

## CP time cutoff -

Process descriptor has CP time field which is zero @ process creation, can have CP time moved in from alloc. blk, and counts down as process runs. When counts to zero, two new items are used in process descriptor provided @ creation

at process destruction  
CP time reverts  
to owning  
alloc blk

Set at creation  
time? or later?

- a) timeout event channel
- b) timeout datum

When process times out, datum is sent on channel if full <sup>or</sup> no event channel (i.e. set later case) too bad.

Alloc Blk: Space can be moved only between

father and son. Moving space doesn't change

charge rate, which is set separately (and

requires an option bit.) ~~Need to allocate M.O.T.!~~

can increment

time integral. Need to allocate M.O.T.!

A daemon who watches allocation blocks and accounting blocks, creates H<sub>y</sub> processes and receives timeout events.

daemon sends interrupts to his buddy  
in user processes to cope with

- ① process <sup>CP</sup> time ~~limit~~ <sup>cutoff</sup> (move a little  
time into proc. desc. 1st)
- ② allocation block <sup>Fixed ECS</sup> space-time cutoff
- ③ Disk sys accounting block space-time  
(Disk or swapped ECS) cutoff

password - weight - other  
money, money warning - split  
defaults on EBS (fixed, swap) Disk  
standard chunks of CP time, disk space  
time, ECS space-time to prefund.

~~add sys~~

User has user profile - logon gives him

1. His standard Fixed ECS ~~amount~~
2. His standard Swapped ECS
3. His standard