			;***COPYRIGH	T 1969, DIGITAL	EQUIPMENT CORP., MAYNARD, MASS.***
			;THIS SUR-PR XLIS LIST	T	WITH SYSTEM PARAMETER FILE - S.MAC(V414)
			;THIS SUB-PR V003)	OGRAM ASSEMBLED	WITH CONFIGURATION DEPENDENT FEATURE SWITCHES - FT50SH
			×LIS LIST		
			IFE FTDISK+F TITLF SCHE	TRC12+2,< DB - SCHEDULING	ALGORITHM FOR SWAPPING SYSTEM(10/50)(BURROUGHS DISK)
			IFE FTDISK+F	TRC12+1,<	
			TITLE SCHE	DD - SCHEDULING	ALGORITHM FOR SWAPPING SYSTEM(10/50)(DATA PR^DUCTS DI
			SUBITL CLKC XP VSCHED,42	<u>1</u> •	TH/RCC TS 02 JUNE 69 V421
				;PUT VERSION	N NUMBER IN GLOB LISTING AND LOADER STORAGE MAP
			INTERNAL	I NEW HOH-IK	IS SOURCE FILE MAY BE ASSEMBLED TO USE EITHER THE D DISK (MODEL RC-10) OR THE OLD PDP-6 DISK (DATA DISK FILE) FOR SWAPPING.
			IFN FTRC:	0, <	
000000			INTERNAL	D ;THIS SYMBOL : BUILDER TO XCKCSW	. IS SOLELY TO PERMIT SYSTEM D RETRIEVE THE CORRECT BINARY FILE,
			> IFE FTRC1 ENTRY XCKCS >		
			EXTERNAL JOB, External jrtg External pjbs		
			INTERNAL NXTJ	OB,FTSWAP	
		000006 000002 000004	QJ=DEVCAT	DEFINITIONS ;QJOB WORD ;STATUS WORD ;JOB NO.	
			;INITIALIZE S ; DEVIC	CHEDULER (CALL) ES ARE INITIALI	ED FROM SYSINI BEFORE ALL OTHER 7ED)
			INTERNAL NXTI	NI	
000001	402001	000011 000244 000275		TAC,MAXQ AVALTB(TAC) REQTAB(TAC)	IMAX, NO, OF QUEUES ICLEAR SHARARLE DEVICE AVAIL, FLAGS ISET SHARABLE DEVICE REQUEST COUNT I TO -1, I.F. NO JOB WAITING OR

57						1 US
58	200023	365040	000001'	SOUGE	TAC, -2	
59	000004	402000	0003221	SETZM	0 J O B	I CLE
60	002025	402000	0003231	SETZM	XJOB	1CLE
61	000076	402000	000321'	SETZM	TOBONE	;CLE

I USING DEVICE OTHER THAN INITIALIZATION

ICLEAR	N9.	OF	JOBS	NEEDING	REQUEING
ICLEAR	ΝΟ,	0F	JORS	NEEDING	EXPANDING
;CLEAR	JUB	N0.	, TO B	E REQUE	JEn

SCHEDB - SCHEDULING ALGORITHM FOR SWAPPING SYSTEM(12/57)(RURHOUGHS DISK) CLKCSW R.KRASIN/AF/IH/RCC IS 02 JUNE 69 V421

MACRO.V36 19:10 4-JUN-69 PAGE 13-1

.

42 000007 263140 000200 POPJ PDP, 63

4L 0	- SCHEDU	R.KRASIN	VAFZTHZ F	OR SWAPP CC IS	ING SYSTEM(10/53)(BUR⊡ 02 JUNE 69 ∋V421	011GPS DISK) MACRO.V36 19:10 4-JUN-6	9 PAGE
54				NXT.IOP	PECREMENTS CURRENT IN	B'S QUANT, AND PROTECT	
45				TIMES	AND REQUETES IT IF QUA	NT. TIME GONE TO Ø.	
66						EQUESTED AT OTHER PRIORITY	
47					THEN CALLS SHUFFLER, S		
× 8					NO ASSUMPTIONS RE. ACS		
€9				RETURN	S NEXT JOE TO RON IN J		
7 Ø							
71							
72				EXTERNAL	_ JBISWP,POILSI		
73			_				
74			600000	NXTJOH:	SKIPN TIMEF	ICLOCK TIC?	
			0000221		JRST NXTJH1	; NO	
		205200			MOVSI J,MJOBN	IYES	
	PØ2013	205170			MCVSI SW1	IDECREMENT IN CORE PROTECT TIME	
		205240			MOVSI DAT, SWP		
		612244			TONE DAT, JETSTS(J)		
		331024	022021'		JRST .+3		
-			002030 003317'		SKIPL JBTSWP(J) ADDM SW:JBTSWP(J)		
83	600021	253220	000015		408JN J,4		
84	A DEECT	200210	N00015.		AUSUN J, =4		
85	202022	336020	C 2 7 2 7 7	NYT ID1.	SKIPN J.J08		-
			220041	W. DET.	JRST CKJP1	ICURRENT JOB NO., IS IT NULL JOB?	
	000024	2013-0			MOVEL OF'S	IYES, GO SEE IF OTHER JOBS NEED RESCHEDUL	ING
	002025		007215		HLAS SWIJSTSTS(J)	JGET READY IN CASE CURRENT JOB UNRUNABLE	
	000026	620100			TRZ SWIRLNMSK+CMWR	JMASK OUT DO NOT CARE BITS	
	000027		440020		CALE SWIRUNARLE	IS CURRENT JOB RUNABLE?	
	700030		000051		JRST CKJR3	INO, REQUE CURRENT JOB	
	000031		000010		SKIPN TIMEF	INC, IS THIS A TIME INTERRUPT?	
93	000032	254000	000041		JRST CKJR1	IND.	
94						# 1812 #	
95	100033	370104	000025		SOS SW.JRTSTS(J)	IDECREMENT QUANT. TIME	
96	000034	-	777777		TRNE SW,-1	THAS TIME GONE TO 0?	
97	000035	254000	000041'		JRST CKJB1	INO	
	000036	201240	0004021		MOVEL DAT, OTIME	IYESREQUEUE AND RESET QUANT. TIME	
99	000037		000321'		MOVE TAC, JOBQUE	C LEGEL NEW NEGEL NEWNING FILLE	
100	ØØØØ4Ø	260140	000162		PUSHJ PDP, OXFER		

SCHEDB - SCHEDULING ALGORITHM FOR SWAPPING SYSTEM(10/50)(BURPOUGHS DISK) CLKCSW R.KRASIN/AF/TH/RCC TS 02 JUNE 69 V421

MACRO,V36 19:10 4-JUN-69 PAGE 15

	02.000						
101	202041	337370	0003221	CKJB1:	SKIPG	01.0109	ISET QUINON ZERO IF ANY REQUEUEING TO DO
102		254000	0000731	01.001	JRST	CKJB5	IND REQUEUEING NECESSARY
103		201220	000000		MOVET	J. JOBMAX	ISTART WITH HIGHEST JOB NUMBER ASSIGNED
174		205100		CKJB2:		SWIJRQ	JOB NEEDS REQUEUEING BIT
				UN J02 •			
125		616174	000033		TDNN	SW, JBTSTS(J)	ITHIS JOB?
106		367200	000045		SOJG	J1	IND, KEEP LOOKING
127	000047	323200	0000731		JUMPLE	J,CKJB5	;YES,LOOKED AT ALL JOBS?
108							; (MAY NOT FIND A KJOBED JOB IF HIGHEST
129							I GO DECR, COUNT QJOB ANYWAY)
110	000050	412104	0000451		ANDCAM	SW,JBTSTS(J)	IND, MARK THIS JOB AS DONE
111	000051	200174	ØØØØ50'	скјвз:	MOVE SW	,JBTSTS(J)	JOB STATUS WORD
112	000052	201240	000330'		MOVEI D	AT,QCMW	JASSUME COMMAND WAIT
113	000053	607120	200000		TLNN SW	, CMWB	IIS JOR IN COMMAND WAIT?
114	000054	254000	0000571		JRST CK	JR9	INO.
115	000055		002001		TINE SW	SWP+JXPN	IYES, IS JOB ON DISK, OR TRYING TO EXPAND?
116		254000	000067		JRST CK		IYES. PUT JOR IN COMMAND WAIT Q
117		325100		CK 1801		SW,CKJB4	IND, WAIT STATUS CODE DETERMINES NEW Q
118	000000		000000	00000	LDB SW,		IYES, GET RUEUE CODE.
119	000061		0000001		CAIN	SWIWSO	J*** EXPERIMENTAL ***
			000132		JRST	CKJB10	JANA EXPERIMENTAL ANA
120							
121		306100	000013		CAIN SW		CURRENT JOB GOING INTO TTY IO WAIT?
122	000064	563004	0010201		HEROS J	BTSWP(J)	IYES, SET IN CORE PROTECT TIME TO -1,
123							I SO HE CAN BE SWAPPED IMMEDIATELY IF SOMEONE
124							I ELSE WANTS TO BE SWAPPED IN
	P0P065					AT, QBITS(SW)	IGET ADDRESS OF TRANSFER TABLE
126	000066	201240				AT, OSTOP	IF RUN BIT WAS OFF
127	000067	260140	000162'	CKJB4A:		DP,QXFER	IREQUEUE THE JOB
128	ØØØØ70	322300	0001131		JUMPE Q	J,SCHFD	; IF FROM NXTJOB GO DIRECTLY TO SCHED
129							; I.F. CURRENT JOB NO LONGER RUNNABLE(IOW)
130							; BUT JRO WASN'T SET SO DON'T DECR QJOB
131	000071	373300	000322'		SOSLE	01.010B	IANY MORE JOBS TO REQUEUE?
132		367290	000044		SOJG	J, CKJB2	IYES, BUT LOOK AT EACH JOB ONLY ONCE PER CLOCK TICK
133						0,0002	
	000073	201330	000011	CK 195:	MOVETO	J.AVLNUM	JCK AVAL FLAGS FOR SHAR. DEVS.
135	000073					VALTB(QJ)	JELAG=0?
136	000075		000074	0,000.	SOJG QJ		JYES - TRY NEXT ONE
137	000076		000112'			QJ,CKJB7	INO - OR FINISHED?
138	000077		000307'			VLQTB(QJ)	INGGET 1ST JOB IN Q
139	000100			CKJB6A:			
140	000101		000111			J,CKJB8	IFINISHED Q? WAIT TILL SWAPPER BRINGS IN JOB
141	000102		000051'			T,JBTSTS(J)	IIS JOB IN CORE?
142	000103		ØØ2ØØØ		TLNE DA		
143	000104	254000	000100'		JRST CK	JB6A	ING, LOOK AT NEXT JOB IN THIS QUEUE
144							I TO SEE IF IN CORE.
145	000105		000307'		HRRZ DA	T,AVLQTB(OJ)	INOGET TRANS, TABLE ADDRESS
146	000106	402006	000244'		SETZM A	VALTB(QJ)	JCLEAR AVAL FLAG
147	000107	301300	000003		CAIL QJ	MINQ	ILESS THAN MIN, SHARABLE DEV. Q?
148	000110		000162			DP, QXFER	REQUEUE THE JOB AND PUT IT IN
149						. = .	I PROCESSOR & SO SCHEDULER WILL RUN IT
150	000111	367300	000074.	скјва:	SOJG DJ	CKJB6	CONTINUE IF ANY MORE FLAGS TO LOOK AT
				-			

							1	
SCHEDB						TEM(10/50)(8 69 V421	SURROU	IGHS DISK) MACRO.V35 19:10 4-JUN-69 PAGE 16
151	202112			CKJB7:				INDNEGO SHUFFLE AND SWAP
152				IFE FTS	WAPIC	EXTERNAL (CHKSHF	
153					PUSHJ	PDP,CHKSHF		
154				>				
155				IFN FTS				
156 157	202112	260140	0004651		PUSHJ	PDP,SWAP>		
158				·SCHEDU	1.4855		IFUES	ACCORDING TO SSCAN TABLE
159						N CORERETU		
160								
161	C02113	402000	80000Ø	SCHED:	SETZM 1	POTLST		ICLEAR POTENTIALLY LOST TIME FLAG
162	000114	201240	000430			DATISSCAN		JADDRESS OF SCAN TABLE
163	000115	265640	0002251			C.OSCAN		BEGIN SCAN
164	007116	254020	000130		JRST S			INC MORE JOBSRETURN NULLJOB
165		476020	0001131		SETOM I			SET POTENTIALLY LOST TIME FLAG FOR CLOCK1
166	707120	2003*4	0001021		MOVE Q.	J, JBTSTS(J)		IS THIS JOB SWAPPED OUT
167	-		-			J,SWP+SHF+J)		IMONITOR WAITING FOR 1/0 TO STOP, OR JOB EXPANDING CON
168	C00121	603330	006001					
169	<i>°</i> 00122	254002	0000000		JRST (TAC1)		IYESCONTINUE SCAN, JOB CANNOT BE RUN
170								
171	000123	576100	000321'		HLREM '	TAC1, JOBQUE		;YESSAVE ITS Q
172	202124	205320	000370		MOVSI (JJ,WTMASK		JCLEAR WAIT CODE
173	70P125	412324	0001201		ANDCAM	GJ, JBTSTS())	
174	000126		000117'		SETZM P	POTLST		ICLEAR POTENTIALLY LOST TIME AS A USER IS TO BE RUN
175	000127	263140	000000		POPJ PI			IRETURN
176	000130		000000	SCHD1:				IRETURN NULL JOB
177	000131	26314Ø	<i>@@@@@@</i>		POPJ PI	ЭР,		
178								
179								
180								
181								ING CHANGE TO PERMIT TTY-1/O-WAIT-SATISFIED JOBS ON
182) BOUNE	D JOBS IN CORE, R.CLEMENTS/D.PLUMER 9 MAY 68
183	000132			CKJB10:		INFLG		
184		331004	000064		SKIPL	JBTSWP(J)		
185		254020	000065		JRST	CKJB4B		
	000135	201240	000137		MOVEI	DAT, CKJBT		
187		254000	000067		JRST	CKJB4A		
	000137	400000		CKJBI:	EXP	EQFIX		
189	000140	000006 100000	777760		XWD	DUTTY, PO2	-	MAKE JOB(LPT) COMPETE WITH CPU BOUND JORS
140	000141	000000	000000	INFLG	ta.	; NUN-ZERO	MŁANS	AT LEAST ONE JOB ON DISK WAITING TO COME IN.

SCHEDB - SCHEDULING CLKCSW R.KR	ALGORITHM FOR SWAPPING SYSTEM(10/50)(BURROUGHS DISK) MACRO,V36 19:10 4-JUN-69 PAGE 17 ASIN/AF/TH/RCC TS 02 JUNE 69 V421
191	SURTEL OCSS R. FRASINZAF TS3,17 22 MAR 68 VØØØ
192	, THIS ROUTINE MUST BE ASSEMBLED WITH THE CONFIGURATION
193	
194	,TAPE TO DEFINE NUMBER OF JORS .This section contains 2 routines for Q manipulation
195	, AND NECESSARY TABLES FOR SPECIFING OPERATIONS PERFORMED
196	
197	, BY THEM.
198	CATERALL INCIDE UNTETE DIRECT
199	EXTERNAL IMGIN,JATSTS,JBTADR.PJBSTS INTERNAL OXFER,GSCAN,FTSWAP,FTDISK
200	INTERNAL WARERJUSSWARTTENSK
201	STORAGE:
202	, STURAGE: , Each o is a ring structured, foward and backward
2713 2714	, LINKEN SRING LIST, THF "FIRST" LINK IN A Q IS
201	, Q-HEARER POINTING TO THE FIRST AND LAST MEMBERS OF THE Q.
200	A NULL O HAS ONE LINK-THE O-HEADER ITSELF. THE LINKS MAKING
208	, UP THE OS ARE CONTAINED IN A TARLE (JBTO) WITH NEGATIVE
2018	INDICIES (ADDRESSES LESS THAN JETO) USED FOR Q-HEADERS AND
2.39	POSITIVE INDICIES USED FOR MEMBERS (JOBS), THUS ONLY ONE WORD
210	PFR LINK IS NECESSARYITS ADDRESS RELATIVE TO JBTQ GIVES THE
211	JCS NO. (OR Q NO. IF VEGATIVE) WHICH IT REPRESENTS WHILF
212	ITS CONTENTS CONTAINS THE LINKING POINTERS, THESE
212	POINTERS ARE ALSO INDICIES RELATIVE TO JETO RATHER THAN
214	APSOLUTE ADDRESSES-RH(LINK)=FOWARD POINTER;
215	LH(LIMK)=BACKWARD POINTER.
216	, A JOB IS ASSUMED TO BE IN NO MORE THAN ONE O AT A TIME, AND
217	, THE NULL JOR (JOB 0) DOES NOT APPEAR IN THE QS (1.E, JATO
218	, ITSELF IS THE G-HEATER FOR (0 0).
219	
220	, ROUTINES:
221	BOTH ROUTINES ARE "TABLE DRIVEN" IN THE SENSE THAT THE
222	CALLING ROUTINE PROVIDES THE ADDRESS OF A TABLE WHICH
223	DEFINES THE SPECIFIC OPERATIONS TO BE PERFORMED.

.

			A. 10 14		48 V982		
224							
225					IN ITIALIZATION		
276				PUT ALL	JOBS IN NULL DUEUE(JOB	NG. NOT ASSIGNED)	
227				;CALLED	ON RESTART AT LOC. 143		
228 229							
230				INTERNAL			
230				EXTERNAL	- CPOPU, JOBYAX, MXQUE, JBTC	2	
232					_ URTOP1 (EQUALS UBTQ+1		
232					_ URTOM1 ;EQUALS UBTO-1		
234				FXIERNAL	_ JRTOMN ;EQUALS JBTQ-NUL	- ⁰	
235	000142	211040	000.000	0.1.4.1.4			
236	000142	211040 504040	003000 000001		MOVNI TAC, MXOUE	IMAKE ALL QUEUE HEADERS POINT TO THEM	ISELVES
237	000144	202041	6061001 606100'		HRL TAC, TAC' MOVEM TAC, JBTQ(TAC)	BACKWARD POINTERS TOO	
238	000145	253040	000144		ADRUN TAC, +1		
239	000146	201040	777763				
240	007147	206040	<i><i>8</i>8888888888888</i>		MOVEN TAC INTODA	PUT JOBS ALL IN NULO QUEUE BACK POINTER FOR JOB 1	
241	000150	201200	200043				
242	000151	202044	9021441			MAX. JOB NO. For. Pointer of Jobmax job no.	
		CDCO			OVER THE BUILTENT	FOR. FUINTER OF JUBMAX JUB NU.	
243	000152	506290	200230		HREM ITEM. UNTOWN	ISET NULD HEADER TO POINT TO JOB1	
244	000153	201040	007001			JOBMAX	
245	000154	542040	0001521		HRRM TAC, JETOMN FORFARD		
246	000155	542204	000200		HERM ITEM, JBTQM1(ITEN)		
247	200156	402020	000151'		SET7M JATO		
248	000157	363220	2300000		SOULF ITEM, CPOPU	IFINISHED?	
249	000160	506274	800147'			JBACK POINTER JOB I+1 POINTS TO JOB I	
25Ø	000161	254000	0001551		JRST QINI1		

SCHEDB - SCHEDULING DCSS R. KRA	; ALGORITHM FOR SWAPPING SYSTEM(10/50)(BURROUGHS DISK) MACRC,V36 19:10 4-JUN-69 PAGE 19 SIN/AF TS3,17 22 MAR 68 V000
251	
252	,DELETES A JOB FROM ITS "SOURCE-O", DETERMINES A "DEST-O"
253	ACCORDING TO ONE OF 3 FUNCTIONS, AND INSERTS THE JOR AT
254	, THE BEGINNING OR END OF THIS DEST-Q. IN ADDITION IT MAY
255	,RESET THE JOB'S QUANTUM TIME (RH JBTSTS).
256	, THE DRIVING TABLES ARE "TRANSFER TABLES";
257	,
258	,T.TABLE: EXP <code></code>
259	, XWD <quant-tab>,<q-tab></q-tab></quant-tab>
260	,
261	, DEPENDING ON <code>, THE SECOND WORD IS EITHER DATA OR THE</code>
262	,ANDRÉSSES OF "CORRESPONDANCE TABLES",
263	
264	, THE PREFIX OF (CODE) SPECIFIES WHETHER THE JOB IS TO BE
265	,INSERTED AT THE BEGINNING OR END OF THE DEST-Q. THE SUFFIX
266	, DETERMINES THE FUNCTION USED TO SELECT THE DEST-0.
267	, THE FOLLOWING ARE THE SIX CODES AND THEIR TABLE FORMATS:
268	
269	
270	,DEST-Q AS A FIXED (PREDETERMINED) Q;
271	,BOFIX: INSERT AT BEG OF DEST-G
272	, EQFIX: INSERT AT END
273	
274	, THE JOB IS TRANSFERED TO THE END OR BEG, OF THE Q <q-tab></q-tab>
275	, IF <quant-tab> = -1, QUANT, TIME IS NOT RESET.</quant-tab>
276	, IF <quant-tab> .G. 0 , QUANT. TIME IS RESET TO <quant-tab>.</quant-tab></quant-tab>
277	, SINCE THIS FUNCTION IS FULLY DEFINED BY THE SECOND WORD
278	, ALONE, NO CORRES, TARLE IS NECESSARY.

SCHEDB - SCHEDULING ALGO DCSS R. KRASIN/A	RITHM FOR SWAPPING SYSTEM(10/53)(BURPOUGHS DISK) MACRO.V36 19:12 4-JJN-69 PAGE (F TS3,17 22 Mar 68 V000	2
279 28Ø 281 282	,DEST-Q AS A FUNCTION OF SOURCE-Q ,BQLINK: INSRT AT BEG OF PEST-Q ,EQLINK: INSERT AT END	
292 283 294	, <q-tab>=ADDRES OF A CORRES. TAPLE "LINKING" SOURCE-QS TO Dest-Qs.</q-tab>	
285 286 287	, IF <quant-tab> = -1, QUANT, TIME IS NOT RESET, , OTHERWISE <quant-tab> IS TAKEN AS THE ADDRESS OF A</quant-tab></quant-tab>	
287 288 299	, TARLE OF QUANT. TIMES CORRESPONDING TO THE Q-LINKING TABLE. , FORMAT OF THE TABLES ARE:	
290 291	, <q-tab>: XWD <sq1>,<dq1> 11ST SOURCE-QIDEST-Q PAIR</dq1></sq1></q-tab>	
202 293 294	XWD <sgn>,<d^n> INTH 7 ;ZERO TERMINATES TABLE</d^n></sgn>	
295 296	, <quant-tab>: EXP <quant1> ICORRES, TO <q-tab>+0</q-tab></quant1></quant-tab>	
297 298	<pre></pre>	
299	7	
300	, UPON A CALL TO OXFER FOR THESE 2 CODES, AC T2 CONTAINES	
371 372	, THE SOURCE-Q (CURRENT Q) OF THE JOR. THE IN OF THE	
302	, <q-tab> ENTRIES ARE SEARCHED FOR A MATCH., IF FOUND, THE</q-tab>	
324	, RH IS TAKEN AS THE DEST-Q AND THE QUANT, TIME IS RESET	
375	, (IF <quant-tab> NOT -1) TO THE CORRRES. ENTRY IN THE , <quant-tab> TABLE.</quant-tab></quant-tab>	
326	, IF NO MATCH FOUND, NO TRANSFER TAKES PLACE,	
377	The second find that it in the place,	
328		
3/29	,DEST-0 AS A FUNCTION OF JOB SIZE	
31Ø 311	, BOUSIZ INSERT AT BEG OF DEST-0	
312	,EQJSIZ INSERT AT END	
313	<pre><q-tab>=ADDRESS OF A TABLE ASSOCIATING JOB SIZE</q-tab></pre>	
314	, (IN 1K BLOCKS) TO DEST-OS.	
315	, <quant-tab> HAS SAME MEANING AS FOR B-EQLINK</quant-tab>	
316		
317	, <q-tab>: XWD <jsiz1>,<dq1></dq1></jsiz1></q-tab>	
318 319		
320	, XHD <jsizn>,<dqn></dqn></jsizn>	
321	· 2	
322	<pre><quant-tab>: Similar to that for b-eqlink</quant-tab></pre>	
323	, CONTRACTOR AND THAT FOR BEGLINK	
324	, THE <jsiz>'S MUST BE IN INCREASING ORDER,</jsiz>	
325	, THE TABLE IS SEARCHED UNTIL SUSIES IS LESS THAN OR	
326 327	I EQUAL TO THE JOB SIZE, THEN THE CORRES. COON IS	
327	, TAKEN AS THE DEST-Q, IF THE TABLE IS EXAUSTED. NO	
329	, TRANSFER TAKES PLACE.	
	, QUANT, TIME IS HANDLED AS IN B-EQLINK.	

SCHEDB						GHS DISK) MACRO.V36 19:10 4-JUN-69 PAGE 21
	9075 R.	KRASINZ	AF TS3,1	/ 22 MAR	66 V×60	
332				,CALLIN	G SEGUENCE:	
331				,	MOVE J.EJOB NUMPER]	ADDI THE AND FOLLOW ONLY
332				,		IBOLINK AND FOLINK ONLY
333				,	POVET TT, TRANS TABLE ADD PUSHJ PDP, GXFER	D#F22
334 335				,		
336						(R)=-1 IF QUANT. TIME NOT
337					TI SCHANT. TIME IF RESET.	
338				,		
339				ACS:		
340			000025		POINTER TO TRANSFER TAP	BLE
341			000004	J=ITEM	;JOB NO.	
342			000077	Q=PROG	IDEST-9 AND QUANT, TIME	ON RETURN
343				T1 = TAC1		
344			000031	T2 = TAC	; TEMP AND SOURCE-Q ON C	ALL TO BIEGLINK
345				EXTERNA	60000	
346 347				EXTERNAL	LERRUR	
348	000162	200345	000071	OXEER:	MOVE Q,1(TT)	GET TRANSFER TABLE ADDRESS
349		254025		ANT ENT	JRST @(TT)	IDISPATCH
350	100					
351				,DEST-0	AS FUNCTION OF SOURCE-Q	
352	000164	336197	200200	OLINK:	SKIPN T1, (Q)	FEND OF TABLE?
353		26314K			PGPJ PDP,	IYES
354		574170			WLRE T1,T1	
355			000001		CAME T1,T2	INOSOURCE-D=LH(TABLE ENTRY)?
356			000164'		ADBUP QARLINK	INO- CONTINUE SEARCH
357	00r1/1	254000	000201'		JRST QX2	IYES
358 359				DEST-O	AS FUNCTION OF JOB SIZE	
359 360	000172	554044	0000000		HLRZ T2, JBTADR(J)	HIGHEST REL. LOC. OF JOB
361	000172		777766	00312	ASH T2, -+D10	CONVERT TO NO. OF 1K BLOCKS - 1.
362	200174		000001		MOVSI T2,1(T2)	IND, OF 1K BLOCKS TO LH,
363	000175	336107	<i>PO9000</i>	QX1:	SKIPN T1, (Q)	JEND OF TABLE?
364		265240	000000		JSP DAT, ERROR	
365	000177	313040	000002		CAMLE T2, T1	JOBSIZE ,LE, LH(TABLE ENTRY)?
366	000200	252340	000175'		AOBUP N,OX1	INDCONTINUE SEARCH, JUMP ALWAYS,
367					_	
368		204040	000607	0×2:	MOVS T2,0	IT2 IS ADDR, OF QUANT, TIME (IF REQUESTED)
369	000202	560347	000000		HRRO Q, (Q)	RH(0)=DEST-QJLH=-1(NO QUANT.TIME RED.)
370	000203	331005			SKIPL 1(TT) HRL 0,(T2)	;WAS QUANT. TIME REQUESTED? ;YESGET 1T
371	000204	504341	900200) ILO0EI II

372							
373					DEST-0		
374			2001561	OFIX:	MOVE T1, JBTQ(J)	IDELETE JOB FROM SOURCE-O	
375	200206	204040	623462		MOVS T2,T1	JT1=FORW, LINK, T2=BACK LINK	
376	000207		0002351		HRRY T1,JBTQ(T2)	IFORW, LINK PAST JOB	
377 378	ØØØ210	506042	000207 ·		HRLM T2, JHTQ(T1)	BACK LINK PAST JOB	
379	700211	335025	000000		SKIPCE (TT)	IEND OR REG, OF Q?	
380	000212	544347	0002101		HLR 7, JETQ(Q)	JEND-THIS WILL LEAVE Q=IDX OF	
381						CURRENT LAST LINK; T2=IDX OF Q-HEADER	
382	000213	200047	000212'		MOVE T2, JBTQ(Q)	JBEGT2=IDX OF CURRENT 1ST LINK	
383						; Q=IDX OF Q-HEADER	
384	000214	542297	0002131		HRRM JIJATQ(Q)	INSERT JOB IN DEST-Q	
385	000215	506201	000214		HRLM J.JBTQ(T2)	FIGURE HOS IN DECK &	
386	000216	542044	0002151		HRRM T2, JBTQ(J)		
387	000217	506344	000216		HRLM G.JSTO(J)		
388			-				
389	606550	32134Ø	0002241		JUMPL 0,0X3	RETURN IF QUANT. TIME NOT REQ.	
390	000221	546344	000125		HLRM Q, JBTSTS(J)	SET QUANT. TIME	
391	000222	201240	000000		MOVEL TT,RNO	ISET JOB STATUS WAIT	
392		137240	RØ2060 1		NPB TT, PUBSTS	CODE TO RUN QUEUE (2).	
393	200224	263140	000020	QX3:	POPJ POP.		
394					- ,		
395			P002751	BOFIX=0	0F1X		
396		400000					
397			000164				
398		400000			QLINK+1BØ		
399			0001721				
400		400000			QJSIZ+1BØ		

SCHEDB - SCHEDU	ING ALGORITHM FOR SWAF KRASIN/AF TS3,17 22 MA	PING SYSTEM(10/ R 68 vøøø	57)(BURROUGHS DISK)	MACRD.V36 19:12	4-JUN-69 PAGE 23
471	SCANS	THE QS RETURNI	NG THE NUMBERS OF THE	JOBS IN THE QS.	
402	THE C	RDER AND MANNER	IN WHICH THE QS ARE	SEARCHED IS	
403	DETER	MINED BY A "SCA"	N TABLE" ADDRESSED IN	THE CALLING SEQ.	
4714	THE	CAN TARLE HAS T	HE FORM:		
495	,	-			
406		AB: XWD <	Q1>, <code1> IS</code1>	CN Q1 ACCEDING TO CODE	1
497					
478		XWD <	QN>, <comen> IQ</comen>	N ACCORDING TO CODEN	
429	,	Ŧ	JZERO TERM	INATES TABLE	
410	,				
411	, EACH	Q MAY BE SCANNE	D IN ONF OF FOUR WAYS	SPECIFIEDRY <code></code>	
412	,THE C	ODES ARE:			
413	,				
414	,QFOR	SCAN WHOLE Q			
415			THE 1ST MEMBER (IF AN	Υ)	
416		SCAN WHOLE O			
417	,QRAK1	SCAN BACKWARD	FOR ALL MEMBERS EXCE	PT THE 1ST	
418	,				
419	, CALLI	NG SEQ.			
420	,				
421	,		TABLE ADDRESS		
422	,	JSP PC.QSCAN	SFT UP PC FOR REP		
423	,		IRETURN HERE WHEN		
424			RETURN HERE WITH		
425	,		; AND ITS Q IN LH(QR)	
426	•				
427	,		ECESSARY TESTING OF T	HIS JOB	
428	,	J,ST,PC,QR MU	ST BE PRESERVED		
429	,				
430	,	JRST (QR)	RETURN TO QSCAN T		
431	,		I IF THIS ONE NOT	ACCEPTABLE	
432	,				
433	,ACS:				
434		JOB NO.			
435		POINTER TO S			
436		RETURN ADDRE			
437	000002 OR=TAC	J FIFERATED RET	URN ADDRESS TO OSCAN		

2.15			AF TS3.1)(BURPOUGHS DISK) MACRO,V	36 19:10 4-JUN-69 PAGE
438	C00225	336125	000000	OSCAN:	SKIPN OR, (ST)	IEND OF SCAN TABLE?	
439	000226	254091	000000		JRST (PC)	IYES-HRETURN TO CALL+1	
44Ø	700227	574270	000002		HLRE J,QR	INGGET NO. OF G	
441	000230	254022	agagag		JRST (OR)	JDISPATCH	
442							
443	000231	201120	000234'	QFOR1:	MOVEL OR, QFOR2	JONLY THE FIRST JOB	
444							
445	007232	570274	0002171	OF OR:	HRRE J.JBTQ(J)	ISCAN FOWARD ALL JOBS	
446	200233	327201	000001		JUMPG J,1(PC)		55
447	000234	344240	0002251	OFOR2:	ADJA ST.OSCAN	JEND OF THIS QGET NEXT Q	- · · ·
448							
449	000235	5742"4	0002321	QBAK1:	HLRE J, JBTQ(J)	ISCAN BACKWARD ALL JOBS EXCEPT :	IST
450	00236	3330*4	0002351		SKIPLE JETO(J)	IS THIS THE FIRST MEMPER?	
451	202237	254001	000001		JRST 1(PC)	INORETURN CALL+2	
452	000240	344240	0002251		ADJA ST, OSCAN	IYESGET NEXT Q	
453							
454	000241	574224	0002361	QBAK:	HLRE J, JRTQ(J)	ISCAN BACKWARD ALL JOBS	
455	707242	327291	000001		JUMPG J.1(PC)	JRETURN CALL+2 WITH JOB NO. UNLE	ss
456	909243	344240	0002251		ADJA ST, OSCAN	BEG OF THIS QGET NEXT Q	

