

## SECTION FIVE: TRAFFIC CONTROLLER--WRAPUP

This section wraps up several ideas which for clarity have been sidestepped in the earlier sections.

### Process Execution Timer Simulation.

The Traffic Controller simulates for every process an execution timer clock identical to the actual processor interval timer except that it only ticks when the process is running. We will name this clock the process execution timer. Two calls from the process are accepted by the Process Timer Module:

Call Load-Timer(time, Process-I.D.)

and

Call Store-Timer(T)

Load-Timer loads the simulated execution timer with the value "time" and saves the process I.D. specified as the process to signal if the timer should run out. Store-Timer returns the time left on the simulated timer as the value "T". If the simulated timer should run out, the process will be interrupted, the specified process signaled, and this process will be blocked.

The process execution timer is simulated by a module of the Traffic Controller which shares the usage of the actual processor timer with the scheduler. The Process Timer Module contains the two entry points mentioned above, and a third called from within the Traffic Controller to determine hardware timer settings. The simulated timer is kept as an entry for this process in the Active Process Table. In those cases when the Traffic Controller must reload the actual processor timer register with a new value it calls

Call Reload-Timer(Stored-Time)

where "Stored-time" is the value that the interval timer had when this process entered the Traffic controller.

The Reload-Timer entry in the Process Timer Module performs the following functions:

1. Compute the time used by this process by subtracting Stored-Time from the "Initial Timer Value" which appears in the Running List.

2. Subtract the time used from the process execution timer limit.
3. Compare the time now remaining on the process execution timer with the scheduled time limit, and store the smaller of these two limits in "Stored-Time" and in the Active Process Table as the "Initial Timer value."
4. If the process execution timer limit was smaller, set a switch in the Active Process Table to show that the execution timer is working for the process, not the scheduler.

The value computed by Reload-Timer may now be loaded into the processor timer register. (We have ignored here the mechanics of accounting; in particular the charging of the time spent in the Traffic Controller, to simplify the discussion.)

On time-out interrupts, the System Interrupt Interceptor time-out handler must distinguish time-outs ordered by the process from those ordered by the scheduler. It can do so by checking the Active Process Table switch mentioned above. If off, this is a scheduled time-out and the Restart entry is called (see Section 4.) If the switch is on, this timer runout is a runout of the simulated process execution timer. A call is made to entry point Wakeup, specifying the process listed. The last step is to unmask the processor and call entry Block in the Process Exchange.

#### Interrupt Masking in the Traffic Controller.

The discussion in section four of the procedures for internal interrupts did not specify in detail the interrupt masking and unmasking that must take place for time-out and pre-emption interrupts. It also did not point out special problems in managing the interval timer. The reason is that internal interrupt masking and timer management are automatically taken care of in the interfaces to and from the Traffic Controller. The reason why this must be so is that an entry to Restart generally causes the processor to be given to some other process. The other process wants neither a masked processor nor any waiting internal interrupts which are hiding behind the masks but which belong to the previous process.

On all entrances to the Traffic Controller, whether system interrupts or calls to Block, Quit, or Wakeup, the interval clock interrupting mechanism is disabled, the contents of the interval timer noted, and the timer is reset to its maximum value. The processor interrupt mask is saved, and at least internal interrupt cells are masked. On all exits from the Traffic Controller, whether return from an interrupt or return from entry point Block, Quit, or Wakeup, the interval timer is reloaded to an appropriate value; the interval timer interrupting mechanism is turned on again, and the processor is appropriately remasked or unmasked.