Rank Xerox

A8

Universal Time-Sharing System (UTS)

Sigma 6/7/9 Computers

Initialization and Recovery Technical Manual

90 19 [illegible] A

Universal Time-Sharing System (UTS)

Sigma 6/7/9 Computers

Initialization and Recovery
Technical Manual

FIRST EDITION

90 19 92A

February 1973

Price: $3.50

NOTICE

This publication documents initialization and recovery processes of the Universal Time-Sharing System (UTS) for Sigma 6/7/9 computers. All material in this manual reflects the C01 version of UTS.

RELATED PUBLICATIONS

Title	Publication No.
UTS Overview and Index Technical Manual[†]	90 19 84
UTS Basic Control and Basic I/O Technical Manual	90 19 85
UTS System and Memory Management Technical Manual	90 19 86
UTS Symbiont and Job Management Technical Manual	90 19 87
UTS Operator Communication and Monitor Services Technical Manual	90 19 88
UTS File Management Technical Manual[†]	90 19 89
UTS Reliability and Maintainability Technical Manual	90 19 90
UTS Interrupt Driven Tasks Technical Manual[†]	90 19 91
UTS Command Processors Technical Manual	90 19 93
UTS System Processors Technical Manual	90 19 94
UTS Data Bases Technical Manual	90 19 95

[†]Not published as of the publication date given on the title page of this manual. Refer to the PAL Manual for current availability.

CONTENTS

ID

INITIAL

PURPOSE

INITIAL prepares a fresh copy of the monitor (from either the system device or tape) for operation.

USAGE

INITIAL is branched to by the tape or system device bootstrap after a fresh copy of the monitor root has been read into memory.

OUTPUT

A running monitor with two users (ALLOCAT and GHOST1), the monitor JIT and XDELTA in their respective physical memory areas, which have been removed from the monitor's page pool.

INTERACTION

GETHGP:	used to restore XDELTA on RAD or disk pack boot. (Section NG)
MONINIT:	used to read from tape, patch, and write to the system device the unlabeled portion of a system tape, monitor root, and system device bootstrap. (Section NB)
T:GJOBSTRT	used to start up ALLOCAT and GHOST1. (Section CC)
T:SE:	used to exit and schedule ALLOCAT and GHOST1. (Section EA)
T:SGRNU	used to release swap space for GHOST1. (Section FA.01.08)

DATA BASES

XPSDS is a 32 word table that INITIAL moves to the trap and interrupt locations X'40' to X'5F'.

IOXPSD	contains the XPSD that belongs in the I/O interrupt location (X'5C') but which is not in XPSDS because BOOTSUBR uses interrupts to signal completion of RAD or disk pack I/O.
CORXPSD	contains the XPSD that is temporarily stored in the non-allowed operation trap location while INITIAL is determining the physical memory size of the computer.

SUBROUTINE

LMA performs three functions:

1. Loads the memory writelocks with (01) for all pages except those which contain any part of the monitor between GETHGP and MONINIT (procedure area), which are given a lock of (11).
2. Loads the memory map so that each page is mapped into itself.
3. Loads the memory access protection registers with a code of (11) for all pages except page 0, for which the code is (10).

LMA is called by means of a BAL, 11 and destroys registers 0 through 6.

RESTRICTIONS

INITIAL must be entered master, unmapped.

DESCRIPTION

The type of computer and mode of SYSGEN is determined. If the system was generated for a large memory Sigma 9 but is running on a Sigma 7, an error message is typed on the OC device and initialization is aborted.

INITIAL then moves the monitor JIT from its loaded location in the root to J:JIT. The external interrupt locations (X'60' to MONORG) are then zeroed to prevent confusion of automatic LOGON. Then the trap and interrupt locations X'40' through X'5F' are initialized from XPSDS and the writelock, memory map, and access protection registers are loaded by LMA. If low order halfword of X'2A' is nonzero (tape boot) the I/O interrupt is armed and enabled for BOOTSBR's RAD or disk pack I/O and a BAL to MONINIT reads and patches the system tape and sets up the monitor area on the system device. Otherwise, GETHGP is used to read in XDELTA. Then the pages containing the monitor root, the monitor JIT, XDELTA, and non-existent memory are removed from the monitor's free page pool (MX:PPUT) and its head, tail and count set appropriately (M:FPPH, M:FPPT, M:FPPC). Nonexistent memory is found by trying to access the last word of decreasing memory sizes until a trap does not occur. LOW is set to the lowest non-monitor page number. HIGH is set to the number of the highest existent page. SL:CORE and S:PCORE are set up to reflect the number of pages found to be available to users. If an extra half page of monitor root exists above JITLOC, it becomes on SPOOL buffer. Then SL:CORE granules of swap area are released, starting at the end of the system-swap area (PSA) of the system RAD, so that GHOST1's swap granules will not conflict with shared processor granules. The DCT index of the system RAD or disk pack and the relative sector number of the last granule released is saved in BOOTSBAND.

INITIAL concludes by moving SYSVERS to X'2B', using T:GJOBSTRT to start up ALLOCAT and GHOST1, and zeroing SNDDX to turn off the symbionts (saving its contents in TSNDDX). It uses T:ADDGHOST to start up ALLYCAT and GHOST1 and sets NSWAP flag in UH:FLG2 to prevent swapping out GHOST1. INITIAL then exits to T:SE after replacing IOXPSD in the I/O interrupt location and arming and enabling the clock, memory parity, I/O, and console interrupts.

ID

BOOTSUBR

PURPOSE

BOOTSUBR contains the code necessary to read the unlabelled portion of a system tape, patch it, and write it to the system device in a bootable form.

OVERVIEW

BOOTSUBR is entered from INITIAL at MONINIT which returns through SWAPINIT and WRTROOT. The function of MONINIT is to get the patching operation started. In order to accomplish this, the operator must be allowed to change the device addresses of the card reader, line printer, and swapping device if the tape doesn't match the hardware, and the tape must be read as far as XDELTA, which contains the patching subroutine. Five subroutines in BOOTSUBR are used by MONINIT to facilitate this task:

MESSG request information from the operator.
VALINP requests, converts and stores new device addresses.
COCINP does the same for COC devices.
RTAPE reads a record from tape.
ERROREC provides error detection and recovery for printer, tape, RAD, and disk pack operations.

The function of SWAPINIT and WRTROOT is to create on the swap RAD copies of all the modules necessary to recover from a crash or to restore the operating system after memory-clobbering maintenance. In addition to RTAPE and ERROREC, SWAPINIT uses RWDSK, which reads and writes to the swap device a boot segment record. All patching is done by the PACHSTRT subroutine in XDELTA. WRTROOT uses WDISC, which is the tail end of RWDSK.

USAGE

MONINIT - BAL, 11 - all registers ignored and clobbered.
MESSG - BAL, 10

enter with:	0=doubleword address of output I/O command
return with:	BUF=3-word buffer containing the operator's response. Terminator is replaced by a zero byte.
	2=terminator byte.
	3=number of characters input, excluding terminator

4=zero
all other registers saved.

VALINP - BAL, 5

enter with:	1=OPLBL index 6=flag, if zero, indicates that a new address is not to be requested for this device.
return with:	1=DCT index 9=device address and if 6 non-zero: 0, 2, 3, 4, 10, 11 clobbered 8=old device address 1=DCT index+1

COCINP - BAL, 5 (Same as VALINP)

except:	7=COC index 1=half word displacement from DCT1 of device address in COH:DN. and 6 is ignored (assumed non-zeros)

ERROREC - SIO, 0 *6

BAL, 11 ERROREC

DATA n where n indicates I/O operation type:
0 - tape read
1 - card reader read
2 - system device (RAD or disk pack) write
3 - Printer write
4 - system device (RAD or disk pack) check-write

enter with:	0=doubleword address of I/O command 6=device address

return is to BAL+2 if the operation was successful or to BAL-1 if unsuccessful except on system device check-write failures, which return to BAL+2 with CC2 set.

RTAPE - BAL, 1

enter with:	7=number of one page records to be read, anything less than one interpreted as one.
return with:	tape record in memory at JBUPVP 1=number of bytes in the record. 0, 6,7, 11, 12, 13 clobbered.

SWAPINIT –entered from MONINIT, exits to WRTROOT.

RWDSK – BAL, 10

enter with: 7=number of pages in the record (on tape), same as RTAPE

return with: next record on tape read, patched and written to the system device.
3 = next available RAD track and sector address.
1 = number of bytes in the record
2 = word address of core image of record.
0, 6, 7, 11, 12, 13, 14 clobbered

WRTROOT – entered from SWAPINIT, exits through 11 saved by MONINIT.

WDISC – BAL, 10

enter with: 1 = record size in bytes
2 = word address of record.

return with: 3 = next available system device disk address
0, 6, 12, 13, 14 clobbered.

INPUT

That portion of the system tape between the end of the monitor root and the end of the unlabelled part.

OUTPUT

MONINIT:

SYSVERS the appropriate contents for X'2B' (monitor information)

CNDD, LLNDD, OCNDD device address of C, LL, and OC device for XDELTA's use.

DCT1, COH:DN,
M:SWAPD may be altered if the operator elects to change card reader, printer, COC, or swapping device addresses.

M:OC the DCT index of the OC device is put into the monitor's DCB.

DLTBIAS start address and execution bias of XDELTA

DLTSZ byte size of XDELTA

SWAPINIT:

P:SA	start addresses of overlays and GHOST1
PB:C#	if system device is disk pack, cylinder part of disk address for overlays and GHOST1.
PB:DCBSZ	number of GHOST1 DCB pages
PB:DC#	if system device is disk pack, cylinder part of disk address for GHOST1 DCBs.
PB:DSZ	number of GHOST1 or ALLOCAT data pages.
PB:HVA	lowest virtual page above GHOST1 or ALLOCAT procedure
PB:PSZ	number of procedure pages of overlays ALLOCAT and GHOST1.
PB:PVA	smallest procedure page number of overlays ALLOCAT, and GHOST1.
PH:DDA	disc address of GHOST1 and ALLOCAT DCB's and DATA
PH:PDA	procedure disc addresses of overlays ALLOCAT and GHOST1
RCVSIZE	number of bytes in RECOVER and its byte load address.
DLTDSC	disc address of XDELTA.
RCVRAD	disc address of unused PSA area
RCVSTART	start address of RECOVER
RCVDISC	disc address of RECOVER

INTERACTION

PACHSTRT is used to read patch cards and make the patches. (Section LA)

SUBROUTINES

MESSG communicates with the operator through the OC device. The I/O command pointed to by register-0 is first fed to the typewriter. When this operation completes, a read is issued which reads into the 3-word buffer BUF a maximum of ten characters, but which will also terminate when a NL, EOM or TAB character is received. If a NL or TAB occurs before the tenth byte, it is replaced in BUF by a zero but saved in register 2, its displacement into BUF is put in register 3, and a return is made through register 10. If this condition is not met, MESSG is re-entered through EOM, which points register 0 to a command which types question marks.

VALINP uses OPLBT2 to obtain DCT index and DCT1 to obtain device address of the device indicated by OPLBL index in (1). Then return if (6) is zero. If not, DC16 is used to generate a request and MESSG is used to get a new device address which is converted to binary in (9). If not valid, the address is re-requested and processed through the EOM entry to MESSG. If valid, it replaces the address in DCT1.

COCINP generates a device address request from COH:DN and enters VALINP at the BAL to MESSG.

ERROREC first checks if the SIO was accepted. If not it is re-issued unless the device was unrecognized or not operational in which case an appropriate message is typed followed by a WAIT instruction and the re-issue logic. Otherwise, a status-checking loop awaits the end of the operation. If the device is RAD or disk pack, the loop is preceded by a WAIT instruction to avoid data overruns. If the device is found by the loop to be in manual mode, a message is typed and the loop is interrupted until automatic mode is detected. If the device is busy, the loop loops. If the operation completes normally, the internal retry counter is reset (unless this was a RAD or disk pack write with check-write to follow) and a return is made. If an unusual end occurs, appropriate corrective action is taken. For the card-reader, a message is typed, followed by a WAIT and the re-issue logic. For RAD, disk pack, or printer, TDV status is examined to choose between retry and operator intervention (message, WAIT, re-issue). For tape, if the error is unrecoverable, operator intervention is chosen. Otherwise, a backspace record is followed by a retry.

RTAPE issues n(contents of R7) read commands (into successive pages starting at J:JIT + 512 - byte count X'800', waits for completion with ERROREC, and then returns in R1 the total number of bytes read.

RWDSK BAL's to RTAPE, then to DELTA's patching routine, then drops into WDISC.

WDISC puts the buffer address into a write command doubleword in registers. The requested size is adjusted to insure that an even number of sectors in written. Then enough command chaining writes are generated to write the whole record, with the last byte count adjusted to write only what is required. The writes follow a seek from DISCLOC and precede a sense to SENSW. The SIO is issued and ERROREC waits for the interrupt. Then the writes are changed to check writes and another SIO-ERROREC sequence follows. If the operation is unsuccessful, WDISC is re-entered at the top. Otherwise, SENSW replaces DISCLOC and WDISC returns.

RESTRICTIONS

Must run master mode, unmapped (or mapped 1-to-1).

DESCRIPTION

MONINIT first moves itself to X'8000' so that the monitor root will checkwrite with the fixed byte count of X'20000'. Then it saves (11) in MONIRTN and finds the OC device address via OPLBT2 and DCT1. The OC DCT index is put into the monitor's OC DCB (M:OC). The operator is asked (through MESSG) if he wants to change any device addresses. If he doesn't (6) is set to zero so that VALINP will not change any addresses. VALINP is then used to find (or change and find) the C and LL device addresses. Then if (6) is non-zero, the GETCOC loop uses COCINP to get new COC device addresses. The GETDC loop uses VALINP to get or change swap device addresses and checks each device to determine that it is what it is supposed to be. If not, a message indicating which is not what is output and VALINP is entered after its check on (6) to change the address. Then the track and sector address of the first overlay is calculated (by reserving granules for bootstrap and HGP) and saved in DISCLOC for WDISC.

If the system device is a Disk Pack, flawed tracks are checked for by reading each header in the PSA and insuring that the first byte is 0. 'PSA Track Flawed" is output to the operator's console if any flawed tracks are found, and the system executes a WAIT so the operator will correct the situation and reboot from tape.

Next, the physical page table in the JIT, JX:CMAP, is initialized with the FMPC constant, starting at JOVVP. The entry in JX:CMAP for the JIT page is set to JJITVP. Then the command list section of the JIT, J:CL, is initialized with blank SEEK and WRITE IOCDs, which have the appropriate flags and byte counts set.

The next record (Monitor Information)is then read by RTAPE from the boot tape. The version level contained therein and other bits appropriate for location X'2B' are saved in SYSVERS. The record is printed line by line and then typed all at once, the first character of each line having been replaced by a NL character during the printing. XDELTA's HEAD is read next. The start address and bias address are saved in DLTBIAS. Then RTAPE reads XDELTA. Its size is saved in DLTSZ and is moved to its bias address. MONINIT exits to SWAPINIT through XDELTA's patching routine after typing the sense switch reminder to the operator.

SWAPINIT consists of two loops, each with initialization, and an end. The larger loop, RHEAD, processes the overlays, goes to the other loop, RWJIT, to do the JIT then processes ALLYCAT and GHOST1 and drops into its end to process RECOVER and write XDELTA. The initialization for RHEAD sets registers 2 through 4 with the buffer address (JBUPVP), the next available disk address (DISCLOC), and the processor table index for ALLYCAT. RHEAD begins by reading the tape (RTAPE). The processor index (4) is decremented and the start address moved from the head to P:SA. Then if (4) indicates overlay, the data bias and size are moved from the head to PB:PVA and PB:PSZ and for Disk Pack the disk address of cylinder 1 set into DISCLOC. Otherwise, the DCB size, data size, and procedure bias and size are moved from the head to PB:DCBSZ, PB:DSZ, PB:PVA, and PB:PSZ, the sum of the last two to PB:HVA, the current disk address (3) to PH:DDA (and if applicable PB:DC#), and the next two records (DCB's and DATA) on tape are read, patched, and written to the system device by RWDSK. Then, in either case, the current disk address goes to PH:PDA (and PB:C# if Disk Pack) and the next record is read, patched and written to the system device by RWDSK. If the processor index is that of an overlay, RHEAD loops. If not, it drops into the end processing.

The end processing reads the RECOVER head and sets up RCVSTART in RCVSIZE and RCVDISC. Then RWDSK reads, patches, and writes RECOVER to the system device. The current disk address is saved in DLTDSC and WDISC writes XDELTA. The command lists used to read RECOVER and XDELTA are modified to handle a byte count of 4 for the seeks if the system device is a Disk Pack.

WRTROOT first initializes the batch bias table (C:SCOB). Then WDISC writes the monitor root to the system device saving the addresses in DABOOT and DABOOT+1 in the bootstrap assembled with BOOTSUBR. Then the current disk address is converted to relative sector number and set in RCVRAD along with the system device's DCT index so that RECOVER and SYSMAK will know where to put crash dumps and shared processors. The bootstrap is written to cylinder - 0 (if disc pack), to track - 0, sector - 0 and WRTROOT returns to INITIAL through MONIRTN.

ID

GHOST1D

PURPOSE

Central control module for the system initializer, GHOST1.

INPUT

OCNDD is used to find the OC device.
JB:CMAP is used to convert virtual addresses to physical
DLTBIAS, DLTSZ are used to return XDELTA's pages to the physical page pool (MB:PPUT)
PASSO's M:BI DCB is used to find the address of the boot tape (if there is one).
Word X'2A' is used to determine the type of boot;

Contents: 0–7	8–15	16–23	24–31	Type of Boot
≠0	0		≠0	Tape Boot
- 1				Tape Boot saving file structure
≠0	0			Cold Start disc boot
1		0		Operator Recovery
0				Crash recovery

OUTPUT

TIME and DATE cells in the monitor root are initialized on non-recovery initializations.