SCHEDB - SCHEDULING ALGORITHM FOR SWAPPING SYSTEM(10/53)(BURROUGHS DISK) OCSS R. KRASIN/AF TS3.17 22 MAR 68 V200 MACRO, V36 19:10 4-JUN-69 PAGE 25 INTERNAL FTCHECK, FTMONP IFN FICHECK+FIMONP. < EXTERNAL AVALTB DEFINE X(A,B),< EXTERNAL A'AVAL INTERNAL A'O A')=72 22=22+1 > **Z =**Ø OUEUES LOC=72 IFE FICHECK+FIMONP. < SHARABLE DEVICE JUST RECOME AVAILABLE(EXTENDED TO OTHER QUEUEW TOO) SAPPROPRIATE ENTRY IS SET NON-ZERO WHEN SCHEDULER SHOULD LOOK SAT THAT QUEUE TO FIND A JOB TO RUN SWSAVAL CONTAINS THE NO. OF JOBS WITH IO WAIT SATISFIED(Ø=NONE) 474 DEFINE X(A,B) <INTERNAL A'AVAL,A'Q A'Q=,-AVALTB A'AVAL: 0 > INTERNAL AVALTB AVALTE: QUEUES ;GENERATE THE AVAL FLAGS addond RNAVAL: 0 WSAVAL: 0 TSAVAL: 9 STAVAL: 0 ØØØØØØØ AUAVAL: 0 aaaaaa MOAVAL: 0 DAAVAL: 0 DTAVAL: 0 DCAVAL: 2 MTAVAL: Ø LOC=.-AVALTB NQUEUE=LOC INO. OF QUEUES COUNTING RUN QUEUE XP MAXQ,NQUEUE-1 XP MINQ,STQ IMAX, STATE CODE WHICH HAS AN AVAL FLAG IMINIMUM SHARABLE DEVICE QUEUE XP AVENUM, MAXQ IMAX. STATE CODE WHICH HAS AN AVAL FLAG IDEFINE STATE CODES WHICH DO NOT HAVE AVAL AND REQ FLAGS DEFINE X(A) <INTERNAL A'Q A'Q=LOC LOC=LOC+1

SCHEMB - SCHE RCSS I	DULING ALGORITHM FOR SWAPPING S R, KRASIN/AF TS3,17 22 MAR 66 V	YSTEN(10/57)(BURROUGHS DISK) 1007 -	MACRO,V36 19:10 4-JUN-60 PAGE 25-1
510	、 、		
511	CODE	S. V. LOW ADDITIONAL ACTO	
512	0. C. E	S+ X IOW, +INTERMAL IOWQ	
513	YP MYCODE.L		
514	202017 P01=LOC	CH1 ;HAX, JOP STATE CODE	
515	002220 LOC=LOC+1		
516	P07020 PU2=LOC		
517	202021 LOC=LOC+1		
518	202221 PQ3=10C		
519	000222 LUC=LUC+1		
<u>5</u> 20	000022 CMC=LOC	COMMAND DELAY QUEUE	

,

SCHEDB					ING SYST 68 V000	Em(1)	Ø/52)(E	บหะอุบุษะ	s c	ISK)	MACR0,V36 19:12 4-JUN-69 PAGE 26
501			• • •								
521 522					RANDENCE	TAD		EEN 100	ST	ATUS CODE	S AND QUEUE TRANSFER TABLES
523					Y SCHEDU			CE 50-	51	A103 0030	
524							ATUS NO		نيا ت	STATE COD	١٢.
525										BLE ADORE	
526										F AS. INDE	
507				I OLLOW	100 1400	L U3	1.0 0.4	J.A.L	000		
528				DEFINE	X / A . D \						
529				<	FXP Q'A	1.0					
530				>							
531				·							
532				INTERNA							
533				1016.00							
534	000256	000070	0003321	OBITS:	QUEUES+	x Ri	N.7 +	F	хP	ORNW	
535	000257	0000.0	0003341	101.0	X WS.6	+		POWSW			
536	000260	000000	000336		X TS.6			P OTSW			
537	000261	0000%0	0003541		X ST.6	+		P OSTW			
538	002262	200220	0003441			X AI	U.4 +	E	XР	QAUW	
539	000263	000000	202346			X M	0,4 +			OMOW	
540	000264	RØØØVØ	000350			X D.	A 4 +	E	ХP	ODAW	
541	000265	3000.20	0003561		X DT,4	t	EX	P ODTW			
542	400266	0000000	0003521		X DC.4	+	ΕX	P ODCW			
543	000267	0000000	0003601		X MT,4	+	E X	P OMTW			
544	000270	000010	000340		CODEST	X I	OW, +	E	ΧP	NIOWW	
545	000271	200000	000342'		X TIOW,	+	ΕX	P OTIOWI	A		
546	000272	300020	000362'		X SLP,	+	ΕX	P GSLPW			
547	000273	000000	0003241		X NUL,	÷	ΓX	P QNULW			
548	000274	2000000	000326'		X STOP,	+	F X	P OSTOPI	A		

j

	OCSS R.	KRASIN	AF TS3.	FOR SWAPPING SYSTEM(12/52)(BURROUGHS DISK) MACRO,V36 19:10 17 22 Mar 68 V000	4-00%-69 PAS
549				IFN FICHFCK+FIMONP,<	
550				DEFINE X(A,B),<	
551				EXTERNAL A'REQ	
552					
553				AUEVES	
554				EXTERNAL REDTAB	
55 5					
556				IFE FICHECK+FTMONP,<	
557					
558				SHARABLE DEVICE REQUEST TABLE (GENERALIZED FOR OTHER QUEUES TOO)	;
559				CONTAINS THE NUMBER OF JOB MAITING TO USE SHARBLE DEVICE	
560				WAREN AND RAREN ARE UNUSED	
561					
562				DEFINE X(A,B)	
563				<a'req: 0<="" td=""><td></td></a'req:>	
564				INTERNAL A'REQ	
565				>	
566 567					
267				INTERNAL REGTAR	
568					
	002275			REQTAB: QUEUES ;GENERATE REQ TABLE	;
	20275	0000 °0	007000	X RN,7 +RNREQ1 Ø	
	300276	0000000	000000	X WS.6 +WSREQ: 0	
	000277	000010	009290	× TS,6 →TSRFQ: Ø	
	202300 202321	000220	000000	X ST.6 +STREQ; Ø	
	202321 202322	1006.10 1006.70	002000	X AU,4 +AUREQ: Ø	
	202323	0000000 0000000	000000 000000	X MQ,4 +MORFQI Ø	
	202323	202020	0000000 0000000	X DA,4 +DAREQ: Ø	:
	302325	2020220 200220	0000000 002000	X DT,4 +DTREQ: @	
	102376	0000000 002000	202000	X DC,4 +DCRFQ; Ø	
580	.2.070	~ U C () ; []	A: KIA: KI 19 KI	X MT,4 +MTRFQ: 2	

		-			48 V. ()								
581													
592								IERE CODE.	RH=QUEUE	TRANSFER T	ABLE ADF	₹ .	
593								JEHE CODE					
594					ARABLE D								
585								ITING FOR			ND		
526								OUN CUFUE					
587				BULEDE	TRANSFER	TABLE	AS SF	PECIFIED BE	LOW BY TH	F JOB WAIT			
598				STATE	CODE.								
589													
590				DEFINE									
591				<	XKD -A'	Q,Q'A'	S						
592				>									
593													
594			200000		IND COR	RESPON	DENCE	TABLES FO	THESE QUE	UES			
39 5			RORGAR	Ω₩SS=10									
596			0000000	QTSS=0									
597													
598				INTERNAL	_ AVLOTE								
399													
630	909397	2002°0	234926	AVLQTB:			7 †	X M D -	RNQ, GRNS				
631	P0P310	777777	602230		× WS,6			- ₩SQ,Q ₩SS					
6.*2	900311	777776	002372		× TS,6			-TSQ.QTSS					
633	200312	777775	0003721		X ST,6			-STQ,QSTS					
6 3 4	202313	777774	0004201			X AU,			AUQ,GAUS				
625	002314	777773	000364				4 +		MQQ,QMQS				
606	ØØC315	777772	000366'				4 +		DAQ,0DAS				
577	000316	777771	0003741		X DI,4			-DTQ,QDTS					
608	909317	777770	0003701					-DCQ,QDCS					
679	200320	777767	000376'		X MT.4	+	XWD	-MTQ,QMTS					

610				IFN FTCHECK+FTM	ONP.<
511				EXTERNAL QJOR, J	OBQUE JOBQUE WILL CAUSE LOAD OF PROPER SCHDAT
612					; DEPENDING ON FTRC10 IN SCHDAT
613				IFN FTSWAP,<	
614				EXTERNAL XJOB	
615 616				IFE FTCHECK+FTM	
617				THE FROMEON #FIN	
618					
619				INTERNAL JOBQUE	
620	000321	0000000	0000000	JOBQUE: 2	JOBS TO BE REQUEUED ON CLOCK INTERRUPT
621					
622				INTERNAL QJOB	
623	000322	0000000	609000	QJOB: Z	;NUMBER OF JOBS MEEDING Q TRANSFERS AT OTHER THAN CLOCK LEVEL
624					
625				IFN FTSWAP,<	WINDER OF WARD WEEDING CODE ENDINGIAN DV CUIDANT IN
626	ØØØ323	200000	000000	XJOB: 7	INUMBER OF JOBS NEEDING CORE EXPANSION BY SWAPOUT-IN
627				INTERNAL XJOR >	
628				,	
62 9				>	

SCHEDB				OR SWAPPI 7 22 mar			5%)(BURROU	GHS DIS	5K)	MACRO,V36	19:10	4-JIJN-60	PAGE 32
63Ø 631				INTERNAL	QSTO	P,QTIME,	SSCAN,QCMW						
632			0002061	BQFIX=QF	1 V	1 DEC 1		nence e	IXED QUEUE				
633		400000							IXED JUEUE		IN F_		
634		4 10 10 10 10 10				O JEND	DF QUEUFS						
				POLINK=Q									
635		400000		EOLINK=O		196							
636				RUJSIZ=Q									
637		400000	000172	EQJSIZ=Q			_						
638				DEFINE T	TAB (FI								
639				<		EXPF							
640						XMD OI	JANT,-QUEU	E					
641				>									
642				DEFINE P	TTAB(I	FCTN,QUEL	JE, QUANT)						
643				<		EXP FO							
644						XWD gl	JANT, QUEUE						
545				>									
646													
647	000324			ONULW:	TTAB B	OF IX, NUL	.0,-1	INULL	QUEUE JOB	NO. NOT A	SSIGNED		
648	200324	400020	0002051			EXP EC	FIX			•			
649	ØØØ325	777777	777763			XWD -1	L ,-NULQ						
650	RØ0326			QSTOP:QS	TOPW:		OFIX STOP	01	LUNRUNAR	LE JOBS T		E STOPO	
651	200326	400070	0002051			EXPE		47 -	10		0 200 00	01010	
652	000327	777777	777762				-STOPQ						
653	000330			QCMW:		QFIX, CMG				WAIT TIL			
654	000330	400000	0002051	4 0.111		EXPEG			1 COMPAND		_ JUB 1	V CONE	
655	000331	777777	777756			XWD -1	-	- 640					
656	000332	//////	////50	QRNW: I				,-CMQ	I UST DU		-		
657	000332	400030	0001721	GIVE 201	FILAD	EXP EG	STAB,QOST	AD	10031 RU	INABLE JOB	5		
658	000333		000404										
659	100333	71071414	1.08.44.4		. Jan ter	XWD QG		, QSTAB					
660					106 VI	1 ARE NUI		T TINC	ATE BELOW,	ENTER PRUC	LISSUR		
661					I G S A I	END AND	GET GUAN	I I I ME	ACCORDING	IN THEIR	SIZE		
662	000334			QWSW:	I I AB E	OFIX, PO1		IIO WA	IT SAT.(EX	CEPT TTY)			
663	000334	300030	000205			EXP BC	•	.					
664	000335	NØ2076	777761			XMD JO		,-PQ1		_			
665								DR QS A	ND GET QUA	NT. TIME			
666							JOB SI7E						
667	000336			OLZM:	TTAB E	OFIX, PQ1		JIYI	O WAIT SAT	'ISFIED(EN'	TER FROM	NT OF PO1)	1
668	000336	0000070	0092951			EXP BG	FIX						
669	000337	2000 76	777761			XWD QG	ATTY	,-PQ1					
670													
671	00034Ø			QIOWW: 1	ΤΤΑΒ Ε	QFIX, IOW	Q,-1	IIOW(E	XDEPT TTY)	HELD IN 1	LOWQ .		
672	000340	4000000	0002051			EXP EG	FIX						
673	000341	777777	777766				,-10WQ						
674	000342			OLIOMM: .	ΤΤΑΒ Ε	QFIX,TIC	WQ1	I YTTE	OW HELD IN	TIOWQ			
675	000342	400270	P00205'			EXP EG	FIX						
676	000343	777777	777765			XWD -1	TIOWO						
677	000344	400000	0002051	QAUW: .	ΤΤΑΒ Ε	OFIX, AUG			EXP EOFI	x			
678	000345	777777	777774			XWD -1							
679	000346			QMQW: 1	TTAB F	QFIX, MOG		IMON.	Q(DISK) WA	τT			
680	000346	400000	0002051			EXP EQ	, -	11011		• '			
681	000347	777777	777773				, -MQQ						
682	000350			ODAW: 1		OFIX, DAG		IDEV	ALLOC.(DIS	2			
						GI INTUAU				n /			

SCHEDB - SCHEDULING ALGORITHM FOR SWAPPING SYSTEM(10/50)(BURROUGHS DISK) MACRO,V36 19:10 4-JUN-60 PAGE 30-1 OCSS R, KRASIN/AF TS3,17 22 MAR 68 V000

683	000350	400020	000205			EXP EQFIX	
684	000351	777777	777772			XWD -1 ,-DAO	
685	000352			QDCW:	TTAB	EQFIX,DCQ,-1	IDATA CONTROL WAIT
686	000352	400000	ØØØ205'			EXP EQFIX	
687	000353	777777	77777Ø			XWD -1 ,-DCO	
688	000354			OSTW:	TTAB	EQFIX,STQ,-1	ISYST TAPE
689	000354	400000	0002051			EXP EQFIX	
690	000355	777777	777775			XWD -1 ,-STO	
691	000356			QDTW:	TTAR	EQFIX, DTQ, -1	IDEC TAPE
		400000	0002051	GO I N	1100	EXP EQFIX	
692	000356						
693	000357	777777	777771			XWD -1 ,-DTO	
694	ØØØ36Ø			OMTW:	TTAB	EQFIX,MTQ,+1	IMAG TAPE
695	000360	400000	000205'			EXP EQFIX	
696	000361	777777	777767			XWD -1 ,-MTQ	
697	000362			OSLPW:	TTAB	EQFIX, SLPQ, -1	ISLEEP UUD
		400070	0002051			EXP EQFIX	, , , , , , , , , , , , , , , , , , , ,
698	@@@362					· · ·	
69 9	000363	777777	777764			XWD -1 ,-SLPQ	

			AF 153.1	/ C.C. 1011	60 V	101			
778				TRANSL	ATION	TABLE FROM WAIT S	TATE TO SATI	SELED STATE	
711				100 NUT	RESE	T QUANTUM RUN TIME			
7.2									
773		777777	777777	QUS()=−1					
774									
7 ≀5	@02364			3435:	TTAB	BOFIX,PQ1,QQSD	ISTART MON	. Q(DISK) AT PQ1	
7:*6	000364	000000	0002051			EXP BOFIX		I GIGISHU AT TUL	
7 7 7	000365	777777	777761			XWD QQSD	,-PQ1		
7218	PØ2366			ODAS:	TTAB	BOFIX, PO1, OOSD		C.(DISK)	
7 79	000366	P02220	000205			EXP BOFIX			
710	000367	777777	777761			XWD DQSD	,-PQ1		
711	200370			ODCS:	TTAB	BOFIX, PC1, QOSD	IDATA CONT	R01	
712	ØØØ37Ø	300000	000205			EXP BOFIX		52111	
713	000371	777777	777761			XWD DQSD	,-PQ1		
714	ØØØ372			OSTS:	TTAB	BOFIX, PO1, DOSD	ISYST TAPE		
715	ØØØ372	700000	000205			EXP BOFIX			
716	000373	777777	777761			XWD QQSD	-PQ1		
717	000374			QDTS:	TTAB	BOFIX PO1,00SD	IDEC TAPE		
718	000374	200020	0002255			EXP POFIX			
719	000375	777777	777761			XMD JOSD	PQ1		
720	000376			DMTS:	TTAG	80FIX, P01, 2050	IMAG TAPE		
721	200376	~0000r0	0002751			EXP ROFIX			
772	000377	777777	777761			280 OKX	,-PQ1		
723	000470			QAUS:	TTAB	BAFIX, PA1, 30SD	JALTER UFD		
724	P00400	000000	@00225			EXP BOFIX			
725	202421	777777	777761			XWD QQSD	,-PQ1		
726	000402		-	QTIME:	PTTAF	P EGLINK, QTTAB, QOS	TAB IM	OVE JOB TO LOWER Q	
127	000402	400000	000164			FXP EQLINK		in a serier w	
728	000403	000414	000410			XWD QQSTAB	QTTAB		

730 ,ENTER PROCESSOR OS ACCORDING TO JOB SIZE 731 ,O00424 000244 77761 733 007425 000202 777761 734 007425 000202 777761 735 002406 702400 777757 XWD +D256,-P03 IP03 IF 16 .L. SIZE 736 737 ,PUT JOB DOWN A 0 IF EXCREDS QUANT. TIME 738 000411 777760 777757 739 000412 777757 777760 740 000412 777757 777760 741 700000 700000 742 ,QUANTUM TABLES ,QUANTUM TABLES 743 ,QUANTUM TABLES ,QUANT. IIMES ACCORDING TO PROCESSOR 0: 743 ,QUANT. IIMES ACCORDING TO PROCESSOR 0: ,QUANT. IIMES ACCORDING TO PROCESSOR 0: 744 ,QUANT. IIMES ACCORDING TO PROCESSOR 0: ,QUANT. IIMES ACCORDING TO PROCESSOR 0: 748 ,QUANT. IIMES ACCORDING TO PROCESSOR 0: ,QUANT. 750 ,QUANT: ,QUANT. IIMES ACCORDING TO PROCESSOR 0: 755 ,QUANT: ,PY +038 ;PQ1: ONE HALF SECOND 756 ,QUAV15 ,QUAV20	PAGE
732 ØØØ424 PØØ2:4 777761 DSTAF: YWD 4,PD1 JPD1 IF SIZE .LE. 4K 733 ØØP425 PØØ202 777760 YWD +D16, PD2 JP2 IF 4K .L. SIZE .LE. 16K 734 ØØP426 700000 777760 YWD +D256, PD3 JP03 IF 16 .L. SIZE .LE. 16K 735 ØØP407 700000 707760 YWD +D256, PD3 JP03 IF 16 .L. SIZE 736 ØØP410 777761 777760 QITAF: YWD -P01, PD2 736 ØØP410 777761 777760 QITAF: YWD -P01, PD2 739 ØØP410 777761 777760 QITAF: YWD -P02, PD3 740 ØØP410 777757 777760 YWD -P03, PD2 JBACK TO P02 TO COMPETE WITH IOWS JOBS 741 77757 777760 XWD -P03, PD2 JBACK TO P02 TO COMPETE WITH IOWS JOBS 742 J00010 ØØ0000 Z Z JOUANTUM TABLES 744 J000010 ØØ0000 Z Z JOUANT. FOR SHAR, DEV. WAITERS 744 J000010 ØØ0000 QUSD=6 JTENTH SEC. INITIAL QUANT. FOR SHAR, DEV. WAITERS	
732 000424 000214 777761 OSTAR: YWD 4,-PD1 ;PD1 IF SIZE .LE. 4K 733 000405 00020 777760 XWD +D16,-PO2 ;PD2 IF 4K .L. SIZE .LE. 16K 735 002407 002000 777761 777760 XWD +D256,-PO3 ;PD3 IF 16 .L. SIZE .LE. 16K 736 002407 002000 777760 777760 7777777 XWD +D256,-PO3 ;PD3 IF 16 .L. SIZE .LE. 16K 736 .PUT JOB DOWN A 0 IF EXCEEDS QUANT. TIME 0TTAF: XWD -P01,-PO2 YWD -P02,-PO3 740 000411 777760 777760 YT7760 YT7660 YWD -P03,-PO2 ;BACK TO P02 TO COMPETE WITH IOWS JOBS 741 300413 700000 000000 7 YWD -P03,-PO2 ;BACK TO P02 TO COMPETE WITH IOWS JOBS 742 .OUANTUM TABLES .OUANTUM TABLES .OUANTUM TABLES .OUANT.TO THE SEC. INITIAL DUANT. FOR SHAR. DEV. WAITERS 744 .OUANT. TIMES ACCORDING TO PROCESSOR 0: .OUANT. TIMES ACCORDING TO PROCESSOR 0: .OUANT. TIMES ACCORDING TO PROCESSOR 0: 750 .OUANT. TIMES ACCORDING TO PROCESSOR 0: .OUANTERNAL RNOUNT: .OUANTA INFOR FXP +D36 .PO1:	
733 M004405 C00020 777760 YKD +D16,-P02 JP02 IF 4K .L. SIZE .LE. 16K 734 C00400 777757 XWD +D256,-P03 JP03 IF 16 .L. SIZE .LE. 16K 735 C00407 C00400 C00220 Z .WD +D256,-P03 JP03 IF 16 .L. SIZE .LE. 16K 736 .PUT JOB DOWN A 0 IF EXCEEDS QUANT, TIME	
734 M04466 200400 77757 XwD +D256,-PQ3 ;PQ3 IF 16 .L. SIZE 735 600407 700000 600220 7 736 ,PUT JOB DOWN & Q IF EXCEEDS QUANT. TIME 738 600410 777761 777760 739 000411 777760 777760 0TTAP: XwD -PQ1,-PQ2 749 000412 777757 777760 XwD -PQ2,-PQ3 741 7000010 000000 7 XwD -PQ2,-PQ3 742 ,QUANTUM TABLES ,QUANTUM TABLES 744 ,QUANTUM TABLES ,QUANT. FOR SHAR, DEV. WAITERS 744 ,QUANT. TIMES ACCORDING TO PROCESSOR Q; ,QUANT. TIMES ACCORDING TO PROCESSOR Q; 748 ,QUANT. TIMES ACCORDING TO PROCESSOR Q; INTERNAL RNGUNT 750 INTERNAL RNGUNT RNGUNT; 751 7607414 700700 200726 754 7007415 700700 200726 754 7007415 700700 2007170	
735 000407 000400 000220 7 736 ,PUT JOB DOWN & Q IF EXCEEDS QUANT, TIME 737 ,PUT JOB DOWN & Q IF EXCEEDS QUANT, TIME 738 000410 777761 739 000411 777760 740 000412 777757 740 000412 777757 741 000010 000000 742 ,QUANTUM TABLES 743 ,QUANTUM TABLES 744 ,QUANTUM TABLES 745 000006 QUSD=0 746 000006 QUSD=0 747 ,QUANTUM TABLES 748 ,QUANT. TIMES ACCORDING TO PROCESSOR Q: 749 ,QUANT. TIMES ACCORDING TO PROCESSOR Q: 750 INTERNAL RNGUNT 751 752 0000414 000200 754 000415 000200 002170 754 000415 000200 002170	
737 ,PUT JOB DOWN & Q IF EXCEEDS QUANT. TIME 738 007410 777761 777760 739 007411 777760 777757 740 000412 777757 777760 741 0070413 000000 007000 XWD -PQ2,-PQ3 742 ,QUANTUM TABLES ,QUANTUM TABLES 744 ,QUANTUM TABLES ,QUANT. FOR SHAR, DEV. WAITERS 746 ,QUANT. TIMES ACCORDING TO PROCESSOR Q: ,QUANT. TIMES ACCORDING TO PROCESSOR Q: 747 ,QUANT. TIMES ACCORDING TO PROCESSOR Q: INTERNAL RNGUNT 750 INTERNAL RNGUNT 751 752 007070 007270 007270 754 007070 007270	
738 #0#410 777761 777760 OTTAB: xwD -PQ1,-PQ2 739 #0#411 777760 777757 XWD -PQ2,-PQ3 740 #0#412 777757 777760 XWD -PQ3,-PQ2 JBACK TO PQ2 TO COMPETE WITH IOWS JOBS 741 #0#413 #0#0010 #0#0000 7 XWD -PQ3,-PQ2 JBACK TO PQ2 TO COMPETE WITH IOWS JOBS 742 #UNANTUM TABLES #UNANTUM TABLES #UNANTUM TABLES #UNANT #UNANT FOR SHAR, DEV. WAITERS 744 #UNANTO #UNANT #UNANT FOR SHAR, DEV. WAITERS 744 #UNANT #UNANT #UNANT FOR SHAR, DEV. WAITERS 744 #UNANT #UNANT #UNANT FOR SHAR, DEV. WAITERS 744 #UNANT #UNANT #UNANT FOR SHAR, DEV. WAITERS 745 #UNANT #UNANT #UNANT FOR TTY IOWS 748 .	
738 007410 777761 777760 QTTAP: XWD -PQ1,-PQ2 739 007411 777760 777760 XWD -PQ2,-PQ3 740 007412 777757 777760 XWD -PQ3,-PQ2 JBACK TO PQ2 TO COMPETE WITH IOWS JOBS 741 007010 007000 Z JBACK TO PQ2 TO COMPETE WITH IOWS JOBS 742 ,QUANTUM TABLES ,QUANTUM TABLES JBACK TO PQ2 TO COMPETE WITH IOWS JOBS 744 ,QUANTUM TABLES ,QUANTUM TABLES 744 ,QUANTUM TABLES ,QUANT. FOR SHAR, DEV. WAITERS 745 M070766 QUSD=6 JTENTH SEC. INITIAL QUANT. FOR THY IOWS 746 ,QUANT. TIMES ACCORDING TO PROCESSOR Q: ,QUANT. TIMES ACCORDING TO PROCESSOR Q: 748 ,QUANT. TIMES ACCORDING TO PROCESSOR Q: INTERNAL RNCUNT 750 INTERNAL RNCUNT GOSTAB: FXP +D30 JP01: ONE HALF SECOND 754 007070 QU770 EXP 2*+D60 JP02: TWO SECONDS	
740 000412 777757 777760 xwn -PQ3,-PG2 ;BACK TO PQ2 TO COMPETE WITH IOWS JOBS 741 707413 000010 000000 z ;guantum tables 743 ,QUANTUM TABLES ;guantum tables ;guantum tables 744 ,QUANTUM TABLES ;guantum tables ;guantum tables 744 ,QUANTUM TABLES ;guantum tables ;guantum tables 745 ,QUANT. ;guantum tables ;guantum tables 746 ,QUANT. ;guantum tables ;guantum tables 747 ,QUANT. ;guantum tables ;guantum tables 748 ,QUANT. ;guantum tables ;guantu tables 750 ,QUANT. ;guantu tables ;guantu tables 751 ,QUANT. ;guantu tables ;guantu tables 752 ,QUANT. ;guantu tables ;guantu tables 753 ,QUANT. ;guantu tables ;guantu tables 754 ,QUANTA ;guantu tables ;guantu tables 754 ,QUANTA ;guantu tables ;guantu tables 754 ,QUANTA ;guantu tables	
741 707413 700000 700000 7 742 743 ,0UANTUM TABLES 7 744 745 700000 00000 7 745 700006 00000 1TENTH SEC, INITIAL QUANT. FOR SHAR, DEV. WAITERS 746 700006 QUTTY=6 ;TENTH SEC, INITIAL QUANT. FOR TTY IOWS 747 748 . QUANT. TIMES ACCORDING TO PROCESSOR Q: 749 INTERNAL RNCUNT 750 INTERNAL RNCUNT 751 7000414 RNQUNT: 752 7000414 700000 200226 754 7002415 2002200 202170 754 7002415 2002200 202170	
741 307413 200010 007000 7 742 ,0UANTUM TABLES ,0UANTUM TABLES 744 ,000006 00SD=0 ITENTH SEC, INITIAL DUANT. FOR SHAR, DEV. WAITERS 745 ,000006 00SD=0 ITENTH SEC, INITIAL DUANT. FOR SHAR, DEV. WAITERS 746 ,00006 00TTY=6 IENTH SEC, INITIAL DUANT. FOR TTY IOWS 747 ,00006 00TTY=6 IENTH SEC, INITIAL DUANT. FOR TTY IOWS 748 ,0UANT. TIMES ACCORDING TO PROCESSOR 0; .0UANT. TIMES ACCORDING TO PROCESSOR 0; 749 .0UANT. TIMES ACCORDING TO PROCESSOR 0; .0UANT. TIMES ACCORDING TO PROCESSOR 0; 750 .0UANT. TIMES ACCORDING TO PROCESSOR 0; .0UANT. TO PROCESSOR 0; 751 .000414 RNOUNT; .005141 752 000414 .000726 .005141; FXP +D30 754 0002000 .007170 .EXP 2**D60 754 .007216 .005141 .005140	
743 ,QUANTUM TABLES 744 000006 QUSD=0 ;TENTH SEC, INITIAL DUANT. FOR SHAR, DEV. WAITERS 745 000006 QUSD=0 ;TENTH SEC, INITIAL DUANT. FOR SHAR, DEV. WAITERS 746 000006 QUTY=6 ;TENTH SEC, INITIAL DUANT. FOR SHAR, DEV. WAITERS 747 000006 QUTY=6 ;TENTH SEC, INITIAL DUANT. FOR SHAR, DEV. WAITERS 748 ,QUANT. TIMES ACCORDING TO PROCESSOR Q; 749 INTERNAL RNGUNT 750 INTERNAL RNGUNT 751 000414 RNGUNT; 752 000414 RNGUNT; 753 000414 00051AB; FXP +D30 ;P01; ONE HALF SECOND 754 000000 007236 QOSTAB; FXP +D30 ;P02; TWO SECONDS	
744 745 000006 QUSD=6 ITENTH SEC, INITIAL QUANT, FOR SHAR, DEV. WAITERS 746 746 90006 QUTTY=6 ITENTH SEC, INITIAL QUANT, FOR TTY IOWS 747 . QUANT. TIMES ACCORDING TO PROCESSOR Q: 748 . QUANT. TIMES ACCORDING TO PROCESSOR Q: 750 INTERNAL RNGUNT 751 RNQUNT: 752 000010 000236 754 000010 000236 754 000015 EXP 2**060 754 000015 EXP 2**060	
745 ØØØØØ6 QUSD=0 ITENTH SEC, INITIAL QUANT, FOR SHAR, DEV. WAITERS 746 ØØØØØ6 QUTTY=6 ITENTH SEC, INITIAL QUANT, FOR STAR, DEV. WAITERS 747	
746 747 747 747 747 748 . RUANT. TIMES ACCORDING TO PROCESSOR 0: 749 INTERNAL RUGUNT 750 INTERNAL RUGUNT 751 752 000000 200236 753 000000 200236 00STAB; FXP +D30 ;P01; ONE HALF SECOND 754 2002200 200170 EXP 2*+D60 ;P02; TWO SECONDS	
747 . QUANT. TIMES ACCORDING TO PROCESSOR Q; 748 . QUANT. TIMES ACCORDING TO PROCESSOR Q; 749 INTERNAL RNGUNT 750 INTERNAL RNGUNT 751 . QUANT: 752 000414 753 . QUANT: 753 . QUANT: 754 . QUANT	
747 748 . QUANT. TIMES ACCORDING TO PROCESSOR Q: 749 750 INTERNAL RNCUNT 751 752 000414 RNQUNT: 753 000414 000000 000236 QUSTAB; FXP +D30 ;P01: ONE HALF SECOND 754 000415 000200 000170 EXP 20+060 ;P02: TWO SECONDS	
749 750 INTERNAL RNCUNT 751 752 000414 753 P00414 RNCUNT: 753 P00414 0002236 754 000415 0002170 754 000415 0002170 754 000415 0002170 754 000415 0002170	
750 INTERNAL RNGUNT 751 752 000414 RNGUNT: 753 000414 000000 000236 QQSTAB; FXP +D30 ;P01: ONE HALF SECOND 754 000415 000200 000170 EXP 20+060 ;P02: TWO SECONDS	
751 752 000414 RNOUNT: 753 000414 000000 000236 DOSTAB: FXP +D30 ;P01: ONE HALF SECOND 754 000415 000270 000170 EXP 2#+D60 ;P02: TWO SECONDS	
752 000414 RNOUNT: 753 000414 000000 000236 QQSTAB; FXP +D30 ;PQ1: ONE HALF SECOND 754 000415 000000 000170 EXP 2#+D60 ;PQ2: TWO SECONDS	
753 000414 000000 000236 105TAB: FXP +D30 ;P01: ONE HALF SECOND 754 000415 000000 000170	
754 000415 000070 007170 EXP 2*+D60 IPO2: TWO SECONDS	
יידע אואאיזער איזער איזער איזער ארא איזער איז	

.

