Rec 7 [1970? – PMcJ 5/18/98]

I) <u>Bead naming conventions</u>

a bead $2 part name will define an entry in
some directory. The desired entry is that
named by the 1ˢᵗ part of the name in that directory
obtained as follows.

If the second part is null, or the same as the
current user name, then the temporary directory
is used; otherwise the 2ⁿᵈ part is looked up in
a special directory. The result, assumed
to be a directory, is used.

note: The special directory contains at least the following
entries. For each logged in user, his temporary
directory under his user name, with all dangerous
option bits turned off. under "S", the system call
directory. under "class", a directory to hold class names.
under "operate" a directory to hold some ecs actions.

te: ecs objects will be held in dynamic or static name tags

note: the "class" and "operate" may instead use special c-lists and
file of names?

II) **Bead calls** (directory actions)

A) B6 =
read directory entry

     An
i) ~~The~~ entry will be looked up as in I). If not found ~~Bina a disk file~~ ~~~~ ~~~~ and the 2nd part of the Bead name was either null or that of the user then a disk file will be created and ~~~~ placed in the entry namedas in I).

ii) ~~If the result of I) is a disk or ecs~~

ii) If the result of ~~I)~~ i) is a name tag, the value of the ~~tag~~ is obtained , as the ~~result~~ new result.

iii) If the result of i) or ii) is a disk or ecs file then
    a) The type field ~~of~~ of the 4 word pseudo Bead entry is set to 0.
    b) The left half of the 4th word is set to the blocksize, the right half of the 4th word is set to the address of the 1st missing word. (via pokes)
    c) If the 1st word of the file is present and negative the system ~~to~~ b.t is set on, else off.

iv) If the result of i) or ii) is not a file, the type field is set to 2 and the 4th word is set to zero.

v) In any case, the 1st part of the B end name is placed in the 1st word, The 2nd part (current user name, if null) is placed in the 2nd word. The index is set to 0. The 3rd word is set to 0.

vi) If the directories do not have ~~may~~ needed option bits or if a file found with many levels funny things may result.

B)   B6 =
write directory entry
        NOOP

C)   B6 =
kill directory entry
    The named object is destroyed and the named entry destroyed.
    If the ~~the~~ entry ~~to hold~~ hold a ~~of~~ dynamic name tag,
    The associated object also destroyed.
        [if proper option bits not on, funnies]

D) B6 =

read a directory entry, ignore Busy bit

same as A)

6) B6 =

set a directory entry (new action)

a dynamic name tag is created so, same as in I).

It is given as value the object passed. (a c.list index B7)

only works if 2nd part of head name is null or current user.

III) temporary version of II

II) will be simulated using a c.list and file
more or less like the current head directory,
main differences from II will be no option bits,
no disk files.