

PPPPPPPPPPPP		AAAAAAA		SSSSSSSSSSSS		RRRRRRRRRRRR		TTTTTTTTTTTTTTT		LLL
PPPPPPPPPPPP		AAAAAAA		SSSSSSSSSSSS		RRRRRRRRRRRR		TTTTTTTTTTTTTTT		LLL
PPPPPPPPPPPP		AAAAAAA		SSSSSSSSSSSS		RRRRRRRRRRRR		TTTTTTTTTTTTTTT		LLL
PPP	PPP	AAA	AAA	SSS		RRR	RRR	TTT		LLL
PPP	PPP	AAA	AAA	SSS		RRR	RRR	TTT		LLL
PPP	PPP	AAA	AAA	SSS		RRR	RRR	TTT		LLL
PPP	PPP	AAA	AAA	SSS		RRR	RRR	TTT		LLL
PPP	PPP	AAA	AAA	SSS		RRR	RRR	TTT		LLL
PPP	PPP	AAA	AAA	SSS		RRR	RRR	TTT		LLL
PPPPPPPPPPPP		AAA	AAA		SSSSSSSSSS	RRRRRRRRRRRR		TTT		LLL
PPPPPPPPPPPP		AAA	AAA		SSSSSSSSSS	RRRRRRRRRRRR		TTT		LLL
PPPPPPPPPPPP		AAA	AAA		SSSSSSSSSS	RRRRRRRRRRRR		TTT		LLL
PPP		AAAAAAAAAAAAAAAA			SSS	RRR	RRR	TTT		LLL
PPP		AAAAAAAAAAAAAAAA			SSS	RRR	RRR	TTT		LLL
PPP		AAAAAAAAAAAAAAAA			SSS	RRR	RRR	TTT		LLL
PPP		AAA	AAA		SSS	RRR	RRR	TTT		LLL
PPP		AAA	AAA		SSS	RRR	RRR	TTT		LLL
PPP		AAA	AAA		SSS	RRR	RRR	TTT		LLL
PPP		AAA	AAA		SSS	RRR	RRR	TTT		LLL
PPP		AAA	AAA	SSSSSSSSSSSS		RRR	RRR	TTT	LLLLLLLLLLLLLLLL	
PPP		AAA	AAA	SSSSSSSSSSSS		RRR	RRR	TTT	LLLLLLLLLLLLLLLL	
PPP		AAA	AAA	SSSSSSSSSSSS		RRR	RRR	TTT	LLLLLLLLLLLLLLLL	

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

PAS

```

PPPPPPPP          AAAAAA          SSSSSSSS          RRRRRRRR          EEEEEEEEEEE          AAAAAA          DDDDDDDD          LL          NN          NN
PPPPPPPP          AAAAAA          SSSSSSSS          RRRRRRRR          EEEEEEEEEEE          AAAAAA          DDDDDDDD          LL          NN          NN
PP      PP      AA      AA      SS          RR      RR          EE          AA      AA      DD      DD          LL          NN          NN
PP      PP      AA      AA      SS          RR      RR          EE          AA      AA      DD      DD          LL          NN          NN
PP      PP      AA      AA      SS          RR      RR          EE          AA      AA      DD      DD          LL          NN          NN
PPPPPPPP          AA      AA          SSSSSS          RRRRRRRR          EEEEEEEEE          AA      AA          DD      DD          LL          NN          NN
PPPPPPPP          AA      AA          SSSSSS          RRRRRRRR          EEEEEEEEE          AA      AA          DD      DD          LL          NN          NN
PP          AAAAAAAAAA          SS          RR      RR          EE          AAAAAAAAAA          DD      DD          LL          NN          NNNN
PP          AAAAAAAAAA          SS          RR      RR          EE          AAAAAAAAAA          DD      DD          LL          NN          NNNN
PP          AA      AA          SS          RR      RR          EE          AA      AA          DD      DD          LL          NN          NN
PP          AA      AA          SS          RR      RR          EE          AA      AA          DD      DD          LL          NN          NN
PP          AA      AA          SSSSSSSS          RR      RR          EEEEEEEEEEE          AA      AA          DDDDDDDD          LLLLLLLLLL          NN          NN
PP          AA      AA          SSSSSSSS          RR      RR          EEEEEEEEEEE          AA      AA          DDDDDDDD          LLLLLLLLLL          NN          NN

