00043	5\$:	PUSHL	NEWIO	:	1561
			FE91	CF	01	FB	00045	CALLS	#1, ATTACH	:		
						04	00G4A	RET		:		
					50	DD	0004B	6\$:	PUSHL	NEWIO	:	1562
			FECE	CF	01	FB	0004D	CALLS	#1, READSINGLE	:		
						04	00052	RET		:		
					50	DD	00053	7\$:	PUSHL	NEWIO	:	1563
			F7FE	CF	01	FB	00055	CALLS	#1, FREEBUF	:		
						04	0005A	8\$:	RET	:	1566	

: Routine Size: 91 bytes, Routine Base: SCODE\$ + 0955

: 1589 1567 1

```

: 1591 1568 1 ROUTINE LINKMBXMSG: NOVALUE =
: 1592 1569 1 ++
: 1593 1570 1
: 1594 1571 1 Functional Description:
: 1595 1572 1 Handle messages received on the link mailbox.
: 1596 1573 1
: 1597 1574 1 Calling Sequence:
: 1598 1575 1 standard
: 1599 1576 1
: 1600 1577 1 Input Parameters:
: 1601 1578 1 none
: 1602 1579 1
: 1603 1580 1 Implicit Inputs:
: 1604 1581 1 none
: 1605 1582 1
: 1606 1583 1 Output Parameters:
: 1607 1584 1 none
: 1608 1585 1
: 1609 1586 1 Implicit Outputs:
: 1610 1587 1 RETSTATUS
: 1611 1588 1
: 1612 1589 1 Routines Called:
: 1613 1590 1 none
: 1614 1591 1
: 1615 1592 1 Routine Value:
: 1616 1593 1 none
: 1617 1594 1
: 1618 1595 1 Signals:
: 1619 1596 1 none
: 1620 1597 1
: 1621 1598 1 Side Effects:
: 1622 1599 1 A new read on the link mailbox may be initiated.
: 1623 1600 1 A $WAKE may be issued to abort the program in case of a link error.
: 1624 1601 1
: 1625 1602 1 --
: 1626 1603 2 BEGIN
: 1627 1604 2 IF (.LINKMAIL[0] EQL MSG$_DISCON) OR (.LINKMAIL[0] EQL MSG$_ABORT) THEN
: 1628 1605 3 BEGIN ! TIME TO QUIT
: 1629 1606 3 $PUTMSG (MSGVEC = UPLIT(2,REMS$_NETDIS,0));
: 1630 1607 3 QUIT;
: 1631 1608 3 END
: 1632 1609 2
: 1633 1610 2 ELSE
: 1634 1611 3 BEGIN ! IGNORE IT
: 1635 1612 3 RETSTATUS =
: 1636 P 1613 3 $QIO (CHAN = .MAILCHAN, ! LINK MAILBOX READ
: 1637 P 1614 3 FUNC = IOS_READVBLK,
: 1638 P 1615 3 ASTADR = LINKMBXMSG,
: 1639 P 1616 3 P1 = LINKMAIL,
: 1640 1617 3 P2 = 40);
: 1641 1618 3 QUIT_ON_ERROR;
: 1642 1619 2 END;
: 1643 1620 1 END;

```

.PSECT \$SPLITS,NOWRT,NOEXE,2

```

00000002 00044 P.AAG: .LUNG 2
00000000G 00048 .ADDRESS REMS_NETDIS
00000000 0004C .LONG 0

.EXTRN SYSSPUTMSG
.PSECT $CODE$,NOWRT,2

000C 00000 LINKMBXMSG:
53 0000' CF 9E 00002 .WORD Save R2,R3 : 1568
52 00000000G 00 9E 00007 MOVAB LINKMAIL, R3
33 63 91 0000E MOVAB RETSTATUS, R2
05 13 00011 CMPB LINKMAIL, #51 : 1604
30 63 91 00013 BEQL 1$
11 12 00016 CMPB LINKMAIL, #48
7E 7C 00018 1$: BNEQ 2$
7E D4 0001A CLRQ -(SP) : 1606
00000000G 00 0000' CF 9F 0001C PUSHAB P.AAG
04 FB 00020 CALLS #4, SYSSPUTMSG
26 11 00027 BRB 3$
7E 7C 00029 2$: CLRQ -(SP) : 1617
7E 7C 0002B CLRQ -(SP)
28 DD 0002D PUSHL #40
53 DD 0002F PUSHL R3
7E D4 00031 CLRL -(SP)
CA AF 9F 00033 PUSHAB LINKMBXMSG
7E 31 7D 00036 MOVQ #49, -(SP)
7E 00000000G 00 00 3C 00039 MOVZWL MAILCHAN, -(SP)
7E D4 00040 CLRL -(SP)
00000000G 00 0C FB 00042 CALLS #12, SYSSQIO
62 50 D0 00049 MOVL R0, RETSTATUS
19 62 E8 0004C BLBS RETSTATUS, 4$
7E D4 0004F 3$: CLRL -(SP)
00000000G 00 01 FB 00051 CALLS #1, SYSSSETAST
00000000G 00 01 90 00058 MOVB #1, WAKEFLAG
7E 7C 0005F CLRQ -(SP)
00000000G 00 02 FB 00061 CALLS #2, SYSSWAKE
04 00068 4$: RET : 1620

