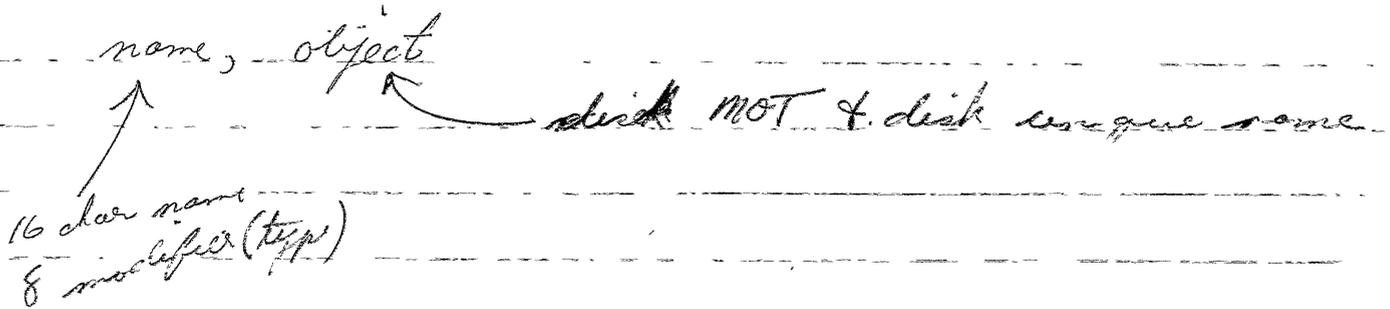


10 apr

directory entry



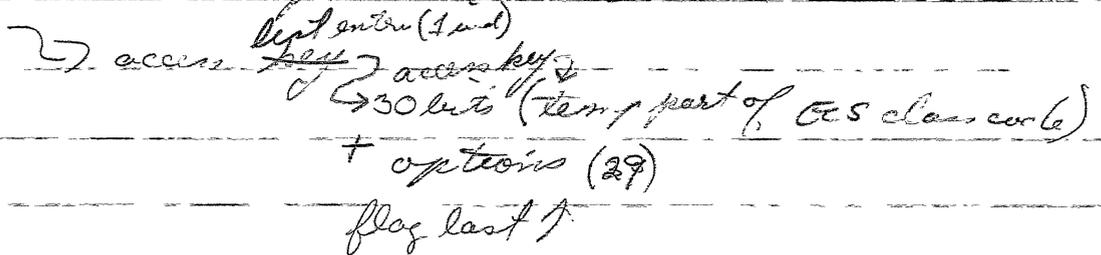
access ~~key~~ with the object on disc

on login - login program

user name match with password

produces an access codes

Access Key \rightarrow 64 word pblock on disc



Disk finding things

① Disk MOT (has unique name & disc addr)

② Dynamic MOT (hash on disc ~~unique name~~ - has # to find disc info table in ECS table)

is not destroyed in ECS until removed from table

18 bits of ECS capability indexes disc info table

~~③ split table (has unique name)~~

Disk info table

- 1) disk unique name
- 2) disk MOT
- 3) attachment info

C memory

directory entries

file: name (variable length)

unique name (disk)

DOT

directory - name

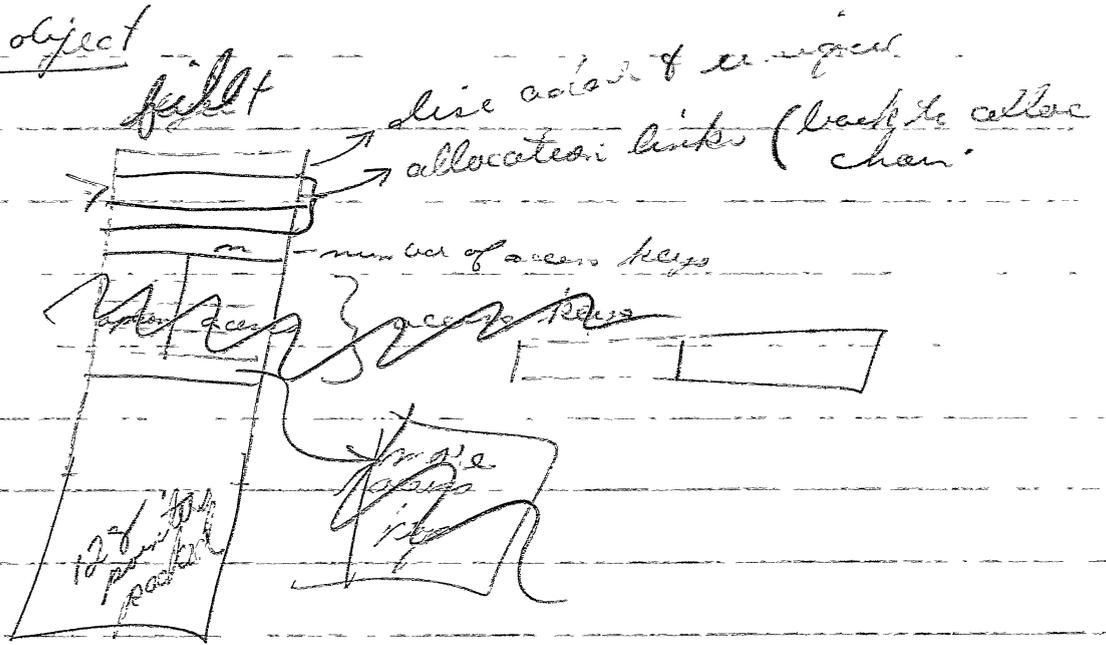
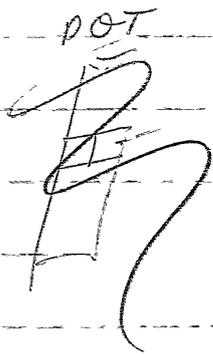
unique name - DOT