```

```

LL          IIIIII          SSSSSSSS
LL          IIIIII          SSSSSSSS
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
LLLLLLLLLLL          IIIIII          SSSSSSSS
LLLLLLLLLLL          IIIIII          SSSSSSSS

```

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

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

```

1 0001 0 MODULE PASSREADLN2 ( %TITLE 'READLN procedure'
2 0002 0 IDENT = '1-003' ! File: PASWRITEL.B32 Edit: SBL1003
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
10 0010 1 * ALL RIGHTS RESERVED. *
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
17 0017 1 * TRANSFERRED. *
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
21 0021 1 * CORPORATION. *
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1 ++
31 0031 1 FACILITY: Pascal Language Support
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 This module contains PASSREADLN2, which implements the
36 0036 1 VAX-11 Pascal READLN procedure. It also contains utility
37 0037 1 procedures used by textfile and string READ procedures.
38 0038 1
39 0039 1 ENVIRONMENT: User mode -- AST reentrant
40 0040 1
41 0041 1 AUTHOR: Steven B. Lionel, CREATION DATE: 1-April-1981
42 0042 1
43 0043 1 MODIFIED BY:
44 0044 1
45 0045 1 1-001 - Original. SBL 1-April-1981
46 0046 1 1-002 - Add PASS$END_READ. SBL 24-May-1982
47 0047 1 1-003 - Use a zeroed descriptor pointer to indicate that the previous
48 0048 1 READV on this string ended with FCBSV LAZY set. This allows the
49 0049 1 proper error actions to occur. SBL 10-Jan-1983
50 0050 1 --
51 0051 1

```

```

53 0052 1 %SBTTL 'Declarations'
54 0053 1
55 0054 1 PROLOGUE DEFINITIONS:
56 0055 1
57 0056 1
58 0057 1 REQUIRE 'RTLIN:PASPROLOG'; ! Externals, linkages, PSECTs, structures
59 0121 1
60 0122 1
61 0123 1 TABLE OF CONTENTS:
62 0124 1
63 0125 1
64 0126 1 FORWARD ROUTINE
65 0127 1 PASSREADLN2: NOVALUE, ! Do a READLN
66 0128 1 PASS$INIT READ: JSB INIT READ NOVALUE, ! Initialize for read
67 0129 1 PASS$END READ: JSB END READ NOVALUE, ! End a read
68 0130 1 PASS$DO_READV: JSB_DO_READV NOVALUE; ! Do a READV
69 0131 1
70 0132 1
71 0133 1 MACROS:
72 0134 1
73 0135 1 NONE
74 0136 1
75 0137 1 EQUATED SYMBOLS:
76 0138 1
77 0139 1 NONE
78 0140 1
79 0141 1 FIELDS:
80 0142 1
81 0143 1 NONE
82 0144 1
83 0145 1 OWN STORAGE:
84 0146 1
85 0147 1
86 0148 1
87 0149 1 + Declare a Pascal File Descriptor to use for a "string file" READV
88 0150 1 -
89 0151 1
90 0152 1 OWN
91 0153 1 PFD_READV: BLOCK [PFD$K SIZE+%CHARCOUNT('-string-'), BYTE]
92 0154 1 FIELD (PFD$FIELDS) PSECT (_PASS$CODE)
93 0155 1 PRESET (
94 0156 1 [PFD$V_TEXT] = 1,
95 0157 1 [PFD$V_NOREAD] = 1,
96 0158 1 [PFD$V_NOWRITE] = 1,
97 0159 1 [PFD$L_LENGTH] = 1,
98 0160 1 [PFD$T_NAME] = %CHARCOUNT('-string-'),
99 0161 1 [PFD$B_NAME1] = %C'-',
100 0162 1 [PFD$B_NAME2] = %C's',
101 0163 1 [PFD$B_NAME3] = %C't',
102 0164 1 [PFD$B_NAME4] = %C'r',
103 0165 1 [PFD$B_NAME5] = %C'i',
104 0166 1 [PFD$B_NAME6] = %C'n',
105 0167 1 [PFD$B_NAME7] = %C'g',
106 0168 1 [PFD$B_NAME8] = %C'-',
107 0169 1 );

