Ada Issue 00357 Support for Deadlines and Earliest Deadline First Scheduling

!standard D.02.6 (01) 05-03-13 AI95-00357/09
!class amendment 03-09-27
!status Amendment 200Y 04-12-02
!status ARG Approved 10-0-0 04-11-20
!status work item 03-09-27
!status received 03-09-27
!priority Medium
!difficulty Medium
!subject Support for Deadlines and Earliest Deadline First Scheduling

!summary

Direct support for deadlines is defined and a new dispatching policy for Earliest Deadline First (EDF) Scheduling is supported.

!problem

Ada does not give any direct support for deadlines (although they are the most important notion in real-time systems). A standard way of representing deadlines via a predefined attribute of a task is needed.

Once deadlines are supported it is possible to define a dispatching policy for EDF scheduling. This being as common an approach as fixed priority scheduling for real-time system. It has the advantage that higher levels of resource utilization are possible.

For many scheduling schemes, including for example EDF, the most effective locking policy for protected objects is one known as the Stack Resource Policy (or preemption level locking). This was defined by Baker in 1991 [1] and has the advantage that it does not enforce the same policy on task dispatching and PO locking; but it does lead to effective implementation of POs. Moreover when priority dispatching is used the SRP policy is the same as ceiling locking.

!proposal

To give support for EDF scheduling with the necessary protocol for sharing protected objects requires the following:

1. a means of representing a task's (absolute) deadline
2. a means of assigning a preemption level to a task
3. a means of assigning a preemption level to a PO

The basic rule for preemption is that a new task T will preempt current running task S if:

1) Deadline of T is before that of S
2) Preemption level of T is higher than preemption level of any locked PO

If preemption levels are assigned (by the programmer) using relative deadlines for each task then a tight upper bound on blocking equivalent to that of priority ceiling inheritance is obtained.

In this proposal

a) deadlines are represented by a new task 'attribute'
b) preemption levels for tasks are represented by base priorities
c) preemption levels for POs are represented by ceiling priorities

The usual behavior of priorities, when tasks execute within a PO, are followed. But the active priority of a task when executing outside a PO may be lower than its base priority (see details below).

A new pragma is provided to allow a task to set a (non-default) relative deadline to control its activation:

pragma Relative_Deadline(expression);
where the expected type of expression is Real_Time.Time_Span.

The pragma can only occur in the specification part of a task.

To support EDF scheduling a new Priority_Policy identifier is defined: EDF_Across_Priorities.

When the Priority policy EDF_Across_Priorities is in effect the following rules apply. Let Base(T) be the base priority of task T and Deadline(T) its absolute deadline.

Rule 1
All ready queues in the specified range System.Any_Priority are ordered by deadline. For two tasks on the same ready queue, S and T: Deadline(S) < Deadline(T) implies S is closer to the head of the queue than T.

Rule 2
When a task T becomes unblocked it is placed on the highest priority ready queue R such that
  A protected object with ceiling priority R is currently locked,
  Deadline(T) < Deadline of the task locking this protected object, and
  Base(T) > Priority level of R.
if no such R exists then add T to Any_Priority'first.

Rule 3
When a task is chosen for execution it runs with the active priority determined by the ready queue from which the task was taken. If preempted it returns to the ready queue for its active priority. If it inherits a higher active priority it will return to its original active priority when it no longer inherits the higher level.

Rule 4
When a task executes a delay_statement that does not result in blocking it is added to the ready queue for its active priority.

Rule 5
Priority ceiling level Priority'first is not allowed.

!wording

D.2.6 Earliest Deadline First Dispatching