allocation block - name

unique name DOT

link: name (variable length)

disk object



disk obj

dir

access key (obj) (access list in directory)

link ✓ (access list in directory)
file

name abbreviation:

leading letters on both parts

contains

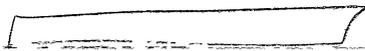
dir cap

normal

access key ✓

STORE on Disk

pointer block



24 bit disk address packed
tight

everything file - directory contains unique names
and disk address of header
record

Reader Record

- 1) check info (disk address & unique name)
- 2) file shape
- 3) pointers or data

Record format

64

some bits on. BS file exp used to
include Disk info table and

DISK

3 June

two local attachment counts → one for map
one for other
file blocks

access list in directory
links record to access files of others

ECS

no op to destroy alloc blk
op to return csp for 1st obj on alloc blk

Disk alloc blk

each obj contains ptr to alloc blk
alloc blk contains list of objects belonging to it

Disk object,

ECS -

jump call - decrement
stack and do stuff

swappable process

file open list
block attach list (counters)
cut

swap obj
process open count
block attach (if processes
are attached)

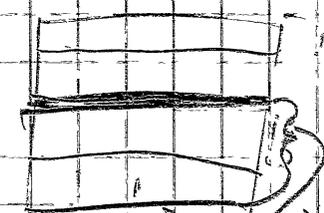
directory

name
with address
end of header block

{ 64
127
257
513

disc. object

unique name, unique name, block size, disk addr



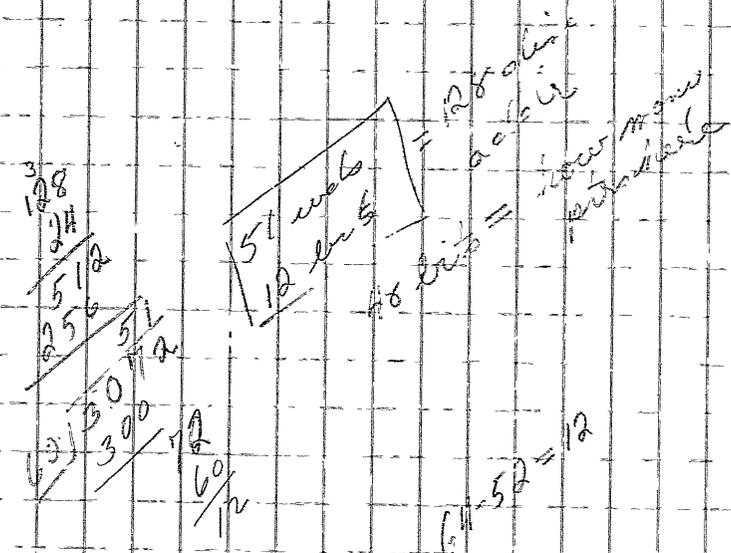
unique name - disk addr
 - shape - like key pin on album card
 allocation (unique name, disk address, size)
 1 bit at high order of field



pointer (actual)
 or data

pointer 24 bit disk addr

size of header
 size highest level
 of file
 fill shape



- 1 - check word
- 2 - shape
- 3 - alloc bit
- 4 -
- 5 -
- 6 -
- 7 -
- 8 -
- 9 -
- 10 -
- 11 -
- 12 -

shape = shape mask
 # levels
 1st level

Disk system

capacity ~~options~~ (3 blocks)

2 blocks ~~to~~ 1000_{10}

disk ~~also~~ master C-leaf

ECS file capability

DIT ~~with~~ header table

pointer ~~array~~

header

$m = 2$

~~summary of~~ ~~header~~ process open count

not on zero level file)