```

```

109 0170 1 %SBTTL 'PASSREADLN2 - READLN procedure'
110 0171 1 GLOBAL ROUTINE PASSREADLN2 (
111 0172 1 PFV: REF $PASSPFV_FILE_VARIABLE,
112 0173 1 ERROR
113 0174 1 ): NOVALUE =
114 0175 1
115 0176 1
116 0177 1 ++
117 0178 1 FUNCTIONAL DESCRIPTION:
118 0179 1 PASSREADLN2 implements the VAX-11 Pascal READLN procedure. It
119 0180 1 flushes the remainder of the current record and positions the
120 0181 1 file at the beginning of the next record.
121 0182 1
122 0183 1 CALLING SEQUENCE:
123 0184 1
124 0185 1 CALL PASSREADLN2 (PFV.mr.r [, ERROR.j.r])
125 0186 1
126 0187 1 FORMAL PARAMETERS:
127 0188 1
128 0189 1 PFV - The Pascal File Variable (PFV) passed by reference.
129 0190 1 The structure of the PFV is defined in PASPFV.REQ.
130 0191 1
131 0192 1 ERROR - Optional. If specified, the address to unwind to
132 0193 1 in case of an error.
133 0194 1
134 0195 1 IMPLICIT INPUTS:
135 0196 1
136 0197 1 NONE
137 0198 1
138 0199 1 IMPLICIT OUTPUTS:
139 0200 1
140 0201 1 NONE
141 0202 1
142 0203 1 ROUTINE VALUE:
143 0204 1
144 0205 1 NONE
145 0206 1
146 0207 1 SIDE EFFECTS:
147 0208 1
148 0209 1 If the file is the standard file INPUT or OUTPUT, it is implicitly opened.
149 0210 1
150 0211 1 SIGNALLED ERRORS:
151 0212 1
152 0213 1
153 0214 1 --
154 0215 1
155 0216 2 BEGIN
156 0217 2
157 0218 2 LOCAL
158 0219 2 FCB: REF $PASSFCB_CONTROL_BLOCK,
159 0220 2 PFV_ADDR: VOLATILE,
160 0221 2 UNWIND_ACT: VOLATILE,
161 0222 2 ERROR_ADDR: VOLATILE;
162 0223 2
163 0224 2 BUILTIN
164 0225 2 ACTUALCOUNT;
165 0226 2

```

! Do a READLN
! File variable
! Error unwind address

! File control block
! Enable argument
! Enable argument
! Enable argument

```

: 166 0227 2  ENABLE
: 167 0228 2  PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR); ! Enable error handler
: 168 0229 2
: 169 0230 2  PFV_ADDR = PFV [PFV$R_PFV]; ! Set PFV address
: 170 0231 2
: 171 0232 2  IF ACTUALCOUNT () GEQU 2
: 172 0233 2  THEN
: 173 0234 2  ERROR_ADDR = .ERROR; ! Set unwind address
: 174 0235 2
: 175 0236 2  !+
: 176 0237 2  ! Validate PFV and get PFV.
: 177 0238 2  !-
: 178 0239 2
: 179 0240 2  PASS$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
: 180 0241 2
: 181 0242 2  !+
: 182 0243 2  ! Set unwind action to unlock file.
: 183 0244 2  !-
: 184 0245 2
: 185 0246 2  UNWIND_ACT = PASS$UNWIND_UNLOCK;
: 186 0247 2
: 187 0248 2  !+
: 188 0249 2  ! Do common initialization.
: 189 0250 2  !-
: 190 0251 2
: 191 0252 2  PASS$INIT_READ (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
: 192 0253 2
: 193 0254 2  !+
: 194 0255 2  ! Move to the end of the line and do a GET.
: 195 0256 2  !-
: 196 0257 2
: 197 0258 2  PFV [PFV$V_EOLN] = 1;
: 198 0259 2  PASS$GET (PFV [PFV$R_PFV], FCB [FCB$R_FCB]);
: 199 0260 2
: 200 0261 2  !+
: 201 0262 2  ! Unlock the file variable.
: 202 0263 2  !-
: 203 0264 2
: 204 0265 2  PFV [PFV$V_LOCK] = 0;
: 205 0266 2
: 206 0267 2  RETURN;
: 207 0268 2
: 208 0269 1  END; ! End of routine PASS$READLN2

