Identification

The Traffic Controller Data Block
Robert L. Rappaport, Michael J. Spier, A. Evans

Purpose

The Traffic Controller Data Block (TCDB) is a collection of miscellaneous items in segment <tc_data>; some of those items are internal to the Traffic Controller, others are of general interest. This section lists all these items and explains their function. No table-declaration is furnished as this is a pre-assembled table and the items are independent of one another; rather, the EPL attributes of each of these items are stated separately.

Itemization

Following are the TCDB items listed by their symbols; to access any of them, say apt_lock, reference should be made to <tc_data>[apt_lock]; "*" means internal to the Traffic Controller.

- **apt_lock**
  - global-lock word used to restrict access into the traffic controller (fixed)
- **pwt_lock**
  - the PWT is the only table that is sometimes accessed without using the global traffic controller interlock; the PWT lock is then used (fixed)
- ***apt_ptr**
  - pointer to the base of the APT (ptr)
- ***apt_size**
  - number of words in the APT (fixed)
- ***pwt_ptr**
  - pointer to the base of the pwt (ptr)
- ***pwt_size**
  - number of words in the pwt (fixed)
- ***aht_ptr**
  - number of words in the APTHSH (ptr)
- ***aht_size**
  - number of words in the APTHSH (fixed)
- **lowest_level**
  - is the lowest-priority (highest-numbered) level number in the ready-list; it is assigned to the idle processes exclusively. (fixed)
lowest_loaded  number of lowest-priority ready-list queue in which a loaded process may be found. Subroutine unload-test uses this variable to determine which process to unload. (fixed)

neligible  number of eligible processes in the system (fixed)

max_eligible  maximum number of eligible processes the system can accommodate. (fixed)

nloaded  number of processes currently loaded, this variable is maintained by the load-daemon (fixed)

nloaded_ready  number of ready and loaded processes (fixed)

max_loaded  maximum number of loaded processes the system can accommodate (fixed)

loaded_depth  the number of processes that must be kept loaded at the head of the ready-list. (fixed)

nprocessors  number of processors currently used (fixed)

activation_level  a constant level number given a newly created process; when rescheduling itself, it will give itself an appropriate level number. (fixed)

wait_enable  a system flag, initially set to zero which is set to 1 after Traffic Controller initialization. (fixed)

loader  pointer to the TC Loader Process’ APT entry (ptr)

loader_id  TC Loader Process’ ID (fixed bin(35))

initializer_id  initializer process’ ID (fixed bin(35))

highest_ready  relative pointer to the topmost process on the ready list; used by the pre-emption mechanism. (fixed)

time_quantum  a system constant; the timer register is never loaded with a value that is higher than time-quantum. A process’ time allotment is used-up as a series of "time-quantas". (fixed)

system_shutdown  a flag to indicate, when "on", that a system shutdown is in progress; this condition "freezes" the APT and prevents further process activation from taking place. (fixed)
hard_core_ring_number  self explanatory  (fixed)

template_dseg_length    (also named dscopyl) the length of
                        the descriptor segment template.  
                        (fixed)

template_pds_length    (also named pdscopyl) the length
                        of the PDS template  (fixed)

dsmsl       descriptor segment maximum length  (fixed)
pdsmsl      PDS maximum length  (fixed)
id          system constant used to generate process IDs. 
            This is a clock reading taken during system
            initialization time.  A process id is generated
            by simply incrementing, then reading of this
            variable.  (fixed bin(35))

lock         lock_word associated with the above-mentioned
            id to allow only one processor at a time to
            manipulate it.  (fixed)

running_list an array of entries, one entry per processor, 
            of the following format:

            dcl 1 running_list(8),
                2 apt_ptr bit(18),  /*rel ptr to APT entry*/
                2 pre empt bit(1),   /*flag*/
                2 filler bit(17);

            the array's size corresponds to the maximum
            number of processors possible; each entry
            corresponds to a processor by index number.

            (the term "sentinel" used below is explained in BJ.1.02)

*empty_q      the first of two contiguous empty-list
               sentinels

*ready_q      the first of an array of ((lowest_level) +1)
               sentinels of the ready-list.

            (while empty_q has pre-assembled values, the ready_q
            array is initialized by subroutine tc_data_init
            together with the five tables)

*level_coefficient  an array of entries, one per level, 
                     which is used by the scheduler's pre-emption
                     algorithm (fixed)
*loaded_list  list of loaded processes; this is an array of size (max_loaded) which contains pointers to the APT entries of all loaded processes.

pwt  pre-initialization pwt (fixed)

**Initialization**

The pre-assembled segment <tc_data> is loaded into core from a Multics System Tape as one of the very first items in the system, and is wired down. At this point, most of the above-mentioned items are pre-set to either zero (for fixed variables) or null (for pointer variables) values.

The exceptions are the two empty_q and the three block_q sentinels which already have their definite values, and the items associated with the pwt. As it is, the Basic File System is engaged in PWN activity long before Traffic Controller initialization; this is recognized by wait_enable's zero-value. However, rather than encumber all four PWN subroutines with conditional statements it was found more elegant to provide a dummy PWT during initialization time, and thus execute most of the actual PWN code (stopping short of giving the processor away, in wait.) Consequently, pwt_size is pre-set to 1, and pwt_ptr points to symbol pwt, thus providing a temporary one-word-long PWT.

Subroutine tc_data_init allocates areas in <tc_data> for the five tables, copies system parameters from the Software Parameter Table (SWPT) into appropriate TCDB items, and initializes all pointers to tables as well as the "level-coefficient" and "loaded-list" arrays.