	0033 N.	KRASINZ	AF 153.1	/ 22 MAR	00	V / (0 2)	
757				IEN ETS	WAP .	ć	
758						CAN, OSCAN	
759	000420					AN FOR INPUT	
76Ø		777756	0002321			-CMQ.QFOR	MONITOR COMMAND WHICH NEEDS CORE IMAGE IN CORE
761			000231				ILOOK FOR 1ST JORS IN SHAR, DEV QUEUES
762	000422		000231			-DAQ,QFOR1	Teon for for one in outsid all defere
763	000423	777774	000231'			-AUQ,QFOR1	
764	000424		000231			-DCQ,QFOR1	
765	000425	777767	000231'			-MTQ.QFOR1	
766			000231			-STQ,QFOR1	
767	000420	777771	000231			-DTQ.QFOR1	
768				CC		-PO1,QFOR	ISCAN PROCESSOR AS SCHEDULER DOES
				SOLAN		-PQ2,QFOR	JOLAN FRUIESDUR AS SUMEDULER DUES
769 770		777760	0002321			-PQ3,QFOR	
			000232'			-Pub,uruk	
771		000000	000000		Z		JPATCH SPACE
772			000030 		Z		
773	000435	000000	000000		7		IFINAL ZERO TO FLAG END
774	_						
775	000436			OSCANI		AN FOR OUTPU	
776			0002321				JUNRUNABLE JOBS FIRST
777			000232'		XWD		
778	001440		000235				JANY SHAR, DEV.WAITERS MORE THAN 1 DEEP
779			000235			-STQ,QBAK1	
78Ø			0002351			-MTQ,QBAK1	
781		777770	000235			-DCQ,QBAK1	
			000235			-AUQ,QBAK1	
783		777772	0002351			-DAQ,QBAK1	
784			0002351			-MOQ,QBAK1	
785		777765	ØØØ232'		XWD	-TIOWQ.QFOR	ITTY IOW
786		777757	000241			-PQ3,QBAK	
787	000451	777771	000231'			-DTQ,QFOR1	
788	000452	777775	000231'		XWD	-STQ,QFOR1	
789	000453	777767	000231'		XWD	-MTQ,QFOR1	INCH SCAN FIRST JOB IN QUEUES
79Ø	001454	77777Ø	000231'		XWD	-DCQ,QFOR1	
791	000455	777774	000231'		XWD	-AUQ,QFOR1	
792	000456	777772	000231'		XWD	-DAQ,QFOR1	
793	000457	777773	000231'		XWD	-MQQ,QFOR1	
794	000460	777760	000241'			-PQ2,QBAK	
795		777761	000241'			-PQ1,QBAK	
796	000462	000000	000000		Z	····	IPATCH SPACE
797	000463		ØØØØØØØ		7		
798	000464	000000	000000		Z		FINAL ZERO TO FLAG END
799			~~~~	>			TATAL LEVE TO LEVE LAU

SCHEDB	- SCHEDULING ALGORITHM F OCSS R. KRASIN/AF TS3.1	OR SWAPPING SYSTEM(10/50)(BURROUGHS DISK) MACRO,V36 19:10 4 7 22 Mar 68 V000	1-JUN-60	PAGE	34
821 822		SUBITL SWAP R, KRASIN/AF TS4,34 03 FEB 69 V406			
803		,SWAPPER CALLED EVERY CLOCK TIC.			
874		SINCE MOST OPERATIONS STARTED BY THE SWAPPER REQUIRE SEVERAL			
805		TICS TO RUN TO COMPLETION, SEVERAL FLAGS(FINISH, FIT, FORCE			
806		JARE USED TO "REMEMBER" PREVIOUS STATES,			
8:47		THE BASIC ALGORITHMI			
808		IS CORE SHUFFLER WAITING FOP IO TO FINISH FOR SOME JOB?			
829		; YESTRY AGAIN TO SHUFFLE (WHEN ID STOPS)			
810		IS CORE SHUFFLER STILL WAITING FOR 10 TO FINISH?			
811		; YESRETURN AND DO NOTHING			
812		JIS SWAPPER STILL BUSY?			
813		; YESRETURN AND DO NOTHING			
814		SCAN OS FOR 1ST JOB OUT OF CORE.			
815		, IF NONERETURN			
816 817		JAI JE ONE WHILL FOUNDER'S SEC SIT IN LADOEST HOLE IN CODES			
818		, IF ONEWILL LOW(HIGH) SEG FIT IN LARGEST HOLE IN CORE? , YESSTART INPUT AND RETURN			
819		, NOIS TOTAL FREE CORE(CORTAL) ENOUGH TO ACCOMMODATE LOW(HIGH)	SEG?		
82Ø		; YESCALL CORE SHUFFLER			
821		IS SHUFFLER WAITING FOR ID TO STOP?			
822		; YESRETURN AND DO NOTHING			
823		I NOGO TO AI			
824		, NO"REMBER" THIS JOB FOR INPUT AND LOOK FOR OUTPUT:			
825 826		ANY JOBS WAITING TO XPAND CORE BY SWAP OUT/IN?			
827		, YESOUTPUT ONE AND RETURN , NOSCAN QS BACKWARD FOR JOB IN CORE WHOSE PROTECT TIME			
828		, (SET ON INPUT) HAS GONE TO Ø.			
829		IE NONE-RETURN			
83Ø		, IF NONERETURN , IF ONEIS IT SWAPPABLE(NO ACTIVE IO AND NOT CURRENT JOB)? , YESOUTPUT HIGH SEC(IF ANY AND NOT ON DISK) THEN LOW SEGMENT , NOSET SWP BIT(SO SCHEDULER WILL NOT RUN), IO WILL CONTINUE , IN LOW SEGMENT AS LONG AS IT CAN			
831		YES-OUTPUT HIGH SEG(IF ANY AND NOT ON DISK) THEN LOW SEGMENT			
832		, NOSET SWP BIT(SO SCHEDULER WILL NOT RUN), IO WILL CONTINUE			
833					
834		, IO ROUTINES NO LONGER STOP IF SWP SET, JUST SHF)			
835					
836 837		EXTERNAL JETSTS			
838		EXTERNAL BIGHOL, CORTAL, ANYDEV, JBTADR, JBTSWP, KCORE1, TRYSWP			
839		EXTERNAL IMGOUT,IMGIN,FINISH,FIT,FORCE FXTERNAL OERROR,CORGET,J9TOAT,J0BDPG,J0BDPD,J0BPC			
840		EXTERNAL JETDAT, SHFWAT, CHKSHF			
841		EXTERNAL FULCAT, ERRENT, EXCALP, PCSTOP, PCORSE, VIRTAL			
842					
843		INTERNAL SWAP			
844		INTERNAL XPAND, FT2REL			
845					
846		T=DEVDAT			
847		T1=TAC1			
848	000001				
849 850	000004	J=ITEM			
851		,ALL DEVICE DEPENDENT CODE MARKED WITH A "*"			
U > #		THE DEFICE DEFENDENT CODE MARKED WITH A """			

SCHEDB					ING SYSTEM(10/50) B 69 V406	(BURROUGHS DISK) MACRO.V36 19:10 4-JUN-69 PAGE 35
852	000465	332000	203000	SWAP:	SKIPE SHEWAT	IS CORE SHUFFLER WAITING FOR IO TO STOP
853						FOR SOME JOB?
854	000466	262140	0000000		PUSHJ PDP, CHKSHF	IYES, CALL CORE SHUFFLER TO SEE IF
855						I 10 STOPPED YET
356	000467	336000	000465		SKIPN SHEWAT	IS SHUFFLER STILL WAITING?
857	000470	332000	001340		SKIPE SORED	: NOIS SWAP SERV. ROUT. STILL BUSY WITH LAST JOB?
858	000471	263140	000000		POPJ PDP,	I * YESRETURN
85 9	000472	402000	000141'		SETZM INFLG	1444 EXPERIMENTAL 444
860	000473	336200	000000		SKIPN J.FINISH	INCANY IN/OUTPUT TO FINISH?
861		254000	000547		JRST SWP2	1N0-
862	000475	321220	0005371		JUMPL J.FINOUT	IYESINPUT OR OUTPUT?
863		332000			SKIPE SERA	INPUT, ANY INPUT ERRORS?
864	000477	254000			JRST INERR	IYES
865	000500			FININØ:		NOTHING TO SWAP IN(HIGH OR LOW SEG EXPANDING FROM Ø)
866				IFN FT2		
867					EXTERN FININ	
868	000500	260140	666666			IS THERE A HIGH SEG WHICH MUST BE SWAPPED IN?
869			000561			YYES, GO SWAP IT IN(J SET TO HIGH SEG NO, JOB # IN INPJOB)
870	V. DV. JV. T	224000	000001			NO, EITHER HIGH SEG ALREADY IN FOR ANOTHER USER
871						I OR THERE IS NONE, J STILL JOB NO, (IE LOW SEG)
872						; OR J IS HIGH SEG WHICH EXPANDED FROM NOTHING(XPANDH)
873						IN WHICH CASE IT HAS NO DISK SPACE AND DIDLING ACS
874						: AND SETTING PROTECT TIME WON'T MATTER EITHER,
875				>		
876	000502	135300	ØØØØØØØ	,	LDB T, IMGIN	INEW CORE SIZE
877		135100			LDB T1, IMGOUT	JOLD SIZE WHEN ON DISK
878		274140			SUB T1.T	JOLD-NEW=DECREASE
879	1.000014	27,110	o Di Diio		30 111	; HAS USER DECREASED VIRTUAL MEMORY FROM M TO N(N GR Ø)
880						, and been dependently finites a finite of the one of the
881						; WHILE OUT ON DISK(R,RUN,GET,KJOB) TO 140 WORDS?
882						I CORE COMMAND ALWAYS FORCES SWAP IN BEFORE
883						CORE REASSIGNMENT SO NOT IN THIS CATEGORY
884						J FRAGMENTED USER TOO HARD TO PARTIALLY RECLAIM DISK SP
885				ACE		, TRIGHERIES OFEN TOO HERE TO TRATIFIET REPERTY FIGH OF
886						I ON REDUCTION WHICH DOES NOT GO TO Ø
887	ØØØ5Ø5	333000	000002		SKIPLE T1	DECREASED?
888	000506	272100	000000		ADDM T1,VIRTAL	IYES, NOW INCREASE VIRTUAL MEMORY AVAILABLE BY
889	000000	2/2100	000000		-0000 117710126	AMOUNT OF DECREASE IN HIGH OR LOW SEG
890	000507	260140	001034		PUSHJ PDP,ZERSWP	
891	100 301	2001-0	001004		FOUND FOFTZEROWE	; AND SWPISHF(JBISTS) TO 0
892	000510	135300	araara		LDB T, PCORSZ	ICOMPUTE AND SET IN CORE IN CORE PROTECT TIME FROM
893	200210	1020.0				SIZE OF JOB(1K BLOCKS=1)
893	000511	220300	900000		MUL T, PROT	ADD VARIABLE AMOUNT DEPENDING ON CORE SIZE
895	000512		000000 000000		ADD T, PROTØ	ADD FIXED AMOUNT INDEPENDENT OF CORE SIZE
896	000512	506304	000133		HRLM T.JRTSWP(J)	
897			003000		MOVE JDAT, JBTDAT	
077	10.014	2 8 7 1 4 4			CIL CORTACIONI	In the for and included that we have for

898					T 50.00 /		
890 899				IFN JUA	T-FROG, <		
930				>	MOVE PROG, JBTADR(J)		
901	000515	202307	202020	/	MOVE T.JOBPC(JDAT)	JOB STOPPED IN EXEC MODE?	
	000516	603300				ITEST PD FLAG	
903		254020	002546		JRST SWP1	INO	
904	000520	550307	007020		HRRZ T, JOBDPG (JDAT)	YES, ADJUST PROG AND PDP IN DUMP AC AREA	
925	000521	275307			SUBI T. (PROG)	JOLD RELOC-NEW RELOC	
926	000522	213000	000006		MOVNS T	INEW RELOC-OLD RELOC	
907	000523	272307	000000		ADOM T, JOBOPD (JDAT)	JADJUST NUMP POP	
908	000524	202347			MOVEM PROG, JOBOPG (JDAT)		
9719	000525	254000	000546'		JRST SWP1		
910							
911	000526	402000		INERR:	SETZM FINISH	ICLEAR FINISH FLAG SO SWAPPING CAN CONTINUE	
912	ØØØ527	200344	ØØØ172'		MOVE PROG, JBTADR(J)	ISETUP RELOC, PROTECTION FOR HIGH OR LOW SEG	
913						JDAT, JBTDAT(J)>	
914		260140			PUSHJ PDP,KCORE1	RETURN CORE	
915 916	000531 000532	265040 516570	000000		JSP TAC, ERRPNT	PRINT ON USER CONSOLE	
910	000502	5105/0	150100		ASCI7 /SWAP READ ERROR/		
	000533	512130	142100				
	000534	426452	247644				
	000535	0000000	000000				
917	000536	254000			JRST PCSTOP	ISTOP JOB AND FORCE RESCHEDULING	
	000537	213000		FINOUT:		FINISH OUTPUT, -FINISH=JOB NO.	
919	000540	332000			SKIPE SERA	IANY ERRORS	
920	000541	254000	0005751		JRST SWPREC	IYES, RECORD ERROR AND TRY AGAIN,	
921						J IN A DIFFERENT PLACE ON DISK	
922	000542	200344	0005271		MOVE PROGIJBTADR(J)	XWD PROTECT, RELOC. FOR LOW SEG	
923				IFN PROD	S-JDAT,<		
924					MOVE JDAT, JBTDAT(J)	JOB DATA AREA	
925	600547			>			
926 927	000543	260140	000530'		PUSHJ PDP,KCORE1	JRETURN CORE FOR LOW OR HIGH SEG JUST SWAPPI	ED OUT
928						I EVEN 1F	
929						ANOTHER JOB STARTED TO SHARE HIGH SEG DUR	ING
930						SWAP OUT (GET) SINCE JOB IS MARKED WITH	
931				ACK IN		I SWP BIT ON AND CANNOT RUN UNTIL HIGH SEG	13 SWAPPED
932				IFN FT2F	REL . C		
933					EXTERN FINOT		
934	000544	260140	ØØØØØØ		PUSHJ PDP, FINOT	IS THIS A HIGH SEG WHICH WAS JUST SWAPPED (ÓUT?
935	000545	254000	0006621		JRST FORCEL	IYES, J SET TO LOW SEG NO, GO TRY SWAP IT DI	
936						I NO, THIS WAS A LOW SEG, ALL SWAPPING FOR	
937						J IS FINISHED,	'
938				>			
			003526	-	SETZM FINISH	ICLEAR FINISH FLAG	
940		332200		SWP2:	SKIPE J.FORCE	IWAITING FOR JOB TO BECOME SWAPPABLE?	
941	000550		000663'		JRST FORCE1	IYES	
942	000551	332200	ØØØØØØ		SKIPE J.FIT	IND WAITING TO FIT JOB IN CORE?	

SCHEDB - SCHEDULING ALGORITHM FOR SWAPPING SYSTEM(10/50)(BURPOUGHS DISK) MACRC, V36 19:10 4-JUN-69 PAGE 37 SWAP R. KRASIN/AF TS4,34 03 FEB 69 V405 944 ,SCAN FOR INPUT 945 900553 201240 0004201 MOVEL DAT, ISCAN JSP TAC, OSCAN 946 200554 265040 0002251 947 000555 254000 0006061 JRST CHKXPN INO INPUT TO DO--CK FOR EXPANDING JOBS ITHIS JOP OUT OF CORE? ISWP ON IF HIGH SEG SWAPPED OUT FOR THIS USER 948 000556 200304 000221' MOVE T, JRTSTS(J) 000557 607370 949 TLNN T, SWP 302000 95Ø ; OR BOTH SEGS SWAPPED OUT 951 952 000560 254022 000020 JRST (TAC1) INO--CONTINUE SCAN 953 000561 202200 000551' FIT1: 200562 135640 000502' ;REMEMBER JOR(OR HIGH SEG) TRYING TO FIT IN ;CORE SIZE NEEDED FOR THIS SEG(@ IF LOW SEG ; OR HIGH SEG WITH UWP OFF ALREADY IN CORE) 954 MOVEM J,FIT 955 LDB AC1. INGIN 956 957 IFE FT2REL .< 958 CAMLE AC1, CORTAL WILL LOW SEG FIT IN FREE+DORMANT CORE? 959 > 960 IFN FT2REL . < 961 EXTERN FITSIZ 962 000563 260140 000000 PUSHJ PDP, FITSIZ SCOMPUTE AMOUNT OF CORE NEEDED TO BRING IN 1 1. THIS JOBS LOW SEG AND HIGH SEG 1 2. THIS JOBS LOW SEG(HIGH ALREADY IN OR NONE) 1 3. THIS HIGH SEG BECAUSE LOW SEG ALREADY IN 1 3. THIS HIGH SEG BECAUSE LOW SEG ALREADY IN 1 3. THIS HIGH SEG BECAUSE LOW SEG ALREADY IN 1 3. THIS HIGH SEG BECAUSE LOW SEG ALREADY IN 1 3. THIS HIGH SEG BECAUSE LOW SEG ALREADY IN 1 3. THIS HIGH SEG BECAUSE LOW SEG ALREADY IN 1 4. THIS JOBS LOW SEG ALREADY IN 1 5. THIS HIGH SEG BECAUSE LOW SEG ALREADY IN 1 5. THIS HIGH SEG BECAUSE LOW SEG ALREADY IN 1 5. THIS HIGH SEG BECAUSE LOW SEG ALREADY IN 1 5. THIS HIGH SEG BECAUSE LOW SEG ALREADY IN 1 5. THIS HIGH SEG BECAUSE LOW SEG ALREADY IN 1 5. THIS HIGH SEG BECAUSE LOW SEG ALREADY IN 1 5. THIS HIGH SEG BECAUSE LOW SEG ALREADY IN 1 5. THIS HIGH SEG BECAUSE LOW SEG ALREADY IN 1 5. THIS HIGH SEG BECAUSE LOW SEG ALREADY IN 1 5. THIS HIGH SEG BECAUSE LOW SEG ALREADY IN 1 5. THIS HIGH SEG BECAUSE LOW SEG ALREADY IN 1 5. THIS HIGH SEG BECAUSE LOW SEG ALREADY IN 1 5. THIS HIGH SEG BECAUSE LOW SEG ALREADY SEG ALREADY IN 1 5. THIS HIGH SEG BECAUSE LOW SEG ALREADY SEG ALREADY SEG ALREADY IN 1 5. THIS HIGH SEG BECAUSE LOW SEG ALREADY SEG ALREA 963 964 965 966 IWILL LOW SEG FIT IN FREE+DORMANT+IDLE CORE? 967 INO,WILL NOT FIT EVEN IF ALL DORMANT SEGS DELETED I AC1=TOTAL CORE NEEDED(IN K) IYES, WILL THIS SEG FIT IN BIGGEST HOLE OF FREE CORE I WITHOUT DELETEING ANY DORMANT OR IDLE SEGS? I (AC1 RESTORED TO SIZE FOR JUST THIS LOW OR HIGH SEG) 968 000564 254020 000610' JEST SCHOUT 969 970 002565 317640 000000 CAMG AC1, BIGHOL 971 972 973 974 000566 254030 0007721 JRST SWAPI IYES, GO SWAP IN THIS LOW OR HIGH SEG IFN FT2REL, < EXTERN FRECR1, HOLEF 975 976 977 000567 336000 aaaaaa SKIPN HOLEF IND, ARE THERE ANY HOLES IN CORE WHICH THE SHUFFLER 978 ; COULD ELIMINATE(NOT COUNTING ONE AT TOP)? INO, GO DELETE ONE DORMANT SEG IN CORE I AND ALWAYS SKIP RETURN(THERE MUST BE AT LEAST 000570 260140 979 000000 PUSHJ PDP, FRECR1 980 981 ; ONE, OTHERWISE CORTAL=BIGHOL)MONITOR ERROR IF NONE 982 > 983 000571 000466' 260140 PUSHJ PDP, CHKSHF IYES, CALL CORE SHUFFLER TO MOVE ONE SEG DOWN 984 336000 000572 0004671 SKIPN SHFWAT SHUFFLER WAITING FOR 10 TO STOP? 985 000573 254000 000551' JRST FITØ 986 000574 263140 000000 POPJ PDP, JYES, RETURN AND WAIT TILL ID STOPS

SCHEDB					ING SYSTEM(10/50)(BURR) 3 69 V406	DUGHS DISK) MACRO.V36 19:10 4-JUN-	-69 PAGE 38
987	-			EXTERN	VIRTAL, SWPERC		
988							
989	000575	200040		SWPREC:		IERROR FLAGS	
99Ø	000576	436040	000000		IORM TAC,SWPERC	ISAVE FOR POSTERITY	
991	PØØ577	135040	000503		LDB TAC,IMGOUT	JDECREASE TOTAL AMOUNT	
992	000600	213000	000001		MOVNS TAC	JOF VIRTUAL CORE IN THE MACHINE	
993	000601	272040	000506'		ADDM TAC, VIRTAL	BY THE AMOUNT BEING GIVEN UP	
994	000602	135040	000577 '		LDB TAC,IMGOUT		
995	000603	661040	000001		TLO TAC.1		
996	000604	272Ø4Ø	000576		ADDM TAC, SWPERC		
997	ØØØ6Ø5	254000	000670'		JRST SWAPO	JGO TRY AGAIN	
998							
999							
1000					JT TO DO, CHECK FOR EX		
1001	000606	337000		CHKXPNI	SKIPG XJOB	IANY JOBS TO EXPAND?	
1002	ØØØ607	26314Ø	000000		POPJ PDP,	ING, RETURN FROM SWAPPER, NOTHING TO IN	
1003						; YES, FALL INTO SCHOUT WHICH WILL SWAF	OUT EXPANUL
1004						JOB SINCE THERE IS ONE	
1005				; INPUT	TO DO, CHECK TO SEE IF	ANY JOBS JUST HAPPEN TO WANT TO EXPAND	
1006					EXTERN HIGHJB, JBTSTS,	ERROR, MAXSIZ, MAXJBN, SUMCOR	
1007	000610	337000	000323'	SCNOUT	SKIPG XJOB	IANY JOBS WAITING TO EXPAND?	
1008	000611	254000	000624'		JRST SCNJOB	INO, SCAN ALL JOBS IN PRIORITY ORDER LC	DOKING
1009						I FOR ONE TO SWAP OUT	
1010	000612	200200	ØØØØØØ		MOVE J.HIGHJB	IYES, START WITH HIGHEST JOB NUMBER ASS	SIGNED
1011	000613	205300	000001		MOVSI T,JXPN	SETUP JOB EXPANDED BIT	
1012	000614	616304	000556'		TDNN T, JBTSTS(J)	IS THIS JOB EXPANDING?	
1013	000615	367200	000614'		SOJG J1	IND, KEEP LOOKING	
1014					IFN FTRCHK.<		
1015	000616	327200	000621'		JUMPG J.SCNOK		
1016	000617	402000	ØØØ323'		SETZM XJOB	ICLEAR XJOB SO MESSAGE WILL PRINT	
1017	ØØØ62Ø	26524Ø	000176'		JSP DAT,ERROR	JERROR IF NONE FOUND	
1018				>			
1019	ØØØ621	370000	000323		SOS XJOB	DECREMENT COUNT OF EXPANDING JOBS	
1020	ØØØ622 ØØØ623	412304 254000	ØØØ614' ØØØ655'		ANDCAM T,JBTSTS(J) JRST FORCEØ	ICLEAR EXPAND BIT IN JOB STATUS WORD IGO TRY TO SWAP JOB OUT	
1021							

SCHENB	- SCHEDU Swap R	LING ALC. KRASIN	GORITHM F Maf ts4.3	OR SWAPP 4 23 Fe	ING SYSTEM(10/50)(BURRO) 8 69 - V406	JGHS DISK) MACRO,V36 19:10 4-JUN-69 PAGE 39
1222 1223 1224 1325 1326				JUST L	N K) NEEDED TO GET THIS	FR TO MAKE ROOM FOR JOB TO COME IN USER IN CORE IS IN AC1(FITSIZ) OR HIGH ALREADY IN, JUST HIGH IF LOW ALREADY IN, D IN
1027 1028		202320 202320	803020 003230	SCV108:	MOVE T,CORTAL MOVEM T,SUMCOR	INITIALIZE FREE CORE COUNTER
1029	000626	402070	000000		SETZM MAXSIZ	JCLEAR SIZE OF LARGEST JOB
1030 1031	000627 000630	201240	000436' 000225'		MOVEL DAT,OSCAN JSP TAC,OSCAN	ISCAN ALL JORS RANKED IN PRIORITY TO BE SWAPPED OUT
1032	000631	254020	000714		JRST NOFIT	INC MORE JOBS LEFT, CANNOT FIT JOB IN CORE
1033	000632	316270	000561 .		CAMN J.FIT	IS THIS JOB WE ARE TRYING TO FIT IN?
		254002			JRST(TAC1)	IYES, GO FIND NEXT JOB TO OUTPUT
	020634		202622		SKIPGE T,JBTSTS(J)	JOB RUN BIT STILL ON(JOB STILL WANT TO RUN)?
1036 1037	000635	33520194	0005131		SKIPGE JBTSWP(J)	IYES, IS PROTECT TIME STILL LEFT?
1038						I PROTECT TIME IS DECREMENTED ONLY WHEN
1039						F A JOB IS RUNABLE, SO LOOK AT IT F ONLY IF RUN BIT STILL ON
1040					TLNE TINSWP+SWP	ING, IS THIS JOB NOT TO BE SWAPPED OR ALREADY SWAPPED?
1041	200636	6033:0	012000			THO, TO THIS SUB NOT TO BE SWAFFED OR ALREADY SWAPPEDY
1042						I (DISPLAY, REAL TIME)?
1043	200637	254072	000200		JRST (TAC1)	IYES, CONTINUE SCAN TO FIND ANOTHER
1344	C02640	554304	000542'		HLRZ T, JATADR(J)	PICK UP SIZE OF JOB
1246	000641	322302	ØØØ220		JUMPE T, (TAC1)	CONTINUE SCAN IF NOT IN CORE (HIGH SEG ALREADY SWAPPED
1047	1.01041	0220-2	100200			L OUT FOR THIS HOFE IS NO LOW OFFICE
1048	000642	240300	777766		ASH 1,-12	; OUT FOR THIS USER IF NO LOW SEG; ;convert to 1K blocks
1049	000643	271320	000001		ADDI T.1	JOONTENT TO IN DECOMS
1050				IFN FT2P		
1051 1052	000444	260140	<i></i>		EXTERN FORSIZ	
1053	100044	200140	******		PUSHJ POP,FORSIZ	IINCREASE SIZE(T) BY HIGH SEG IF THIS JOB
1054				>		; IS ONLY ONE IN CORE USING HIGH SEG(J= JOB # STILL)
1055	000645	317300	0006261		CAMG T.MAXS17	LARGEST SO FAR?
	000646	254000	000651		JRST FORCE2	IND
1057	000647	202300	0006451		MOVEM T, MAXSIZ	IYES, SAVE SIZE
1058	909650	202270	000000		MOVEM J.MAXJBN	IAND JOB NUMBER
1059 1060	000651 000652	272320 313640	000625		ADDM T, SUMCOR	IADD TO TOTAL
	000653	254002	000651' 000000		CAMLE AC1,SUMCOR JRST (TAC1)	FOUND ENOUGH CORE FOR JOB TO BE FIT IN?
1062	000654	200200	000650'		MOVE JAMAXJBN	JNG. LOOK FOR MORE JYES, SWAP OUT LARGEST
						TEST OWAR OUT LANGEST