INTERACTION

ERRLOG	to record the startup event. (Section KE.01)
PASSO	to read, patch and write the system account files. (Section HG)
RECOVER2	to restore the system environment following a crash or ZAP Keyin. (Section OG)
SYSMAK	to put shared processors on the swapping RAD. (Section NE)

T:GJOBSTRT	to logon the ANLZ ghost
T:BTSCHED	to start up batch job still enqueued after a crash. (Section EA)
COCINIT	to start up the COC device. (Section DC.01)

SUBROUTINES

MESSG communicates with the operator through the OC device. It is identical to the subroutine of the same name in BOOTSUBR except that it has a 9 word buffer, converts I/O command doubleword addresses to physical addresses before issuing SIO's. If the keyin has an obvious default, a wait loop is entered which aborts the read after five seconds unless the operator has started inputting.

DATETIME obtains from the operator the current date and time and stores them into the DATE and TIME monitor cells. MESSG is used to request the date (MM/DD/YY). The number of fields is set to 3 (month, day, year) and the GETDIG loop is entered to scan the response. GETDIG scans from left to right shifting valid decimal characters into the right end of the accumulator register (1) (initially minus one) until a delimiter is encountered. If an invalid character is encountered or if the ninth input character is not the delimiter, EOM is entered to re-request the input. When the delimiter is found, GOTIT checks the accumulator. If still minus one, EOM. If only one character was shifted in (byte 2 is X'FF') byte-2 becomes X'F0'. Then the result is checked against MIND and MAXD which are halfword tables of minimum values indexed by field number (counted right to left, e.g. MM is field 3). EOM if the value is too small or too large. The acceptable value is stored at the halfword displacement from DATE contained in the byte table STORE (indexed by field number, contains 0, 3, 1, 0 for date, 0, 5, 4 for time). Then a BDR on field number to GETDIG-1 (re-initializing the accumulator) continues to scan the input. When the BDR drops through and the delimiter is not "/", DATETIME returns to the caller. Otherwise the final contents of DATE is checked against a list of bad dates (BADATES) not caught by MIND and MAXD, i.e. 31st day of months with 30 days and February 30. If the date is February 29, the year must be a leap year. If all is well, MIND, MAXD, and STORE are changed to reflect time requirements, MESSG is used to request the time. The delimiter is set to ':' and stored over the terminator, the number of fields is set to 2 and GETDIG is used to check and store the time and return.

RESTRICTIONS

GHOSTID must run master mapped with special JIT access and must be the first ROM on the GHOST1 LOAD command.

DESCRIPTION

The I/O command buffer addresses are converted from virtual to physical addresses. The operator is asked (through MESSG) if XDELTA is to be retained as a debugging aid. If the response is not "Y", LEES WATERING HOLE is disabled and XDELTA's page chain is tacked onto the end of the monitor's free page pool chain, adjusting M:FPPT, M:FPPC, MB:PPUT and S:PCORE (free page pool tail and count, the chain itself, and the physical core size available to users) appropriately.

If this is a crash or operator recovery, GHOSTID BAL's to RECOVER2 and SYSMAK and if the recovery was successful, to COCINIT and T:BTSCHED. Then ERRLOG records the startup and TSNDDX replaces SNDDX to turn on the symbionts. The ANLZ ghost is started up and GHOST1 exits to FILL.

For a tape boot, GHOSTID first BAL's to PASS0 and rewinds off-line the boot tape (whose address is in X'25') and, if it is on a different drive, the system account tape (whose DCT index is in the M:BI DCB). Then GHOSTID BAL's to SYSMAK, DATETIME COCINIT, T:BTSCHED and ERRLOG, turns on the symbionts and exits to FILL.

For a boot-under-files, GHOSTID first BAL's to RECOVER2 to restore the system environment if the system was last shut down via a ZAP keyin. For a disc boot, MESSG is used to ask the operator if he wants HGPRECONSTRUCTION, in which case GHOSTID BAL's to HGPRECON. Then, or otherwise, GHOSTID continues as for a tape boot but skipping the PASS0 and tape rewind phases.

ID

PHASEC

PURPOSE

To get the input tape copied to system account files and then perform the modification to those files dictated by GENMD and GENDICT commands.

USAGE

CCI0 branches to PASS1NXT when it runs out of control command cards. PHASEC returns to PASS0's calling program by pulling the temp stack empty and using the last word pulled as the return address.

On entry: (R0) = Temp stack pointer address
(R6) = SEGNAM table control words address (see PHASEB)
(R7) = CCPL address (control card PLIST for using SEGSERCH).
(R14) = Current SEGNAM table pointer (see PHASEB usage)

INTERACTION

BITOTM	to copy the input tape
ERRABN0 (CCI0)	to type IO errors/abnormals
GENABS (PHASED)	to do nothing
MODIFY	to perform GENMD and GENDICT modifications
SEGSERCH (CCI0)	to check segment names in TREE's

M:GP, M:OPEN, M:CLOSE, M:READ, M:WRITE

SUBROUTINE

A BAL, 11 to READSEG with the address of a TREE segment name in R10 and a buffer address in R9 will cause a modify parameter list (MODPLIST) to be generated and the segment read into core. Only the REFDEF stack, relocation dictionaries, and protection type 00, 01 and 10 records are read. If the module is ABS, the relocation dictionaries are not read. If the module is paged (type-85), only the REFDEF stack is read. The MODPLIST is set up first and the information contained therein is used to calculate buffer addresses and sizes for the reads. If a record will not fit in PHASEC's buffer, a buffer size of one byte is used to force an IO abnormal since GHOST1 runs master mode. An alternate entry point called WRITESEG writes the segment the same way the read worked except that the MODPLIST is assumed to be already set up and if the module is paged, only the page that is currently in CORE (if any) is written.

RESTRICTION

The end of PHASEC (PAGES) is the SEGNAM table origin so it must be loaded after PHASEB and any GHOST1 module that is exexecuted after PHASEB.

DESCRIPTION

All of memory is requested after releasing all of GHOST1's data the upper limit is saved in COREND, and the IO buffer address is calculated from R14 (end of SEGNAM table). BITOTM is called to copy files from tape to disc. If there are no modifications, BITOTM returns to PASS0's calling routine. Otherwise, PHASEC's MODTMTM loop is entered.

MODTMTM opens M:TM with update and next file options (and a zero file name the first time). If the next file cannot be opened and it is not because it is synonymous, PHASEC frees memory and returns to PASS0's caller. If the file is not keyed, M:TM is closed and the next file is tried. The HEAD record is read to determine the TREE size and whether the module is ABS or paged. The TREE is read on top of the HEAD and the NXTREENT loop searches the TREE for segments which have modifications.

In NXTREENT, each segment name is moved to the CCPL as though it had just been picked off a GENMD card. Then a BAL to SEGSERCH determines whether modifications exist. If they do, READSEG reads in the segment and one BAL to MODIFY for each chained Change Description Table in the SEGNAM table does the modifications. If the module is paged, MODIFY BALs to PAGEMOD in PHASEC with what to store where in registers. PAGEMOD stores the patch if the affected page is in core. Otherwise it rewrites the page that is in core (if there is one), reads the right one, and patches it.

UTS TECHNICAL MANUAL

ID

SYSMAK, SYSMAK 1

PURPOSE

SYSMAK - To copy shared processors (except GHOST1) to the swap device from files in the system account.

SYSMAK 1 - To copy the specified shared processor to the swap device from the input file.

USAGE

BAL, 11 SYSMAK all registers clobbered (except 11)

BAL, 11 SYSMAK 1 all registers clobbered (except 11)
Reg6 = address of buffer to use
Reg7 = processor number
M:EI open to the processor load module file.

INPUT

J:DLL	to release GHOST1 data pages
MB:SDI	swap device DCT index
P:NAME	names of shared processors
RCVRAD	relative sector number of shared processor area on RAD
MB:GAM4	to reserve space (RCVRDSZ)
HIGH	to calculate required RCVRDSZ
BOOTSBAND	end of processor area on RAD
PB:C#, PB:DC#	cylinder number of area on device (SYSMAK 1)
PH:DDA, PH:PDA	disc address of area on RAD (SYSMAK 1)
SPSIZE	size of processor replacement slots (in pages)

OUTPUT

P:NAME	processor overlay names
P:AC	access image for special shared processors (double word table)
P:SA	start address
P:TCB	TCB address
PB:C#	cylinder part of procedure disk address for Disk Pack
PB:DC#	cylinder part of data and DCB disk address for Disk Pack
PB:DCBSZ	DCB size (pages)

PB:DSZ	Data size	
PB:HVA	Next page above procedure (including largest overlay)	
PB:LNK	Next overlay processor index	
PB:PSZ	Procedure size	
PB:PVA	Procedure bias (page)	
PH:DDA	DCB's - Data disc address	
PH:PDA	Procedure disc address	
RCVRDSZ	Size of RECOVER's RAD dump area	SYSMAK only
M:SBAND	Lowest valid swapping track and sector	

INTERACTION

PUTHGP	to save the file structure	
M:OPEN	to open files (JIT's, Processors)	
M:CLOSE	to close files	
M:GP, M:FP	to get and release memory	SYSMAK only
T:SGRNU	to release swapping RAD granules	
T:SGAJIT	to swipe swapping RAD granules	
M:TYPE	to type messages	
SCREECH	to crash	
M:READ	to read files	
GMB	to get a monitor buffer	
RMB	to release a monitor buffer	
NEWQ	to write to the swap device	

SUBROUTINES

SET$PROC$TAB moves information from the HEAD (for P:SA, P:TCB) and TREE records to all processors tables except PB:LNK, PH:PDA. P:AC is set as follows (bits per page):

> Bits 0 to 1 = 00 if PB:DSZi =1; 01 if PB:DSZi = 0; read, write, execute access.
> Bits 2 to 2n = 1 = 01 where PB:PSZi = n (n pages of procedure). Read, execute.
> Bits 2n + 2 to 63 = 11 no access.

RADWRITE writes to the swap device. It is called with a BAL,8 with (9) containing the word displacement from the buffer (input for SYSMAK 1 or the page boundary above SYSMAK) of the data to be written (must be on a page boundary), and (15) containing the first byte and the appropriate disk address table address (either PH:PDA or PH:DDA) in the rest.

PH:PDA or PH:DDA is first set up as computed from SENSW. The relative sector number is specified in SENSW and each write of one page is followed by incrementing SENSW and decrementing #PGSLEFT. Prior to writing procedure or data and DCBs, a check is made to insure that the number of pages to write is greater than #PGSLEFT so that each will be contained on one cylinder if system device is a Disk Pack. If the swapping device is a Disk Pack, PH:C# or PB:DC#, the cylinder part of the disk address, is set up at this time; whichever is appropriate is determined by the disk address table address in register 6.

The seek address is specified in SENSW. The seek address of the last granule written is in DISCLOC and the next granule to write to is in SENSW. To write to the RAD NEWQ is called with end action. The end action routine, placed in a monitor buffer so that the routine may be executed unmapped as required, saves the type of complete. The completion code is checked for errors when NEWQ returns. If a swapping error occurs or if the write is to the address in BOOTSBAND, a message is typed and SYSMAK skips to the next processor or if SYSMAK 1, exits. RADWRITE clobbers all registers except 4, 5, 6, and 7.

ERRORS

TYPE is entered with (14) pointing to a message (TEXTC). If SYSMAK 1, this is converted to an error code in (5) and SYSMAK 1 exits. It appends the message to the current processor name and types this. It then skips to the next processor. It is entered on IO errors and abnormals or if a RADWRITE tries to write above BOOTSBAND.

The messages are:

"UNREADABLE"	if an I/O error occurs opening or reading the file
"NOT IN SYSTEM"	as indicated when trying to open the file
"OVERFLOWS"	if there is no more space on the system device
"SWAP IO ERROR"	if an error occurs writing to the swap device
"ILLEGAL LM"	if the load module is illegal
"CANT OVERLAY"	if no overlay slots are available.

RESTRICTIONS

SYSMAK or SYSMAK 1 must run master, mapped. Since SYSMAK uses all memory above itself as a buffer, it must be loaded after any module in GHOST1 which must execute after it, and must run with the special JIT access flag. NSWAP flag in UH:FLG2 is set by INITIAL to prevent swapping out GHOST1.

DESCRIPTION

SYSMAK 1 performs a subset of the SYSMAK logic. The entry and exit points for SYSMAK 1 are noted below.

SYSMAK initialization consists of: saving return in TSTACK, getting all available memory, calculating and saving in BUFFER the page boundary above SYSMAK, and moving the starting relative sector number from RCVRAD to SENSW. #PGSLEFT is set up with a number greater than the possible processor area (if RAD), or with S:CYLSZ, the number of granules in a cylinder (if Disk Pack).

The processor index is initialized to MAXOVLY (the first shared processor which is not a monitor overlay). An attempt is made to open the file named in P:NAME unless it is M:DUMLM, in which case the relative sector number from SENSW is put in PH:DDA and SENSW is bumped SPSIZE granules. If I/O errors or abnormals occur opening or reading, M:DUMLM is put into the P:NAME entry to create a replacement slot.

SYSMAK 1 enters at this point with a flag indicating SYSMAK 1 and the processor number in register 7, having saved in BUFFER the buffer address provided in register 6, and setting up in SENSW the processor's data relative sector number from PH:DDA. M:EI has been opened to the appropriate file by whoever called SYSMAK 1. If Disk Pack, #PGSLEFT is initialized by computation from SENSW and S:CYLSZ; otherwise, it is set with a number larger than the size of the processor area.

The HEAD record is read into SYSMAK's data area HEADER, the TREE is read into the IO buffer (pointed to by BUFFER) and a BAL to SET$PROC$TAB fills in most of the tables. The next relative sector number (from SENSW) is put in PH:DDA and the DCB's and data records, if present, are read from the file and written to RAD. The procedure record is read and written with its disc address going into PH:PDA. If this is a SYSMAK 1 call and the processor is a monitor overlay, then the size and disc address of the data is put into the processor's pure procedure size and disc address tables instead of the tables for data since the data in the load module of a monitor overlay is really pure procedure. If the TREE size is 12 words, this processor is complete, and the processor is complete, and the processor number is incremented. The calling routine is now returned to if this is SYSMAK 1.

If the processor number is now that of GHOST1 or ALLOCAT, it is incremented again. If the processor is TEL or CCI, SENSW is bumped up 4 granules to leave space for them to grow is they are later replaced. If M:EI is open, an M:CLOSE closes it. If there are still processors to process, the processor loop loops. If the processor index is PNAMEND, all the processors are complete, the disc address in SENSW is set into the 1st entry of PH:DDA (and PB:DC# if Disk Pack). The required RCVRDSZ is calculated (HIGH+3). If current RCVRDSZ is not adequate, it is set to the required size and the end RAD address of the dump area is calculated. If the swap device is a Disc Pack, RCVRDSZ is set to the next higher cylinder boundary. The user cylinder table UB:C# is set up. The user slots assigned to cylinders outside the PSA area are taken out of the system by removing them from state 0 and outputting a message to the operator indicating how many users are now in the system. If this is greater than BOOTSBAND, RCVRDSZ is decremented until it isn't. Then all available swapping granules above BOOTSBAND are swiped and their addresses saved in a table. M:SBAND is set to zero and the swiped granules and all granules between BOOTSBAND and the end of the dump area are released. Then memory is freed, the return address is restored, and SYSMAK exits. If the TREE is bigger than 12, the processor's overlay segments must be put on RAD. The last partial page (unless it is a full page) of the root segments procedure (still in memory) is moved down to the second page of the IO buffer (the first page has the TREE in it). Overlay segments will be read in at the end of the root portion of this page and the whole page will be written to the system device as the first page of the overlay so that pages on the system device will correspond to pages of execution memory. PB:PSZ and PB:HVA are decremented for the root segment, and DISCLOC is moved to SENSW to back up the disk address one page, since the last page of the root is no longer part of the root. Then a loop is entered which processes the TREE from the last segment to the second one. For each segment, the name is moved from the TREE to P:NAME, in any zero entry in the processor overlay portion (NAMEND to PPROCS). If no such zero entry exists, SYSMAK aborts this processor through the I/O error logic. The segments procedure size (including the last page of the root) is put into PB:PSZ. Each segment is linked to the previous one by storing the previous index (initially zero) in PB:LNK. PB:PVA is set for the overlays from PB:HVA of the root and the largest segment's size is saved. Only the procedure record of the segment is read from the file and written to the system device, with its disc address going into PH:PDA. When the TREE runs out, the last overlay index used is put in PB:LNK for the root, PB:HVA for the root is increased by the largest segment's size, and the main processor loop is re-entered after its TREE size check.

ID

GPHGP - GETHGP, PUTHGP, GETHGPND

PURPOSE

GETHGPND- To move file and symbiont granule pools and account directory information from RAD to memory.

GETHGP- Same as GETHGPND except also moves XDELTA from RAD to memory.

PUTHGP- Same as GETHGPND except movement is from memory to RAD.

USAGE

BAL, 11

Clobbers registers 0-8.

Does not return if transmission data or memory error or memory address error occurs repeatedly.

DATA BASES

HGPCL is an I/O command list for reading or writing HGPSIZE-3 words starting at HGP+4 to RAD at track-0 Sector-1.

RDLT is a command list for reading XDELTA from RAD.

