

ASSIGN MICI, (FILE, SWAPPER, :DOOCI)
METASYM CI, LØ

H01 12:24 SEP 08, 1975

38
39
40
41
42
43
44
45
46
47
48
49
50
51

P
P
P
P
P
P
P
P
P
P
P
P
P
P

INSWAP USERS JIT/AJIT. THE PROCESSOR AND JIT/AJIT²
ARE THEN READ INTO MEMORY.
NEXT THE SWAPPER INITIALIZES THE USERS JIT COMMAND
LIST AND JXICMAP TABLE. THE USER IS THEN READ.
PROCESSORS ASSOCIATED WITH THE INSWAP USER
HAVE THEIR PHYSICAL PAGES PLACED IN HIS
JXICMAP TABLE AND HE IS NOW READY TO RUN.

IN SOME CASES THE USER MAY ALREADY BE IN CORE
AND THE PURPOSE OF THE INSWAP IS TO ALLOCATE
PHYSICAL PAGES, (THERE MAY NOT HAVE BEEN ANY
WHEN THE USER NEEDED THEM AND HE WAS BLOCKED)
OR TO BRING IN A PROCESSOR.

H01 1212* SEP 08, 1975

52
53
54
55
56
57

PAGE
DEF
DEF
DEF
DEF
DEF

SPMAP
SWAPIN
SWAPOUT
SADRINC
LOGOFF

PUT PROCS INTO CMAP
IN SWAP USER AND PROCS
OUTSWAP USERS
INCREMENT SEEK ADDRESS
LOGON TEXTC

58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94

PAGE

* ALL REFS *

* USER LABEL REFS

REF	UX:JIT	PHYS JIT PAGE NR.
REF	UH:FLG	USER FLAGS
REF	UH:JIT	JIT DISC ADDRESS
REF	UH:AJIT	AJIT DISC ADDRESS
REF	UB:PCT	USER SIZE IN PAGES
REF	UB:ACP	USER COMMAND PROCESSOR
REF	UB:APR	USER SHARED PROCESSOR
REF	UB:APB	SHARED PROCESSOR OVERLAY
REF	UB:ASP	SPEC SHARED PROC (TEL, LINKETC)
REF	UB:DB	USER DEBUGGER
REF	UB:BV	USER MONITOR OVERLAY
REF	UB:US	USERS STATE
REF	UB:BL	LINK BACK TO NEXT USER
REF	UB:PRIO	CURRENT PRIORITY
REF	S:GJOBACN	GHOST JOB ACCOUNT
REF	S:CUP	CURRENT USER PRIORITY

*
* JIT REFERENCES FOR SCHEDULER
*

REF	J:ASSIGN	SWAP PURE PROC FLAG
REF	J:JIT	JIT
REF	JX:MAP	PHYS PAGE TABLE IN JIT
REF	JB:NRG	NR. REMAINING GRANULES
REF	JB:PCC	CONTEXT PAGE COUNT
REF	J:CL	BEG OF CL IF NO AJIT
REF	J:CLPA	COMMAND LIST PHY ADR
REF	J:CLE	CL END IE DISP FROM BEG OF CL
REF	J:CC	MAX COMMAND CHAIN LENGTH IN JIT
REF	J:CLP	DISP IN CL WHERE TIC WENT
REF	J:CLT	SAVED WORD IN CL WHERE TIC WENT

95	REF	JAJ	AJIT'S PHY PG ADR
96	REF	JVLCS	VIRTUAL LINK CHAIN STOP
97	REF	JJITVP	JIT VIRT PAGE NR.
98	REF	JAJITVP	AJIT VIRT PAGE NR.
99	REF	JXBUFVP	SPARE BUFFER VIRT PAGE NR.
100	REF	JBUPVP	BEGINNING USER PAGE VIRT NR.
101	REF	JSBUF1VP	SPEC WINDOW PAGE NR.
102	REF	JQVVP	FIRST OVERLAY PAGE
103	REF	JSPBFLG	SPARE BUFFER MAPPED FLAG
104	REF	JHIDA	DISC ADDRESS TABLE IN JIT
105	REF	JDA	DISPLACEMENT TO DA TABLE
106	REF, 1	JBINFP00L	INITIALIZED BY SWAPIN
107	REF	JLMAP	VIRT PAGE TABLE
108	REF	JCMAP	PHYS PAGE TABLE
109	REF	JPPH	PHYS PAGE CHAIN HEAD
110	REF	JPPT	PHYS PAGE CHAIN TAIL
111	REF	JPPC	PHYS PAGE CHAIN COUNT
112	REF	HANSWAPS	NUMBER OF OUTSWAPS DONE
113	REF	J:NR5	OVERFLOW BIT ON NR. OUTSWAPS
114	REF	JVLH	VIRTUAL PAGE CHAIN HEAD
115	REF	JVLT	VIRTUAL PAGE CHAIN TAIL
116	REF	JPCP	PROCEDURE PAGE COUNT
117	REF	JBNASP	CHOICE FOR NEXT GRANULE NR.
118	REF, 1	JBIPRIV	USER PRIVILEGE
119	REF, 1	JBIPNR	INITIALIZED AT FIRST INSWAP
120	REF	PRDCRM	INITIALIZED AT FIRST INSWAP
121	REF	PRDPRM	INITIALIZED AT FIRST INSWAP
122	REF	TMDCRM	INITIALIZED AT FIRST INSWAP
123	REF	TMDPRM	INITIALIZED AT FIRST INSWAP
124	REF	TMPDCPK	INITIALIZED AT FIRST INSWAP
125	REF	TMPDPPK	INITIALIZED AT FIRST INSWAP
126	REF	MIUC	INITIALIZED AT FIRST INSWAP
127	*		
128	* MEM MGMT REFS		
129	*		
130	REF	TIFFP	FREE PAGE
131	REF	MX:PPUT	MONITOR PHYSICAL PAGE TABLES

132		REF	FPMC	FREE PG MAP CONSTANT
133		REF	NPMC	UNALLOCATED PAGE MAP CONSTANT
134		* LINE TABLE REFS		
135		*		
136		SREF	LBIUN	LINE TO USER NO TABLE
137		*		
138		* PERFORMANCE MEASUREMENT REFS		
139		*		
140	00000001	PERFORM	EQU	PMONOFF
141		REF	DID*IB	INDICATES I/O IN PROGRESS
142	00000001		DB PERFORM	
143		REF	CURNTIM	RETURNS CURRENT TIME
144		REF	OSWAP*MEAS	MEASURE OUTSWAP TIME
145		REF	ISWAP*MEAS1	MEASURE INSWAP PROC TIME
146		REF	ISWAP*MEAS2	MEASURE INSWAP USER TIME
147		REF	TIDSTRB	MEASURE NR. OUTSWAP USERS
148		REF	CHIDDS	DIST TABLE FOR OUTSWAP NR.
149		FIN		
150		* PRDC. TABLE REFERENCES		
151		*		
152		*		
153		REF	PXIHPP	HEAD OF PHYSICAL PAGES
154		REF	PXITPP	TAIL
155		REF	PBIPSZ	PROCEDURE SIZE
156		REF	PBIDSZ	DATA SIZE
157		REF	PBIDCBSZ	DCB SIZE
158		REF	PH: PDA	PROCEDURE DISCARD
159		REF	PH: DDA	DATA DISC ADD
160		REF	PBIPVA	PROCEDURE VIRTUAL ADD
161		*		
162		*		
163		REF	T:SEXIT	EXIT TO SCHEDULER

	PAGE		
164			
165	*		
166	* STEP REF		
167	*		
168	REF	TIDELUS	INITIALIZE KEYIN'S JIT
169	REF	OPSD	BUILD JIT TSTACK AFTER ERROR*
170	*		
171	* SCHEDULER REFS		
172	*		
173	REF	SIHIR	
174	REF	ATITLE	
175	REF	SVILIM	
176	REF	TIPGCHK	CHECK PAGE CHAINS
177	REF	PGCHKM	CHECK MONITOR PAGE CHAIN
178	REF	TSS2	RECYCLE THROUGH SWAP SCHED.
179	REF	TIRE	REPORT EVENT
180	REF	GETJIT	GET A JIT FOR ADD1
181	REF	GIVEUP	GIVE UP SWAP AFTER GETJIT
182	REF	EIUOFAC	UNQUE FOR ALLYCAT
183	* SWAPPER REFS		
184	*		
185	REF	M:HLTIC	TIC TO SENSE COMMAND
186	REF	UB:SWAP1	USERS SWAP DEVICE INDEX
187	REF	M:GASLIM	#SECTORS/TRACK ON SWAPPER
188	REF	MB:GAM6	MASK FOR SECTOR ADDRESS
189	REF	MIADRINCR	ADD TO LAST SECTOR TO GET NEXT
190	*,*		TRACK*
191	REF	UMI#LG2	SWAP ERROR FLAGS
192	REF	TSC3	TEMP DATA
193	REF	TSC4	TEMP DATA
194	REF	TSC2	TEMP DATA
195	REF	TSC1	TEMP DATA
196	REF	SCL\$END	END SHELL CLIST
197	REF	S:BECL	BEGIN/END CLIST ADDRESS TABLE
198	REF	SXIHPP	TEMP PROCESSOR PAGE CHAIN HD
199	REF,2	SDA\$END	END SHELL DISC ADDR TABLE
200	REF	SIISUN	INSWAP USER NUMBER

201	REF	SBINP	NUMBER PROCESSORS IN
202	REF	SBIPNL	PROCESSOR NUMBER TABLE
203	REF	SI:PCT	NR. PAGES NEEDED FOR INSWAP
204	REF	SI:FPPH	SWAPPER PAGE CHAIN HEAD
205	REF	SI:FPPT	SWAPPER PAGE CHAIN TAIL
206	REF	SI:FPPC	SWAPPER PAGE CHAIN COUNT
207	REF	SI:AJP	AJIT PAGE NUMBER
208	REF	SB:BSN	NR. OUTSWAP USERS
209	REF	SB:BSUL	OUTSWAP USER TABLE
210	REF	SIBDA	BEGIN DISC ADDRESS ATBLE
211	REF	SH:EDA	END DISC ADDRESS TABLE
212	REF	SI:BEL	BEGIN CLIST TABLE
213	REF	SI:ECL	END CLIST TABLE
214	REF	SI:SCL	SHELL COMMAND LIST
215	REF	SH:SDA	SHELL DISC ADDRESS TABLE
216	REF	SI:JCL	JIT SHELL COMMAND LIST
217	REF	SH:JAJDA	JIT DISC ADDRESS SHELL
218	REF	CHNMSK	SORT MASK
219	REF	COMSK	SORT MASK
220	REF	SI:JSP	INITIAL JIT INSWAP FLAG
221	REF	BANPMC	BA(NULL PAGE MAP CONSTANT)
222	00000001	DB PERFORM	
223		REF TSCO	TEMP DATA
224		FIN	
225	*		
226	* MISC REFS		
227	*		
228	REF	J:ACCN	INITIALIZE JIT
229	REF	IBIG	DISTINGUISH BIG MEMORY
230	REF	ALLOUT	ALLYCAT SWAP FLAG
231	REF	SI:SYMDB	INITIALIZE JIT
232	REF	SB:IRBMX	INITIALIZE JIT
233	REF	SB:IRGMX	INITIALIZE JIT
234	REF	SB:IRDMX	INITIALIZE JIT
235	REF	SB:IRBDF	INITIALIZE JIT
236	REF	SB:IRGDF	INITIALIZE JIT
237	REF	SB:IRODF	INITIALIZE JIT

238	REF	SLIBDF	INITIALIZE JIT
239	REF	SLIBDF	INITIALIZE JIT
240	REF	SLIBMX	INITIALIZE JIT
241	REF	SLIGMX	INITIALIZE JIT
242	REF	SLIBMX	INITIALIZE JIT
243	REF	J:CCBUF	INITIALIZE JIT
244	REF	JITITLE	INITIALIZE JIT
245	REF	JBICUR	INITIALIZE JIT
246	REF	JBIMAX	INITIALIZE JIT
247	REF	SVIRSIZ	INITIALIZE JIT
248	REF	MPP0	INITIALIZE JIT
249	REF	MP0	INITIALIZE JIT
250	REF	MDP0	INITIALIZE JIT
251	REF	MUP0	INITIALIZE JIT
252	REF	MAXG	INITIALIZE JIT
253	REF	ONLN	INITIALIZE JIT
254	REF	GJOB	INITIALIZE JIT
255	REF	JPUF	INITIALIZE JIT
256	REF	DAJ	READ MASTER JIT
257	REF	SB:GJOBUN	GHOST USER NUMBER
258	REF	S:GJOBTL	INITIALIZE JIT
259	REF	TIOV	INITIALIZE JIT
260			
261	REF	M7	CONSTANT
262	REF	M8	CONSTANT
263	REF	M21	CONSTANT
264	REF	M24	CONSTANT
265	REF	Y8	CONSTANT
266	REF	Y01	CONSTANT
267	REF	Y02	CONSTANT
268	REF	Y08	CONSTANT
269	REF	Y008	CONSTANT
270	REF	YFF	CONSTANT
271	REF	X3FFE00	CONSTANT
272	REF	XFFF800	CONSTANT
273	REF	X8000	CONSTANT
274	REF	X7	CONSTANT

H01 12124 SEP 08, 195
275

REF

X4

CONSTANT

10

Line	Address	Hex	Label	Page	Content
276					
277		0000002C	AJFLGS	EQU	X'2C'
278		0000004C	JFLGS	EQU	X'4C'
279		0000004C	HFLGS	EQU	X'4C'
280		00000200	JITIC	EQU	JIC
281		00000010	PPSWAP	EQU	PPSWP
282		00000004	SDLAY	EQU	4
283		00000001	R1	EQU	1
284		00000002	R2	EQU	2
285		00000003	R3	EQU	3
286		00000004	R4	EQU	4
287		00000005	R5	EQU	5
288		00000006	R6	EQU	6
289		00000007	R7	EQU	7
290		00000008	R8	EQU	8
291		00000009	R9	EQU	9
292		0000000A	R10	EQU	10
293		0000000B	R11	EQU	11
294		0000000C	R12	EQU	12
295		0000000D	R13	EQU	13
296		0000000E	R14	EQU	14
297		0000000F	R15	EQU	15
298				BOUND	8
299	01 00000	03C3C3C9	CCI	TEXTC	'CCI'
300	01 00001	40404040		TEXT	' '
301	01 00002	05D3D6C7	LOGOFF	TEXTC	'LOGON'
	01 00003	06D54040			
302	01 00004	00400000	OVSTART	DATA	X'400000'+T;0V
303	01 00005	03000000	SHSDA	GEN,8,24	3,BA(SH;SDA)
304	01 00006	03000000	19AJCL	GEN,8,24	X'03',BA(JH;DA)
305	01 00007	2E000002		GEN,8,24	X'2E',2
306	01 00008	02000000		GEN,8,24	X'02',0
307	01 00009	2C800800		GEN,8,4,20	X'2C',8,X'800'

COM CHAIN, INT ON ERROR & HLAT
 INT ON BC=0 OR ERROR, NO COM CHAIN

```

308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338 01 0000A 72100000 X
339 01 0000B 72400000 A
340 01 0000C 75400000 X
341      01 0000D
342 01 0000D 72220000 X
343 01 0000E 75220000 X
344 01 0000F 72240000 X
    
```

```

PAGE
NAME: SWAPOUT
PURPOSE: TO SWAP OUT ONE OR MORE USERS SPECIFIED IN
        SBIOSUL TABLE.
DESCRIPTION: THIS PROGRAM SELECTS ALL USERS FOR EACH
        SWAP DEVICE AND BUILDS A COMMAND LIST FOR EACH USERS
        JIT AND AJIT IN THE SHELL COMMAND LIST. EACH
        USERS JXICMAP IS EXAMINED AND IF SPARE
        BUFFERS ARE MAPPED INTO WINDOW PAGES THEY ARE
        REPLACED BY THEIR INDEX NUMBER. THE USERS
        ARE ADDED TO THE SWAPPER PAGE CHAIN AND THE
        COMMAND LIST ORDER CODES ARE SET
        TO WRITE. ALL USERS FOR THE SAME DEVICE ARE
        SORTED BY DISC ADDRESSES TO OBTAIN MINIMUM
        LATENCY, AND THEN THE I/O IS PERFORMED BY CALLING
        TSI0/DPSI0.
        1 = RAD TBL INDEX(BYTE 0), AND INDEX TO CL, DA TABLES
        3 = USER #
        7 = USERS JIT ADR
        8 = ACC
        6 = INDEX FOR ADDRESSING & EDA
        12 = BCL
        13 = ECL
        5 = BA(USERS JIT CL)
        9 = SEEK ORDER WITH BA(DA TABLE)
        4 = INDEX TO DA TABLE FOR JIT CL
        2 = POINTER TO NEXT CL ENTRY FOR JIT CL
* GET # OF USERS TO SWAP OUT
SWAPOUT EQU $
REF SBIOSULT
LB,R1 SBIOSN MOVE USER LIST TO TEMP TABLE
LI,4 0
STW,4 #SWAP$DEV
MV$BSUL EQU $
LB,R2 SBIOSUL,R1
STB,R2 SBIOSULT,R1
LB,2 UBISWAPI,2
    
```

H01 12:24 SEP 08, 1975

345 01 00010 22300001 A
 346 01 00011 25340000 A
 347 01 00012 31300004 A
 348 01 00013 69400016
 349
 350 EXT
 351 01 00014 33100000 X
 352 01 00015 49400003 A
 353 01 00016 64100000
 354
 355 EXT
 356 00000000
 357 *S*
 358 *S*
 359
 360 01 00017 2210001A
 361 01 00018
 362 01 00018 22700000 N
 363 01 00019 68000000 X
 364 01 0001A 6A100000 X
 365 01 0001B 22400000 A
 366 01 0001C 32900005
 367 01 0001D 22200000 N
 368 01 0001E 68000022
 369 01 0001F
 370 01 0001F 32200000 X
 371 01 00020 32400000 X
 372 01 00021 32900000 X
 373 01 00022
 374 01 00022 22100000 A
 375 01 00023 72700000 X
 376 01 00024 68300000 X
 377
 378 01 00025
 379 01 00025 723E0000 X
 380 01 00026 6830000B
 381 01 00027 72360000 X

