

file -  
MSPM

6-1-70

Identification

Traffic Controller Data Bases

S. H. Webber

Introduction

The traffic controller data bases contain all information needed by the traffic controller. Some of this information is stored in tc data which is a hardcore, system-wide <sup>(shared)</sup> segment in the address space of every process. Other data is stored in the process data block of a process and is only accessible by that process. Most traffic control information about a process which is needed by other processes is stored in tc\_data.

The Structure of tc data

tc\_data consists of four sections. These are:

- 1) the traffic controller data block,
- 2) the active process table,
- 3) the inter-process transmission table (ITT), and
- 4) the device signal table.

The traffic controller data block contains system wide constants and variables of interest to the traffic controller. Some of the more important variables are described briefly below:

lock            The lock is an interlock over all critical data within tc\_data.  
In general, no process will ever reference tc\_data without

first setting this lock (by looping until it can be set). That is, no two processes will ever reference `tc_data` at the same time. Every traffic controller procedure respects this lock.

`max_eligible` This variable limits the number of processes which may be eligible at one time.

`tefirst` A scheduling parameter. The amount of cpu time for which a process is guaranteed to remain eligible (if it needs to) the first time it runs after an interaction. See Section BJ.2 of this manual.

`telast` A scheduling parameter. The amount of cpu time for which a process is guaranteed to remain eligible when it is in the last scheduling queue. See Section BJ.2 of this manual for a description of "last scheduled *waiting queue*".

`timax` A scheduling parameter. The default amount of cpu time a process remains in the last scheduling queue before being rescheduled (at the end of that queue).

*see above*  
controller. It consists of an active process table entry (APT entry) for each process known to the traffic controller. Within the APT entry for a process exists all the information specific to that process which is needed by the traffic controller. The traffic controller can access this data while executing in any process. Among the variables stored in the APT entry for a process are:

- 1) scheduling parameters,
- 2) characteristics of the process (state, etc.), and
- 3) process variables of interest to the traffic controller.

A brief description of some of the more important items within a process's APT entry is given below:

processid

The processid of a process is the name of a process as far as the traffic controller is concerned. When one process wants to refer to another process, it passes the traffic controller the processid of the target process which is used in a look-up algorithm to find that process's APT entry. A processid is unique for all time at a given Multics installation.

multi-programming state

The multi-programming state of a process is the value of two switches indicating first whether the process is eligible and second whether the process is loaded.

execution state

The execution state of a process indicates whether the process is (1) running, (2) ready, (3) waiting, (4) blocked, or (5) stopped.

dbr value

The dbr value is used by the traffic controller when switching from one process to another. It is the value placed in the "descriptor segment base register"

which defines the address space of the process.

wait event

The wait event of a process is the name of the event for which a process is waiting. If the wait event is zero for a given process, the process is not "waiting" for any event.

ITT list pointer

The ITT list pointer points to a threaded list of ITT entries relevant to this process. For details of ITT management see Section BJ.4 of this manual.

wakeup waiting

The wakeup waiting flag for a process indicates that the process has been sent a wakeup which it has not yet processed. The next time the process goes blocked any unprocessed ITT messages for the process are passed on to the inter-process-communication facility for handling and block immediately returns (after resetting this flag).

The following variables within each APT entry are used for scheduling purposes. For details of the scheduling algorithm see Section BJ.2 of this manual.

- te The cpu time a process has run since last gaining eligibility.
- ts The cpu time a process has run since being scheduled. That is, the time since the process's relative priority changed.
- ti The cpu time a process has run since it last interacted, i.e., since it was last scheduled with its per-process interaction indicator ON (in general a higher ti means lower priority).
- temax The amount of cpu time for which a process may remain eligible, i.e., the eligible time. (A variable between tefirst and telast.) Note, the hardware timer register is set to "go off" whenever temax is exceeded. This <sup>forces</sup> the process running to invoke the traffic controller to reschedule itself.
- tymax The amount of cpu time a process may run before being rescheduled. (A variable between tefirst and tymax). This variable is actually not kept in the APT entry but rather is derived from other variables in the APT entry when it is needed (See BJ.2). It is, however, useful to treat it as a separate variable.

timax

A parameter which defines the "last scheduling queue" for this process. See section BJ.2 of this manual.

#### The Eligible Queue and the Priority Queue

The multi-programming control feature limits the number processes which may be eligible. These processes are the ones which currently have the privilege of using core memory and hence should be awarded high priority. In fact, before a process can be run it must be loaded which in turn requires that it be eligible. All eligible processes are placed in what is called the eligible queue. This is a first-in-first-out queue of all eligible processes. The highest-priority (ineligible) process is placed at the tail of this queue whenever the number of processes which are eligible drops below max eligible. When a process loses its eligibility for some reason, it is removed from the queue (even if it is not at the head of the queue).

The eligible queue consists of all running and waiting processes and those ready processes which are eligible. When the traffic controller searches for a process to run it searches the eligible queue first (from the beginning). If it finds a process in the eligible queue which is loaded and which is not waiting for a system event the process is run. If no such

process is found the traffic controller searches what is called the priority queue. The priority queue is an ordered list of all processes which are ready but not eligible. If the traffic controller finds that the number of eligible processes is less than max eligible the first process in the priority queue is awarded eligibility and placed at the tail of the eligible queue. If the process is not loaded the appropriate call is made to get the process loaded.

The queues consist of one or more APT entries threaded together. In fact the head of the priority queue is threaded to the tail of the eligible queue making certain implementation details work out easily.