DLTSZ-byte size of XDELTA DLTDSC- disc address of XDELTA DLTBIAS- load bias and start address of XDELTA	Initialized by BOOTSUBR

DESCRIPTION

On all I/O operations, a retry is attempted if any errors occur.

GETHGP first executes the RDLT command list. Then, and this is GETHGPND's entry point, the HGPCL is set to a read operation. This is also the entry point for PUTHGP, except HGPCL is set to write operation. Three words at HGP+4 are saved in registers and replaced in memory by the word at ACNCFU+1, the byte at ACNCFU+3, and the word at ACNCFU+5. Then the command list is executed and the three words at HGP+4 are saved in registers and the old values restored.

If it was a check write, PUTGP exits. If it was a read, the ACNCFU information (in registers) is moved to ACNCFU and GETHGP (ND) exits. If it was a write, PUTHGP changes it to a check write and starts over.

ID

RCVCTL - Recovery Control Program (UTS - C01)

RCVCTL contains the following subroutines

TYOUT
TYIN
COCOUT
HEXCVT
RBOUT

PURPOSE

The RCVCTL module controls the flow of the system recovery.

INPUT

a. Saved Register 15 (SAVEREGS+15)
   = -1, Operator initiated recovery
   <X'FF', Screech Code
   >X'FF', Unknown recovery activation
b. DATE, DATE+1 MMDD -- YY
c. TIME HHMM
d. Error Logging Information (See Section VK)
   1) Error Log Pointers (SGRAN, BGRAN, CURGRAN, FGRAN1, CURBUF)
   2) Error Log Buffer (CURBUF contents and preceding control words)
e. Physical address of System Boot Device in M:SWAPD
f. Administrative message in COCMESS
g. Number of words in recovery buffer in BUFLN.
h. RCVRAD containing RAD address for core dump
i. RCVDISC containing disc address of RECOVERY
j. The first disk address of the released granule chain in RELFDA
k. R:TSTACK - temp stack

OUTPUT

a. Recovery Buffer containing items b, c, d, f and j above.
b. Bootstrap Routine to Simulate RAD Boot.
c. Operator/User Messages: (Messages originate from the various routines called by RECOVERY). Additional messages are output by HGP reconstruction during second phase of recovery.
d. RCVRAD address for core dump
e. Location X'2A'

X'0'	-	Crash
X'00010000'	-	Operator recovery
X'40404'	-	Shut Down

MESSAGES

Identify the cause of the recovery procedure:

OPERATOR RECOVERY
RECOVERY FOR UNKNOWN REASON
SOFTWARE CHECK XX

All users are advised:

RECOVERY SAYS - STAND BY -

When the HGP or CFU table is deemed destroyed, All users are informed:

RECOVERY ERROR - ATTEMPTING FILE SYSTEM RECOVERY

Correspondingly, the operator is told:

RECOVERY ERROR - STARTING HGP RECONSTRUCTION

Inform Operator that the HGP table has been destroyed.

HGP MALFORMED.

Inform the operator the Granule Stack double words overlap.

BUFSPD OVERLAP

Inform the operator that the granule stacks have been destroyed.
BAD DA IN GRAN STACKS

Inform the operator that an active private pack was found and its HGP was restored but may have lost some cylinders.

PRIVATE VOLUME XXXX MAY HAVE LOST SPACE

Inform the operator that an entry in the AVR table of Private volume serial numbers does not match the serial number on volume mounted.

PRIVATE VOLUME XXXX IN AVRTAB DOESN'T MATCH DPndd.

Indicate to the operator that file FFFF in account AAAAAAAA will no longer be accessible by its name. This is a result of the file having been defective as the result of the crash.

FILE FFFF ACT AAAAAAAA MADE INACESSIBLE

If bad files are detected, the operator is asked:

BAD FILES DETECTED, DO YOU WISH TO BREAK SERVICE

If the operator responds with a 'Y' to the above question, all users are informed:

RECOVERY SAYS WE HAVE PROBLEMS

and the operator is informed:

THE SYSTEM IS YOURS

The operator is asked to enter the I/O address of a tape unit having a scratch tape mounted which will receive the core dump. If 0 is entered, no dump will be performed.

ENTER I/O ADDRESS FOR TAPE DUMP (0=NONE) (e.g., A80, 080, 80)

an invalid response has been received. The initial request will be repeated.

EH?

Trouble with the tape dump device. A request for a new tape dump device will be issued.

I/O ERROR, TAPE DUMP DEVICE

The tape dump was successful, dismount and save the tape for later analysis.

DUMP TAPE - RCVT - IS ON UNIT XX

The job with IDXXXX was partially completed at time of Recovery. The remainder of the job will not be run.

JOB XXXX PARTIALLY COMPLETED

The partial symbiont output files will have as their last record.

LAST RECORD SUPPLIED BY RESTART

Impossible recovery conditions found by RECOVER2.

RECOVERY TABLE CLOBBERED - UNABLE TO CONTINUE

If the HGP or CFU tables in core are destroyed and the account directory on disc is bad, the users are informed:

RECOVERY IMPOSSIBLE - MUST RELOAD WITH FILL TAPES

Similarly, the operator is informed:

RECOVERY IMPOSSIBLE - SYSTEM UP FOR FILE SYSTEM SAVE ONLY

For the I/O subroutines, MAPFLG is preset to non-zero if core is mapped or zero if core is not mapped.

External:

All routines entered via BAL, 11

Name	Purpose	Manual Reference
TSTHGP	Validity check Disk allocation	OC
TSTUSR	Test user tables	OF
RCVDMP	Recovery dump (Swap RAD)	OG
SV1	Save one word in RECOVERY buffer	OC
SYMFILS	Package symbiont files	OB
CYCUSR	Verify and save users' JIT's, DCB's, etc	OD
SAVHGP	Save file allocation tables	OC
TAPDMP	Tape Dump	OC
SAVSYM	Restore JIT and AJIT disk address of Symbiont Ghost job from ALLOCAT. Save symbiont communication buffer.	OD
SAVSYM1	Save symbiont tables, remote batch tables, and the symbiont communication buffer in recovery buffer.	OD
SVDNDEV	Save down device list	OC
MVEBUF	Move RECOVERY buffer to swap RAD	OC
SYSLIM	Save system limits	OC
RRBG10	Truncate release granule buffer	
CKRAD	Check validity of Seek address	OE
R:CHKDA	Check validity of Disk address	OE

DESCRIPTION

The date and cause of the recovery are typed to the operator and the users are asked to stand by. The HGP tables are now checked by TSTHGP; TSTUSR is called to verify all user tables. Next, RCVDMP is called to write all of core to the swap RAD if space is available, otherwise, to tape. SAVSYM is called to restore the symbiont ghost JIT, AJIT disk addresses and SWAP I index in core. These values are saved by ALLOCAT at system start up time. The symbiont Communication buffer is moved from Monitor data to a buffer in recovery so that entries may be added by recovery without destroying any monitor data. Then CYCUSR is called to verify every user JIT, close all files and remember the JIT's for accounting purposes. If any bad files are found or the CFU tables are clobbered a flag is set.

SVDNDEV is called to save the down device list. The partial error log and pointers are saved, and SYSLIM and SYMFILS are called to save system limits, partition limits and process active symbiont files for continuation. If no bad files have been detected, SAVHGP is called to empty the granule allocation stacks and rewrite Private Pack HGP's.

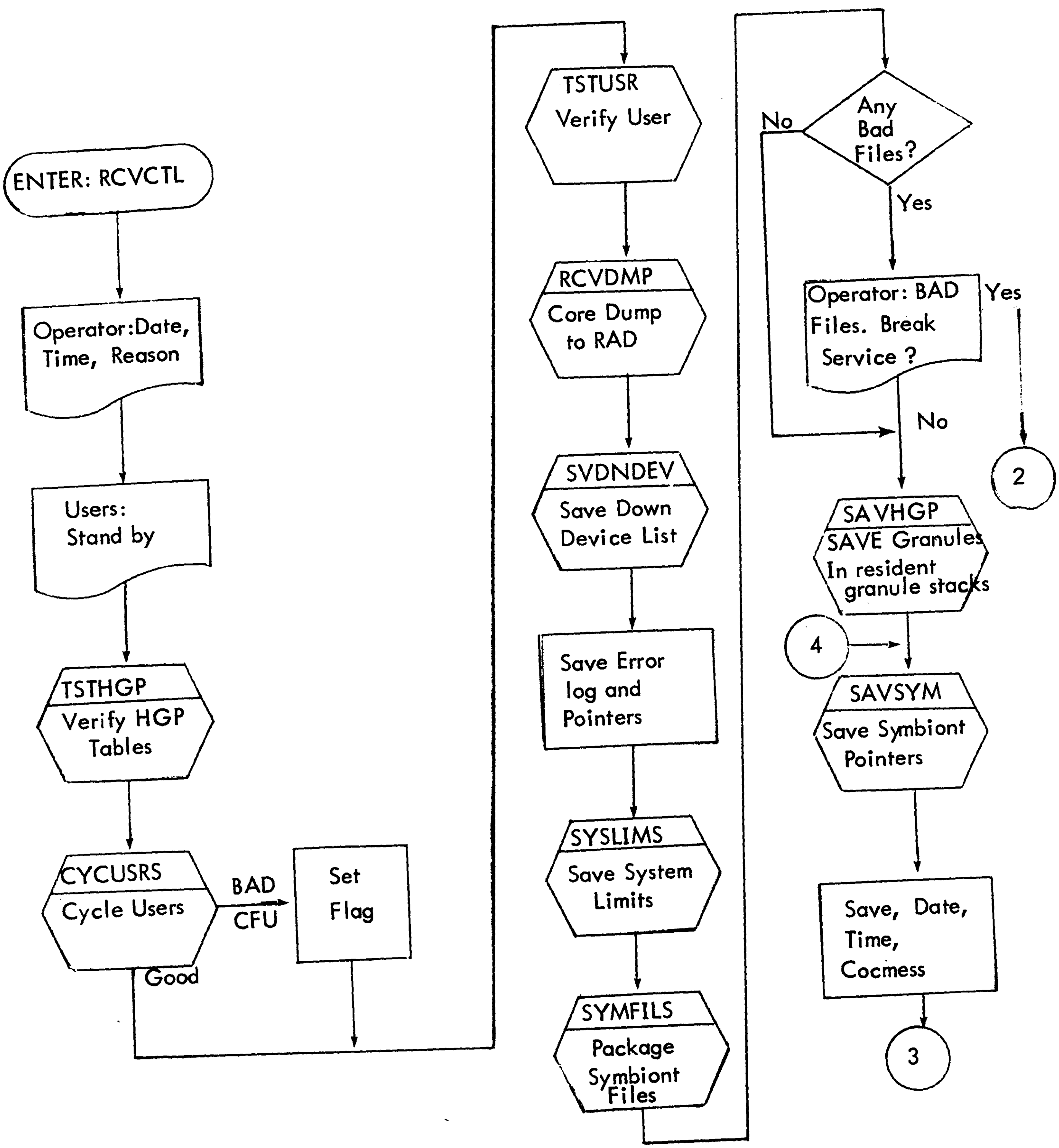
SAVSYM1 is called to put symbiont tables, remote batch tables (in a remote batch system), and the symbiont communication buffer in the recover buffer. The date, time and administration messages are put into the recover buffer. The disk address of the granule that contains the addresses of granules to be released by recover 2 is put into the recover buffer.

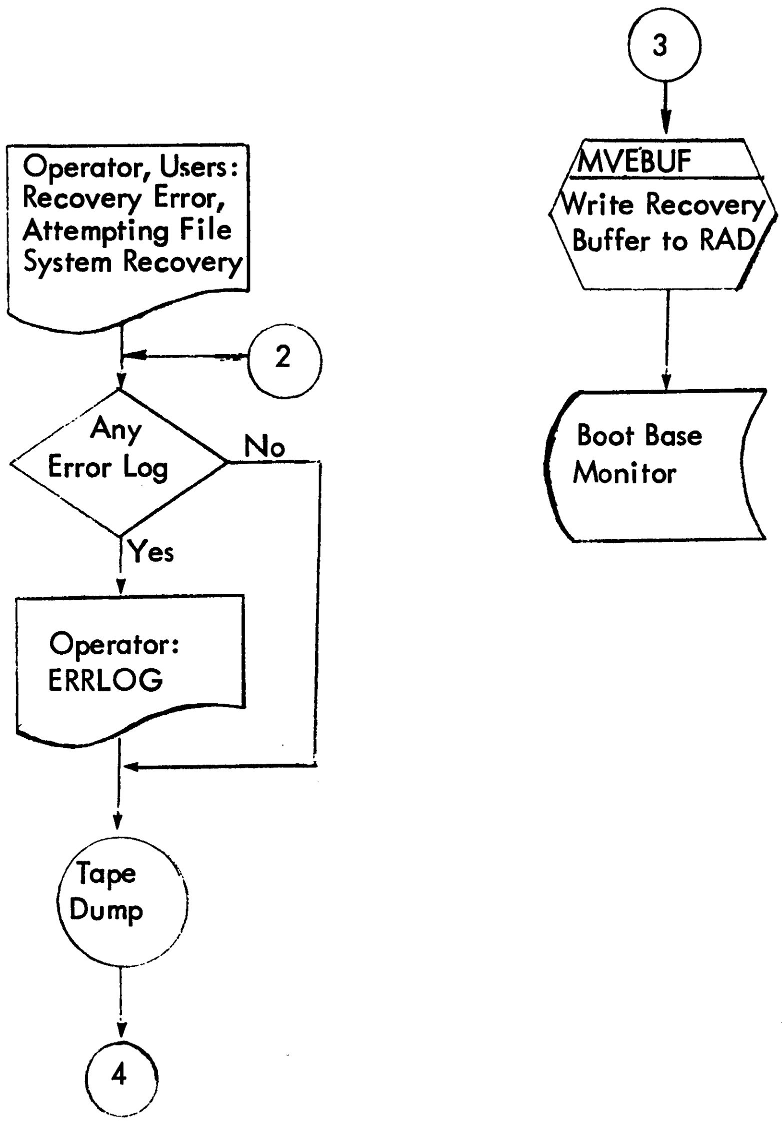
MVEBUF is called to put the recover buffer on the system swapping RAD at RCVRAD.

The monitor is booted from the system swapping RAD with the contence of X'2A' saved accross the boot.

If bad files are detected, the operator is asked if he wished to break service. If he says no, recovery continues as if no bad files were detected. If he says yes, core is dumped to tape, and the users are informed that file reconstruction will be attempted. The operator should then do a RAD boot and ask for a HGP reconstruction.

UTS TECHNICAL MANUAL




Operator, Users: Recovery Error, Attempting File System Recovery
2
Any Error Log
No
Yes
Operator: ERRLOG
Tape Dump
4
3
MVEBUF
Write Recovery Buffer to RAD
Boot Base Monitor

UTS TECHNICAL MANUAL

SUBROUTINES

Internal:

TYOUT	Type message to operator. Address of TEXTC Message in Register 4: BAL, 11 TYOUT
TYIN	Input one typed character from operator: BAL, 11 TYIN Returns input character in byte 3, register 0
COCOUT	Output message to all users. Address of TEXTC message in register 4: BAL, 11 COCOUT
HEXCVT	Convert hex to decimal for output. Hexdecimal input in register 0: BAL, 11 HEXCVT Returns hexadecimal numbers in EBCDIC in registers 2 and 3.
RBOUT	'RECOVERY SAYS - STAND-BY' to all dialed-up work stations. BAL, 11 RBOUT no arguments needed.

UTS TECHNICAL MANUAL

ID

TSTHGP

This module contains the following routines:

TSTHGP
SAVHGP
RRSG
RRBG
RRBG10
R:RNBG
R:RNCYL
CLSFILS
CLSFIL

UTS TECHNICAL MANUAL

ID

TSTHGP

PURPOSE

Check the consistency of the in core granule stacks and HGP's.

USAGE

BAL, 11 TSTHGP

INPUT

HGP	Head of Granule Pool Table
BUFSPD	Stack pointers for granule stacks
R:HGP	Copy of original HGP's in RECOVER overlay

OUTPUT

Message to operator if errors are detected.

DESCRIPTION

Checks the four stack pointer double words starting at BUFSPD for validity and overlap. If any inconsistencies are found a flag is set for SAVHGP to ignore the stacks and the message:

BUFSPD OVERLAP

is sent to the operator.

All disc addresses in the stacks are checked via R:CHKDA for validity. If a bad one is found, the flag for SAVHGP is set and the message:

BAD DA IN GRAN STACKS

is sent to the operator.

Next, the first 6 words of each HGP are compared with the copy of the HGPs built into recovery. If an error is found, the message

HGP MALFORMED

is sent to the operator.

SUBROUTINES

R:CHKDA	check validity of Disc Address
TYOUT	type message to operator

UTS TECHNICAL MANUAL

ID

SAVHGP

PURPOSE

Save the granules in the core resident granule stacks, restore Private Pack HGP's.

USAGE

BAL, 15 SAVHGP

INPUT

RCBUF	I/O buffer
HGP	Head of Granule Pool Table
BUFSPD	Stack pointers for granule stacks
AVRTBL	Table of Tape/Private Pack serial numbers and flags
DCT16	Text YYndd for devices

DESCRIPTION

The flag set by TSTHGP is checked and if non-zero, SAVHGP ignores the granule stacks. If the flag is zero, all disc addresses in the granule stacks are released via calls to RRBG or RRSG.

AVRTBL is searched for private disc packs that have been AVR'ed. When one is found, the VTOC of the volume is read and if the serial number matches, the HGP in core is written to the pack and the message:

Private Volume XXXX may have last space

is sent to the operator, where XXXX is the serial number on the pack.

