Identification

Process Execution Timer Simulation
J. H. Saltzer, J. L. Bash

Purpose

This section describes timer simulation by the traffic controller to determine process running time.

Introduction

For every process, the traffic controller simulates an execution timer identical to the actual processor interval timer, except that it ticks only when the process is running. The clock is called the process execution timer.

Calls to the Process Execution Timer

The process execution timer accepts two calls from the process:

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

and

Call Store-Timer(T)

Load-Timer loads the simulated 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 traffic controller procedure that shares the usage of the actual timer with   Schedule . The process execution timer contains a third entry point, 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. When the
traffic controller must reload the actual processor timer register with a
new value it makes the following call:

Call Reload-Timer(stored-Time)

where "$tored-time" is the value that the interval timer had when this
process entered the traffic controller.

The Reload-Timer entry in the process execution timer performs the following
functions:

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

2. Subtracts time used from the process execution timer limit.

3. Compares time now remaining on the process execution timer with
   the scheduled time limit, and stores 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, sets 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.

Time-Out Interrupts

On time-out interrupts, the Interrupt Interceptor time-out handler must distin-
guish time-outs ordered by the process from those ordered by Schedule. 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. 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 point, Block, in the Process Exchange.