```

```

.TITLE PASS$READLN2 READLN procedure
.IDENT \1-003\
.PSECT _PASS$CODE,NOWRT, SHR, PIC,2

```

```

00# 0000 PFD_READV:
19 00004 .BYTE 0[4]
00# 00005 .BYTE 25
0000001 00008 .BYTE 0[3]
2D 67 6E 69 72 74 73 2D 08 0000C .LONG 1
.BYTE 8, 45, 115, 116, 114, 105, 110, 103, 45

```

			00FC 00000		.EXTRN	PASSREADLN2, PASS\$INIT_READ	
			08 C2 00002		.EXTRN	PASS\$END_READ, PASS\$DO_READV	
			7E D4 00005		.EXTRN	PASS\$IO_HANDLER	
			AE 7C 00007		.EXTRN	PASS\$VALIDATE_PFV	
			CF DE 0000A		.EXTRN	PASS\$GET	
		04	AC D0 0000F				
		002F	56 D0 00013		.ENTRY	PASSREADLN2, Save R2,R3,R4,R5,R6,R7	0171
08		04	6C 91 00017		SUBL2	#8, SP	
			04 1F 0001A		CLRL	ERROR_ADDR	0216
			AC D0 0001C		CLRQ	UNWIND_ACT	
		08	00 16 00020	1\$:	MOVAL	2\$, (FP)	
			01 D0 00026		MOVL	PFV, R6	0230
			30 0002A		MOVL	R6, PFV_ADDR	
04		00000000G	08 88 0002D		CMPB	(AP), #2	0232
			00 16 00031		BLSSU	1\$	
			8F 8A 00037		MOVL	ERROR, ERROR_ADDR	0234
			04 0003C		JSB	PASS\$VALIDATE_PFV	0240
			0000 0003D	2\$:	MOVL	#1, UNWIND_ACT	0246
			AC D0 0003F		BSBW	PASS\$INIT_READ	0252
			A0 D0 00043		BISB2	#8, 6(R6)	0258
			A0 9F 00047		JSB	PASS\$GET	0259
			A0 9F 0004A		BICB2	#128, 7(R6)	0265
			A0 9F 0004D		RET		0269
			03 DD 00050		.WORD	Save nothing	0216
			5E DD 00052		MOVL	8(AP), R0	
			AC 7D 00054		MOVL	4(R0), R0	
			03 FB 00058		PUSHAB	ERROR_ADDR	
			04 0005F		PUSHAB	UNWIND_ACT	
					PUSHAB	PFV_ADDR	
					PUSHL	#3	
					PUSHL	SP	
					MOVQ	4(AP), -(SP)	
					CALLS	#3, PASS\$IO_HANDLER	
					RET		

: Routine Size: 96 bytes, Routine Base: _PASS\$CODE + 0015

: 209 0270 1
: 210 0271 1 !<BLF/PAGE>

```

212 0272 1 %SBTTL 'PASS$INIT_READ - Common setup for READ procedures'
213 0273 1 GLOBAL ROUTINE PASS$INIT_READ (                               ! Common setup
214 0274 1     PFV: REF $PASS$PFV FILE_VARIABLE,                          ! File variable
215 0275 1     FCB_IN: REF $PASS$FCB CONTROL_BLOCK;                       ! Input FCB
216 0276 1     FCB: REF $PASS$FCB CONTROL_BLOCK                           ! Output FCB
217 0277 1 ): JSB_INIT_READ NOVALUE=
218 0278 1
219 0279 1 ++
220 0280 1 FUNCTIONAL DESCRIPTION:
221 0281 1
222 0282 1     This procedure performs the common setup steps for all READ procedures.
223 0283 1     These are:
224 0284 1         Verify that the file is a textfile.
225 0285 1         If it is the standard file OUTPUT or INPUT, open it.
226 0286 1         Verify that the file is in Inspection mode.
227 0287 1         Resolve any lazy lookahead in progress.
228 0288 1         Move character in user file buffer to record buffer.
229 0289 1
230 0290 1 CALLING SEQUENCE:
231 0291 1
232 0292 1     JSB_INIT_READ PASS$INIT_READ (PFV.mr.r, FCB_IN.mr.r; FCB)
233 0293 1
234 0294 1 FORMAL PARAMETERS:
235 0295 1
236 0296 1     PFV           - The Pascal File Variable (PFV) passed by reference in R2.
237 0297 1                 The structure of the PFV is defined in PASPFV.REQ.
238 0298 1
239 0299 1     FCB_IN        - The address of the FCB for the file, if any.  If
240 0300 1                 the file is not open, it will be opened and the
241 0301 1                 resultant address stored in FCB.
242 0302 1
243 0303 1     FCB           - The address of the File Control Block (FCB) is an
244 0304 1                 output register parameter.
245 0305 1
246 0306 1 IMPLICIT INPUTS:
247 0307 1
248 0308 1     NONE
249 0309 1
250 0310 1 IMPLICIT OUTPUTS:
251 0311 1
252 0312 1     NONE
253 0313 1
254 0314 1 ROUTINE VALUE:
255 0315 1
256 0316 1     NONE
257 0317 1
258 0318 1 SIDE EFFECTS:
259 0319 1
260 0320 1     If the file is the standard file OUTPUT, it is implicitly opened.
261 0321 1
262 0322 1 SIGNALLED ERRORS:
263 0323 1
264 0324 1     FILNOTOPE - file not open
265 0325 1     FILNOTTEX - file is not a textfile
266 0326 1     FILNOTINS - file not in Inspection mode
267 0327 1     GETAFTEOF - GET attempted after end-of-file
268 0328 1

