

1/22/70

EB

objects in directories

- files
- directories
- access keys
- subprocess descriptors ← public ^{subprocess} files
- ecs goodies
- soft links

note on ecs goodie

basic ops to present

1. present an ecs goodie and offer an
ecs object in return [one gets the option bits on the goodie
and add with the bits with the
object?]
2. present an ecs goodie and an ecs object
and ask that they be ident. fixed [implies a special option
bit on the goodie?]

Representation of files, directories, access keys, etc goodies

[note: s-b process descriptors later]

directories

file d. slot address + unique name

directory "

access key a number

ecs goodie a number + unique name

[a slot address of an ecs

~~table~~ table of the ecs objects

] hence the name would be checked]

ecs

special type capability

data word holds disk address + unique

" (different from)

special type capability

data word holds the

[see for uniformity, in procedure

to using ~~the~~ the class code mac 407.301]

same as file

directory structure for an entry

name, object, accesskey, list

<entry> ::= <name> <object> <accesskey, list>

<accesskey, list> ::= ~~<keynumber> <option bits>~~ <access> | <accesskeylist> <access>

<access> ::= <keynumber> <option bits>

note: associated with each directory will be a special accesskey
That access key will appear on all accesskey, lists in the directory
with full option bits.

a directory is being "held"
 if not these actions
 will cause at least a disk read
 & maybe a read-write if
 directory modified

used if
 multiple actions
 dependent on
 directory?

Hold a directory

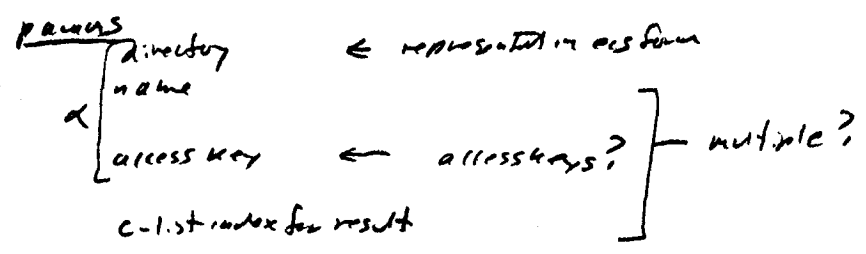
directory

Release a directory

directory

Directory operations [by user process]

Fetch an entry



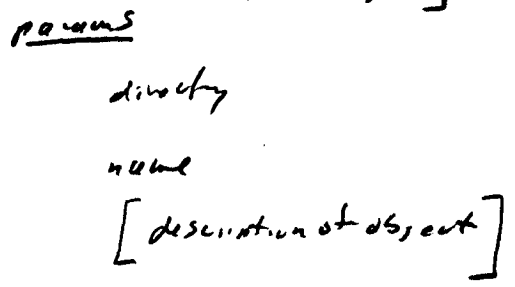
result

is representation of the object
 at indicated place in c-list

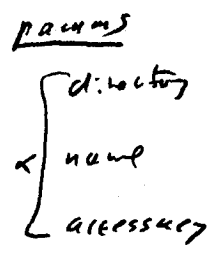
This should
 be user
 code, interba
 if we
 know the
 command proce
 others can be
 written

Log on
 g:uid
 user pass
 2 d. returns
 temp
 perm
 for an access key
 if more than
 one log on on
 same machine
 then different
 temp parts
 but same perm
 |
 |
 |

Create an entry [i.e. create a disk system object]

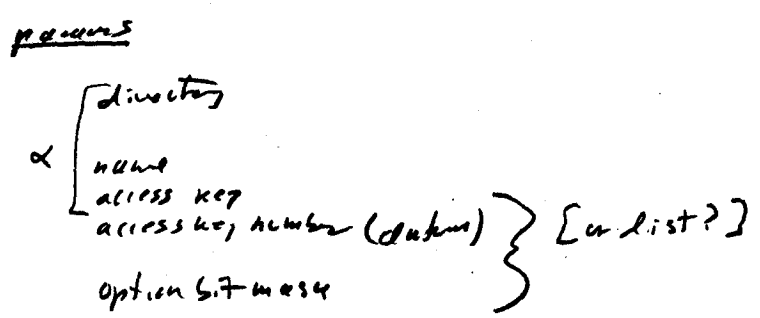


Destroy an entry

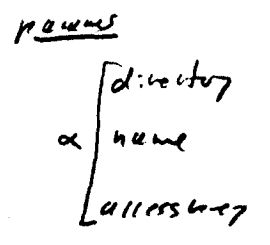


[note: destroys the object]

add access keys to an entry



Remove an entry (?)



delete access keys from an entry

same as add access keys

sub process descriptor

class code



class code of further?



#map entries

need individual map entries

compiled map site

subprocess file

entry point

cd.it

- file
- file address
- cm address
- cnt
- RO

need c-list size entries

calling operations defined by sub process descriptor

3 ways in which objects at arons might be known

1. known to constructor of the sub process descriptor

e.g. The file containing the code for the sub process

The class code for the sub process

2. known to constructor of the sub process

e.g. (have none) (but clear this is needed)

3. Local to the sub process itself

e.g. scratch rw file of local class



how what directly do we put this in

or is there files not in directories

or can't have user home access because it could contain secret data at very least, don't want user writing in it.

4. PIS objects

e.g. standard c-lists (over only in int)

3. by force certain PIS actions

which objects could be controlled by access