If the serial number in AVRTBL does not match the one on the pack, the message:

Private Volume XXXX in AVRTAB doesn't match DPndd

is sent to the operator. XXXX is the serial number in AVRTBL and ndd is the device address of the pack whose serial number doesn't match.

UTS TECHNICAL MANUAL

SUBROUTINES

RRSG, RRBG	release granules
TYOUT	type message to operator
RDDISK	Read data from disc
WRDISK1	Write block to disc
R:CHKDA	Check disc address validity

UTS TECHNICAL MANUAL

ID

RRSG, RRBG

PURPOSE

Release granules back to the system

USAGE

BAL, 11 Entry point
R8 contains disc address to release

INPUTS

Disc address in R8

DESCRIPTION

These routines are used to free a file or symbiont granule respectively. The routine builds a core buffer of granules to be released (symbiont granules have bit 0 set) and when full, writes it to one of the granules being released. The routine always remembers the first disc address written and puts a forward link disc address at the start of each buffer written.

The entry point RRBG10 truncates the core buffer and writes it to the disc.

The format of the blocks of disc addresses on the disc is:

Word	Contents
0	0
1	number of valid entries
2	forward link disc address
3	disc address of first block
4-255	disc addresses to be released.

SUBROUTINES

R:CHKDA	check disc address validity
WRDISK	write block to disc

UTS TECHNICAL MANUAL

ID

RRBG10

PURPOSE

Truncate granule release buffer chain created by RRBG, RRSG.

USAGE

BAL, 11 RRBG10
R 15 contains 0

INPUT

R15 must contain 0

OUTPUT

last block of free granule chain is written to Disc.

DESCRIPTION

If no granules have been released, RELFDA is set to zero and the routine exits. Otherwise, the forward link of the current block is set to zero and the block is written to the disc.

UTS TECHNICAL MANUAL

ID

R:RNBG

PURPOSE

Release contiguous background (PFA) granules.

USAGE

BAL, 11 R:RNBG
R8 contains first disc address
R15 contains the number to release

INPUT

FDA in R8
number of granules in R15

OUTPUT

The disc addresses are released back to the system.

DESCRIPTION

This routine calls RRBG to release FDA, increments FDA by the number of sectors per granule from the HGP corresponding to this disc address. R15 is decremented and if not zero, the process is repeated.

SUBROUTINES

RRBG	release background granule
R:FNDHGP	find HGP for disc address

UTS TECHNICAL MANUAL

ID

R:RNCYL

PURPOSE

Release contiguous cylinders

USAGE

BAL, 11   R:RNCYL
R8 contains disc address of first cylinder
R15 contains number to release

OUTPUT

Cylinders are released to the system.

DESCRIPTION

FDA (in R8) is released via RRBG, FDA is incremented by 60, and R15 is decremented by 1. If R15 is not zero the process is repeated.

SUBROUTINES

RRBG	release background granule

UTS TECHNICAL MANUAL

ID

CLSFILS

PURPOSE

Close all open files for a user

USAGE

BAL, 11 CLSFILS
R5 has address of JIT

INPUT

JDCBVP	DCB Table virtual page
JBUPVP	Monitor buffer virtual page
J:DCBLINK	Pointer to User's DCB table.
CFU	Current File User Tables
RCBUF	Input Buffer (used for I/O buffer area)
JJITVD	JIT virtual page number.
J:OC	Address of DCB for operator's console.
HACCBD	(Half word address) current data record in blocking buffer.

OUTPUT

All files for the user are closed.

DESCRIPTION

J:DCBLINK in JIT points to the top of the next (only one allowed) DCB table. The FLINK pointed to by J:DCBLINK must point to the bottom of the DCB table and the bottom word must contain a zero. The entire DCB table and all DCB's must be in the next pages above the JIT. If these tests are not passed, the error exit is taken. The DCB address is extracted from each entry in the DCB table, verified, and passed to CLSFIL in register 6. The verification consists of verifying that the BLINKS and DCB addresses are between JIT and JBUPVP or equal to M:OC.

SUBROUTINES

CLSFIL	Close user DCB.

UTS TECHNICAL MANUAL

ID

CLSFIL

PURPOSE

Close a users DCB

USAGE

INPUT

CFU	Current File User tables
DCT 1	Device physical addresses

DESCRIPTION

If the DCB is assigned to device, is not open, or is a private file no further processing is required. If it is an open FILE DCB, the ACTIVE byte in the CFU is set to zero, and the file mode is tested to determine file processing for recovery.

IN Mode File:

If there are no current outstanding I/O operations for the file, no further processing is required. If there is an outstanding I/O operations, the current disc address in the DCB is verified via R:CHKDA. If the address is invalid, the bad file flag is set for recovery control and ZAPFIL is called to overwrite the file name to disallow further usage of the file. If the disk address is valid, no further processing is required.

INOUT Mode File:

The file's master index updated flag in the DCB is tested. If the index has been updated, it is written from core (DCB BUF2 buffer) to the disc address allocated for the file's master index. WRDISC1 is called to effect the I/O transfer and any errors encountered are ignored. A check is then made to determine whether the primary blocking buffer has been updated, the current block displacement is tested. If the displacement is non-zero, it is used to compute the number of sectors available for the file's record. The updated items are stored in the CFU table. If the buffer has not been updated, maximum buffer size is used to output the block instead of buffer size. WRDISC1 is then called to write the blocking buffer on the disc.

After transferring buffers to the disc or if no updates were indicated, the File Information Table is read into core. The following items are extracted from the CFU table and stored in the FIT core image:

a) available sector address for file's master index (SMI).
b) available sector address for file's data records (SREC).
c) the first disc address of file's master index (FDA).
d) sector left for the file's master index (SS MI).
e) sector left for the file's records (SSREC).
f) number of granules remaining in last cylinder allocated to the file(NGAVAL).
g) disc address of the last granule used out of the last cylinder allocated to the file (GAVAL).
h) disc address of top of perimid structure (TDA).
i) number of inserted MI blocks (SLIDES).
j) indicator that pyramid structure exists (O).

and the FIT is re-written onto the disc. The logic described above for an "IN" mode file is then executed.

OUT or OUTIN Mode File:

If the file is keyed, the first disc address of the file's master index (FDA) is used as the initial forward link to examine file status. If FDA = 0, processing is terminated. RRBG is called to release the file map granule pointed to by FDA.

A check is made to determine whether the granule is to be read into RCBUF by RDDISC. If it is already in core, it is moved from the monitor buffer to RCBUF. If errors are incurred attempting to input the index granule, processing is terminated. The forward link within the master index is used to either input and test the next granule or terminate index granule processing. Each granule is checked for proper FLINKs and BLINKs. Failure causes processing to be terminated. If there are significant data words, the disc address for each data block is used to release the data granules via RRBG. This process continues until all granules used for index and data for the file are released.

If the file is RANDOM, FDA from the CFU is used to release the granules. The number of granules to be released is picked up from RSTORE in the DCB and R:RNBG/R:RNCYL is called depending on whether the file is granule or cylinder allocated.

If the file is consecutive, the data blocks are read into RCBUF, via the FLINKs and all granules are checked via R:CHKDA and released via RRBG. If a bad disc address is found, processing is terminated.

LABELED TAPE

EOP (ending operation) is checked, and if not set to write, nothing is done. If the last operation was a write, the current blocking buffer is written to the tape if necessary. Then the series:

Tape mark
:EOF
Tape mark
:EOR
Tape mark
Tape mark

is written to the tape.

SUBROUTINES

Internal:

FINDFIL — Find a file by reading the account and file directories from the disk.
Address of account name in register 8;
Address of file name in register 9

```
BAL, 11        FINDFIL
```

Error return if not found or if BLINK/FLINK check failure.
If found, byte displacement of name in RCBUF is in register 3; disc address of file directory is in register 8; fil- directory in RCBUF.

GETFIT — GET File Information Table
Address of DCB in register 6:

```
BAL, 11        GETFIT
```

FIT disc address saved internally if the file information table is successfully input.

GETFLACN — Gets address of file name (R9) and address of account (R8) in variable parameters in DCB.

WRTFIT — Write File Information Table.
Used disc address saved internally (FITDA) by GETFIT or ZAPFIL:

```
BAL, 11        WRTFIT
```

Writes FIT image from RCBUF to disc if disc address is valid.

LOCCOD — Find parameter code in a specified buffer. Code for search placed in register 12; buffer address to search in register 7:

```
BAL, 11        LOCCOD
```

If code found, register 3 contains buffer pointer for the code word. Used by CLSFIL to find length entry (code of 13) in File Information Table.

ZAPNAM — Destroy name field for clobbered files. RCBUF name pointer in register 3:

```
BAL, R15       ZAPNAM
```

Stores EBCDIC blanks followed by an asterisk (*) over the requested name field. Used by ZAPFIL.

ZAPFIL — make a file inaccessible.

```
BAL, 11        ZAPFIL
```

R8 has address of account
R9 has address of file name

The internal routine FINDFIL is called to locate and read in the file directory for the requested file name for the account indicated. If the file is not found, the routine exits. If found, ZAPNAM is called to overwrite the file name in the file directory and the directory is re-written to the disk. The file information table is input via GETFIT. If not found, the routine exits. Otherwise, ZAPNAM is again called to overwrite the file name in the file information table, the table is re-written to the disc and the operator is informed that the file has been made inaccessible.

External:

Name	Purpose	Manual Reference
RDDISC	Read data from disc	OE.01
R:CHKDA	Check disc address validity	OE.03
WRDISC1	Write a specified buffer to disc	OE.01
TYOUT	Type message to operator	OA.01
RRSG	Release symbiont granule.	OB.03
R:RNBG	Release N background granules.	OB.05
R:RNCYL	Release N cylinders.	OB.06

UTS TECHNICAL MANUAL

ID

CYCUSR

This module is included in the monitor overlay RECOVER and contains the following routines:

CYCUSRS
CHKCFU
MAPSET
SVDNDEV
TSTUSR
RCVDMP
TAPDMP
SV1
MVEBUF
SYSLIM

ID

CYCUSRS - CYCLE users

PURPOSE

For each user: verify JIT, close all files, package all COOP files, and save the JIT for accounting during recovery second phase.

For the Symbiont Ghost job: write the Data (00), AJIT, and JIT pages back to their assigned spaces on swap RAD if Symbiont Ghost job was in core at time of recovery.

USAGE

```
BAL, 11        CYCUSRS
Error Return
Normal Return
```

INPUT

JBUPVP, FPMC, NPMC, HIGH, LOW - Possible page numbers

JOVVP	Virtual page numbers
SMUIS	Maximum number of users (EQU def)
UH:FLG	Bit 6 user in-core flag
UX:JIT	Physical page number of JIT, if in core
UH:JIT, UH:AJIT	Disc address of JIT and AJIT
JB:VLH	Virtual page link head
JB:LMAP	Allocated page map
JB:PPC, JX:PPH, JX:PPT, MX:PPUT	Physical page chain
JH:DA	Disc addresses for allocated pages
JJITVP+2, +25	Virtual page numbers for JIT's, DCB's, MI's
INITIAL & CORED	Page limits for users
JCMAP	Physical page map
S:SIP	Swap-in-progress flag
DID$IO	Swap I/O in progress flag
SB:OSUL	Out-swap user list
S:ISUM	In-swap user number
BGRCFU-LASTCFR	User CFU tables
JCLPA	Address of swap command list
JCLE	Length of swap command list
JBPCP	Page count of page procedure
JBPCDD	Page count of dynamic data
JBPCC	Page count of context
JAJ	Page address of AJIT

OUTPUT

The JIT's for the in-core users are written to their assigned spaces on swap RAD. Their disc addresses are placed in the recovery buffer by SV1.

If in core at time of recovery, write the Symbiont Ghost job's (RBBAT) data, AJIT, and JIT pages back to their assigned spaces on swap RAD.

The "JOB id PARTIALLY COMPLETED" message is typed on the OC by PARCOM for running batch jobs.

Cryptic error messages are noted on the OC by ABNXT.

ERRORS

The ABNXT routine is used to type cryptic messages on the OC. They relate to problems encountered while verifying user information. They are meaningful only to systems programmers who study the ANLZ dump following the crash. The format is: user number; location in recovery; message. The messages and their meanings are:

BAD JIT	TSTACK check failed
PHY PG MAP	Failure in JX:PPH, MX:PPUT, or JXPPT
DCB TABLES	DCB Table flinks or DCB addresses are bad
JIT DA	UH:AJIT contains invalid disc address
SWAP DA	JH:DA table contains invalid disc addresses
AJIT DA	UH:AJIT contains invalid disc address
CONTEXT DA	JH:DA of out-of core user contains invalid disc addresses for context area
USR CNTL T	Any failures encountered by TSTUSR
BAD MAP	Failure encountered by MAPSET
READ CHECK	Swapper read-check failed for context area of out of core user.
SYMBT LOST	Bad disk address in RBBAT JIT

CHKCFU subroutine verifies any CFU which is flagged active. Any failure here implies that the CFU tables have been clobbered and recovery is impossible. CHKCFU takes its error return which causes CYCUSRS to execute its error return to recovery control.

SUBROUTINES

External:

TYOUT type on OC

HEXCVT	convert hex for printing
RRAD1	read RAD
CKRAD	check RAD address
CLSFILS	close user files
COOPFILS	package and close COOP files
SV1	save JIT address in Recovery Buffer
WRRAD1	write RAD

Internal:

MAPSET	verify user map and set map above INITIAL L1,6 JIT + JCMAP BAL,11 MAPSET Error causes return to find the next user
ABNXT	type error message BAL,11 ABNXT
CHKCFU	Close any files whose DCB's were clobbered BAL,11 CHKCFU error return normal return
PARCOM	type partially completed message BAL,12 PARCOM

DESCRIPTION

MAPSET

For each in-core user, MAPSET is called to verify the map in the user's JIT, move the map into the map registers and go mapped. The verification consists of checking each physical page in the map between JJITVP and JBUPVP to be not equal to JJITVP, not lower than LOW nor higher than HIGH unless they equal FPMC or NPMC.

CHKCFU

CHKCFU is called after all files have been closed. If a JIT or DCB tables have been clobbered, it is possible that some files might not have been closed. Any CFU which is active is checked to see if it might be clobbered. These checks consist of:

1. If FDA is non-zero it must be a valid disc address.
2. If SSMI (byte 0) is non-zero, SMI must be a valid disc address.
3. If CCBD (byte 0) is non-zero, SREC must be a valid disc address.

Failure of any of these checks causes recovery to be deemed impossible.

Processing continues as a function of the mode of the open file.

INPUT or not active files are ignored

INOUT file names are placed in the recovery buffer so that RECOVER 2 can copy them.

ACCPGE

ACCPGE is called in with register zero pointing to a limit doubleword that contains a lower and upper virtual page number. The subroutine searches through the user's virtual page chain in JB:LMAP and forms a list of all virtual pages, and their disk addresses, that occur between the specified limits.

CYCUSR

CYCUSR must examine each user, locate all DCB's, close all open files and save the JIT for accounting information. Validity checking is performed on JIT's, swapping RAD addresses and physical page numbers.

All users currently in core are processed first. As each is processed, the map is set by MAPSET to correspond to the user's map. Before processing users that are disk resident, the map is turned off. Each users context is then read in from the disk and processed in the same manner as the core resident users. Users in the process of being swapped in or out are flagged either entirely in or entirely out according to the following tests: If the swap-in-progress flag (S:SIP) is zero or the number of users being swapped out (DID#10) is zero, no users are in transition and no flags are changed. If DID#10 is negative, all the users in the out swap list are flagged as in-core. If DID$10 is positive, the in-swap user (S:ISUN) is flagged out-of-core. 1

SMUIS establishes the length of the user control tables. The tables are scanned twice-once to locate in-core users and the second time to locate out-of-core users. Let N stand for a user number. UB:US(N) is the user's state; if zero or 2, N is not a user. $UH:FLG(N)_6$ is in-core or out-of-core flag.

In-core users:

Each JIT is validity checked by checking the user's TSTACK. The contents of the TSTACK pointer (Location TSTACK) minus the space used (TSTACK+1, bits 16-32) must equal the address TSTACK+1.

Each swapping RAD address in JH:DA is validity checked by CKRAD.

Each physical page number corresponding to a virtual page in JB:LMAP is validity checked by being >INITIAL↓9 and <CORED↓9.

The number of pages in the user's physical page chain in MX:PPUT is checked against JB:PPC. The tail of the user's physical page chain in MX:PPUT must equal JX:PPT.

Failure of any of these checks causes CYCUSR to type an error message on the operator's console, skip this user, and continue to the next user.

A pointer to the JIT is passed to CLSFILS subroutine, which closes all open user DCBs. A pointer to the JIT is also passed to COOPFILS, which tracks down and packages all COOP files. Subroutine PARCOM is called to type partially completed messages if the user was a batch job. The disk address of the JIT is then saved in the recovery buffer so that RECOVERY2 can update the accounting log. Each in-core JIT is written to its assigned RAD space.

As the in-core users are processed, a check is made to determine if the Symbiont Ghost job (RBBAT) is in-core. If not, no processing is needed. When in-core, the data (00 protection) pages and AJIT page are written back to swap RAD. The page addresses are found in the command list and the disk addresses and AJIT page addresses are found in the JIT. These pages must be on swap storage so they can be read by RECOVER2.

