Resource Control & Accounting on Cal TSS

I Resources. Some are controlled strictly on the basis of the user's ability to pay for them; others have artificial ~~limits~~ necessary for efficient operation of the system.

A) CPU time (μs) - ~~scheduling of proces~~ the CP is not discussed here

B) Central Memory ~~(cells)~~

C) PPU time

D) ECS space (words * μs/1024?)
   0) MT slots
   1) fixed
   2) swapped?

E) Disk space 1) permanent (words *?)
   2) temporary

F) Peripheral Equipment
   1) Printer (typeset lines?)
   2) Reader (cards?)
   3) Tape drives (tape mounting; time)
   4) TTY's & connect time?
   5) User screens & keyboards

II Accounting Considerations

A) User profile.

The user negotiates his ~~profile~~ account with the CC business office, resulting in data known as the user profile. The profile contains

1) $ worth of resources he may use

2) ~~$~~ warning $ limit a) written notification?
   2½) weight b) warning at each logon?

3) ΔCP a) ∃ users who can't use more in one session
   b) notification at ΔCP - may cream the stack!

4) fixed ECS ~~limit~~ a) ~~on cells~~ limit
   b) default

II A) 5) ~~temporary~~ swapped ECS?
a) ~~cells~~ limit
b) ~~default~~
c) Δtime * cells

6) permanent disk space. Renegotiated with 201 Campbell

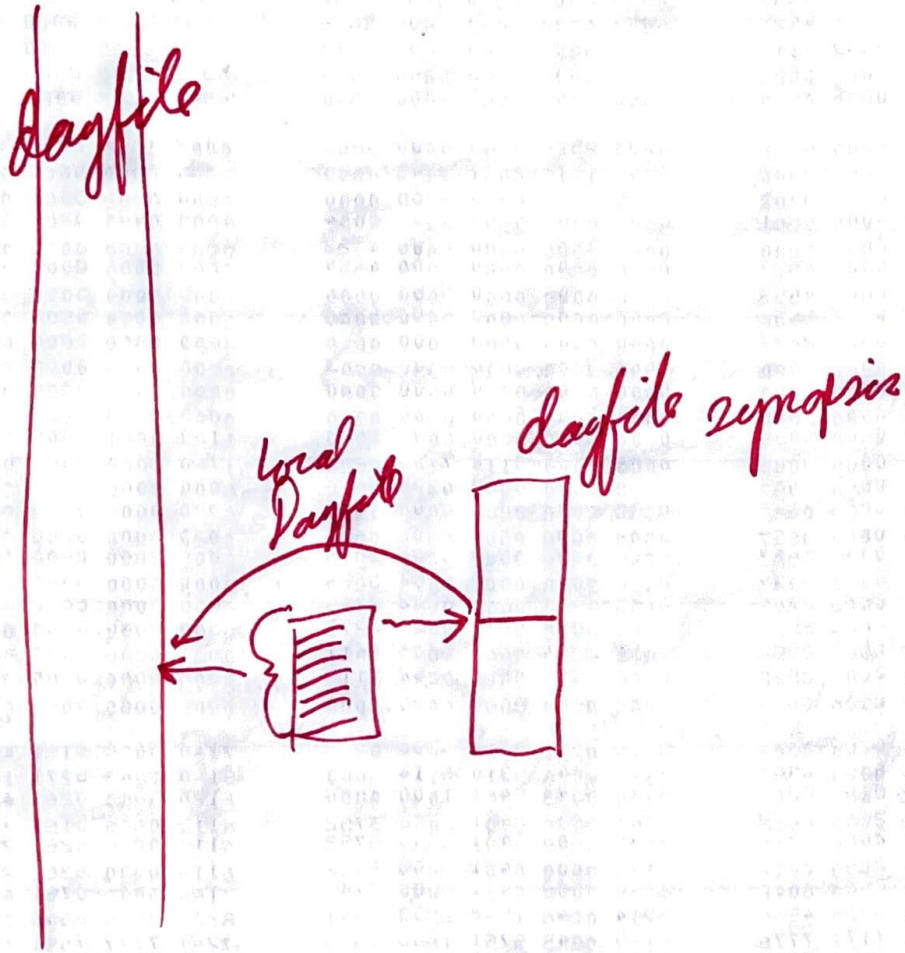
7) temporary disk space? a) limit
b) default
c) Δ

