

[illegible]

```

LL          IIIII
LL          IIIII
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LLLLLLLLLLL
LLLLLLLLLLL

SSSSSSSS
SSSSSSSS
SS
SS
SS
SS
SSSSSS
SSSSSS
SS
SS
SS
SS
SSSSSSSS
SSSSSSSS

```



```
1 0001 0 MODULE showusers (IDENT = 'V04-000',
2 0002 0 ADDRESSING_MODE (EXTERNAL = 'GENERAL')) =
3 0003 1 BEGIN
4 0004 1
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
32 0032 1 FACILITY: SHOW utility
33 0033 1
34 0034 1 ABSTRACT:
35 0035 1 This module contains the option routine for SHOW USERS.
36 0036 1
37 0037 1 ENVIRONMENT:
38 0038 1 VAX native, user mode.
39 0039 1
40 0040 1 AUTHOR: Gerry Smith CREATION DATE: 25-Jun-1982
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 V03-008 MHB0120 Mark Bramhall 6-Apr-1984
45 0045 1 Fix for the case of no interactive users.
46 0046 1
47 0047 1 V03-007 MHB0114 Mark Bramhall 20-Mar-1984
48 0048 1 Remove underscores from terminal names.
49 0049 1
50 0050 1 V03-006 MHB0103 Mark Bramhall 29-Feb-1984
51 0051 1 Add physical terminal name for virtual terminals.
52 0052 1
53 0053 1 V03-005 MHB0101 Mark Bramhall 2-Feb-1984
54 0054 1 Add "(Disconnected)" to the display line for
55 0055 1 disconnected processes.
56 0056 1
57 0057 1 V04-004 PDG0001 Peter D Gilbert 04-Nov-1983
```


SHOWUSERS
V04-000

C 2
16-Sep-1984 01:23:38
14-Sep-1984 12:09:49

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]SHOWUSER.B32;1

Page 2
(1)

```
: 58      0058  1  :
: 59      0059  1  :
: 60      0060  1  :
: 61      0061  1  :
: 62      0062  1  :
: 63      0063  1  :
: 64      0064  1  :
: 65      0065  1  :
: 66      0066  1  :
: 67      0067  1  :
: 68      0068  1  :
: 69      0069  1  :
: 70      0070  1  :
: 71      0071  1  :
: 72      0072  1  :
: 73      0073  1  :
: 74      0074  1  :--
```

```
Don't request 0 bytes ffrom LIB$GET_VM.
Also, check for overflowing the allocated memory.
V03-003 GAS0150      Gerry Smith      28-Jun-1983
Well, there's more than one way to skin a cat.
Rearrange the display, and order by username.
Also, increase the terminal length, to accomodate
the increasing size of terminal names.

V03-002 GAS0106      Gerry Smith      12-Mar-1983
For devices with unit numbers greater than one
digit, correct the sorting method so that they
are displayed after devices with one-digit numbers.

V03-001 GAS00102      17-Jan-1983
Fix display of PIDs.
```


SHOWUSERS
V04-000

D 2
16-Sep-1984 01:23:38
14-Sep-1984 12:09:49

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]SHOWUSER.B32;1

Page 3
(2)

```
: 76      0075 1
: 77      0076 1
: 78      0077 1  ! Include files
: 79      0078 1  !
: 80      0079 1
: 81      0080 1  LIBRARY 'SYSS$LIBRARY:LIB';
: 82      0081 1  REQUIRE 'SRC$:SHOWDEF';
: 83      0180 1
: 84      0181 1  MACRO
: 85      0182 1      list$w_length = 0,0,16,0%;
: 86      0183 1      list$w_item   = 2,0,16,0%;
: 87      0184 1      list$l_buff   = 4,0,32,0%;
: 88      0185 1      list$l_rlen   = 8,0,32,0%;
: 89      0186 1      zero          = 0,0,32,0%;
```

```
! VAX/VMS system definitions
! Get SHOW definitions
```