Out-of-core users:

After all in-core users are processed, the map is turned off and each out-of-core user is examined. Out-of-core users' pages are read into physical memory as though mapped one-to-one, virtual to physical. The user's JIT is read into JJITVP, and his AJIT, if he had one, is read into JJITVP+1. A list of the user's context virtual pages and their disk addresses is formed by searching through the JB:LMAP chain. Each page in this list is read in. All the disk addresses in the JH:DA table are verified. As was done for the in-core users, the CLSFILS, COOPFILS, and PARCOM routines are called to close up the files and type the partially completed messages. Finally, each out-of-core user's JIT disk address is added to the table of users' JIT disk addresses in the recovery buffer.

After all users have been processed, CHKCFU is called to find any files remaining open. If the CFUs have been clobbered, recovery is judged impossible and the error exit is taken.

UTS TECHNICAL MANUAL

ID

SVDNDEV - Save down devices

PURPOSE

Loop through DCT3 and determine which devices are down. Save the down device list so that it can be merged with the DCT3 by recovery second phase.

USAGE

BAL, 11 SVDNDEV

INPUT

Bit 2 of DCT3 - when set, indicates the device is marked down.
DCTSIZ is the length of the DCTs.

OUTPUT

One word is moved to the recovery buffer for each device which is flagged down. Byte 0 contains code X'06', which signifies to second phase that this is a down device. Byte 3 contains the index into the DCTs for the down device.

Future: If diagnostic has a bit to indicate temporary-down or diagnostic-down; when that bit is set, the device will not be remembered as down by SVDNDEV.

UTS TECHNICAL MANUAL

ID

TSTUSR - Verify user control tables

PURPOSE

Check all monitor controlling user tables and type an error message if tables have been clobbered.

USAGE

BAL, 11 TSTUSR
normal return

INPUT

UH:FLG	-	User flags
UX:JIT	-	JIT physical page number
UH:JIT	-	JIT disc address
UH:AJIT	-	AJIT disc address
UB:US	-	User state
SNSTS	-	Number of states
INITIAL	-	Lowest user virtual address
CORED	-	Highest user virtual address

ERROR

Failure of any of the tests causes a cryptic message "USR CNTL T" to be typed on the OC by ABNXT as described in section OD.

SUBROUTINES

ABNXT - Error message typing

DESCRIPTION

TSTUSR checks as many user tables as can be checked without reading JITs in from the RAD.

TESTS:
If UB:US is zero or 2, there is no assigned user.
UH:FLG: 1) if bit 15 is set, 6 must also be set because if a user is in core and

ready to run, his JIT must also be in core.

2) Bits 0 and 1 must be 0 because they are unused.

3) If bit 9 is set, bit 5 must also be set because if a debugger is in control, a debugger must be associated with the user.

UX:JIT: If UH:FLG bit 6 is set, then the JIT is in core and the JIT physical page number in UX:JIT must be greater than the page number of INITIAL (right shift 9) and less than CORED.

UH:JIT and UH:AJIT are either 0 or legitimate swapping RAD addresses.

UB:US must be less than or equal to SNSTS, the number of possible states.
If UB:US is zero, UH:JIT and UH:ID must also be zero.
If UB:US is 2, UH:FLG must be non-zero.

UTS TECHNICAL MANUAL

ID

RCVDMP - Recovery dump to RAD

PURPOSE

If RECOVERY decides that the file system is probably okay and just the swapping RAD and core need be initialized, all of core is dumped on to a reserved area of the swapping RAD for later analysis.

USAGE

BAL, 11 RCVDMP

INPUT

RCVRAD	defines the first available address on the swapping RAD to be used for the dump
RCVRDSZ	defines the number of available granules
CORED	defines the top of physical memory
RCVRCNT	the number of recoveries executed since last start of the system

OUTPUT

Beginning with page 0, each page of physical core is written to the swapping RAD beginning with granule RCVRAD+2. The value of RCVRAD is saved in the RECOVERY buffer by SV1. The next available granule after the dump and the RCVRCNT are also saved by SV1. The address originally contained in RCVRAD is saved in TRCVRAD. This is the address where the recovery buffer will later be placed.

ERRORS

If the amount of available RAD space is less than that needed to save the core space, the tape dump is called and RAD addresses are not remembered by SV1.

SUBROUTINES

External:	WRRAD1	Write to RAD
	SV1	move a word to RECOVERY buffer
	TAPDMP	tape dump
Internal:	INCRDA	Compute next granule disc address (system RAD)

DESCRIPTION

CORE is converted to a page number and the space required is compared with the RAD space available (RCVRDSZ). If space needed exceeds space available, TAPDMP is called. Otherwise, RCVRAD+2 is entered into the RECOVERY buffer.

Each physical page is then written to the swapping RAD via calls to WRRAD1. The next available granule address is computed by INCRDA.

Following system reboot and core initialization, but before swapping RAD initialization, the dump space on RAD is written as a keyed file on the file RAD.

The keys for core pages are | 03 | 00 | 00 | Page # |

The user JITs are added to the keyed file with keys of | 03 | 00 | user # | 00 |.
This file is described in section KB.11.

UTS TECHNICAL MANUAL

ID

TAPDMP Tape dump

PURPOSE

TAPDMP performs a full core physical page dump on to tape. Following the system reload, the crash analysis program can read the dump tape for analysis processing.

USAGE

```
BAL, 11    TAPDMP
```

OUTPUT

The dump tape is a labeled tape and each logical record is one page of core memory.
The label sentinel is :LBL RCVT.
The identification sentinel is :ACN :SYS bbbb Date of the crash in format from DATE and DATE+1.
Tape mark
The beginning of file sentinel is

```
:BOF
01   00   02   02
 7    T    A    P       file name is TAPDUMP
 D    U    M    P
09   01   00   02       ORG is conseq., VOL is 1
01   00   00   00
00   00   00   00
```

Tape Mark

Control record

Each data record is 512 words.

The tape mark record is

```
tape mark
:EOF
3*4                previous block size
```

Tape mark

End of reel sentinel terminates the dump

tape mark
:EOR
NULL

Tape mark
Tape mark

ERRORS

If the requested unit is not supplied by the operator or is not a valid unit id, the tape dump is not performed.

SUBROUTINES

External:

SV1	Save tape identification in RECOVERY buffer
TYOUT	request to operator

Internal:

GETDEV	read operator response and convert to device number
REWTAP	Rewind tape
TAPBAD	Request new tape and start again
WRTAP	write tape
WREOF	write tape mark

DESCRIPTION

TAPDMP runs unmapped so that physical pages are dumped sequentially. Only one tape dump is performed per crash. The operator is asked to mount a scratch tape and key in the device identification. This identification is saved in the RECOVERY buffer.

UTS TECHNICAL MANUAL

ID

RECOVERY Buffer, SV1 and MVEBUF

PURPOSE

The RECOVERY buffer is used to save information which is used after the system is rebooted in order to restore any information which should not be lost.

USAGE

The subroutine SV1 is called to add a word to the buffer.

```
        LW, 15      word
        BAL, 11     SV1
```

Move the buffer to swap RAD

```
        BAL, 11     MVEBUF
```

SUBROUTINES

RDRAD1	Read RAD
WRRAD1	Write RAD

DATA BASE

Each item of information within the RECOVERY buffer is followed by an identification word. The identification consists of an id code in byte 0 and the word count of the information item in bytes 2 and 3. The items are not necessarily in the buffer in id order.

id code	probable count	Item
01	64 +5+2	SGRAN, BGRAN, CURGRAN, FGRAN1, CURBUF + contents of CURBUF and 2 preceding words
02	19	Date/Time and administrative message
03	3	Initial and final RCVRAD from RCVDMP and RCVRCNT
04	size	size of RECOVERY buffer
05	1	dump tape identification
06	value	down device number
07	value	number of locked symbiont devices + 1, RCVRCNT, and SGB.
08	logged on users	user #, swap index, seek address (8, 8, 16)

DATA BASE (cont'd)

id code	probable count	Item
09	149	Partition limits, system limits.
0A	≤10	Two words of account and up to 8 words of file name in TEXTC format
0B	0	HGP reconstruction required
0C	≤10	Same as for 0A above except file's granules will not be released.
0D	1	First disc address of granule stack for release
0E	8+4*#of RB's genned	Symbiont tables and Remote batch tables.
0F	4	Write symbiont ghost recovery files
10	≥61(SGCBUFSZ)	Symbiont ghost communication buffer
11	1	Symbiont ghost error word.

buffer layout

. . . .	DATE	DATE+1	TIME	02 3	04	SIZE

DESCRIPTION

The RECOVERY buffer is a RES of 1024 words in the data area of RECOVERY. When RECOVERY processing is completed, the size of the recovery buffer and its contents are moved to the first 2 granules of RCVRAD on the system RAD. Since the JIT's are used by the second phase, any JITs occupying the area of the system swapping RAD where GHOST1 will be swapped must be moved to some other space on the RAD. MVEBUF accomplishes this by investigating the JIT addresses saved in the Recovery buffer and calling RDRAD1 and WRRAD1.

The data (00 protection), and AJIT pages of the symbiont ghost job (RBBAT) also must not be in the area of the system swapping RAD where GHOST1 will be swapped. These pages are moved at the same time as user JITs.

UTS TECHNICAL MANUAL

ID

SYSLIM - Save system limits

PURPOSE

During execution of UTS the system limits can be modified by the CONTROL program. It, therefore, behooves recovery to save the system limits so that they can be restored by recovery second phase. The system id is saved for uniqueness.

USAGE

```
        BAL, 11             SYSLIM
```

INPUT

S:GUAIS 14 words

this includes:

S:GUAIS
S:BUAIS
SL:TB
SL:UB
SL:QUAN
SL:QMIN
SL:BB
SL:IOC
SL:IOPC
CO:LTO
SL:LTO
SL:OLTO
CO:ITO
SL:ITO
SL:OITO
S:OUAIS
SL:PI

SL:BTIME 14 words
this includes:

SL:BTIME
SL:BLO
SL:BOP
SL:BDO
SL:BUO
SL:BT
SL:BFP
SL:BPS
SL:BTS
SL:BIP
SL:BC
SL:BF
SL:BSP

SL:OTIME 14 words
this includes:

SL:OTIME
SL:OLO
SL:OPO
SL:ODO
SL:OUO
SL:OT
SL:OFP
SL:OPS
SL:OTS
SL:OIP
SL:OC
SL:OF
SL:OSP

SL:USID	1 word
SL:SQUAN	1 word
SL:OXMF	4 words

this includes:

SL:OXMF
SL:BXMF
SL:OIMF
SL:BIMF

SL:7T	7 words	
SL:9T	7 words	
SL:SP	7 words	
SL:C	7 words	
PLH:TL		LPART+1 entries saved
PLH:TU		
PLH:QN		
PLH:TOL		
PLH:FLG		
PL:MAX		
PL:MIN		

OUTPUT

The above mentioned system limits Recovery buffer by SV1 with an id code of X'09'.

SUBROUTINES

SV1 – Save one word in the Recovery buffer.

UTS TECHNICAL MANUAL

ID

SYMFILS - Symbiont Files
SAVSYM contains the following subroutines

SAVSYM
SAVSYM1
SYMFILS
COOPFILS

PURPOSE

Save the symbiont ghost communication buffer in the recovery buffer. Save selected symbiont tables and selected remote batch tables in the recovery buffer. Release current imput cooperative files. Package current active symbiont files from card readers and printers both local and remote. Current output cooperative files are truncated with a record containing 'LAST RECORD SUPPLIED BY RESTART'.

UTS TECHNICAL MANUAL

ID

SAVSYM - Save symbiont information in recovery.

PURPOSE

Restore UH:JIT, UH:AJIT, and UB:SWAPI entries for the Symbiont Ghost job (RBBAT) from ALLOCAT data on the system swap RAD. Move the symbiont communication buffer from monitor data to a buffer in recover.

USAGE

```
        BAL, 11      SAVSYM
    Return to BAL+1
```

INPUT

MB:SDI	-	Swap DCT index
UH:JIT	-	JIT Disk Address
SGCBUFSZ	-	Symbiont Ghost communication buffer size
SGCHD	-	Symbiont Ghost communication buffer
RCVRGFCI	-	Recovery Code in SG communication buffer entry.

OUTPUT

UH:JIT	-	JIT Disk address of SG job (RBBAT)
UB:SWAPI	-	Swap index of SG job (RBBAT)
UH:AJIT	-	AJIT Disk address of SG job (RBBAT)

DESCRIPTION

At system start up ALLOCAT saves in its data on the system swap RAD the UH:JIT, UH:AJIT, and UB:SWAPI entries for the symbiont ghost job. At recovery these values are restored, in case core has been clobbered.

The monitor data symbiont communication buffer is moved to a buffer in recovery. This is done because additional entries are added by recovery, these entries cannot clobber monitor data.

ID

SAVSYM1 - Put symbiont tables and remote batch tables in recovery buffer.

USAGE

```
        BAL, 11  SAVSYM1
    Return to BAL+1
```

INPUT

SSIG	- Symbiont Control Character
RCVRCNT	- Recover count
SGB	- Number of available symbiont granules.
S:BFIS	- Batch files in system
BL:IFS	- Number of input symbiont file spaces remaining
BL:OFS	- Number of output symbiont file spaces remaining
GI:SDA	- Starting disk address of selected symbiont input file
GIB:UN	- User number of selected symbiont input file
RBLIMSZ	- Number of Remote Batch entries
RB:XFLG	- If NZ RBX keyed in
RB:FLAG	- Remote Batch flags
RBB:LPZ	- Max line length for Remote Batch printer
RBB:CPZ	- Max record length for Remote Batch punch
RBD:WSN	- Remote Batch work station name
RBB:ID	- Remote Batch ID

OUTPUT

All of the preceeding tables are moved in the recovery buffer.

DESCRIPTION

Any locked symbiont device (SSIG) is saved, the recover count cell (RCVRCNT) is saved, and the number of available symbiont granules (SGB) is saved in the recovery buffer by calling SV1. Next a selected group of symbiont tables and remote batch tables are saved by calling SV1. The symbiont ghost communication buffer from recovery is moved to the recovery buffer.

ID

RELSYMS

PURPOSE

Release a Symbiont file.

USAGE

BAL, 11 RELSYMS
R8 contains first disc address

INPUT

None other than R8

OUTPUT

All granules in the symbiont file are released to the system via calls to RRSG.

DESCRIPTION

Starting with the disc address provided, RELSYMS reads the first word of the block and uses it as the next block address in the chain. Each granule thus located is released by RRSG. An invalid disc address or zero, signifies the end of the granule chain.

SUBROUTINES

RDDISK - Read a data block from the DISC
RRSG - Recovery Release Symbiont Granule

UTS TECHNICAL MANUAL

ID

SYMFILS

PURPOSE

Package input and output symbiont files.

USAGE

BAL, 11 SYMFILS
Return to BAL+1

INPUT

SNDDX	Number of symbiont devices, DCT index
SSTAT	Symbiont status
SCNTXT	Symbiont context Block address
SYMX	Symbiont - input/output flag.
SCJOBX	Count of records read through symbiont
SCBINFOX	Current buffer address
SCDCBX+8	Current disc address
BL:IFS	Number of input symbiont file spaces remaining
SCDEVTYP	Symbiont device type
SRET	Displacement into symbiont cummunication buffer
SCBESTDA	Best disc address
SCSVDGI	Remote Batch information (RRI, RBID, SYSID)

OUTPUT

Entries in the recovery symbiont ghost communication buffer.

DESCRIPTION

Each active symbiont device, both local and remote, is processed. An entry is made in the recovery symbiont ghost communication buffer for each symbiont file at the time of the crash.

INPUT SYMBIONT FILES

If a card has been read, a FLINK of zero is put into the current buffer and that buffer is written to file storage. An add Input File entry is put into the recovery symbiont ghost communication buffer.

OUTPUT SYMBIONT FILES

An add Output File Partial entry is put into the recovery symbiont ghost communication buffer for each output symbiont file. The disk address at SCBESTDA in the Context block is used as the first disk address of the file. If the output symbiont file has not been connected to the communication buffer entry that entry is unlinked from the in use chain.

UTS TECHNICAL

ID

COOPFILS

PURPOSE

Close all cooperative files

USAGE

BAL, 11 COOPFILS
R 5 contains JIT address

INPUT

JITUSCDX	Start of user context Data Buffer chain
CDLINK	Link to next context data buffer
SNDDX	Number of symbiont devices
SCDCBX	Current disc address
SCJOBX	Count of reconds written through COOP
R:DCT4	Type of device
JRBID	Remote Batch ID
BAABC	Priority of job
SBSECTS	Sectors/Buffer

OUTPUT

Entry in recovery symbiont ghost communication buffer for each output cooperative file.

INTERNAL SUBROUTINE

LNKSGCBF - Link a communication buffer entry to the in use chain.

DESCRIPTION

Close all COOP files associated with the Job Information Tabl e. The beginning address of the used context data buffer chain (JITUSCDX) is used to start processing the input-output files opened by a cooperative. The link-to-next- context block is always examined and the routine exits when the end-of-chain is indicated.

Output Cooperative File:

The DCB area of the context block is used to extract the CDA and WRDISK is called to write a trailer record as the last record of the file. The trailer record corresponds to the Output Symbiont File Buffer format given in Section VI. 04.

If errors are detected by WRDISK/R:CHKDA no further COOP processing is done for this JIT. Each output cooperative file is truncated with a buffer that contains the 'LAST RECORD SUPPLIED BY RESTART' message. This buffer is written to the current disk address (SCDCBX+8) with a FLINK of zero and the proper BLINK (SCDCBX+9). An entry is placed in the symbiont ghost communication buffer for each output cooperative file processed.

Input Cooperative File:

Beginning with the disc address for the current I/O operation (selected from the DCB area of the context block), the disc granules allocated for the file are released by RELSYMS. The disk address is then adjusted using the number of sectors/buffer as input to RRSG to free the granules containing the partial job input.