LI,3 1
 SLS,3 0,2
 CW,3 4
 BANZ *+3
 REF S;#SWAP\$DEV
 #SWAP\$DEV EQU S;#SWAP\$DEV
 MTW,1 #SWAP\$DEV
 OR,4 3
 BDR,1 MV\$SUL
 OPEN SB;SUL
 SB;SUL SET SB;SULT
 DB DEBUG
 LI,1 7
 BAL,0 RECORD
 FIN
 LI,1 PGCHKM1
 PGCHKS RES 0
 LI,7 S;FppH
 B T;PGCHK
 PGCHKM1 BAL,1 PGCHKM
 LI,4 0
 LW,9 SHSDA
 LI,2 S;SCL
 B S00
 OS\$CNT EQU \$
 LW,2 TSC1
 LW,4 TSC2
 LW,9 TSC3
 S00 EQU \$
 LI,R1 0
 LB,R7 SB;SN
 BEZ T;EXIT
 * S00A PICK UP ALL USERS THAT GO OUT TO SAME DEVICE
 EQU \$
 LB,R3 SB;SUL,R7
 BEZ S0J6
 LB,R3 UB;SWAP1,R3

INDEX FOR SDA (FOR JIT CL)
 SEEK ORDER
 SCL ADR (FOR JIT CL)

RESTORE 2,4,8,9

INITIALIZE CL,DA TABLES INDEX
 # USERS IN OUTSWAP LIST

BAIL OUT
 ALL USERS THAT GO OUT TO SAME DEVICE

GET USER #
 THIS ONE ALREADY PROCESSED
 GET RAD INDEX

H01 12124 SEP 08, 175

382 01 00028 20100000 A
 383 01 00029 4930002B
 384 01 0002A 75300001 A
 385 01 0002B 71300001 A
 386 01 0002B 4930000B
 387 01 0002C 20100001 A
 388 01 0002D 09700000 N
 389 01 0002E 723E0000 X
 390 01 0002F 72760000 N
 391 01 00030 25700009 A
 392 01 00031 32500007 A
 393 01 00032 25500002 N
 394 01 00033 728A0000 N
 395 01 00034 32600000 X
 396 01 00035 49300039
 397 01 00036 35800000 X
 398 01 00037 6800003A
 399 01 00038 01 00039 758C0000 N
 400 01 0003A 728A0000 N
 401 01 0003B 35800000 X
 402
 403
 404
 405
 406
 407
 408
 409
 410 01 0003C 22800000 N
 411 01 0003D 25800000 N
 412 01 0003E 30800005 A
 413 01 0003F 22600001 N
 414 01 00040 30600008 A
 415 01 00041 22CFFFFF N
 416 01 00042 30C00008 A
 417 01 00043 22D00000 N
 418 01 00044 30D00008 A

AI,R1 0
 BNE S00B
 STB,R3 R1
 S00B EQU \$
 CB,R3 R1
 BNE S0J6
 * GET USER # & USER'S PHY JIT ADR
 AI,R1 1
 PUSH R7
 LB,R3 SB10SUL,R7
 LOAD,7 UXIJIT,3
 SLS,7 9
 LW,5 7
 SLS,5 2=;BIG
 LOAD,8 JPPH,5
 LW,6 SIFPPT
 BNEZ S02
 * SIFPP CHAIN MUST BE INITIATED
 STW,8 SIFPPH
 B S02+1
 * ADD TO SWAPPER'S FREE PG POOL
 S02 EQU \$
 ST0RE,8 MXIIPPT,6
 LOAD,8 JPPT,5
 STW,8 SIFPPT
 *
 * SAVE SPARE BUFFER INDEX IN WINDOW ENTRY.
 *
 LI,8 JXCMP
 SLS,8 =;BIG
 AW,8 5
 LI,6 JAJITVP+1
 AW,6 8
 LI,12 JBUPVP+1
 AW,12 8
 LI,13 JXBUFVP
 AW,13 8

IS IT LOOP 1 FOR THIS DEVICE
 NO
 INITIALZE RAD INDEX
 DOES USER MATCH DEVICE
 NO
 BUMP CL AND DA TABLE INDEX
 SAVE INDEX INTO USER LIST
 PROCESS THIS USER #
 GET PP # OF USERS JIT
 JIT ADR
 BA OR WA (JIT ADR)
 GET HEAD OF USERS PP CHAIN
 GET TAIL OF SWAPPER FREE PP CHAIN
 CHAIN HAS BEEN STARTED
 TO START CHAIN - HEAD TO HEAD
 TAIL TO TAIL
 USER HEAD TO PP POINTED TO BY S TAIL
 GET USER TAIL
 TO SWAPPER TAIL
 BYTE DISPLACEMENT INTO JIT
 B/H DISPLACEMENT INTO JIT
 B/H ADDR OF CMAP
 INDEX TO 1ST WINDOW PG IN CMAP
 B/H ADDR OF 1ST WINDOW PG IN CMAP
 INDEX TO LAST WINDOW PG IN CMAP
 B/H ADDR OF LAST WINDOW PG IN CMAP
 INDEX TO 1ST SPARE IN CMAP
 B/H ADDR OF 1ST SPARE IN CMAP

H01 12124 SEP 08, 175

419	01	00045	09400000	N
420	01	00046	224FFFFFF	N
421	01	00047	30400008	A
422	01	00048	72EC0000	N
423	01	00049	21E00000	N
424	01	0004A	68300054	N
425	01	0004B	71E80000	N
426	01	0004C	68300052	N
427	01	0004D	204FFFFFF	A
428	01	0004E	3140000D	A
429	01	0004F	6810004B	N
430				
431				
432				
433				
434				
435				
436				
437				
438	01	00050	0F000000	X
	01	00051	006B0001	A
439	01	00052	38400008	A
440	01	00053	754C0000	N
441	01	00054	20600001	A
442	01	00055	3160000C	A
443	01	00056	68200046	N
444	01	00057	08400000	N
445	01	00058	32D00000	X
446	01	00059	47DE0000	X
447				
448				
449	01	0005A	22800000	A
450	01	0005B	758A0000	N
451	01	0005C	758A0000	N
452	01	0005D	22600000	N
453	01	0005E	031C0007	A
454	01	0005F	69300062	N

S022 PUSH 4
LI,4 J0VVP-1 INDEX TO LAST SPARE IN CMAP
AW,4 8 B/H ADDR OF LAST SPARE IN CMAP
LOAD,14 0,6 GET PP MAPPED INTO WINDOW
CI,14 FPMC IS IT NOW MAPPED
BE S025 NO

S023 COMPARE,14 0,4 IS IT MAPPED IN THIS SPARE
BE S024 YES
AI,4 *1 NO-SET TO LOOK AT LOWER SPARE
CW,4 13 HAVE ALL SPARES BEEN SEARCHED
BGE S023 NO-KEEP LOOKING

S SCREECH CODE: 6B=01
S REPORTED BY: SWAPPER
S MESSAGE: ERROR IN SPARE BUFFER TABLES.
S REGISTERS: 14=PHYSICAL PAGE FROM WINDOW.
6=PHYSICAL BYTE ADDRESS OF WINDOW PAGE.
REMARKS: THE PHYSICAL PAGE MAPPED INTO THE VIRTUAL
WINDOW PAGE CANNOT BE FOUND IN THE SPARE
BUFFER AREA, BAD JIT INPUT.

S SCREECH X'6B',1

S024 SW,4 8 CONVERT SPARE B/H ADDR TO VP INDEX
STORE,4 0,6 PUT INDEX IN WINDOW OF CMAP

S025 AI,6 1 NEXT WINDOW PAGE
CW,6 12 HAVE ALL WINDOWS BEEN SEARCHED
BLE S022 NO
PULL 4
LW,13 Y008 FLAG FOR INDICIES IN WINDOW
STS,13 JSPBFLG,7

* CLEAR USER'S PP CHAIN HEAD, TAIL & COUNT AND ADD COUNT TO SWAP COUNT
* IS THE CLEARING NECESSARY

LI,8 0 CLEAR USER HEAD
STORE,8 JPPH,5 TAIL
STORE,8 JPPT,5
LI,6 HANSWAPS HA(NUMBER OF OUTSWAPS)
MTH,1 *7,6 INCREMENT THE COUNT
BNEZ *+3 HAVE WE OVERFLOWED

H01 12124 SEP 08, '75

455	01	00060	22D10000	A
456	01	00061	47DE0000	F
457	01	00062	25500000	N
458	01	00063	72CA0000	X
459	01	00064	758A0000	X
460	01	00065	66C00000	X
461				
462	01	00066	32CE0000	X
463	01	00067	32DE0000	X
464	01	00068	30DE0000	X
465	01	00069	35D0000E	A
466	01	0006A	22600000	N
467	01	0006B	728A0000	X
468	01	0006C	F2BC0007	A
469	01	0006D	30B00008	A
470	01	0006E	36B00000	X
471	01	0006F	30B00008	A
472	01	00070	25B00001	A
473	01	00071	38E0000B	A
474				
475	01	00072	52060000	X
476	01	00073	702A0000	F
477	01	00074	6920007A	
478	01	00075	21000010	A
479	01	00076	6940007A	
480				
481	01	00077	728A0000	X
482	01	00078	6830007A	
483	01	00079	35E0000D	A
484				
485		01 0007A		
486	01	0007A	8260000D	A
487	01	0007B	3280000D	A
488	01	0007C	3880000C	A
489	01	0007D	69300080	
490	01	0007E	35C20000	X
491	01	0007F	680000AD	

LI,13	X'100001	SET OVERFLOW BIT
STS,13	J;NRS=J;JIT,7	
SLS,5	IBIG	NOP OR SHIFT FROM HA TO BA(JIT)
LB,12	JPPC,5	
STB,8	JPPC,5	
AWM,12	S;FPPC	
* SET UP BUT SWAP ORDER LIST		
LW,12	JCLPA,7	GET BEG OF CL - PHY ADR
LW,13	JCLPA,7	
AW,13	JCLE,7	ADD DISP = END OF CL
STW,13	14	
LI,6	JBNRG	FIND THE END OF THE NON PURE
LB,8	JPCP,5	PROCEDURE PORTION OF THE
LB,11	*7,6	CLIST
AW,11	8	
DW,11	X4	
AW,11	8	
SLS,11	1	
SW,14	11	
* TEST - PURE PROCEDURE SWAP		
LH,0	UH:FLG,3	
LC	J;ASSIGN=J;JIT,5	CHECK FOR LONG TERM
BCS,2	S04	
CI,0	RPSWAP	SWAP PURE P TEST
BANZ	S04	
* PURE PROCEDURE NOT CHANGED - DON'T SWAP IT - CORRECT ECL		
LB,8	JPCP,5	ANY PP
BEZ	S04	NO
STW,14	13	TRUNCATE CLIST AT END NON-PP
* PUT IN JIT, DISP OF TIC FROM CL & CONTENTS OF SAME IF GOOD OTHERWISE		
S04	EQU	S
LW,6	*13	SAVE WD WHERE TIC WILL BE INSERTED
LW,8	13	ECL
SW,8	12	DISP IN CL WHERE TIC WILL GO
BNEZ	*+3	
STW,12	SIECL,1	
B	S0J1	

H01 12:24 SEP 08, 175

492	01	00080	32A0000C	A
493	01	00081	32B00000	X
494	01	00082	45A00007	A
495	01	00083	49300085	
496	01	00084	30800000	X
497	01	00085	358E0000	X
498	01	00086	356E0000	X
499				
500	01	00087	48000005	N
	01	00088	55060000	X
501				
502	01	00089	22600000	A
503	01	0008A	206FFFF8	A
504	01	0008B	F28C000D	A
505	01	0008C	21800003	A
506	01	0008D	4910008A	
507	01	0008E	2560007E	A
508	01	0008F	826C000D	A
509	01	00090	2560007F	A
510	01	00091	48600000	X
511	01	00092	526C0000	A
512				
513	01	00093	35D20000	X
514	01	00094	55620000	X
515				
516	01	00095	32A00000	X
517	01	00096	32B00000	X

	LW,10	12	
	LW,11	X3FFFE00	
	CS,10	7	
	BNE	\$+2	
	AW,8	Y8	SET IF CL IN JIT
	STW,8	JCLP,7	SAVE DISP IN JIT
	STW,6	JCLT,7	SAVE WORD THAT TIC DESTROYES
*	RESET PPSWAP		
	RSETSST	PPSWP,0,3	
*	GET END DA		
	LI,6	0	
	AI,6	=8	
	LB,8	*13,6	SEARCH BACK FROM END OF CL
	CI,8	3	FOR LAST SEEK 10CD.
	BL	\$=3	
	SLS,6	=2	
	LW,6	*13,6	GET BA OF END DA POINTER
	SLS,6	=1	HA OF END DA POINTER
	AND,6	M24	
	LH,6	0,6	END DA
*	SET UP USERS	ENTRY IN ORDER TABLES =	END OF CL & DA
	STW,13	S:IECL,1	
	STH,6	SHIEDA,1	
*			
	LW,10	Y01	WRITE ORDER
	LW,11	YFF	

```

518
519
520
521
522
523
524
525
526
527
528
529
530
531 01 00097 02200070 A
      01 00098 08100000 N
532 01 00099 32600007 A
533 01 0009A 32500000 X
534 01 0009B 3270000C A
535 01 0009C 32300000 X
536 01 0009D 22100004 A
537 01 0009E 20700002 A
538 01 0009F 3170000D A
539 01 000A0 681000A9
540 01 000A1 47AE0000 A
541 01 000A2 324E0000 A
542 01 000A3 45400000 X
543 01 000A4 693000A6
544 01 000A5 473E0001 A
545 01 000A6 6410009E
546 01 000A7 20700002 A
547 01 000A8 6800009D
548      01 000A9
549 01 000A9 02200070 A
      01 000AA 0A100000 N
550 01 000AB 82300000 X
551 01 000AC 72360000 X
    
```

PAGE

```

* SET USER CL TO WRITE *
* SET WRITE ORDER IN CL & DO WRITE PART OF RD CK
* IF NO PHY PG ALLOCATED, PUT USER # IN CL, OTHERWISE
* SAVE A HALF WD FROM PHY PG IN CL & PUT DA INTO HALF WD
* SAVE A HALF WD FROM PHY PG IN CL & PUT SWAP ID INTO HALF WD
* 3 = USER #, NOW = Y008, INDICATES NPMC
* 4 = PHY PG MEM ADR
* 5 = MASK FOR COMPARING FOR NO PHY PG
* 6 = ADR OF DA & DA
* 7 = WD 2 OF CL ENTRY
    
```

```

      PUSH      7,1
      LW,6      7
      LW,5      M24      BYTE ADR MASK
      LW,7      12      BEGIN CLIST PTR.
      LW,3      Y01
      LI,1      4
      AI,7      2      INC TO NEXT IBCD IN CLIST.
      CW,7      13      CHECK FOR END
      BGE      SOCK8      POSSIBLE
      STS,10    0,7      STORE WRITE ORDER
      LW,4      0,7
      CS,4      BANPMC   WAS A PHYSICAL PAGE ASSIGNED
      BNE      $+2
      STS,3     1,7      NO, SET SKIP FLAGS
      BDR,1     SOCK1    DO FOUR COMMANDS.
      AI,7      2      SKIP SEEK.
      B        SOCK1-1
      EQU      $
      PULL     7,1
      LW,R3    *TSTACK   INDEX INTO USER LIST
      LB,R3    SB:OSUL,R3 GET USER #
    
```

SOCK1

SOCK2

SOCK8

```

552
553
554
555
556
557
558      01 000AD
559 01 000AD      35220000 X
560 01 000AE      52860000 X
561 01 000AF      6830000B
562
563 01 000B0      55880000 X
564 01 000B1      35820000 X
565 01 000B2      32AE0000 X
566 01 000B3      25A0000B A
567 01 000B4      30A00000 X
568 01 000B5      35A40002 A
569 01 000B6      35940000 A
570 01 000B7      20200004 A
571 01 000B8      20400001 A
572 01 000B9      20900002 A
573 01 000BA      6800000B
574
575      01 000BB
576 01 000BB      52860000 X
577 01 000BC      35820000 X
578
579      01 000BD
580 01 000BD      52860000 X
581 01 000BE      72A60000 N
582 01 000BF      25A0000B A
583 01 000C0      30A00000 X
584 01 000C1      35A40002 A
585 01 000C2      35940000 A
586 01 000C3      55880000 X
587 01 000C4      20200004 A
588 01 000C5      31C20000 X
    
```

```

PAGE
*****
*      SET UP AJIT AND JIT CL      *
*      FOR OUT SWAP USER          *
*****
*      SET UP BCL
SBJ1      EQU      $
          STW,2      SIBCL,1      SFT BCL
          LH,8       UH:AJIT,3    GET AJIT DISC ADR
          BEZ        SBJ2        NO AJIT
*      SET UP AJIT CL & SET UP BDA
          STH,8      SH:SDA,4    AJIT'S DA TO SHELL DA TABLE
          STW,8      SIBDA,1     SET BDA
          LW,10     JAJ,7       GET AJIT'S PP#
          SLS,10    11
          AW,10     Y01         WRITE ORDER
          STW,10    2,2        AJITS WRITE ORDER AND MEM ADR TO SCI
          STW,9     0,2        SET SEEK WD
          AI,2      4          UPDATE SCL
          AI,4      1          UPDATE SDA
          AI,9      2          UPDATE ADR IN SEEK WD
          B         SBJ4
*      NO AJIT - SET UP BDA
SBJ2      EQU      $
          LH,8       UH:JIT,3    GET JIT'S DA
          STW,8      SIBDA,1     SET UP BDA
*      SET UP JIT CL & PUT TIC TO REMAINDER OF CL
SBJ4      EQU      $
          LH,8       UH:JIT,3    GET JIT'S DA
          LOAD,10   UX:JIT,3    GET JIT PP #
          SLS,10    11          GET JIT'S MEM ADR
          AW,10     Y01         WRITE ORDER
          STW,10    2,2        SET IN SCL, JITS WRT ORDER & MEM ADI
          STW,9     0,2        SET SEEK WD
          STH,8     SH:SDA,4    SET IN SDA, JITS DA
          AI,2      4
          CW,12     S:IECL,1
    
```