```



```

269 0329 1 !--
270 0330 1
271 0331 2 BEGIN
272 0332 2
273 0333 2
274 0334 2
275 0335 2
276 0336 2
277 0337 2 IF NOT .PFV [PFV$V_VALID]
278 0338 2 THEN
279 0339 2     PASS$LOOK_AHEAD (PFV [PFV$R_PFV], FCB_IN [FCB$R_F(B); FCB]);
280 0340 2
281 0341 2
282 0342 2
283 0343 2
284 0344 2
285 0345 2 IF NOT .PFV [PFV$V_OPEN]           ! Not open
286 0346 2 THEN
287 0347 2     $PASS$IO_ERROR (PASS$_FILNOTOPE,0);           ! File not open
288 0348 2
289 0349 2
290 0350 2
291 0351 2
292 0352 2
293 0353 2
294 0354 2 IF .FCB [FCB$V_STRING]
295 0355 2 THEN
296 0356 2     RETURN;
297 0357 2
298 0358 2
299 0359 2
300 0360 2
301 0361 2
302 0362 2 IF NOT .FCB [FCB$V_TEXT]
303 0363 2 THEN
304 0364 2     $PASS$IO_ERROR (PASS$_FILNOTTEX,0);           ! File is not a textfile
305 0365 2
306 0366 2
307 0367 2
308 0368 2
309 0369 2
310 0370 2 IF NOT .FCB [FCB$V_INSPECTION]
311 0371 2 THEN
312 0372 2     $PASS$IO_ERROR (PASS$_FILNOTINS,0);
313 0373 2
314 0374 2
315 0375 2
316 0376 2
317 0377 2
318 0378 2 IF .FCB [FCB$V_EOF]
319 0379 2 THEN
320 0380 2     $PASS$IO_ERROR (PASS$_GETAFTEOF,0);
321 0381 2
322 0382 2
323 0383 2
324 0384 2
325 0385 2

```

```

: 326      0386  2
: 327      0387  2      IF NOT .PFV [PFV$V_EOLN] AND NOT .FCB [FCB$V_NOWRITE]
: 328      0388  2      THEN
: 329      0389  2          CH$WCHAR (..PFV [PFV$A_BUFFER], .FCB [FCB$A_RECORD_CUR]);
: 330      0390  2
: 331      0391  2      RETURN;          : Success
: 332      0392  2
: 333      0393  1      END;

```

! End of routine PASS\$INIT_READ

```

.EXTRN PASS$LOOK_AHEAD
.EXTRN PASS$SIGNAL, PASS$FILNOTOPE
.EXTRN PASS$FILNOTTEX
.EXTRN PASS$FILNOTINS
.EXTRN PASS$GETAFTEOF

