

RELATIONSHIPS OF THE VARIOUS TABLES USED BY THE MEMORY MANAGER

Judy Simon

April 5, 1974

Technical Memo

TM.74-16

RELATIONSHIPS OF THE VARIOUS TABLES USED BY THE MEMORY MANAGER

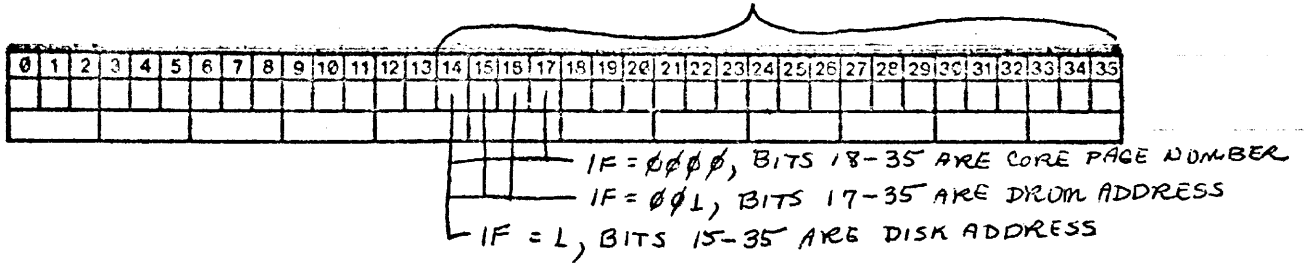
Page Table (PT)

The page table is the software map of a fork (process). Each page in the fork is represented in the fork's page table by one of three types of pointers; private, shared or indirect. The private pointer, indicated by BITS $\emptyset, 1 = \emptyset\emptyset$, contains the current address of the page (drum, disk or core). The shared pointer, indicated by BITS $\emptyset, 1 = \emptyset 1$, contains the index into the Special Pages Table (SPT). The SPT entry contains the current address of the page. (The page in the shared case is a page of a file). The indirect pointer, indicated by BITS $\emptyset, 1 = 1\emptyset$, contains an index into the SPT and a page number. The SPT entry pointed to contains the address of another page table and the page number is used as an index into that page table. Thus the indirect pointer allows a user to access a page in another user's address space.

INDEX BLOCKS (XB)

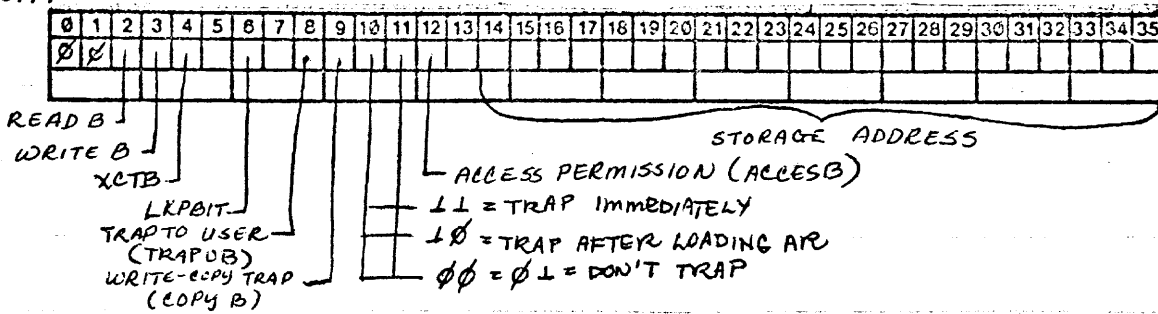
For a file, there are index blocks which are used like page tables. There is one entry in the XB for each page in the file. The entries are private pointers for pages on drum or disk and shared pointers for pages in core.