```

; Routine Size: 105 bytes, Routine Base: \$CODE\$ + 0980

```

: 1645      1621 1 ROUTINE INDREAD =
: 1646      1622 1 ++
: 1647      1623 1
: 1648      1624 1 Functional Description:
: 1649      1625 1     Read a record from an indirect command file.
: 1650      1626 1
: 1651      1627 1
: 1652      1628 1 Calling Sequence:
: 1653      1629 1     standard
: 1654      1630 1
: 1655      1631 1 Input Parameters:
: 1656      1632 1     none
: 1657      1633 1
: 1658      1634 1 Implicit inputs
: 1659      1635 1     INDDATA
: 1660      1636 1     INDFLAG
: 1661      1637 1     SYSINRAB
: 1662      1638 1     SYSINFAB
: 1663      1639 1
: 1664      1640 1 Output Parameters:
: 1665      1641 1     none
: 1666      1642 1
: 1667      1643 1 Implicit Outputs:
: 1668      1644 1     SYSINRAB
: 1669      1645 1
: 1670      1646 1 Routines Called:
: 1671      1647 1     $GET
: 1672      1648 1     $CLOSE
: 1673      1649 1     FREEBUF
: 1674      1650 1
: 1675      1651 1 Routine Value:
: 1676      1652 1     Status of the $GET
: 1677      1653 1
: 1678      1654 1 Signals:
: 1679      1655 1     none
: 1680      1656 1
: 1681      1657 1 Side Effects:
: 1682      1658 1     If an EOF is read, the indirect command file is closed.
: 1683      1659 1
: 1684      1660 1 --
: 1685      1661 2 BEGIN
: 1686      1662 2 RETSTATUS =
: 1687      1663 2 $GET (RAB = SYSINRAB); ! READ A RECORD
: 1688      1664 2 IF .RETSTATUS EQL RMS$_EOF THEN
: 1689      1665 3 BEGIN ! END OF FILE
: 1690      1666 3 $CLOSE (FAB = SYSINFAB); ! CLOSE THE COMMAND FILE
: 1691      1667 3 FREEBUF(.INDDATA); ! GET RID OF THE BUFFER
: 1692      1668 3 INDDATA = 0; ! NO MORE DATA
: 1693      1669 3 INDFLAG = 0; ! NO MORE FILE
: 1694      1670 3 END
: 1695      1671 2 ELSE
: 1696      1672 3 BEGIN
: 1697      1673 3 IF (.RETSTATUS AND 1) EQL 0 THEN RETURN .RETSTATUS; ! ERROR
: 1698      1674 3 (.INDDATA+26+.SYSINRAB[RAB$W_RSZ])<0,8> = 'X'OD'; ! ADD TERMINATOR
: 1699      1675 3 INDDATA[RTP_IOC] = .SYSINRAB[RAB$W_RSZ]; ! RECORD SIZE
: 1700      1676 3 INDDATA[RTP_IOS] = .RETSTATUS; ! STATUS FROM THE $GET
: 1701      1677 2 END;

```

: 1702  
: 1703  
1678 2  
1679 1  
RETURN .RETSTATUS;  
END;