UTS TECHNICAL MANUAL

ID

RCVRIO - Recovery disc I/O routines

This module contains the following subroutines

RDDISK
RDRAD
RDDISK1
RDRAD1
WRDISK
WRRAD
WRDISK1
WRRAD1
CKRAD
R:CHKDA
R:DSCCVT
R:FNDHGP

ID

RDDISK, RDRAD, RDDISK1, RDRAD1, WRDISK, WRRAD, WRDISK1, WRRAD1

PURPOSE

To read and write to RADS and Disks without using the monitor I/O Routines.

USAGE

The entry points whose names end in DISK require a disc address, i.e., DCT index and relative sector number, whereas the entry points whose names end in RAD require a SEEK ADDRESS. The SEEK ADDRESS may take one of two formats:

```
USER#, 0,  SEEK        (8, 8, 16)
0, SWAPI,  SEEK        (8, 8, 16)
BAL, 11
Error Return
Normal Return
```

Calling arguments:

RDDISK/RDRAD

R15	Size in Bytes
R8	Disc/Seek address
RCBUF	is the buffer address

RDDISK1/RDRAD1

R3	Size in bytes
R4	Buffer address
R8	Disc/Seek address

WRDISK/WRRAD

R4	Buffer address
R8	Disk/Seek address
Size	=1024 bytes

WRDISK1/WRRAD1

R3	Size in bytes
R4	Buffer address
R8	Disk/Seek address

UTS TECHNICAL MANUAL

INPUT

RCBUF	512 word buffer RDDISK/RDRAD
DCT4	Device types
DCT1	Physical device addresses
SMUIS	Maximum number users
UB:SWAPI	Swap device table index
MB:GAM5	Swap device table
M:GATLIM	Swap device table
MB:GAM6	Swap device table
M:GASLIM	Swap device table
MB:SDI	DCT index of swap device
MAPFLG	Flag provided by caller, 0 = > not mapped, ≠ 0 = > mapped.
JOVVP	Overlay virtual page address
JB:CMAP	JIT map

OUTPUT

The number of bytes requested is read or written from/to the proper device.

SUBROUTINES

Name	Purpose	Manual Reference
R:CHKDA	Check validity of disc address	
R:DSCCVT	Convert disc address to seek address	
CKRAD	Check validity of seek address	

DESCRIPTION

The buffer address is set in R3 and the buffer size is set in R4. Then a copy of the proper command list (read or write) is moved to a work area.

The Disc or Seek address is checked via a call to the appropriate subroutine. If invalid, the error return is taken. If the area the I/O is to be done into/out of is mapped, the command list is changed to contain the physical addresses and the command list is changed to data chain if necessary. If RCVRIO is in GHOST1, then it is mapped itself so the command list is moved so that it is wholely contained in one page and the physical address of the command list is calculated. Next, the SIO is done. If a no device recognition response is received, the routine takes the error exit. Otherwise, a TIO loop is entered until the I/O is completed.

UTS TECHNICAL MANUAL

ID

CKRAD - Check Validity of Disc Address

PURPOSE

CKRAD checks the validity of a specified SEEK address to see if it is a valid disc pack, file RAD, or swapping RAD address.

USAGE

Swapping:

LW, 8	disc address in form USER#, 0, SEEK address (8, 8, 16)
LI, 11	error return
BAL, 0	CKRAD
normal return	

INPUT

DCTSIZ	Device control table length
DCT1	Device control table containing physical device addresses
DCT4	Device control table containing index into the type mnemonic table (TYPNME)
SMUIS	Equals maximum number of users in system
UB:SWAPI	RAD parameter tables index table (by user number)
MB:GAM5	Shift count to right-justify track (by UB:SWAPI)
M:GATLIM	Number of tracks on RAD (by UB:SWAPI)
MB:GAM6	Sector mask (by UB:SWAPI)
M:GASLIM	Number of sectors per track (by UB:SWAPI)
MB:SDI	RAD DCT index (by UB:SWAPI)

OUTPUT

If no error is detected, 11 is incremented by 1.

DESCRIPTION

The user number field (byte zero) is checked for validity, i.e., less than or equal to SMUIS. If it is invalid, the subroutine exits to the address contained in R11. If the user number is valid, the track/sector address contained in the right half-word of R8 is checked for validity. If invalid, the error return is taken; otherwise, the subroutine increments the error return in R11 by one and exits normally. The track number is valid if it is less than or equal to the appropriate entry in M:GATLIM. The sector number is valid if it is less than or equal to the appropriate entry in M:GASLIM. When byte zero contains zero, byte one contains the swap index (UB:SWAPI).

UTS TECHNICAL MANUAL

ID

R:CHKDA

PURPOSE

Check the validity of a disk address.

USAGE

BAL, 11	R:CHKDA
R8	has disc address to check on return;
CC=0	if Disc address is bad
CC=F	if Disc address is not bad

INPUT

HGP Tables

SUBROUTINES

Name	Purpose	Manual Reference
R:FNDHGP	Finds HGP for disc address in R8	OE.04

OUTPUT

Sets CC as above.

DESCRIPTION

Calls R:FNDHGP to find the HGP for the disc address. If none is found, the error exit is taken. The device type is found from the number of sectors/track in the HGP. The relative sector number in the disc address is then compared with the maximum physically on the device. If the sector number is too large, the error exit is taken. Otherwise the routine takes the normal exit.

UTS TECHNICAL MANUAL

ID

R:FNDHGP

PURPOSE

To locate the HGP for a given disc address.

USAGE

BAL, 3 R:FNDHGP
with disc address in R8
Returns HGP address or 0 in R7
Sets CC via LW,7 7 before exiting.

INPUT

HGP Header Granule Pools

SUBROUTINES

None

OUTPUT

Address of proper HGP

DESCRIPTION

Searches the HGP tables for one with the same DCT index as the DCT index in the disc address input to the routine.

UTS TECHNICAL MANUAL

ID

R:DSCCVT – disc convert

PURPOSE

To convert a disc address to a seek address.

USAGE

BAL, 11	R:DSCCVT
R8	has disc address

on Return:

R8	has seek address or zero (left justified if necessary)
CC	set by LW, 8 8 before return

INPUT

HGP Tables

SUBROUTINES

Name	Purpose	Manual Reference
R:FNDHGP	Finds HGP for disc address	OE.04

OUTPUT

SEEK ADDRESS

DESCRIPTION

The routine calls R:FNDHGP to locate the HGP for the disc address. Once the HGP is located, the routine has the necessary information to calculate the Seek address and it proceeds to do so.

UTS TECHNICAL MANUAL

ID

RCVDEF

PURPOSE

This module's only purpose is to define the address of recovery's copy of the resident HGP's.

DESCRIPTION

This module is placed in the LOCCT for recovery after MONSTK and just before IOTABLE. Thus it can define the address of recovery's HGP by the loader evaluatable expression:

```
R:HGP  EQU  $ + (HGP-IOTABLE)
```

where HGP and IOTABLE are defs in the monitor root.

ID

RECOVER2 - Recovery second phase

PURPOSE

The second phase of recovery is executed after the monitor is rebooted but before the shared processors are reinitialized to the system swapping RAD. The system parameters from the Recovery buffer are merged in with the system tables and the full core dump from the system RAD is written as a keyed file. The accounting information from the JITs is added to the accounting log and any TFILEs are released. Any update files which were not properly closed are copied. The HGP tables are reconstructed if necessary.

USAGE

```
BAL, 11    RECOVER2
```

RECOVER2 is called by GHOST1 for every start up except a cold boot from the P0 tape. Normal exit is to BAL+2, error exit is to BAL+1 (indicating that HGP reconstruction was unsuccessful or the recovery tables were clobbered.)

INPUT

The Recovery buffer is located in the first 2 granules of RCVRAD on the system RAD. The full core RAD dump is located at granule RCVRAD+2 and the users' JITs are located from data saved on the recovery buffer.

OUTPUT

The full core dump is written as a keyed file by the name MONDMPi. :SYS where i is the EBCDIC value of the recovery count (RCVRCNT) module 8. Each record of the file is a page of core with the key | 3 | 0 | 0 | Page # | or a user's JIT with the key | 3 | 0 | user number | 0 |.

The system tables which are restored include the DATE, TIME, administrative message, Error log and pointers, recovery count, down device list, symbiont tables, system limits, and the number of unscheduled batch jobs.

INTERACTION

CAL1, 8 - Get a common page for recovery buffer
CAL1, 8 - Get a virtual page for JIT
M:OPEN, M:CLOSE, M:WRITE, M:READ - for MONDMPi and scarey files
M:MESSAGE - Error message to OC
M:SNAP - Snap the clobbered recovery tables.
COCOUT - Output message to all users
TYOUT - Output message to operator console
TYWAIT - wait for OC I/O to complete
ACCNTSUM- LOGOFF accounting routine
RDRAD1 - Read a RAD granule
CKRAD - Verify disc address
INCRDA - Compute next swapping RAD granule address (within core dump)
MAILBOX - Warning to users
JULIAN - Date Conversion
HGPRECON- Reconstruct HGP tables
RDDISK1 - Read disc

ERRORS

If the Recovery buffer contains inconsistencies in the id codes or word counts, recovery cannot continue, the system should be rebooted, and the operator is informed:

RECOVERY TABLES CLOBBERED - UNABLE TO CONTINUE

While copying files which were improperly closed during Recover, file system errors '75' are related to the operator as "ERR xxxx ACT count FILE filename". The third digit of the error code is T for an output file, or an I for an input file.

If HGPRECON is called and the reconstruction attempt fails, the operator is notified:

RECOVERY IMPOSSIBLE-SYSTEM UP FOR FILE SAVE ONLY

and the users are similarly informed:

RECOVERY IMPOSSIBLE-MUST RELOAD FROM FILL TAPES

This error results in RECOVER2 exiting to the BAL+1, causing GHOST1 to inhibit the COC and batch job initialization. RECOVER2 sets location X'2A' to a one, inhibiting the analyze ghost.

Error Message output by HGPRECON are described in Section OH.

DESCRIPTION

The Recovery buffer is read into dynamic common from the swapping RAD. The id codes are used to branch to a transfer vector in order to perform the appropriate processing.

code = 1, the error log buffer and pointers are restored from the Recovery buffer. This includes SGRAN, BGRAN, CURGRAN, FGRAN1, CURBUF and its contents.

code = 2, the cells DATE, DATE+1 and TIME are restored. The administrative message is restored to COCMESS.

code = 3, the recovery dump is read from the system RAD and written by the file system as a keyed file MONDMPi in account :SYS. RCVRAD and RCVRCNT are restored.

code = 4, is always the last word of the last granule since it includes the number of the significant words in the buffer.

code = 5, is currently a NOP as it tells that a tape dump was taken rather than a swapping RAD dump. If there is no MONDMP corresponding to recovery count, ANALYZE will request that the tape be mounted

code = 6, the down device indicated in the code word is marked down in DCT3.

code = 7, the locked symbiont device list plus RCVRCNT and SGB

code = 8, the users' JITs are read into virtual page ='1B000' (DEFed as P:JIT) and are added to the dump file MONDMPi. The RATE file is read and each JIT is then passed to the ACCNTSUM subroutine of LOGOFF to do the accounting and TFILE releasing.

code = 9, the system limits are restored:

SL:ONCB 1 word
S:GUAIS 14 words
this includes:

S:GUAIS
S:BUAIS
SL:TB
SL:UB
SL:QUAN
SL:QMIN
SL:BB
SL:IOC
SL:IOPC
CO:LTO
SL:LTO
SL:OLTO
CO:ITO
SL:ITO
SL:OITO
S:OUAIS
SL:PI

SL:BTIME 14 words

this includes:

SL:BTIME
SL:BLO
SL:BPO
SL:BDO
SL:BUO
SL:BT
SL:BFP
SL:BPS
SL:BTS
SL:BIP
SL:BC
SL:BF
SL:BSP

UTS TECHNICAL MANUAL

code = 9(cont'd)

SL:OTIME 14 words

this includes:

SL:OTIME
SL:OLO
SL:OPO
SL:ODO
SL:OUO
SL:OT
SL:OFP
SL:OPS
SL:OTS
SL:OIP
SL:OC
SL:OF
SL:OSP

SL:ONCB	1 word
S:USID	1 word
SL:SQUAN	1 word
SL:OXMF	4 words

this includes:

SL:OXMF
SL:BXMF
SL:OIMF
SL:BIMF

SL:7T	7 words	words +1, +3, +5 set to 0
SL:9T	7 words	
SL:SP	7 words	
SL:C	7 words	
PLH:TL		LPART+1 entries saved
PLH:TU		
PLH:QN		
PLH:TOL		
PLH:FLG (ANDed with PL:JIF)		
PL:MAX		
PL:MIN		

S:MBSF is set to 0
PL:CHG is set to X'FFFF'

code = A, the files identified by name-account are copied first to a temporary file and then copied again to a file of the original name. The original file is released so that it cannot cause futher trouble to the file system. Any file system errors which occur during the copy are typed for the operator. Each file copied results in a message in the user's MAILBOX file of "FILE COPIED BY RECOVERY filename".

code = B, the HGP tables in ALLOCAT or the CUF's were clobbered. The ALLOCAT HGP tables must be reconstructed. (The subroutine is described in Section KB. 12).

code = C, same as code A except files granules are not released with the original file.

code = D, the disc address in the granule chain are released via calls to RBG or RSG.

code = E, the symbiont and Remote Batch tables are restored.

The tables restored are:

S:BFIS	- Batch files in system
BL:IFS	- Number of input symbiont file spaces remaining
BL:OFS	- Number of output symbiont file spaces remaining
GI:SDA	- Starting disk address of selected symbiont input file
GIB:UN	- User number of selected symbiont input file
RB:XFLG	- If NZ RBX keyed in
RB:FLAG	- Remote Batch flags
RBB:LPZ	- Max line length for Remote Batch printer
RBB:CPZ	- Max record length for Remote Batch punch
RBD:WSN	- Remote Batch work station name
RBB:ID	- Remote Batch ID

code = F, the Symbiont Ghost data pages both, static data and dynamic data, are written to the :RBBRVR. :SYS file.

code = 10, the Symbiont Ghost communication buffer record is written to the :RBBRVR. :SYS file.

code = 11, the Symbiont Ghost error communication word is stared at SGCHD+3 in monitor data.

UTS TECHNICAL MANUAL

ID

HGPRECON

PURPOSE

Reconstruct the HGP tables if they are destroyed on the system RAD.

USAGE

BAL, 11 HGPRECON
normal return is + 2
error return is + 1

INPUTS

ALLOCAT DATA	old system HGP's
DCT1	Physical device addresses
HGP	Monitor copy of HGP heads
MB:SDI	SWAPPER DCT INDEX
GIB:UN	Qued uses number in partion table
GI:SDA	Qued symbiont file first disc address
SGCHD	Symbiont Ghost communication buffer

OUTPUT

The system HGP's are reconstructed and re-written into ALLOCAT's data area on the swap RAD.

DESCRIPTION

Upon entry, HGPRECON performs the following initialization steps:

1. Obtains 4 buffers:
   a) BUF1 256 words
   b) BUF2 256 words
   c) BUF3 512 words
   d) DATA BUF1 512 words
   via M:GVP cal's
2. Obtains enough core via an M:GP cal to construct 4 working copies of the system HGP's. The copies are called:
   a) HGPAD account directory granules
   b) HGPFD file directory granules
   c) HGPCUR current file's granules
   d) HGPSUM sum of all previous file's granules
   note that HGPRECON has a master copy of the System HGP's built into it
3. Reads the first 8 words of ALLOCAT's data and extracts the account directory's first Disc address.

HGPRECON next starts the first of its 4 phases. The 4 phases are:

1. Check and allocate all account directory granules
2. Check and allocate all file directory and FIT granules
3. Check and allocate all file granules
4. Check and allocate all symbiont granules

Phase 1 (Account directory checks)

Each sector of the Account directory is read. The Blink and Flink are checked for validity and the Blink is compared with last Flink. If an error occurs, it is noted on the line printer and the AD is truncated at that point. Each granule is marked in use in the HGPCUR copy of the HGP's. See sample output for normal Phase 1 output. When the account directory is finished HGPCUR is merged into HGPAD and HGPCUR is re-initialized.

Phase 2-File directory checks

The first sector of the account directory is re-read. The disc address of the start of the lst file directory is extracted and read. The disc address for the fit of the lst file is extracted from the file directory and read. The file name in File Directory is compared to the file name in the FIT, if they don't compare, a diagnostic message is output. The granules are marked used in CURHGP and the process is repeated for each file in the file directory and each file directory in the account directory. As each file in the file directory is processed, the sequence of the names is checked.

If a file name is out of order it is removed from the directory and an appropriate diagnostic message is printed.

At the end of each file directory, HGPCUR is merged into HGPFD and re-initialized. Any overlaps are noted by an appropriate diagnostic.

Phase 3 -File integrity checks

The account directory first sector is read again and the first accounts file directory disc address is extracted and read. The FIT disc address for the first file is extracted and read. From the FIT, the file's lst disc address is available. The level zero sector chain for the file is read, Flinks, Blanks and all data granule addresses are validated and marked used in HGPCUR. If the chain is broken at any point, LDA is extracted from the FIT and the level zero chain is read backwards until either the two ends meet or the chain is broken again. The two ends of the chain are then made to point to each other.

If the file has an upper level structure, the links of the entire upper level are checked and any errors noted. The upper level is not repaired since the logic in the system that uses the upper level is prepared to handle error conditions.