1064 00 1065 1066 1067 1068 1069 1070 00 1071 1072 1074 1074 1075 1074 1075 1076 1077 1078 1078 1078 1078 1078 1080 1080	00656	254072	007230	IFN FT2RF	PUSHJ PDP,TRYSWP JRST (TAC1) FL.K Extern forhgh Pushj PDP,forhgh	 ING, NSWP OR NSHF SET(DISPLAY, REAL TIME) OR SAVE OR GET IN PROGRESS WITH DEVICE STILL AC LOOK FOR AN OTHER JOB TO SWAP IIS THERE A HIGH SEG TO BE WRITTEN BEFORF TRYING TO SWAP OUT LOW SEGMENT? WRITE HIGH SEG IF ALL OF THE FOLLOWING ARE TF I, JOB HAS A HIGH SEG AND I THAS NOT BEEN SWAPPED FOR THIS USFR (SWP=Ø FOR JOB) IT IS IN CORE(NOT XPANDH) I FIS ONLY USER USING IN CORE HIGH SEG NOT ON DISK YET 6. THIS HIGH SEG IS NOT THE SAME ONE AS JOB BEING FITTED IN IS GOING TO WANT RETURN HIGH SEG NO. IN J IF YES, OTHERWISE
1066 1067 1068 1069 1070 00 1071 1072 1073 1074 1075 1076 1075 1076 1077 1078 1079 1080 1081 1083 1084 1085 1086 1087 00 1089 1090 1090	ØØ657	260140		5	EXTERN FORHGH	IS THERE A HIGH SEG TO BE WRITTEN BEFORF TRYING TO SWAP OUT LOW SEGMENT? WRITE HIGH SEG IF ALL OF THE FOLLOWING ARE TF 1. JOB HAS A HIGH SEG AND 2. IT HAS NOT BEEN SWAPPED FOR THIS USFR (SWP=Ø FOR JOB) 3. IT IS IN CORE(NOT XPANDH) 4. IF IN-CORE COUNT IS EXACTLY 1 MEANING THIS ONLY USER USING IN CORE 5. HIGH SEG NOT ON DISK YET 6. THIS HIGH SEG IS NOT THE SAME ONE AS JOB BEING FITTED IN IS GOING TO WANT RETURN HIGH SEG NO. IN J IF YES, OTHERWISE
1068 1069 1070 00 1071 1072 1073 1074 1075 1076 1077 1078 1079 1080 1081 1083 1084 1085 1086 1087 00 1088 1088 1088 1089 1090 1091 1092 000	ØØ657	260140		5	EXTERN FORHGH	<pre>; TRYING TO SWAP OUT LOW SEGMENT? ; WRITE HIGH SEG IF ALL OF THE FOLLOWING ARE TF ; JOB HAS A HIGH SEG AND ; IT HAS NOT BEEN SWAPPED FOR THIS USFR ; (SWP=0 FOR JOB) ; 3. IT IS IN CORE(NOT XPANDH) ; 4. IF IN-CORE COUNT IS EXACTLY 1 MEANING ; THIS ONLY USER USING IN CORE ; 5. HIGH SEG NOT ON DISK YET ; 6. THIS HIGH SEG IS NOT THE SAME ONE AS JOB ; BEING FITTED IN IS GOING TO WANT ; RETURN HIGH SEG NO. IN J IF YES, OTHERWISE</pre>
1069 1070 00 1071 1072 1073 1074 1075 1076 1077 1078 1079 1080 1079 1080 1081 1083 1084 1085 1086 1087 00 1089 1090 1091 1092 000	ØØ657	260140		5	EXTERN FORHGH	<pre>; TRYING TO SWAP OUT LOW SEGMENT? ; WRITE HIGH SEG IF ALL OF THE FOLLOWING ARE TF ; JOB HAS A HIGH SEG AND ; IT HAS NOT BEEN SWAPPED FOR THIS USFR ; (SWP=0 FOR JOB) ; 3. IT IS IN CORE(NOT XPANDH) ; 4. IF IN-CORE COUNT IS EXACTLY 1 MEANING ; THIS ONLY USER USING IN CORE ; 5. HIGH SEG NOT ON DISK YET ; 6. THIS HIGH SEG IS NOT THE SAME ONE AS JOB ; BEING FITTED IN IS GOING TO WANT ; RETURN HIGH SEG NO. IN J IF YES, OTHERWISE</pre>
1071 1072 1073 1074 1075 1076 1076 1077 1078 1079 1080 1081 1082 1083 1084 1085 1086 1087 006 1088 006 1089 1090 1092 000	ØØ657	260140	803020			<pre>; TRYING TO SWAP OUT LOW SEGMENT? ; WRITE HIGH SEG IF ALL OF THE FOLLOWING ARE TF ; JOB HAS A HIGH SEG AND ; IT HAS NOT BEEN SWAPPED FOR THIS USFR ; (SWP=0 FOR JOB) ; 3. IT IS IN CORE(NOT XPANDH) ; 4. IF IN-CORE COUNT IS EXACTLY 1 MEANING ; THIS ONLY USER USING IN CORE ; 5. HIGH SEG NOT ON DISK YET ; 6. THIS HIGH SEG IS NOT THE SAME ONE AS JOB ; BEING FITTED IN IS GOING TO WANT ; RETURN HIGH SEG NO. IN J IF YES, OTHERWISE</pre>
1072 1073 1074 1075 1076 1077 1078 1079 1080 1081 1083 1084 1085 1086 1087 006 1088 006 1089 1090 1091 1092 000						<pre>; TRYING TO SWAP OUT LOW SEGMENT? ; WRITE HIGH SEG IF ALL OF THE FOLLOWING ARE TF ; JOB HAS A HIGH SEG AND ; IT HAS NOT BEEN SWAPPED FOR THIS USFR ; (SWP=0 FOR JOB) ; 3. IT IS IN CORE(NOT XPANDH) ; 4. IF IN-CORE COUNT IS EXACTLY 1 MEANING ; THIS ONLY USER USING IN CORE ; 5. HIGH SEG NOT ON DISK YET ; 6. THIS HIGH SEG IS NOT THE SAME ONE AS JOB ; BEING FITTED IN IS GOING TO WANT ; RETURN HIGH SEG NO. IN J IF YES, OTHERWISE</pre>
1073 1074 1075 1076 1077 1078 1079 1080 1081 1082 1083 1084 1085 1086 1085 1088 000 1088 1089 1090 1091 1092 000						 WRITE HIGH SEG IF ALL OF THE FOLLOWING ARE THE JOB HAS A HIGH SEG AND IT HAS NOT BEEN SWAPPED FOR THIS USFR (SWP=Ø FOR JOB) IT IS IN CORE(NOT XPANDH) IF IN-CORE COUNT IS EXACTLY 1 MEANING THIS ONLY USER USING IN CORE HIGH SEG NOT ON DISK YET IT HIS HIGH SEG IS NOT THE SAME ONE AS JOB BEING FITTED IN IS GOING TO WANT RETURN HIGH SEG NO. IN J IF YES, OTHERWISE
1074 1075 1076 1077 1078 1079 1080 1081 1082 1083 1084 1085 1084 1085 1086 1087 000 1088 000 1099 1090 1092 000						 1, JOB HAS A HIGH SEG AND 2. IT HAS NOT BEEN SWAPPED FOR THIS USFR (SWP=Ø FOR JOB) 3. IT IS IN CORE(NOT XPANDH) 4. IF IN-CORE COUNT IS EXACTLY 1 MEANING THIS ONLY USER USING IN CORE 5. HIGH SEG NOT ON DISK YET 6. THIS HIGH SEG IS NOT THE SAME ONE AS JOB BEING FITTED IN IS GOING TO WANT RETURN HIGH SEG NO. IN J IF YES, OTHERWISE
1075 1076 1077 1078 1079 1080 1081 1082 1083 1084 1085 1086 1087 006 1089 1090 1090 1092 000						<pre>(SWP=Ø FOR JOB) 3. IT IS IN CORE(NOT XPANDH) 4. IF IN-CORE COUNT IS EXACTLY 1 MEANING 5. THIS ONLY USER USING IN CORE 5. HIGH SEG NOT ON DISK YET 6. THIS HIGH SEG IS NOT THE SAME ONE AS JOB 5. BEING FITTED IN IS GOING TO WANT 5. RETURN HIGH SEG NO. IN J IF YES. OTHERWISE</pre>
1076 1077 1078 1079 1080 1081 1083 1083 1085 1086 1085 1086 1087 000 1089 1090 1092 000						; 3. IT IS IN CORE(NOT XPANDH) ; 4. IF IN-CORE COUNT IS EXACTLY 1 MEANING ; THIS ONLY USER USING IN CORE ; 5. HIGH SEG NOT ON DISK YET ; 6. THIS HIGH SEG IS NOT THE SAME ONE AS JOB ; BEING FITTED IN IS GOING TO WANT ; RETURN HIGH SEG NO. IN J IF YES, OTHERWISE
1077 1078 1079 1080 1081 1082 1083 1084 1085 1086 1085 1086 1088 000 1089 1090 1091 1092 000						 4. IF IN-CORE COUNT IS EXACTLY 1 MEANING THIS ONLY USER USING IN CORE 5. HIGH SEG NOT ON DISK YET 6. THIS HIGH SEG IS NOT THE SAME ONE AS JOB BEING FITTED IN IS GOING TO WANT RETURN HIGH SEG NO. IN J IF YES, OTHERWISE
1078 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1088 1089 1090 1090 1092 000						; THIS ONLY USER USING IN CORE ; 5. HIGH SEG NOT ON DISK YET ; 6. THIS HIGH SEG IS NOT THE SAME ONE AS JOB ; BEING FITTED IN IS GOING TO WANT ; RETURN HIGH SEG NO. IN J IF YES, OTHERWISE
1079 1080 1081 1083 1083 1084 1085 1086 1087 000 1089 1090 1090 1092 000						; 5. HIGH SEG NOT ON DISK YET ; 6. THIS HIGH SEG IS NOT THE SAME ONE AS JOB ; BEING FITTED IN IS GOING TO WANT ; RETURN HIGH SEG NO. IN J IF YES, OTHERWISE
1080 1081 1082 1083 1084 1085 1086 1087 0086 1087 000 1089 1090 1090 1092 000						; 6. THIS HIGH SEG IS NOT THE SAME ONE AS JOB ; BEING FITTED IN IS GOING TO WANT ; RETURN HIGH SEG NO. IN J IF YES, OTHERWISE
1081 1082 1083 1084 1085 1086 1087 1088 1088 1089 1090 1090 1091 1092 000						BEING FITTED IN IS GOING TO WANT RETURN HIGH SEG NO. IN J IF YES, OTHERWISE
1083 1084 1085 1086 1087 000 1088 000 1089 1090 1091 1092 000						RETURN HIGH SEG NO. IN J IF YES, OTHERWISE
1084 1085 1086 1087 001 1088 000 1089 1090 1091 1092 000						
1085 1086 1087 000 1088 000 1089 1090 1091 1092 000						I RETURN LOW SEG NO.
1086 1087 000 1088 000 1089 1090 1091 1092 000						I IF JOB JUST HAS LOW SEG, SHE BIT IS SET IN JE
1087 000 1088 000 1089 1090 1091 1092 000				>		FOR JOB SO IO WILL STOP NEXT BUFFER
1088 000 1089 1090 1091 1092 000	00440	205300	002000		OVEL T CHELLER FTODEL	
1089 1090 1091 1092 000		436324		۲. ۲.	OPM T INTSTOCIN	LICSHF> ISET SWAPPED OUT BIT FOR LOW OR HIGH SEG ISCHEDULER WILL NO LONGER RUN THIS JOB
1090 1091 1092 000	0001	400004	000034	1	0.44 1,081313(0)	I SET SHE BIT IF ONE SEG SOFTWARE, SO IO WILL
1091 1092 000						STOP AFTER NEXT BUFFERFUL.
1092 000						J OTOT AFTER NEXT BOFFERFOL
1//03	00662	202270	000547'	FORCEL: M	OVEM J.FORCE	ASSUME NOT SWAPPABLEIS IT?
1070						THE PROPERTY OF THE TO THE
1094 000	00663			FORCE1:		
1095				IFN JDAT-		
1096				м	OVE JDAT, JBTDAT(J)	
1097				>		
	00663	336344	000640	S	KIPN PROGIJATADR(J)	ILOC, IN PHYSICAL CORE, IS CORE
1099			0.0.0.0.0.0.0			1 ASSIGNED IN MEMORY?
		254000	000670		RST SWAPO	INC, CANNOT HAVE ACTIVE DEVICES
1101 000 1132	20665	312200	0000221	С	AME J.JOB	JIF THIS IS CURRENT JOB, WAIT UNTIL
	33111	268440	000000	-		I PROTECTED AREA IS MOVED BACK TO JOB DATA AREA
		260140 263140	0000000 000000		USHJ PDP, ANYDEV OPJ PDP.	IANY ACTIVE DEVICES? (2ND HALF OF ANYACT ROUT.)

.

HENB	- SCHEDU SWAP R.	LING ALG KRASIN/	AF TS4.3	DR SWAPP 4 2°3 Fe	NG SYSTEM(10/57)(BURG B 69 V406	OUGHS DISK) MACRO,V36 19:10 4-JUN-69 PAGE
1105				ISWAP C	OUT LOW OR HIGH SEGMENT	
1176						
1127				INTERNA	L FTTRACK	
1108						
1109	000670			SWAPO:		
111Ø				IFN FTT	RACK, KEXTERN LASOUT	
1111					MOVEM J,LASOUT	ISAVE LAST SWAP OUT FOR DEBUGGING ONLY
1112				>		
1113			000662		SETZM FORCE	ICLEAR FORCE FLAG
1114	000671	554394	000663		HLRZ T, JRTADR(J)	COMPUTE CORE IMAGE
1115	000672	322330	000546'		JUMPE T,SWP1	IDONT OUTPUT IF Ø CORE(IMGOUT ALREADY SET TO 2
1116						; WHEN CORE WAS RETURNED
1117						
1118	ØØØ673		000671'		HRRZ T1, JBTADR(J)	
1119	000674	212300	000001		MOVNM T.T2	ISAVE COUNT FOR CALL TO SCOUT
1120	000675	240300	777766		ASH T,-+D10	CONVERT TO 1K BLOCKS
1121	ØØØ676	27130Ø	000001		ADDI T,1	
1122	ØØØ677		0006021		DPB T, IMGOUT	FRECORD AS OUT IMAGE
1123	000700	505131	777777		HRLI $T1, -1(T2)$	STEUILD AND SAVE IOWD FOR SQOUT
1124	000701	261140	000002		PUSH PDP, T1	
1125	000702		000562'		LDB DAT, IMGIN	HAS SIZE OF CORE NEEDED WHEN NEXT SWAPPED IN
1126	000703	336020	009095		SKIPN DAT	JALREADY BEEN SET (XPAND)
1127			000702		DPB T, IMGIN	INO, SO SET TO # 1K BLOCKS OF CORE NEEDED
1128	aøa705		000006		MOVE DAT,T	I*CONVERT CORE IMAGE TO 128 WD BLOCKS
1129			001206		PUSHJ PDP, GXSAT	; *GET DEVICE STORAGE
1130	000707		000717'		JRST FULL	J*NONE AVAILABLE
1131			000635		HRLM TAC, JBTSWP(J)	J*SAVE DEVICE ADDRESS
1132	000711	212270	000546'	OUTP2:	MOVNM J,FINISH	JDISK SWAP SPACE ASSIGNED, NOW SET FINISH FLAG
1133						; SO THAT SWAPPER WILL KNOW WHICH SEG FINISHE
1134		_				; WHEN IN COMPLETED(SQREQ BECOMES ZERO)
1135		262140			POP PDP,TAC1	J*GET IOWD
1136	000713	254020	001066'		JRST SOOUT	F*START OUTPUT AND RETURN
1137 1138	000714	402030	0006321	NOFIT	SETZM FIT	FORGET ABOUT FITTING IN A JOB ON DISK
1139	000715		000141'		SETOM INFLG	1444 EXPERIMENTAL *** MARK DESIRE TO INPOT
1140	000710	263140	0000000		POPJ PDP,	JALL JOBS IN CORE ARE HIGHER PRIORITY.

•

1142				COME H	ERE WHEN	THE AMOUNT	OF SPACE NEEDED ON THE DISK	
$1143 \\ 1144$				IS NUT	AVAILAB	LE IN ONE C	CONTIGUOUS BLOCK	
1144					EXTERN	GETECR		
1145					ENTENN			
1147	000717	506240	000017	FULL:	HRLM	DAT,AC3	ISAVE DAT (LARGEST AVAILABLE HOLE)	
1148	000720	260140	000755'		PUSHJ	PDP, FULCOR	IGET 4 FREE CORE LOCS	
1149	000721	554240	000017		HLRZ	DAT, AC3	RESTORE DAT	
1150	000722	200740	000072		MOVE	AC3,TAC1	ILOC OF 1ST FREE CELL	
1151	000723	505740	777774		HRLI	AC3,-4	14 LOCS	
1152	000724	660100	400000		TRO	TAC1, FRGSE	G JLIGHT FRAGMENTED BIT	
1153	000725	506104	0007101		HRLM	TAC1, JBTSW	P(ITEM) ISAVE LOC OF TABLE IN JETSWP	
1154	000726	261140	000035	FULL1:	PUSH	PDP, DAT	ISAVE AMOUNT OF SPACE BEING REQUESTED	
1155	000727	260140	0012061	FULL1A:	PUSHJ	PDP, GXSAT	IGET SOME SWAPPING SPACE	
1156	000730	254000	0007421		JRST	FULL2	CANT HAVE THAT MUCH	
1157	000731	542057	ØØ0000Ø		HRRM	TAC, (AC3)	ISAVE LOC OF THE DISK SPACE	
1158	000732	262140	000005		POP	PDP,DAT	IRESTORE AMT GOTTEN	
1159		506257	009000		HRLM	DAT, (AC3)	ISAVE AMOUNT IN TABLE	
1150	000734	274300	000035		SUB	T,DAT	IAMOUNT STILL NEEDED	
1161	000735	322300	000751		JUMPE	T,FULSET	ITHROUGH IF NEED Ø K NOW	
1162	000736	260140	000760'		PUSHJ	PDP, 8MPAC3		. :
1163	000737	200240	000006		MOVE	DAT, T	ITRY TO GET ALL WE NEED NOW IN 1 CHUN	
1164	000740	254000	0007261		JRST	FULL1		
1165	000741	200240	000006	FULL18:	MOVE DA	.T,T. 3R	RESET AMOUNT OF SPACE NEEDED	
1166								
1167							THE CHUNK REQUESTED	
1168	ØØØ742	202243	000000	FULL2:		DAT, (PDP)	DAT HAS LARGEST CHUNK AVAILABLE	
1169	000743	327240	ØØØ727'		JUMPG	DAT, FULL1A		
117Ø				IFN FT2			; -1=Ø MEANS NO MORE LEFT	
1171					EXTERN	I ALOMP		
1172	000744	260140				DP, FRESWP	ITRY TO DELETE AN UNUSED HIGH SEG FROM	DISK
1173	000745	254000	000741'		JRST FL	JL[18	FOUND ONE, TRY AGAIN, J PRESERVED	
1174							I NONE FOUND, PRINT MONITOR ERROR	
1175				>				

	SWAP R.	KRASINA	AF TS4.3	4 243 FE	B 69 V	406	GPS DISK) MACRO,V36 19:10 4-JUN	
1176					EXTERN	CFRROR		
1177								
1178	900746	262140	000001		POP	PDP, TAC	WHAT? NONE LEFT?	
1179	000747	262140	000001		606 606		ISET POP TO RIGHT VALUE	
1190	000750	265240	007200		JSP	DAT.CERROR	FERROR IN MONITOR AT	
1181					•		Printer In Continue at 11111	
1182				HERE W	HEN THE	TOTAL AMOUNT OF	SPACE NEEDED HAS BEEN OBTAINED	
1183	000751	260140	0007601	FULSETI	PUSHJ	PDP, PMPAC3	ISTEP TO NEXT (LAST) TABLE LOCATION	
1184	ØØØ752	402017	<i>©00000</i>		SETEM	(AC3)	JZERO MEANS END OF TABLE	
1185	000753	554044	0007251		HLRZ	TAC, JBTSWP(ITEM)	LOC OF TABLE OF FRAGMENTS	
1186	000754	2540 40	000711'		JRST	CUTP2	IGO START OUTPUT	
1187								
1188				HERE T	O GET 4	LOCS OF FREE CORE		
1189	000755	261140	000004				IGETFOR USES ITEM	
119Ø	000756	260140	000000		PUSHJ	PDP, GETFCR	IGET 4 CELLS	
1191	000757	254022	000030		JRST I		RETORE ITEM AND RETURN	
1192							The fear free had actions	
1193				STEP A	C3 TO NI	EXT LOC OF TABLE E	REING BUILT	
1194	202760	253740	0031571	RMPAC3:	AGBUN		IOK IF MORE LOCS OF TABLE	
1195	000761	260140	Ø01755'		PUSHJ	PDP.FULCOR	IGET 4 MORE LOCS	
1196	ØØØ762	505740	777774		HRLI	AC3,-4		
1197	000763	306117	002000		CAIN	TAC1.(AC3)	ARE THEY CONTIGUOUS?	
1198	000764	263140	003000		POPJ		IYES. RETURN	
1199	700765	200257	777777		MOVE		IND. CONVERT LAST GOOD LOC	
1270	000766	562117	777777		HRPCM		ITO A POINTER TO NEXT PART OF TABLE	
1271	000767	202042	0000000		MOVEM	TAC, (TAC1)	ISTORE GOOD DATA IN 1ST WD OF NEW PART	
1272					-			
1273	000770	540740	000022		HRR	AC3,TAC1	INEW TABLE LOC	
1224	000771	253740	0007601				COUNT WORD AND RETURN	

•

HEUR					IB 69 V426	GHS DISK) MACRO.V36 19:10 4-JUN-69	FAGE 4
1235							
1296				SWAP	IN A JOB OF HIGH SEGMENT		
1277							
1238	001772			SWAP1:			
1229							
1210				IFN FT	TRACK, KEXTERN LASIN		
1211					MOVEM J.LASIN	SAVE LAST SWAP IN FOR DEBUGGING ONLY	•
1212				>			
1213							
1214		202200				SET FINISH FLAG TO INPUT	
1215	000773	402010			SETZM FIT	ICLEAR FIT FLAG	
1216	000774	135040	000704		LDB TAC, IMGIN	ISIZE OF CORE TO BE ASSIGNED WHEN SWAPPED	IN (IN
1217	000775	242040	ØØØØ12		LSH TAC, +D10	ICONVERT TO HIGHEST ADR	
1218	000776	275040	000001		SUBI TAC,1	;-1 FOR CALL TO CORGET	
1219	000777		0006731		SKIPE PROG,JBTADR(J)	IIS (LOW)SEG ALREADY IN CORE?	
1220	001000	254000	000590		JRST FININØ	IYES, POSSIBLE IF THIS IS LOW SEG AND ONLY	Y
1221						I HIGH SEG WAS SWAPPED OUT.	
		260140			PUSHJ PDP,CORGET	INC, GET CORE FOR LOW OR HIGH SEG	
1223	001002	265240	007070		JSP DAT: DERROR	INCT AVAILABLE-SHOULD NEVER HAPPEN(TELL OF	PÉR)
1224							
1225	_		-	IFN FT2	REL, KEXTERN FITHGH		
1226	001003	260140	000270		PUSHJ PDP,FITHGH	INCREASE INCORE COUNT FOR THIS JOB'S HIGH	I SEG.
1227			_	>			
1228		135300	0026771			IGET OUTPUT IMAGE	
1229	001005	322320	000500		JUMPE T,FININØ	JDONT INPUT IF OUT IMAGE IS Ø	
1230	001006	135100	000774		LDB TAC1, IMGIN JIS SIZ	E OF CORE SMALLER THAN DISK SPACE?	
1231	ØØ1007	315100	0000006		CAMGE TAC1,T	JWELL?	
1232	001010	200300	ØØØØ032		MOVE T, TAC1	JYES, ONLY INPUT SMALLER AMOUNT(R,RUN,GET,	KJOB)
1233	001011	242300	000034		LSH T, +D18++D10	I*BUILD IOWD FOR SQIN	
1234	001012	210100	007006		MOVN TAC1,T] *	
1235	001013		000777'		HRR TAC1, JBTADR(J)	; 4	
1236	001014		000753			#GET DEVICE ADDRESS	
1237	001015	254000	0010651		JRST SQIN	J*START INPUT	

,

.

SCHEDB					ING SYSTEM(10/50)(BU B 69 V406	RROUGHS DISK) MACRO,V36 19:10 4-JUN-69 PAGE 4
1238				ROUTIN	E TO CHANGE DISK SWA	PPING SPACE ALLOCATION(OR SET TO 0)
1239						T VIRTUAL TALLY FOR SYSTEM IS ALSO CHANGED
1240					FROM COREØ	THE SEE THEET THE SECONDARDED
1241					MOVE ITEM, JOB OR HI	GH SEG NO.
1242				1		TO BE NEW ASSIGNMENT
1243				1	PUSHJ PDP, CHGSWP	
1244				1	ALWAYS RETURN	
1245				CALLED		HYSICAL CORE ROUTINE COREØ
1246					oner pro spinskers	and toke oblice additine oddies
1247				INTERN	CHCSWP	
1248					JETSTS, IMGIN, IMGOUT,	INTSWP, VIRTAL, IPOP.I
1249						
1250	001016	135100	0010061	CHGSWP:	LDB TAC1,IMGIN	ISIZE WHEN SEG NEXT SWAPPED IN
1251	001017		0010351		JUMPE TAC, CHG1	JIS ZERO BEING ASKED FOR?
1252	001020		777766		LSH TAC,-12	INO, CONVERT TO 1K BLOCKS
1253	001021				ADDI TAC.1	JBUT DO NOT ATTEMPT TO RETURN DISK SPACE
1254						I SINCE IT MIGHT BE FRAGMENTED (SWAPPER WILL
1255						RETURN ALL OF DISK SPACE ON NEXT SWAPIN)
1256						HAPPENS ONLY ON R, RUN, GET, KJOB
1257	001022	137040	0010161		DPB TAC, IMGIN	STORE NEW SIZE WHEN NEXT SWAPPED IN
1258	001023	261140	000004		PUSH PDP.ITEM	ISAVE AN AC
1259	001024		001074		LOB ITEM, IMGOUT	IGET OLD DISK SIZE OF THIS USER (USES ITEM)
1260	001025	315130	000004		CAMGE TAC1.ITEM	IS OLD IN-CORE SIZE BIGGER?
1261	001026	200120	000004		MOVE TAC1, ITEM	ING, USE DISK SIZE AS USER'S OLD VIRTUAL CORE
1262	001027	315040	000004		CAMGE TAC, ITEM	IS NEW IN-CORE SIZE BIGGER?
1263	001030	200040	007004		MOVE TAC, ITEM	IND, USE DISK SIZE AS USER NEW
1264						I VIRTUAL CORE
1265	001031	274170	000001		SUB TAC1.TAC	IDECREASE OF USER VIRT. CORE=OLD-NEW
1266		272100	000601		ADDM TAC1, VIRTAL	
1267						J CORE
1268	001033	254000	0007571		JRST IPOPJ	JRESTORE ITEM AND RETURN

			-		ING SYSTEM(10/50)(BURROU B 69 V406						
1269						SPACE FOR A LOW OR HIGH SEG					
1270						N ONLY AND HAS NO EFFECT ON A SEGMENTS					
1271					L CORE ASSIGNMENT						
1272				CALLI	ALL: MOVE ITEM, JOP NO. OR HIGH SEG NO.						
1273				1	PUSHJ PDP,ZERSWP						
1274					LLED FROM SEGCON IN MANY PLACES(5)						
1275				JAND FI	NINA HERE IN SWAP						
1276											
1277					INTERN ZERSWP						
1278	_										
1279	701034	634040				IREQUEST O SPACE ON DISK AND ALWAYS SKIP					
1280 1281	ØØ1Ø35	272100	001032'	CHG1:	ADDM TAC1,VIRTAL	JINCREASE SIZE OF VIRTUAL CORE AVAILABLE IN SY J AND THEN RETURN ALL OF DISK SPACE(CHGSWP)	SILM				
1282	001036	205100	006000		MOVSI TAC1, SWP!SHF	ICLEAR SWAPPED OUT BIT IN JOB OR SEG					
1283	001037	412124	000661'		ANDCAM TAC1, JBTSTS(ITEM)ISTATUS WORD(SHE SET IF IO WAS TO BE STOPPED					
1284						; FOR SWAP OR CORE SHUFFLE					
1285											
1286	001040	261140	000005		PUSH PDP,DAT	JSAVE TTY OUTPUT BYTE POINTER(COMMAND DECODER)					
1287	001041	135240	001024'		LDB DAT, IMGOUT	I*SIZE ON DISK(1K BLOCKS)					
1288	001042	322240	001045'		JUMPE DAT, CHG3	IDID SEG HAVE ANY DISK SPACE?	1				
1289	001043	554044	001014'		HLRZ TAC, JBTSWP(ITEM)	J*YES, LOGICAL DISK BLOCK+FRGSEG BIT					
1290	001044	260140	001224'		PUSHJ PDP,FXSAT	I*FREE THE DISK BLOCKS NO LONGER NEEDED					
1291	ØØ1045	26214Ø	000005	CHG3:	POP PDP,DAT	RESTORE TTY OUTPUT BYTE POINTER					
1292	001046	201040	000000		MOVEI TAC,Ø	10 IS NEW DISK ASSIGNMENT	-				
1293	ØØ1047	137040			DPB TAC, IMGOUT	SET DISK ASSIGNMENT TO Ø					
1294 1295	ØØ105Ø	137040	ØØ1Ø22'		DPB TAC, IMGIN	SET NEW CORE IMAGE BLOCK SIZE WHEN NEXT SWAPP HERE FROM CHGSWP IF NOT ASKING FOR Ø	EDI				
1296	001051	263140	000000		POPJ PDP,	JRETURN					

SCHEDB	- SCHEDULING ALGORITHM FOR SWAPPIN SWAP R. KRASIN/AF TS4,34 Ø3 FEB		OUGHS DISK)	MACRO,V36 19:10	4-JUN-69 PAGE 47
1297 1298	E	XTERN PROTØ,PROT	PROT AND PROTO	OCCUR IN COMMON	
1299	IFE F	TRC10. <			
1370	XP ICPRT1		PROTECT TIME IN	SLOCK TICS-	
1371	XP ICPROT		I((JOBSIZE/1K)+		
1372	••••			4 PRODUCE PROTECT	TIMES ROUGHLY
1323				DISK SWAP(1-WAY)	
1394	>				1.1.201
1305	IFN F	TRC10, <			
1376	;SIMILAR	IN-CORE PROTECT TIME	PARAMETERS FOR FAS	STER RD-10 DISK	
1307	XP ICPRT1	,3+1+3	JZERO CORE PLUS		
1308	XP ICPROT		MULTIPLY BY K-1		
1309	>				

SCHEDB	- SCHEDU Swap R,	ULING ALC , krasin/	GORITHM F Mar ts4.3	DR SWAPP 4 03 FE	ING SYSTEM(10/50)(BURROU B 69 V406	IGHS DISK) MACRO,V36 19:10 4-JUN-69 F	AGE 48
1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 1320				,THEN R ,JOBS P ;Called ;ASSUME ;This I	ACK IN TO DESIRED AMOUNT OSITION IN OS NOT AFFECT ONLY FROM CORE COMMAMD S CALL FOR CURRENT JON I	ÉD. F EXPANDING HIGH SEG,IE ASSUME AT UUD LEVEL CORE COMMAND WHICH GAN EXPAND HIGH SEG	
1321 1322 1323 1324 1325	001053	242040 271040 137040	000001		LSH TAC,-12 Addi tac,1 OPB tac,imgin	ICONVERT HIGHEST DESIRED ADDRESS ITO 1K BLOCKS ISTORE, SO SWAPPER WILL KNOW HOW MUCH CORE I TO REQUEST WHEN NEXT SWAPPED IN	
1326 1327 1328				BFCAUS	E TO FLAG JOB TO BE STOP E IT HAS JUST BEEN CONNE Ith Way IN OR OUT, THE	PED AND SWAPPED OUT CTED TO A HIGH SHARABLE SEG WHICH IS ON DISK SIZE OF THE HIGH SEG IS UNCHANGED	1
1329 1330 1331 1332 1333 1334				JEXPAND	B MUST BE STOPPED UNTIL En High Seg(MUST be call Move Item,High seg numb) Pushu PDP,XPandh Intern XPANDH	HIGH SEG SWAPPED IN JUST AS IF JOB HAS ED FROM UUD LEVEL FOR CURRENT JOB IF HIGH SE ER	Ģ)
1335 1336 1337	001055			XPANDH:	EXTERN IPOPJ		
1337 1338 1339 1340 1341 1342	001055	261140 303200 200200	000004 000150' 000665'	TEN CTO	PUSH PDP,ITEM CAILE ITEM,JOBMAX	ISAVE JOB NUMBER JIS THIS A LOW OR HIGH SEG? JHIGH,SO GET JOB NO.(MUST BE CURRENT JOB)	
1343 1344 1345 1346 1347	001062	205100 616124 350020 436104	000001 001037 000323 001061		TDNN TAC1, JBTSTS(ITEM) AOS XJOB IORM TAC1, JBTSTS(ITEM)	ISET THIS JOR EXPANDING BIT SO IT WILL NOT IS IT ALREADY SET FOR THIS JOB?(UNLIKELY) ING, INCREMENT COUNT ONLY ONCE FOR EACH JOB IAND SET JOB EXPANDING BIT	
1348 1349 1350					POPJ PDP,	JRETURN	
	001064	25403Ø	201233'			FRESTORE JOB OR HIGH SEG NUMBER (ITEM) AND H	RETURN

SCHEDB	- SCHEDU	LING ALG	ORITHM F	OR SWAPP	ING SYSTEM (1	LØ/50)(BURROUG	HS DISK)	MACR0,V36 19:12	4-JUN-69 PAGE 49
1353	SWAP R.	KRASINZ	AF 154.3		8 69 V406 SWPSER R.KR	ASIN/AF TS4.3	4 Ø3 FEB 69	V 4 Ø 6	
1354				0-0	•				
1355					L SQIN, SQOUT	,SQG0,SQG01			
1356					L FTSWAP				
1357				EXTERNA	L DEBUSY, DEF	ED, DEWRT, CPOP	J, JOBDAC, MJOB	CK,CHECK,JBTCHK	
1358								T SOIN FOR	
1359 1360						HE SWAPPING Q IT FOR OUTPUT	UEUE, ENTER A	I SUIN FUR	
1360				3			ROS. FIRST COR	E LOC.(IE IOWD+1)	
1362				1		SK BLOCK NO.			
1363				i	PUSHJ PDP,S				
1364				1	RETURN HERE	ALWAYS			
1365				1	CONTENTS OF	TAC, TAC1 LOS	т		
1366									
1367	001065	661040	400000	SQIN:	TLO TAC,400	0000	ISET READ IND		
1368	701066	202040	001342	SQOUTI	MOVEM TAC.S	ERA	ISTORE THE BLO		
1369			001340				ISTORE THE ION	OF DISK ERROR ON F	PACHENTED 10P
1370			001341' 000001		MOVEM TAC1. MOVNI TAC.1	ESUREU	IS THE DEVICE		RAGMENTER JUB
1371 1372			0000001		EXCH TAC, DF	BUSY	IS THE DEVICE	- 80311	
1373			000771'		JUMPN TAC,C		IYES IF JUMP		
1374	001075	3200-0	000771				JIEO II 00		
1375									
1376			000003	ERATRY=	3 ;NC	. OF TIMES TO	READ AND WRI	TE ON ERRORS	
1377									
1378							EQUEST, THIS I		
1379								L AS FROM ABOVE.	
1380				IIF A S	WAPPER REQUE	ST IS WAITING	(SOREQ WILL BI	L NON-ZERO)	
1381		004400	000007		MOVEL TAC1,	COATON			
1382 1383		201100 202100	000003 001343'		MOVEN TACI				
1384		205040	400000		MOVSI TAC.4				
1385		612040	0013421		TDNE TAC.SE		WRITE?		
1386			001107		JRST SOGO1		INO		
1387	001101	550040	001340'		HRRE TAC, SG	REQ			
	001102		000000		ADDI TAC, JO				
1389			000000		HRLI TAC,M.				
		260140			PUSHJ PDP,C				
			000772'		MOVM TAC				
1392	001106	202171	0000000		MUVEM TAU	C1,JBTCHK(TAČ)			

SCHEDB			GORITHM FO				20UGHS DISK) MACRO,V36 19:17 4-JUN-69 PAGE 52
1393	001107	402000	001344'	S0G01:	SETZM	SQLEN	ZERO AMOUNT TRANSFERRED SO FAR
1394	001110	200100	001340'		MOVE TA	C1.SOREO	*PUT IOWD INTO TAC1
1395	001111		200000			AC,200000	#SET "SWAPPER I/O GOING" FLAG ON
1396		437040	001342'		ORB TAC		
1397			400000		TRZN	TAC, FRGSEG	1*FRAGMENTED?
1398			001142'		JRST	SQGO2	IND, READ IN ENTIRE (OR PART) OF SEG
1399							
1400					EXTERN	CL COR1	
1471					C.A. C.	010011	
1402	001115	261140	000005	FRAGIO:	PUSH	POP, DAT	
1403	001116	210240	001344		MOVN	DAT, SQLEN	JAMOUNT PREVIOUSLY TRANSFERRED
1404	001117	274100	000005		SUB	TAC1,DAT	INCREASE CORE ADDRESS BY WORDCOUNT PREVIOUS
1405			000000	FRGIO1:		DAT, (TAC)	
							ISWAPPING ADDRESS FOR THIS DISK CHUNK
1407	001122	325240	0011271		UMPGE	DAT, FRGIO2	POINTER TO NEW CORE LIST IF NEG.
1408	001123	201240	077777		MOVET	AT,77777	ICLEAR OUT ADR(15 BITS)
1409	001124	412240	0013421		ANDCAM	TAC,(TAC) DAT,FRGIO2 DAT,7777 DAT,SERA SERA	J J
1410	001125	436040	001342'		ORM TAC	SFRA	INSERT NEW ADDRESS
1411		254000	001120		JRST FR		JINDERN VER ROBREOG
					0		
1412							
1413	201127	242240	000012	FRGI02:	LSH	DAT, 12	ICONVERT FROM K TO WORDS
1414	ØØ113Ø	27224Ø	001344'		ADDM DA	T, SQLEN	ADD TO PREVIOUSLY TRANSFERRED AMOUNT
1415	001131	213000	000005		MOVNS D	AT	I-N WORDS
1416	001132	506240	000002		HRLM	DAT, TAC1	FIOND IN TACI
1417	001133	524240	000005		HRLO DA	T, DAT	I-NO, OF WRDS FOR THIS DISK TRANSFER TO UH
1418	001134	317240	001340'		CAMG DA	T, SOREQ	COMPARE WITH - NO. WORDS FOR REST OF SEC
1419	001135	500100	001340'		HLL TAC	1,SOREQ	ISWAPPER ONLY WANTS TO READ A PORTION OF SEG
1420							I NOT ALL OF IT(R, RUN, GET, KJOB COMMAND)
1421	001136	213000	000005		MOVNS D	AT	I+NO. OF WORDS FOR THIS NEXT TRANSFER
1422	001137	271240	777777		ADDI DA	T,77777	
1423	001140	272240	001340		ADDM DA	T. SOREQ	JUPDATE LH OF IOWD FOR ENTIRE SEG, SO IT HAS
1424						• =	I -NO. OF WORDS LEFT AFTER THIS TRANSFER IS DONE
1425	001141	262140	000005		POP	PDP, DAT	
4 4 9 4		101 010	777747				

,

 1425
 001141
 262140
 000005
 PDF
 < #*CLEAR POSSIBLE TRASH IN LH. #*RE-POSITION DISK LOGICAL BLOCK NUMBER. #*TEST AND CLEAR READ/WRITE BIT. ;*YES ;*NO, WRITE,

SCHEDB					ING SYSTEM(10/50)(BURROU(Er 69 v406	GHS DISK) MACRO,V36 19:10 4-JUN-69 PAGE 51
1431				ISERVICE	A SWAPPING INTERRUPT	
1432					DINT4R, CKSMCT	
1433				INTERNAL		
1434				1.4.6.6.6.6.6.6		
1435	001147	602000	700000	SWPINIT	TRNE IOS, IODTER: IODERR:	TOTMPM
1436	001150	254020	ØØ1176'		JRST SWPERR	IERRORS
1437	001151	622040	400000		TRZE TAC, FRGSEG I*FRAGME	
1438	001152	331100	001340'		SKIPL TAC1, SOREQ	INYES, MORE LOWD TO GO?
1439	001153	254000	001157'		JRST DINT8B	IND, ALL DONE SWAP IN OR OUT
1440	001154	350000	001342'		AOS SERA	IYES, FRAGMENTED AND MORE TO GO
1441	001155	332001	000001		SKIPE 1 (TAC)	IS THIS THE END OF SWAP COMMAND LIST?
1442	001156	344040	001115'		AOJA TAC, FRAGIO	ING, BO DO NEXT PIECE OF FRAGMENTED SEG
1443						
1444	001157	607040	400000	DINT88:	TLNN TAC,400000	J*INPUT?
1445	001160	254000	001170'		JRST DINT8A	3 # NO
1446	001161	550040	001340'		HRRZ TAC, SQREQ	
1447	001162	271040	001102'		ADDI TAC,JOBDAC	
1448	001163	505040	0011031		HRLI TAC,MJOBCK	
1449	001164	260140	001104'		PUSHJ PDP,CHECK	
1450	ØØ1165	214040	001105'		MOVM TAC, FINISH	
1451	PØ1166	312101	001106'		CAME TAC1, JBTCHK(TAC)	
1452	001167	254020	001173'		JRST SWPER1	
1453	001170	552020			HRRZM IOS, SERA	
1454		402000	001340'		SETZM SOREO	
1455	001172	254090	ଷଷଷଷଷ		JRST DINT4B	
1456						
1457	001173	205040		SWPER1:	MOVSI TAC, 100	
1458	001174	272040	000000		ADDM TAC, CKSMCT	
1459	001175	660000	100000		TRO IOS, IODTER	I*RESET SERA IN CASE OF FRAGMENTED JOB
1460	001176	214040		SWPERKI		INCASE OF FRAGMENTED JUB INSWAP LOC (DISK ADR OR TABLE ADR)
1461	001177	554041	001043			INFRANCE SERA
1462	001200	542040	001342		HRRM TAC, SERA Move tac,esqreq	RESTORE ESGREG IN CASE OF FRAGMENTED JOR
1463	001201	200040	001341' 001340'		MOVE TAC, ESGREG	PRESIDIE ESQUED IN CASE OF LUXONENTED JUR
1464 1465	ØØ1202 ØØ1203	202040 373000	001340 001343			J*TRIED ENOUGH?
1465	001203	254000	ØØ1343'			INNO, TRY AGAIN
1460	001204 001205		001170			J*YES, TOUGH.
140/	NO1205	224000	ENTING.		Avai Mturov	iairai Ionout

.

