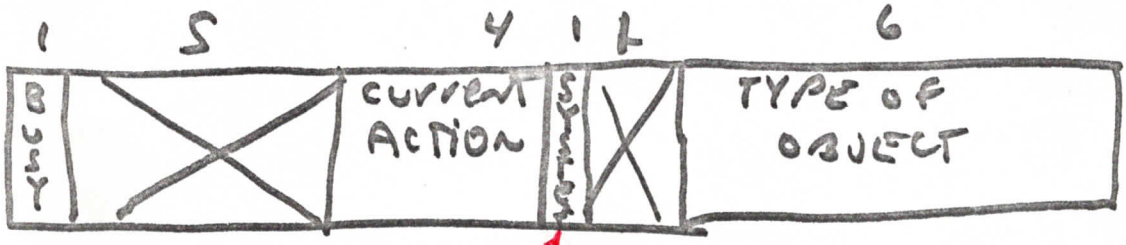
THE BEAD

THE BEAD IS THE FIRST SUBPROCESS CREATED IN EVERY PROCESS WHEN THE SYSTEM IS INITIALIZED. ITS FUNCTION IS TO ACT AS AN INTERIM MONITOR PACKAGE TO COORDINATE FILE ACTIVITY AND NAMING, ELEMENTARY UTILITIES, AND BE A COMMAND PROCESSOR FOR THE USER.

EVERY BEAD IS GIVEN A CLIST WITH CAPABILITIES FOR ITS TELETYPE AND FOR THE SYSTEM DIRECTORY. THE SYSTEM DIRECTORY CONTAINS NAMES AND TYPES OF OBJECTS IN A STANDARD FILE AND A GIANT CLIST WITH CAPABILITIES FOR THESE OBJECTS. THE NAME FILE HAS ENTRIES FORMATTED LIKE THIS

1	2	3	4	5	6	7	8	9	10
NAME OF OBJECT									TYPE/STATUS
NAME of OWNER									OBJECT NUMBER
SCOPE file info	SCOPE File INFO				CURRENT File Pointer				
	BLOCK SIZE				CURRENT LENGTH OF File NEXT BLOCK IS created AT THIS ADDRESS				

TYPE/STATUS Fields



SYSTEXT set to 1 if SYSTEXT File

CURRENT ACTION codes:

3 - WRITING } used only by SCOPE
4 - READING }
5 - REWOUND
6 - AT END of information

CALLS ON THE BEAD: (save registers first)

X1 - OBJECT NAME
X2 - USER NAME OR ZERO IF CURRENT USER NAME IS REQUESTED
B1 - ADDRESS OF 4 WORD AREA FOR ENTRY
B7 - LINK LIST INDEX FOR ENTRY CAPABILITY
B6 - ACTION

ACTIONS

0 - LOCATE NAMED OBJECT. MAKE BUSY. PUT ENTRY AT B1 AND CAPABILITY AT B7. IF THE OBJECT IS BUSY, GIVE BUSY NOTIFICATION.

1 - UPDATE OBJECT FROM OBJECT NUMBER field IN ENTRY AT B1

2 - delete OBJECT FROM directory. OBJECT NUMBER ? FROM ENTRY AT B1

3 - CALL SUBPROCESS. GET DESCRIPTOR FROM File name AND USER NAME IN X1 AND X2

.4 — STOP
(back of prev page)

PF, "COPIES", "CFILE", , ,

COMMANDS TO THE BEAD:

{C / CALL}, file, USER, Par 1, Par 2 (CR)

A subprocess descriptor is obtained from the named file and created. A call is then made to transfer control with the two parameters in X4 and X5 left justified display code.

{S / SNATCH}, file, USER (CR)

THE named file is made NOT BUSY.

{D / Delete}, file, USER (CR)

The named file is deleted

CONTINUE (CR)

entered to continue anyway after a busy notification is made

TRY (CR)