12:24 SEP 08, 1955

```

589 01 000C6 693000CA
590 01 000C7 35220000 X
591 01 000C8 32820000 X
592 01 000C9 55820000 X
593 01 000CA 000CA
594 01 000CA 48C00000 X
595 01 000CB 25C0007F A
596 01 000CC 30C00000 X
597 01 000CD 35C40000 A
598 01 000CE 20200004 A
599 01 000CF 20400002 A
600 01 000D0 20900004 A
601
602 01 000D1 22800000 A
603 01 000D2 358E0000 X
604 01 000D3 35200000 X
605 01 000D4 35400000 X
606 01 000D5 35900000 X
607 01 000D6 08700000 N
608 01 000D7 758E0000 X
609 01 000D8 32C20000 X
610 01 000D9 32D20000 X
611 01 000DA 15CE0000 X
612 01 000DB 000DB
613 01 000DB 64700025
614 01 000DC 20100000 A
615 01 000DD 68300000 X
616 01 000DE 32800001 A
617 01 000DF 72E00001 A
618 01 000EO 09E00000 N
619 01 000E1 4B800000 X

```

S0J5

* SET STOP OF VIRTUAL LINK CHAIN TO ZERO

S0J6

```

BNE S0J5
STW,2 S;ECL,1
LW,8 S;BDA,1
STH,8 SHIEDA,1
EQU $
AND,12 M21
SLS,12 =1
AW,12 Y08
STW,12 0,2
AI,2 4
AI,4 2
AI,9 4
LI,8 0
STW,8 JVLCS,7
STW,2 TSC1
STW,4 TSC2
STW,9 TSC3
PULL R7
STB,R8 SB;BSUL,R7
LW,12 S;BCL,1
LW,13 S;ECL,1
STD,12 S;BECL,7
EQU $
BDR,R7 SBOA
AI,R1 0
BE TISEXIT
LW,R8 R1
LB,14 1
PUSH 14
AND,R8 M8

```

```

COMMAND ADR TO DBL WD
ADD TCI TO USERS BCL
TIC FROM JIT'S CL TO USER'S CL

```

```

UPDATE SDA PAST TIC
UPDATE ADR IN SEEK WD

```

PUT VP # IN VIRT LINK CHAIN STOP

```

GET INDEX INTO USER LIST
FLAG THIS USER ELIMINATED

```

SAVE FOR ERROR PROCESSING IN TSIB

```

GO TRY FOR ANOTHER ONE
IS EVERYBODY QID UP FOR OUTSWAP
YES.(TO SCDLER, OR INTRPT RETURN)
CL AND DA TABLE SIZE TO R8 FOR BSAC
RADX TO 14

```

```

620
621
622
623
624
625 01 000E2 21800001 A
626 01 000E3 693000E7
627 01 000E4 32500001 N
628 01 000E5 32600001 N
629 01 000E6 680000EE
630
631      01 000F7
632 01 000E7 6AF00422
633 01 000E8 08E00000 N
634 01 000E9 35800000 X
635 01 000EA 6AF00000 X
636 01 000EB 72E0000E A
637 01 000EC 09E00000 N
638 01 000ED 6AF00474
639
640      01 000FE
641 01 000EE B2700000 X
642 01 000EF 328E0000 X
643 01 000F0 358A0000 A
644 01 000F1 22700001 A
645
646 01 000F2 33F00000 X
647      00000001
648 01 000F3 6A000000 X
649 01 000F4 35100000 X
650
651
652 01 000F5 08E00000 N
653 01 000F6 22B0001F
654 01 000F7 09B00000 N
655 01 000F8 6AF00000 X
656
    
```

```

PAGE
*****
* SWAP OUT = SORT ORDER CHAIN*
*****
* IF ONE USER SKIP ORDER ETC
      CI,8      1      IS THERE ONLY ONE OUT SWAP USER
      BNE      S08      MORE THAN ONE USER
      LW,5     SIECL+1   FOR 1, GET ECL
      LW,6     SIBCL+1   FOR 1, GET BCL
      B        S09
* DO EVERYTHING FOR MORE THAN ONE USER
S08 EQU $
      BAL,15   OSAC      ORDER SORT & CHAIN
      PULL     R14      GET RAD INDEX
      STW,8    TSC4     SAVE USER COUNT
      BAL,15   SENSE    FIND DISC SECTOR
      LB,R14   R14      RADX RETURNED AS END ACTION INFO
      PUSH     R14      SAVE END ACTION INFO
      BAL,15   ULCLC    UNLINK CL CHAIN
* FOR SIO 6 = BEG OF CHAIN
S09 EQU $
      LW,7     *TSTACK   GET RAD INDEX
      LW,R8    M:HLTIC,R7 HALT COMMAND
      STW,8    0,5      SET UP LAST TIC TO THE HALT ORDER
      LI,7     01       WRITE FCN
*
      MTW,-1   DID$IB   SET SWAP OUT I/O FLAG TO 'TRUE'
      DB      PERFORM
      BAL,0    CURNTIM  GET CURRENT TIME
      STW,1    TSC0
FIN
*
      PULL     R14      GET SWAP RAD INDEX
      LI,11    OS$CONT  RETURN TO OS$CONT FROM TSIO
      PUSH     R11
      BAL,15   SIO
*
    
```

H01

12:24 SEP 08, 1975

22

657 00000001
 658 01 000F9 32300000 X
 659 01 000FA 02200020 A
 01 000FB 0B400000 N
 660 01 000FC 6A400000 X
 661 01 000FD 02200020 A
 01 000FE 0A400000 N
 662
 663

DE PERFORM

LW,3

TSC0

PUSH

2,4

BAL,4

0SWAP,MEAS

RECORD OUT SWAP TIME

PULL

2,4

FIN

*

664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700

01 000FF
0000000

S
S

PAGE
NAME: SWAPIN
PURPOSE: TO BRING IN PROCESSORS AND USERS.
DESCRIPTION: IN SWAP BEGINS BY RETURNING EXCESS
PAGES OBTAINED FROM THE OUTSWAP, TO THE FREE
PAGE POOL. THE SWAPPER PAGE CHAIN CONTAINS
EXACTLY THE NUMBER OF PAGES NEEDED FOR THE
USER AND PROCESSORS COMING IN.
NEXT, AN INSWAP COMMAND LIST IS BUILT IN THE
SHELL COMMAND LIST AREA FOR EACH INSWAP
PROCESSOR. PHYSICAL PAGES ARE TAKEN FROM THE
SWAPPER PAGE CHAIN AND CHAINED TOGETHER
FOR EACH PROCESSOR. THE HEAD IS KEPT IN
A TEMP HEAD TABLE UNTIL THE PROCESSOR IS IN,
AND IS THEN PLACED IN PXIHPP. THE PROCESSORS
ARE SORTED BY DISC ADDRESS TO OBTAIN MINIMUM
A COMMAND LIST IS BUILT IN SJJCL TO BRING
IN THE USERS JIT AND AJIT, IF NEEDED, AND
CHAINED TO THE PROCESSOR COMMAND LIST. A
TEST IS MADE FOR THE INITIAL JIT INSWAP AND
THE SEEK ADDRESS CHANGED TO THE MASTER JIT
DISC ADDRESS IF NECESSARY. THE I/O IS THEN DONE
FOR BOTH PROCESSORS AND JIT.
ONCE THE JIT IS IN THE USERS COMMAND LIST IS
INITIALIZED AND THE PHYSICAL PAGE NUMBERS
INSERTED INTO JXICMAP. THE I/O IS DONE
TO READ IN THE USER AND THE SPARE
BUFFERS REPLACED IN THE WINDOW PAGES AS REQUIRED.
ANY PROCESSORS REQUIRED BY THE USER ARE
PLACED IN JXICMAP AND THE INSWAP IS NOW COMPLETE.
\$
DEBUG
8
RECORD
FIN

PAGE
NAME: SWAPIN
PURPOSE: TO BRING IN PROCESSORS AND USERS.
DESCRIPTION: IN SWAP BEGINS BY RETURNING EXCESS
PAGES OBTAINED FROM THE OUTSWAP, TO THE FREE
PAGE POOL. THE SWAPPER PAGE CHAIN CONTAINS
EXACTLY THE NUMBER OF PAGES NEEDED FOR THE
USER AND PROCESSORS COMING IN.
NEXT, AN INSWAP COMMAND LIST IS BUILT IN THE
SHELL COMMAND LIST AREA FOR EACH INSWAP
PROCESSOR. PHYSICAL PAGES ARE TAKEN FROM THE
SWAPPER PAGE CHAIN AND CHAINED TOGETHER
FOR EACH PROCESSOR. THE HEAD IS KEPT IN
A TEMP HEAD TABLE UNTIL THE PROCESSOR IS IN,
AND IS THEN PLACED IN PXIHPP. THE PROCESSORS
ARE SORTED BY DISC ADDRESS TO OBTAIN MINIMUM
A COMMAND LIST IS BUILT IN SJJCL TO BRING
IN THE USERS JIT AND AJIT, IF NEEDED, AND
CHAINED TO THE PROCESSOR COMMAND LIST. A
TEST IS MADE FOR THE INITIAL JIT INSWAP AND
THE SEEK ADDRESS CHANGED TO THE MASTER JIT
DISC ADDRESS IF NECESSARY. THE I/O IS THEN DONE
FOR BOTH PROCESSORS AND JIT.
ONCE THE JIT IS IN THE USERS COMMAND LIST IS
INITIALIZED AND THE PHYSICAL PAGE NUMBERS
INSERTED INTO JXICMAP. THE I/O IS DONE
TO READ IN THE USER AND THE SPARE
BUFFERS REPLACED IN THE WINDOW PAGES AS REQUIRED.
ANY PROCESSORS REQUIRED BY THE USER ARE
PLACED IN JXICMAP AND THE INSWAP IS NOW COMPLETE.
\$
DEBUG
8
RECORD
FIN

H01 12124 SEP 08 175

701		01 000FF		
702	01	000FF	32800000	X
703	01	00100	38800000	X
704	01	00101	6830010E	
705	01	00102	69200105	
706	01	00103	0F000000	X
	01	00104	001F0000	A
707				
708				
709				
710				
711				
712				
713				
714				
715				
716		01 00105		
717				
718				
719				
720	01	00105	32300000	X
721	01	00106	72160000	N
722	01	00107	6A200000	X
723	01	00108	33F00000	X
724	01	00109	35100000	X
725	01	0010A	64800105	
726	01	0010B	32200001	A
727	01	0010C	6930010E	
728	01	0010D	35200000	X
729	01	0010E	22500000	A
730	01	0010F	46500000	X
731	01	00110	68300118	
732	01	00111	526A0000	X
733	01	00112	21600200	A
734	01	00113	68400116	
735	01	00114	35500000	X
736	01	00115	68000118	

24

SIO	EQU	*	
	LW,8	SIFPPC	PG CNT FROM SWAPPER FREE PG POBL
	SW,8	SIPCT	SUBST PG CNT NEEDED FOR IN USER
	BEZ	SI1A	
	BQZ	SI1	RELEASE EXTRA PGS
	SCREECH	XI1FI	
S	SCREECH CODE: IF		
S	REPORTED BY: SWAPPER		
S	MESSAGE: NOT ENOUGH PAGES TO PERFORM THIS SWAP.		
S	REGISTERS: 8=DEFICIENT PAGE COUNT.		
S	REMARKS: THE NUMBER OF PAGES IN THE SWAPPER PAGE CHAIN		
S	IS LESS THAN THE NUMBER OF PAGES THE SWAP		
S	SCHEDULER THOUGHT WERE NEEDED FOR THE INSWAP(SIPCT)		
S	SOMETHING IS PROBABLY WRONG WITH THE SWAPPER		
S	PAGE CHAIN AFTER THE OUTSWAP.		
SI1	EQU	*	
*	RELEASE EXTRA PGS		
*	1 3 = PG TO RELEASE		
*	4 = REG USED INTERNALLY		
	LW,3	SIFPPH	1ST PP
	LOAD,1	MX:PPUT,3	
	BAL,2	TIFPP	
	MTW,-1	SIFPPC	
	STW,1	SIFPPH	
	BDR,8	SI1	
	LW,2	1	
	BNEZ	*+2	
	STW,2	SIFPPT	
SI1A	LI,5	0	CHECK FOR AND CLEAR ALLYEATS FLAG
	XW,5	ALLOUT	
	BEZ	SI1B	
	LH,6	UHIFLG,5	
	CI,6	JIC	
	BAZ	*+3	
	STW,5	ALLOUT	
	B	SI1B	

H01 12:24 SEP 08, 1975

737	01	00116	22600000	N
738	01	00117	6AB00000	X
739		01 00118		
740	01	00118	32500000	X
741	01	00119	68300308	
742	01	0011A	22100001	A
743	01	0011B	35100000	X
744				

SI1B

LI,6	E:UGFAC
BAL,11	TIRE
EQU	*
LW,5	S:ISUN
BEZ	SIU14
LJ,1	1
STW,1	#SWAP*DEV

IT WAS HIM, REPORT IT

ANY INSWAP USER
NOPE GET OUT

```

745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761 01 0011C 6A100000 X
762 01 0011D 6A100018
763 01 0011E 22700000 N
764 01 0011F 22600000 A
765 01 00120 32900005
766 01 00121 32800000 X
767 01 00122 72100000 X
768 01 00123 68300145
769
770      01 00124
771 01 00124 72320000 X
772
773 01 00125 72460000 X
774 01 00126 3AC00004 A
775 01 00127 66C00000 X
776 01 00128 35720000 X
777 01 00129 52C60000 X
778 01 0012A 35C20000 X
779 01 0012B 75820000 N
780 01 0012C 6800012E
781

```

```

PAGE
*****
* SET UP SWAP IN PROCESSORS *
*****
* SET UP TO SWAP IN PROCESSORS
*
* 1 = PROCESSOR LIST INDEX
* 2 = PP #
* 3 = PROC #
* 4 = PROC SIZE (# OF PGS)
* 6 = INDEX TO DA TABLE
* 7 = POINTER TO NEXT CL ENTRY
* 8 = PHY PG ADR IN 3RD WD OF 4 WORD CL
* 9 = MASK FOR STS
* 12 = DA
* 13 = DA MASK
BAL,1 PGCHKM
BAL,1 PGCHKM
LI,7 S:ISCL GET BEG OF SHELL CL
LI,6 0 INIT DA TABLE POINTER
LW,9 SHSDA SEEK ORDER
LW,8 S:IFPPH GET 1ST PP #
LB,1 SB:INP GET # OF PROCESSORS
BEZ SI2 THERE ARE NONE
* GET NEXT PROCESSOR NUMBER
SIP1 EQU *
LB,3 SB:PNL,1 GET PROC #
* GET PROC SIZE & SET UP BCL & BDA
LB,4 PB:PSZ,3 GET PROC SEG SIZE
LCW,12 4
AWM,12 S:IFPPC
STW,7 S:IBCL,1 SET BCL FOR THIS PROCESSOR
LW,12 PH:PD,3 GET SEG DISC ADR
STW,12 S:IBDA,1 SET BDA FOR THIS PROC
STORE,8 SX:HPP,1 SET PROC TEMP PP HEAD
B SIP3+1
* GET PP # SET UP CL & DAT FOR PROC IN SHELLS

```

H01 12:24 SEP 08, 1975

27

```

782      01 0012D      72840000 N
783      01 0012D      72840000 N
784      01 0012E      32200008 A
785      01 0012F      20804000 A
786      01 00130      25800008 A
787      01 00131      359E0000 A
788      01 00132      20900002 A
789      01 00133      358E0002 A
790      01 00134      55CC0000 X
791      01 00135      22500000 A
792      01 00136      6A000487
793      01 00137      20600001 A
794      01 00138      20700004 A
795      01 00139      6440012D
796
797      01 0013A      35720000 X
798      01 0013B      20CFFFFE A
799      01 0013C      55C20000 X
800
801      01 0013D      75260000 N
802      01 0013E      72840000 N
803      01 0013F      75440000 N
804
805      01 00140      20700004 A
806      01 00141      20900002 A
807      01 00142      20600001 A
808      01 00143      64100124