STORAGE ADDRESS

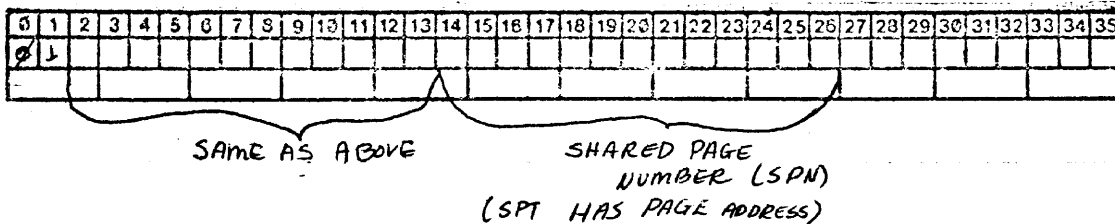


PAGE TABLE ENTRIES

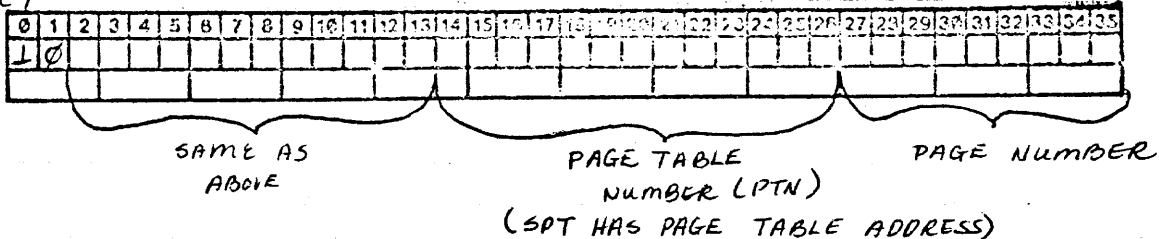
PRIVATE



SHARED



INDIRECT



SPECIAL PAGES TABLE (SPT)

The first part of the SPT called the OFN section (of length NOFN) contains entries pointing to index blocks. These entries are pointed to from the Job Storage Block (JSB) to associate files with jobs. The entries above the OFN section are pointers to shared file pages, page tables (see PAGE TABLE), Process Storage Blocks, (PSB) and Job Storage Blocks(JSB). FKPGS (indexed by FORKX, a system wide fork index) contains a "pointer" to the fork page table entry and a "pointer" to the fork PSB entry to tie the PSB and PT with the fork. (These "pointers" are actually indexes into the SPT. The SPT entry indexed to points to the PT or PSB.) Similarly, FKJOB (also indexed by FORKX) contains a pointer to the fork JSB entry to tie the fork to the job. Each fork (potentially shares a common job area with other forks tied to the same job. This sharing is accomplished by having the fork have indirect pointers in the PSB point to the JSB map area which actually points to the pages used in common (by forks) for the entire job. Just directly above the entries pointing to index blocks is an entry pointing to the monitor map (MMAP). This entry is for consistency with the entries for the users' page tables. (MMSPTN contains the SPTN which points to this entry.)

Because the entries in the SPT point to various data structures, the index into the SPT is referred to by these various names: SPTN (SPT index), SPN (shared page index), OFN (open file number, index), and PTN (page table index).

SPTH

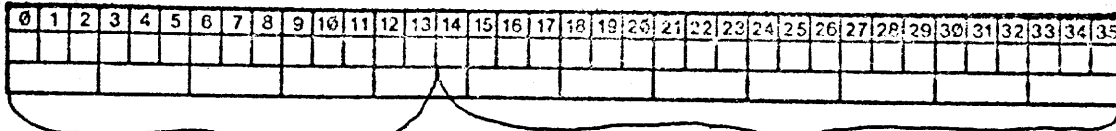
The SPTH parallels the SPT and is, therefore indexed identically.

The OFN section of the SPTH contains the disk address of the XB and the other section contains OFN's and page numbers indicating from where the shared file page originated. (For PSB's, JSB's and UPT's, the entry is zero.)

Drum Status Table (DST)

This table is indexed as a function of drum address and used as a place to store the disk address of a page while the page is in core.

SPECIAL PAGES TABLE

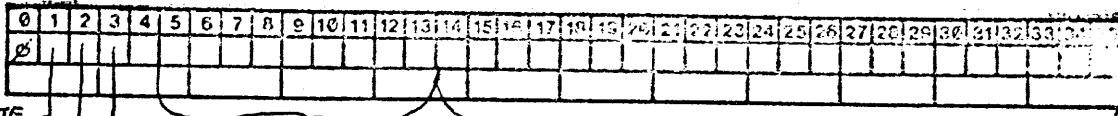


SHARE COUNT
 FILES - +1/ OPENING & SHARING
 OTHER - +1/ SHARING

STORAGE ADDRESS

FREE SPT'S CONTAIN ADDRESS OF NEXT FREE SPT

SPTH - OFN SECTION, POINTERS TO KB'S OF OPEN FILES



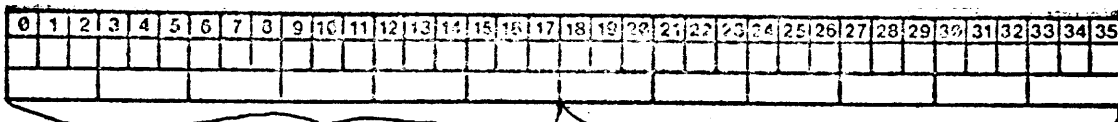
WRITE BIT
 THAWED BIT
 NEW FILE BIT

CLASS FIELD

DISK ADDRESS OF KB

FREE SLOTS = -1

SPTH - OTHER SECTION



SHARED FILE PAGES:
 PSB's, JSB's, UPT's:

OFN
 ∅

PN
 ∅

(ORIGINATING)

FREE SLOTS - LIST CHAINED THROUGH SPT.