				.EXTRN SYSS\$GET, SYSS\$CLOSE		
		001C 00000	INDREAD:	.WORD	Save R2,R3,R4	: 1621
	54	0000'	CF 9E 00002	MOVAB	INDDATA, R4	:
	53	00000000G	00 9E 00007	MOVAB	RETSTATUS, R3	:
		00000000G	00 9F 0000E	PUSHAB	SYSINRAB	: 1663
00000000G	00		01 FB 00014	CALLS	#1, SYSS\$GET	:
	63		50 D0 0001B	MOVL	R0, RETSTATUS	:
	52		63 D0 0001E	MOVL	RETSTATUS, R2	: 1664
0001827A	8F		52 D1 00021	CAPL	R2, #98938	:
			1E 12 00028	BNEQ	1\$	:
		00000000G	00 9F 0002A	PUSHAB	SYSINFAB	: 1666
00000000G	00		01 FB 00030	CALLS	#1, SYSS\$CLOSE	:
			64 DD 00037	PUSHL	INDDATA	: 1667
F756	CF		01 FB 00039	CALLS	#1, FREEBUF	:
			64 D4 0003E	CLRL	INDDATA	: 1668
		00000000G	00 94 00040	CLRB	INDFLAG	: 1669
			1E 11 00046	BRB	3\$	: 1664
	04		52 E8 00048	1\$:	BLBS	: 1673
	50		52 D0 0004B	MOVL	R2, R0	:
			04 0004E	RET		:
	50		64 D0 0004F	2\$:	MOVL	: 1674
	51	00000000G	00 3C 00052	MOVZWL	INDDATA, R0	:
1A A140			0D 90 00059	MOVAB	SYSINRAB+34, R1	:
0A A0			51 B0 0005E	MOVW	#13, 26(R1)(R0)	: 1675
08 A0			52 B0 00062	MOVW	R1, 10(R0)	: 1676
	50		63 D0 00066	3\$:	MOVW	: 1678
			04 00069	MOVL	R2, 8(R0)	: 1679
				RET	RETSTATUS, R0	:

; Routine Size: 106 bytes, Routine Base: \$CODE\$ + 0A19

```

: 1705      1680  1 ROUTINE GETTERMCHAR(BUFFER): NOVALUE =
: 1706      1681  1 ++
: 1707      1682  1
: 1708      1683  1 Functional Description:
: 1709      1684  1     Return the terminal characteristics
: 1710      1685  1
: 1711      1686  1 Calling Sequence:
: 1712      1687  1     standard
: 1713      1688  1
: 1714      1689  1 Input Parameters:
: 1715      1690  1     BUFFER = address of buffer from link
: 1716      1691  1
: 1717      1692  1 Implicit Inputs:
: 1718      1693  1     none
: 1719      1694  1
: 1720      1695  1 Output Parameters:
: 1721      1696  1     none
: 1722      1697  1
: 1723      1698  1 Implicit Outputs:
: 1724      1699  1     none
: 1725      1700  1
: 1726      1701  1 Routines Called:
: 1727      1702  1     none
: 1728      1703  1
: 1729      1704  1 Routine Value:
: 1730      1705  1     none
: 1731      1706  1
: 1732      1707  1 Signals:
: 1733      1708  1     none
: 1734      1709  1
: 1735      1710  1 Side Effects:
: 1736      1711  1     none
: 1737      1712  1
: 1738      1713  1 --
: 1739      1714  2 BEGIN
: 1740      1715  2 LOCAL
: 1741      1716  2 CHARPTR : REF VECTOR[,BYTE],
: 1742      1717  2 CHARBUF : VECTOR[3];
: 1743      1718  2 MAP
: 1744      1719  2 BUFFER : REF RTP_BUF;
: 1745      1720  2 BIND
: 1746      1721  2 TERMTYPE = CHARBUF+1 : BYTE,
: 1747      1722  2 TERMWIDTH = CHARBUF+2 : WORD,
: 1748      1723  2 TERMCHAR = CHARBUF[1] : BLOCK[,BYTE],
: 1749      1724  2 TERMLENGTH = CHARBUF[1]+3 : BYTE,
: 1750      1725  2 TERMCHAR2 = CHARBUF[3] : BLOCK[,BYTE];
: 1751      1726  2
: 1752      1727  2 RETSTATUS =
: 1753      1728  2 SQUIW (CHAN = .CNTRLCHAN,
: 1754      1729  2       FUNC = IOS_SENSEMODE,
: 1755      1730  2       P1 = CHARBUF,
: 1756      1731  2       P2 = 12);
: 1757      1732  2 QUIT ON ERROR;
: 1758      1733  2 CHARPTR = BUFFER[RTP_DAT];      ! POINT TO THE CHARACTERISTICS LIST
: 1759      1734  2 UNTIL .CHARPTR[0] EQ 0
: 1760      1735  2 DO
: 1761      1736  2 BEGIN