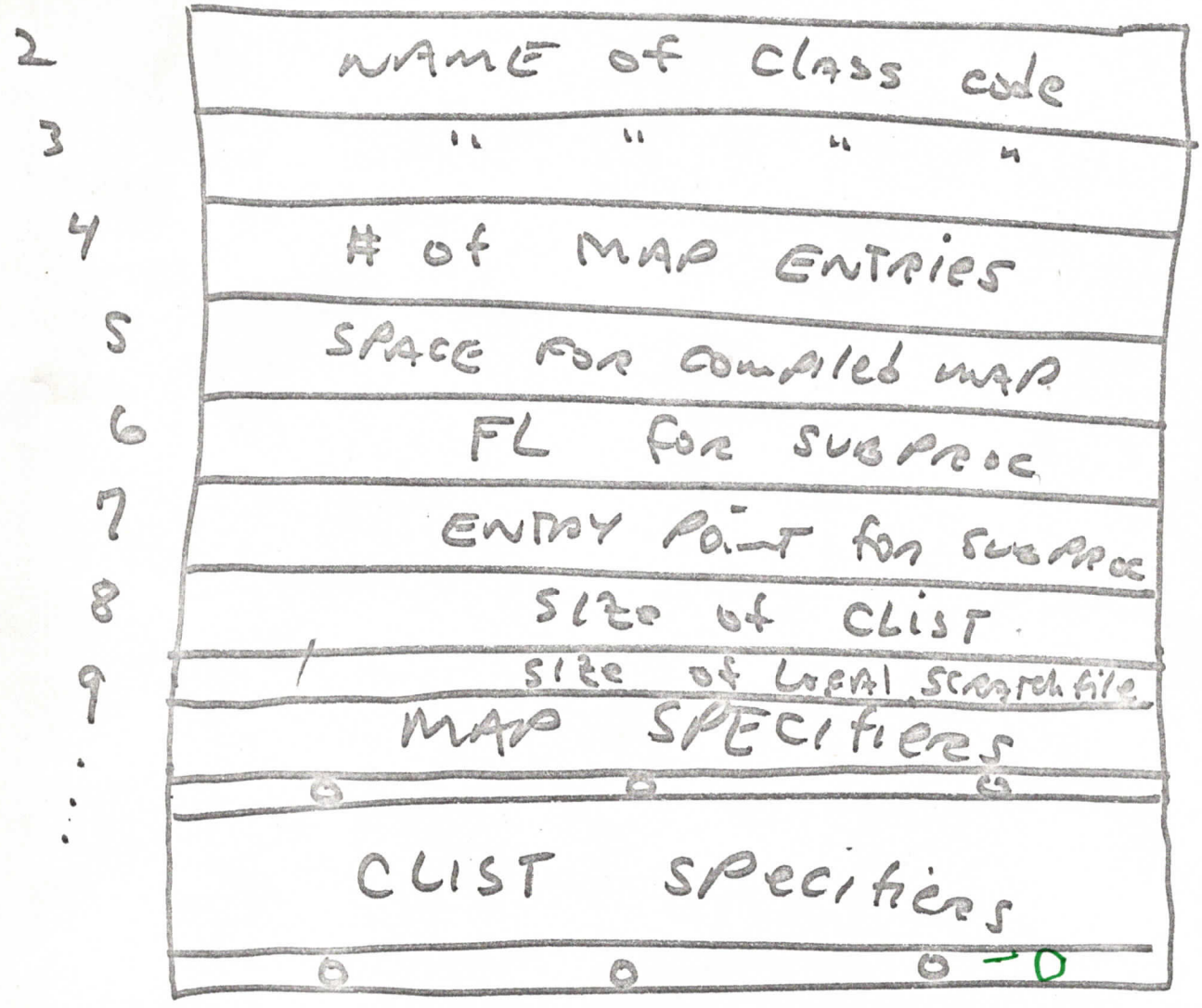
entered to try again and check if the file is still busy after a busy notification is made