```

		06	06	A6	E8	00000	PASS\$INIT_READ::				
							B[BS	6(PFV), 1\$: 0337		
			00000000G	00	16	00004	JSB	PASS\$LOOK_AHEAD	: 0339		
08	07	A6		05	E0	0000A	1\$:	BBS	#5, 7(PFV), 2\$: 0345	
				7E	D4	0000F		CLRL	-(SP)	: 0347	
				7E	00G	8F	9A	00011	MOVZBL	#PASS\$FILNOTOPE, -(SP)	
				29	11	00015		BRB	5\$		
3B	FE	A7		04	E0	00017	2\$:	BBS	#4, -2(FCB), 7\$: 0354	
				08	F8	0001C		BLBS	-8(FCB), 3\$: 0362	
				7E	D4	00020		CLRL	-(SP)	: 0364	
				7E	00G	8F	9A	00022	MOVZBL	#PASS\$FILNOTTEX, -(SP)	
				18	11	00026		BRB	5\$		
08	FD	A7		07	E0	00028	3\$:	BBS	#3, -3(FCB), 4\$: 0370	
				7E	D4	0002D		CLRL	-(SP)	: 0372	
				7E	00G	8F	9A	0002F	MOVZBL	#PASS\$FILNOTINS, -(SP)	
				0B	11	00033		BRB	5\$		
0E	FD	A7		05	E1	00035	4\$:	BBC	#5, -3(FCB), 6\$: 0378	
				7E	D4	0003A		CLRL	-(SP)	: 0380	
				7E	00G	8F	9A	0003C	MOVZBL	#PASS\$GETAFTEOF, -(SP)	
		00000000G	00	02	FB	00040	5\$:	CALLS	#2, PASS\$SIGNAL		
				05	00047			RSB			
0A	06	A6		03	E0	00048	6\$:	BBS	#3, 6(PFV), 7\$: 0387	
05	F8	A7		04	E0	0004D		BBS	#4, -8(FCB), 7\$		
	EC	B7		00	B6	90	00052	MOVB	@0(PFV), @-20(FCB)	: 0389	
				05	00057	7\$:		RSB		: 0393	

; Routine Size: 88 bytes, Routine Base: _PASS\$CODE + 0075

```

: 334      0394  1
: 335      0395  1 !<BLF/PAGE>

```

337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393

```

0396 1 %SBTTL 'PASS$END_READ - Common epilogue for READ procedures'
0397 1 GLOBAL ROUTINE PASS$END_READ (
0398 1     PFV: REF $PASS$PFV_FILE_VARIABLE,           ! File variable
0399 1     FCB: REF $PASS$FCB_CONTROL_BLOCK           ! File control block
0400 1 ): JSB_END_READ NOVALOE =
0401 1
0402 1 !++
0403 1 FUNCTIONAL DESCRIPTION:
0404 1
0405 1     This procedure is called at the end of every READ procedure
0406 1     to perform common functions. These are:
0407 1     1. Move next character to user file buffer (do a GET).
0408 1     2. Set STATUS to zero and unlock file variable.
0409 1
0410 1 CALLING SEQUENCE:
0411 1
0412 1     JSB_END_READ PASS$END_READ (PFV.mr.r, FCB.mr.r)
0413 1
0414 1 FORMAL PARAMETERS:
0415 1
0416 1     PFV           - The Pascal File Variable (PFV) passed by reference in R2.
0417 1                   The structure of the PFV is defined in PASPFV.REQ.
0418 1
0419 1     FCB           - The File Control Block of the file.
0420 1
0421 1 IMPLICIT INPUTS:
0422 1
0423 1     It is assumed that the file is a textfile, is locked and is
0424 1     in Generation mode.
0425 1
0426 1 IMPLICIT OUTPUTS:
0427 1
0428 1     NONE
0429 1
0430 1 ROUTINE VALUE:
0431 1
0432 1     NONE
0433 1
0434 1 SIDE EFFECTS:
0435 1
0436 1     NONE
0437 1
0438 1 SIGNALLED ERRORS:
0439 1
0440 1     NONE
0441 1 --
0442 1
0443 2 BEGIN
0444 2
0445 2 !+
0446 2 | If file buffer is not NOREAD, store the next character to be read
0447 2 | into the user's file buffer.
0448 2 | -
0449 2
0450 2 IF .FCB [FCB$A_RECORD_CUR] GEQA .FCB [FCB$A_RECORD_END]
0451 2 THEN
0452 3     BEGIN