2 planes
data data
block
array

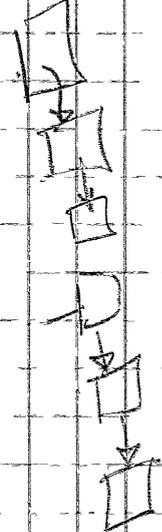
attach count

status

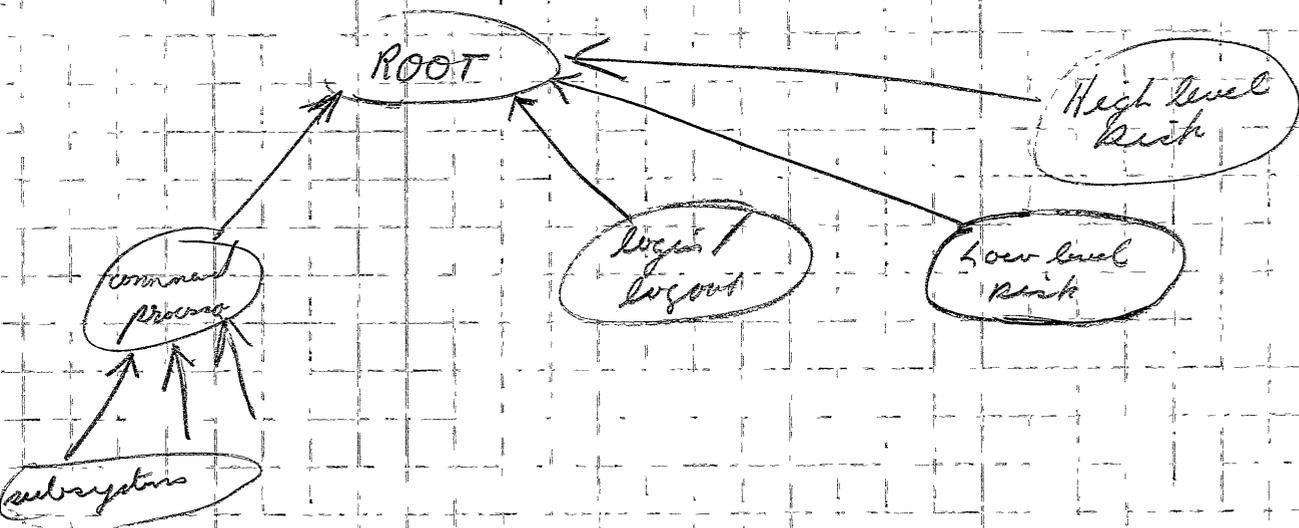
2)

point to block
disk
addr

pointer 2
pointer 1



DIT ~~with~~ table



ROOT

interrupt
 lens collector code

LOGIN/LOGOUT

login \Rightarrow create
 close up
 logout

LOW LEVEL DISK

Ersturn on file read/write
 open/close
 attach/detach \Rightarrow swappable process
 alloc disk implementation

command processor

create subroutines &
 call it

High level DISK

directory
 alloc blk
 swap process (simulate swaps)

line collector

→ for command mode processing - the line collector code in the ROOT should be used with the data area in the subsystem using the tty.

→ for subsystem FO (eg. editor - interactive systems) the line collector code may be included in the subsystem - NOTE: this precludes the switching of input and should only be used with systems designed to operate with a terminal

SUBSYSTEM FORMULA (ready)

- 1) subprocess name - further name
- 2) C-list
- 3) files \leftrightarrow map : instructions
- 4) call ops
- 5) entry point, etc to construct subp

suggest \rightarrow formula is in C-list

C-list has

- 1) name & father name
- 2) file contains map^{to} instructions to construct C-list and call ops & entry points etc

Thus call to have subsystem created would have

- 1) C: C-list (formula)
- 2) R: C-list to return operations
- 3) D: index in APP for 1st operation

15 July

ECS action

when ^{interrupt} interrupt strikes, set lock out

ECS action(s)

set and reset lockout
(note on release - check for pencils)

6 Aug

ECS system

actw does subp exist?

