Published: 11/25/66

Identification

The Active Process Table
J. H. Saltzer, R. L. Rappaport

Purpose

The Active Process Table is a system-wide data base, accessible to and maintained by the modules of the Traffic Controller. It contains an entry for every process in the system which is active. An active process is one which meets one or more of the following conditions:

1. It is running.
2. It is ready.
3. It is responsible for a system interrupt.
4. It is waiting for a page to come in.
5. It is being loaded.
6. It is blocked but has not yet been completely paged out of core.

The Active Process Table must remain in core memory at all times, since it is the primary source of information for scheduling during system interrupts. A characteristic of an active process is that enough information is stored in the Active Process Table that the process can be scheduled and switched to running status without getting any page faults.

An inactive process must be placed in the Active Process Table ("activated") before it is possible to send it a wakeup signal. Activating a process involves copying appropriate information about the process from the Known Process Table, which is paged, and making the Known Segment Table for the process an active segment. (This allows the process to retrieve its own pages.) Both of these actions generally require access to paged tables and directories.

A process is deactivated by the basic file system when, while operating for some other process and looking for core space, the file system notices that the descriptor segment has fallen into disuse. If the process in question does not meet one of the first five conditions above, its descriptor segment is discarded and its Known Segment Table is paged out and deactivated (see section BG.2). The process is then removed from the Active Process Table to the Known Process Table.

Contents of the APT

The Active Process Table contains a table interlock, needed to insure orderly access and modification of the table, and the following information for each active process:

1. Process Identification number. This is a serial number used to identify the process within the Traffic Controller.

2. Descriptor Segment Base Register Value. This is an absolute core address, the base of the hardcore ring descriptor segment for this process. If the process not loaded switch is on, this entry is meaningless.

3. Process not loaded switch. If this switch is on, the descriptor segment for the process has been paged out of core memory, and special action is needed to switch control to the process.

4. Process loading switch. If this switch is on, this process is attempting to retrieve its process data block, and must not be unloaded.

5. Process time limit, scheduler imposed. This is the maximum time that the process should be allowed to run in a single shot without the scheduler reconsidering its priority. This time limit will generally be re-evaluated each time that the process schedules itself.

6. Time last run. This item contains the calendar time at which this process last left running status. This information is used to help select candidates for unloading and deactivation.

7. Running Switch. This switch is set on whenever the process is running.

8. Ready Switch. This switch is set on whenever the process is ready. If neither the Running nor Ready switches are on, the process is blocked.

9. Process Segment Table pointer. For a process which is unloaded but still active, this pointer represents the end of a (long) string which, if pulled hard enough, will retrieve enough of the process to allow it to execute.

10. Processor Currently in use. This item only has meaning if the Running Switch (7 above) is on. It contains the identification number of the processor being used by this process. This information is needed, for example, to interrupt the process.

11. Processor number required. Certain processes have sufficient authority to demand that they only run on a specific processor. Such a requirement is indicated by this entry being non-zero. This service is provided for test and diagnostic purposes only.

12. Ready list Pointers. The ready list is a thread through the active process table. If this process is ready (switch 8 above), these pointers link the ready list forward and backward. If this entry in the Active Process Table is empty (13, below) these pointers link the empty entry list forward and backward.

13. Empty Entry Switch. If on, this process entry is empty, and the pointers of item 12, above, link the APT empty list.

14. Wakeup Waiting Switch. If a call to Wakeup comes in this switch is set on. If a process calls Block and this switch is on, the switch is reset and Block returns immediately.

15. Waiting for Hardware Switch. If this switch is on, this process is responsible for some system interrupt and cannot be deactivated.

16. Waiting for Page in Switch. If this switch is on, this process is waiting for a page to come in from secondary storage and cannot be deactivated.

17. Current Priority. This is the process' priority as computed by its own personal scheduler. It is examined by other process' schedulers to determine the appropriate point to insert them into the ready list.

18. Wakeup-lock. If on, some process is in the midst of trying to wakeup this process. Another process desiring to wakeup this process need not proceed.

19. Quit-waiting switch. This switch, if on, indicates another process is trying to "quit" this process.

20. The intermediate-state switch. This switch, if on, indicates the process may be "executing" on a processor regardless of the process state as defined in its ready and running switches.

21. The process-chosen switch. This switch, if on, indicates that this process has been chosen to run by another process operating in getwork, but that it has not yet begun to run. Getwork in choosing subsequent processes will pass over this process.

22. The process-state lock. Processes referring to or changing the settings of this process' running or ready switches must set this lock prior to accessing these data items. Other processes wishing to access this data must wait for the process-state lock to be reset before going ahead.