SHOWUSERS
V04-000

E 2
16-Sep-1984 01:23:38
14-Sep-1984 12:09:49

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]SHOWUSER.B32;1

Page 4
(3)

```
: 91      0187 1 |
: 92      0188 1 | Table of contents
: 93      0189 1 |
: 94      0190 1 |
: 95      0191 1 FORWARD ROUTINE
: 96      0192 1     show$users : NOVALUE;
: 97      0193 1 |
: 98      0194 1 EXTERNAL
: 99      0195 1     sys$gw_ijobcnt;
: 100     0196 1 |
: 101     0197 1 EXTERNAL LITERAL
: 102     0198 1     show$_nosuchuser;
: 103     0199 1 |
: 104     0200 1 EXTERNAL ROUTINE
: 105     0201 1     cli$get_value,
: 106     0202 1     show$write_line : NOVALUE,
: 107     0203 1     lib$get_vm;
: 108     0204 1 |
```



```
110 0205 1 GLOBAL ROUTINE show$users : NOVALUE =
111 0206 2 BEGIN
112 0207 2
113 0208 2 ---
114 0209 2
115 0210 2 This routine performs the SHOW USERS function. Using $GETJPIW, data on
116 0211 2 all the interactive users is gathered and displayed. This routine runs
117 0212 2 in user mode, but needs WORLD privilege in order to get all users.
118 0213 2
119 0214 2 ---
120 0215 2
121 0216 2 LOCAL
122 0217 2     status,                                ! General status return
123 0218 2     maxusers,                             ! Maximum interactive users
124 0219 2     index,                                ! Pointer to current process data
125 0220 2     pid_context,                          ! 'Wildcard PID' context
126 0221 2     tt_phydevnam_buf : VECTOR[15,BYTE],    ! Physical device name buffer
127 0222 2     tt_phydevnam_len,                       ! and its length
128 0223 2     arglist : VECTOR[9],                    ! Argument list
129 0224 2     username : REF BLOCKVECTOR[,12,BYTE],
130 0225 2     user_len : REF VECTOR,
131 0226 2     terminal : REF BLOCKVECTOR[,1+15,BYTE],
132 0227 2     term_len : REF VECTOR,
133 0228 2     procname : REF BLOCKVECTOR[,15,BYTE],
134 0229 2     proc_len : REF VECTOR,
135 0230 2     pid : REF VECTOR,
136 0231 2     devchar2 : $BBLOCK[4],
137 0232 2     user_desc : $BBLOCK[dsc$ s bln],
138 0233 2     jpilist : BLOCKVECTOR[5,12,BYTE];
139 0234 2
140 0235 2
141 0236 2 ! See if there was a specific username to search for
142 0237 2
143 0238 2 $init_dyndesc(user_desc); ! Initialize dynamic descriptor
144 0239 2 cli$get_value(%ASCII 'USERNAME', user_desc);
145 0240 2
146 0241 2
147 0242 2 ! Get the current number of interactive jobs, add 20 percent, and a few extra,
148 0243 2 ! and use that for the maximum number of users.
149 0244 2
150 0245 2 maxusers = .sys$gw_ijobcnt + (.sys$gw_ijobcnt/5) + 4;
151 0246 2
152 0247 2
153 0248 2 ! Based on the maximum number of users, go allocate space to store info
154 0249 2 ! about them. This is based on there being
155 0250 2     12 bytes per username
156 0251 2     16 bytes per terminal name
157 0252 2     15 bytes per process name
158 0253 2     (4 bytes per length) for each of the above