```

```

SIP3 EQU $
      LOAD,8 MX:PPUT,2 GET A PP # FROM S:FPF
      LW,2 8 SAME PP # IS INDEX TO NEXT PP #
      AI,8 X'4000' READ ORDER
      SLS,8 11
      STW,9 0,7 SEEK
      AI,9 2 TO NEXT DA TABLE ENTRY
      STW,8 2,7 READ & BA(PP ADR)
      STH,12 SH:SDA,6 DA INTO DA TABLE
      LI,5 0 SWAPPER INDEX
      BAL,0 SADRINC INC DISC ADDRESS.
      AI,6 1 INCR INDEX TO NEXT DA TABLE ENTRY
      AI,7 4 INCR TO NEXT CL ENTRY
      BDR,4 SIP3 CONT SETTING UP PROCESSOR
* SET UP ECL & EDA
      STW,7 S:IECL,1 SET ECL FOR THIS PROC
      AI,12 *2 SET TO LAST DA
      STH,12 SH:EDA,1 SET EDA FOR THIS PROC
* GET NEXT PP # AND SET UP PROC CHAIN,S TAIL
      STORE,2 PX:TPP,3 SET TAIL IN PROC PP TAIL
      LOAD,8 MX:PPUT,2 GET NEXT PP #
      STORE,4 MX:PPUT,2 END PROC CHAIN WITH 0
* INCR CLP PAST SPACE FOR TIC AND GO FOR NEXT PROC
      AI,7 4 INCR CL POINTER PAST TIC SPACE
      AI,9 2
      AI,6 1
      BDR,1 SIP1 BACK FOR NEXT PROCESSOR

```

```

809
810
811
812
813
814 01 00144 35800000 X
815      01 00145
816
817 01 00145 35100000 X
818 01 00146 35100000 X
819 01 00147 32100000 X
820 01 00148 52020000 X
821 01 00149 21000200 A
822 01 0014A 48400150
823 01 0014B 72720000 N
824 01 0014C 25700009 A
825 01 0014D 32CE0000 X
826 01 0014E 35C00000 X
827 01 0014F 68000177
828      01 00150
829 01 00150 22700000 N
830 01 00151 22600000 A
831 01 00152 32900000 X
832 01 00153 32800000 X
833 01 00154 35800000 X
834
835 01 00155 2280002C A
836 01 00156 52C20000 X
837 01 00157 4830015A
838 01 00158 52C20000 X
839 01 00159 4930016A
840 01 0015A 35C00000 X
841      01 0015B
842 01 0015B 2280004C A
843 01 0015C 52C20000 X
844 01 0015D 4930016A
845 01 0015E 52220000 X
    
```

```

PAGE
*****
* SET UP SWAP IN JIT *
*****
* SET NEW SIFPP HEAD
STW,8 SIFPPH UPDATE SWAPPER'S FREE PG POOL HEAD
SI2 EQU $
* DETERMINE WHETHER JIT IS IN CORE & SET UP ITS CL IF IT ISN'T
STW,1 SIAJP 0 TO AJIT PP#
STW,1 S:JSP 0 TO JIT SECTOR PGS
LW,1 S:ISUN GET IN SWAP USER #
LH,0 UHIFLG,1 IS JIT IC
CI,0 JITIC JIT IN CORE TEST
BAZ SI35
LOAD,7 UXIJIT,1
SLS,7 9
LW,12 JAJ,7 GET AJIT PP #
STW,12 SIAJP
B SIP4
SI35 EQU $
LI,7 S:JCL BEG OF JIT CL
LI,6 0 BEG OF JIT DA TABLE
LW,9 XFFF800 SET UP PHY PG ADR MASK
LW,8 SIFPPH GET A PP #
STW,8 SIAJP SAVE ALT JIT'S PP #
* IS THERE AN ALTERNATE JIT - YES, GET AJIT FLAGS - NO, GET JIT FLAGS
LI,11 AJFLGS AJIT FLAGS
LH,12 UH:JIT,1
BEZ SIJ4*1
LH,12 UH:AJIT,1 GET ALT JIT DISC ADR
BNEZ SIJ6 NOT = 0, AJIT EXISTS
STW,12 SIAJP SET AJIT PP # TO 0
SIJ4 EQU $
LI,11 JFLGS JIT FLAGS (TERMINATES COMMAND CHAIN)
LH,12 UH:JIT,1 GET JIT DISC ADR
BNEZ SIJ6
LH,2 UH:AJIT,1
    
```

```

846 01 0015F 212FFFFF A
847 01 00160 69300166
848 01 00161 32400000 X
849 01 00162 22000000 A
850 01 00163 35000000 X
851 01 00164 22B00000 N
852 01 00165 68000000 X
853 01 00166
854 01 00166 55220000 X
855 01 00167 35200000 X
856
857 01 00168 55C20000 X
858 01 00169 22C00000 N
859 01 0016A 55CC0000 X
860 01 0016A 32200008 A
861 01 0016B 2580000B A
862 01 0016C 478E0002 A
863 01 0016D 20700003 A
864 01 0016E F5B00007 A
865 01 0016F 20600001 A
866 01 00170 20700001 A
867 01 00171 72840000 N
868 01 00172 2180004C A
869 01 00173 6930015B
870 01 00174 75220000 N
871 01 00175 35800000 X
872 01 00176 72E00000 X
873 01 00177 72800000 X
874 01 00178 22600000 N
875 01 00178 21800001 A
876 01 00179 69200189
877 01 0017A 69100180
878 01 0017B
879 01 0017C 32500001 N
880 01 0017D 32600001 N
881 01 0017E
882 01 0017E

```

```

CI,2 =1
BNE SIJ41
LW,4 SI:ISUN
LI,0 0
STW,0 SB:BSN
LI,11 GIVEUP
B GETJIT
SIJ41 RES 0
      STH,2 UH:JIT,1
      STW,2 SIJSP
*
      STH,12 UH:AJIT,1
      LI,12 DAJ
SIJ6 EQU $
      STH,12 SH:JAJDA,6
      LW,2 8
      SLS,8 11
      STS,8 2,7
      AI,7 3
      STB,11 *7
      AI,6 1
      AI,7 1
      LOAD,8 MX:PPUT,2
      CI,11 JFLGS
      BNE SIJ4
      STORE,2 UX:JIT,1
      STW,8 SI:FPPH
SIP4 EQU $
      LB,R14 UB:SWAPI
      LB,8 SB:INP
      LI,6 SIJCL
      CI,8 1
      BG SI5
      BL SI4
* ONE PROCESSOR TO SWAP IN
      LW,5 SI:ECL+1
      LW,6 SI:BCL+1

```

```

CHECK FOR NO JIT DISC
NO, HE HAS A SPOT
GET USER INDEX TO PROPER REG
CLEAR OUTSWAP LIST
FOR GIVEUP
SET RETURN

SET FIRST TIME FLAG
AND SAVE SEEK ADR FOR JBNASP.

ALL JITS COME FROM DAJ

DA TO JIT DA TABLE
SAME PP# IS INDEX TO NEXT PP #
CONVERT PP# TO BYTE MEM ADR
MEM ADR TO CL+2
INCR CL POINTER TO LAST WD OF ENTRY
SET FLAGS
INCR TO NEXT DAT ENTRY
INCR TO NEXT CL ENTRY
GET NEXT PP #
WAS JIT CL SET UP
NO AJIT CL = DO JIT
SET JIT PP #
UPDATE SI:FPPH

GET RAD INDEX
GET # OF PROCESSORS TO SWAP IN

COMPARE # TO ORDER & CHAIN WITH 1
MORE THAN ONE PROCESSOR TO SWAP IN
NO PROCESSORS
GET BEG & END OF COMMAND CHAIN
GET ECL FOR 1 PRBC
BCL

```

```

883 01 0017F 6800018E
884
885 01 00180
886 01 00180 21000200 A
887 01 00181 694001B5
888 01 00182 32500007 A
889 01 00183 205FFFFFF A
890 01 00184 22600000 N
891 01 00185 32800000 X
892 01 00186 69300188
893 01 00187 72E20000 X
894 01 00188 680001AA
895
896 01 00189
897 01 00189 6AF00422
898 01 0018A 35800000 X
899 01 0018B 6AF00000 X
900 01 0018C 72E0000E A
901
902 01 0018D 6AF00474
903
904 01 0018E
905 01 0018E 32100000 X
906 01 0018F 52020000 X
907 01 00190 21000200 A
908 01 00191 694001A7
909
910 01 00192 72220000 X
911 01 00193 6830019E
912 01 00194 32800000 X
913 01 00195 6930019E
914 01 00196 2280019A
915 01 00197 09800000 N
916 01 00198 33100000 X
917 01 00199 680001A7
918 01 0019A
919 01 0019A 22600000 N

```

```

      B          SI6
* NO PROCESSORS MUST JIT COME IN
SI4 EQU $
      CI,0      JITIC
      BANZ     SI10      PROCESSORS AND JIT ARE IN
      LW,5     7
      AI,5     -1
      LI,6     $I,JCL      GET JCL ADR FOR SI8
      LW,8     $I,JSP
      BNEZ     $+2
      LB,R14   UBISWAPI,R1  GET RAD INDEX
      B        SI9
* PROCESSORS TO SWAP SO SORT, ORDER & CHAIN
SI5 EQU $
      BAL,15   BSAC        ORDER, SORT & CHAIN
      STW,8    TSC4       SAVE USER COUNT
      BAL,15   SENSE      FIND THE DISC SECTOR
      LB,R14   R14        RADX RETURNED AS END ACTION INFO
* UNLINK CL CHAIN INPUT IN 10 = PRESENT SECTOR POSITION
      BAL,15   ULCLC      UNLINK CL CHAIN
* IS JIT IN CORE = NO, SET TIC = YES, AT END OF PROC CL STOP CHAINING
SI6 EQU $
      LW,1     $I,ISUN
      LW,0     UH:FLG,1
      CI,0     JITIC      IS JIT IN CORE
      BANZ     SI8        YES
* JIT MUST BE BROUGHT IN
      LB,2     UBISWAPI,R1  GET USERS SWAP TABLE INDEX
      BEZ     SI7          SAME AS PROCESSORS
      LW,8     $I,JSP
      BNEZ     SI7
      LI,8     SI65
      PUSH    $
      MTW,1    $SWAP,DEV
      B        SI8
SI65 EQU $
      LI,6     $I,JCL

```

H01 12124 SEP 08, 175

920	01	0019B	32100000	X
921	01	0019C	72E20000	X
922	01	0019D	680001A2	
923		01 0019E		
924	01	0019E	22800000	N
925	01	0019F	2580007F	A
926	01	001A0	30800000	X
927	01	001A1	358A0000	A
928		01 001A2		
929	01	001A2	22500003	N
930	01	001A3	33000000	X
931	01	001A4	683001AA	
932	01	001A5	20500004	A
933	01	001A6	680001AA	
934				
935		01 001A7		
936	01	001A7	205FFFFFF	A
937	01	001A8	2280004C	A
938	01	001A9	F5800005	A

	LW,1	S:ISUN	
	LR,R14	UBISWAPI,R1	GET RAD INDEX
	B	SI75	
SI7	EQU	*	
	LI,8	SIJCL	GET JIT CL ADR
	SLS,8	=1	
	AW,8	Y08	CREATE TIC
	STW,8	0,5	CORRECT T/C TO JIT CL
SI75	EQU	*	
	LI,5	SIJCL+3	END OF JIT CL
	MTW,0	S:AJP	IS THERE AJIT CL
	BEZ	SI9	NO
	AI,5	4	CORRECTION FOR PRESENCE OF AJIT CL
	B	SI9	
	* JIT IS IC SB SET HALT FLAG IN LAST COMMAND		
SI8	EQU	*	
	AI,5	=1	
	LI,8	HFLGS	FLAGS TO STOP COMMAND CHAIN
	STB,8	*5	SET HALT FLG

939
 940
 941 01 001AA
 942
 943 01 001AA 22700002 A
 944
 945 01 001AB 33200000 X
 946 00000001
 947 01 001AC 6A000000 X
 948 01 001AD 35100000 X
 949
 950
 951 01 001AE 6AF00000 X
 952
 953 00000001
 954 01 001AF 32300000 X
 955 01 001B0 02200020 A
 01 001B1 0B400000 N
 956 01 001B2 6A400000 X
 957 01 001B3 02200020 A
 01 001B4 0A400000 N
 958
 959

PAGE
 * START UP PROC & JIT SWAP IN
 SI9 EQU \$
 * FOR SI0 6 = BEG OF CHAIN
 LI,7 02 READ FCN
 *
 MTW,2 DID\$10 SET SWAP IN I/O FLAG TO 'TRUE'
 DB PERFORM
 BAL,0 CURNTIM
 STW,1 TSC0
 FIN
 *
 BAL,15 SI0
 *
 DB PERFORM
 LW,3 TSC0
 PUSH 2,4
 BAL,4 ISWAP,MEAS1 RECORD IN SWAP TIME
 PULL 2,4
 FIN
 *

```

960
961
962
963
964
965
966      01 001B5
967 01 001B5 32100000 X
968 01 001B6 72720000 N
969 01 001B7 25700009 A
970 01 001B8 52220000 X
971 01 001B9 21200001 A
972 01 001BA 694003DE
973 01 001BB 32200000 X
974 01 001BC 4830024C
975 01 001BD 22200000 N
976 01 001BE 483001CF
977 01 001BF 32200000 X
978 01 001C0 35200000 X
979 01 001C1 25200009 A
980 01 001C2 20200000 N
981 01 001C3 22400000 N
982 01 001C4 2540047F A
983 01 001C5 12800006
984 01 001C6 12A00008
985 01 001C7 223FFFFC A
986 01 001C8 95880002 A
987 01 001C9 20800002 A
988 01 001CA 654001CC
989 01 001CB 680001CF
990 01 001CC 95A80002 A
991 01 001CD 453001CA
992 01 001CE 654001C7
993      01 001CF
994 01 001CF 52220000 X
995 01 001D0 12A00000
996 01 001D1 22400000 A
    
```

```

PAGE
*****
* PROCESSORS & JIT ARE IN *
* CORRECT SHELL CL *
* SET HEAD OF PROC CHAIN *
*****
SI10 EQU $
      LW,1 SI:ISUN IN SWAP USER #
      LOAD,7 UX:JIT,1
      SLS,7 9
      LH,2 UHIFLG2,1
      CI,2 1
      BANZ SI:JITERR
      LW,2 SI:JSP
      BEZ SI:105
      LI,2 IBIG
      BEZ SI:102 NOT BIG9
      LW,2 SI:FPFH ALLOCATE AN AJIT FOR BIG9..
      STW,2 SI:AJP SAVE ITS PAGE#
      SLS,2 9
      AI,2 JDA DISPLACEMENT TO DISC ADDRESS
      LI,4 *JDA TABLE...
      SAS,4 *1
      LD,8 19AJCL DUMMY COMMAND LIST ENTRIES.
SI102A LD,10 19AJCL+2
      LI,3 *4
      STD,8 *2,4 STORE SEEK
      AI,8 2 INC DA ADDRESS
SI102B BIR,4 $+2
      B SI:102
      STD,10 *2,4 IBCD
      BIR,3 SI:102B
      BIR,4 SI:102A
SI102 EQU $
      LH,2 UHIFLG,1
      LD,10 CCI
      LI,4 0
    
```

H01 12124 SEP 08, '75

997 01 001D2 21200100 A
 998 01 001D3 6940021D
 999 01 001D4 22500000 N
 1000 01 001D5 711A0000 X
 1001 01 001D6 6930021A
 1002 01 001D7 12AA0000 X
 1003 01 001D8 22400000 N
 1004 01 001D9 228000E0 A
 1005 01 001DA 22300000 N
 1006 01 001DB F5860007 A
 1007 01 001DC 22300000 N
 1008 01 001DD F5560007 A
 1009 01 001DE 2230000B N
 1010 01 001DF 21500001 A
 1011 01 001E0 693001E6
 1012 01 001E1 32800004
 1013 01 001E2 358E0000 F
 1014 01 001E3 22800000 N
 1015 01 001E4 358E0000 F
 1016 01 001E5 22300007 N
 1017 01 001E6 02200020 A
 1018 01 001E7 ABAE0003 A
 1019 01 001E8 12AA0000 X
 1020 01 001E9 02200020 A
 1021 01 001EA 2BAE0000 F
 1022 01 001EB 2BAE0000 F
 1023 01 001EC 6800021F

SI10GCK

SI105GNK

CI,2
 BANZ
 LI,5
 CB,1
 BNE
 LD,10
 LI,4
 LI,8
 LI,3
 STB,8
 LI,3
 STB,5
 LI,3
 CI,5
 BNE
 LW,8
 STW,8
 LI,8
 STW,8
 LI,3
 LCI
 STM,10
 LD,10
 LCI
 STM,10
 STM,10
 B

BAT
 SI101
 MAXG
 SBIGJOBUN,5
 SI10NG
 SIGJOB,5
 GJOB
 X'E0'
 JBI,PRIV=BA(JIJIT)
 *7,3
 JBI,PNR=BA(JIJIT)
 *7,3
 TSTACK+5+6=JIJIT
 1
 SI105GNK
 BVSTART
 TSTACK+2=JIJIT,7
 TIDELUS
 TSTACK+5+11=JIJIT,7
 TSTACK+5+2=JIJIT
 2
 *3,7
 S;GJOBACN,5
 2
 TSTACK+5+13=JIJIT,7
 JIACCN=JIJIT,7
 SI101G