```

```

CASE .CHARPTR[0] FROM 0 TO RC_MAX OF
    SET
    [RC_HHT]: CHARPTR[1] = .TERMCHAR[TT$V_MECHTAB];
    [RC_NEC]: CHARPTR[1] = .TERMCHAR[TT$V_NOECHO];
    [RC_TTP]:
        SELECT ONE .TERMTYPE OF
            SET
            [DT$VT100]: CHARPTR[1] = 13;
            [DT$VT52]: CHARPTR[1] = 9;
            [OTHERWISE]: ;
        TES;
    [RC_SCP]: CHARPTR[1] = .TERMCHAR[TT$V_SCOPE];
    [RC_BIN]: CHARPTR[1] = .TERMCHAR[TT$V_PASSALL];
    [RC_TPL]: CHARPTR[1] = .TERMLENGTH;
    [INRANGE]: ;
    [OUTRANGE]: ;
    TES;
    CHARPTR = .CHARPTR + 2;
END;
BUFFER[RTP_STS] = RS_SFC; ! GOOD STATUS
RETSTATUS =
$QIO (CHAN = .LINKCHAN, ! WRITE TO LINK
      FUNC = IOS$WRITEVBLK,
      IOSB = BUFFER[RTP_IOS],
      ASTADR = LINKWRTDONE,
      ASTPRM = .BUFFER,
      P1 = BUFFER[RTP_FNC],
      P2 = (.CHARPTR + 2 - BUFFER[RTP_FNC]));
IF .RETSTATUS EQL SSS_ABORT THEN
    RETURN; ! LINK GONE - MAILBOX MESSAGE WILL TELL WHY
QUIT_ON_ERROR;
END;

```

```

: 1762      1737      3
: 1763      1738
: 1764      1739
: 1765      1740
: 1766      1741
: 1767      1742
: 1768      1743
: 1769      1744
: 1770      1745
: 1771      1746
: 1772      1747
: 1773      1748
: 1774      1749
: 1775      1750
: 1776      1751
: 1777      1752
: 1778      1753
: 1779      1754
: 1780      1755
: 1781      1756
: 1782      1757
: 1783      1758
: 1784      1759
: 1785      1760
: 1786      1761
: 1787      1762
: 1788      1763
: 1789      1764
: 1790      1765
: 1791      1766
: 1792      1767
: 1793      1768
: 1794      1769
: 1795      1770
: 1796      1771
: 1797      1772
: 1798      1773
: 1799      1774
: 1800      1775

```

P  
P  
P  
P  
P  
P  
P

```

000C 0000 GETTERMCHAR:
           .WORD      Save R2,R3
53 0000000G 00 9E 0002      MOVAB    RETSTATUS, R3
5E               OC C2 0009      SUBL2   #12, SP
           7E 7C 0000C      CLRQ   -(SP)
           7E 7C 0000E      CLRQ   -(SP)
           14      OC DD 00010     PUSHL  #12
           AE 9F 00012     PUSHAB CHARBUF
           7E 7C 00015      CLRQ   -(SP)
7E           27 7D 00017     MOVQ   #39, -(SP)
7E 0000000G 00 3C 0001A     MOVZWL CNTRLCHAN, -(SP)
           7E D4 00021      CLRL  -(SP)
0000000G 00 0C  FB 00023     CALLS  #12, SYSSQIOW

```

```

: 1680
:
:
: 1731
:

```



			63		50	D0	0002A		MOVL	R0, RETSTATUS		
			03		63	E8	0002D		BLBS	RETSTATUS, 1\$		
					00D4	31	00030		BRW	15\$		
			51		04	AC	D0 00033	1\$:	MOVL	BUFFER, R1		1733
			50		1A	A1	9E 00037		MOVAB	26(R1), CHARPTR		
						60	95 0003B	2\$:	TSTB	(CHARPTR)		1734
						03	12 0003D		BNEQ	3\$		
					008B	31	0003F		BRW	14\$		
					60	BF	00042	3\$:	CASEB	(CHARPTR), #0, #28		1737
0081	1C		00		0081		00046	4\$:	.WORD	13\$-4\$,-		
0081	0081	0081			0081		0004E			13\$-4\$,-		
0081	0081	0081			0081		00056			13\$-4\$,-		
0081	0081	0081			0081		0005E			13\$-4\$,-		
0044	003C	0081			0081		00066			13\$-4\$,-		
0068	004C	0081			0081		0006E			13\$-4\$,-		
0081	0081	0081			007G		00076			13\$-4\$,-		
					007C		0007E			13\$-4\$,-		
										13\$-4\$,-		
										5\$-4\$,-		
										6\$-4\$,-		
										13\$-4\$,-		
										13\$-4\$,-		
										7\$-4\$,-		
										9\$-4\$,-		
										10\$-4\$,-		
										13\$-4\$,-		
										13\$-4\$,-		
										13\$-4\$,-		
										12\$-4\$		
										13\$		
52	05	AE		01	45	11	00080		BRB	13\$		
					00	EF	00082	5\$:	EXTZV	#0, #1, TERMCHAR+1, R2		1740
					32	11	00088		BRB	11\$		
52	04	AE		01	01	EF	0008A	6\$:	EXTZV	#1, #1, TERMCHAR, R2		1742
					2A	11	00090		BRB	11\$		
					60	52	01	01	01	01	01	
					8F	AE	9A 00092	7\$:	MOVZBL	TERMTYPE, R2		1744
						52	91 00096		CMPB	R2, #96		1746
						06	12 0009A		BNEQ	8\$		
					01	A0	0D 90 0009C		MOVB	#13, 1(CHARPTR)		1747
						25	11 000A0		BRB	13\$		
					40	8F	52 91 000A2	8\$:	CMPB	R2, #64		1748
						01	1F 12 000A6		BNEQ	13\$		
						01	A0		MOVB	#9, 1(CHARPTR)		1749
						19	11 000AC		BRB	13\$		
52	05	AE		01	04	EF	000AE	9\$:	EXTZV	#4, #1, TERMCHAR+1, R2		1753
						06	11 000B4		BRB	11\$		
52	04	AE		01	00	EF	000B6	10\$:	EXTZV	#0, #1, TERMCHAR, R2		1755
					01	A0	52 90 000BC	11\$:	MOVB	R2, 1(CHARPTR)		
						05	11 000C0		BRB	13\$		