8) access to peripherals
line limit
card limit
tape mounting limit etc

II B) Nature of the users accounting report

1) Accounting points. ~~Certain~~ user interactions with the system which indicate changes in ~~his resource~~ the nature and/or rate of his resource utilization are called a.p. These interactions may be triggered by the user at a keyboard or by his running code. Accounting points are

a) logon
b) changes to ECS profile
c) " " disk "
d) renewal of ΔCP?
e) requests for peripheral service
f) log~~off~~out



how long is the dayfile kept
(& in what form) ??

a local
& global ???

c appropriate

II B) 2. Dayfile. A file is maintained by the system into which entries relevant to system activity are made from time to time. An entry is made at each user accounting point including the real time, userID, & a description of the AP.

3) Users report. At some time interval selected by the CC office (one month?), the dayfile is processed to prepare reports for ~~the~~ each individual user. The report can potentially be incredibly detailed, including every A.P.; details are left to the office to decide.

4) Actual charges are based on the report presented to the user.

5) Approximate account.

x C) General Philosophy of the Charging Algorithm. 1) The system ~~adopts~~ enforce a "pay in advance" attitude. That is, whenever a user is about to utilize a resource, he presents an upper limit ~~on what~~ for the task he is about to do. The system converts this limit to a $ charge & decrements the users ~~total~~ approx. account; if his approx. account is insufficient, the user is so informed & may take appropriate action, i.e. curtailing his request or obtaining more $. If the account is sufficient, & the task is allowed to proceed until it terminates or reaches the limit. In the former case, the users account is refunded the appropriate number of $; ~~in the latter case, the user is ...~~

use its approx. account

II C) 1) (cont) If a user task reaches the limit he specified, he is so informed + is allowed to extend his limit whenever it is practical to do so. If it is impossible to communicate with the user (as in ? + ?? below), the task is, alas, terminated at the specified limit with some hopefully illuminating diagnostic.

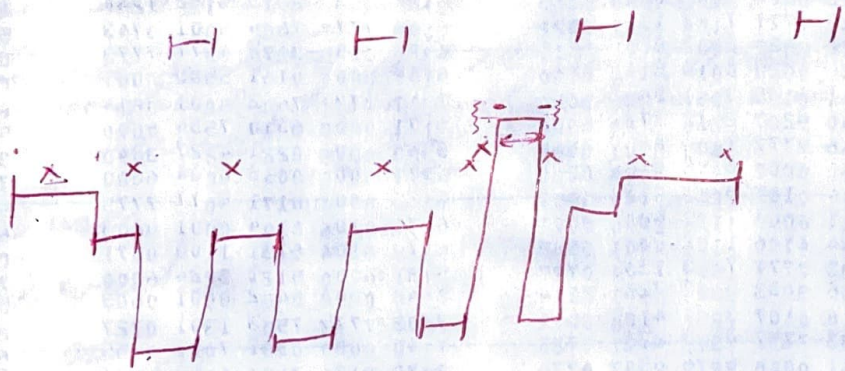
Examples:

a) Δ CP

b) remote file print requests

c) permanent disk storage

44 MR



200 μsec

40 μsec

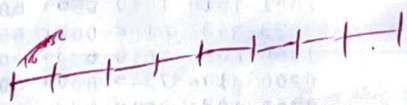
$\frac{1}{37}$

50
32
2500

60

$x_1 + \frac{1}{x_1}$ 60+ 50 3000 60

55



210 μs

40,000
32,512
≈16000

$\sqrt{16000 * \frac{1}{4} + \frac{3}{4}}$

$\sqrt{\frac{16 * 3000}{16}} = \sqrt{3000} = \frac{55}{16000}$ 2 1/2 %

III Mechanisms underlying. Brief def of process + allocation blocks.

A) Time

1) The S.MASTR (S.CHORE) us clock attached to the 6400 ~~[illegible]~~ is read periodically by a PP + its value is set in a cell, S.MASTR?, in central memory. S.MASTR this contains the # us since the system was deadstart + is the basis of all system timing. Because the PP can only update S.MASTR about every 40 us due to other duties the PP must perform, S.MASTR does not advance uniformly. ~~It is not suitable for timing small intervals~~

?