CORE STATUS TABLES (CST0, 1, 2, 3).

These four tables are parallel and map real core i.e. they are indexed into by core page number.

CST0

BITS 0-8 contain the age of the page if the value is greater than 100; otherwise it contains a code indicating the status of the page such as read in progress or on Replaceable Queue (free list). CST0 also contains the modified bit (dirty bit).

CST1

Left half: lock count used by pager to hold page tables in core while it diddles with their pages.

Right half: drum address for this page.

CST2

CST2 contains the home map location of this page i.e. the page table, index block or SPT entry which points here. For a page table or index block, the entry is

PTN.PN (left half PTN and right half page number).

For a SPT pointer, the entry is 0.SPTN

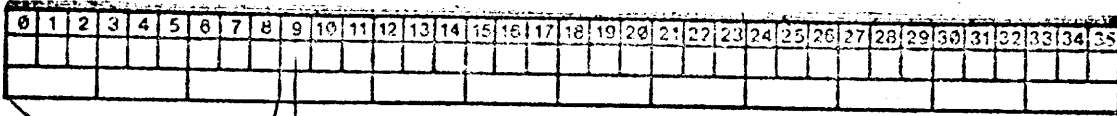
CST3

CST3 contains the fork number for an assigned page or

BACKWARD-LIST-POINTER.FORWARD-LIST-POINTER for a page on the Replaceable Queue.

CORE STATUS TABLES

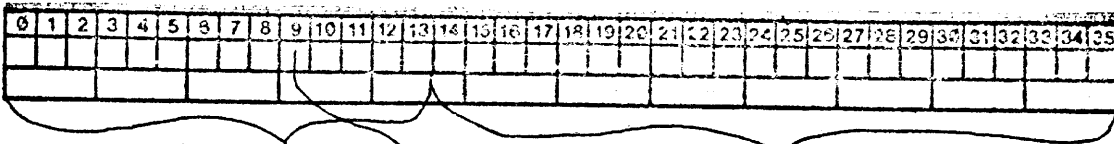
CST0 - AGE



AGE
 000 - ON REPLACEABLE QUEUE (RPLQ)
 010 - TO BE PUT ON RPLQ
 020 - READ COMPLETELY
 040 - WRITE IN PROGRESS
 060 - READ IN PROGRESS
 100 - ASSIGNED

MODIFIED BIT

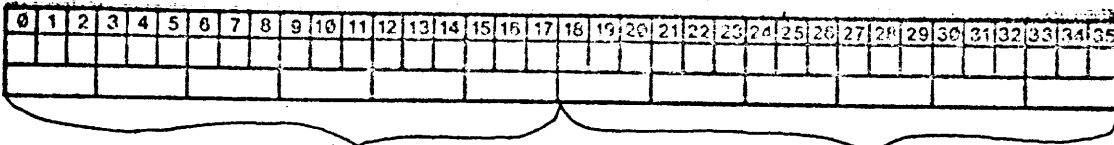
CST1 - LOCK



LOCK COUNT (MUST BE = 0 TO BE SWAPPED)
 PAGER LOCK

DRUM ADDRESS

CST2 - HOME MAP LOCATION

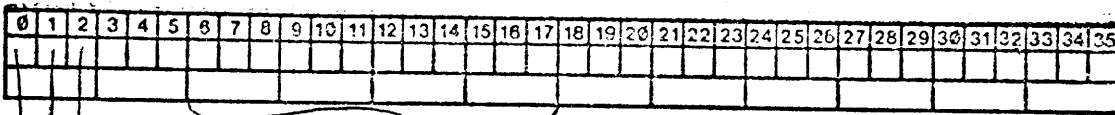


PT OR XB:
 SPT:

PTN
0

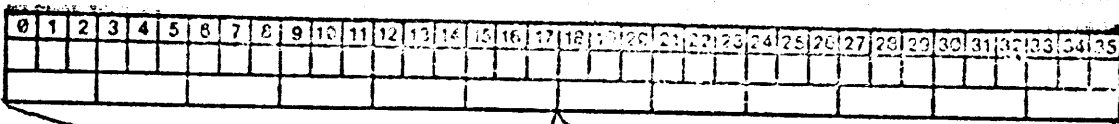
PN
SPTN

CST3 - OWNER



L SWAP TO DISK REQUESTED BY DDMP
 SUPER
 WRITE IN PROGRESS
 FORK No.

CST3 - ON REPLACEABLE QUEUE



BACKWARD LIST POINTER

FORWARD LIST POINTER