01	A0	07	AE	90	000C2	12\$:	MOVB	TERMLENGTH, 1(CHARPTR)	:	1757
	50		02	C0	000C7	13\$:	ADDL2	#2, CHARPTR	:	1761
			FF6E	31	000CA		BRW	2\$	:	1734
		13	A1	94	000CD	14\$:	CLRB	19(R1)	:	1763
			7E	7C	000D0		CLRQ	-(SP)	:	1771
			7E	7C	000D2		CLRQ	-(SP)	:	
	52	10	A1	9E	000D4		MOVAB	16(R1), R2	:	
	50		52	C2	000D8		SUBL2	R2, R0	:	
		02	A0	9F	000DB		PUSHAB	2(R0)	:	
		10	A1	9F	000DE		PUSHAB	16(R1)	:	
			51	DD	000E1		PUSHL	R1	:	
		FA03	CF	9F	000E3		PUSHAB	LINKWRTDONE	:	
		08	A1	9F	000E7		PUSHAB	8(R1)	:	
			30	DD	000EA		PUSHL	#48	:	
	7E	00000000G	00	3C	000EC		MOVZWL	LINKCHAN, -(SP)	:	
			7E	D4	000F3		CLRL	-(SP)	:	
00000000G	00		0C	FB	000F5		CALLS	#12, SYSSQIO	:	
	63		50	D0	000FC		MOVL	R0, RETSTATUS	:	
	2C		50	D1	000FF		C MPL	R0, #44	:	1772
			1C	13	00102		BEQL	16\$	:	
	19		50	E8	00104		BLBS	R0, 16\$	:	1773
			7E	D4	00107	15\$:	CLRL	-(SP)	:	
00000000G	00		01	FB	00109		CALLS	#1, SYSSSETAST	:	
00000000G	00		01	90	00110		MOVB	#1, WAKEFLAG	:	
			7E	7C	00117		CLRQ	-(SP)	:	
00000000G	00		02	FB	00119		CALLS	#2, SYSSWAKE	:	
			04	00120	16\$:		RET		:	1775

: Routine Size: 289 bytes, Routine Base: \$CODE\$ + 0A83

: 1801	1776	1	
: 1802	1777	1	
: 1803	1778	1	END
: 1804	1779	0	ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	164	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$PLITS	80	NOVEC, NOWRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
PROTOTBL	6	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(0)
\$CODE\$	2980	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
------	----------------	-------------------	------------------	-----------------	--------------------

RSXRT  
V04-000

E 7  
16-Sep-1984 02:18:51  
14-Sep-1984 13:04:57

VAX-11 Bliss-32 V4.0-742 Page 65  
DISK\$VMSMASTER:[RTPAD.SRC]RSXRT.B32;1 (26)

```
:  
:  $255$DUA28:[SYSLIB]LIB.L32;1      18619      45      0      1000      00:01.4  
:  $255$DUA28:[SYSLIB]CLIMAC.L32;1    14          2      14          9      00:00.0
```

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:RSXRT/OBJ=OBJ\$:RSXRT MSRCS\$:RSXRT/UPDATE=(ENH\$:RSXRT)

```
: Size:          2980 code + 250 data bytes  
: Run Time:      00:37.4  
: Elapsed Time: 02:35.6  
: Lines/CPU Min: 2850  
: Lexemes/CPU-Min: 36583  
: Memory Used:  222 pages  
: Compilation Complete
```