20HEUB	- SCHEDULING ALGORITHM FOR SWAPP SWPSER R.KRASIN/AF 184,34 //3 F	1%G SYSTEM(10/53 ER 69 - 4426)(BURROUGHS DISK)	MACR0.V36 19:10	4-JUN-69 PAGE 52
1468	IFF	FTRC10, K			
1469			LO POP-6 (DATA PRODUCTS		
1470		20110 00 012 012 0		UISK FILE	
1471.	;FIND A	SERIES OF BLOCK	S ON THE DISK TO SWAP OF	NTO, CALLED	1
1472	JAT CLO	CK LEVEL.		TOP URLEED	
1473	;CALL:	MOVEL DAT. NO. O	F 1K PLOCKS DESIRED		
1474	;	PUSHJ POP, GXSAT			
1475 1476	;	ERROR EXIT	(DISK IS FULL)		
1470	1	NORMAL EXIT	IC(TAC) = BLOCK NO.		
1478					
1479	IUNIEN	IS OF ACS TAC, TA	C1,DAT WILL BE LOST.		
1480	INTERNAL	CVEAT			
1481		_ GETRIT, [POPJ1			
1482		- OFFICIAL OF DT			
1483	GXSAT:	MOVE AC1, XSAT1	ISAVE ACT. SET	IT TO TABLE LOC.	
1494		MOVE AC2.XSAT2	I I I I I I I I I I I I I I I I I I I	IT TO TARLE LTC.	
1485		LSH DAT, CONVHD		NORD DISK PLOCKS	
1486		PUSH POP, ITEM	ISAVE C(ITEM)		
1487		HOVE ITEM, DAT	CETRIT EVERATE	PARAMETER IN ITEM	. i
1488		PUSHU PDP GETBI			
1489		JRST IPUPU		ITEM AND ERROR RET	1106.
1490		MOVEL TACI-1(TAC	(1)	THE FAR SERVER REI	UKN I
1491		CAIL TAC, BLOCKS		172	2
1492		AODI TAC, DIFF	IYFS		
1493		LSH TAC, -CONVMD	ICARRY DISK ADD	RESS SHIFTED TO FI	T IN JATSWP.
1494 1495		JRST IPOP⊎1	ISKIP RETURN AN	D RESTORE JOB NUMB	ER(ITEN)
1495			_		
1497	FREE UP	A SERIES OF BLO	OCKS ON THE SWAPPING DEV	ICE, CALLED	
1498		K LEVEL.			
1499			1K CORE BLOCKS TO FREE		
1500		PUSHJ PDP, FXSAT	10. OF FIRST DISK BLOCK	TO FREE	
15/1		ALWAYS RETURN HE	RF		
1572		all all all all all all all			
15"3	; CONTENT	S OF ACS TAC, TAC	1 WILL BE LOST.		
1574					
1525	INTERNAL				
1526	FXTERNAL	CLABIT, 180PJ			
1528					
1578		LSH TAC.CONVMD		DISK ADDRESS.	
1511		CAIL TAC, HISWAP	101 DISK 172		
1511		SUBI TAC, DIFF LSH DAT, CONVMD	IYES		
15:2		MOVE AC1,XSAT1		WORD DISK PLOCKS	
1		MOVE AM2, XSAT2	IAC2 TOO	H POINTER TO SAT B	LUCK
1514		PUSH POR, HIEM	SAVE C(ITEM)		
1515		MOVE ITEN CAT		PARAMETER IN DAT	
1516		PUSH, FOP, CLRBIT		CONTRACTOR OF CAL	
1517		JAST IPOPJ	RETURN, AND RE	STORE ITEM	
1518					
1519	; INITIAL	12F SAAPPER DISK	STORAGE TABLE		
1520					