At the end of each file, HGPCUR is merged into HGPSUM and compared with HGPAD and HGPFD. If any dual allocation exists a diagnostic message is printed. HGPCUR is then reinitialized.

This process is repeated for each file in the file directory and each file directory in the account directory.

Phase 4 - symbiont checks

The starting disc address of each symbiont file is obtained from: the Symbiont Ghost static data record, the Symbiont Ghost communication buffer record, and the table GI:SDA. The Blink and Flink are checked for validity and if OK, the Blink is marked used in the allocation table. The Flink is then read and the process repeated until the end of the file is reached. If an error is detected in the chain, the Flink of the last good block is set to zero and the block is re-written thus truncating the file at that point. This process is repeated for each symbiont SDA found.

The reconstruction process is now complete. The 4 working copies of the HGP's are now merged into 1 and written to the Data area of ALLOCAT, starting at sector 8 of the first swapping rad in the system.

ERROR MESSAGES

**NAME SEQUENCE ERROR IN FILE DIRECTORY

File Directory entries were found that were not in alphabetic order. The entries out of order are deleted.

EOF BIT SET - AND FLINK NON - ZERO

Self explanatory. No correction is made.

BAD LINKAGE

The file being processed had a link failure. The link is fixed if possible, otherwise it is left as it is.

INVALID DATA ADDRESS

The disk address of a data block was found to be invalid. No correction.

BAD FREE SECTOR POOR LINKAGE

Self explanatory. The Free Sector Pool is truncated at that point.

INVALID FREE SECTOR # POOL ENTRY: xxxxxx

Self explanatory. xxxxxx is the entry, it is deleted.

DUAL ALLOCATION IN FREE SECTOR POOL

Granules were found in the free sector pool that were also allocated to previous files. No correction.

BAD F.I.T. ADDRESS

An illegal FIT disc address was found in the file directory. The file is deleted.

DUAL ALLOCATION

Self explanatory. No correction.

DATA GRANULE ALLOCATION ERROR

A data disc address was found that was either illegal or dually allocated. No correction.

MASTER INDEX ALLOCATION ERROR

Same as above except the disc address is for a master index.

****************FDA BAD FOR ABOVE FILE -- FILE DELETED ******

Self explanatory

DISC I/O FAILURE

Physical device error. No correction.

RECONSTRUCTION FAILURE - USE BACKUP TAPE

Self explanatory.

ACCOUNT DIRECTORY FIRST SECTOR BLINK IS NON-ZERO-RECONSTRUCTION HALTED

Self explanatory

ACNCFU CONTAINS INVALID DISC ADDRESS

The Account Directory FDA extracted from ALLOCAT Data is invalid. ALL files are lost.

****LINK FAILURE IN ACCOUNT DIRECTORY

Self explanatory. The Account Directory is truncated at the point of the error.

****LINK FAILURE IN FILE DIRECTORY

Self explanatory. The File Directory is truncated at the point of the error.

PYRAMID FLINK ERROR:LEVEL

LOCATION	LINK
AAAA	DDDD

Self explanatory. LL is the level, AAAA is the Disc address of the granule that contains the bad link DDDD.

PYRAMID LOCATION, ERROR: TDA = xxxxxx

TDA (xxxxxx) from the FIT is an illegal disc address. No correction.

NEXT LEVEL ADDRESS INVALID: xxxx

The disc address xxxx in the pyramid structure is invalid. The disc address was supposed to be a pointer to the next lower level.

RANDOM FILE ADDRESS ERROR: xxxx

FDA (xxxx) from the FIT of a Random file is illegal. The file is deleted.

FILE NAME DOES NOT CORRESPOND TO THE FIT.

The file name in the file directory is not the same as the file name in the FIT pointed to by the entry. The file is deleted.

xxxxxxxxxxxxxxxx ACCOUNT AND FILE DIRECTORIES CONFLICT.

Dual allocation between Account and File directories. No correction.

BLINK ERROR	LOCATION
bbbbbb	LLLLL

The Blink (bbbbbb) in the granule at LLLLLL is an illegal disc address.

FLINK ERROR	LOCATION
fffff	LLLLL

The Flink (ffffff) in the granule at LLLLLL is an illegal disc address. The linkage is fixed if possible.

*************FILE CONFLICTS WITH ACCOUNT DIRECTORY

Self explanatory. No correction.

********FILE CONFLICTS WITH FILE DIRECTORIES

Self explanatory. No correction.

********FILE CONFLICTS WITH PREVIOUS FILES

Self explanatory. No correction.

SAMPLE OUTPUT

Phase 1

Account Directory Check

SECTOR #	LOCATION	BLINK	FLINK	HGP
001	000B007C	00000000	00000000	D3E8

Phase 2

File Directory and File Information Table Checks

ACCOUNT	FD LOCATION	HGP	BLINK	FLINK	FSP
:SYS	000B007E	D3E8	00000000	000B0180	000B0583
	000B0180	D3E8	000B007E	000B0009	
	000B0009	D3E8	000B0180	000B039B	
	000B039B	D3E8	000B0009	000B001C	
	000B001C	D3E8	000B039B	000B0634	
	000B0634	D3E8	000B001C	000B016D	
	000B016D	D3E8	000B0634	000B01DA	
	000B01DA	D3E8	000B016D	000B0392	
	000B0392	D3E8	000B01DA	000B0235	
	000B0235	D3E8	000B0392	000B03[illegible]	
	000B038F	D3E8	000B0235	00000000	

UTS TECHNICAL MANUAL

Phase 3
File Integrity Checks

Account: :SYS

FIT DA	FILE NAME	PASSWORD	SIZE	FDA	LDA	ORG	KEYM	DEV#	TDA
000B0004	$::COBLIB		0002	000B0006	000B0006	KEY	03	2F0	
000B0005	$::LIB		0002	000B0002	000B0002	KEY	03	2F0	
000B006C	$::REST		0002	000B0000	000B0000	KEY	03	2F0	
000B0393	:ACCTLG		000D	0015158A	0015286A	CON	03	2F0	00000076
000B0183	:BACKUP		0002	00000000	00000000	KEY	08	2F0	
000B0079	:BLIB		0013	0015005C	00150038	CON	03	2F0	0000014E
000B0181	:BREC		0003	000B0202	000B0202	KEY	0B	2F0	
000B006D	:DIC		000C	000B006A	000B005D	KEY	0B	2F0	000B005A
000B0056	:JO		0004	000B0058	000B0058	KEY	0B	2F0	
000B0057	:J1		0017	000B0054	000B0054	KEY	0B	2F0	
000B0024	:LIB		0023	000B0052	000B0028	KEY	0F	2F0	000B0026
000B0025	:PO		0008	000B0022	000B0022	KEY	0B	2F0	
000B009E	:POO		0021	000B0020	000B0021	KEY	0F	2F0	
000B009F	:P1		0004	000B009C	000B009C	KEY	0B	2F0	
000B0098	:P11		000F	000B009A	000B009A	KEY	0F	2F0	
000B0214	:USERS		003B	000B05B0	000B0686	KEY	15	2F0	000B0684
000B0099	ALLOCAT		001D	000B0096	000B0097	KEY	0F	2F0	
000B00C6	ANLZ		0021	000B00C8	000B00C9	KEY	0F	2F0	
000B00C7	APL		003C	000B00C4	000B00D2	KEY	0F	2F0	000B00C5
000B00DA	BASIC		0022	000B00D0	000B00D1	KEY	0F	2F0	
000B00DB	BASICLOCCT		0002	000B00D8	000B00D8	KEY	03	2F0	
000B00C2	BATCH		0007	000B00CE	000B00CE	KEY	0F	2F0	
000B00C3	BJMAP		0002	000B00C0	000B00C0	KEY	03	2F0	
000B00D6	BLOAD		001F	000B0094	000B00CA	KEY	0F	2F0	000B0095
000B00D7	BLOAD2		001F	000B00D4	000B00DC	KEY	0F	2F0	000B00D5
000B00E8	BPASS2		0027	000B00E0	000B00EA	KEY	0F	2F0	000B00E1
000B00E9	BPASS3		000A	000B00E6	000B00E6	KEY	0B	2F0	
000B00F4	BPASS32		000A	000B00E4	000B00E4	KEY	0F	2F0	

XEROX Publication Revision Sheet

MAY, 1973

CORRECTIONS TO XEROX UNIVERSAL TIME-SHARING SYSTEM (UTS) INITIALIZATION AND RECOVERY TECHNICAL MANUAL (SIGMA 6/7/9 Computers)

PUBLICATION NO. 90 19 92A, FEBRUARY, 1973

The attached pages contain changes for the D00 version of UTS. Pages in the A edition of the manual that are to be replaced are: 1-8, 14-18, 44, 45, 47, 48, 50, and 51. New pages to be inserted are 2-1, 6-1, and 8-1.

These changes will be incorporated into the next edition of the manual.

Revision bars in the margins of replacement pages identify changes. Pages without the publication number 90 19 92A-1(4/73) at the bottom are included only as backup pages.

ID

INITIAL

PURPOSE

INITIAL prepares a fresh copy of the monitor (from either the system device or tape) for operation.

USAGE

INITIAL is branched to by the tape or system device bootstrap after a fresh copy of the monitor root has been read into memory.

OUTPUT

A running monitor with two users (ALLOCAT and GHOST1), the monitor JIT and XDELTA in their respective physical memory areas, which have been removed from the monitor's page pool.

INTERACTION

GETHGP:	used to restore XDELTA on RAD or disk pack boot. (Section NG)
MONINIT:	used to read from tape, patch, and write to the system device the unlabeled portion of a system tape, monitor root, and system device bootstrap. (Section NB)
T:GJOBSTRT	used to start up ALLOCAT and GHOST1. (Section CC)
T:SE:	used to exit and schedule ALLOCAT and GHOST1. (Section EA)
T:SGRNU	used to release swap space for GHOST1. (Section FA.01.08)

DATA BASES

XPSDS is a 32 word table that INITIAL moves to the trap and interrupt locations X'40' to X'5F'.

IOXPSD	contains the XPSD that belongs in the I/O interrupt location (X'5C') but which is not in XPSDS because BOOTSUBR uses interrupts to signal completion of RAD or disk pack I/O.
CORXPSD	contains the XPSD that is temporarily stored in the non-allowed operation trap location while INITIAL is determining the physical memory size of the computer.

UTS TECHNICAL MANUAL

SUBROUTINE

LMA performs three functions:

1. Loads the memory writelocks with (01) for all pages except those which contain any part of the monitor between GETHGP and MONINIT (procedure area), which are given a lock of (11).
2. Loads the memory map so that each page is mapped into itself.
3. Loads the memory access protection registers with a code of (11) for all pages except page 0, for which the code is (10).

LMA is called by means of a BAL, 11 and destroys registers 0 through 6.

RESTRICTIONS

INITIAL must be entered master, unmapped.

DESCRIPTION

The type of computer and mode of SYSGEN is determined. If the system was generated for a large memory Sigma 9 but is running on a Sigma 7, an error message is typed on the OC device and initialization is aborted.

INITIAL then moves the monitor JIT from its loaded location in the root to J:JIT. The external interrupt locations (X'60' to MONORG) are then zeroed to prevent confusion of automatic LOGON. Then the trap and interrupt locations X'40' through X'5F' are initialized from XPSDS and the writelock, memory map, and access protection registers are loaded by LMA. If low order halfword of X'2A' is nonzero (tape boot) the I/O interrupt is armed and enabled for BOOTSBR's RAD or disk pack I/O and a BAL to MONINIT reads and patches the system tape and sets up the monitor area on the system device. Otherwise, GETHGP is used to read in XDELTA. Then the pages containing the monitor root, the monitor JIT, XDELTA, and non-existent memory are removed from the monitor's free page pool (MX:PPUT) and its head, tail and count set appropriately (M:FPPH, M:FPPT, M:FPPC). Nonexistent memory is found by trying to access the last word of decreasing memory sizes until a trap does not occur. LOW is set to the lowest non-monitor page number. HIGH is set to the number of the highest existent page. SL:CORE and S:PCORE are set up to reflect the number of pages found to be available to users. If an extra half page of monitor root exists above JITLOC, it becomes on SPOOL buffer. Then SL:CORE granules of swap area are released, starting at the end of the system-swap area (PSA) of the system RAD, so that GHOST1's swap granules will not conflict with shared processor granules. The DCT index of the system RAD or disk pack and the relative sector number of the last granule released is saved in BOOTSBAND.

RESTRICTION

The end of PHASEC (PAGES) is the SEGNAM table origin so it must be loaded after PHASEB and any GHOST1 module that is exexecuted after PHASEB.

DESCRIPTION

All of memory is requested after releasing all of GHOST1's data the upper limit is saved in COREND, and the IO buffer address is calculated from R14 (end of SEGNAM table). BITOTM is called to copy files from tape to disc. If there are no modifications, BITOTM returns to PASS0's calling routine. Otherwise, PHASEC's MODTMTM loop is entered.

MODTMTM opens M:TM with update and next file options (and a zero file name the first time). If the next file cannot be opened and it is not because it is synonymous, PHASEC frees memory and returns to PASS0's caller. If the file is not keyed, M:TM is closed and the next file is tried. The HEAD record is read to determine the TREE size and whether the module is ABS or paged. The TREE is read on top of the HEAD and the NXTREENT loop searches the TREE for segments which have modifications.

In NXTREENT, each segment name is moved to the CCPL as though it had just been picked off a GENMD card. Then a BAL to SEGSERCH determines whether modifications exist. If they do, READSEG reads in the segment and one BAL to MODIFY for each chained Change Description Table in the SEGNAM table does the modifications. If the module is paged, MODIFY BALs to PAGEMOD in PHASEC with what to store where in registers. PAGEMOD stores the patch if the affected page is in core. Otherwise it rewrites the page that is in core (if there is one), reads the right one, and patches it.

UTS TECHNICAL MANUAL

ID

SYSMAK, SYSMAK 1

PURPOSE

SYSMAK - To copy shared processors (except GHOST1) to the swap device from files in the system account.

SYSMAK 1 - To copy the specified shared processor to the swap device from the input file.

USAGE

BAL, 11 SYSMAK all registers clobbered (except 11)

BAL, 11 SYSMAK 1 all registers clobbered (except 11)
Reg6 = address of buffer to use
Reg7 = processor number
M:EI open to the processor load module file.

INPUT

J:DLL	to release GHOST1 data pages
MB:SDI	swap device DCT index
P:NAME	names of shared processors
RCVRAD	relative sector number of shared processor area on RAD
MB:GAM4	to reserve space (RCVRDSZ)
HIGH	to calculate required RCVRDSZ
BOOTSBAND	end of processor area on RAD
PB:C#, PB:DC#	cylinder number of area on device (SYSMAK 1)
PH:DDA, PH:PDA	disc address of area on RAD (SYSMAK 1)
SPSIZE	size of processor replacement slots (in pages)

OUTPUT

P:NAME	processor overlay names
P:AC	access image for special shared processors (double word table)
P:SA	start address
P:TCB	TCB address
PB:C#	cylinder part of procedure disk address for Disk Pack
PB:DC#	cylinder part of data and DCB disk address for Disk Pack
PB:DCBSZ	DCB size (pages)

PB:DSZ	Data size	
PB:HVA	Next page above procedure (including largest overlay)	
PB:LNK	Next overlay processor index	
PB:PSZ	Procedure size	
PB:PVA	Procedure bias (page)	
PH:DDA	DCB's - Data disc address	
PH:PDA	Procedure disc address	
RCVRDSZ	Size of RECOVER's RAD dump area	SYSMAK only
M:SBAND	Lowest valid swapping track and sector	SYSMAK only

INTERACTION

PUTHGP	to save the file structure	
M:OPEN	to open files (JIT's, Processors)	
M:CLOSE	to close files	
M:GP, M:FP	to get and release memory	SYSMAK only
T:SGRNU	to release swapping RAD granules	SYSMAK only
T:SGAJIT	to swipe swapping RAD granules	SYSMAK only
M:TYPE	to type messages	
SCREECH	to crash	
M:READ	to read files	
GMB	to get a monitor buffer	
RMB	to release a monitor buffer	
NEWQ	to write to the swap device	

SUBROUTINES

SET$PROC$TAB moves information from the HEAD (for P:SA, P:TCB) and TREE records to all processors tables except PB:LNK, PH:PDA. P:AC is set as follows (bits per page):

> Bits 0 to 1 = 00 if PB:DSZi =1; 01 if PB:DSZi = 0; read, write, execute access.
> Bits 2 to 2n = 1 = 01 where PB:PSZi = n (n pages of procedure). Read, execute.
> Bits 2n + 2 to 63 = 11 no access.

RADWRITE writes to the swap device. It is called with a BAL,8 with (9) containing the word displacement from the buffer (input for SYSMAK 1 or the page boundary above SYSMAK) of the data to be written (must be on a page boundary), and (15) containing the first byte and the appropriate disk address table address (either PH:PDA or PH:DDA) in the rest.

PH:PDA or PH:DDA is first set up as computed from SENSW. The relative sector number is specified in SENSW and each write of one page is followed by incrementing SENSW and decrementing #PGSLEFT. Prior to writing procedure or data and DCBs, a check is made to insure that the number of pages to write is greater than #PGSLEFT so that each will be contained on one cylinder if system device is a Disk Pack. If the swapping device is a Disk Pack, PH:C# or PB:DC#, the cylinder part of the disk address, is set up at this time; whichever is appropriate is determined by the disk address table address in register 6.