```

00
00
00
06
06
06
06


```

418 0476 1 %SBTTL 'PASS$DO READV - Do a READV'
419 0477 1 GLOBAL ROUTINE PASS$DO READV (
420 0478 1     PFV: REF $PASS$PFV_FILE_VARIABLE,           ! PFV of string file
421 0479 1     STRING: REF BLOCK-[ , BYTE],           ! String to read from
422 0480 1     ARGUMENT_LIST: REF VECTOR [ , LONG],   ! Argument list
423 0481 1     ROUTINE_ADDR;                          ! Routine to call
424 0482 1     RESULT_R0,                             ! R0 of result
425 0483 1     RESULT_R1                             ! R1 of result
426 0484 1     ): JSB_DO_READV NOVALUE =
427 0485 1
428 0486 1  +-+
429 0487 1  FUNCTIONAL DESCRIPTION:
430 0488 1
431 0489 1  CALLING SEQUENCE:
432 0490 1
433 0491 1      JSB_DO_READV PASS$DO READV (PFV.mr.r, STRING.mq.r,
434 0492 1      ARGUMENT_LIST.rlu.r, ROUTINE_ADDR.fzem.r;
435 0493 1      RESULT_R0, RESULT_R1)
436 0494 1
437 0495 1  FORMAL PARAMETERS:
438 0496 1
439 0497 1      PFV           - Pascal File Variable allocated by our caller.
440 0498 1      It is filled in here.
441 0499 1
442 0500 1      STRING        - A descriptor of the string to read from.
443 0501 1
444 0502 1      ARGUMENT_LIST   - Argument list for a CALLG
445 0503 1
446 0504 1      ROUTINE_ADDR    - The address of the PASS$READ_xxx routine to call
447 0505 1      to actually do the work.
448 0506 1
449 0507 1      RESULT_R0      - The R0 of the function result, if any.
450 0508 1
451 0509 1      RESULT_R1      - The R1 of the function result, if any.
452 0510 1
453 0511 1  IMPLICIT INPUTS:
454 0512 1
455 0513 1      Caller's AP (for the address of the string descriptor.
456 0514 1
457 0515 1  IMPLICIT OUTPUTS:
458 0516 1
459 0517 1      NONE
460 0518 1
461 0519 1  ROUTINE VALUE:
462 0520 1
463 0521 1      NONE
464 0522 1
465 0523 1  SIDE EFFECTS:
466 0524 1
467 0525 1      See called routine
468 0526 1
469 0527 1  SIGNALLED ERRORS:
470 0528 1
471 0529 1      NONE
472 0530 1
473 0531 1  --
474 0532 1