1501INTERNAL SMPINI1502SWPINI: HOVE TAC,YSAT21503NOVE TAC,YSAT31504NOVE TAC,YSAT31505SET2H XSAT41506NOVE TAC,YSAT31507SET2H XSAT41508NOVE TAC,YSAT31509NOVE TAC,YSAT51511NOVE TAC,YSAT61522NOVE TAC,YSAT61523NOVE TAC,YSAT61524NOVE TAC,YSAT61525FN FTCHECK+FTMON?1526FN FTCHECK+FTMON?1527FN FTCHECK+FTMON?1528IFF FTCHECK+FTMON?1529IFF FTCHECK+FTMON?1530IFF FTCHECK+FTMON?1540IFF FTCHECK+FTMON?1541IATA AN STPAACE AFEA FOB SWAPPING, ON THE 278 DISK, DISKS1543IATA AN STPAACE AFEA FOB SWAPPING, ON THE 278 DISK, DISKS1544IFF FTCHECK+FTMONP.1544IATA AN STPAACE AFEA FOB SWAPPING, ON THE 278 DISK, DISKS1545IATA AN STPAACE AFEA FOB SWAPPING, ON THE 278 DISK, DISKS1546INTERNAL1547SOREG1 71548IATA AN STPAACE AFEA FOB SWAPPING, ACO T INTE NOOT D READ/WRITE1549SERGE1 71540ICAN+SUCC NUMBER BEFORE READING,1551SERAI 71553SERAI 71554SERAI 71555SULENT 81556SATI1 EXP XSAT3+11557YEA 4080001558SULENT 81559XAT11 KEM 4080001550YEA 4000001551SERACTI 81556YEAL	SCHEDB - SCHEDULING ALGORI SWPSER R.KRASIN/AF	THM FOR SWAPPING SYSTEM(10/50)(BUR 5 TS4,34 - 03 FER 69 - V406	ROUGHS DISK) MACRO,V36 19:10 4-JUN-69 PAGE 52-1
1533 SMPIAIL HOVE TAC,XSAT2 1524 HOVET TAC,XSAT3 1525 HOVET TAC,XSAT3 1526 HOVET TAC,XSAT4 1527 SET2M XSAT4 1528 HOVET TAC,XSAT4 1529 HOVET TAC,XSAT4 1530 HOVET TAC,XSAT4 1531 HOVET TAC,XSAT5 1532 HOVET TAC,XSAT5 1533 HOVET TAC,XSAT6 1534 POPU PDP, 1535 HOVET TAC,XSAT6 1536 FTHERTEKTHONPA 1537 EXTERNAL SORED,SERA,SERACT,XSAT1,XSAT4,XSAT5,XSAT6,XSAT7 1538 EXTERNAL SORED,SERA,SERACT,SCONTAND,BLOCKS,XSAT31,XSAT4,XSAT6,XSAT7 1540 IFE FTCHECK+FTMONPA 1541 IDATA AND STORAGE AREA FOR SWAPPING, ON THE 278 DISK, DISKS 1542 IDATA AND STORAGE AREA FOR SWAPPING, ON THE 278 DISK, DISKS 1544 IFE FTCHECK+FTMONPA 1545 ICATA AND STORAGE AREA FOR SWAPPING, ON THE 278 DISK, DISKS 1546 INTERNAL SORED,SERA,SERACT 1547 IDATA AND STORAGE AREA FOR SWAPPING, ON THE 278 DISK, DISKS 1548 IFE FTCHECK+FTMONPA 1544 IDATA AND ST		INTERNAL SWPINI	
1524 HUVÉ Y TAC, XSAT31 1525 HUVÉ Y TAC, XSAT3 1526 MUVE Y TAC, XSAT3 1527 STTAY XSAT4 1528 MUVE Y TAC, XSAT4 1529 HOVE Y TAC, XSAT4 1529 HOVE Y TAC, XSAT4 1529 HOVE Y TAC, XSAT4 1520 HOVE Y TAC, XSAT4 1531 HOVE Y TAC, XSAT4 1532 HOVE Y TAC, XSAT4 1533 HOVE Y TAC, XSAT4 1533 HOVE Y TAC, XSAT4 1533 HOVE Y TAC, XSAT4 1534 HOVE Y TAC, XSAT4 1535 HOVE Y TAC, XSAT4 1536 HOVE Y TAC, XSAT6 1537 EXTERNAL XSAT6, SERA, SERACT, XSAT3, XSAT4, XSAT5, XSAT6, XSAT7 1538 EXTERNAL XSAT6, SERA, SERACT 1540 IFE FTCHECK+FTMONP. 1541 IOATA ANN STORAGE AFEA FOR SWAPPING, ON THE 278 DISK, DISKS 1542 IOATA ANN STORAGE AFEA FOR SWAPPING, ON THE 278 DISK, DISKS 1543 : P & 17 AFE USEA FOR SWAPPING, ON THE 278 DISK, DISKS 1544 : DATAE, MORE OF SERA, SERACT 1545 INTERNAL SOREO! SERA, SERACT 15		CHRINIA MONE THE VEATO	
1550 MÜYSİ TAÇ,YABIA 1550 MÜYY TAÇ,YSAT3 1577 SETZM XSAT4 1578 MÜYE TAÇ,YSAT3 1579 RLT TAÇ,YSAT3 1570 MÜYE TAÇ,YSAT3 1571 MÜYE TAÇ,YSAT3 1572 MÜYE TAÇ,YSAT3 1573 MÜYE TAÇ,YSAT3 1530 MÜYE TAÇ,YSAT3 1531 MÜYE TAÇ,YSAT3 1532 MÜYE TAÇ,YSAT3 1533 MÜYE TAÇ,YSAT3 1534 MÜYE TAÇ,YSAT3 1535 IPN FICHECK+FIMONP, 1536 IPN FICHECK+FIMONP, 1537 EXTERMAL SORGO,SERA,SERACT,SKAT3,XSAT4,XSAT5,XSAT4,XSAT5,XSAT4 1548 IFE FICHECK+FIMONP, 1549 IDATA AND STORAGE AREA FOR SWAPPING, ON THE 270 DISK, DISKS 1540 IFE FICHECK+FIMONP, 1541 IDATA AND STORAGE AREA FOR SWAPPING, EACH DISK CONTALNING 1542 IDATA AND STORAGE AREA FOR SWAPPING, COF SIZE OF READ/WRITE 1544 IDATA AND STORAGE AREA FOR SWAPPING, COF SIZE OF READ/WRITE 1545 SORGOI 7 IC(HI)+REGO. OF SIZE OF READ/WRITE 1546 INTERNAL SORGOI 7			
1526 MOVEY TAC, XSAT3 1527 STT2W XSAT4 1528 MOVE TAC, XSAT4P 1529 RLT TAC, XSAT4 1529 MOVE TAC, XSAT4 1530 MOVE TAC, XSAT5 1533 MOVE TAC, XSAT5 1534 MOVE TAC, XSAT5 1535 MOVE TAC, XSAT5 1533 MOVE TAC, XSAT6 1534 MOVE TAC, XSAT6 1535 FXTERMAL SORE0, SERA, SERACT, XSAT2, XSAT3, XSAT4, XSAT5, XSAT6, XSAT7 1535 EXTERMAL SORE0, SERA, SERACT, MOVP, ALL 1536 IFF FTCHECK+FTMONP, A 1540 IFF FTCHECK+FTMONP, A 1541 IOATA AND STORAGE AREA FOR SWAPPING, ON THE 270 DISK, DISKS 1542 IOATA AND STORAGE AREA FOR SWAPPING, CACH DISK CONTAINING 1543 IA TA ARE USED FOR SWAPPING, CACH DISK CONTAINING 1544 IA TA ARE USED FOR SWAPPING, CACH DISK CONTAINING 1545 INTERNAL SOREGI 7 1546 INTERNAL SOREGI 7 1547 SOREGI 7 ICCHA)=REG OF SIZE OF READ/WRITE 1548 INTERNAL SOREGI 7 1549 INTERNAL SOREGI 7			
527 SETZM XSAT4 528 MOVE TAC, XSAT4P 529 RLT TAC, XSAT4 530 MOVEM TAC, XSAT5 531 MOVEM TAC, XSAT6 532 MOVEM TAC, XSAT6 533 MOVEM TAC, XSAT6 534 MOVEM TAC, XSAT6 535 MOVEM TAC, XSAT6 536 IFN FICHECK+FTMONP, 537 EXTERNAL SOREQ.SERA, SERACT, XSAT1, XSAT2, XSAT4, XSAT5, XSAT6, XSAT7 538 EXTERNAL SOREQ.SERA, SERACT, XSAT1, XSAT2, XSAT4, XSAT5, XSAT6, XSAT7 539 S 540 IFF FICHECK+FTMONP, 541 JDATA AND STORAGE AREA FOR SWAPPING, ON THE 278 DISK, DISKS 542 JDATA AND STORAGE AREA FOR SWAPPING, EACH DISK CONTAINING 544 IFF FICHECK+FTMONP, 545 IFA P (CAL) RECORDS. 546 IFE FICHECK+FTMONP, 547 SOREQI ? 548 IFF TCHECK+FTMONP, 544 IFF TCHECK+FTMONP, 545 IFF TCHECK+FTMONP, 544 IFF TCHECK+FTMONP, 545 IFF TACKASCA <t< td=""><td></td><td></td><td></td></t<>			
1520 NOVE TAC,XSAT4P 1520 RLT TAC,XSAT61 1530 MOVE TAC,XSAT51 1531 NOVE TAC,XSAT51 1532 MOVE TAC,XSAT51 1533 NOVE TAC,XSAT61 1534 POP,PDP, 1535 IFN FICHECK+FIMOUP, 1536 EXTERNAL SORED,SERA,SERACT,XSAT1,XSAT2,XSAT3,XSAT4,XSAT5,XSAT6,XSAT51 1536 EXTERNAL SORED,SERA,SERACT 1537 IFE FICHECK+FIMOUP, 1540 IFE FICHECK+FIMOUP, 1541 IFE FICHECK+FIMOUP, 1542 IDATA AND SIGNAGE AREA FOR SWAPPING, ON THE 270 DISK, DISKS 1543 : # A 17 ARE USED FOR SWAPPING, EACH DISK CONTAINING 1544 : GEN,SERA,SERACT 1545 : UTENNAL 1546 INTERNAL 1547 SORED: 7 1548 : C(LH)=NEG, OF SHEE0, IN CASE OF 1549 ESORFUI 7 1540 INTERNAL 1541 : DATA AND STORAGE AREA FOR SWAPPING, ALLOCAND TO READ/WRITE 1542 : DATA AND STORAGE AREA FOR SWAPPING, ON THE 270 DISK, DISKS 1544 : FROMENT, CONTARTARCHUTE 1545 : C(
1529RLT TAC.VSAT11530NOVET TAC.VSAT51531NOVET TAC.VSAT51532NOVET TAC.VSAT61533NOVET TAC.VSAT61534POPP POP,1535IFN FTCHECK+FTMONP.1536EXTERNAL SORED.SERA.SERACT.XSAT1.VSAT2.XSAT3.XSAT4.VSAT5.XSAT6.XSAT71537EXTERNAL SORED.SERA.SERACT.SCAT1.NSAT2.XSAT3.XSAT4P.XSAT611538SAT601540IFF FTCHECK+FTMONP.1540IFF FTCHECK+FTMONP.1541IDATA ANN STORAGE AREA FOR SWAPPING, EACH DISK CONTAINING1544.1545INTERNAL1546INTERNAL1547SORED.71548.1549ESOREUI 71549.1550SERA.T1551SERA.T1553.1554.1555SQLEN! 01556XSAT3! XWA1557XSAT3! NWA1558SQLEN! 01559XSAT3! NWA1550XSAT3! NWA1551SERACT! 01552.1553SQLEN! 01554SERACT! 21555SQLEN! 01556XSAT3! NWA POPIE.XSAT41557XSAT3! NWA POPIE.XSAT41560.1571XSAT2! NWD POPIE.ALDER DY GETBIT.CLRBIT1560.1571.1571.1572.1571.1572.1571.1571.1572.			
1530 HÖVE TAC,XSAT7 1531 HÖVE TAC,XSAT5 1532 HÖVE TAC,XSAT5 1532 HÖVE TAC,XSAT5 1533 HÖVE TAC,XSAT5 1534 POP, POP, 1535 IFN FICHECX-FIMONP, 1536 IFN FICHECX-FIMONP, 1537 EXTERNAL SORD,SERA,SERA,CT,XSAT1,XSAT3,XSAT4,XSAT5,XSAT6,XSAT7 1538 EXTERNAL SORD,SERA,SERACT 1540 IEF FICHECK+FIMONP, 1541 IDATA ANN STORAGE AREA FOR SWAPPING, ON THE 270 DISK, DISKS 1542 IDATA ANN STORAGE AREA FOR SWAPPING, EACH DISK CONTAINING 1544 SOREOI - 1545 INTERNAL 1546 INTERNAL 1547 SOREOI - 1548 : SOREOI - 1549 ESOREUI 7 1540 INTERNAL 1541 SOREOI - 1542 ISON IF A READ 1544 SOREOI - 1545 INTERNAL 1546 INTERNAL 1547 SOREOI - 1548 INTERNAL			
1531 HOVEY TAC,YSAT5 1532 HOVEY TAC,YSAT5 1533 HOVEY TAC,YSAT6 1534 HOVEY TAC,YSAT6 1535 IFN FICHECK+FIMONP, 1536 EXTERNAL SOREO,SERA,SERACT,XSAT1,XSAT4,XSAT5,XSAT4,XSAT6,XSAT7 1537 EXTERNAL SOREO,SERA,SERACT,XSAT1,XSAT4,XSAT5,XSAT4,XSAT6,XSAT7 1538 EXTERNAL SOREO,SERA,SERACT,XSAT1,XSAT4,XSAT6,XSAT6,XSAT7 1540 IFE FICHECK+FIMONP. 1541 IDATA AND STORAGE AREA FOR SWAPPING, ON THE 270 DISK, DISKS 1544 ; p & 1 ARE USED FOR SWAPPING, EACH DISK CONTAINING 1545 INTERNAL 1546 INTERNAL 1547 SOREO1 7 1548 INTERNAL 1549 ESOREU1 7 1549 ESOREU1 7 1550 SERACTI R 1551 SERACTI R 1555 SULEN: 8 1556 SULEN: 8 1557 XSAT1! EXP XSAT3-1 1558 SULEN: 8 1559 XSAT1! EXP XSAT3-1 1560 YSAT1! AND 480000K 1551 SERACTI R 1600HTER USED RY GETALXANSEERED SO FAR - F			
1532 MOVE TAC, XSAT8 1533 MOVEN TAC, YSAT6 1534 POPU PDP, 1535 IFN FTCHECK+FTMONP, 1536 IFN FTCHECK+FTMONP, 1537 EXTERMAL XSAT8, SERACT, XSAT1, XSAT2, XSAT3, XSAT4, XSAT6, XSAT7 1538 EXTERMAL XSAT8, SERACT, XSAT1, XSAT2, XSAT3, XSAT4, XSAT6, XSAT7 1539 > 1540 IFE FTCHECK+FTMONP, 1541 :DATA AND STORAGE AREA FOR SWAPPING, EACH DISK CONTAINING 1542 :DATA AND STORAGE AREA FOR SWAPPING, EACH DISK CONTAINING 1544 :SAP@ (OCTAL) RECORDS. 1545 INTERNAL SOREOI 7 1546 INTERNAL SOREOI 7 1547 SOREOI 7 IC(LH)=NEG, OF SIZE OF READ/WRITE 1548 :SOREOI 7 IC(LH)=NEG, OF SIZE OF READ/WRITE 1549 ESOREOI 7 IC(LH)=NEG, OF SIZE OF READ/WRITE 1541 :SOREOI 7 IC(LH)=NEG, OF SIZE OF READ/WRITE 1542 :SOREOI 7 IC(LH)=NEG, OF SIZE OF READ/WRITE 1543 :SOREOI 7 IC(LH)=NEG, OF SIZE OF READ/WRITE 1544 :SOREOI 7 IC(LH)=NEG, OF SIZE OF READ/WRITE 1545 :SERA: 7<			
1533HOVEW TAC,XSAT6 PDFy PDF,1534PDFy PDF,1535IFN FICHECK+FTMOVP,1536EXTERMAL SOREO,SERA,SERACT,XSAT1,XSAT2,XSAT3,XSAT4,XSAT5,XSAT6,XSAT71537EXTERMAL SAT8,SWP517,HISMAP,DIFF,CONVMO,BLOCKS,XSAT3,XSAT4,XSAT6,XSAT71538EXTERMAL XSAT8,SWP517,HISMAP,DIFF,CONVMO,BLOCKS,XSAT3,XSAT4,XSAT6,XSAT61539>1540IFE FICHECK+FTMONP,1541IDATA AND STORAGE AREA FOR SWAPPING, EACH DISK CONTAINING1542IDATA AND STORAGE AREA FOR SWAPPING, EACH DISK CONTAINING1543: 9 & 17 ARE USED FOR SWAPPING, EACH DISK CONTAINING1544: 0CAL) RECORDS.1544: 0CAL) RECORDS.1544: 0CAL) RECORDS.1545SOREOI 71546INTERMAL1547SOREOI 71548: CRAIPAIDOCK NUMBER OF READ/WRITE1549ESOREOI 71551SERA: 71551SERA: 71553SERA: 71554SERACTI 81555SULENI 81555SULENI 81556XSAT11 EXP XSAT3-11557XSAT11 EXP XSAT3-11558XSAT21 XND 4&0000041560INUMER OF RECORDS AT TOP oF TABLE1561: SIT TABLE FOR SWAPPING ALLOCATION1562: BIT TABLE FOR SWAPPING ALLOCATION1563: P BLOCKS/F0361564: BIT TABLE FOR SWAPPING ALLOCATION1565: SAT21 XND 4&000001566: ELEDLOCKS-H14*10361567: SAT31 XND 4&000001568: P BLOCKS/F036 </td <td></td> <td></td> <td></td>			
1334POPJ PDP,1335IFN FICHECK+FIMONP,1336EXTERNAL SORCO.SERA.SERACT,XSAT1,XSAT2,XSAT4,XSAT5,XSAT6,XSAT71338EXTERNAL SAT0,SWPSIF.HISWAP,DIFF.CONVMD,BLOCKS,XSAT31,XSAT4P,XSAT611340IFE FTCHECK+FIMONP.1540IFE FTCHECK+FIMONP.1541IDATA AND STORAGE ARCA FOR SWAPPING, ON THE 270 DISK, DISKS1542IDATA AND STORAGE ARCA FOR SWAPPING, EACH DISK CONTAINING1543, PA 17 ARE USED FOR SWAPPING, EACH DISK CONTAINING1544; SAP@ (OCTAL) RECORDS.1545INTERNAL1546INTERNAL1547SOREO,SERA,SERACT1548; SAP@ (OCTAL) RECORDS.1549ESOREUI 71549ESOREUI 71550SERAI. 71551SERAI. 71552; EROPI IN FRACHENTED JOG1553SERACTI Ø1554SERACTI Ø1555SULENI Ø1556XSAT11 EXP XSAT3-11557XSAT11 EXP XSAT3-11558XSAT3: YND AUPSIZ:XSAT41559XSAT3: YND AUPSIZ:XSAT41560XSAT3: YND AUPSIZ:XSAT41561SERACTI Ø1562INUMER OF RECORDS/DISK PLATE1559XSAT3: YND AUPSIZ:XSAT41560XND SENPSIZ:XSAT41561SERACTI Ø1562INUMER OF RECORDS/DISK PLATE1563YP BLOCKS.544001564INUMER OF ZEROUSS IN WI-HIST WORD1565YP BLOCKS.54401566INUMER OF ZEROUSS IN WI-HIST WORD1566ISERO WORDS AT BOTTOM OF TABLE			
13351FN FICHECK+FIMONP,1336IFN FICHECK+FIMONP,1337EXTERNAL SOREO,SERA.SERACT,XSAT1,XSAT4,XSAT4,XSAT5,XSAT4,XSAT5,XSAT4,XSAT71338EXTERNAL SOREO,SERA.SERACT,XSAT1,XSAT4,XSAT5,XSAT4,XSAT5,XSAT4,XSAT51339>1340IFE FICHECK+FIMONP,1541:DATA AND STORAGE AREA FOR SWAPPING, ON THE 270 DISK, DISKS1542:DATA AND STORAGE AREA FOR SWAPPING, EACH DISK CONTAINING1543: 5400 (OCTAL) RECORDS,1544:DATA AND STORAGE AREA FOR SWAPPING, EACH DISK CONTAINING1545: SOREO: 71546INTERNAL1547SOREO: 71548: OCPY OF SOREQ, IN CASE OF1549ESORFU: 71549: SERA: 71551SERA: 71553: CRMD'IN FRACKENTED JOB1554SERACT: 01555: SULEN: 01556XSAT1: EXP XSAT3-11557XSAT3: XND 400400;1558: SENACT: 11560: NUMBER OF OF RECORDS OF SAR - FRAC SEG1557XSAT3: XND 400400;1560: DATA SATA1561: FRACT: 01562: BIT TABLE FOR SWAPPING ALLOCATION1563: NUMBER OF RECORDS/DISK PLATE1564: BIT TABLE FOR SWAPPING ALLOCATION1564: HELOCKS'+0361565: WP BLOCKS-F0361566: FEROWORDS AT BOTTOM OF TABLE1567: YP BLOCKS-F0361568: FERO WORDS AT BOTTOM OF TABLE1569: YP XSAT31,XSAT3+11571: XSAT3: BLOCK W2+1 <td></td> <td></td> <td></td>			
1336IFM FICHECK+FIMONP,1337EXTERNAL SORCO.SERA.SERACT,XSAT1,XSAT2,XSAT3,XSAT4,XSAT5,XSAT6,XSAT71338EXTERNAL XSAT8,SWPSI7,HISWAP,DIFF,CONVMD,BLOCKS,XSAT31,XSAT4P,XSAT611340IFE FTCHECK+FIMONP,1341JOATA AND STORAGE ARCA FOR SWAPPING, ON THE 270 DISK, DISKS1342JOATA AND STORAGE ARCA FOR SWAPPING, EACH DISK CONTAINING1343; 0 A 17 ARC USED FOR SWAPPING, EACH DISK CONTAINING1344; 5400 (OCTAL) RECORDS.1345INTERNAL1346SOREO.SERA.SERACT1347SOREO1 71348: 0COPY OF SOREO, IN CASE OF1349ESOREU1 71340: 0COPY OF SOREO, IN CASE OF1354: 0COPY OF SOREO, IN CASE OF1354: 1EROR IN FRACMENTED JOB1355SERACTI ?1356SERACTI ?1357SERACTI ?1358SERACTI ?1359: 0CONTER FOR ERAOS1357XSAT11 EXP XSAT3-11358YAND -SWPSIZ.XSAT41359: SERACTI ?1350: SERACTI ?1351: SERACTI ?1355: SULEN! ?1356: SULEN! ?1357: SAT11 EXP XSAT3-11358: SULEN! ?1359: SERACTI ?1359: SULEN! ?1350: SERACTI ?1351: SERACTI ?1355: SULEN! ?1356: SULEN! ?1357: SAT11 EXP XSAT3-11358: SULEN! ?1359: SULEN! ?1350: SULEN! ? </td <td></td> <td></td> <td></td>			
1337EXTERNAL SORED.SERA.SERACT, XSAT1, XSAT2, XSAT3, XSAT4, XSAT5, XSAT6, XSAT71338EXTERNAL XSAT8, SMS17.HISWAP.DIFF, CONVMO, BLOCKS, XSAT3, XSAT4, XSAT611349>1540IFFTCHECK+FTMONP,1541IOATA AND STORAGE AREA FOR SWAPPING. ON THE 278 DISK, DISKS1543; # & 17 ARE USED FOR SWAPPING, EACH DISK CONTAINING1544;1545INTERNAL1546INTERNAL1547SORED: 71548; C(RH)=LOC, OF SIZE OF READ/WRITE1549SOREU: 71551SERA: 71551SERA: 71553SERA: 71554SERACTI #1555SULEN: #1556SULEN: #1556YAT1: EXP XSAT3.1557XSAT1: EXP XSAT3.11558YAT1: EXP XSAT3.11559YSAT2: XKD 4000#, IZ-MOR DITS AFTER READING, IZ-MOR DITS AFTER		IEN FICHECK+FIMONP.C	
1538EXTERNAL XSAT8,SwPSI7,HISWAP,DIFF,CONVMD,BLOCKS,XSAT31,XSAT4P,XSAT611549IFE FTCHECK+FTMONP,1540IFE FTCHECK+FTMONP,1541IDATA AND STORAGE AREA FOR SWAPPING, ON THE 270 DISK, DISKS1542IDATA AND STORAGE AREA FOR SWAPPING, EACH DISK CONTAINING1543; # 4 17 ARE USED FOR SWAPPING, EACH DISK CONTAINING1544; # 4 17 ARE USED FOR SWAPPING, EACH DISK CONTAINING1545: CICH1: ECORDS,1546INTERNALSOREO,SERA,SERACT1547SOREO: 7: CICH1:LOC, OF FIRST WORD TO READ/WRITE1548: CORP OF SOREO, IN CASE OF1549ESOREU: 7: CICH1:LOC, OF FIRST WORD TO READ/WRITE1549: SOREU: 7: CICH1:LOC, OF FIRST WORD TO READ/WRITE1550: SERA: 7: CICH1:LOC, OF FIRST WORD TO READ/WRITE1551: SERA: 7: CICH1:LOC, OF FIRST WORD TO READ/WRITE1553: SUBEN: 7: CICH1:LOC, OF FIRST WORD TO READ/WRITE1554: SERACT: 7: CICH1:LOC, OF FIRST WORD TO READ/WRITE1555: SULEN: 7: CICH1:LOC, OF FIRST WORD, CICHNER, C			SAT1, XSAT2, XSAT3, XSAT4, XSAT5, XSAT6, XSAT7
1339>1540IFE FTCHECK+FTMONP,1541IDATA AND STORAGE AREA FOR SWAPPING, EACH DISK CONTAINING1542IDATA AND STORAGE AREA FOR SWAPPING, EACH DISK CONTAINING1543. # # # # # # # # # # # # # # # # # # #			
1540IFE FTCHECK+FTMONP.1541;DATA AND STORAGE AREA FOR SWAPPING, ON THE 270 DISK, DISKS1542;DATA AND STORAGE AREA FOR SWAPPING, EACH DISK CONTAINING1543;DATA AND STORAGE AREA FOR SWAPPING, EACH DISK CONTAINING1544;DATA AND STORAGE AREA FOR SWAPPING, EACH DISK CONTAINING1545; A 17 ARE USED FOR SWAPPING, EACH DISK CONTAINING1546INTERNALSOREOLSERA,SERACT1547SOREOL 7;C(RL)=NEG, OF SIZE OF READ/WRITE1548:SOREOL 7;C(RL)=NEG, IN CASE OF1559SERAL 7;SIGN IS 1 IF A READ1551SERAL 7;SIGN IS 1 IF A READ1552:C(RL)=NEGR NUMBER BEFORE READING,1553:SERACTI 8;CONTER FOR EROER READING,1554SERACTI 8;CONTER FOR EROER READING,1555:SULEN! 9;AMOUNT TRANSFERRED SO FAR - FRAG SEG1556:STATI! EXP XSAT3-1;POINTER USED RY GETBIT, CLRBIT1557XSAT1! EXP XSAT3-1;POINTER TO BIT TABLE1558XSAT2! XWD -SWPSIZ,XSAT4; BIT TABLE FOR SWAPPING ALLOCATION1560:SPESIZ,XSAT4; BIT TABLE FOR SWAPPING ALLOCATION1561:SERO WORDS AT TOP OF TABLE1564:SERO WORDS AT TOP OF TABLE1565:E1=BLOCKS/*D36;LEADING ZEROES IN W1+ST WORD1566:E1=BLOCKS/*D36;LEADING ZEROES IN W1+W2+2ND WORD1567:SERT2, W2+N36;LEADING ZEROES IN W1+W2+2ND WORD1568:F2=EX-W22+N36;LEADING ZEROES IN W1+W2+2ND WORD1569:YE XSAT31,XSAT3+11570:YE XSAT31,X			
154110 ATA AND STORAGE AREA FOR SWAPPING, ON THE 270 DISK, DISKS1542; 0 & 17 ARE USED FOR SWAPPING, EACH DISK CONTAINING1544; 5400 (OCTAL) RECORDS,1544; 5400 (OCTAL) RECORDS,1545INTERNALSOREG, SERA, SERACT1546INTERNALSOREG, SERA, SERACT1547SOREG1 7; C(RH)=LOC, OF FIRST WORD TO READ/WRITE1548is converted and is a stress of the stress		-	
1542:DATA AND STORAGE AREA FOR SWAPPING, ON THE 270 DISK, DISKS1543: A TA ARE USED FOR SWAPPING, EACH DISK CONTAINING1544: 5400 (OCTAL) RECORDS.1545:1546INTERNAL1547SOREQ: 71548: (C(H)=NEG, OF SIZE OF READ/WRITE1549: SOREQ: 71550: (C(H)=LOC, OF FIRST WORD TO READ/WRITE1551SERA: 71552: C(RY)=LOC, OF SOREG, IN CASE OF1553SERA: 71554: SERACT: 01555: COUPY OF SOREG SINCHARER BEFORE READING,1556: ERROR BITS AFTER READING,1557: SULEN: 01558: COUPY CF OR ERRORS1559: SULEN: 01550: SULEN: 01551: SERACT: 11555: SULEN: 01556: SULEN: 01557: SAT1: EXP XSAT3-11558: SULEN: 01559: SULEN: 01560: SAT2: XHO -SWPSIZ,XSAT41561: BIT TABLE FOR SWAPPING ALLOCATION1563: YP BLOCKS,54001564: BIT TABLE FOR SWAPPING ALLOCATION1565: SURSKS/0361566: LEBLOCKS,5408: NUMBER OF RECORDS/DISK PLATE1566: EI=HOCKS+IE+036: SERO FOR EROBS AT DOP OF TABLE1567: ZEX-W2*/036: SURSKS/C36: LEADING ZEROES IN W1+W2*2ND WORD1568: ZEX-W2*/036: SURSKS/C364: LEADING ZEROES IN W1+W2*2ND WORD1569: X=HOCK W1: SAT31, XSAT3+1 <t< td=""><td></td><td></td><td></td></t<>			
1543:Ø. Å 17 ÅRE UŠED FOR SWAPPINĠ, EACH DISK CONTÁINÍNG1544:5400 (OCTAL) RECORDS.1545INTERNALSQREQ: 71546INTERNALSQREQ: 71547SQREQ: 7IC(LH)=NEG. OF SIZE OF READ/WRITE1548:.C(RH)=LQC. OF FIRST WORD TO READ/WRITE1549ESQREQ: 7.C(RH)=LQC. OF FIRST WORD TO READ/WRITE1549ESQREQ: 7.C(RH)=REQC NC ASE OF1550SERA: 7.C(RH)=REQC NUMBER BEFORE READING.1551SERA: 7.C(RH)=REQC NUMBER BEFORE READING.1552		DATA AND STORAGE AREA FOR S	WAPPING, ON THE 270 DISK, DISKS
1544:5400 (OCTAL) RECORDS,1545INTERNALSOREQ: SERA, SERACT1546INTERNALSOREQ: P1547SOREQ: PIC(LH)=NEG, OF SIZE OF READ/WRITE1548ISOREQ: PIC(RH)=LOC, OF FIRST WORD TO READ/WRITE1549ESOREQ: PJCOPY OF SOREQ, IN CASE OF1550SERA: PISIGN IS 1 IF A READ1551SERA: PISIGN IS 1 IF A READ1552SERACT: RICOUNTER FOR ERADING,1553SERACT: RICOUNTER FOR ERADING,1554SERACT: RICOUNTER FOR ERADING,1555SQLEN: RIMOUNT TRANSFERED SO FAR - FRAG SEG1556XSAT11 EXP XSAT3-1JPOINTER USED BY GETBIT, CLRBIT1558XSAT21 XHD -SWPSIZ, XSAT4JPOINTER TO BIT TABLE1559XSAT31 XHD 400000,IZ-HORD TABLE ALTERED BY GETBIT, CLRBIT1560YP BLOCKS, 5400INUMBER OF RECORDS/DISK PLATE1561YP BLOCKS, 5400ILEADING ZEROES IN WI-IST WORD1565F2=EX-W2+D36JZERO WORDS AT FOP OF TABLE1566F2=EX-W2+D36JZERO WORDS AT BOTTOM OF TABLE1569YP XSAT31, XSAT3+11570XP XSAT31, XSAT3+11571XSAT41 RLOCK W11572XSAT41 RLOCK W1			
1546 INTERNAL SQREQ, SERA, SERACT 1547 SQREQ; 7 IC(LH)=NEG, OF SIZE OF READ/WRITE 1548 IC(RH)=LOC, OF FIRST WORD TO READ/WRITE 1549 ESOREU; 7 IC(RH)=LOC, OF FIRST WORD TO READ/WRITE 1550 IC(RH)=LOC, OF FIRST WORD TO READ/WRITE IC(RH)=LOC, OF FIRST WORD TO READ/WRITE 1550 IC(RH)=LOC, OF SQREQ, IN CASE OF IC(REOR IN FRACMENTE JOB 1551 SERA: ISIGN IS 1 IF A READ 1552 IC(RH)=LOC K NUMBER BFFORE READING, IC(RH)=LOC K NUMBER BFFORE READING, 1553 SERACTI R ICOUNTER FOR ERRORS 1554 SERACTI R ICOUNTER FOR ERRORS 1555 SQLEN: Ø IAMOUNT TRANSFERRED SO FAR - FRAG SEG 1556 XSAT1: EXP XSAT3-1 IPOINTER USED RY GETBIT, CLRBIT 1556 XSAT2: XWD -SWPSIZ,XSAT4 IPOINTER TO BIT TABLE 1560 XSAT3: XWD 40000, I2-WORD TABLE ALTERED BY GETBIT, CLRBIT 1561 INUMER OF RECORDS/DISK PLATE INUMER OF RECORDS/DISK PLATE 1562 I BIT TABLE FOR SWAPPING ALLOCATION IXWD -SWPSIZ,XSAT4 1563 XP BLOCKS,5400 INUMER OF ZERO WORDS AT TOP OF TABLE 1564 IELBLOCKS-			
1546 INTERNAL SQREQ, SERA, SERACT 1547 SQREQ; 7 IC(LH)=NEG, OF SIZE OF READ/WRITE 1548 IC(RH)=LOC, OF FIRST WORD TO READ/WRITE 1549 ESOREU; 7 IC(RH)=LOC, OF FIRST WORD TO READ/WRITE 1550 IC(RH)=LOC, OF FIRST WORD TO READ/WRITE IC(RH)=LOC, OF FIRST WORD TO READ/WRITE 1550 IC(RH)=LOC, OF SQREQ, IN CASE OF IC(REOR IN FRACMENTE JOB 1551 SERA: ISIGN IS 1 IF A READ 1552 IC(RH)=LOC K NUMBER BFFORE READING, IC(RH)=LOC K NUMBER BFFORE READING, 1553 SERACTI R ICOUNTER FOR ERRORS 1554 SERACTI R ICOUNTER FOR ERRORS 1555 SQLEN: Ø IAMOUNT TRANSFERRED SO FAR - FRAG SEG 1556 XSAT1: EXP XSAT3-1 IPOINTER USED RY GETBIT, CLRBIT 1556 XSAT2: XWD -SWPSIZ,XSAT4 IPOINTER TO BIT TABLE 1560 XSAT3: XWD 40000, I2-WORD TABLE ALTERED BY GETBIT, CLRBIT 1561 INUMER OF RECORDS/DISK PLATE INUMER OF RECORDS/DISK PLATE 1562 I BIT TABLE FOR SWAPPING ALLOCATION IXWD -SWPSIZ,XSAT4 1563 XP BLOCKS,5400 INUMER OF ZERO WORDS AT TOP OF TABLE 1564 IELBLOCKS-	1545	· · · · · · · · · · · · · · · · · · ·	
1547 SQREQ: 7 IC(L)=NEG. OF SIZE OF READ/WRITE 1548 IC(L)=NEG. OF SIZE OF READ/WRITE 1548 IC(H)=LOC. OF FIRST WORD TO READ/WRITE 1549 ESOREU: 7 IC(H)=LOC. OF FIRST WORD TO READ/WRITE 1550 IC(H)=NEG. OF SIZE OF READ/WRITE IC(H)=NEG. OF SIZE OF READ/WRITE 1551 ESOREU: 7 IC(H)=NEG. OF SIZE OF READ/WRITE 1551 SERA: 7 ISGN IS 1 IF A READ 1552 IF RROR IN FRACMENTED JOB IF AREAD 1553 SERA: 7 ISGN IS 1 IF A READ 1554 SERACT: 0 ICONTER FOR ERAORS 1555 SULEN: 0 ICONTER FOR ERAORS 1556 SULEN: 0 IAMOUNT TRANSFERED SO FAR - FRAG SEG 1556 SULEN: 0 IAMOUNT TRANSFERED SO FAR - FRAG SEG 1557 XSAT1: EXP XSAT3-1 IPOINTER USED BY GETBIT.CLRBIT 1568 XND -SWPSIZ.XSAT4 IPOINTER TO BIT TABLE 1560 XND -SWPSIZ.XSAT4 IPOINTER TO BIT TABLE ALTERED BY GETBIT.CLRBIT 1562 IBIT TABLE FOR SWAPPING ALLOCATION 1563 YP BLOCKS.+1036 ILEADING ZEROES IN W1+1ST WORD 1564 W1=BLOCKS.+1036 ILEADING ZEROES IN		INTERNAL SQREQ, SERA, S	ERACT
1549ESOREU: 7JCOPY OF SOREG, IN CASE OF1550SERA: 7JEROR IN FRAGMENTED JOB1551SERA: 7JSIGN IS IF A READ1553FRAGTJCONTER FOR ERADING,1554SERACT: 0JCONTER FOR ERRORS1556JS56JCONTER FOR ERRORS1557XSAT1: EXP XSAT3-1JPOINTER USED RY GETBIT, CLRBIT1558XSAT2: XWD -SWPSIZ, XSAT4JPOINTER USED RY GETBIT, CLRBIT1560XSAT3: XWD 400000,J2-WORD TABLE ALTERED BY GETBIT, CLRBIT1561XWD -SWPSIZ, XSAT4JNUMBER OF ZECOMS/DISK PLATE1562YP BLOCKS, 5400JNUMBER OF ZECOMS/DISK PLATE1564W1=BLOCKS/+D36JLEADING ZEROES IN W1+1ST WORD1565E1=BLOCKS-W1+TD36JZERO WORDS AT BOTTOM OF TABLE1566F2=EX-W2+D36JZERO WORDS AT BOTTOM OF TABLE1569XP XSAT31, XSAT3+11570XP ASAT31, XSAT3+11571XSAT4: RLOCK W12-1			
1550SERA:7I ERROR IN FRACMENTED JOB1551SERA:7ISIGN IS 1 IF A READ1552I CRNH-BLOCK NUMBER BEFORE READING,1553SERACT:01554SERACT:01555SULEN:01556XSAT1:EXP XSAT3-11558XSAT2:XWD -SWPSIZ,XSAT41560XSAT3:XWD 400000,1561XWD -SWPSIZ,XSAT41562I BIT TABLE FOR SWAPPING ALLOCATION1563XP BLOCKS,54001566INUMBER OF ZECORS/DISK PLATE1566ISECOKS/+D361566E1=BLOCKS/+D361566E2=BLOCKS/+D361567W2=EX/+D361568F2=EX-W2++D361569XP XSAT31, XSAT3+11570XP ASAT31, XSAT3+11571XSAT4:1572XSAT4:1572XSAT4:	1548		; C(RH)=LOC, OF FIRST WORD TO READ/WRITE
1551SERA: 7ISIGN IS 1 IF A READ1552ISIGN IS 1 IF A READ1553ICRNP BELOCK NUMBER BEFORE READING,1554SERACT: Ø1555SULEN! Ø1556IAMOUNT TRANSFERRED SO FAR - FRAG SEG1557XSAT1: EXP XSAT3-11558XSAT2! XHD -SWPSIZ,XSAT41559XSAT3: XHD 4ØØØØØ,1560ISIT TABLE FOR SHAPPING ALLOCATION1561INUMBER OF RECORDS/DISK PLATE1562ISIT TABLE FOR SMAPPING ALLOCATION1563XP BLOCKS,54ØØ1564ISIT TABLE FOR SHAPPING ALLOCATION1565ISIT CKS+U1*D361566ISIENCKS+U1*D361567W2=EX/+D361568F2=EX-W2*D361570XP XSAT31,XSAT3+11570XP XSAT31,XSAT3+11571XSAT4: RLOCK W11572XSAT5: BLOCK V2+1	1549	ESOREU: 7	ICOPY OF SOREQ, IN CASE OF
1552j C(RH)=RLOCK NUMBER BEFORE READING,1553SERACT: Ø1554SERACT: Ø1555SULEN: Ø1556JAMOUNT TRANSFERRED SO FAR - FRAG SEG1557XSAT1: EXP XSAT3-11558XSAT2: XWD -SWPSIZ,XSAT41559XSAT3: XWD 40000,1560J2-WORD TABLE ALTERED BY GETBIT,CLRBIT1561XWD -SWPSIZ,XSAT41562I BIT TABLE FOR SWAPPING ALLOCATION1563XP BLOCKS,54001564JUMBER OF RECORDS/DISK PLATE1565E1=BLOCKS+W1**D361566F2=EX-W2**D361568F2=EX-W2**D361569XP XSAT31,XSAT3+11570XP XSAT31,XSAT3+11571XSAT4: RLOCK W11572XSAT5: BLOCK W2+1	1550		; ERROR IN FRAGMENTED JOB
1553; ERROR BITS AFTER READING,1554SERACT: Ø1555SÜLEN: Ø1556idount transferred so far - frag seg1556xsat1: EXP xsat3-11557xsat1: EXP xsat3-11558xsat2: xWD -sWPSIZ,xsat41559xsat3: xWD 400000,1560xwD -sWPSIZ,xsat41561iz-word table altered by GETBIT,CLRBIT1562i BIT Table for swapping allocation1563xP BLOCKS,54001564w1=BLOCKS+M1*tD361565E1=BLOCKS+M1*tD361566E1=BLOCKS+M1*tD361567w2=EX/tD361568file for swapping allocation1569xP BLOCKS,54001564w1=BLOCKS+M1*tD361565file for swapping allocation1566file for swapping allocation1567w2=EX/tD361568file for swapping allocation1569xP xsat31, xsat3+11570xsat4: RLOCK w11571xsat5: BLOCK w2+1	1551	SERA: Z	ISIGN IS 1 IF A READ
1554SERACT: ØJCOUNTER FOR ERRORS1555SGLEN: ØJAMOUNT TRANSFERRED SO FAR - FRAG SEG1556XSAT1: EXP XSAT3-1JPOINTER USED BY GETBIT, CLRBIT1558XSAT2: XWD -SWPSIZ, XSAT4JPOINTER TO BIT TABLE1559XSAT3: XWD 4ØØØØØ, XWD -SWPSIZ, XSAT4J2-WORD TABLE ALTERED BY GETBIT, CLRBIT1560XWD -SWPSIZ, XSAT4JUNMBER OF RECORDS/DISK PLATE1561XWD -SWPSIZ, XSAT4JNUMBER OF RECORDS/DISK PLATE1563XP BLOCKS, 54ØØJNUMBER OF RECORDS/DISK PLATE1564M1=BLOCKS/*D36JLEADING ZEROES IN W1+1ST WORD1565E1=BLOCKS+E1-*D35JZERO WORDS AT BOTTOM OF TABLE1568F2=EX-W2**D36JLEADING ZEROES IN W1+W2*2ND WORD1570XP XSAT31, XSAT3*11571XSAT4: RLOCK W11572XSAT5: BLOCK W2+1	1552		; C(RH)=BLOCK NUMBER BEFORE READING,
1555SQLEN: 0JAMOUNT TRANSFERED SO FAR - FRAG SEG1556XSAT1: EXP XSAT3-1JPOINTER USED BY GETBIT, CLRBIT1558XSAT2: XWD -SWPSIZ, XSAT4JPOINTER TO BIT TABLE1559XSAT3: XWD 40000, XWD -SWPSIZ, XSAT4J2-WORD TABLE ALTERED BY GETBIT, CLRBIT1560XWD -SWPSIZ, XSAT4JNUMBER OF RECORDS/DISK PLATE1563XP BLOCKS, 5400JNUMBER OF RECORDS/DISK PLATE1564E1=BLOCKS-+D36JLEADING ZEROES IN W1+IST WORD1565E2=SUCKS+E1-+D35JZERO WORDS AT BOTTOM OF TABLE1568F2=EX-W2+D36JLEADING ZEROES IN W1+W2+2ND WORD1570XP XSAT31, XSAT3+1XSAT4: RLOCK W11572XSAT5: BLOCK W2+1	1553		I ERROR BITS AFTER READING,
1556XSAT1:EXPXSAT3-1JPOINTERUSEDBYGETBIT, CLRBIT1557XSAT2:XWD-SWPSIZ, XSAT4JPOINTERTABLEI1559XSAT3:XWD400000,J2-WORDTABLEALTEREDBYGETBIT, CLRBIT1560XWD-SWPSIZ, XSAT4J2-WORDTABLEALTEREDBYGETBIT, CLRBIT1561XWD-SWPSIZ, XSAT4J2-WORDTABLEFORSWAPPINGALLOCATION1563XPBLOCKS, 5470INUMBEROFRECORDS/DISKPLATE1564W1=BLOCKS/+D36INUMBEROFZERO WORDSATTOPOFTABLE1565E1=BLOCKS-W1*+D36ILEADINGZEROESIN <w1+st< td="">WORDSAT4EKAT31, XSAT3+1SAT41:RLOCKKMKAT41:KAT41:RLOCKKMKAT41:KAT41:RLOCKKMKAT41:<td></td><td>SERACT: Ø</td><td>ICOUNTER FOR ERRORS</td></w1+st<>		SERACT: Ø	ICOUNTER FOR ERRORS
1557 XSAT1: EXP XSAT3-1 JPOINTER USED BY GETBIT, CLRBIT 1558 XSAT2: XWD -SWPSIZ, XSAT4 JPOINTER TO BIT TABLE 1559 XSAT3: XWD 40000, J2-WORD TABLE ALTERED BY GETBIT, CLRBIT 1560 XWD -SWPSIZ, XSAT4 JPOINTER TO BIT TABLE 1561 XWD -SWPSIZ, XSAT4 JUNUBER OF RECORDS/DISK PLATE 1562 JNUMBER OF RECORDS/DISK PLATE JNUMBER OF ZEROW WORDS AT TOP OF TABLE 1563 XP BLOCKS, 5400 JULMBER OF ZEROW WORDS AT TOP OF TABLE 1564 W1=BLOCKS/+D36 JLEADING ZEROES IN W1+1ST WORD 1565 E1=BLOCKS+E1-+D35 JZERO WORDS AT BOTTOM OF TABLE 1568 F2=EX+WZe+D36 JLEADING ZEROES IN W1+W2+2NO WORD 1569 XP XSAT31, XSAT3+1 JSAT3: RLOCK W1 1570 XSAT3: RLOCK W1 YSAT3: BLOCK W2+1	1555	SQLEN: 0	IAMOUNT TRANSFERRED SO FAR - FRAG SEG
1558 XSAT2: XWD -SWPSIZ,XSAT4 ; POINTER TO BIT TABLE 1559 XSAT3: XWD 40000, ;2-WORD TABLE ALTERED BY GETBIT,CLRBIT 1560 XWD -SWPSIZ,XSAT4 ; BIT TABLE FOR SWAPPING ALLOCATION 1561 XWD -SWPSIZ,XSAT4 ; BIT TABLE FOR SWAPPING ALLOCATION 1562 ; BIT TABLE FOR SWAPPING ALLOCATION 1563 XP BLOCKS,5400 ; NUMBER OF RECORDS/DISK PLATE 1564 W1=BLOCKS/*D36 ; LEADING ZEROES IN W1*1ST WORD 1565 E1=BLOCKS+W1*D36 ; LEADING ZEROES IN W1*1ST WORD 1566 F2=EX-W2*D36 ; LEADING ZEROES IN W1*W2*2NO WORD 1569 XP XSAT31,XSAT3*1 ; SAT4: RLOCK W1 1570 XSAT4: RLOCK W1 ; SAT5: BLOCK W2*1	1556		
1559 XSAT3: XWD 40000, XWD -SWPSIZ,XSAT4 ;2-WORD TABLE ALTERED BY GETBIT,CLRBIT 1560 XWD -SWPSIZ,XSAT4 ; 1561 XWD -SWPSIZ,XSAT4 ; 1562 ; BIT TABLE FOR SWAPPING ALLOCATION 1563 XP BLOCKS,5400 JNUMBER OF RECORDS/DISK PLATE 1564 W1=BLOCKS/+D36 ILEADING ZEROES IN W1+1ST WORD 1565 E1=BLOCKS+W1*+D36 JLEADING ZEROES IN W1+1ST WORD 1566 EX=BLOCKS+E1-+D35 ;ZERO WORDS AT BOTTOM OF TABLE 1568 F2=EX-W2*+D36 ;LEADING ZEROES IN W1+W2+2ND WORD 1569 XP XSAT31,XSAT3+1 1571 XSAT4: RLOCK W1 1572 XSAT5: BLOCK W2+1	1557	XSAT1: EXP XSAT3-1	POINTER USED BY GETBIT, CLRBIT
1560 XWD -SWPSIZ,XSAT4 1561 : BIT TABLE FOR SWAPPING ALLOCATION 1562 : BIT TABLE FOR SWAPPING ALLOCATION 1563 XP BLOCKS,5400 INUMBER OF RECORDS/DISK PLATE 1564 W1=BLOCKS/*D36 INUMBER OF ZERO WORDS AT TOP OF TABLE 1565 E1=BLOCKS-W1**D36 ILEADING ZEROES IN W1*1ST WORD 1566 EX=BLOCKS+E1-*D35 IZERO WORDS AT BOTTOM OF TABLE 1568 F2=EX-W2**D36 ILEADING ZEROES IN W1*W2*2ND WORD 1569 XP XSAT31, XSAT3*1 IST1 1570 XSAT4: RLOCK W1 XSAT5: BLOCK W2*1	1558	XSAT2: XWD -SWPSIZ,XSAT4	IPOINTER TO BIT TABLE
1561 ; BIT TABLE FOR SWAPPING ALLOCATION 1562 ; BIT TABLE FOR SWAPPING ALLOCATION 1563 XP BLOCKS,5400 JNUMBER OF RECORDS/DISK PLATE 1564 W1=BLOCKS/+D36 INUMBER OF ZERO WORDS AT TOP OF TABLE 1565 E1=BLOCKS+W1*+D36 JLEADING ZEROES IN W1+1ST WORD 1566 EX=BLOCKS+E1-+D35 JZERO WORDS AT BOTTOM OF TABLE 1567 W2=EX/+D36 JZEROES IN W1+W2+2NO WORD 1568 F2=EX-W2*+D36 JLEADING ZEROES IN W1+W2+2NO WORD 1569 XP XSAT31,XSAT3+1 JS70 1571 XSAT4: RLOCK W1 JSAT5: BLOCK W2+1			12-WORD TABLE ALTERED BY GETBIT, CLRBIT
1562 ; BIT TABLE FOR SWAPPING ALLOCATION 1563 XP BLOCKS,5400 JNUMBER OF RECORDS/DISK PLATE 1564 W1=BLOCKS/+D36 JNUMBER OF ZERO WORDS AT TOP OF TABLE 1565 E1=BLOCKS-W1*+D36 JLEADING ZEROES IN W1+IST WORD 1566 EX=BLOCKS+E1-+D35 JZERO WORDS AT BOTTOM OF TABLE 1567 W2=EX/+D36 JLEADING ZEROES IN W1+W2+2ND WORD 1569 F2=EX-W2*+D36 JLEADING ZEROES IN W1+W2+2ND WORD 1570 XSAT31,XSAT3+1 JS71 1572 XSAT5: BLOCK W2+1 KATS		XWD -SWPSIZ,XSAT4	
1563 XP BLOCKS,5400 INUMBER OF RECORDS/DISK PLATE 1564 W1=BLOCKS/*D36 INUMBER OF ZERO WORDS AT TOP OF TABLE 1565 E1=BLOCKS-W1**D36 ILEADING ZEROES IN W1*1ST WORD 1566 EX=BLOCKS+E1-*D35 IZERO WORDS AT BOTTOM OF TABLE 1568 F2=EX-W2**D36 IZERO WORDS AT BOTTOM OF TABLE 1569 XP XSAT31,XSAT3*1 IZEADING ZEROES IN W1*W2*2ND WORD 1571 XSAT4: RLOCK W1 IST1 1572 XSAT5; BLOCK W2+1 IST1			
1564 W1=BLOCKS/+D36 INUMBER OF ZERO WORDS AT TOP OF TABLE 1565 E1=BLOCKS-W1*+D36 JLEADING ZEROES IN W1+1ST WORD 1566 EX=BLOCKS+E1-+D35 1567 W2=EX/+D36 JZERO WORDS AT BOTTOM OF TABLE 1568 F2=EX-W2++D36 JLEADING ZEROES IN W1+W2+2ND WORD 1569 XP XSAT31,XSAT3+1 1571 XSAT4: RLOCK W1 1572 XSAT5: BLOCK W2+1			
1565 E1=BLOCKS-W1**D36 JLEADING ZEROES IN W1*1ST WORD 1566 EX=BLOCKS+E1-*D35 1567 W2=EX/*D36 JZERO WORDS AT BOTTOM OF TABLE 1568 F2=EX-#D36 JLEADING ZEROES IN W1*W2*2ND WORD 1569 XP XSAT31,XSAT3*1 1570 XSAT4: RLOCK W1 1572 XSAT5: BLOCK W2+1			
1566 EX=BLOCKS+E1-+035 1567 W2=EX/+D36 1568 F2=EX-W2+D36 1569 JLEADING ZEROES IN W1+W2+2ND WORD 1570 XP XSAT31,XSAT3+1 1571 XSAT4: RLOCK W1 1572 XSAT5: BLOCK W2+1			
1567 W2=EX/+D36 JZERO WORDS AT BOTTOM OF TABLE 1568 F2=EX-W2++D36 JLEADING ZEROES IN W1+W2+2ND WORD 1569 XP XSAT31,XSAT3+1 1571 XSAT4: RLOCK W1 1572 XSAT5: BLOCK W2+1			ILEADING ZEROES IN W1+1ST WORD
1568 F2=EX-W2+D36 ;LEADING ZEROES IN W1+W2+2ND WORD 1569 XP XSAT31,XSAT3+1 1571 XSAT4: RLOCK W1 1572 XSAT5: BLOCK W2+1			
1569 1570 XP XSAT31,XSAT3+1 1571 XSAT4: RLOCK W1 1572 XSAT5: BLOCK W2+1			
1570 XP XSAT31,XSAT3+1 1571 XSAT4: RLOCK W1 1572 XSAT5: BLOCK W2+1		E2=EX-W2++D36	ILEADING ZEROES IN W1+W2+2ND WORD
1571 XSAT4: RLOCK W1 1572 XSAT5: BLOCK W2+1			
1572 XSAT5: BLOCK W2+1			
15/3 XSAT6: BLOCK 1	• • •		
	15/3	XSAI6: BLUCK 1	

SCHEDB -	- SCHEDULING ALGORITHM FOR SWAPPING SYSTEM(10/57)(BURFOUGHS DISK)	-69 PAGE 52-2
1574		
1575	REPEAT 1.<	
1576	1FE F1, <x=184></x=184>	
1577	IFN £1, <x=1< td=""><td></td></x=1<>	
1578	REPFAT +035-F1, <x=x+2>></x=x+2>	
1579	1FE [2, <z=-1></z=-1>	
1590	tEN E2, <7=?	
1581	v=1	
1532	PEPEAT + 936-12, <7=7+Y	
1583	Y = Y \$ 2>	
1584	>>	
1585	XP XSAT61,XSAT6-1	
1586	XSAT7: FXP X	
1597	XSATS: EXP 7	
1588		
1589	SWPSIZ=XSAT6-XSAT4+1: ISIZE OF TABLE	
1590	HISWAP=17#PLOCKS JLOGICAL BLOCK NUMBER OF FIRST	
1591	I BLOCK ON DISK	
1592	DIFF=HISMAP-BLOCKS-1	
1593	XP CONVERSION FROM 14 CORE BLOCKS TO 128 WORD	
1594	(DISC BLOCKS(SHIFT COUNT)	
1595		
1596	XP BLKSPK,CONVMD INO, OF BLOCKS PER K, SAME AS CONVMD	
1597	XSAT4P: XWP XSAT4,XSAT4+1	
1598	>	
1599	> ;END OF SWPSER LOGIC FOR THE OLD PDP-6 DISK.	

