

## local C-list

bottom of full path C-list in core →

read only C-list

percons → write in core copy  
& whump changes to ECS

other changes made both

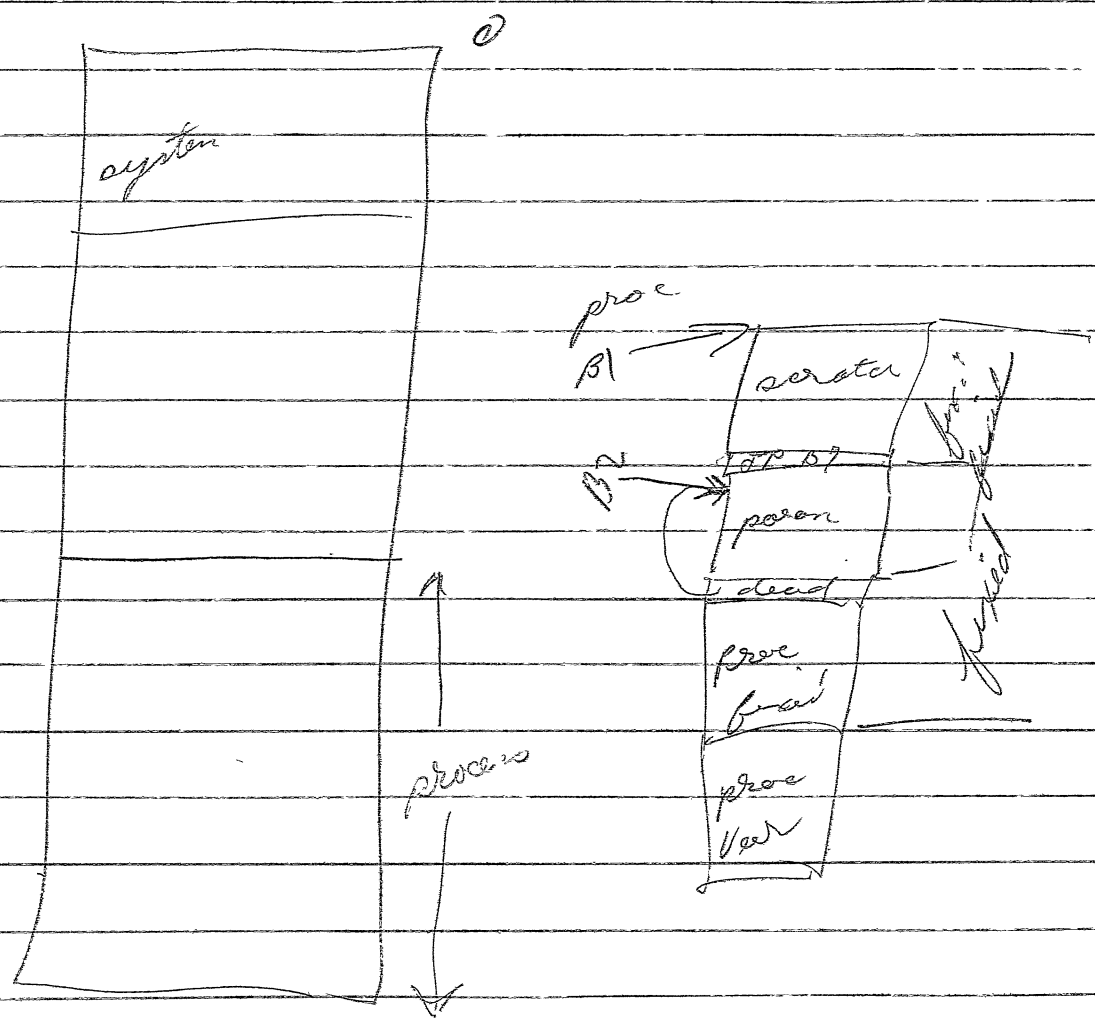
also read in after a  
swap out on timer slice

buffer between proc descriptor & 1<sup>st</sup> subp  
for local C-list → buffer is as big as  
largest C-list - PA of subp relocated  
accordingly

#44 - j, list  
#5 - p, data  
A2 - p, list  
A6 - main list  
~~A4 - C-list~~  
A4 - C-list  
B5 - C-list  
B5 - C-list

⇒ event on event channel + process hung on E.C.

if in system code ; place process on deferred queue and change CEF to user to a TP to scheduler which will schedule deferred queue & replace the CEF



record in Disc sys of every file which exists in ~~both systems~~ <sup>disc sys</sup>  
and ~~has~~ <sup>has</sup> capabilities in ECS sys

entry in master object table remains until last  
capability is destroyed - destroy obj in ECS  $\equiv$  to remove  
from MOT

disc sys present data <sup>to ECS</sup> to create obj file +  
capability  $\&$ : the name from capability is used to  
create new capability for file with new master  
object table index for both if necessary

disc sys. to swap out reduces file to one  
word (empty) - to reconstitute file

if old capability for file is dragged out & ~~given~~  
unique name is in MOT then ECS oper to change  
MOT index in capability

if old unique name not in

NOTE a capability can never go to disc & return  
 $\Rightarrow$  capability list ~~cannot~~ must stay in ECS

Not approved

18 Feb

class codes

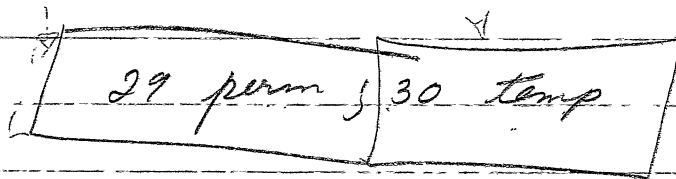
used for

- ① explicitly subprocess
- ② disk sys access and keys

→ use temp part (hand out access code after turning off "set temp" option)

for subprocess "perm" part defines a group of subp  
 "temp" " " a particular subp  
 ↳ used to be subp in use

How divide



ECS actions

perm - cap and key

1) new perm part

2) set temp part (can option bit control this)

3) read

14 Feb

operation

add new PS "any capability"

means type & option bits of capability not checked

TTY input interrupts

① normal (~~go~~ <sup>reboot</sup>) break also sent to root but with different data

② <sup>normal</sup> go to top of <sup>(root)</sup> sub tree if <sup>interrupt</sup> ~~(break)~~ <sub>reboot</sub> root may then do "return with error"

errors

guy who caused error gets 1st crack at it

ESM may be freely reset

when error is accepted - turn off appropriate ESM bit

ESM operation - return with error (remove top of stack & proceed with error processing)