BATCH

NOT GHOST
 GET NAME
 SET JIT FLAG
 AND PRIV

GJOBUN GOES IN PNR
 NAME MEBBE GOES IN REGS6,7
 UNLESS ITS KEYIN

SET LINK REG11

NAME GOES IN 2,3

GET ACCOUNT FOR GHOST

SET ACCOUNT IN REGS

Line	Code	Address	Flags	Label	Page	Details
1024					PAGE	
1025					DRIVE TABLES FOR LIMIT DEFAULT STORING	
1026				LIM	COM,2,6,24 CF(2),AF(1),AF(2)	
1027					DEFAULT SERVICE LIMITS	
1028	01	001ED	329A0000 X	LIMLOAD	LW,9	SL:BDP,5
1029	01	001EE	329A0000 X		LW,9	SLIGMX,5
1030						***TEMP UNTIL GHOSTS LOG ON***
1031	01	001EF	329A0000 X		LW,9	SL:BDP,5
1032					SYSTEM MAX SERVICE LIMITS	
1033	01	001F0	00000000 N	SMLoad	DATA	SL:BMX
1034	01	001F1	00000000 N		DATA	SL:GMX
1035	01	001F2	00000000 N		DATA	SL:GMX
1036	01	001F3	00000000 N	SMST0	DATA	J:TITLE=J:JIT
1037	01	001F4	00000000 N		DATA	J:CCBUF=J:JIT
1038	01	001F5	00000000 N		DATA	J:CCBUF=J:JIT
1039	01	001F6	B59C0007 A	LIMSTBRE	STW,9	*7,6
1040	01	001F7	F59C0007 A		STB,9	*7,6
1041	01	001F8	D59C0007 A		STH,9	*7,6
1042	01	001F9	6ABC0000 A		BAL,11	0,6
1043	01	001F9		LIMINF0	EQU	*=1
1044	01	001FA	11000013 N		LIM,WORD	17,ATITLE+19 TIME
1045	01	001FB	11000000 N		LIM,WORD	17,MPP0 L0
1046	01	001FC	11000000 N		LIM,WORD	17,MP0 P0
1047	01	001FD	11000000 N		LIM,WORD	17,MDP0 D0
1048	01	001FE	11000000 N		LIM,WORD	17,MUP0 U0
1049	01	001FF	00000204		LIM,SPEC	0,PSTST PSTORE
1050	01	00200	00000208		LIM,SPEC	0,TSTST TSTORE
1051	01	00201	40000000 N		LIM,BYTE	0,JB:INFP00L=BA(J:JIT)
1052	01	00202	0000020B		LIM,SPEC	0,TDISK
1053	01	00203	00000206		LIM,SPEC	0,PDISK
1054		00000000		WORD	EQU	0
1055		00000001		BYTE	EQU	1
1056		00000003		SPEC	EQU	3
1057	01	00204	359E0000 X	PSTST	STW,9	PRDCRM,7
1058	01	00205	E800000B A		B	+11
1059	01	00206		PDISK	RES	0
1060	01	00206	359E0000 X		STW,9	PRDPRM,7

H01 12124 SEP 08, 175

1061	01	00207	F800000B	A		B	*11
1062	01	00208	359E0000	X	TSTST	STW,9	TMDCRM,7
1063	01	00209	359E0000	X		STW,9	TMPDCPK,7
1064	01	0020A	F800000B	A		B	*11
1065	01	0020B			TDISK	RES	0
1066	01	0020B	359E0000	X		STW,9	TMDPRM,7
1067	01	0020C	359E0000	X		STW,9	TMPDPPK,7
1068	01	0020D	F800000B	A		B	*11
1069					*	RESOURCE	MAXIS
1070	01	0020E	00000000	N	RMLoad	DATA	SBIRBMX
1071	01	0020F	00000000	N		DATA	SBIRGMX
1072	01	00210	00000000	N		DATA	SBIR0MX
1073	01	00211	00000000	N	RMST0	DATA	JBICUR=J:JIT
1074	01	00212	00000000	N		DATA	JB:MAX=J:JIT
1075	01	00213	00000000	N		DATA	JB:MAX=J:JIT
1076					*	RESOURCE	DEFAULTS
1077	01	00214	00000000	N	RDL0AD	DATA	SBIR0DF
1078	01	00215	00000000	N		DATA	SB:RGDF
1079	01	00216	00000000	N		DATA	SBIR0DF
1080	01	00217	00000000	N	RDST0	DATA	JB:MAX=J:JIT
1081	01	00218	00000000	N		DATA	JB:MAX=J:JIT
1082	01	00219	00000000	N		DATA	JB:MAX=J:JIT

1083
 1084 01 0021A 645001D5
 1085 01 0021B 12A00002
 1086 01 0021C 22400000 N
 1087 01 0021D 02200020 A
 1088 01 0021E 2BAE0000 F
 1089 01 0021F F5400007 A
 1090 01 00220 2540007A A
 1091
 1092 01 00221 22500002 N
 1093 01 00222 F51A0007 A
 1094
 1095 01 00223 22500000 N
 1096 01 00224 323801F0
 1097 01 00225 326801F3
 1098 01 00226 30600007 A
 1099 01 00227 B29A0003 A
 1100 01 00228 B59A0006 A
 1101 01 00229 64500227
 1102
 1103 01 0022A 3238020E
 1104 01 0022B 32680211
 1105 01 0022C 30600007 A
 1106 01 0022D 22500000 N
 1107 01 0022E F29A0003 A
 1108 01 0022F F59A0006 A
 1109 01 00230 6450022E
 1110
 1111 01 00231 32380214
 1112 01 00232 32680217
 1113 01 00233 30600007 A
 1114 01 00234 22500000 N
 1115 01 00235 F29A0003 A
 1116 01 00236 F59A0006 A
 1117 01 00237 64500235
 1118
 1119 01 00238 32980000 X

PAGE
 \$I10NG BDR,5 \$I10GCK
 LD,10 LOGOFF
 LI,4 ONLN ONLINE
 \$I101 LCI 2
 STM,10 TSTACK+5+6=J;JIT,7 START UP PROCESSOR
 \$I101G STB,4 *7 SET JOB TYPE IN JIT
 SLS,4 *6 O=BATCH,1=GH0ST,2=ONLINE
 REF J;PUF
 LI,R5 BA(J;PUF)-BA(J;JIT)+2
 STB,R1 *R7,R5 USER NUMBER IN JIT
 * STORE MAXIS SERVICE LIMITS
 LI,5 SV;LIM
 LW,3 SMLBAD,4
 LW,6 SMST0,4
 AW,6 7 PHY JIT
 LW,9 *3,5
 STW,9 *6,5 SAVE IN JIT
 BDR,5 \$=2
 * MAX RESOURCES
 LW,3 RMLBAD,4
 LW,6 RMST0,4 WHERE
 AW,6 7 PHY JIT
 LI,5 SV;RSIZ
 LB,9 *3,5
 STB,9 *6,5
 BDR,5 \$=2
 * DEFAULT
 LW,3 RDLBAD,4
 LW,6 RDST0,4
 AW,6 7
 LI,5 SV;RSIZ
 LB,9 *3,5
 STB,9 *6,5
 BDR,5 \$=2
 * PERIPHERAL DEFAULTS
 LW,9 S;SYMDB,4

```

1120
1121 01 00239 25900070 A
1122 01 0023A 22300000 N
1123 01 0023B 05960007 A
1124 01 0023C 22500000 N
1125 01 0023D 670801ED
1126 01 0023E 326A01F9
1127 01 0023F 72300006 A
1128 01 00240 25960200 A
1129 01 00241 2530007A A
1130 01 00242 670601F6
1131 01 00243 6450023D
1132 01 00244 21400002 A
1133 01 00245 6930024B
1134 01 00246 222FFFFFF A
1135 01 00247 20200001 A
1136 01 00248 71140000 X
1137 01 00249 69300247
1138 01 0024A 352E0000 F
1139 01 0024B 471E0000 A
1140      01 0024C
1141
1142 01 0024C 72100000 X
1143 01 0024D 6830025A
1144
1145 01 0024E 222FFFFC A
1146 01 0024F 2230002C A
1147      01 00250
1148 01 00250 32620000 X
1149 01 00251 53400006 A
1150 01 00252 64100250
1151
1152      01 00253
1153 01 00253 22600000 A
1154 01 00254 72100000 X
1155 01 00255 72320000 X
1156 01 00256 72220000 N
    
```

```

REF,2      JH:LDCF
SLS,9      =16
LI,3       JH:LDCF=HA(J;JIT)
STH,9      *7,3
LI,5       SVILIM
EXU        LIMLOAD,4      PUT IN DEFAULT LIMS
LW,6       LIMINFO,5     GET NEXT VALUE
LB,3       6              GET STORE INFORMATION
SCS,9      0,3           SHIFT VALUE & RESOLUTION
SLS,3      =6            SHIFT IT
EXU        LIMSTORE,3    ONLY RESOLUTION NOW
BDR,5      SI101L        STORE IT
CI,4       2              TO NEXT LIM
BNE        SI1057        ARE WE ONLINE
LI,2       =1            NO, DONT SET C8CLN
AI,2       1
CB,1       LB:UN,2
BNE        =2
STW,2      M:UC=J;JIT+8,7
STS,1      0,7
EQU        $
* GET # OF PROCESSORS SWAPPED IN
LB,1       SB:NP
BEZ        SI12          NONE
* RECREATE SHELL CL WHERE DESTROYED BY PROCESSOR TIC
LI,2       =4
LI,3       AJFLGS
SIP5       EQU          $
LW,6       SIECL,1
STB,3      +6,2
BDR,1      SIP5
* SET UP PROCESSOR'S PP CHAIN HEAD
SIP6       EQU          $
LI,6       0
LB,1       SB:NP          GET # OF PROCESSORS
LB,3       SB:PNL,1      GET A PROCESSOR #
LOAD,2     SX:HPP,1     GET PROC TEMP PP HEAD
    
```

H01 12:24 SEP 08, 175

1157 01 00257 75260000 N
 1158 01 00258 75620000 N
 1159 01 00259 64100255
 1160
 1161 01 0025A
 1162 01 0025A 32100000 X
 1163
 1164 01 0025B 72220000 N
 1165 01 0025C 22500000 N
 1166 01 0025D 25500000 N
 1167 01 0025E 20500000 N
 1168 01 0025F F52A0007 N
 1169 01 00260 52620000 X
 1170 01 00261 21600200 A
 1171 01 00262 69400264
 1172 01 00263 33F00000 X
 1173 01 00264 32200000 X
 1174 01 00265 6830026D
 1175 01 00266 22500000 N
 1176 01 00267 25500000 N
 1177 01 00268 20500000 N
 1178 01 00269 F52A0007 N
 1179 01 0026A 21600200 A
 1180 01 0026B 6940026D
 1181 01 0026C 33F00000 X
 1182
 1183 01 0026D
 1184 01 0026D 32600000 X
 1185 01 0026E 356E0000 X
 1186 01 0026F 25600009 A
 1187 01 00270 20600000 A
 1188 01 00271 69300274
 1189 01 00272 32600007 A
 1190 01 00273 20600000 N
 1191
 1192 01 00274 356E0000 X
 1193

STORE,2 PXIHPP,3 SET HEAD OF PRCS PP CHAIN
 STORE,6 SXIHPP,1 ZAP TEMP HD FOR ANLZ.
 BDR,1 SIP6+2
 * GET IN SWAP USER # & JIT ADR
 SI12 EQU \$
 LW,1 S;ISUN IN SWAP USER #
 * PUT JIT & AJIT PP # IN CMAP
 LOAD,2 UXIJIT,1
 LI,5 JXCMP
 SLS,5 =;BIG
 AI,5 JIJTVP DISP FROM 0 OF JIT IN CMAP
 STORE,2 *7,5 SET JIT PP IN CMAP
 LW,6 UHIFLG,1
 CI,6 JITIC
 BANZ \$+2
 MTW,=1 SIFPPC
 LW,2 SIAJP
 BEZ SI125
 LI,5 JXCMP
 SLS,5 =;BIG DISP FROM 0 OF AJIT IN CMAP
 AI,5 JAJITVP
 STORE,2 *7,5
 CI,6 JITIC
 BANZ \$+2
 MTW,=1 SIFPPC
 * PUT AJIT'S PP ADR IN JIT & CREATE CL ADR
 SI125 EQU \$
 LW,6 SIAJP AJIT PP #
 STW,6 JAJ,7 PUT INTO JIT
 SLS,6 9 CL ADR IF IN AJIT
 AI,6 0
 BNEZ \$+3 IF = 0,
 LW,6 7 GET JIT ADR AND
 AI,6 JCL ADD CL DISP TO IT
 * SET CL PHY ADR
 STW,6 JCLPA,7 SET UP CL'S PHY ADR
 * RESTORE CL WHERE TIC WAS PUT

H01 12'24 SEP 08, '75

1194	01	00275	328E0000	X
1195	01	00276	6830027E	
1196	01	00277	325E0000	X
1197	01	00278	6830027E	
1198	01	00279	6920027D	
1199	01	0027A	48500000	X
1200	01	0027B	21500000	N
1201	01	0027C	6830027E	
1202	01	0027D		
1203	01	0027D	858C0005	A
1204				
1205	01	0027E	325E0000	X
1206	01	0027F	68300293	
1207	01	00280	32800007	A
1208	01	00281	22000000	N
1209	01	00282	68300284	
1210	01	00283	328E0000	X
1211	01	00284	25800002	A
1212	01	00285	32E00000	X
1213	01	00286	32F00000	X
1214	01	00287	32900000	X
1215	01	00288	305E0000	X
1216	01	00289	22000003	A
1217	01	0028A	205FFFFE	A
1218	01	0028B	31500006	A
1219	01	0028C	69100293	
1220	01	0028D	F1000005	A
1221	01	0028E	68200291	
1222	01	0028F	07E00005	A
1223	01	00290	6800028A	
1224	01	00291	07800005	A
1225	01	00292	6800028A	
1226	01	00293		

	LW,8	JCLT,7
	BEZ	SI128
	LW,5	JCLP,7
	BEZ	SI128
	BGZ	SI127
	EOR,5	Y8
	CI,5	JCCL
	BE	SI128
SI127	EDU	*
	STW,8	*5,6
* MASK PHY DISC	ADR POINTER INTO CL	IN JIT (OR AJIT)
SI128	LW,5	JCLG,7
	BEZ	SI113
	LW,8	7
	LI,0	IBIG
	BEZ	*+2
	LW,8	JCLPA,7
	SLS,8	2
	LW,14	Y02
	LW,15	YFF
	LW,9	XFFF800
	AW,5	JCLPA,7
	LI,0	3
SI129	AI,5	*2
	CW,5	6
	BL	SI113
	CB,0	*5
	BLE	*+3
	STS,14	*5
	B	SI129
	STS,8	*5
	B	SI129
SI113	EDU	*

SACED CL WORD

GET DISP OF TIC FROM CL

IS IT MAX CL IN JIT
YES NO NEED TO RESTORE

RESTORE CL

GET # OF WDS IN CL

GET JITS (DATIS) PP #

DAT IS IN AJIT PAGE..
MAKE IT A BA
READ

BA PP # MASK

NEXT CLIST IBCD

END OF CLIST
IS THIS IBCD A SEEK

NO STORE READ ORDER

STORE BA(DISC ADDRESS) FOR SEEK

```

1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249 01 00293 15700008 A
1250 01 00294 20800000 N
1251 01 00295 20900000 N
1252 01 00296 22500000 N
1253 01 00297 32000000 X
1254 01 00298 350A0000 A
1255 01 00299 205FFFFE A
1256 01 0029A 35500000 X
1257 01 0029B 52020000 X
1258 01 0029C 22E00000 A
1259 01 0029D 22F00000 A
1260
1261 01 0029E 21002800 A
1262 01 0029F 684002B5
1263
    
```

```

PAGE
*****
* IF ANY PROCESSORS ARE *
* ASSOCIATED, MOVE JIT DA TO *
* SHELL DA *
* SET PROC DA IN A REG, *
* OTHERWISE SET TO 0 *
* IF DELTA, SET THE ONE DA *
*****
*
* 0 = USER FLAGS
* 2 = INDEX TO 1ST VP LINK
* 3 = # OF ENTRIES IN DA TABLE FOR MOVING JIT'S DA TO SDA
* 4 = PROCESSOR ROOT #
* 5 = ADR OF NEXT DA TABLE ENTRY
* 6 = DA TABLE ADR = 2 FOR MOVING JIT'S DA TO SDA
* 7 = JIT ADR
* 8 = JLMAP
* 9 = JICMAP
* 14 = DA FOR PROCESSOR DCB & INIT OR 0 IF NO PROCESSOR
*
* SET UP SOME REGS FOR THIS SECTION & THE NEXT SECTION
*
STD,7 8
AI,8 JLMAP
AI,9 JICMAP
LI,5 SCL,END
LW,0 MIMLTIC
STW,0 0,5
AI,5 =2
STW,5 TSC3
LW,0 UHIFLG,1 GET USER FLAGS
LI,14 0
LI,15 0
* IF NOT INIT, SKIP DA MOVE
CI,0 INITIDCBS INIT OR DCBS TO BE BROUGHT IN
BAZ SIUO NB
* GET APPRO PROC # AND COMPUTE DISC ADR
    
```

H01 12124 SEP 08, '75

Line	Code	Address	Value	Mode	Label	Value	Label	Value
1264	01	002A0	72420000	X	LB,4	UBIACp,1	GET COMMAND PROC #	
1265	01	002A1	21000080	A	CI,0	TIC		
1266	01	002A2	694002A9		BANZ	SIP9		
1267	01	002A3	72420000	X	LB,4	UBIAPR,1		
1268	01	002A4	693002A9		BNEZ	SIP9		
1269	01	002A5	72420000	X	LB,4	UBIASp,1		
1270	01	002A6	21000400	A	CI,0	DELA		
1271	01	002A7	684002A9		BAZ	#+2		
1272	01	002A8	72420000	X	LB,4	UBIDB,1		
1273	01	002A9			EQU	\$		
1274	01	002A9	72D80000	X	LB,13	PBIDCSZ,4	GET DCB SIZE	
1275					*			
1276	01	002AA	21000800	A	CI,0	INIT		
1277	01	002AB	684002AD		BAZ	#+2	NO INIT	
1278	01	002AC	72F80000	X	LB,15	PBIDSZ,4		
1279	01	002AD	30F0000D	A	AW,15	13		
1280	01	002AE	22500000	A	LI,5	0	SWAP INDEX	
1281	01	002AF	52C80000	X	LH,12	PH;DDA,4		
1282	01	002B0	72D00000	X	LB,13	MB;GAM6		
1283	01	002B1	220002B2		LI,0	#+1		
1284	01	002B2	64F00488		BDR,15	SADRINC+1	INC DISC ADDRESS...	
1285	01	002B3	35C0000E	A	STW,12	14		
1286	01	002B4	52020000	X	LH,0	UHIFLG,1		
1287	01	002B5			EQU	\$		