-

SCHEDB	- SCHEDU Swpser	JLING ALC R.KRASIM	GORITHM.F NZAF TS4.	OR SWAPF 34 03 F	PING SYS TER A9	TEM(10/50)(BURRO V426	UGHS DISK)	MACRO.V36 19:	10 4-JUN-69 PAGE 53
1600				TEN	⊂TRC10				
1671						FOR THE NEW PDP-	14 (1005) 06 441	DI SK	
1602				JUNIAL	COGIC	FOR THE NEW FUP=	IN CHUNEL RCAIRS	DISK	
1633				INTERNA		GXSAT,FXSAT,SW	D I A. I		
1604				EXTERNA		GETBIT, CLRBIT	F I 'V I		
1605							Pop (1		
1626				CALC: NA		Content (ec-0) I	- U- JI		
1607									
1608				:SUBROU	ITINE "G	XSAT" IS CALLED	TO FIND & SERIES		EBEE DI OCKO ON
1609				I THE D		SWAP SOME JOB OU	TO FIND A DENIES TONTO ITIS O	TALLED AT CLOCK	FREE BLOCKS UN
1610						and 3000 500 50		ALLED AT GLOCK L	_ [V [] •
1611				CALLIN	G SEQUE	NCF			
1612				1	PUSHJ	PDP, GXSAT			
1613				;	ERROR	EXIT THE DISK	IS FULL. NO SWA	PPING SPACE AVA	TLABLE.
1614				;	NORMAL	EXIT			TEROLE.
1615				; ENTRY	CONDITI	0N5			
1616				;	C(DAT)	= NUMBER OF 1K E	BLOCKS OF DISK S	TORAGE NEEDED.	
1617				;EXIT C	ONDITIO	NS			
1618				;	C(TAC)	= LOGICAL BLOCK	NUMBER (DIVIDED	BY 8) OF THE FI	IRST DISK BLOCK IN
1619				;		THE SERIES OF	F CONSECUTIVE BL	OCKS WHICH SATIS	SEY THIS REDUEST.
1620				JACCUMU	LATORS	DAT, TAC, TAC1, A	AC1, AND AC2 ARE	DESTROYED BY TH	IS SUBROUTINE.
1621									
1622		261140		GXSATI		PDP,ITEM	ITHIS ROUTINE	SAVES AND RESTOR	RES ACCUMULATOR "ITEM".
1623 1624	ØØ1207	2002:10	000005		MOVE	ITEM, DAT			
1625	ØØ121Ø ØØ1211	201640 200700	001301' 001301'		MOVET	AC1, SWPENT	ISET UP ENTRY	CONDITIONS FOR T	THE "GETBIT" SUBROUTINE,
1626	001211	260140			MOVE	AC2, SWPENT			
1627	001212	254070			PUSHJ JRST	PDP,GETBIT			
1628	001213		777777		ADDI	IPOPJ	INO ROOM IN TH	E SWAPPING PART	OF THE DISK.
1629	001215	313100			CAMLE	TAC1,-1(ITEM) TAC1,MAXSNP	IRUUM FOUNDC	OMPUTE LOGICAL B	BLOCK NUMBER OF THIS
163Ø	001210	2021-20	001300'		MOVEM	TAC1, MAXSWP	IFIRST BLUCK I	N A SERIES OF CO	DNSECUTIVE FREE DISK
1631	001217	242130	0000003			TAC1, BLKSPK	18400K5, 186	"SWPIAB" TABLE S	SEARCHED HAS ONE BIT
1632	ØØ1220	274100	607000			TAC1, LBHIGH	IPACKWADDE (40	T SWAPPING DISK	SPACE, BUT IS STORED
1633		214040			MOVM	TAC, TAC1	IN SO HODATE #	A BIL OF SWPIAB	= LAST 1K OF DISK), IOUS MAXIMUM EXCEEDED.
1634	001222	242040			ISH TA	- BLKSPK	COMPRESS DICK	ADDRESS TO 17 D	UUS MAXIMUM EXUEEDED.
1635	001223	344040	000000		40.JA TA	AC, IPOPJ1		AND RESTORE ITEM	AND SKIP RETURN
							THE THE	NEW NEW OFFEN	AND SKIF KEIDAN

1636 1637 1638							TO RETURN A SERIES OF CONSECUTIVE DISK BLOCKS TO MAKING THEM AVAILABLE FOR RE-USE IN HANDLING
1639					C DAR PI	INSTRENOESIS.	IT IS CALLED AT CLOCK LEVEL.
1641 1642				ICALLIN	VG SEQUEN PUSHJ NORMAL	PUP, FXSAT	
1643 1644				ENTRY	CONDITIO	0NS	
1645 1646				;		SERIES WHICH	K NUMBER OF THE FIRST DISK BLOCK IN THE IS TO BE MADE AVAILABLE.
1647				;		ARE TO BE MADE	E AVAILABLE.
1649				;	THE REQ	NS Ruested bits in	THE SWAPPING SPACE AWAILARY ITY TARLE
1650 1651				; ; accumui			AC1, AND AC2 ARE DESTROYED BY THIS SUBROUTINE.
1652 1653		626240	407020	FXSAT:			
1654	001225	254240	0012471	1	JKST	TAC,FRGSEG FXSAT1	FRAGMENTED? FNO. DO IN REGULAR WAY
1655 1656	001227	55025 7	009071 009090	FRAGRK: FRGBK1:	HKRZ	AC3, TAC TAC, (AC3)	YES, LOC OF TABLE IN AC3
1657	001230	574257	000000 001234 •		нĽКЕ	DAT, (AC3)	LOC OF NEXT DISK ADDRESS
1659	ØØ1232	267140	0212471		PUSHJ	PDP, FXSAT1	IGIVE UP FREE CORE IF NOT REAL ADDRESS IGIVE UP THE DISK SPACE FOR THIS PART
1661	001234	550040	001227 · 000017	FRGBK2:	AURJP HRR⊉	AC3, FRGBK1 TAC, AC3	COUNT WORD OF TABLE, GET NEXT
1662 1663	ØØ1235 ØØ1236	261140	000004		PUSH	PDP, ITEM	
1664	001237	274040	000024		SUB	ITEM,AC3 TAC,ITEM	INUMBER OF WOS IN TABLE IPOINT TAC TO 1ST WORD OF BLOCK
1666	001241	350000	000004		LSH AOS	I TEM, -2 I TEM	14 WOS PER BIT
1668	ØØ1243	262140			PUSHJ POP	PDP,CLCOR1 PDP,ITEM	IGIVE UP FREE CORE
1669 1670	001245	254000	000000 001226 '		SKIPE JRST	TAČ, (AČ3) FRAGRK	JEND OF TABLE? JNO. GO CHASE NEXT PART
.671 .672		263140			POPJ	PDP,	IYES, DONE
1674	001250	2002%0	000004 000005			PDP,ITEM ITEM.DAT	ITHIS ROUTINE SAVES AND RESTORES ACCUMULATOR "ITER
675	ØØ1251	242:40	000003 001220'			RLKSPK	REGENERATE THREE LOW ORDER & BITS IN DICK ADDRESS
.677		217020	000001		MOVMS	TAC,LBHICH TAC	AVAILABILITY TABLE BY TRANSFORMING THE LOCICAL
.679	001255	211640	000003			TAC,1 AC1,BLKSPK	I BLOCK NUMBER TO AN EQUIVALENT BIT NUMBER IN THIS TARLE (WHICH IS STORED IN REVERSE ORDER AND HAS
	001257	274040			LSH	TAC, (AC1) TAC, DAT	I ONE BIT PER 1K OF SWAPPING SPACE ON THE DISK).
682 683	ØØ126Ø ØØ1261	201640 200720	ØØ1301' ØØ1301'		MOVEI	AC1, SWPENT	
684 685	001262	260140	002020	FXSATC:	PUSHJ	AC2,SWPENT PDP,CLRBIT	ISET UP ENTRY CONDITIONS FOR THE "CLRBIT" SUBROUTI ITHE "CLRBIT" SUBROUTINE IN "ISKSER" ACTUALLY DOES
686 687 688	ØØ1263	254000	0012131		JRST IPO)PJ	FITHE WORK OF CLEARING THE BITS. FRESTORE ITEM AND RETURN

SCHEDB			ORITHM F ZAF TS4.			YSTEM(10/50)(BUR¤ V406	201645 DISK) MACRO.V36 19:10 4-JUN-69 PAGE 54-1
1689 1690							SWAPPING AVAILABILITY TABLE, CALLED AT
1691 1692 1693				I SYSTE		TIALIZATION TIME. Rn virtal,swphgh	
1694							
1695	001264	200040	ØØ1252'	SWPINI:		TAC,LBHIGH	ISET HIGHEST LOGICAL BLOCK
1696	201265	202240	000000		MOVE	M TAC, SWPHGH	FOR SWAPPING IN COMMON
1697	001266	211040	000001		MOVN	1 TAC,1	ISET ENTIRE TABLE TO ONES, I.E., MARK EVERYTHING
1698	001267	200120	001321'		MOVE	TAC1, SWPENT	; AS BEING UMAVAILABLE FOR SWAPPING.
1699	901270	202042	000020		MOVE		
1720	801271	2531 0	ØØ1270'		AOBJ		
1701	001272	201640	001301'		MOVE		ITHEN USE THE "CLRBIT" ROUTINE TO CLEAR OUT AS MANY
1702	001273	200720	001301'		MOVE		BITS AT THE FRONT OF THE TABLE AS THERE ARE 1K DISK
1703	001274	200200	RABARA		MOVE		AREAS ALLOCATED FOR SWAPPING USE. THE PARAMETER
1724	PØ1275	202220	001035'		MOVE		ITOTAL AMOUNT OF VIRTUAL CORE ALLOWED
17.5	1012/2	2101.2 2	001002		1012	1 TENTAL	IS JUS EQUAL TO AMOUNT OF SWAPPING SPACE
							I EVEN IF USER CAN NOT HAVE ALL OF PHYSICAL USER CORE
1796							I EVEN IF OBER DAN NUT HAVE ALL OF PHIBIGAL OBER CORE

SCHEDB	- SCHEDU SWPSER	LING ALG R.KRASIN	GRITHM FOR VAF TS4,34	SWAPPING SYS Ø3 FER 69	TEM(10/50 V406	BURPOUGHS DISK) MACRD.V36 19:10 4-JUN-60 PAGE 61
1727	PØ1276	201040	002300	MOVE I	ТАС,Й	; "KASWAP" IS SET UP DURING DISK REFRESHING, AND
	001277	254030	ØØ1262'	JRŠT	CLREIT	; RECALLED FROM THE DISK AT EACH SYSTEM-INITIALIZE. ;IN LIEU OF "PUSHJ CLRBIT" FOLLOWED BY "POPJ".

SCHEDB - SCHEDULING ALGORITHY FOR SWAPPI'G SYSTEM(10/5%)(BURPOUGHS DISK) MACRO, V36 19:10 4-JUN-69 PAGE 62 SWESER R.KRASIN/AF 154.34 03 FER 69 - V426 TEN FTOGECKFETMONP, K L SCYSDT :DUMMY GLOPAL SYMBOL TO PERMIT THE SYSTEM BUILDER TO 1710 17:1 EXTERNAL 1712 ; RETRIEVE THE CORRECT BINARY COPY OF "SCHDAT 1713 EXTERNAL SORED, SERA, SERACT 1714 EXTERNAL LEHIGH, BLKSPK, MXX2SWP, SWPENT, SWPTAB, MAXSWP 1715 1716 IFE _____FTGHFCK+FTMONP, < ;DATA ASSOCIATED WITH THE SWPSER LOGIC FOR THE NEW PDP-10 DISK ----1717 1718 1719 1720 INTERN MXK2SWP, CONVMD, BLKSPK, SWPTAB, MAXSWP 1721 FXTERN K4SWAP 1722 THE ABOVE ARE REFERENCED OR INITIALIZED BY THE "ONCE" ROUTINE. 1723 1724 (C(LOWSWP)=LOWEST LOGICAL PLOCK NUMBER TO BE USED FOR SWAPPING DETERMINED BY ONCE ONLY REFRESH DIALOG (K45WAP)=MAX, NO, OF K DISK WORDS ALLOCATED FRO SWAPPING, ALSO (DETERMINED BY ONCE ONLY REFRESH DIALOG 1725 1726 1727 001300 000000 000000 MAXSEP: 2 IMAX. NO. OF K FROM TOP TO LOWEST JOB 1728 1729 I USED SINCE SYSTEM REGAN C(MAXSWP)=MAX, NO. OF K EVER ASSIGNED COUNTING FROM TOP DOWN TO SLOWEST FXCURSION, INCLUDING HOLES, WHEN = C(K4SWAP), THEN FRAGMENTATION HAD 1730 1731 1732 HAPPENED SCHETIME DURING THE PAST SHIFT FACTOR TO CONVERT K OF CORE TO DISK BLOCKS, I.E., 1733 002023 BLKSPK=3 SHIFT FACTOR TO CONVERT K OF CORE TO DISK BLOCKS, I.E., ; EIGHT DISK BLOCKS PER K OF CORE, ;ONLY FOR CONSISTENCY WITH OLDER PDP-6 ROUTINE, ;MAXIMUM NUMRER OF IK DISK BLOCKS WHICH MIGHT BE ALLOCATED ; FOR SWAPPING (UPPER BOUND ON THE VALUE OF K4SWAP WHICH ; MAY BE REQUESTED AT DISK REFRESH TIME). (ONE MILLION WORDS ; FOR SWAPPING SEEMS LIKE A NON-RESTRICTIVE ARBITRARY LIMIT.) >07441 1734 002003 CONVMD=BLKSPK 1736 001750 MXK2SWP=+01220 1737 1738 1739 1740 000034 SWPS12=MXK2SWP/*D36+1 SIZE OF SWPTAB ALLOCATION TABLE. 1741 -SWPSIZ,SWPTAB ;THREE WORD POINTER TABLE 403000,0 ; recuired by the "getrit" and -SWPSIZ,SWPTAB ; "CLRBIT" SUBROUTINES, 1742 001304' SWPENT: XWD 201321 777744 1743 001302 400020 0000000 XWD 1744 001303 777744 001304' XWD 1745 1746 1747 001304 SWPTAB: BLOCK SWPSIZ ; SWAPPING SPACE AVAILABILITY TABLE. 1748 1749 DATA CARRIED OVER FROM THE COMMON PART OF OLD AND NEW SWPSER ROUTINES ---1750 INTERNAL SURED, SERA, SERACT, ESORED 1751 001340 000210 000000 SORESI C(LH)=NEG. OF SIZE OF READ/WRITE 1752 I C(PH)=LOC. OF FIRST WORD TO READ/WRITE ICOPY OF SQREQ IN CASE OF SWAP ERROR ON FRAGMENTED JOB 1753 ESOREQ: 7 1754 701341 0022"0 653626 1755 201342 222222 000002 SEPA: ISIGN IS 1 IF A READ 7 ; C(RH)=RLOCK NUMBER BEFORE READING, ; ERROR BITS AFTER READING, ;COUNTER FOR ERRORS 1756 1758 001343 000010 000000 SERACT: 7 1759 AMOUNT TRANSFERRED SO FAR FOR FRAGMENTED JOB 001344 200020 000000 SULENI 2 1760 1761 > JEND OF SWPSER LOGIC FOR THE NEW PDP-10 DISK.

SCHEOB - SCHEDULING ALGORITHM Swpser R.KRAfin/Af 194	FOR SWAPPING SYSTEV(1275*)(BHRPOUGHS DISK) ,34 - 03 FER 69 - V476	MACRC.V36 19:10	4-JUN-60 PAGE 63
1762 201345	SCHEND: FND		
NO ERRORS PETECTED			

PROGRAM PREAK IS 661345

AC1 020215 INT AC2 200016 INT AC3 200017 INT AVPEV 000666' EXT AUAL 200250' INT AJD 200017 INT AVPEV 000371' INT AUAL 200250' INT AJD 200011 INT AVEC0 000371' INT AUAL 200250' INT AJD 200011 INT AVEC0 000371' INT PIGH0L 900555' EXT BLSSPK 000073 INT BMPAC3 000760' BGLINK 000760' CK 001164' FXT CHG1 000760' CHS37' CHK3P 000666' CKJB3 000051' CKJB10 00012'2' CHK3P 000066' CKJB4 000065' CKJB4 000051' CKJB4 000666' CKJB4 200074' CKJB6 000051' CKJB6 000051' CKJB4 000663 INT CKJB6 200011' CKJB6 000051' CKJB6 000051' CKJB4 000127' CKJB8	E 64
ANDEV 000666' EXT AUAVAL 000250' INT AUA AUAVAL 000250' INT AUA AUAVAL 000261' INT AUAVAL 000260' INT CHECK 001164' FXT CHECK 001164' INT CHESS 00066' CKJB1 000061' CKJB1 000061' CKJB1 000061' CKJB1 000061' CKJB1 000065' CKJB4 000065' CKJB4 000065' CKJB4 000065' CKJB1 000126' CKJB1 000126' CKJB1 00006'	
AURE0 2003°1' INT AVALTB 200244' INT AVLUM 200011 INT AVL0TB 2003°7' INT PIGH0L 200264' INT BUSST BUSST 20075' BMPAC3 200760' BGFIX 200750' EXT BLKSPK 20072' BQLINK 200760' BGFIX 200750' EXT CHECK 201164' FXT CHSSF 2002371' EXT CHEXPN 20066' CKJB1 200041' CHSSF 200247' CHSSF 200241' CKJB3 200041' CKJB1 2000412' CKJB2 20024' CKJB3 200051' CKJB4 200066' CKJB4 200267' CKJB4 200065' CKJB5 200273' CKJB6 202074' CKJB4 202000' CKJB7 200112' CKJB8 20011' CKJB6 20200' CKJB7 200137' CKS6 202074' CKJB8 202000' CKJB7 200137' CKJB8 202000' INT 20202' CKJB7 200137' CKJB8 202000' INT 20202' CMB8 <td< td=""><td></td></td<>	
AVLOTB 90037' INT PIGHOL 90065' EXT BLKSPK 90007 INT BMPAC3 000760' PGFIX 900255' BQUS1Z 00017' BMLINK 900164' CEPROR 900750' EXT CHECK 001164' FXT CHG1 901035' CHG3 201045' CHGSWP 900051' CHSF 90077' CHSWP 900066' CKJB1 900051' CKJB4 900066' CKJB4 900065' CKJB4 900065' CKJB5 90073' CKJB4 900065' CKJB4 900065' CKJB5 90073' CKJB4 900065' CKJB4 900065' CKJB5 90073' CKJB4 900065' CKJB4 900065' CKJB5 900112' CKJB8 200111' CKJB4 900065' CKJB5 900137' CKSMCT 70174' EXT CLCC01 901242' FXT CLRB1T 90127' CKJB5 900006'	
BMPAC3 000760' RGF1X 00025' RGJSIZ 000172' BQLINK 000164' CEPRIR 000750' EXT CHECK 001164' FXT BQLINK 000172' EXT CHECK 001164' FXT CHECK 001164' FXT CHG1 00105' CHG3 201045' CHGSWP 00116' FXT CHKSWF 000571' EXT CHKXPN 000606' CKJB1 000041' CKJB10 00073' CKJB2 20074' CKJB4 00065' CKJB5 000073' CKJB6 200074' CKJB6A 000124' FXT CLRB1 00127' EXT CHKJB8 200111' CKJB6A 000057' CKJB7 000127' CKSB6 200074' CKJB6A 000057' CKJB7 000127' CKJB8 200111' CKJB6 200000' TNT CLRB1T 001277' EXT CMG4' FXT CLRB1' 000005' CKJB6	
BQLINK ØØP164' CEPROR ØØP760' EXT CHECK ØØ164' FXT CHG1 Ø01035' CHG3 201045' CHECK Ø0164' FXT CHSUF ØVP73' EXT CHSUP Ø0164' FXT CKJB10 ØVP132' CKJB2 200066' CKJB3 ØØØ05' CKJB4 ØØP073' CKJB4 ØØP073' CKJB4 ØØP073' CKJB4 ØØP073' CKJB7 ØØP073' CKJB6 200074' CKJB9 ØØP075' CKJB7 ØØP073' CKJB6 200074' CKJB9 ØØP075' CKJB7 ØØP073' CKJB6 20011' CKJB9 ØØP075' CKJB7 ØØP073' CKSMCT 701174' EXT CLCOR1 ØØ1242' FXT CLRB1T ØØP073' EXT CMG6ET 701017' EXT CLCOR1 ØØ1242' FXT CLRB1T ØØP073' EXT CMG6ET 701017' EXT CLCOR1 ØØ0257' INT CDVPJ Ø01073' EXT DAVAL 700252' INT DAQ ØØ0006' INT <t< td=""><td></td></t<>	
CHG1 001035' CHG3 201045' CHGS 001016' INT CHKSWF 000771' EXT CHKXPN 000606' CKJB1 000041' CKJB10 00013' CKJB2 200044' CKJB3 00005' CKJB4 000066' CKJB4 000073' CKJB4 00005' CKJB5 00073' CKJ66 200074' CKJB9 00005' CKJB7 00012' CKJB6 200074' CKJB9 00005' CKJB7 00012' CKJB6 200074' CKJB9 00005' CKJB7 000137' CKSMCT 201174' EXT CLCOR1 001242' FXT CLRB1T 001277' EXT CMG 700174' EXT CLCOR1 00005' INT CDNVMD 000033' INT CDRGET 701071' EXT CORTAL 000254' FXT CDAVAL 700252' INT DAQ 000065' INT 024VAL 000254' INT DAREQ 000303' INT CDREQ 700375' INT </td <td></td>	
CHKSUF 000571'EXT CHKXPN 000606' CKJB1 00041' CKJB10 000132' CKJB2 00044' CKJB3 00065' CKJB4 00066' CKJB4A 00067' CKJB4B 00005' CKJB5 000073' CKJB6 200074' CKJB6A 00005' CKJB7 000112' CKJB8 200111' CKJB9 000057' CLRBT 000137' CKSMCT 201174'EXT CLCOR1 001242'FXT CLNBT 000137' CKSMCT 2010''EXT CMWB 200000 INT CONVMD 200003 INT CORGET 7012''EXT CMWB 200006 INT DAREQ 000303' INT DAT 0000''S INT DCAVAL 000254'INT DCG 000001 INT DCREO 7003''S INT DCAVAL 00026'' INT DFRUSY 001072'EXT DINT8A 031170' DINT8B 001146'EXT DIAVAL 7002'S'INT DEVDAT 000300' INT DCGW000 00030' INT 00000''''''''''''''''''''''''''''''''	
CKJB10 002132' CKJB2 200044' CKJB3 000051' CKJB4 000066' CKJB4 000067' CKJB4 000051' CKJB5 000073' CKJB6 200074' CKJB6 000051' CKJB7 000112' CKJB8 200111' CKJB9 000057' CKJB7 000137' CKSMCT 201174' EXT CLCR1 001242' FXT CLRBIT 001277' EXT CM0 90022 CMWB 200000 INT CONVMD 0000303' INT CORECT 70121' EXT CORTAL 000224' FXT CPOPJ 001073' EXT DAAVAL 700252' INT DAQ 000006' INT DAREQ 0000303' INT DAT 00007' DINT DCAVAL 000254' INT DFUSY 001072' EXT DINT8A 001170' DINT8B 001164' FXT DIAVAL 000253' INT DIA 000007 INT DEVDAT 000006 INT DEVDAT 0000253' <	
CKJB4 000066' CKJB4A 00027' CKJB4A 00027' CKJB5 000073' CKJB6 00074' CKJB6A 00005' CKJB7 00012' CKJB8 200111' CKJB9 00005' CKJB7 00012' CKJB8 200111' CKJB9 00005' CKJB1 000127' EXT CMG0 00022' CMWB 200000 INT CLRBIT 001277' EXT CMG0 00022' CMWB 200000 INT CONVMD 0002033 INT CDRGET 0102' INT DAQU22' CMWB 200000 INT CPOPJ 001073' EXT DAAVAL 20022' INT DAQ 000061 NT DAREQ 000173' EXT DAAVAL 20022' INT DAQ 000061 NT DCG 000173' EXT DAAVAL 20022' INT DCAVAL 00024' FXT DAREQ 000172' EXT DAAVAL 20025' INT DCAVAL 00024' INT DFWSY 000	
CKJB5 000073' CKJE6 20074' CKJE6 20010' CKJB7 000112' CKJE6 20074' CKJE6 200074' CKJB7 000112' CKJE6 200074' CKJE6 200074' CKJB7 000137' CKJE6 20074' CKJE6 200074' CKJB7 000137' CKJE6 20074' CKJE6 20000' CLRB1T 00127' EXT CMO 20000' 001242' FXT CDNVMD 000073 INT CDRGET 0010'' EXT CONTAL 000224' FXT CDAREQ 000030' INT CDRGET 0000'' INT DCAVAL 00024'' FXT DAREQ 000030' INT DAT 0000'' INT DCAVAL 00024'' INT DGG 000010 INT DCREQ 0000'' INT DCAVAL 00024'' INT DGG 000010 INT DCREQ 00014'' EXT 00000'' INT DFWRT 001146'' FXT DINT4B 0	
CKJB7 000112' CKJB8 200111' CKJB9 000057' CKJBT 000137' CKSMCT 201174' EXT CLCOR1 001242' FXT CLRBIT 001277' EXT CMQ 20022 CMWB 200000 INT CONVMD 000057' EXT CMQ 20022 CMWB 200000 INT CONVMD 000073' EXT CAAVAL 20022' INT CORTAL 000254' FXT DAREQ 0000303' INT DAT 000005 INT DCAVAL 000254' INT DAREQ 000303' INT DAT 000005 INT DCAVAL 000254' INT DAREQ 000172' EXT DAAVAL 20000' SINT DCAVAL 000254' INT DGG 000303' INT DAT 00000' SINT DCAVAL 000254' INT DAT 00000' SINT DCAVAL 000254' INT DCAVAL 000254' INT DFUSY 001072' EXT DFREO 001145' EXT DFWRT 00116' FXT DINT4B 00172' EXT DINT8A 00117' DINT8B 00157' DIAVAL 000253' INT DIQ 00000' TINT DTREQ </td <td></td>	
CKJBT 000137' CKSMCT 201174' EXT CLC0R1 001242' FXT CLRBIT 001277' EXT CMO 202022 CMWB 200000 TNT CONVMD 002023 INT CDRGET 70174' EXT CORTAL 00224' FXT CPOPJ 001073' EXT DAAVAL 20022' INT DAQ 000006 TNT DAREQ 000303' INT DAT 000252' INT DAQ 000006 TNT DAREQ 000303' INT DAT 000075 INT DCAVAL 000254' INT DGG 000010 INT DCRED 001145' EXT DFWRT 00146' FXT DFUSY 001072' EXT DFRED 001145' EXT DFWRT 00146' FXT DINT4B 00172' EXT DINT8A 001170' DINT8B 001157' DTAVAL 000253' INT DIQ 000007 INT DTREQ 000304' INT EQFIX 400000 00225' EQJSIZ 400000 000172' EQLINK 400000 000164' ESORFQ 001341' INT EXCALP 00060' EXT FININ 000530' FXT FININ 000530' FXT	
CLRBIT Ø01277'EXT CMQ Ø0022 CMWB 200001 INT CONVMD Ø00033 INT CORGET °010°1'EXT CORTAL Ø00624'FXT CPOPJ Ø01073'EXT DAAVAL ?00252'INT DAQ @00006 INT DAREQ Ø00204'INT DAT Ø00252'INT DAQ @00026'INT DAREQ Ø00210'INT DAT Ø00025'INT DCAVAL @00254'INT DCQ Ø00210'INT DCRC ?0033°5'INT DCVAL @00266'INT DFUSY Ø01072'EXT DINTBA Ø01145'EXT DFWRT ?01146'FXT DINT4B Ø0172'EXT DINTBA Ø01170' DINTBB ?01157' DTAVAL ?000253'INT DTA Ø00070'INT DTREQ Ø00304'INT EQFIX 400000 Ø0255' EQJSIZ 400000'PO172' EQLINK 400000'PO164' ESQRFQ Ø0134'INT EXCALP Ø0000'EXT FININ Ø00530'FXT FININ FININC Ø00520' FINISH Ø0176'EXT FININ Ø00530'FXT Ø00530'FXT FININC Ø00530'	
CONVED 000003 INT CORGET 000021 EXT CORGET 000022 INT DAQ 000024 FXT DAREQ 0000010 INT DAT 0000025 INT DCAVAL 000024 000006 INT DCG 0000010 INT DCRED 0001145 EXT DFWRT 000006 INT DFUSY 001072 EXT DFRED 001145 EXT DFWRT 001146 FXT DIAVAL 001253 INT DTA 000007 INT DTREQ 000000 000000 INT EQFIX 400000 0002255' EQJSIZ 400000 0002000 EXT ERP	
CPOPJ Ø01073'EXT DAAVAL 200252'INT DAQ 000006 INT DAREQ Ø00303'INT DAT 000005 INT DCAVAL 000254'INT DAREQ Ø00303'INT DAT 000005 INT DCAVAL 000254'INT DGG Ø0010'INT DAT 000005 INT DCAVAL 000254'INT DFUSY Ø01072'EXT DFRED Ø01145'EXT DFWRT Ø01164'FXT DINT4B Ø0172'EXT DINTBA Ø01170' DINT8B Ø0157' DTAVAL Ø00253'INT DTQ Ø00007 INT DTREQ Ø00304'INT EQFIX 400000 Ø0225' EQJSIZ 400000 Ø00172' EQLINK 400000 Ø00164' ERATRY Ø0000'' EROR Ø0020'EXT ERRPNT Ø00530' FXT FININ0 Ø00520' FINISH Ø01176' EXT FININ Ø00544' FXT FININ0 Ø00537' FIT Ø00773' EXT FITØ Ø00551'	
DAREQ ØØØ303' INT DAT ØØØ205 INT DCAVAL ØØØ254' INT DGG ØØØ302' INT DCREO ØØØ325' INT DEVDAT ØØØ264' INT DFUSY ØØ1072' EXT DFREO ØØ145' EXT DFREO ØØ145' EXT DFWRT ØØ146' EXT DINT4B ØØ172' EXT DINT8A ØØ1170' DINT8B ØØ1157' DTAVAL ØØ0253' INT DTA ØØØ267 INT DTREQ ØØØ304' INT EQFIX 4ØØ000 ØØ253' INT DTA ØØØ262'S' EQJSIZ 4ØØ000 ØØ172' EQLINK 4ØØ000 ØØ0164' ERATRY ØØ00'33 ERFOR ØØ00'0' EXT ERRPNT ØØ0530' FXT FININØ ØØ0520' FINISH Ø0176' EXT FININ ØØ0530' FXT FININØ ØØ0520' FINISH Ø0176' EXT FININ ØØ0544' FXT FININØ ØØ0520' FINISH Ø0176' EXT FININ ØØ0544' FXT FINOIT ØØ0533' FIT ØØ0773' EXT FITØ ØØ0551'	
DCG 00010 INT DCRED 00035 INT DEVDAT 000206 INT DFRUSY 001072' EXT DFRED 001145' EXT DFWRT 001146' FXT DINT4B 00172' EXT DINT8A 021170' DINT8B 021157' DTAVAL 000253' INT DTA 00007 INT DTREQ 000304' INT EQFIX 400000 00225' EQJSIZ 400000 000172' EQLINK 400000 000304' INT EQFIX 400000 00253' INT EQJSIZ 400000 000164' EQJSIZ EQUSIZ' EXT ERPNT 000531' FXT ESQRFQ 00134' INT EXCALP 00000' EXT FININ 000530' FXT FINING 000520' FINISH 00176' EXT FINOT 000544' FXT FINOT 000537' FIT 000773' EXT FIT0 000551' <	
DFPUSY ØØ1072'EXT DFRED ØØ1145'EXT DFRED ØØ1145'EXT DFWT ØØ1146'FXT DINT4B ØØ1172'EXT DINT8A ØØ1170' DINT8B ØØ1157' DTAVAL ØØ0253'INT DIN ØØ0007 INT DTREQ ØØ0304'TNT EQFIX 40000 Ø0225' EQJSIZ 40000 ØØ172' EQLINK 40000 ØØ164' ERATRY ØØ0235' EROR ØØ0620'EXT ERPNT ØØ050'FXT ESOREQ Ø01341'INT EXCALP ØØ0176'EXT FININ ØØ0500'FXT FININ0 ØØ6520' FINISH Ø01176'EXT FININ ØØ0544'FXT FINOIT ØØ0537' FIT ØØ773'EXT FITØ ØØ0551'	
DINT4B Ø01172'EXT DINT8A Ø01170' DINT8B Ø01157' DTAVAL Ø00253'INT DIN DIN Ø00007 INT DTREQ Ø00304'TNT EQFIX 400000 Ø00255' EQJSIZ 400000 Ø00275' EQLINK 400000 Ø00164' ERATRY Ø00003 EROR Ø00620'EXT ERPNT Ø00500'FXT ESOREQ Ø01341'INT EXCALP Ø00176'EXT FININ Ø00500'FXT FININ0 Ø06520' FINISH Ø01176'EXT FINOT Ø00544'FXT FINOIT Ø00537' FIT Ø00773'EXT FITØ Ø00551'	
DTAVAL 000253' INT DTA 000007 INT DTRG 000304' INT EQFIX 400000 000205' EQJSIZ 400000 000172' EQLINK 400000 000164' ERATRY 00000'3 ERCQ 000620' EXT ERRPNT 000530' FXT ESORFQ 001341' INT EXCALP 00000' EXT FININ 000530' FXT FININ0 000520' FNISH 00176' EXT FINOT 000544' FXT FINOIT 000537' FIT 000773' EXT FIT0 000551'	
EQFIX 400000 000205' EQJSIZ 400000 000172' EQLINK 400000 000164' ERATRY 00000'3 ERDR 000602' EXT ERPNT 000531' FXT ESORFO 001341' INT EXCALP 00010' EXT FININ 000500' FXT FINING 000520' FINISH 001170' EXT FINIT 000544' FXT FINING 000520' FIT 000773' EXT FITO 000551'	
ERATRY ØØØ02/3 ERROR ØØØ620' EXT ERRONT ØØØ531' FXT ESQREQ Ø01341' INT EXCALP ØØØ020' EXT FININ ØØ0500' FXT FININ0 ØØØ520' FINISH Ø01176' EXT FINOT ØØ0544' FXT FINOUT ØØ0537' FIT ØØ0773' EXT FITØ ØØ0551'	
ESGREQ 001341'INT EXCALP 00000 EXT FININ 000500'FXT FININO 000520' FINISH 001176'EXT FININ 000544'FXT FINOUT 000537' FIT 000773'EXT FITO 000551'	
FINING 000520' FINISH 001176'EXT FINOT 000544'FXT FINOUT 000537' FIT 000773'EXT FIT0 000551'	
FINOUT 000537' FIT 000773' EXT FIT0 000551'	
FODDE AND FILL FILLE WOUDDS' FX	
COCOT PRAGIU DOLLO	
FDCOVO COLORAD PROBAL PROBAL	
FRANCE PRAIRIE	
FTCUAD 777777 777777 INT FTRUHK ////// /////	
FULCOD FULCAT MOUTO FX	
EVENTO READER DUIZA	
CYCAT GETFUR W00/20 FX	
TNCOUT GG10471 EVT INTO INTO INTO INCOM	
100ERR 200030 LNT 100150 100020 NT	
TOP TOTAL TOTAL TOTAL TOTAL TOTAL	
1000 11 1000 1001203' FX1	
IDTSUD 0014771 EVT 00101 EXT 001003 FX1	
108MAY 3014541 EVT 100000 0000241 EXT 3080PG 0000241 EXT	
IBO GRADO THE JOB GRADO THE JOB GRADE DROSEL INT	
JKU 002002 INT JXPN 000001 INT K4SWAP 001274' FXT	