159 0254 2     4 bytes per PID
160 0255 2     ----
161 0256 2     59 bytes per interactive process
162 0257 2
163 0258 3 IF NOT (status = LIB$GET_VM( %REF(.maxusers*60), pid))
164 0259 2 THEN SIGNAL_STOP(show$_insvirmem, 0, .status);
165 0260 2
166 0261 2 !
```

```
167 0262 2 ! Assign the starting locations for each block
168 0263 2 !
169 0264 2 !
170 0265 2 user_len = pid[.maxusers];
171 0266 2 term_len = user_len[.maxusers];
172 0267 2 proc_len = term_len[.maxusers];
173 0268 2 username = proc_len[.maxusers];
174 0269 2 terminal = username[.maxusers,zero];
175 0270 2 procname = terminal[.maxusers,zero];
176 0271 2 !
177 0272 2 !
178 0273 2 ! Set up the $GETJPI item list, and initialize the block pointer and
179 0274 2 ! Process ID context.
180 0275 2 !
181 0276 2 !
182 0277 2 jplist[0,list$w_length] = 12; ! Username
183 0278 2 jplist[0,list$w_item] = jpi$username;
184 0279 2 !
185 0280 2 jplist[1,list$w_length] = 15; ! Terminal name
186 0281 2 jplist[1,list$w_item] = jpi$terminal;
187 0282 2 !
188 0283 2 jplist[2,list$w_length] = 15; ! Process name
189 0284 2 jplist[2,list$w_item] = jpi$prcnam;
190 0285 2 !
191 0286 2 jplist[3,list$w_length] = 4; ! Process ID
192 0287 2 jplist[3,list$w_item] = jpi$pid;
193 0288 2 !
194 0289 2 jplist[4,0,0,32,0] = 0; ! Zero longword to end list
195 0290 2 !
196 0291 2 index = 0; ! Pointer to blocks
197 0292 2 pid_context = -1; ! Initialize the pid
198 0293 2 !
199 0294 2 !
200 0295 2 ! Loop thru all processes on system, gathering data about the interactive
201 0296 2 ! ones.
202 0297 2 !
203 0298 2 !
204 0299 2 DO
205 0300 2 BEGIN
206 0301 2 LOCAL
207 0302 2 iosb : VECTOR[2]; ! Final status from $GETJPI
208 0303 2
209 0304 2 jplist[0,list$l_buff] = username[.index, zero]; ! Point to next block
210 0305 2 jplist[0,list$l_rlen] = user_len[.index]; ! of user data
211 0306 2 terminal[.index, zero] = ;
212 0307 2 jplist[1,list$l_buff] = terminal[.index, zero] + 1;
213 0308 2 jplist[1,list$l_rlen] = term_len[.index];
214 0309 2 jplist[2,list$l_buff] = procname[.index, zero];
215 0310 2 jplist[2,list$l_rlen] = proc_len[.index];
216 0311 2 jplist[3,list$l_buff] = pid[.index];
217 0312 2 jplist[3,list$l_rlen] = 0;
218 0313 2
219 0314 2
220 P 0315 2 status = $GETJPIW(PIDADR = pid_context, ! Get info on next process
221 P 0316 2 ITMLST = jplist,
222 0317 2 IOSB = iosb);
223 0318 2
```



```
224 0319 3 IF .status THEN status = .(io.b[0])<0,16>; | Get final status
225 0320 3 IF .status | If status is good
226 0321 4 AND (.term_len[.index] NEQ 0) | and this is an interactive user
227 0322 3 AND | and either
228 0323 4 BEGIN | no username was specified
229 0324 4 IF .user_desc[dsc$w_length] EQL 0 | or the usernames match
230 0325 4 THEN true
231 0326 4 ELSE CH$EQL(.user_desc[dsc$w_length], .user_desc[dsc$a_pointer],
232 0327 4 .user_desc[dsc$w_length], username[.index, zero])