SIP9

SIP94

1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319

01 002B5
01 002B5 32D00000 X
01 002B6 326E0000 X
01 002B7 306E0000 X
01 002B8 206FFFFE A
01 002B9 32100007 A
01 002BA 25100002 A
01 002BB 32F00007 A
01 002BC 32BE0000 X
01 002BD 72720000 X
01 002BE 21B001FF A
01 002BF 683002C2
01 002C0 31B00007 A
01 002C1 693002C3
01 002C2 F2BE0008 A
01 002C3 22700000 N

PAGE
* PREPARE FOR SETTING CMAP, CL & PPUT & PROCESSORS DA IF NEEDED
*
* 0 = UH:FLG
* 1 = IN SWAP USER #
* 2 = PRES VP #
* 3 = NEW PP #
* 4 = NEXT PP # IN USERS PP CHAIN
* 5 = ADR OF NEXT DA TABLE ENTRY
* 6 = CL ENTRY
* 7 = JIT ADR
* 8 = LMAP ADR
* 9 = CMAP ADR
* 10 = ACCUM TO GET NEXT PPUT LINK
* 12 = PP ADR FOR CL
* 13 = PP ADR MASK

SIUO EQU \$
LW,13 XFFF800
LW,6 JCLE,7
AW,6 JCLPA,7
AI,6 =2
LW,1 7
SLS,1 2
LW,15 7
LW,11 JVLCS,7
LB,7 JVLW,1
CI,11 X'1FF'
BE \$+3
CW,11 7
BNE \$+2
LB,11 *8,7
LI,7 SDA\$END

MASK FOR SELECT STORE PP ADR TO CC
GET DISP OF CL END FROM CL BEG
ADD

BA OF JIJIT

```

1320
1321 01 002C4 32200000 X
1322 01 002C5 21000200 A
1323 01 002C6 694002EA
1324
1325 01 002C7 72340000 N
1326 01 002C8 72C40000 N
1327 01 002C9 32A00000 X
1328 01 002CA 6930031E
1329 01 002CB 72440000 X
1330 01 002CC 21400001 N
1331 01 002CD 68300324
1332
1333 01 002CE 32400000 X
1334 01 002CF 683002D3
1335 01 002D0 73120000 X
1336 01 002D1 75380000 N
1337 01 002D2 32C00004 A
1338 01 002D3 22A00000 A
1339 01 002D4 75A60000 N
1340 01 002D5 73120000 X
1341 01 002D6 25100000 N
1342 01 002D7 75320000 N
1343
1344 01 002D8 32400000 X
1345 01 002D9 683002E4
1346 01 002DA 72A80000 N
1347 01 002DB 75420000 N
1348 01 002DC 25100000 N
1349 01 002DD 75C80000 N
1350 01 002DE 72220000 X
1351 01 002DF 6830033A
1352 01 002E0 F2C40009 N
1353 01 002E1 21C00000 N
1354 01 002E2 69300312
1355 01 002E3 680002F6
1356 01 002E4 32400000 X
    
```

```

PAGE
LW,2 SIISUN
CI,0 JITIC
BANZ SIU1 JIT IC
* JIT NOT IN CORE IS THIS 1ST TIME
LOAD,3 UXIJIT,2
LOAD,12 UXIJIT,2
LW,10 SIJSP
BNEZ SIU7 1ST TIME JIT IN
LB,4 UBIPCT,2
CI,4 1+IBIG HOW MANY PAGES
BE SIU75 NOT 1ST TIME IN BUT JIT ONLY
* SET AJIT & JIT IN PP CHAIN TAIL & 0 END
LW,4 SIAJP
BEZ $+4 NO AJIT PP
MTB,1 JPPC,1
STORE,3 MXIPPUT,4 LINK AJIT TO JIT
LW,12 4
LI,10 0
STORE,10 MXIPPUT,3 0 TO JIT
MTB,1 JPPC,1
SLS,1 IBIG NOP OR SHIFT FROM BA TO HA(JIT)
STORE,3 JPPY,1
* SET IN 1ST PP OF CHAIN TO GET HEAD CORRECT
LW,4 SIFPPH GET 1ST FREE PP
BEZ SI01
LOAD,10 MXIPPUT,4 SAVE NEXT FREE PP
STORE,4 JPPH,1 PUT 1ST PP IN PPH
SLS,1 IBIG NOP OR SHIFT FROM HA TO BA(JIT)
STORE,12 MXIPPUT,4 LINK 1ST P TO JIT OR AJIT
LB,2 JVLH,1 GET 1ST V LINK
BEZ SIU8
LOAD,12 +9,2
CI,12 NPMC
BNE SIU5
B SIU2
SIB01 LW,4 SIAJP NO PAGES TO INSWAP...
    
```

W01 12124 SEP 08, 1975
 1357 01 002E5 693002E7
 1358 01 002E6 32400003 A
 1359 01 002E7 35400000 X
 1360 01 002E8 75420000 N
 1361 01 002E9 680003A3
 1362
 1363 01 002FA
 1364 01 002EA 72220000 X
 1365 01 002EB 6830033A
 1366 01 002EC 32400000 X
 1367 01 002ED F2340009 N
 1368 01 002EE 21300000 N
 1369 01 002EF 69300319
 1370
 1371 01 002F0 72A80000 N
 1372 01 002F1 25100000 N
 1373 01 002F2 72520000 N
 1374 01 002F3 75420000 N
 1375 01 002F4 25100000 N
 1376 01 002F5 75580000 N
 1377
 1378 01 002F6
 1379 01 002F6 21000010 A
 1380 01 002F7 694002FA
 1381 01 002F8 21002800 A
 1382 01 002F9 68400312
 1383 01 002FA 22C00312
 1384 01 002FB 68000409

BNEZ \$+2 USE THE AJIT PAGE, OR
 LW,4 3 THE JIT PAGE, FOR THE HEAD.
 STW,4 SIFPPH
 STORE,4 JPPH,1
 B SIU12
 * JIT WAS IN BUT IF 1ST PP NOT SET (IE=0) SET HEAD PROPER
 SIU1 EQU \$
 LB,2 JVLH,1 GET 1ST V LINK
 BEZ SIU8
 LW,4 SIFPPH
 LOAD,3 *9,2 GET 1ST CMAP
 CI,3 NpMC
 BNE SIU6
 * SET 1ST PP
 LOAD,10 MX:PPUT,4 SAVE NEXT FREE PP
 SLS,1 *IBIG
 LOAD,5 JPPH,1
 STORE,4 JPPH,1
 SLS,1 IBIG
 STORE,5 MX:PPUT,4
 * SET PROC DISC ADR IF PPSWAP SET
 SIU2 EQU \$
 CI,0 PPSWAP
 BANZ \$+3
 CI,0 INIT,DCBS
 BAZ SIU5
 LI,12 SIU5 RET ADDR
 B S\$INIT

```

1385
1386      01 002FC
1387      01 002FC 32300002 A
1388      01 002FD F2240008 A
1389      01 002FE 6830033A
1390      01 002FF 3120000B A
1391      01 00300 6830033A
1392      01 00301 21000010 A
1393      01 00302 69400309
1394      01 00303 21002800 A
1395      01 00304 6840030D
1396      01 00305 F2C40009 N
1397      01 00306 21C00000 N
1398      01 00307 6830030C
1399      01 00308 6800030D
1400      01 00309
1401      01 00309 F2C40009 N
1402      01 0030A 21C00000 N
1403      01 0030B 69300319
1404      01 0030C 6AC00409
1405      01 0030D
1406      01 0030D 72A80000 N
1407      01 0030E F2560009 N
1408      01 0030F 72CA0000 N
1409      01 00310 754A0000 N
1410      01 00311 75C80000 N
1411      01 00312
1412      01 00312 F5440009 N
1413      01 00313 33F00000 X
1414      01 00314 32C00004 A
1415      01 00315 3240000A A
1416      01 00316 25C0000B A
1417      01 00317 C7C00006 A
1418      01 00318 73120000 X
1419      01 00319
1420      01 00319 206FFFFE A
1421      01 0031A F2C00006 A
    
```

```

SIU3 PAGE
      EQU $
      LW,3 2
      LB,2 *8,2 GET V LINK
      BEZ SIU8
      CW,2 11 VLCS
      BF SIU8
      CI,0 PPSWAP
      BANZ SIU35
      CI,0 INITDCBS
      BAZ SIU4
      LOAD,12 *9,2 CMAP
      CI,12 NPMC
      BE SIU35+3
      B SIU4
SIU35 EQU $
      LOAD,12 *9,2 CMAP
      CI,12 NPMC
      BNE SIU6
SIU4 BAL,12 S$INIT
      EQU $
      LOAD,10 MXIPPUT,4 SAVE NEXT FREE PP
      LOAD,5 *9,3 GET PREV PP FROM CMAP
      LOAD,12 MXIPPUT,5
      STORE,4 MXIPPUT,5 LINK BACK
      STORE,12 MXIPPUT,4 LINK F0RE
SIU5 RES 0
      STORE,4 *9,2 CMAP
      MTW,=1 SIFPPC
      LW,12 4 PP INSERTED
      LW,4 10 NEXT FREE PP
      SLS,12 11
      STS,12 *6 SET PP IN CLIST
      MTB,1 JPPC,1
SIU6 EQU $
      AI,6 =2 DEC CLIST PTR
      LB,12 *6 TO NEXT ENTRY
    
```

H01 12124 SEP 08, 175

1422	01	0031B	21C00003	A
1423	01	0031C	68100319	
1424	01	0031D	680002FC	
1425		01	0031E	
1426	01	0031E	72540000	X
1427	01	0031F	725A0000	X
1428	01	00320	4BA00005	A
1429	01	00321	20A0001E	A
1430	01	00322	22500000	N
1431	01	00323	F5AA000F	A
1432		01	00324	
1433	01	00324	22500000	N
1434	01	00325	22A00001	N
1435	01	00326	F5AA000F	A
1436	01	00327	25100000	N
1437	01	00328	75320000	N
1438	01	00329	32400000	X
1439	01	0032A	68300335	
1440	01	0032B	75380000	N
1441	01	0032C	75420000	N
1442	01	0032D	32200000	X
1443	01	0032E	68300335	
1444	01	0032F	32200000	X
1445	01	00330	52C40000	X
1446	01	00331	55C40000	X
1447	01	00332	72540000	X
1448	01	00333	6A000487	
1449	01	00334	55C40000	X
1450		01	00335	
1451	01	00335	75320000	N
1452	01	00336	25100000	N
1453	01	00337	73120000	N
1454	01	00338	22A00000	A
1455	01	00339	75A60000	N
1456		01	0033A	
1457	01	0033A	3270000F	A
1458				

	CI,12	3
	BGE	*-3
	B	SIU3
SIU7	EQU	*
	LB,5	UBISWAPI,2
	LB,5	MBIGAM6,5
	AND,10	5
	AI,10	30
	LI,5	JBNASP
	STB,10	*15,5
SIU75	EQU	*
	LI,5	JBPC
	LI,10	1+1BIG
	STB,10	*15,5
	SLS,1	*1BIG
	STORE,3	JPPH,1
	LW,4	SIJJP
	BEZ	SIU76
	STORE,3	MX:PPUT,4
	STORE,4	JPPH,1
	LW,2	SIJSP
	BEZ	SIU76
	LW,2	SIISUN
	LH,12	UH:JIT,2
	STH,12	UH:AJIT,2
	LB,5	UBISWAPI,2
	BAL,0	SADRINC
	STH,12	UH:JIT,2
SIU76	EQU	*
	STORE,3	JPPT,1
	SLS,1	1BIG
	MTB,1+1BIG	JRPC,1
	LI,10	0
	STORE,10	MX:ppUT,3
SIU8	EQU	*
	LW,7	15

CHECK ORDER CODE
SKIP SEEKS
AGAIN

JIT IN FOR FIRST TIME. SET
NASP FOR FIRST PAGE 30 SECTORS FROM
JIT SECTOR.

2 IF BIG 9 OR
1 TO CONTEXT COUNT

IF HE HAS AN AJIT FIRST TIME

FIRST TIME THROUGH
NO, SKIP WALKING THE JIT

AND SETUP JIT AND AJIT
DISC ADDRESSES...

PUT JIT ON 2ND GRANULE.

INCREMENT BY 2 IF BIG 9

SET JIT ADR BACK TO 7

* INIT VLCS TO LARGEST POSSIBLE P # SINCE IT INDICATES HOW FAR DOWN


```

1459
1460 01 0033B 22A001FF A
1461 01 0033C 35AE0000 X
1462
1463
1464 01 0033D 32200000 X
1465 01 0033E 52340000 X
1466 01 0033F 6930034D
1467 01 00340 22200000 N
1468 01 00341 F2340009 N
1469 01 00342 21300000 N
1470 01 00343 6830034D
1471
1472 01 00344 22300000 N
1473 01 00345 F2560009 N
1474 01 00346 72320000 X
1475 01 00347 F2360009 N
1476 01 00348 75460000 N
1477 01 00349 75580000 N
1478 01 0034A 73120000 X
1479 01 0034B F5440009 N
1480 01 0034C 33F00000 X
1481      01 0034D
1482
1483 01 0034D 32200000 X
1484 01 0034E 52040000 X
1485 01 0034F 21002800 A
1486 01 00350 69400356
1487 01 00351 21000210 A
1488 01 00352 694003A3
1489 01 00353 72540000 X
1490 01 00354 21500001 N
1491 01 00355 683003A3
1492      01 00356
1493 01 00356 325E0000 X
1494 01 00357 305E0000 X
1495 01 00358 35500000 X
    
```

```

* VIRTUAL CHAIN, CORE IMAGE DOES NOT LIVE WITH DISC IMAGE
  LI,10  XI1FFI
  STW,10 JVLCS,7
    
```

```

* SET UP AJIT IF = BOTH ITS DA=0 AND, ITS CMAP= FPMC
    
```

```

  LW,2  SIISUN
  LH,3  UHIAJIT,2
  BNEZ  SIU9  ALREADY HAVE AJIT
  LI,2  JAJITVP
  LOAD,3 *9,2  CMAP
  CI,3  FPMC
  BE    SIU9
    
```

```

* 4 = NEXT FREE PP
    
```

```

  LI,3  JJITVP
  LOAD,5 *9,3  JIT PP IN CMAP
  LB,3  JVL,1
  LOAD,3 *9,3  LAST PP (LINKED TO JIT PP)
  STORE,4 MXI,PUT,3  LINK LAST TO AJIT
  STORE,5 MXI,PUT,4  LINK AJIT TO JIT
  MTB,1  JPPC,1  UPDATE PP CNT
  STORE,4 *9,2  PUT AJIT PP INTO CMAP
  MTW,=1  SIFPPC
    
```

```

SIU9 EQU $
* IS BOTH READY TO RUN RESET AND SWAP PURE P SET . NO I/O, HAD VP NO PI
    
```

```

  LW,2  SIISUN
  LH,0  UHIFLG,2
  CI,0  INITIDCBS
  BANZ  SIU10
  CI,0  PPSWAP+JITIC
  BANZ  SIU12
  LB,5  UBIPCT,2
  CI,5  1+IBIG
  BE    SIU12
    
```

```

SIU10 EQU $
  LW,5  JCLE,7  CL LENGTH
  AW,5  JCLRA,7  LAST CDW+1
  STW,5  TSC1
    
```

HO1 12:24 SEP 08, 175

1496 01 00359 72140000 X
 1497 01 0035A 32A20000 X
 1498 01 0035B 72E40000 X
 1499 01 0035C 69300363
 1500 01 0035D 21002800 A
 1501 01 0035E 68400363
 1502 01 0035F 32A00000 X
 1503 01 00360 20A00002 A
 1504 01 00361 25A0007F A
 1505 01 00362 30A00000 X
 1506 01 00363 46AA0000 A
 1507 01 00363 46AA0000 A
 1508 01 00364 35A00000 X
 1509 01 00365 33200000 X
 1510 00000001
 1511 01 00366 6A000000 X
 1512 01 00367 35100000 X
 1513 01 00368 35100000 X
 1514
 1515 01 00369 22100001 A
 1516 01 0036A 35100000 X
 1517 01 0036B 326E0000 X
 1518 01 0036C 22100000 A
 1519 01 0036D 351E0000 X
 1520 01 0036E 22700004 A
 1521 01 0036F 52040000 X
 1522 01 00370 21000200 A
 1523 01 00371 6940037B
 1524 01 00372 22F00380
 1525 01 00373 21002800 A
 1526 01 00374 68400000 X
 1527 01 00375 20E00000 A
 1528 01 00376 68300000 X
 1529 01 00377 22F0037B
 1530 01 00378 09F00000 N
 1531 01 00379 33100000 X
 1532 01 0037A 68000000 X

SIU10A

LB,1 UBISWAPI,2
 LW,10 MIHLTIC,1
 LB,14 UBISWAPI,2
 BNE SIU10A
 CI,0 INIT+DCBS
 BAZ SIU10A
 LW,10 TSC3
 AI,10 2
 SLS,10 *1
 AW,10 Y08
 EQU \$
 XW,10 0,5
 STW,10 TSC2
 MTW,2 DID\$10
 DB PERFORM
 BAL,0 CURNTIM
 STW,1 TSC0
 STW,1 TSC0
 FIN
 LI,1 1
 STW,1 #SWAP,DEV
 LW,6 JCLPA,7
 LI,1 0
 STW,1 JCLT,7
 LI,7 4
 LW,0 UHIFLG,2
 CI,0 JITIC
 BANZ SIU11A
 LI,15 SIU11B
 CI,0 INIT+DCBS
 BAZ SI0
 AI,14 0
 BE SI0
 LI,15 SIU11A
 PUSH 15
 MTW,1 #SWAP,DEV
 B SI0