2) The system* (ECS) partitions its own activities into 5? categories, the time for each being accumulated as us since deadstart in ~~a certain~~ some cell in CM.

System clocks

any operation to display system clocks is available to higher level systems + the user. It gives:

a) interrupt (PP) real time

b) user (central)

c) wait

d) swap

e) system

~~f) compaction~~ ?

3) User clocks

Each process* in the system has a set of clocks which record the CP time consumed by that particular process in various categories? An operation allows higher level systems to display ~~these~~ the clocks of any process for which a suitable capability is possessed. The cp gives

a) real

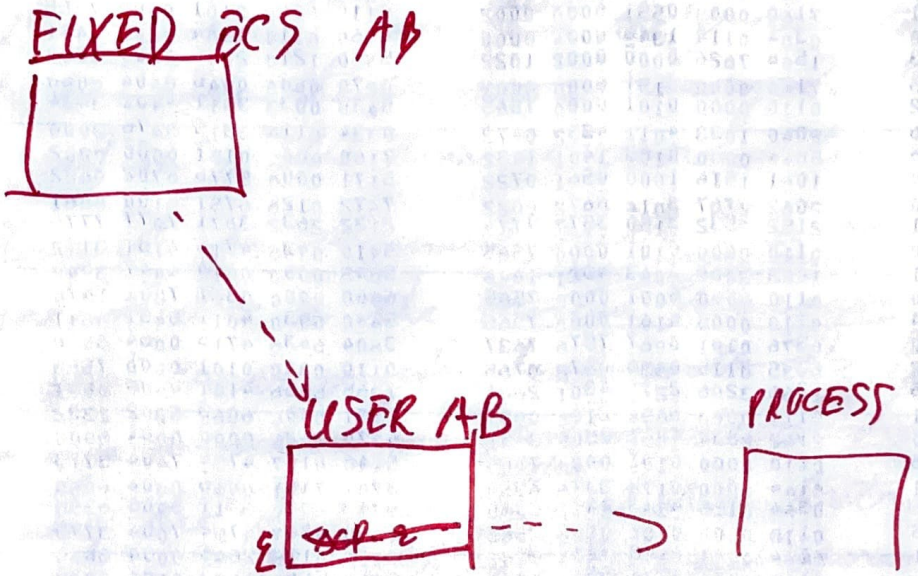
b) user

c) system

d) swap

e) quantum

?



mechanism.

of time Enforcement. Each process contains a timer & a message ~~channel~~

III A) 4) ~~Process timer. Each process in the system contains a timer.~~ Time may be transferred ~~to & from~~ between the process timer & ~~the~~ an allocation block. ~~to which the process is charged.~~ Whenever a process is swapped out, its timer is decremented by the amount of CP time it has consumed; if its timer has run out, it is descheduled + ~~&~~ the ~~Process Manager~~ message ~~interrupt~~ mechanism is utilized ~~is notified by the swapper. Prearranged deals with the Process Manager~~ Avail ~~may determine reactivate a process descheduled for timeout. Otherwise~~ Reproc ~~it will~~ will either lie dormant until it is destroyed or until some other process bails it out. ~~Retrieval operations~~:

(it's ownway)?

The mon. ~~constants~~ provide for an event channel + an event; if present, the event is sent to the PC; if not present, no action beyond descheduling is taken.

a) ~~more time to process~~ Timer operations

i) more time to process
IPI C: AB (OB.TIMP)
2 C: Process (OB.TIM)
3 $\Delta t \geq 0$
error if Δt not available in AB;
if process is currently descheduled for timeout, it is rescheduled;
processes are created with timeout

ii) more time from process
IPI C: AB (OB.TIMP)
2 C: Process (OB.TIM)
3 $\Delta t \geq 0$
if timer $T \leq \Delta t$, process descheduled + caller notified of T (F-return?)