233 0328 4 END
234 0329 3 THEN index = .index + 1; | then accept this process for display
235 0330 3 END
236 0331 2 UNTIL .status EQL SS$_NOMOREPROC or .index geq .maxusers;
237 0332 2
238 0333 2 | Check to see if the user was found. If not, signal and go away.
239 0334 2
240 0335 2 IF .user_desc[dsc$w_length] NEQ 0
241 0336 2 AND .index LEQ 0
242 0337 2 THEN (SIGNAL(show$_nosuchuser, 1, user_desc); RETURN);
243 0338 2
244 0339 2
245 0340 2 | Make a heading showing the date and total number of users
246 0341 2
247 0342 2 show$write_line(%ASCII '!!10< !>VAX/VMS Interactive Users!!11< !>!%D',
248 0343 2 %REF(0),
249 0344 2 %ASCII '!4< !>Total number of interactive users = !ZW',
250 0345 2 %REF(.sys$gw_ijobcnt));
251 0346 2
252 0347 2
253 0348 2 | Exit now if nothing to display
254 0349 2
255 0350 2 IF .index LEQ 0
256 0351 2 THEN RETURN;
257 0352 2
258 0353 2
259 0354 2 | Make a heading for displayed information
260 0355 2
261 0356 2 show$write_line(%ASCII '!/ Username Process Name PID Terminal',
262 0357 2 0);
263 0358 2
264 0359 2
265 0360 2 | Set up $GETDVI item list for "disconnected" process checking.
266 0361 2
267 0362 2
268 0363 2 |pilist[0,list$w_length] = 4;
269 0364 2 |pilist[0,list$w_item] = dvi$devchar2;
270 0365 2 |pilist[0,list$l_buff] = devchar2;
271 0366 2 |pilist[0,list$l_rlen] = 0;
272 0367 2 |pilist[1,list$w_length] = 15;
273 0368 2 |pilist[1,list$w_item] = dvi$tt_phydevnam;
274 0369 2 |pilist[1,list$l_buff] = tt_phydevnam_buf;
275 0370 2 |pilist[1,list$l_rlen] = tt_phydevnam_len;
276 0371 2 |pilist[2,zero] = 0;
277 0372 2 tt_phydevnam_len = 0;
278 0373 2
279 0374 2
280 0375 2 | Print all the entries, in username order.
```

```
281 0376 2 !
282 0377 2 INCR i FROM 0 TO .index-1 DO
283 0378 BEGIN
284 0379 LOCAL
285 0380 p; ! Pointer to "lowest" username
286 0381
287 0382 p = 0; ! Use first username
288 0383 INCR i FROM 1 TO .index-1 ! Go thru all the usernames
289 0384 DO
290 0385 IF CH$LESS(.user_len[i], username[i,zero],
291 0386 user_len[p], username[p,zero])
292 0387 THEN p = .i;
293 0388
294 0389 arglist[0] = .user_len[p];
295 0390 arglist[1] = username[p,zero];
296 0391 arglist[2] = .proc_len[p];
297 0392 arglist[3] = procname[p,zero];
298 0393 arglist[4] = .pid[p];
299 0394 arglist[5] = .term_len[p] + 1;
300 0395 arglist[6] = terminal[p,zero];
301 0396 arglist[7] = 0;
302 0397 arglist[8] = tt_phydevnam_buf + 1;
303 0398
304 0399 IF $GETDVIW(devnam = arglist[5], itmlst = jpilist)
305 0400 THEN
306 0401 IF .devchar2[dev$y_det]
307 0402 OR .tt_phydevnam_len EQL 0
308 0403 THEN
309 0404 BEGIN
310 0405 arglist[7] = 12;
311 0406 arglist[8] = UPLIT BYTE ('Disconnected');
312 0407 END
313 0408 ELSE
314 0409 IF CH$NEQ(.tt_phydevnam_len, tt_phydevnam_buf,
315 0410 .arglist[5], .arglist[6])
316 0411 THEN
317 0412 arglist[7] = .tt_phydevnam_len - 1;
318 0413
319 0414 arglist[5] = .arglist[5] - 1;