RADX
 RADX
 USER NOT ON SAME RAD
 NO INITIAL OR DCBS
 START OF INIT CL
 TIC ORDER
 SAVE SEEK WORD
 SET SWAP IN I/O FLAG TO ITRUE!
 AVOID RESTORING CLIST WORD
 IF NO SWAP OUT OCCURS
 READ W/READ CHECK
 USER IC
 RETURN ADDRESS
 USER ONLY
 USER AND INIT ON SAME RAD
 RETURN IMMEDIATELY
 2 I/O OPERATIONS

```

1533      01 0037B
1534      01 0037B 32600000 X
1535      01 0037C 20600002 A
1536      01 0037D 22700002 A
1537      01 0037E 22E00000 A
1538      01 0037F 6AF00000 X
1539      01 00380
1540
1541      00000001
1542      01 00380 32300000 X
1543      01 00381 02200020 A
           01 00382 0B400000 N
1544      01 00383 6A400000 X
1545      01 00384 02200020 A
           01 00385 0A400000 N

1546
1547      01 00386 32A00000 X
1548      01 00387 85A00000 X
1549      01 00388 32100000 X
1550      01 00389 52020000 X
1551      01 0038A 4900048D
1552      01 0038B 55020000 X
1553
1554
1555
1556
1557
1558      01 0038C 72220000 N
1559      01 0038D 25200009 A
1560      01 0038E 32500000 X
1561      01 0038F 31540000 X
1562      01 00390 684003A2
1563      01 00391 22400000 A
1564      01 00392 47440000 X
1565      01 00393 25200002 N
1566      01 00394 22400000 N
1567      01 00395 25400000 N
    
```

SIU11A

```

EQU $
LW,6 TSC3
AL,6 2
LI,7 2
LI,14 0
BAL,15 SI0
    
```

SWAP IN INIT&/OR DCBS

SIU11B

```

EQU $
DB PERFORM
LW,3 TSC0
PUSH 2,4
BAL,4 ISWAP$MEAS2
PULL 2,4
    
```

RECORD IN SWAP TIME AND TIME FROM

END OF

FIN

```

LW,10 TSC2
STW,10 *TSC1
LW,1 SIISUN
LW,0 UH:FLG2,1
BR,0 =X'18'
STH,0 UH:FLG2,1
    
```

RESTORE SEEK WORD

SET JUST SWAPPED IN
AND DONT SWAP FLAGS
AFTER PHYSICAL IO ONLY

*
*
*
*
*

PLACE SPARE BUFFER PAGES IN WINDOWS

```

LOAD,2 UX:JIT,1
SLS,2 9
LW,5 Y008
CW,5 JSPBFLG,2
BAZ SIU11C
LI,4 0
SYS,4 JSPBFLG,2
SLS,2 2-:BIG
LI,4 JXCMP
SLS,4 =:BIG
    
```

PHYS JIT WORD ADDRESS

ARE INDICIES IN WINDOW
NO
YES-RESET FLAG

PHYS JIT BYTE/HALFWORD ADR
BYTE DISPLACEMENT INTO JIT
B/H DISPLACEMENT INTO JIT

H01 12:24 SEP 08 '75

1568	01	00396	30400002	A
1569	01	00397	22200001	N
1570	01	00398	30200004	A
1571	01	00399	22600000	N
1572	01	0039A	72540000	N
1573	01	0039B	21500000	N
1574	01	0039C	683003A0	
1575	01	0039D	30500004	A
1576	01	0039E	725A0000	N
1577	01	0039F	75540000	N
1578	01	003A0	20200001	A
1579	01	003A1	6460039A	
1580	01	003A2	52020000	X

SIU11C

AW,4	2
LI,2	JAJITVP+1
AW,2	4
LI,6	JBUPVP,JSBUF1VP
LOAD,5	0,2
CI,5	FPMC
BE	\$+4
AW,5	4
LOAD,5	0,5
STORE,5	0,2
AI,2	1
BDR,6	\$=7
LH,0	UH:FLG,1

51

B/H ADDR OF CMAP
INDEX TO 1ST WINDOW PG IN CMAP
B/H ADDR OF 1ST WINDOW PG IN CMAP
OF WINDOW PAGES
GET SPARE INDEX FROM WINDOW

NOTHING MAPPED IN WINDOW
GET B/H ADDR OF SPARE
GET PHYS PG FROM SPARE
AND PLACE IN WINDOW
LOOK AT NEXT WINDOW

```

1581
1582          01 003A3
1583
1584 01 003A3 32100000 X
1585 01 003A4 72720000 N
1586 01 003A5 25700008 A
1587 01 003A6 20700000 N
1588 01 003A7 25700000 N
1589 01 003A8 72220000 X
1590 01 003A9 6A4003FE
1591 01 003AA 21000080 A
1592 01 003AB 684003AE
1593 01 003AC 72220000 X
1594 01 003AD 680003B8
1595          01 003AE
1596 01 003AE 72220000 X
1597 01 003AF 6A4003FE
1598 01 003B0 72220000 X
1599 01 003B1 6A4003FE
1600 01 003B2 72220000 X
1601 01 003B3 21000040 A
1602 01 003B4 694003B8
1603 01 003B5 72220000 X
1604 01 003B6 693003B8
1605 01 003B7 72220000 X
1606          01 003B8
1607 01 003B8 6A4003FE
1608
1609 01 003B9 72220000 N
1610 01 003BA 25200008 N
1611 01 003BB 72440000 N
1612 01 003BC 72540000 N
1613 01 003BD 25200000 N
1614 01 003BE 72640000 X
1615 01 003BF 22700004 A
1616 01 003C0 6A100000 X
1617 01 003C1 32100000 X
    
```

```

PAGE
SIU12 EQU *
* SET UP PROCESSOR'S PP IN CMAP
LW,1 SIISUN
LOAD,7 UX:JIT,1
SLS,7 11
AI,7 JXCMP
SLS,7 :BIG
LB,2 UB:BV,1
BAL,4 SPMAP
CI,0 TIC
BAZ SIU121
LB,2 UB:ACP,1
B SIP11
SIU121 EQU *
LB,2 UB:APR,1
BAL,4 SPMAP
LB,2 UB:APB,1
BAL,4 SPMAP
LB,2 UB:DB,1
CI,0 DIC
BANZ SIP11
LB,2 UB:ASP,1
BNEZ SIP11
LB,2 UB:DB,1
SIP11 EQU *
BAL,4 SPMAP
* SET UBIFLG APROPOS & RESET SWAP IN PRBG FLG
LOAD,2 UX:JIT,1
SLS,2 11:BIG
LOAD,4 JPPH,2
LOAD,5 JPPT,2
SLS,2 IBIG
LB,6 JPPC,2
LI,7 4
BAL,1 T:PGCHK
LW,1 SIISUN
    
```

SHIFT TO BYTE RESOLUTION
R7 IS USED IN SPMAP BELOW
SHIFT TO BYTE OR HALFWORD RES

GET COMMAND PROC #

NAP OR SHIFT TO BA(JIT)

NO1 12:24 SEP 08, 175

1618 01 003C2 52920000 X
 1619 01 003C3 4990000A N
 01 003C4 4890000C N
 1621 01 003C5 4890000E N
 1622 01 003C6 20900001 A
 01 003C7 55920000 X
 1623 01 003C8 72320000 X
 1624 01 003C9 72520000 X
 1625 01 003CA 31500000 X
 1626 01 003CB 492003D0
 1627 01 003CC 33100000 X
 1628 01 003CD 492003D0
 1629 01 003CE 22B00001 A
 1630 01 003CF 35B00000 X
 1631 01 003D0
 1632 01 003D0 32B00000 X
 1633 01 003D1 483003D4
 1634
 1635
 1636
 1637
 1638
 1639
 1640
 1641
 1642
 1643 01 003D2 0F000000 X
 01 003D3 001B0000 A
 1644 01 003D4
 1645 01 003D4 22B00000 A
 1646 01 003D5 35B00000 X
 1647 01 003D6 35B00000 X
 1648 00000000
 1649 *S*
 1650 *S*
 1651

SIU13

EQU *
 LSETR JIC,9,1
 RSETS INIT,9
 RSETS DCBS,9
 SETST RTR,9,1
 LB,3 UBIUS,1
 LB,5 UBIPRI0,1
 CW,5 S;CUP
 BG SIU132
 MTW,1 S;HIR
 BG SIU132
 LI,11 1
 STW,11 S;HIR

GET HIS PRIORITY
 IS HE HIGHER THAN CURRENT USER
 NO
 COUNT FOR HIR DUDE
 ALREADY GOT SOME
 FORCE COUNT TO ONE

SIU132

RES 0
 LW,11 SIFPPC
 BE SIU135

S SCREECH CODE: 1B
 S REPORTED BY: SWAPPER
 S MESSAGE: SWAPPER PAGE CHAIN COUNT IS NON ZERO AFTER
 S INSWAP.
 S REGISTERS: 1-IN SWAP USER NUMBER
 S 2-PHYSICAL BYTE ADDRESS OF THE INSWAP JIT
 S REMARKS: ALL OF THE PAGES ALLOCATED TO THE INSWAP WERE NOT
 S DOLED OUT. THERE IS PROBABLY SOMETHING WRONG
 S WITH THE INSWAP JIT OR SWAPPER PAGE CHAIN.

SCREECH X11B:

SIU135

RES 0
 LI,11 0
 STW,11 SIFPPH
 STW,11 S;FPPT
 DB DEBUG
 LI,1 9
 BAL,0 RECORD
 FIN

W01 12:24 SEP 08 1975
 1652 01 003D7 22E00002 A
 1653 01 003D8 22000000 A
 1654 01 003D9 35000000 X
 1655 00000001
 1656 01 003DA 22400000 N
 1657 01 003DB 72100000 X
 1658 01 003DC 22500000 N
 1659 01 003DD 68000000 X
 1661
 1662 *S*
 1663

SIU14

LI,14 2
 EQU \$
 LI,0 0
 STW,0 \$IISUN
 DB PERFORM
 LI,4 TSS2
 LB,1 SBIBSN
 LI,5 CHIDBS
 B T;DSTRB
 ELSE
 B TSS2
 FIN

GO REENTER SS

```

1664
1665
1666
1667
1668      01 003DE
1669 01 003DE 72200000 X
1670 01 003DF 683003E4
1671
1672      01 003E0
1673 01 003E0 72340000 X
1674 01 003E1 72840000 N
1675 01 003E2 75860000 N
1676 01 003E3 642003E0
1677
1678      01 003E4
1679 01 003E4 32400000 X
1680 01 003E5 72680000 N
1681 01 003E6 32700006 A
1682 01 003E7 25700009 A
1683 01 003E8 22500000 N
1684 01 003E9 25500000 N
1685 01 003EA 20500000 N
1686 01 003EB F56A0007 N
1687 01 003EC 2570007F A
1688 01 003ED 12C00000 X
1689 01 003EE 15CE0000 F
1690 01 003EF 52880000 X
1691 01 003F0 21800200 A
1692 01 003F1 694003F7
1693 01 003F2 33F00000 X
1694 01 003F3 32300000 X
1695 01 003F4 683003F7
1696 01 003F5 6A200000 X
1697 01 003F6 33F00000 X
1698      01 003F7
1699 01 003F7 32500000 X
1700
    
```

```

PAGE
*
* ERROR READING JIT
*
S:JITERR EQU $
LB,2 SB:NP
BEZ JE4
*
JE3 EQU $
LB,3 SB:PN,2
LOAD,8 SX:HPP,2
STORE,8 PX:HPP,3
BDR,2 JE3
*
JE4 EQU $
LW,4 SI:SUN
LOAD,6 UX:JIT,4
LW,7 6
SLS,7 9
LI,5 JXC:MAP
SLS,5 =IBIG
AI,5 JJITVP
STORE,6 *7,5
SLS,7 =1
LD,12 OPSD
STD,12 TSTACK=JJIT,7
LW,8 UH:FLG,4
CI,8 JITIC
BANZ JE45
MTW,=1 S:FPPC
LW,3 SIAJP
BEZ JE45
BAL,2 T:FPP
MTW,=1 S:FPPC
*
JE45 EQU $
LW,5 S:FPPH
    
```

DISP TO JIT PG IN CMAP
 SET PHY PG IN CMAP

INIT TSTACK

TAKE JIT OUT OF COUNT
 PHY PG # OF AJIT

RELEASE IT
 AND TAKE IT OUT OF COUNT

RL:

H01 12124 SEP 08, 175

56

1701 01 003F8
1702 01 003F8 32300005 A
1703 01 003F9 683003C2
1704 01 003FA 72560000 N
1705 01 003FB 33F00000 X
1706 01 003FC 222003F8
1707 01 003FD 68000000 X
1708

JE5

EQU \$
LW,3 5
BEZ SIU13
LOAD,5 MX:PPUT,3
MTW,=1 S:PPC
LI,2 JE5
B TIFPP

RELEASE USER PAGES

*

```

1709
1710
1711      01 003FE
1712 01 003FE 68300408
1713 01 003FF 72640000 X
1714 01 00400 30600007 A
1715 01 00401 72340000 N
1716 01 00402 72840000 X
1717 01 00403 68000405
1718      01 00404
1719 01 00404 72360000 N
1720 01 00405 753C0000 N
1721 01 00406 20600001 A
1722 01 00407 64800404
1723 01 00408 68080000 A
    
```

```

PAGE
* SET PRBC S CORE MAP
SPMAP EQU $
      BEZ SPMAP20
      LB,6 PB1PVA,2
      AW,6 7
      LOAD,3 PX1HPP,2
      LB,8 PB1P8Z,2
      B $*2
SPMAP2 EQU $
      LOAD,3 MX1PPUT,3
      STORE,3 0,6
      AI,6 1
      BDR,8 SPMAP2
SPMAP20 B 0,4
    
```

```

1724
1725
1726
1727
1728      01 00409
1729      01 00409 21002800 A
1730      01 0040A F840000C A
1731      01 0040B 02200020 A
           01 0040C 0BC00000 N
1732      01 0040D 55EE0000 A
1733      01 0040E 32C00004 A
1734      01 0040F 25C0000B A
1735      01 00410 30C00000 X
1736      01 00411 B5C00000 X
1737      01 00412 33E00000 X
1738      01 00413 32C00007 A
1739      01 00414 25C00001 A
1740      01 00415 30C00000 X
1741      01 00416 B5C00000 X
1742
1743      01 00417 207FFFFFF A
1744      01 00418 33E00000 X
1745      01 00419 72C00000 X
1746      01 0041A 4BC0000E A
1747      01 0041B 20CFFFFFFE A
1748      01 0041C 4810041E
1749      01 0041D 38E00000 X
1750      01 0041E 20EFFFFFFE A
1751      01 0041F 02200020 A
           01 00420 0AC00000 N
1752      01 00421 F800000C A
1753
    
```

PAGE
 * BUILD COMMAND LIST USED TO SWAP IN INIT AND DCBS
 *

*
 *
 * S\$INIT

```

EQU      $
CI,0     INIT+DCBS
BAZ      *12
PUSH     2,12

STH,14   0,7
LW,12    4
SLS,12   11
AW,12    Y02
STW,12   *TSC3
MTW,=2   TSC3
LW,12    7
SLS,12   1
AW,12    Y03
STW,12   *TSC3
REF      Y03
AI,7     =1
MTW,=2   TSC3
LB,12    MBIGAM6
AND,12   14
AI,12    =2
BGEZ     $*2
SW,14    M:ADRINCR
AI,14    =2
PULL     2,12

B        *12
    
```

SET DA

READ COMMAND

SEEK BRDR

*

1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790

01 00422
01 00422 21800001 A
01 00423 6830045B
01 00424 32700008 A
01 00425 207FFFFFFF A
01 00426
01 00426 32AE0001 N
01 00427 5570000A A
01 00428 35AE0001 N
01 00429 64700426
01 0042A 3220000E A
01 0042B 72B40000 X
01 0042C 22200000 A
01 0042D

PAGE

* BSAC *
* * *
* ORDER *
* SBRT *
* AND *
* CHAIN *

* I 8 = # OF USERS WHEN 2ND INDEX*(8) THEN ORDERING DONE
* 7 = ORDER # INSERTED IN 1ST HALF OF BDA
* 1 = 1ST INDEX
* 2 = 2ND INDEX
* 9 = SAVES 2ND INDEX TO CONTINUE RIPPLE UP AFTER RIPPLE DOWN
* 10 = PRESENT BDA BEING USED FOR COMPARISON
* 11 = SECTOR MASK USED FOR COMPARE SELECTIVE
* 12 = ACC TO RIPPLE A BDA UP WHEN RIPPLING DOWN
* 15 = SUBROUTINE LINK
*
* FIRST ORDERING OF LIST
BSAC EQU \$
CI,8 1
BE BUT,1
* GENERATE ORDER LIST INDEXES
LW,7 8 GET NUMBER OF USERS
AI,7 =1
F01 EQU \$
LW,10 \$IBDA+1,7
STH,7 10 PUT INDEX INTO 1ST HALF OF BDA WORD
STW,10 \$;BDA+1,7 PUT BDA BACK
BDR,7 F01 CNT
* GET NEXT ENTRY SET
LW,2 14
LB,11 MBIGAM6,2 SECTOR MASK
LI,2 0
F02 EQU \$