~~* The Process Manager has good vibes (small $\frac{W_i}{r_i}$) + runs next?!?~~
~~Supervisor~~

III A) 4) a) iii) move time to AB

1 C: AB (OB.TIME) from
2 C: AB (OB.TIME) to
3 Δt

error if Δt not avail

iv) install message mechanism

1 C: process (OB.WARN)
2 C: event channel
3 D: event

v) display timer, message mechanism

note: ~~perhaps~~ time moves ~~should~~ will be restricted to son of AB?
when a process is destroyed, its time is given to its owning AB.
when an AB is destroyed, all its resources revert to its owning AB.

~~III A) 5) Process supervisor - hacks on process~~ ~~AH keeps only most recent request for given process~~

~~b)~~ deals with the Process ~~Manager~~ Supervisor:

~~i) send time~~
C: AB
C: process in question
D: Δt

iii) delete request

i ~~ii~~) send interrupt
C: process in question
C: process to recieve interrupt
C: subprocess
D: interrupt datum

ii ~~iii~~) send time + interrupt
i + ii params

note: Process supervisor forgets a request if it gets an error trying to comply (no AB, not enough time)
perhaps it makes a logfile entry?
If its request queue OVs, or every ST in any case, PS probes its requests + deletes ones which are no longer relevant.
If the request queue really OVs, ?
~~P.S. communicates with logfile? Vs duplicate requests?~~

Tree structure on processes?

(interrupt father on timeouts? Xfer time to descendents + ancestors?)

III A) 6) Allocation blocks.

ABs ~~are the memory mechanism~~ serve several purposes:

~~i) limit ECS space usage to what is available~~

~~ii) more specifically, limit the ECS space~~

SPACE RESERVED	SPACE IN USE
PTR HEAD OBJ	PTR TAIL OBJ
TIME OF LAST BILL	SPACE CHARGED
DISCONT.	TIME*SPACE
CONT.	"
CP us available	
CP us used ???	

Every object which uses space in ECS is charged to an allocation block (including allocation blocks; the initial Master Allocation Block is kludged together at system initialization time). Thus BCS objects are arranged in a tree structure with the MAB as the root. ABresources move up & down the tree. Reserved / INuse fields

NOT elts

a) control the amount of space available for different users

b) accumulate information ~~on the~~ for charging for ECS usage - TOLB, charged, DTS, CTS

c) provide a structure on objects occupying ECS ~~so that~~ which will allow cleanup of objects (even when the creator of the objects has run ~~on~~ into problems & lost track of them). Head/Tail Ptrs

d) serve as a repository for CP time - CP us

e) operations (in addition to III A 4a i-iii)

i create

ii destroy

iii display

iv read charges (resets DTS)

v bequeath reserved space to son

C: donor AB

C: son of donor AB

D: Δ space

decrements reserved in donor;

increments reserved & charged in donee

vi revert space to father

C: son

D: Δ space

decrements reserved & charged in son;

increments reserved in father

vii create capability for nth object

viii increment charged space (or steal reserved space?)

~~f) AB supervisor~~

not elt

III A) 7) ~~Accounting blocks~~ Allocation Block Supervisor. It is hoped that ECS is cheap enuf that spying on ABs at an interval of say ≥10 mins is feasible. In any case, ABS runs at regular intervals, say ΔT_{ABS}, & looks at each AB for which it has a request. It sends the prescribed interrupt for each one that has passed its limit & deletes the request. No duplicate requests? (Then user can't spy on his AB.)

III A) 8) Accounting Blocks

IV System use of control & accounting features: A) Logon

~~A CP time.~~ a TTY, the ~~TTY Supervisor~~ system ~~creates~~ locates the user's profile. ~~uses data in it to create~~

~~i An AB with Charged Reserved = fixed ECS~~
~~CPuse = ΔCP − ε~~

~~ii Now swapped ECS ?!?~~
~~iii a process with times = ΔCP − ε, taken from the AB~~

1) The system deletes $ from the user account for:

a) ΔCP
b) 3rd * (fixed ECS * ΔT_{ABS})
c) swapped ECS ?
d) Temporary disk space?