320 0415 arglist[6] = .arglist[6] + 1;
321 0416
322 0417 show$write_line(%ASCII ' !12AF !15AF !XL !15AF!AF',
323 0418 arglist);
324 0419
325 0420 CH$FILL(-1, 12, username[p,zero]);
326 0421 END;
327 0422
328 0423 RETURN true;
329 0424 1 END;
```

```
.TITLE SHOWUSERS
.IDENT \V04-000\
.PSECT $PLITS,NOWRT,NOEXE,2
```

```
45 4D 41 4E 52 45 53 55 0000 P.AAB: .ASCII \USERNAME\
```



```
010E0008 00008 P.AAA: .LONG 17694728
00000000' 0000C .ADDRESS P.AAB
20 53 4D 56 2F 58 41 56 3E 21 20 3C 30 31 21 00010 P.AAD: .ASCII \!10< !>VAX/VMS Interactive Users!//!11< !\
65 73 55 20 65 76 69 74 63 61 72 65 74 6E 49 0001F
21 20 3C 31 31 21 2F 21 73 72 0002E
44 25 21 3E 00038
010E002C 0003C P.AAC: .ASCII \>!%D\
00000000' 00040 .LONG 17694764
6D 75 6E 20 6C 61 74 6F 54 3E 21 20 3C 34 21 00044 P.AAF: .ASCII \!4< !>Total number of interactive users \
74 63 61 72 65 74 6E 69 20 66 6F 20 72 65 62 00053
20 20 20 20 65 6D 61 6E 72 65 73 55 20 2F 21 00062
20 20 65 76 69 0006C
00 00 00 57 5A 21 20 3D 00074 P.AAE: .ASCII \= !ZW\<0><0><0>
010E002D 00078 .LONG 17694765
00000000' 0007C P.AAH: .ADDRESS P.AAF
20 20 20 20 65 6D 61 6E 72 65 73 55 20 2F 21 0007C P.AAH: .ASCII \!/ Username Process Name PID \
20 20 65 6D 61 4E 20 73 73 65 63 6F 72 50 20 0008B
20 20 20 20 6C 61 6E 69 6D 72 65 54 20 20 0009A
00 00 6C 61 6E 69 6D 72 65 54 20 20 000A4
010E0032 000B0 P.AAG: .ASCII \ Terminal\<0><0>
00000000' 000B4 .LONG 17694770
58 21 20 64 65 74 63 65 6E 6E 6F 63 73 69 44 000B8 P.AAI: .ASCII \Disconnected\
20 46 41 35 31 21 20 46 41 32 31 21 20 000C4 P.AAK: .ASCII \ !12AF !15AF !XL !15AF!AF\<0><0>
00 00 46 41 21 46 41 35 31 21 20 20 4C 000D3
010E001A 000E0 P.AAJ: .LONG 17694746
00000000' 000E4 .ADDRESS P.AAK
```

```
.EXTRN SYSS$GW_IJOB$CNT, SHOW$ NOSUCHUSER
.EXTRN CLIS$GET_VALUE, SHOW$WRITE_LINE
.EXTRN LIB$GET_VM, SYSS$GETJPIW
.EXTRN SYSS$GETDVIW
```

```
.PSECT $CODE$,NOWRT,2
```

```
.ENTRY SHOW$USERS, Save R2,R3,R4,R5,R6,R7,R8,R9,- 0205
R10,R11
MOVAB -160(SP), SP
MOVL #34471936, USER_DESC 0238
CLRL USER_DESC+4
PUSHAB USER_DESC 0239
PUSHAB P.AAA
CALLS #2, CLIS$GET_VALUE
MOVL SYSS$GW_IJOB$CNT, R0 0245
DIVL3 #5, R0, R1
MOVAB 4(R1)[R0], MAXUSERS
PUSHAB PID 0258
MULL3 #60, MAXUSERS, 16(SP)
PUSHAB 16(SP)
CALLS #2, LIB$GET_VM
MOVL R0, STATUS
BLBS STATUS, 1$
PUSHL STATUS 0259
CLRL -(SP)
PUSHL #7869170
CALLS #3, LIB$STOP
MOVAL @PID[MAXUSERS], USER_LEN 0265
MOVAL @USER_LEN[MAXUSERS], TERM_LEN 0266
MOVAL @TERM_LEN[MAXUSERS], PROC_LEN 0267
```

```
OFFC 00000
64 SE FF60 CE 9E 00002
AE 020E0000 8F D0 00007
68 AE D4 0000F
64 AE 9F 00012
0000' CF 9F 00015
00000000G 00 02 FB 00019
50 00000000G 00 D0 00020
50 05 C7 00027
54 04 A140 9E 0002B
10 AE 10 AE 9F 00030
AE 54 3C C5 00033
10 AE 9F 00038
00000000G 00 02 FB 0003B
55 50 D0 00042
11 55 E8 00045
55 DD 00048
7E D4 0004A
00000000G 007812F2 8F DD 0004C
08 03 FB 00052
OC AE 10 BE44 DE 00059 1$:
OC AE 08 BE44 DE 0005F
SA 0C BE44 DE 00065
```

