4/23/70

0) <u>open and close, general</u>

for objects of all kinds, the following generalities are true.
[some objects cannot be opened! (access keys)]

1) a global open count is maintained for each object.
This is the number of processes holding the object open

2) a local open count is maintained in each process for each object it holds open.

The global open count for an object is increased and decreased by 1 each time a local open count for the object departs from 0, or reaches 0, respectively.

3) for some kinds of objects, there is a different capability represent the open form of the object from that representing the closed form of the object.

for these objects, closing is sometimes done by presenting a capability for the open form & sometimes by presenting a capability for the closed form.

4) whenever a disk process is destroyed, the global open count is decremented for all objects which have a non zero local open count,

[representation of lowlevel disk objects.
diskaddress and unique name]

I) directories

(These are lowlevel disk objects)

A) creation

[see object creation of document of 4/17/70]

additional parameters would be:

i) max size (wds, ~~[illegible]~~ entries?)

ii) whether this directory is to ~~[illegible]~~ have a directly associated accounting block, and if so, how much of each allocated item to ~~[illegible]~~ ~~[illegible]~~ place in this one.

B) open action

does not return any capability, but does insure that the lowlevel file representing the directory is open. (i.e. if a 0-level file, all in ECS) Hence the open directory represented by the same capability as the closed directory

C) close action

applied to the same capability as the closed directory. after enough closes, the ~~[illegible]~~ lowlevel file representing the directory will be moved back to the disk. (global open count = 0)

note: all directory actions may be performed on a closed directory. It will be temporaly opened for the action.

II) <u>disk files</u>

(These are low level disk objects)

A) <u>creation</u>

additional parameters (over object creation of 4/17/70) would be only shape.

B) <u>open action</u>

This returns a new kind of capability (ers file)

C) <u>close action</u>

See Bruce for what kind of capability needed here

III) subprocess descriptors

(These are low level disk objects)

A) Creation

See a special document on subprocess descriptors (a new one not yet ready)

B) open

similar to directories, This just causes The low level/file representing the subprocess descriptor to be brought into ECS. No new capability is returned.
all actions may be performed by the user on an unopened descriptor. It will be temporarily opened for The action.

C) close

Just present the same capability

Note: For directories and subprocess descriptors, The open action just makes repeated reference to The object more efficient.

IV access keys

each access key is just a number
represented in the directory by that number
represented in its capabilities by that number

access keys may not be opened and closed

creation requires no additional parameters.
creation does not make an ownership entry, thus
access keys may not be destroyed.

V) Global ECS object

A) implementation

a clist containing the capabilities for the objects
and a file containg a unique name for each of the objects

B) directory representation

The index in the c-list of A) along with the corresponding
unique name in the file of A)

C) creation

(will be used only by system routines)
supply a capability for an ecs object (with destruction bit on)
The capability will be placed in an empty slot in
The c-list of A, a new unique name will be assigned to it
and the result placed in the directory
(This creates an ownership entry) (Thus permitting destruction, which removes the object from the c-list of A and resets the unique name in the file of A)

D) open

returns the capability from the c-list of A), option bits
anded with the option bits of the closed form of the object.
(also unique name is checked)

E) close

(not applicable)

VI) ecs goodie (do not remember new name)

A) implementation

a clist containing capabilities for open objects and a hash table, hashed on ecs-goodie unique name, pointing to corresponding c-list location and containing global open count.

B) Creation

Just produces a new ecs-goodie unique name which is placed in directory. Does not make an ownership entry.

C) open

2 kinds of open

1) open an ecs goodie with a presented ecs object

F-return if the unique name of the ecs goodie already in the Hash table of 4)

error if the destruction bit not on in the capability for the presented object

otherwise, the unique name is entered in the hash table, a free slot is found in the c-list and the capability for the presented ecs object is placed in the c-list. Also the hash table entry is made to point to the slot in the c-list and the global open count becomes 1.

2) Open an ecs goodie, presenting no ecs object.

F-return if the unique name of the ecs goodie is not in the hash table

otherwise; bump the open counts as needed
and return ~~th~~ the capability from the indicated
place in the c-list, option bits masked by the
option bits of the capability for the closed
form of the object.

D) <u>close</u>

must present a capability for the closed form of the object
If global count goes to zero, the entry is removed from
the hash table and the corresponding ecs object
is destroyed.