```

```

475 0533 2 BEGIN
476 0534 2
477 0535 2 LOCAL
478 0536 2 FCB_BLOCK: $PASSFCB_CONTROL_BLOCK; ! File Control Block
479 0537 2
480 0538 2 BIND
481 0539 2 FCB = FCB_BLOCK + FCB$K_BLN: $PASSFCB_CONTROL_BLOCK;
482 0540 2
483 0541 2 BUILTIN
484 0542 2 CALLG;
485 0543 2
486 0544 2 !+
487 0545 2 ! Set up the Pascal File Variable
488 0546 2 !-
489 0547 2
490 0548 2 PFV [PFV$A_BUFFER] = 0; ! No file buffer
491 0549 2 PFV [PFV$B_VERSION] = PFV$K_CUR_VERSION; ! PFV version number
492 0550 2 PFV [PFV$W_FLAGS] = 0; ! Initially zero bits
493 0551 2 PFV [PFV$V_VALID] = 1; ! PFV is valid
494 0552 2 PFV [PFV$V_FCB_VALID] = 1; ! FCB address is valid
495 0553 2 PFV [PFV$V_OPEN] = 1; ! File is "open"
496 0554 2 PFV [PFV$V_EOF_DEFINED] = 1; ! EOF(f) is defined
497 0555 2 PFV [PFV$V_DFB] = 1; ! File buffer is defined
498 0556 2 PFV [PFV$A_PFD] = PFD_READV [PFD$R_PFD]; ! Address of PFD
499 0557 2 PFV [PFV$A_FCB] = FCB [FCB$R_FCB]; ! Address of FCB
500 0558 2
501 0559 2 !+
502 0560 2 ! Set up the File Control Block (FCB). Only set those fields which
503 0561 2 ! will be used.
504 0562 2 !-
505 0563 2
506 0564 2 FCB [FCB$A_PFV] = PFV [PFV$R_PFV]; ! Address of PFV
507 0565 2 FCB [FCB$A_PFD] = PFD_READV [PFD$R_PFD]; ! Address of name
508 0566 2 FCB [FCB$W_ATTRIB] = PFD_READV [PFD$W_ATTRIB]; ! Attributes
509 0567 2 FCB [FCB$L_OPTIONS1] = 0; ! Initially 0
510 0568 2 FCB [FCB$V_INSPECTION] = 1; ! In Inspection mode
511 0569 2 FCB [FCB$V_STRING] = 1; ! String file
512 0570 2 FCB [FCB$L_RECORD_NUMBER] = 1; ! "Record 1"
513 0571 2
514 0572 2 !+
515 0573 2 ! Get the "line" pointer and length, and set the FCB accordingly.
516 0574 2 ! If the last READV on this string ended with FCB$V_LAZY set, we
517 0575 2 ! zeroed the descriptor pointer. Look for that here.
518 0576 2 !-
519 0577 2
520 0578 2 FCB [FCB$A_RECORD_BEG] = .STRING [DSC$A_POINTER];
521 0579 2 FCB [FCB$A_RECORD_CUR] = .FCB [FCB$A_RECORD_BEG];
522 0580 2 IF .FCB [FCB$A_RECORD_CUR] EQLA 0
523 0581 2 THEN
524 0582 2 BEGIN
525 0583 2 FCB [FCB$V_LAZY] = 1;
526 0584 2 PFV [PFV$V_VALID] = 0;
527 0585 2 END;
528 0586 2 FCB [FCB$L_RECORD_LEN] = .STRING [DSC$W_LENGTH];
529 0587 2 FCB [FCB$A_RECORD_END] = .FCB [FCB$A_RECORD_BEG] + .FCB [FCB$L_RECORD_LEN];
530 0588 2
531 0589 2 !+

```


PASSREADLN2
1-003

READLN procedure
PASS\$DO_READV - Do a READV

J 4
16-Sep-1984 01:54:22
14-Sep-1984 12:51:46

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASREADLN.B32;1

Page 14
(6)

5E 04 A2 D4 0006F CLRL 4 (STRING)
 44 AE 9E 00072 MOVAB 68(SP), SP
 05 00076 RSB

: 0611
: 0615
:

: Routine Size: 119 bytes, Routine Base: _PASSCODE + 00F6

: 558 0616 1
: 559 0617 1 !<BLF/PAGE>

PAS
1-0

: R

:
:

PASS\$READLN2
1-003

READLN procedure
PASS\$DO_READV - Do a READV

K 4
16-Sep-1984 01:54:22
14-Sep-1984 12:51:46

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASREADLN.B32;1

Page 15
(7)

PAS
1-C

: 561 0618 1 END
: 562 0619 1
: 563 0620 0 ELUDOM

! End of module PASS\$READLN2

PSECT SUMMARY

Name	Bytes	Attributes
_PASS\$CODE	365	NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	2	0	581	00:01.0
_\$255\$DUA28:[PASRTL.OBJ]PASLIB.L32;1	427	124	29	33	00:00.4

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS\$:PASREADLN/OBJ=OBJ\$:PASREADLN MSRC\$:PASREADLN/UPDATE=(ENH\$:PASREADLN)

: Size: 344 code + 21 data bytes
: Run Time: 00:12.0
: Elapsed Time: 00:40.5
: Lines/CPU Min: 3110
: Lexemes/CPU-Min: 20688
: Memory Used: 101 pages
: Compilation Complete

The image displays a grid of 150 small document thumbnails, arranged in 10 rows and 15 columns. Each thumbnail contains technical text, diagrams, and code snippets, typical of a software manual or technical specification. The thumbnails are arranged in a grid pattern, with some larger text labels overlaid on the grid. The labels include:

- PASREAB00 LIS
- PASREAST1 LIS
- PASREAREH LIS
- PASREASETK LIS
- PASREAREG LIS
- PASREAB LIS
- PASREASET2 LIS
- PASREAREF LIS
- PASREAVAR LIS
- PASREARED LIS
- PASREACHA LIS
- PASREADUT LIS
- PASREARENU LIS
- PASREAREINT LIS
- PASREASTR LIS