50	57	6A44	DE	0006A	MOVAL	(PROC LEN)[MAXUSERS], USERNAME	0268
58	54	0C	C5	0006E	MULL3	#12, MAXUSERS, R0	0269
50	50	57	C1	00072	ADDL3	USERNAME, R0, TERMINAL	
5B	54	04	78	00076	ASHL	#4, MAXUSERS, R0	0270
	50	58	C1	0007A	ADDL3	TERMINAL, R0, PROCNAME	
28	AE	0202000C	8F	D0 0007E	MOVL	#33685516, JPILIST	0277
34	AE	031D000F	8F	D0 00086	MOVL	#52232207, JPILIST+12	0280
40	AE	031C000F	8F	D0 0008E	MOVL	#52166671, JPILIST+24	0283
4C	AE	03190004	8F	D0 00096	MOVL	#51970052, JPILIST+36	0286
		58	AE	D4 0009E	CLRL	JPILIST+48	0289
			56	D4 000A1	CLRL	INDEX	0291
14	AE		01	CE 000A3	MNEGL	#1, PID CONTEXT	0292
2C	52		0C	C5 000A7	MULL3	#12, INDEX, R2	0305
AE	56		52	C1 000AB	ADDL3	R2, USERNAME, JPILIST+4	
	57		08	BE46	MOVAL	@USER_LEN[INDEX], JPILIST+8	0306
50	30		AE	DE 000B0	ASHL	#4, INDEX, R0	0307
	56		04	78 000B6	PUSHAB	(R0)[TERMINAL]	
			6048	9F 000BA	MOVZBL	#95, @ (SP)+	
	9E		5F	8F 9A 000BD	MOVAB	1(R0)[TERMINAL], JPILIST+16	0308
	38		01	A048	MOVAL	@TERM_LEN[INDEX], JPILIST+20	0309
	3C		0C	BE46	MULL3	#15, INDEX, R0	0310
44	50			0F C5 000CD	ADDL3	PROCNAME, R0, JPILIST+28	
AE	56		5B	C1 000D1	MOVAL	(PROC LEN)[INDEX], JPILIST+32	0311
	50		6A46	DE 000D6	MOVAL	@PID[INDEX], JPILIST+40	0312
	48		10	BE46	CLRL	JPILIST+44	0313
	50		54	AE D4 000E1	CLRL	JPILIST+44	0317
			7E	7C 000E4	CLRL	JPILIST+44	
			28	AE 9F 000E6	PUSHAB	IOSB	
			34	AE 9F 000E9	PUSHAB	JPILIST	
			7E	D4 000EC	CLRL	-(SP)	
			28	AE 9F 000EE	PUSHAB	PID CONTEXT	
			7E	D4 000F1	CLRL	-(SP)	
00000000G	00		07	FB 000F3	CALLS	#7, SYSSGETJPIW	
	55		50	D0 000FA	MOVL	R0, STATUS	0319
	1D		55	E9 000FD	BLBC	STATUS, 4\$	
	55		20	AE 3C 00100	MOVZWL	IOSB, STATUS	0320
	16		55	E9 00104	BLBC	STATUS, 4\$	0321
			0C	BE46	TSTL	@TERM_LEN[INDEX]	
			10	13 0010B	BEQL	4\$	
	50		64	AE 3C 0010D	MOVZWL	USER_DESC, R0	0324
			08	13 00111	BEQL	3\$	
6247	68	BE	50	29 00113	CMPC3	R0, @USER_DESC+4, (R2)[USERNAME]	0327
			02	12 00119	BNEQ	4\$	
			56	D6 0011B	INCL	INDEX	0329
000009A8	8F		55	D1 0011D	CMPL	STATUS, #2472	0331
			08	13 00124	BEQL	5\$	
	54		56	D1 00126	CMPL	INDEX, MAXUSERS	
			03	18 00129	BGEQ	5\$	
			FF79	31 0012B	BRW	2\$	
			64	AE B5 0012E	TSTW	USER_DESC	0336
			17	13 00131	BEQL	6\$	
			56	D5 00133	TSTL	INDEX	0337
			13	14 00135	BGTR	6\$	
			64	AE 9F 00137	PUSHAB	USER_DESC	0338
			01	DD 0013A	PUSHL	#1	
00000000G	00	00000000G	8F	DD 0013C	PUSHL	#SHOWS NOSUCHUSER	
			03	FB 00142	CALLS	#3, LIB\$SIGNAL	
			04	00149	RET		