2) If the account is sufficient, the ~~user's~~ system

comes from FIXED ECS AB? ⟷

a) creates an AB, the USER AB, containing
   i Charged = Reserved = fixed ECS from the profile
   ii CPuse = ΔCP − ε
b) creates a process, the USER PROCESS, charged to the USER AB, ~~&~~ sets the process times to ΔCP − ε ~~from~~ taken from USER AB
c) the process then started/reconstructs its system subprocess tree

iii makes a request on the AB Super to watch the user AB for FIXEDECS * ΔT_{ABS} ~~and it~~

iv-begins dialogue with user

ii) makes a request on Process Supervisor to send a time from FIXED ECS AB to the process & interrupt a system subprocess

IV A) 3) If the account is insufficient, the user is ~~so~~ informed of the physical problem. The system may or may not ~~decide~~ bargain with the user to let him make smaller requests to use up his last small amount of time. ~~Such~~

IV B) ΔCP enforcement.

~~1)~~ Whenever a timeout interrupt arrives, the system subprocess responsible for accounting first checks whether the user has requested local control. ~~(see below).~~ Actions taken when local ~~1) If local control is in effect, time~~ control is in effect are described below. When not under local control, the user profile and/or keyboard requests determine what happens

1) Automatic renewal (maybe requested from TTY, ~~but possibly not if~~ allowed by profile). ~~Adjustments are made to~~ The user account is charged for another ΔCP, modified by any remainder of the previous ΔCP not actually consumed. ~~Processing resu~~ If the account is sufficient, the process timer is reset to ΔCP−ε, the request to the process supervisor is renewed, & processing continues. If account not sufficient, physical problem processing.

IV B) 2) Optional renewal. The user is informed of expiration of CP time & allowed to renew, as in 1, or log off.

3) Forced logoff. Some users may only be allowed to use SCP at one sitting?

IV C) Fixed ECS time & space enforcement. When the AB interrupt arrives, The DTS field of the AB is read & the user account is charged for the amount read.

1) If the account is sufficient, a new request is posted with the AB supervisor & processing resumes (thought of as automatic, although user options as IV B 2 are possible).

2) If account not sufficient, the user is informed that he has ΔT_{ABS} to terminate gracefully & a request is posted with the AB supervisor. If the user fails to logout before the interrupt arrives, he is involuntarily logged out.

D) Swapped ECS?

IV E) Disk space being cheap, it can merely be charged for by some crude algorithm. 1) Permanent disk space can be charged for X months in advance. Then when the user's account is depleted, he can replenish his account or preserve his files on tape within X months without dire consequences. The code doing the charging can run infrequently & need only inform the CC business office when the account runs dry.

2) Temporary disk space may be charged ~~in any~~ for in advance in terms of any convenient time interval $\gg \Delta T_{ABS}$. ~~Then it~~ The code needn't detect account deficiency, as ΔCP or ΔECS will inevitably handle the problem shortly.

IV F) Logout. The temporary files are destroyed & the objects attached to the USER AB are deleted from the system and ~~any resour~~ a refund is made to the user account for any of the resources charged for under IV A1 but not yet consumed.

G) Peripheral usage.

1) Remote printing, etc.

2) Use of printers, tape, etc.

V User access to control & accounting features. Facility for the user to control the CP time and/or ECS and/or disk space used by some subtask initiated during his run is fairly general, & hopefully convenient. The mechanisms described in III are available to him with only modest restrictions.

A) CP time

1) Suspicious subsystem usage. If about to call a subsystem which he does not trust, he may request an interrupt after Δt sec of processing ~~for~~ either from the TTY or by a system call from his running code. The system will thereupon ~~move the timer in~~ move all but Δt of the USER process times back to the USER AB & note that the process is under local control.*

If & when the interrupt arrives, the process times is restored & the user informed & presented with the option of continuing (with or without a new Δt) or terminating.

2) Satellite processes

* local control $\Leftrightarrow$ CP time in USER AB?

① process
Time runs out
find 1st owning allocation
block that is "special"
lend it to current to designated
process to suspend
deallocate the run of process.

in an allocation block that is special
a process to access more than one current address

also some allocation blocks are "marked"
They are chained.

①

process

Time runout

find 1st unused allocation
block matrix "special"
lead interrupt to designated
process & suspend

descendde the run of process.

in an allocation block matrix special
a process & a class are & an instrument datum

also some allocation blocks are "marked"
They are chained.