The seek address is specified in SENSW. The seek address of the last granule written is in DISCLOC and the next granule to write to is in SENSW. To write to the RAD NEWQ is called with end action. The end action routine, placed in a monitor buffer so that the routine may be executed unmapped as required, saves the type of complete. The completion code is checked for errors when NEWQ returns. If a swapping error occurs or if the write is to the address in BOOTSBAND, a message is typed and SYSMAK skips to the next processor or if SYSMAK 1, exits. RADWRITE clobbers all registers except 4, 5, 6, and 7.

ERRORS

TYPE is entered with (14) pointing to a message (TEXTC). If SYSMAK 1, this is converted to an error code in (5) and SYSMAK 1 exits. It appends the message to the current processor name and types this. It then skips to the next processor. It is entered on IO errors and abnormals or if a RADWRITE tries to write above BOOTSBAND.

The messages are:

"UNREADABLE"	if an I/O error occurs opening or reading the file
"NOT IN SYSTEM"	as indicated when trying to open the file
"OVERFLOWS"	if there is no more space on the system device
"SWAP IO ERROR"	if an error occurs writing to the swap device
"ILLEGAL LM"	if the load module is illegal
"CANT OVERLAY"	if no overlay slots are available.

RESTRICTIONS

SYSMAK or SYSMAK 1 must run master, mapped. Since SYSMAK uses all memory above itself as a buffer, it must be loaded after any module in GHOST1 which must execute after it, and must run with the special JIT access flag. NSWAP flag in UH:FLG2 is set by INITIAL to prevent swapping out GHOST1.

DESCRIPTION

SYSMAK 1 performs a subset of the SYSMAK logic. The entry and exit points for SYSMAK 1 are noted below.

SYSMAK initialization consists of: saving return in TSTACK, getting all available memory, calculating and saving in BUFFER the page boundary above SYSMAK, and moving the starting relative sector number from RCVRAD to SENSW. #PGSLEFT is set up with a number greater than the possible processor area (if RAD), or with S:CYLSZ, the number of granules in a cylinder (if Disk Pack).

The processor index is initialized to MAXOVLY (the first shared processor which is not a monitor overlay). An attempt is made to open the file named in P:NAME unless it is M:DUMLM, in which case the relative sector number from SENSW is put in PH:DDA and SENSW is bumped SPSIZE granules. If I/O errors or abnormals occur opening or reading, M:DUMLM is put into the P:NAME entry to create a replacement slot.

SYSMAK 1 enters at this point with a flag indicating SYSMAK 1 and the processor number in register 7, having saved in BUFFER the buffer address provided in register 6, and setting up in SENSW the processor's data relative sector number from PH:DDA. M:EI has been opened to the appropriate file by whoever called SYSMAK 1. If Disk Pack, #PGSLEFT is initialized by computation from SENSW and S:CYLSZ; otherwise, it is set with a number larger than the size of the processor area.

The HEAD record is read into SYSMAK's data area HEADER, the TREE is read into the IO buffer (pointed to by BUFFER) and a BAL to SET$PROC$TAB fills in most of the tables. The next relative sector number (from SENSW) is put in PH:DDA and the DCB's and data records, if present, are read from the file and written to RAD. The procedure record is read and written with its disc address going into PH:PDA. If this is a SYSMAK 1 call and the processor is a monitor overlay, then the size and disc address of the data is put into the processor's pure procedure size and disc address tables instead of the tables for data since the data in the load module of a monitor overlay is really pure procedure. If the TREE size is 12 words, this processor is complete, and the processor is complete, and the processor number is incremented. The calling routine is now returned to if this is SYSMAK 1.

If the processor number is now that of GHOST1 or ALLOCAT, it is incremented again. If the processor is TEL or CCI, SENSW is bumped up 4 granules to leave space for them to grow is they are later replaced. If M:EI is open, an M:CLOSE closes it. If there are still processors to process, the processor loop loops. If the processor index is PNAMEND, all the processors are complete, the disc address in SENSW is set into the 1st entry of PH:DDA (and PB:DC# if Disk Pack). The required RCVRDSZ is calculated (HIGH+3). If current RCVRDSZ is not adequate, it is set to the required size and the end RAD address of the dump area is calculated. If the swap device is a Disc Pack, RCVRDSZ is set to the next higher cylinder boundary. The user cylinder table UB:C# is set up. The user slots assigned to cylinders outside the PSA area are taken out of the system by removing them from state 0 and outputting a message to the operator indicating how many users are now in the system. If this is greater than BOOTSBAND, RCVRDSZ is decremented until it isn't. Then all available swapping granules above BOOTSBAND are swiped and their addresses saved in a table. M:SBAND is set to zero and the swiped granules and all granules between BOOTSBAND and the end of the dump area are released. Then memory is freed, the return address is restored, and SYSMAK exits. If the TREE is bigger than 12, the processor's overlay segments must be put on RAD. The last partial page (unless it is a full page) of the root segments procedure (still in memory) is moved down to the second page of the IO buffer (the first page has the TREE in it). Overlay segments will be read in at the end of the root portion of this page and the whole page will be written to the system device as the first page of the overlay so that pages on the system device will correspond to pages of execution memory. PB:PSZ and PB:HVA are decremented for the root segment, and DISCLOC is moved to SENSW to back up the disk address one page, since the last page of the root is no longer part of the root. Then a loop is entered which processes the TREE from the last segment to the second one. For each segment, the name is moved from the TREE to P:NAME, in any zero entry in the processor overlay portion (NAMEND to PPROCS). If no such zero entry exists, SYSMAK aborts this processor through the I/O error logic. The segments procedure size (including the last page of the root) is put into PB:PSZ. Each segment is linked to the previous one by storing the previous index (initially zero) in PB:LNK. PB:PVA is set for the overlays from PB:HVA of the root and the largest segment's size is saved. Only the procedure record of the segment is read from the file and written to the system device, with its disc address going into PH:PDA. When the TREE runs out, the last overlay index used is put in PB:LNK for the root, PB:HVA for the root is increased by the largest segment's size, and the main processor loop is re-entered after its TREE size check.

UTS TECHNICAL MANUAL

ID

CYCUSR

This module is included in the monitor overlay RECOVER and contains the following routines:

CYCUSRS
CHKCFU
MAPSET
SVDNDEV
TSTUSR
RCVDMP
TAPDMP
SV1
MVEBUF
SYSLIM

ID

CYCUSRS - CYCLE users

PURPOSE

For each user: verify JIT, close all files, package all COOP files, and save the JIT for accounting during recovery second phase.

For the Symbiont Ghost job: write the Data (00), AJIT, and JIT pages back to their assigned spaces on swap RAD if Symbiont Ghost job was in core at time of recovery.

USAGE

```
BAL, 11        CYCUSRS
Error Return
Normal Return
```

INPUT

JBUPVP, FPMC, NPMC, HIGH, LOW - Possible page numbers

JOVVP	Virtual page numbers
SMUIS	Maximum number of users (EQU def)
UH:FLG	Bit 6 user in-core flag
UX:JIT	Physical page number of JIT, if in core
UH:JIT, UH:AJIT	Disc address of JIT and AJIT
JB:VLH	Virtual page link head
JB:LMAP	Allocated page map
JB:PPC, JX:PPH, JX:PPT, MX:PPUT	Physical page chain
JH:DA	Disc addresses for allocated pages
JJITVP+2, +25	Virtual page numbers for JIT's, DCB's, MI's
INITIAL & CORED	Page limits for users
JCMAP	Physical page map
S:SIP	Swap-in-progress flag
DID$IO	Swap I/O in progress flag
SB:OSUL	Out-swap user list
S:ISUM	In-swap user number
BGRCFU-LASTCFR	User CFU tables
JCLPA	Address of swap command list
JCLE	Length of swap command list
JBPCP	Page count of page procedure
JBPCDD	Page count of dynamic data
JBPCC	Page count of context
JAJ	Page address of AJIT

OUTPUT

The JIT's for the in-core users are written to their assigned spaces on swap RAD. Their disc addresses are placed in the recovery buffer by SV1.

If in core at time of recovery, write the Symbiont Ghost job's (RBBAT) data, AJIT, and JIT pages back to their assigned spaces on swap RAD.

The "JOB id PARTIALLY COMPLETED" message is typed on the OC by PARCOM for running batch jobs.

Cryptic error messages are noted on the OC by ABNXT.

ERRORS

The ABNXT routine is used to type cryptic messages on the OC. They relate to problems encountered while verifying user information. They are meaningful only to systems programmers who study the ANLZ dump following the crash. The format is: user number; location in recovery; message. The messages and their meanings are:

BAD JIT	TSTACK check failed
PHY PG MAP	Failure in JX:PPH, MX:PPUT, or JXPPT
DCB TABLES	DCB Table flinks or DCB addresses are bad
JIT DA	UH:AJIT contains invalid disc address
SWAP DA	JH:DA table contains invalid disc addresses
AJIT DA	UH:AJIT contains invalid disc address
CONTEXT DA	JH:DA of out-of core user contains invalid disc addresses for context area
USR CNTL T	Any failures encountered by TSTUSR
BAD MAP	Failure encountered by MAPSET
READ CHECK	Swapper read-check failed for context area of out of core user.
SYMBT LOST	Bad disk address in RBBAT JIT

CHKCFU subroutine verifies any CFU which is flagged active. Any failure here implies that the CFU tables have been clobbered and recovery is impossible. CHKCFU takes its error return which causes CYCUSRS to execute its error return to recovery control.

SUBROUTINES

External:

TYOUT type on OC

HEXCVT	convert hex for printing
RRAD1	read RAD
CKRAD	check RAD address
CLSFILS	close user files
COOPFILS	package and close COOP files
SV1	save JIT address in Recovery Buffer
WRRAD1	write RAD

Internal:

MAPSET	verify user map and set map above INITIAL L1,6 JIT + JCMAP BAL,11 MAPSET Error causes return to find the next user
ABNXT	type error message BAL,11 ABNXT
CHKCFU	Close any files whose DCB's were clobbered BAL,11 CHKCFU error return normal return
PARCOM	type partially completed message BAL,12 PARCOM

DESCRIPTION

MAPSET

For each in-core user, MAPSET is called to verify the map in the user's JIT, move the map into the map registers and go mapped. The verification consists of checking each physical page in the map between JJITVP and JBUPVP to be not equal to JJITVP, not lower than LOW nor higher than HIGH unless they equal FPMC or NPMC.

CHKCFU

CHKCFU is called after all files have been closed. If a JIT or DCB tables have been clobbered, it is possible that some files might not have been closed. Any CFU which is active is checked to see if it might be clobbered. These checks consist of:

1. If FDA is non-zero it must be a valid disc address.
2. If SSMI (byte 0) is non-zero, SMI must be a valid disc address.
3. If CCBD (byte 0) is non-zero, SREC must be a valid disc address.

Failure of any of these checks causes recovery to be deemed impossible.

Processing continues as a function of the mode of the open file.

INPUT or not active files are ignored

INOUT file names are placed in the recovery buffer so that RECOVER 2 can copy them.

ACCPGE

ACCPGE is called in with register zero pointing to a limit doubleword that contains a lower and upper virtual page number. The subroutine searches through the user's virtual page chain in JB:LMAP and forms a list of all virtual pages, and their disk addresses, that occur between the specified limits.

CYCUSR

CYCUSR must examine each user, locate all DCB's, close all open files and save the JIT for accounting information. Validity checking is performed on JIT's, swapping RAD addresses and physical page numbers.

All users currently in core are processed first. As each is processed, the map is set by MAPSET to correspond to the user's map. Before processing users that are disk resident, the map is turned off. Each users context is then read in from the disk and processed in the same manner as the core resident users. Users in the process of being swapped in or out are flagged either entirely in or entirely out according to the following tests: If the swap-in-progress flag (S:SIP) is zero or the number of users being swapped out (DID#10) is zero, no users are in transition and no flags are changed. If DID#10 is negative, all the users in the out swap list are flagged as in-core. If DID$10 is positive, the in-swap user (S:ISUN) is flagged out-of-core.

SMUIS establishes the length of the user control tables. The tables are scanned twice- once to locate in-core users and the second time to locate out-of-core users. Let N stand for a user number. UB:US(N) is the user's state; if zero or 2, N is not a user. $UH{:}FLG(N)_6$ is in-core or out-of-core flag.

In-core users:

Each JIT is validity checked by checking the user's TSTACK. The contents of the TSTACK pointer (Location TSTACK) minus the space used (TSTACK+1, bits 16-32) must equal the address TSTACK+1.

Each swapping RAD address in JH:DA is validity checked by CKRAD.

Each physical page number corresponding to a virtual page in JB:LMAP is validity checked by being >INITIAL↓9 and <CORED↓9.

The number of pages in the user's physical page chain in MX:PPUT is checked against JB:PPC. The tail of the user's physical page chain in MX:PPUT must equal JX:PPT.

Failure of any of these checks causes CYCUSR to type an error message on the operator's console, skip this user, and continue to the next user.

A pointer to the JIT is passed to CLSFILS subroutine, which closes all open user DCBs. A pointer to the JIT is also passed to COOPFILS, which tracks down and packages all COOP files. Subroutine PARCOM is called to type partially completed messages if the user was a batch job. The disk address of the JIT is then saved in the recovery buffer so that RECOVERY2 can update the accounting log. Each in-core JIT is written to its assigned RAD space.

As the in-core users are processed, a check is made to determine if the Symbiont Ghost job (RBBAT) is in-core. If not, no processing is needed. When in-core, the data (00 protection) pages and AJIT page are written back to swap RAD. The page addresses are found in the command list and the disk addresses and AJIT page addresses are found in the JIT. These pages must be on swap storage so they can be read by RECOVER2.

Out-of-core users:

After all in-core users are processed, the map is turned off and each out-of-core user is examined. Out-of-core users' pages are read into physical memory as though mapped one-to-one, virtual to physical. The user's JIT is read into JJITVP, and his AJIT, if he had one, is read into JJITVP+1. A list of the user's context virtual pages and their disk addresses is formed by searching through the JB:LMAP chain. Each page in this list is read in. All the disk addresses in the JH:DA table are verified. As was done for the in-core users, the CLSFILS, COOPFILS, and PARCOM routines are called to close up the files and type the partially completed messages. Finally, each out-of-core user's JIT disk address is added to the table of users' JIT disk addresses in the recovery buffer.

After all users have been processed, CHKCFU is called to find any files remaining open. If the CFUs have been clobbered, recovery is judged impossible and the error exit is taken.

UTS TECHNICAL MANUAL

ID

SVDNDEV - Save down devices

PURPOSE

Loop through DCT3 and determine which devices are down. Save the down device list so that it can be merged with the DCT3 by recovery second phase.

USAGE

BAL, 11 SVDNDEV

INPUT

Bit 2 of DCT3 - when set, indicates the device is marked down.
DCTSIZ is the length of the DCTs.

OUTPUT

One word is moved to the recovery buffer for each device which is flagged down. Byte 0 contains code X'06', which signifies to second phase that this is a down device. Byte 3 contains the index into the DCTs for the down device.

Future: If diagnostic has a bit to indicate temporary-down or diagnostic-down; when that bit is set, the device will not be remembered as down by SVDNDEV.

UTS TECHNICAL MANUAL

ID

TSTUSR – Verify user control tables

PURPOSE

Check all monitor controlling user tables and type an error message if tables have been clobbered.

USAGE

BAL, 11 TSTUSR
normal return

INPUT

UH:FLG	-	User flags
UX:JIT	-	JIT physical page number
UH:JIT	-	JIT disc address
UH:AJIT	-	AJIT disc address
UB:US	-	User state
SNSTS	-	Number of states
INITIAL	-	Lowest user virtual address
CORED	-	Highest user virtual address

ERROR

Failure of any of the tests causes a cryptic message "USR CNTL T" to be typed on the OC by ABNXT as described in section OD.

SUBROUTINES

ABNXT – Error message typing

DESCRIPTION

TSTUSR checks as many user tables as can be checked without reading JITs in from the RAD.

TESTS:
If UB:US is zero or 2, there is no assigned user.
UH:FLG: 1) if bit 15 is set, 6 must also be set because if a user is in core and

ready to run, his JIT must also be in core.

2) Bits 0 and 1 must be 0 because they are unused.

3) If bit 9 is set, bit 5 must also be set because if a debugger is in control, a debugger must be associated with the user.

UX:JIT: If UH:FLG bit 6 is set, then the JIT is in core and the JIT physical page number in UX:JIT must be greater than the page number of INITIAL (right shift 9) and less than CORED.

UH:JIT and UH:AJIT are either 0 or legitimate swapping RAD addresses.

UB:US must be less than or equal to SNSTS, the number of possible states.
If UB:US is zero, UH:JIT and UH:ID must also be zero.
If UB:US is 2, UH:FLG must be non-zero.

UTS TECHNICAL MANUAL

ID

RCVDMP – Recovery dump to RAD

PURPOSE

If RECOVERY decides that the file system is probably okay and just the swapping RAD and core need be initialized, all of core is dumped on to a reserved area of the swapping RAD for later analysis.

USAGE

BAL, 11 RCVDMP

INPUT

RCVRAD	defines the first available address on the swapping RAD to be used for the dump
RCVRDSZ	defines the number of available granules
CORED	defines the top of physical memory
RCVRCNT	the number of recoveries executed since last start of the system

OUTPUT

Beginning with page 0, each page of physical core is written to the swapping RAD beginning with granule RCVRAD+2. The value of RCVRAD is saved in the RECOVERY buffer by SV1. The next available granule after the dump and the RCVRCNT are also saved by SV1. The address originally contained in RCVRAD is saved in TRCVRAD. This is the address where the recovery buffer will later be placed.

ERRORS

If the amount of available RAD space is less than that needed to save the core space, the tape dump is called and RAD addresses are not remembered by SV1.