04	AE	00000000G	00	D0	0014A	6\$:	MOVL	SYSS\$GW_IJOBcnt, 4(SP)	0346
		04	AE	9F	00152		PUSHAB	4(SP)	
		0000'	CF	9F	00155		PUSHAB	P.AAE	0344
		08	AE	D4	00159		CLRL	8(SP)	
		08	AE	9F	0015C		PUSHAB	8(SP)	
		0000'	CF	9F	0015F		PUSHAB	P.AAC	0343
00000000G	00		04	FB	00163		CALLS	#4, SHOW\$WRITE_LINE	
			56	D5	0016A		TSTL	INDEX	0351
			01	14	0016C		BGTR	7\$	
			04	0016E			RET		
		0000'	7E	D4	0016F	7\$:	CLRL	-(SP)	0357
			CF	9F	00171		PUSHAB	P.AAG	
00000000G	00		02	FB	00175		CALLS	#2, SHOW\$WRITE_LINE	
28	AE	00E60004	8F	D0	0017C		MOVL	#15073284, JPILIST	0363
2C	AE		18	AE	9E	00184	MOVAB	DEVCHAR2, JPILIST+4	0365
			30	AE	D4	00189	CLRL	JPILIST+8	0366
34	AE	0112000F	8F	D0	0018C		MOVL	#17956879, JPILIST+12	0367
38	AE		F0	AD	9E	00194	MOVAB	TT_PHYDEVNAM_BUF, JPILIST+16	0369
3C	AE		1C	AE	9E	00199	MOVAB	TT_PHYDEVNAM_LEN, JPILIST+20	0370
			40	AE	D4	0019E	CLRL	JPILIST+24	0371
			1C	AE	D4	001A1	CLRL	TT_PHYDEVNAM_LEN	0372
		59	01	CE	001A4		MNEGL	#1, J	0377
			00B9	31	001A7		BRW	14\$	
			54	7C	001AA	8\$:	CLRL	P	0382
			1D	11	001AC		BRB	10\$	0383
51		55	0C	C5	001AE	9\$:	MULL3	#12, I, R1	0385
50		54	0C	C5	001B2		MULL3	#12, P, R0	0386
			08	BE44	DF	001B6	PUSHAL	@USER_LEN[P]	
			0C	BE45	DF	001BA	PUSHAL	@USER_LEN[I]	
9E		00	9E	2D	001BE		CMPC5	@(SP)+, (R1)[USERNAME], #0, @(SP)+, (R0)-	
			6047		001C4			[USERNAME]	
			03	1E	001C6		BGEQU	10\$	
		54	55	D0	001C8		MOVL	I, P	0387
	DF	55	56	F2	001CB	10\$:	AOBLSS	INDEX, I, 9\$	0385
		6C	08	BE44	D0	001CF	MOVL	@USER_LEN[P], ARGLIST	0389
	55	54	0C	C5	001D5		MULL3	#12, P, R5	0390
70	AE	57	55	C1	001D9		ADDL3	R5, USERNAME, ARGLIST+4	
		74	6A44	D0	001DE		MOVL	(PROC_LEN)[P], ARGLIST+8	0391
	50	54	0F	C5	001E3		MULL3	#15, P, R0	0392
78	AE	50	5B	C1	001E7		ADDL3	PROCNAME, R0, ARGLIST+12	
		7C	10	BE44	D0	001EC	MOVL	@PID[P], ARGLIST+16	0393
E0	AD	0C	01	C1	001F2		ADDL3	#1, @TERM_LEN[P], ARGLIST+20	0394
			10	C4	001F9		MULL2	#16, R4	0395
E4	AD		58	C1	001FC		ADDL3	TERMINAL, R4, ARGLIST+24	
			AD	D4	00201		CLRL	ARGLIST+28	0396
		EC	F1	AD	9E	00204	MOVAB	TT_PHYDEVNAM_BUF+1, ARGLIST+32	0397
			7E	7C	00209		CLRL	-(SP)	0399
			7E	7C	0020B		CLRL	-(SP)	
			38	AE	9F	0020D	PUSHAB	JPILIST	
			E0	AD	9F	00210	PUSHAB	ARGLIST+20	
			7E	7C	00213		CLRL	-(SP)	
		00000000G	08	FB	00215		CALLS	#8, SYSS\$GETDVIW	
			50	E9	0021C		BLBC	R0, 13\$	
05		18	01	E0	0021F		BBS	#1, DEVCHAR2, 11\$	0401
			1C	AE	D5	00224	TSTL	TT_PHYDEVNAM_LEN	0402
			0C	12	00227		BNEQ	12\$	
		E8	0C	D0	00229	11\$:	MOVL	#12, ARGLIST+28	0405