	CHEDULING ALGORITHM FOR MBOL TABLE	SWAPPING SYSTEM(1	a/57)(BURPOUGHS DISK)	MACRO.V36 1	9:17 4-JUN-69 PAGE 64-1
KCORF1	200543' EXT	LBHIGH	201264' FXT	LOC	800022
MAXJEN	000654' EXT	MAXQ	000011 INT	MAXSIZ	000647' =xT
MAXSWP	201320' INT	MINQ	200003 INT	MJOBCK	701163' FXT
MJOBN	200012' EXT	MGAVAL	200251 ' INT	MQQ	000005 INT
MQRED	0003-2' INT	MTAVAL	000255' INT	мТQ	000011 INT
MTREQ	200326' INT	MXCODE	000016 INT	MXK2SW	001750 INT
MXQUE	000142' EXT	NUFIT	000714	NQUEUE	000012
NSWP	010070 INT	NULQ	000015 INT	NXTINI	200000 TNT
NXTJ91	200022'	NIX TJOB	200010' INT	OERROR	201002' FXT
OSCAN	000436' INT	OUTP2	200711'	PC	200001
PCORSZ	000510' EXT	PCSTOP	000536' EXT	PDP	200003 INT
PJBSTS	000223' EXT	POTLST	200126' FXT	P01	200017
PQ2	ADADER	PQ3	000021	PROG	200027 INT
PROT	000511' EXT	PROTØ	000512' FXT		
QAUS	0004/01	DAUW .	000344	Q Q	200007
QBAK1				OBAK	000241
QDAS	@ØØ235' ØØØ366'	OBITS Odaw	000256' INT 000350'	Q C M W	000330' INT
QDCW	000352			ODCS	000370
		QUIS	000374	QDTW	000356'
QFIX	000225	QFOR	000232	QFOR1	000231'
QFOR2	000234	QINI	000142' INT	QINI1	000155
QIOWW	000340	01	000006	QJOB	000322' INT
QUSIZ	000172	OLINK	000164	QMQS	000364'
QMQW	000346	OMTS	000376'	QMTW	000360
QNULW	000324	00SD	000006	QOSTAB	000414'
QQTTY	PDP026	OK	000002	QRNS	000000
ORNW	000332'	OSCAN	000225' INT	QSLPW	000362'
OSTAR	0004V4	OSTOP	000326' INT	QSTOPW	000326'
QSTS	000372'	nstw	0003541	QTIME	000402' INT
WWOITQ	000342	QTSS	ØØØØØØ	QTSW	000336'
QTTAP	000410	OWSS	000000	QWSW	000334'
QX1	000175'	0X2	000201'	QX3	000224'
QXFER	000162' INT	RCXSKD	000000' INT	REQTAB	000275' iNŤ
RNAVAL	000244' INT	RNQ	000000 INT	RNQUNT	000414' INT
RNRED	000275' INT	RUNARL	440000 INT	RUNMSK	111404 INT
SCHD1	000130'	SCHED	000113'	SCHEND	01345'
SCNUOB	000624	SCNOK	000621'	SCNUUT	000610'
SERA	001342' INT	SERACT	001343' INT	SHF	004000 INT
SHFWAT	000572' EXT	SLPQ	000014 INT	SQGO	001074' INT
SQC01	001107' INT	SQGOZ	001142'	SQIN	001065' INT
SQLEN	221344'	SQOUT	001066' INT	SQREQ	001340' INT
SSCAN	000430' INT	۲P	000005	STAVAL	000247' INT
STOPQ	000016 INT	STO	00003 INT	STREQ	000300 INT
SUMCOR	000652' EXT	SW	000002	SWAP	000465' INT
SWAPI	AQ0772 '	SWAPO	0006701	SWP	002000 INT
SWP1	0005461	SWP2	000547'	SWPENT	001301'
SWPER1	001173'	SWPERC	000604' EXT	SWPERR	001176'
SWPHCH	001265' EXT	SWPINI	001264' INT	SWPINT	001147' INT
SWPREC	000575'	SWPS1Z	000034	SWPTAB	001304' INT
т	000026	τ1	000002	T2	000001
TAC	000001 INT	TAC1	000002 INT	TIMEF	000031' FXT
TIOWO	600013 INT	TRYSWP	000655' EXT	TSAVAL	000246' INT
TSO	000002 INT	TSREQ	000277' INT	TT	000005
USRMOD	010030 INT	VIRTAL	001275' EXT	VSCHED	000421 INT
	ar an in an in an Ar 1911	TINING.	1046/2 [A!	VJUHLU	UDDACT INC

SCHERB - SC Sym	"HEDULING ALGORITHM FOR 'BOL TABLE	SWAMPING SYSTEM(1	0/50)(BURROUGHS DISK)	MACRO,V36 1	9:10 4-JUN-69 PAGE 64-2
WSAVAL	000245' INT	WSC	000071 INT	WSREQ	000276' INT
WTMASK	000370 INT	Xokcsw	200020' INT	XJOB	000323' INT
XPAND	001052' INT	Xpannh	201355' INT	Zerswp	001034' INT

A	6#	6												
A C 1.	6#	6	955	970	1250	1624	1679	1680	1652	1721				
AC2	6#	6	1625	1683	1722									
AC3	6#	1147	1149	1150	1151	1157	1159	1184	1194	1196	1197	1199	1200	12 0 3
	1274	1655	165c	1657	1050	1661	1663	1669						
AEFERR	6#	6												
AL	6#	6												
ANYDEV	837	113												
ASSCON	6#	6												
ASSPRG	6#	6												
AUAVAL	490	490#												
AUQ	490	490#	624	678	763	782	791							
AUREO	574#	575												
AVALTB	54	135	146	483	485#	496	4,97	488	489	490	491	492	493	494
	495	496	6.0.0											
AVLNUM AVLQTB	134	572#	502	(
B	138 6#	145 6	598	600#										
BIGHOL	837	97Ø											1	
BLKSPK	1427	97Ø 1631	1634	1675	1679	1720	1733#	1735						
BMPAC3	1162	1183	1194#	10/2	10/9	1/20	1/03#	1/35						
BQFIX	395#	632#	663	668	706	709	712	715	718	721	724			
BQJSIZ	399#	636#	000	000	7:0		/ 12	115	/10	12.1	124			
BQLINK	397#	634#												
BUFPNT	6#	6												
BUFWRD	6#	6												
CERROR	1176	1180												
CHECK	1357	1390	1449											
CHG1	1251	1280#												
CHG3	1288	1291#												
CHGSWP	1247	1250#												
CHKSHF	840	854	983											
CHKXPN	947	1021#												
CKJB1	86	93	97	121#										
CKJB10	120	183#												
CKJB2	104#	132												
CKJB3	91	111#												
CKJB4	117	126#	_											
CKJB4A	116	127#	187											
CKJB4B	125#	185												
CKJB5	102	127	134#											
CKJB6	135#	136	150											
CKJB6A	139#	143												
CKJB7 CKJB8	137 140	151# 150#												
CK1B9	114	117#												
CKJBT	186	188#												
CKSMCT	1432	1458												
CLCOR1	1400	1667												
CLKR	6#	6												
CLRBIT	1604	1684	1709											
CLSIN	6#	6												
CLSOUT	6#	6												
		-												

CMO	520#	655	760											
CMWB	6#	6	89	113										
CONVMD	1720	1735#												
CORCNT	6#	6												
CORGET	8.39	1222												
CORTAL	837	1027												
CPOPJ	230	248	1194	1204	1357	1373								
D	6#	6												
DAAVAL	492	492#												
DAO	492	492#	626	684	762	783	792							
DAREQ	576#	577			,		1.12							
DAT	6#	6	78	79	98	112	125	126	141	142	145	162	186	340
	364	435	945	1017	1030	1125	1126	1128	1147	1149	1154	1158	1159	
	1163	1165	1168	1169	1180	1223	1286	1287	1288	1291	1402	1403	1404	1160 1405
	1407	1408	1409	1413	1414	1415	1416	1417	1418	1421	1422	1403		
	1657	1658	1674	1681	1414	1415	1410	1411	1410	1421	1422	1425	1425	1623
DCAVAL	494	494#	10/1	1001										
DCL	6#	6												
DCLI	6#	6												
DCLO	6#	6												
DCLR	6#	6												
000	494	494#	608	687	764	781	79Ø							
DCREO	578#	579	010	0.077	704	101	790							
DDI	6#	6												
000	6#	6												
DEN	6#	6												
DEVADR	6#	6												
DEVBUF	6#	6												
DEVCHR	6#	6												
DEVCTR	6#	6												
DEVDAT	6#	6	43	846										
DEVEXT	6#	6	40	040										
DEVFIL	6#	6												
DEVIAD	6#	6												
DEVIOS	6#	6												
DEVLOG	6#	6												
DEVMOD	6#	6												
DEVNAM	6#	6												
DEVOAD	6#	6												
DEVPPN	6#	6												
DEVPTR	6#	6												
DEVSER	6#	6												
DFBUSY	1357	1372												
DFRED	1357	1429												
DEWRT	1357	1430												
DGF	6#	6												
DHNG	6#	6												
DIN	6#	ő												
DINI	6#	6												
DINT4B	1432	1455												
DINT8A	1445	1453#	1467											
DINT8B	1439	1444#												
DLK	6#	6												

•

1

DMT DNAERR DOU DR DRL DRN DSER DSER DSER DSKRLB	6# 6# 6# 6# 6# 6#	6 6 6 6 6 6 6 6												
DSO DTAVAL	6# 493	6 493#												
DTO	493	493#	607	693	767	778	787							
DTREQ	577#	578	01.7	0,0	, . ,	///0	707							
DVAVAL	6#	6												
DVCDR	6#	6												
DVDIR	6#	6												
DVDIRI	6#	6												
DVDIS	6#	6												
DVDSK	6#	6												
	6#	6												
DVIN	6#	6												
DVLNG	6#	6												
DVLPT DVMTA	6# 6#	6 6												
DVOUT	0# 6#	6												
DVPTP	6#	6												
DVPTR	6#	6												
DVTTY	6#	6												
ENTRS	6#	6												
EQFIX	188 695	396# 698	633#	648	651	654	672	675	677	68Ø	683	686	689	692
EQJSIZ	400#	637#	657											
EQLINK	398#	635#	727											
ERATRY	1376#	1382												
ERROR	346	364	1006	1017				-						
ERRENT	841	915											1	
ESGREQ	1370	1463	1750	1753#										
EXCALP	841													
FBMERR	6#	6												
FININ FININØ	867	868 122Ø	1000											
FINISH	865# 838	122Ø 860	1229 911	939	1132	1214	1391	145Ø	1460					
FINOT	933	934	711	7 37	1102	1217	1391	1450	1400					
FINOUT	862	918#												
FIT	838	942	954	1033	1138	1215								
FITØ	942#	985		120-										
FIT1	869	943	954#											
FITHGH	1225	1226												
FITSIZ	961	962												
FNFERR	6#	6												
FORCE	838	940	1092	1113										
FORCEØ	1021	1063#												
FORCE1	941	1094#												

DMT DNAERR	6# 6#	6 6												
DOU	6#	6												
000 0R	6#	6												
DRL	6#	6												
DRN	6#	6												
DSER	6#	6												
DSI	0# 6#	6												
DSKRLB	6#	6												
DSO	6#	6												
DIAVAL	493	493#	697	693	767	778	787							
DTO DTREQ	493 577#	493#	607	090	101	//0	/0/							
DVAVAL	577#	578												
DVCDR	0# 6#	6 6												
DVDIR	6#	6												
DVDIGI	6#	6												
DVDIS	6#	6												
DVDSK	6#	6												
DVDJA	6#	6												
DVIN	6#	6												
DVLNG	6#	6												
DVLPT	6#	6												
DVMTA	6#	6												
DVOUT	6#	6												
DVPTP	6#	6												
DVPTR	6#	6												
DVTTY	6#	6											-	
ENTRS	6#	6												
EQFIX	188	396#	633#	648	651	654	672	675	677	68Ø	683	686	689	692
7. GF 1 A	695	590# 698	030#	040	001		0/2	0/5	0//	000	083	000	. 009	092
EQUSIZ	400#	637#	657											
EQLINK	400# 398#	635#	727											
ERATRY	1376#	1382	1.1											
ERROR	346	364	1006	1017										
ERRPNT	841	915	1000	1011										
ESOREO	1370	1463	1750	1753#										
EXCALP	841	1.400	17,56	1/20#										
FBMERR	6#	6												
FININ	867	868												
FININØ	865#	1220	1229											
FINISH	838	860	911	939	1132	1214	1391	145Ø	1460					
FINOT	933	934	/ 1 4	207	TIOF		1011	1420	1400					
FINOUT		918#												
	862													
FIT	862 838		954	1033	1138	1215								
FIT FITØ	838	942	954	1033	1138	1215								
FITØ	838 942#	942 985		1033	1138	1215								
FITØ FIT1	838 942# 869	942 985 943	954 954#	1033	1138	1215								
FITØ FIT1 FITHGH	838 942# 869 1225	942 985 943 1226		1033	1138	1215								
FITØ FIT1 FITHGH FITSIZ	838 942# 869 1225 961	942 985 943 1226 962		1033	1138	1215								
FITØ FIT1 FITHGH FITSIZ FNFERR	838 942# 869 1225 961 6#	942 985 943 1226 962 6	954#		1138	1215								
FITØ FIT1 FITHGH FITSIZ FNFERR FORCE	838 942# 869 1225 961 6# 838	942 985 943 1226 962 6 940		1033 1113	1138	1215								
FITØ FIT1 FITHGH FITSIZ FNFERR	838 942# 869 1225 961 6#	942 985 943 1226 962 6	954#		1138	1215								

FORCE2 FORCEL FORHGH FORSIZ FRAGEK FRAGIO FRECR1 FRESWP FRGBK1 FRGBK2	1056 935 1051 1655# 1402# 976 1171 1656# 1658	1259# 1292# 1270 1252 1670 1442 979 1172 1640 1661#									·			
FRGI01	1425#	1411												
FRGI02	1427	1413#												
FRGSEG FT2REL	6# 6#	6	1152	1397	1437	1653						_		
FIZRE	1350	844	866	932	957	96Ø	975	1050	1068	1Ø87	1170	1225	1338	1347
FTATTA	6#													
FICCL	10#													
FTCHEC	6#	457	458	470	549	556	610	616	1710	1717				
FTDISK	10#	11	14	200	490	538	574	604						
FTEXAM	6#													
FTFIN1 FTGETT	6# 6#													
FTHALT	6#													
FTKCT	6#													
FTLOGI	10#													
FTMONP	6#	457	458	470	549	556	610	616	1710	1717				
FTPRV	6#						0.0	010	1/10	T , T ,				
FTRA1Ø	6#													
FTRC1Ø	10#	11	14	22	26	31	1299	1305	1468	1600				
FTRCHK	6#	1014												
FTREAS	6#													
FTSLEE FTSWAP	6#	4.0	450			(5 F								
FTTALK	10# 6#	4Ø	152	155	200	625	75 7	1356						
FTTIME	6#									,				
FTTRAC	6#	1107	1110	1210										
FTTRPS	6#													
FTTTYS	6#													
FULCNT	841													
FULCOR	1148	1189#	1195											
FULL FULL1	1130 1154#	1147#												
FULL1A	1155#	1164 1169												
FULL18	1165#	1173												
FULL2	1156	1168#												
FULSET	1161	1183#												
FXSAT	1290	1623	1653#											
FXSAT1	1654	1659	1673#											
FXSATC	1684#													
GETBIT GETFCR	1624	1626												
GXSAT	1145 1129	1190 1155	1633	1622#										
HIGHJB	1006	1010	1020	1026#										
	1000	2010												

HOLEF	976	977												
HSAMSK	¢/0	6												
HSAPOS	6#	6												
HSASIZ	6#	6												
HUNGCT	6#	6												
HUNGST	6#	6												
I	6#	6												
IB	6#	6												
IBUFR	6#	6												
ICLOSB	6#	6												
ICPROT	1379#	1379												
ICPRT1	1328#	1378												
ILM	6#													
ILUERR	6#	6												
IMGIN	199	838	b76	955	1125	1127	1216	1230	1248	1250	1257	1001		
IMGOUT	838	877	991	994	1122	1228	1248	1259	1297	1290	1277	1294	1323	
INBER	6#	6						* 2 2 7	12.57	1295				
INERR	864	911#												
INFLG	183	190#	859	1139										
INITE	6#	6												
INPB	6#	6												
10	6#	6												
IOACT	6#													
IOREG	0# 6#	6 6												
IORKTL	6#	6												
IOBOT	6#	6												
IOCON	6#	6												
TODEND	6#	6												
IODERH	6#	6	1435											
IDDTER	6#	6	1435	1459										
IOENO	6#	6		1 4 5 7										
IOFST	6#	6												
IOIMPM	6#	6	1435											
IONRCK	6#	6	1.02											
IOPAR	6#	6												
105	6#	6	1435	1453	1459									
IOTEND	6#	6			1427									
IOUSE	6#	6												
IOW	6#	6												
IOWC	6#	6												
IOWQ	511	512#	673											
IPOPJ	1191	1248	1268	1335	1351	1695	1627	1686						
IPOPJ1	1605	1635						1000						
IPPERR	6#	6												
ISCAN	758	759#	945											
ITEM	6#	6	45	241	242	243	246	248	249	341	434	849	1153	1105
	1189	1258	1259	1260	1261	1262	1263	1283	1289	1339	1340	1341	1344	1185
	1622	1623	1628	1662	1663	1664	1665	1666	1668	1673	1674	1703	1704	1346
J	45#	76	79	81	82	33	85	88	95	1073	1074	196	104	1
	111	122	132	138	139	140	141	166	173	176	184	341#	360	110 374
	384	385	386	387	390	434#	440	445	440	449	450	454	455	3/4 849#
	860	862	896	897	912	918	922	940	942	948	954	1010	1012	1013
	1015	1020	1033	1035	1236	1044	1058	1062	1088	1092	1098	1101	1114	1118
											-			

- ·

JACCT	1131 6#	1132	1214	1219	1235	1236								
JBFADR	6#	6												
JBFCTR	6#	6 6												
JBFPTR	6#	-												
JBTADR	199	6	077	912		1			4					
JBTCHK	1357	36Ø 1392	837 1451	912	922	1044	1008	1114	1118	1219	1235			
JBTDAT														
JBTQ	839 37	840	897	077	0.4.0	o 47								_
2010		139	230	237	242	247	374	376	377	38Ø	382	384	385	386
JBTQM1	387	445	449	450	454									
	232	246	0.45											
JBTQMN	233	243	245											
JBTQP1	231	240	249											
JBTSTS	36	79	88	95	105	110	111	141	166	173	199	390	836	948
INTOWN	1076	1012	1020	1035	1088	1248	1283	1344	1346					
JBTS⊮P	72 1461	°1	82	122	184	837	896	1036	1131	1153	1185	1236	1248	1289
JBUF	6#	6												
JDAT	6#	6	897	898	971	904	907	908	913	923	1095			
JERR	6#	6									10/0			
JLOG	6#	6												
JNA	6#	6												
JOB	36	85	1121	1341										
JOBDAC	1357	1388	1447											
JOBDPD	839	997												
JOBDPG	839	924	908											
JOBMAX	103	23Ø	241	1340										
JOBPC	839	991												
JOBQUE	61	99	171	619	62Ø#									
JRQ	6#	6	174											
JWPOS	6#	6												
JWSIZ	6#	6												
JXPN	6#	6	115	167	1011	1343								
K4SWAP	1703	1721			_	-								
KCORE1	837	914	926											
LBHIGH	1605	1632	1676	1695										
LISTSN	6													
LOC	496#	498	512	512#	514	515	515#	516	517	517#	518	519	519#	52Ø
LOOKB	6#	6	-					510	211	51/#	210	219	519#	526
MAXJBN	1006	1058	1062											
MAXQ	53	500#	500	502										
MAXSIZ	1006	1029	1055	1057										
MAXSWP	1629	1630	1720	1728#										
MEDDLE	6#	6												
MINQ	147	501#	501											
MJOBCK	1357	1389	1448											
MJOBN	38	76												
MQAVAL	491	491#												
NQQ	491	491#	605	681	761	784	793							
MQREO	575#	576			+	,	, , , ,							
MTAVAL	495	495#												
MTQ	495	495#	629	696	765	78Ø	789							
MTRED	579#	58Ø												

MXCODE	514#	514												
MXK2SW	1720	1736#	1740											
MXQUE	230	235												
NECERR	6#	6												
NLEERR	6#	6												
NOFIT	1032	1138#												
NQUEUE	498#	570												
NSDERR	6#	6												
NSFERR	6#	6												
NSHF	6#	6												
NSRBIT	6#	6												
NSWP	6#	6	1040											
NULQ	239	512#	649											
NXM	6#		•••											
NXTINI	50	53#												
NXTJR1	75	85#												
NXTJOB	40	74#												
OBUER	6#	6												
OCLOSE	6#	6												
DERROR	839	1223												
OSCAN	758	775#	1030											
OUTBEB	6#	6	2000											
		-												
OUTP2	1132#	1186												
OUTPR	6#	6												
PC	436#	439	446	451	455									
PCORSZ	841	892												
PCSTOP	841	917												
PDP	6#	6	62	100	127	148	156	175	177	353	393	854	0.5.0	
	892	914	926	934	962	979	983	986	1002	1052	1063		858	868
	1124	1129	1135	1140	1148	1154	1155	1158	1162			1070	1103	1124
	1189	1190	1195	1198	1222	1226	1258	1286	1290	1168 1291	1172	1178	1179	1183
	1425	1449	1622	1626	1659	1662	1667	1668	1671		1296	1339	1390	1402
PJRSTS	37	38	118	199	392	1002	1007	1000	10/1	1673	1684			
POTLST	72	161	165	174	0.42									
POV	6#	707	100	17.1										
P01	514#	664	669	7Ø7	710	713	716	719	722	725	770	738	7 4 0	7.5
PQ2	189	516#	733	738	739	740	769	794	122	125	732	/38	768	795
PQ3	518#	734	739	740	770	786	107	/ 7 7						
PROG	6#	6	342	898	905	928	912	913	922	923	1005	1000	1010	
PROT	894	1297	0.12	0,0	,0,0	700	712	913	922	923	1095	1098	1219	
PROTO	895	1297												
PRTERR	6#	6												
PVSPYA	6#	6												
PVSPYM	6#	6												
PVTRPS	6#	6												
Q	342#	348	352	356	363	366	368	369	371	380	382	704	7 . 7	7.0.0
	390		0.2	• - •	000	000	000	307	3/1	200	302	384	387	389
QAUS	694	723#												
QAUW	538	677#												
QBAK	454#	786	794	795										
QBAK1	449#	778	779	780	781	782	783	784						
OBITS				100	101	/04	100	/04						
00113	125	5.52	5.14#											
DCMW	125 112	532 630	534# 653#											

QDAS	676	728#												
NDAW	540	6.2#												
ODCS	608	711#												
QDCW	542	685#												
ODIS	607	717#												
QDTW														
OFIX	541	691#	2.0.4	(= 2										
	374#	395	396	632	633									
OFOR	445#	760	768	759	770	776	777	785						
QFOR1	443#	761	762	765	764	765	766	767	787	788	789	79Ø	791	792
	793													
QFOR2	443	447#												
QINI	229	235#												
QINI1	246#	250												
QIOWW	544	671#												
01	43#	87	101	128	131	134	135	136	137	138	145	146	147	150
	166	167	172	173							-			100
0 J O B	59	121	131	622	623#									
QJSIZ	360#	399	400	630	637									
OLINK	352#	356	397	398	634	635								
QMQS	605	725#	0.1	• / •	0	0								
QMQW	539	679#												
OMTS	679	7210#												
OMTW	543	694#												
GNULW	547	647#												
QQSD	793#	777	740	747	744									
QQSTAB			710	713	716	719	722	725	745#					
	658	728	753#											
QQTTY	189	664	669	746#										
QR	437#	438	440	441	443									
ORNS	594#	670												
QRNW	534	656#												
QSCAN	163	270	438#	447	452	456	946	1031						
QSLPW	546	697#												
QSTAB	658	732#												
QSTOP	126	630	650/#											
QSTOPW	548	65Ø#												
QSTS	603	714#												
OSTW	537	688#												
QTIME	98	63Ø	726#											
OTIOWW	545	674#												
QTSS	596#	672												
OTSW	536	667#												
OTTAB	728	738#												
QWSS	595#	671												
QWSW	535	662#												
OX1	363#	366												
0X2	357	368#												
QX3	389	393#												
QXFER	100	127	148	200	348#									
RCXSKD	27	28#	T.40	C 10 10	340#									
RENMB	27 6#	6												
REGTAB			5104											
RNAVAL	55	567	569#											
RNQ	486	486#	4044	(04										
TON G	391	486	486#	600										

					•									
RNQUNT	750	752#												
RNREO	572#	571												
RUM	576#				•									
RUNARL		6												
	6#	6	90											
RUNMSK	6#	6	89											
SCHD1	164	176#												
SCHER	128	151#												
SCHEND	1762#													
SCNUCH	1260	1:27#												
SCNOK	1215	1219#												
SCNOUT	968	1127#												
SD	6#	6												
SERA	863	919	989	1368	13°5	1396	14.49	1410	1440	1453	4440	4.7.6.4		
SERACT	1383	1465	1750	1758#	10 2	1070	14 9	1416	TANK	1453	1462	1750	1755#	
SHF	6#	6	167	1282										
SHFWAT	840	852	856	984										
SHRSEG	6#	6	0.70	104										
SLEVEL	6#	6												
SLICE	6#	6												
SLPQ	512	512#	699	777										
SNA	6#	6	099	///										
SPYSEG	6#	6												
51 1 5 2 6	0#	D												
SQGO	1355	1382#												
SQGD1	1355	1386	1393#	1466										
SOGOS	1398	1426#												
SQIN	1237	1355	1367#											
SQLEN	1393	1423	1414	1759#										
SQOUT	1136	1355	1368#											
SQRED	857	1369	1387	1394	1418	1419	1423	1438	1446	1454	1464	1750	1751#	
SSCAN	162	630	768#					1.00	1 1 0	1424	1404	1756	1791#	
ST	435#	438	447	452	456									
STAVAL	489	499#												
STOPIO	6#	6												
STOPO	512#	652	776											
STQ	489	489#	501	693	690	766	779	788						
STRED	573#	574	21.4	0110	0-0	780	//9	/00						
STTYR1	6#	6												
STTYRE	6#	6												
SUMCOR	1006	1028	1059	1060										
SW	44#	77	82	88	89	90								
0.11	117	118	119	121		90	٥5	96	174	105	110	111	113	115
SWAP	156	843		121	125									
SWAPI	156 974		852#											
SWAP1	997	1228# 1120	1120#											
SWP	997 6#		1139#											
SWP1		6	78	115	142	167	949	1040	1797	1282				
	903	9:9	939#	1110										
SWP2	861	940#												
SWPCLR	6#	6												
SWPENT	1624	1625	1682	1683	1698	1701	1772	1742#						
SWPER1	1452	1457#												
SWPERC	987	99Ø	996											
SWPERR	1436	1460#												
SWPHGH	1693	1696												

SWPINI	1603	1695#												
SWPINT	1433	1435#												
SWPRFC	920	989#												
SWPSIZ	1740#	1742	1744	1746										
SWPTAB Sysdev	1720 6#	1742 6	1744	1746#										
Т	846#	876	878	892	894	895	896	901	902	904	905	906	907	948
	949	1011	1012	1020	1027	1028	1035	1040	1044	1045	1048	1049	1055	1257
	1059	1087	1088	1114	1115	1119	1120	1121	1122	1127	1128	1160	1161	1163
	1165	1228	1229	1231	1232	1233	1234				110			-100
T1	343#	352	354	355	363	365	374	375	376	377	847#	877	878	887
	888	1118	1123	1124	000		0, .	0,1	0,0	0,,	0.17#	0//	0,0	007
T2	344#	355	360	361	362	365	368	371	375	376	377	382	385	386
	848#	1119	1123	001	002	001	000	0/1	0,2	070	0,,,	002	002	000
TAC	6#	6	53	54	55	58	9	163	235	236	237	238	239	240
	242	244	245	344	436	848	915	946	989	990	991	992	993	994
	995	996	1231	1131	1157	1178	1179	1185	1199	1201	1216	1217	1218	1236
	1251	1252	1253	1257	1262	1263	1265	1279	1289	1292	1293	1294	1321	1322
	1323	1367	1368	1371	1372	1373	1384	1385	1387	1388	1389	1391	1321	1395
	1396	1397	1425	1406	1410	1426	1427	1428	1437	1441	1442	1444	1446	
	1448	1450	1451	1457	1458	1460	1461	1462	1463	1464	1633	1634	1635	1447
	1655	1656	1661	1664	1669	1675	1676	1677	1678	1680	1681	1695	1696	1653
	1699	1707	1001	100 .	1007	1075	10/0	10//	10/0	1000	1001	1040	1990	1697
TAC1	6#	6	44	169	171	343	437	847	952	1034	1043	1045	1061	1064
	1135	1150	1152	1153	1197	1200	1201	1203	1230	1231	1232	1234	1235	1250
	1260	1261	1265	1266	1280	1282	1283	1343	1344	1346	1369	1370	1382	
	1392	1394	1404	1416	1419	1429	1430	1438	1451	1628	1629	1630	1631	1383
	1633	1698	1699	1700	1.11	- 12 /	1 40 P	1400	7427	1020	1029	1030	1031	1632
TEM	6#	6	1077	1700										
TIMEF	38	74	92											
TIOWO	121	512	512#	676	785									
TRNERR	6#	6		0,0	102									
TRYSWP	837	1063												
TSAVAL	488	488#												
TSQ	488	488#	602											
TSRED	572#	573	01/2											
TT	340#	348	349	370	379	391	392							
TTYATC	6#	6	0.17	• / -	0.7	0,1	0.1							
TTYBIU	6#	6												
TTYUSE	6#	6												
UCHN	6#	6												
UIOMOD	6#	6												
USRMOD	6#	6	902											
UUO	6#	6												
UWP	6#	6												
UWPOFF	6#	6												
VIRTAL	841	888	987	993	1248	1266	1280	1693	1704					
VSCHED	19#	19					~	10.0	*/07					
WSAVAL	487	487#												
WSO	119	487	487#	6Ø1										
WSREC	571#	572												
WTMASK	6#	6	172											
XCKCSW	29	52#												

•

X J O B	60	626#	627	1001	1237	1016	1019	1345
XPAND	844	1321#						
XPANDH	1334	1337#						
ZERSWP	892	1277	1279#					

CODES	6#	511	544											
DISABL	6#	214	241											
ENABLE	6#													
NOSCHE	6#													
NOSHUE	6#													
PTTAB	643#	656	726											
QUEUES	6#	485	534	569	670									
SCHEDU	6#													
SHUFFL	6#													
STARTD	6#													
TTAB	639#	647	650	653	662	667	671	674	677	679	682	685	688	691
	694	697	7015	708	711	714	717	720	723	0, ,	0.2	000	000	091
X	478#	486	487	488	489	490	491	492	493	494	495	507#	511	512
	512#	529#	534	535	536	537	538	539	540	541	542	543	544	545
	546	547	548	563#	570/	571	572	573	574	575	576	577	578	579
	591#	680	601	602	6/13	674	675	606	607	608	609	2	570	273
XP	6#	6	18	499	590	501	513#	1307	1378					