GET NUMBER OF USERS
PUT INDEX INTO 1ST HALF OF BDA WORD
PUT BDA BACK
CNT
SECTOR MASK

HO1 12:24 SEP 08, '75

1791	01	0042D	32100002	A
1792	01	0042E	20200001	A
1793	01	0042F	31200008	A
1794	01	00430	68300443	
1795	01	00431	32A20001	N
1796				
1797	01	00432	45A40001	N
1798	01	00433	6910042D	
1799				
1800	01	00434	32900002	A
1801	01	00435	46A40001	N
1802		01 00436		
1803	01	00436	201FFFFFF	A
1804	01	00437	6910043E	
1805	01	00438	202FFFFFF	A
1806	01	00439	45A20001	N
1807	01	0043A	69200440	
1808	01	0043B	32C20001	N
1809	01	0043C	35C40001	N
1810	01	0043D	68000436	
1811				
1812		01 0043E		
1813	01	0043E	35A00001	N
1814	01	0043F	68000441	
1815		01 00440		
1816	01	00440	35A40001	N
1817	01	00441	32200009	A
1818	01	00442	6800042D	

	LW,1	2
	AI,2	1
	CW,2	8
	BF	CCL0
	LW,10	S;BDA+1,1
* MAKE RIPPLE UP	COMPARISON	
	CS,10	S;BDA+1,2
	BL	F02
* RIPPLE BACK DOWN		
	LW,9	2
	XW,10	S;BDA+1,2
RBD1	EQU	\$
	AI,1	=1
	BLZ	RBDF1
	AI,2	=1
	CS,10	S;BDA+1,1
	BG	RBDF2
	LW,12	S;BDA+1,1
	STW,12	S;BDA+1,2
	B	RBD1
* RIPPLE DOWN COMPLETE = RESTORE RIPPLE UP POINTER & CONT		
RBD1	EQU	\$
	STW,10	S;BDA+1
	B	#+2
RBD2	EQU	\$
	STW,10	S;BDA+1,2
	LW,2	9
	B	F02

UPDATE 1ST TO 2ND
 ! 2ND
 2ND INDEX = N
 GET A BDA
 COMPARE WITH NEXT
 LESS = IN ORDER = CONT
 SAVE 2ND INDEX
 PULL LESSER BDA, PUT GREATER BDA UP
 FINISH RIPPLE DOWN
 YES = RETURN TO RIPPLE UP
 ! 2ND !
 ONE FOUND, STILL OUT OF ORDER
 NO = RETURN RIPPLE UP
 RIPPLE BDA
 CONT RIPPLE BACK DOWN
 RIPPLE UP POINTER & CONT
 INSERT 1 OUT OF ORDER IN 1ST ENTRY
 INSERT ONE OUT OF ORDER IN
 RESTORE 2ND INDEX
 CONT RIPPLE UP

H01 12:24 SEP 08, 175

1856 01 00452 4B100000 X
 1857 01 00453 32A20001 N
 1858 01 00454 25A0007F A
 1859 01 00455 30A00000 X
 1860 01 00456 32580001 N
 1861 01 00457 35AA0000 A
 1862 01 00458 22A00080 A
 1863 01 00459 30A00004 A
 1864 01 0045A 75A00001 N
 1865 01 0045B EQU \$
 1866 01 0045B FB00000F A B *15
 1867 * SET UP FOR NEXT SEARCH
 1868 CCL2 EQU \$
 1869 01 0045C 32100002 A LW,1 2
 1870 01 0045D 32300004 A LW,3 4
 1871 01 0045E 20400001 A AI,4 1
 1872 01 0045F 52A80000 X LW,10 SH:EDA,4
 1873 01 00460 3220000E A LW,2 14
 1874 01 00461 72240000 X LB,2 MB:GAM6,2
 1875 01 00462 4BA00002 A AND,10 2
 1876 * INITIALIZE SEARCH INDEX
 1877 01 00463 EQU \$
 1878 01 00463 22200000 A LI,2 0
 1879 * COMPARE NEXT ENTRY
 1880 CCL6 EQU \$
 1881 01 00464 20200001 A AI,2 1
 1882 01 00465 4AC40001 N LS,12 S:BD A+1,2
 1883 01 00466 31A0000C A CW,10 12
 1884 01 00467 6910046C BL CCF
 1885 * IF FIRST SEARCH DONE, SET XIC01 IN 1ST BYTE OF EDA
 1886 01 00468 31200006 A CW,2 6
 1887 01 00469 69300464 BNE CCL6
 1888 01 0046A 30A00000 X AW,10 COMSK
 1889 01 0046B 68000463 B CCL4
 1890 * FOUND CHAIN, GET INDEX TO OTHERS & SET TIC
 1891 01 0046C EQU \$
 1892 01 0046C 32440001 N LW,4 S:BD A+1,2

AND,1 M8
 LW,10 S:BDCL+1,1
 SLS,10 =1
 AW,10 Y08
 LW,5 S:ECL+1,4
 STW,10 0,5
 LI,10 X:801
 AW,10 4
 STB,10 S:BD A+1
 EQU \$
 B *15
 * SET UP FOR NEXT SEARCH
 CCL2 EQU \$
 LW,1 2
 LW,3 4
 AI,4 1
 LW,10 SH:EDA,4
 LW,2 14
 LB,2 MB:GAM6,2
 AND,10 2
 * INITIALIZE SEARCH INDEX
 CCL4 EQU \$
 LI,2 0
 * COMPARE NEXT ENTRY
 CCL6 EQU \$
 AI,2 1
 LS,12 S:BD A+1,2
 CW,10 12
 BL CCF
 * IF FIRST SEARCH DONE, SET XIC01 IN 1ST BYTE OF EDA
 CW,2 6
 BNE CCL6
 AW,10 COMSK
 B CCL4
 * FOUND CHAIN, GET INDEX TO OTHERS & SET TIC
 CCF EQU \$
 LW,4 S:BD A+1,2

AND F&BL OUT
 GET BCL FOR 1ST ENTRY
 FIND END OF CHAIN WHERE TIC BELONGS
 PUT TIC IN CL
 GET FLG
 ADD BACK LINK
 SET UP F&BL INTO 1ST RDA
 UPDATE PRES TO SEARCH INDEX
 UPDATE PRES INDEX TO TABLES
 PRES EDA
 INIT SEARCH INDEX
 INCR SEARCH INDEX
 GET F&BL AND BDA
 EDA LESS THAN BDA
 YES = FOUND
 FINISHED 1ST TIME THRU
 NO
 START AT BEG OF LIST
 & SET TIC
 GET INDEX TO OTHER TABLES

H01 12124 SEP 08, 175

1893	01	0046D	52400004	A
1894	01	0046E	32A80001	N
1895	01	0046F	25A0007F	A
1896	01	00470	30A00000	X
1897	01	00471	32560001	N
1898	01	00472	35AA0000	A
1899	01	00473	6800044C	

LW,4	*
LW,10	SIBCL+1,4
SLS,10	*1
AW,10	Y08
LW,5	SIECL+1,3
STW,10	0,5
B	CCL1

FRBM BDA

PUT TIC TO (ECL)


```

1900
1901
1902
1903
1904
1905
1906
1907
1908
1909      01 00474
1910
1911      01 00474      32700000 X
1912      01 00475      3220000E A
1913      01 00476      72B40000 X
1914      01 00477      20A00004 A
1915      01 00478      45A40000 X
1916      01 00479      6820047B
1917      01 0047A      30A40000 X
1918      01 0047B      0047B
1919      01 0047B      45AE0000 X
1920      01 0047C      6910047F
1921      01 0047D      6470047B
1922      01 0047E      22700001 A
1923      01 0047F      0047F
1924      01 0047F      32CE0000 X
1925      01 00480      7230000C A
1926      01 00481      4B300000 X
1927      01 00482      5240000C A
1928      01 00483      4B400000 X
1929      01 00484      32560001 N
1930      01 00485      32680001 N
1931      01 00486      E800000F A
    
```

```

PAGE
*****
*      ULCLC      *
*      *      *
*      UNLINK CL CHAIN      *
*****
*      I      10 = PRESENT SECTOR POSITION
*      0      5 = POINTS TO END OF CHAIN
*      0      6 = POINTS TO BEG OF CHAIN
ULCLC      EQU      $
*      UNLINK CL CHAIN
      LW,7      TSC4      USER CT
      LW,2      14
      LB,11     MBIGAM6,2
      AI,10     SDLAY
      CS,10     M;GASLIM,2
      BLE      $+2
      AW,10     MIADRINCR,2
UC2      EQU      $
      CS,10     S;BDA,7      IS PRES SECT POSIT < THIS BDA
      BL      UC4      YES-FOUND
      BDR,7     UC2
      LI,7      1      NONE LESS = GET FIRST
UC4      EQU      $
      LW,12     S;BDA,7      GET F&BL, INDEX AND BDA
      LB,3      12      GET F&BL = FOR HALT
      AND,3     M7      GET RID OF FLAG, KEEP BACK LINK
      LH,4      12      GET INDEX TO OTHER TABLES = FOR SIO
      AND,4     M8      GET INDEX ONLY
      LW,5      S;ECL+1,3    POINTS TO HALT NEEDED
      LW,6      S;BCL+1,4    ADR OF CL FOR SIO
      B      +15
    
```

H01 12124 SEP 08, 175

65

1932				PAGE		
1933	01	00487	72DA0000 X	SADRINC LB,13	MBIGAM6,5	MAX SECTOR MASK
1934	01	00488	20C00002 A	AI,12	2	INC SECTOR ADDRESS
1935	01	00489	45CA0000 X	CS,12	MIGASLIM,5	MAX ALLOWABLE SECTOR #.
1936	01	0048A	F8200000 A	BLE	*0	
1937	01	0048B	30CA0000 X	AW,12	MADRINCR,5	INCREMENT TO NEXT BAND.
1938	01	0048C	F8000000 A	B	*0	

HO1 12:24 SEP 08, 1975

1939
1940
1941
1942
1943

EXT
EXT

SIO
SENSE

PAGE
REF
EQU
EQU
END

T:SI0,T:SENSE
T:SI0
T:SENSE

01 0048D 00000018 A

CONTROL SECTION SUMMARY; 01 0048E PT 0

H01 12:24 SEP 08, '75
* SYMBOL VALUES

67

AJFLGS/0000002f
BYTE/00000001
CCL1/01 0044C
DCBPRBC/00000000
DIC/00000040
HFLGS/0000004C
JE45/01 003F7
JITIC/00000200
MBNPRBC/00000000
OSAC/01 00422
PERFORM/00000001
PPSWAP/00000010
RBD2/01 00440
RML0AD/01 0020F
R10/0000000A
R14/0000000E
R4/00000004
R8/00000008
SB10SUL/EXT
SIJ4/01 0015B
SI01/01 002E4
SIP4/01 00177
SIP94/01 002B5
SIU10A/01 00363
SIU12/01 003A3
SIU135/01 003D4
SIU35/01 00309
SIU7/01 0031E
SIU9/01 0034D
SI10/01 00118
SI101/01 0021D
SI102A/01 001c7
SI1055/01 00246
SI127/01 0027D
SI2/01 00145
SI6/01 0018E

ANSPRBC/00000000
CCF/01 0046c
CCL2/01 0045C
DCBS/00002000
DISCBPRBC/00000000
INIT/00000800
JE5/01 003F8
LIMINF0/01 001F9
MPBITS/00000000
OUT1/01 0045B
PGCHKM1/01 0001A
PPSWP/00000010
RBD1/01 00436
RMST0/01 00211
R11/0000000B
R15/0000000F
R8/00000005
R9/00000009
SDLAY/00000004
SIJ41/01 00166
SIP1/01 00124
SIP5/01 00250
SIU0/01 002B5
SIU11A/01 0037B
SIU121/01 003AE
SIU14/01 003D8
SIU4/01 0030D
SIU75/01 00324
SI0/01 000FF
SI10/01 001B5
SI101G/01 0021F
SI102B/01 001cA
SI1057/01 00248
SI128/01 0027E
SI35/01 00150
SI65/01 0019A

BAT/00000100
CCI/01 00000
CCL4/01 00463
DEBUG/00000000
F01/01 00426
JE3/01 003E0
JFLGS/0000004C
LIML0AD/01 001ED
MV0SUL/01 0000D
0VSTART/01 00004
PGCHK5/01 0001g
PSTST/01 00204
RDLOAD/01 00214
RTR/00000001
R12/0000000C
R2/00000002
R6/00000006
S\$INIT/01 00409
SENSE/EXT
SIJ6/01 0016A
SIP11/01 003B8
SIP6/01 00253
SIU1/01 002EA
SIU11B/01 003g0
SIU13/01 003c2
SIU2/01 002F6
SIU5/01 00312
SIU76/01 00335
SI1/01 00105
SI10GCK/01 00105
SI101L/01 0023D
SI105/01 0024c
SI12/01 0025A
SI129/01 0028A
SI4/01 001g0
SI7/01 0019E

BITS/00000000
CCL0/01 00443
CCL6/01 00464
DELA/00000400
F02/01 0042D
JE4/01 003E4
JIC/00000200
LIMST0RE/01 001F6
0S0CENT/01 0001F
PDISK/01 00206
PM0N0FF/00000001
RBD1/01 0043E
RDST0/01 00217
R1/00000001
R13/0000000D
R3/00000003
R7/00000007
SIJITERR/01 003DE
SHSDA/01 00005
SI0/EXT
SIP3/01 0012D
SIP9/01 002A9
SIU10/01 00356
SIU11C/01 003A2
SIU132/01 003D0
SIU3/01 002FC
SIU6/01 00319
SIU8/01 0033A
SI1A/01 0010E
SI10NG/01 0021A
SI102/01 001CF
SI105GNK/01 001E6
SI125/01 0026D
SI13/01 00293
SI5/01 00189
SI75/01 001A2

H01 12:24 SEP 08, 175

SI8/01 001A7
SMST8/01 001F3
S8J1/01 000AD
S8J6/01 000DB
S82/01 00039
S825/01 00054
SPEC/00000003
TDISK/01 00208
UC4/01 0047F
WORD/00000000

SI9/01 001AA
S8CK1/01 0009E
S8J2/01 00088
S80/01 00022
S822/01 00046
S84/01 0007A
SPMAP2/01 00404
TIC/00000080
UFLAGS/00000001
!9AJCL/01 00006

SJAC/00001000
S8CK2/01 000A1
S8J4/01 000BD
S80A/01 00025
S823/01 00048
S88/01 000E7
SPMAP28/01 00408
TSTST/01 00208
ULCLC/01 00474
*SWAP*DEV/EXT

SML8AD/01 001F0
S8CK8/01 000A9
S8J5/01 000CA
S80B/01 00028
S824/01 00052
S89/01 000EE
S69PR8C/00000001
UC2/01 0047B
UTSPR8C/00000001

* EXTERNAL DEFINITIONS

LOG8FF/01 00002
SWAP8UT/01 0000A

SADRINC/01 00487
SWAPPER/01 00000

SPMAP/01 003FE

SWAPIN/01 000FF

* PRIMARY REFERENCES

ALL88UT
COMSK
GJ8B
JITIT
JBIMAX
JBUPVP
JCMAP
JPCP
JVLCS
M:HLTIC
MUP8
NPMC
PGCHKM
RCVPSD
SIECL
SIISUN
SBINP
SBIRGDF
SHIJAUDA
SVILIM
TIPGCHK
TMPDCK
TSS2

ATITLE
DAJ
HANSWAPS
JINRS
JBINFP88L
JCLL
JDA
JPPC
JVLH
M:UC
MX:PPUT
ONLN
PHIDDA
S:SWAP*DEV
SIFPPC
SIJCL
SB:8SN
SBIRGMX
SH:SDA
SVIRSIZ
T:RE
TMPDPPK
TSTACK

BANPMC
DID:8
ISWAP*MEAS1
JIPUF
JB:PNR
JCL
JH:DA
JPPH
JVL
MAXG
M21
8SWAP*MEAS
PH:DA
S:AJP
SIFPPH
SIJSP
SB:8SUL
SB:R8DF
SL:8DF
SX:HPP
T:SENSE
TSCO
UB:ACP

BT31T80
E:UGFAC
ISWAP*MEAS2
JITITLE
JBIPRIV
JCLE
JH:LDCE
JPPY
JXBUFVP
MB:GAM6
M24
PB:DCBSZ
PRDCRM
SIBCL
SIFPPT
SIFCT
SB:8SULT
SB:R8MX
SL:8MX
TIDELUS
TISEXIT
TSC1
UB:AP8

CH:88S
FPMC
JIACCN
JAJ
JBNASP
JCLP
JJITVP
JPUF
JXCMP
MDP8
M7
PB:DSZ
PRDPRM
SIBDA
SIGJ8BACN
SISCL
SB:PNL
SCL*END
SL:8MX
TIDSTRB
TIS:8
TSC2
UB:APR

CHNMSK
GETJIT
JIASSIGN
JAJITVP
JBNRG
JCLPA
JLMAP
J8BUF1VP
MIADRINC
MP8
M8
PB:PSZ
PX:HPP
SIBECL
SIGJ8BTBL
SISYMDB
SB:R8DF
SDA*END
SL:8DF
TIFPP
TMDCRM
TSC3
UB:ASP

CURNTIM
GIVEUP
JICCBUF
JB:CUR
JBPC
JCLT
J8VVP
J8PBFLG
M:GASLIM
MPP8
NB31T80
PB:PV
PX:TPP
SICUP
S:HIR
SB:GJ8BU
SB:R8MX
SH:EDA
SL:8MX
T:8V
TMDPRM
TSC4
UB:BL

H01 12124 SEP 08, '75

UB1DB UB1DV
UH:FLG UH:FLG2
X7 X8000
Y08 Y8

UB:PCY
UH:JIT
YFF
OPSD

UB:PRI0
UX:JIT
Y008
IBIG

UB:SWAP1
XFFF800
Y01

UB:US
X3FFE00
Y02

UH:AJIT⁶⁹
X4
Y03

- * SECONDARY REFERENCES
LBIUN
- * NO UNDEFINED SYMBOLS
- * ERROR SEVERITY LEVEL: 0
- * NO ERROR LINES