SHOWUSERS
V04-000

M 2
16-Sep-1984 01:23:38
14-Sep-1984 12:09:49

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]SHOWUSER.B32;1

Page 12
(4)

			EC	AD	0000'	CF	9E	0022D		MOVAB	P,AAI, ARGLIST+32	:	0406
						12	11	00233		BRB	13\$	:	0401
EO	AD	00	FO	AD	1C	AE	2D	00235	12\$:	CMPC5	TT,PHYDEVNAM_LEN, TT,PHYDEVNAM_BUF, #0, -	:	0409
					E4	BD		0023D			ARGLIST+20, @ARGLIST+24	:	
						06	13	0023F		BEQL	13\$	:	
		E8	AD	1C	AE	01	C3	00241		SUBL3	#1, TT,PHYDEVNAM_LEN, ARGLIST+28	:	0412
					EO	AD	D7	00247	13\$:	DECL	ARGLIST+20	:	0414
					E4	AD	D6	0024A		INCL	ARGLIST+24	:	0415
					6C	AE	9F	0024D		PUSHAB	ARGLIST	:	0417
					0000'	CF	9F	00250		PUSHAB	P,AAJ	:	
						02	FB	00254		CALLS	#2, SHOW\$WRITE_LINE	:	
OC	FF	8F		00		00	2C	0025B		MOVC5	#0, (SP), #-1, #12, (R5)[USERNAME]	:	0420
				6E		6547		00261				:	
		01		59		56	F2	00263	14\$:	AOBLSS	INDEX, J, 15\$	:	0377
						04		00267		RET		:	0424
						FF3F	31	00268	15\$:	BRW	8\$	:	0377

; Routine Size: 619 bytes, Routine Base: \$CODE\$ + 0000

SHOWUSERS
V04-000

N 2
16-Sep-1984 01:23:38
14-Sep-1984 12:09:49

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]SHOWUSER.B32;1

Page 13
(5)

: 331 0425 1 END
: 332 0426 0 ELUDOM

.EXTRN LIB\$SIGNAL, LIB\$STOP

PSECT SUMMARY

Name	Bytes	Attributes
\$PLITS	232	NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$CODE\$	619	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	31	0	1000	00:01.9

COMMAND QUALIFIERS

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

: Size: 619 code + 232 data bytes
: Run Time: 00:16.1
: Elapsed Time: 00:44.6
: Lines/CPU Min: 1591
: Lexemes/CPU-Min: 20589
: Memory Used: 202 pages
: Compilation Complete

0058 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

