CONTROL OF CPU TIME

Two new features are being implemented for processes, a timer and an associated message mechanism. Time may be moved between the CPU time field of an allocation block and the timer of any of its owned proceses. When a process is swapped out, its timer is decremented by the [redacted] time it just used, and if the result is negative, the process is descheduled; the negative residual sits in the timer.

The message mechanism, which is set by a separate system action, consists of an event channel and an event. When a process is descheduled, if the message mechanism is set, the swapper sends the event on the event channel (any errors, such as event channel gone or full, are ignored). If the message mechanism is not present, nothing further is done.

The operation which moves time into a process timer increments the current timer. If the process is descheduled and the timer goes positive, the process is rescheduled.

I heavily favor creating processes descheduled, but it is not too late to argue for an additional parameter on process creation to initialize the timer (that is the only alternative that I can see). A graceful phase-in will beprovided in any case.