RESTORE — undoes STOP

SUBPROCESS descriptors

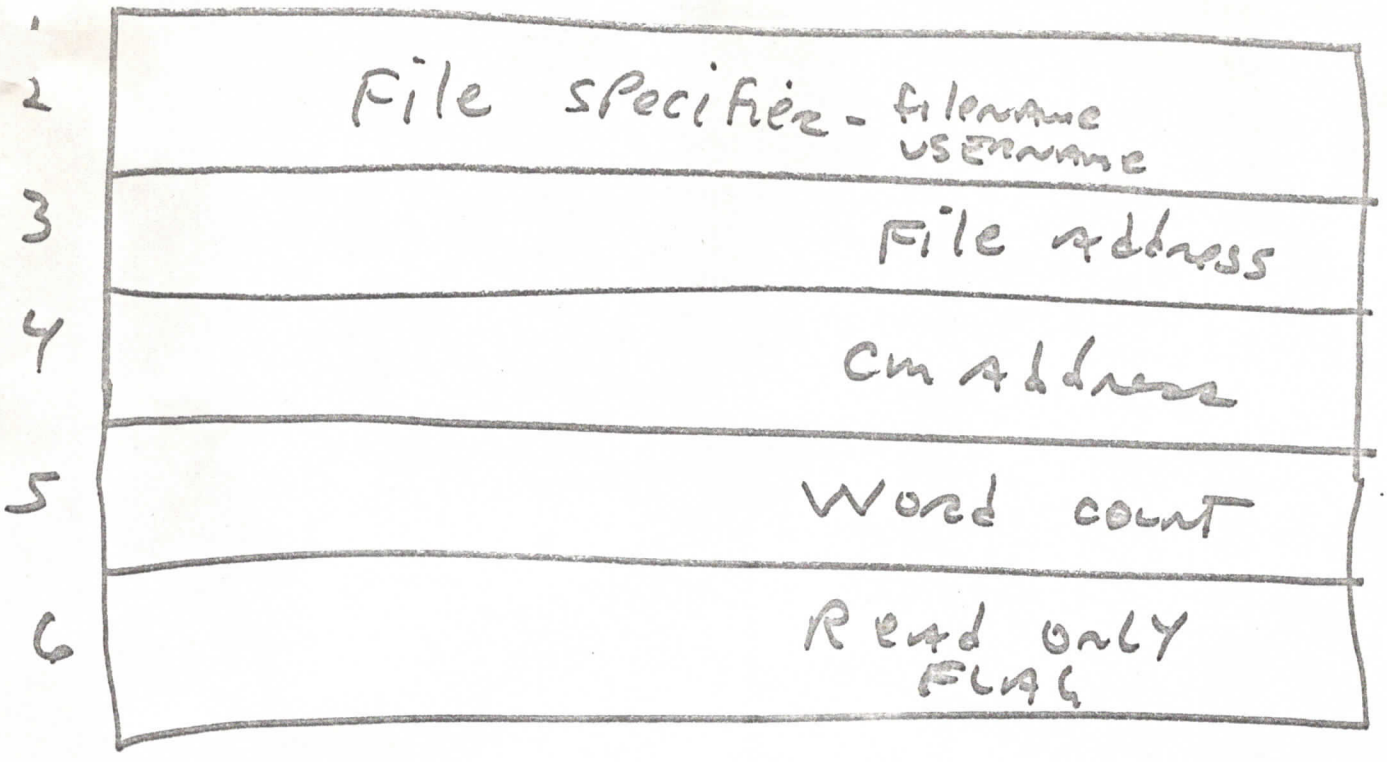
THE FIRST WORDS of A BINARY OUTPUT file ~~contain~~ FROM THE ASSEMBLER CONTAIN THE FOLLOWING INFORMATION:

ADDRESS IN ASSEMBLY LISTING



C LIST SPECIFIERS CONTAIN THE 2 WORD NAMES of OBJECTS. N APPEARS in THE FIRST PART of THE LOCAL CLIST.

MAP SPECIFIERS



IF THE FILE NAME IS ZERO, THE LOCAL SCRATCH FILE WILL BE USED.

NOTE THAT THE ASSEMBLY addresses AND THE FILE addresses differ BY 17_{10} BECAUSE of THE 50 TABLE AND THE 77 TABLE AND THE SCOPE Logical RECORD WORD AT THE BEGINNING of THE FILE.

NAMES of SPECIAL OBJECTS IN THE DIRECTORY:

SYSTEM OPERATIONS HAVE USER NAMES of 'OPERATE'. THE LIST includes:

READ
WRITE
SENDE (SENDE)
HANG
GETE
CCLIST - CREATE C' LIST
CFILE - CREATE FILE
CBLK
CPROC
CEVCH
CSPROC
CCC - CREATE CLASS CODE
SAVE - SAVE REGISTERS
RESTOR
DSPCAP - DISPLAY CAPABILITY
DSPANY - DISPLAY ANY CAPABILITY
MVECAP - MOVE CAP IN FULL C LIST
CAP IN - MOVE CAP INTO FULL C LIST
CAP OUT - MOVE CAP OUT OF FULL C LIST
RSHAPE - READ FILE SHAPE
COPER - CREATE OPERATION
MBLOCK - MOVE BLOCK

PROBE - CHECK BLOCK
DBLK - DESTROY BLOCK

CONDITIONS AT INITIAL ENTRY TO A called SUB PROCESS:

C-LIST:

0 ALLOCATION BLOCK
1 TTY cap 2 (TTY File)
2 TTY cap 2 (CP to PP event channel)
3 TTY CAP 3 (PP to CP event channel)
4 CALL TO BEAD ←
5 Call TO SELF ←
6 Cap of self ← class code
7 Cap of THIS C-LIST

1 E-BEAD

3 read

4 write

5 sendc

6 hang