API: c: class code

return $XG = 0$ is no such subp
 $XG \neq 0$ if subp exists

option bit : option bit to permit move & mask \Rightarrow

actw : move w/o mask

fast actw ~~option bits (high order) index~~
~~which actw~~ which actw in
middle of operation

Subsystem formula (long term)

user opern C-ent: all ops allowed user
on disc system

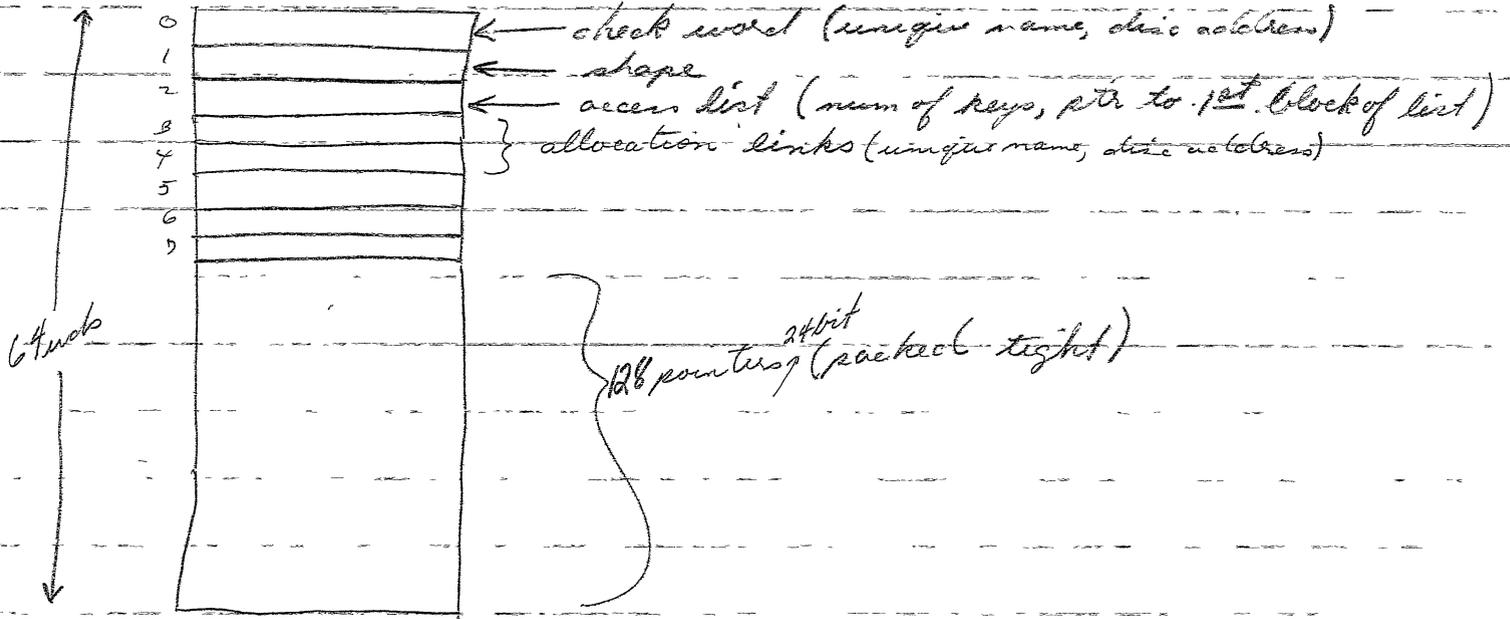
subsystem formula on a disc file

make "ready" formula from map
list of user ops \rightarrow add file info

- \rightarrow new class code will be assigned
to the map \rightarrow father will be
command process (default)
- \rightarrow (may name other files in same director)

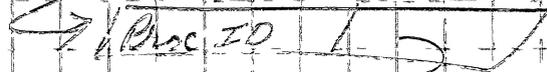
(subsystem?)
parent pointer to make transformation

disc file - on disc



file suspension queue

DHT header



Each disk process has EC used
for suspension queuing - each proc
queues itself for suspension (lock) &
then hangs

DISK SYSTEM

allocation block (survives post logout + perm)

Q-list (same as ECS)

disc capabilities have good word to identify object in disc app MOT

class-ware

access key (survives)

fixed file

swappable file (survives post logout)

fixed proc

swappable proc

directory K (survives post logout)

contains 'swap file' & i.e. → survives
access key / list of triple
alloc bit / name → access list, → perm object

operations (same as ECS)

arb length name & type
str, str, -str3, str4
↑
arb length
1st char of file name is alphabetic

3 char (24 char)
10 char name
5 char type (eg. binary, list)
prefix allowed w/ whole table for duplicate whole name match over-rides

Types also in disc MOT to allow for bag collection

contains disc MOT info by prod type

access key + options

filename = letter [letter | digit | .]⁰⁰⁰ [: [letter | digit | .]⁰⁰⁰]

To match an input string to a filename in a directory, break the string just before every . or :, then look for a filename containing these substrings in the same order (the first substring must match starting from the left end of the filename)



ECS action
API - C: class with
P: data

if perm API \Rightarrow
then construct class capability
w/ type of temp API \leftarrow
value = data and
 \uparrow 2nd and

not temp API to
new type