Ada Issue 333 Additional Locking_Policies with FIFO_Within_Priorities

!standard D.02.02 (05) 04-11-09 AI95-00333/06
!standard D.02.02 (03)
!class binding interpretation 03-07-25
!status Amendment 200Y 04-09-27
!status WG9 approved 04-11-18
!status ARG Approved 8-0-1 04-09-18
!status work item 03-07-25
!status received 03-02-34
!qualifier Clarification
!priority Low
!difficulty Easy
!subject Other Locking_Policies can be used with FIFO_Within_Priorities

!summary
Other Locking_Policies can be used with FIFO_Within_Priorities. However, Ceiling _Locking must work with FIFO_In_Priorities. Moreover, similar rules apply to Round_Robin_Within_Priorities (AI-298) and Non_Preemptive_FIFO_Within_Priorities (AI-355).

!question

Why does D.2.2(5) require that Ceiling_Locking be used whenever FIFO_Within_Priorities is used?

If an implementation supports another locking policy, why shouldn't it be allowed to combine that policy with FIFO_Within_Priorities?

!recommendation
(See Wording.)
!wording

Delete D.2.2(5).

Add after D.2.2(13):

Implementation Requirements

An implementation shall allow specifying both the task dispatching policy as FIFO_Within_Priorities and the locking policy (see D.3) as Ceiling_Locking for a single partition.

Similar changes apply to Non-Preemptive dispatching (AI-298) and Round-robin (AI-355).

!discussion

There is no identified reason for this restriction.

While analysis of a real-time program requires the use of well-defined locking and dispatching policies, such an application should already be specifying the policies that they are assuming.

Moreover, if an implementation provides its own well-defined policy, why shouldn't it be allowed to combine that policy with FIFO_Within_Priorities? There is no value in forcing an implementation to support a second name for the same dispatching policy just to be able to use a different locking policy.

Finally, an implementation's default policy should be selected by its user's requirements (such as performance or compatibility with a target OS), not by the standard. The predefined policies may not be the best on a given target, and forcing users to specify some implementation-defined policy to get the best performance is just over specification (and also makes code less portable).

However, there is a benefit to insuring that Ceiling_Locking can be used with FIFO_Within_Priorities, so that carefully designed systems can be ported to new targets. Such insurance can cleanly be accomplished with an Implementation Requirement. Thus, we've adopted the Implementation Requirement given above to replace D.2.2(5).

!corrigendum D.2.2(04)
!comment This is a fake to trigger conflict processing with AI-355. The real !comment change is in conflict text.
@drepl
A Task_Dispatching_Policy pragma is a configuration pragma.
@dby
A Task_Dispatching_Policy pragma is a configuration pragma.

!corrigendum D.2.2(05)

@ddel
If the FIFO_Within_Priorities policy is specified for a partition, then the Ceiling_Locking policy shall also be specified for the partition.

!corrigendum D.2.2(13)

@dinsa
In addition, when a task is preempted, it is added at the head of the ready queue for its active priority.
@dinst
@i<@s8<Implementation Requirements>>

An implementation shall allow specifying both the task dispatching policy as FIFO_Within_Priorities and the locking policy (see D.3) as Ceiling_Locking for a single partition.

!corrigendum D.2.4(01)
!comment This is a fake to trigger conflict processing with AI-298. The real !comment change is in conflict text.
@dinsc

@i<@s8<Implementation Requirements>>

An implementation shall allow specifying both the task dispatching policy as Non_Preemptive_FIFO_Within_Priorities and the locking policy (see D.3) as Ceiling_Locking for a single partition.

!corrigendum D.2.5(01)
!comment This is a fake to trigger conflict processing with AI-355. The real !comment change is in conflict text.
@dinsc

@i<@s8<Implementation Requirements>>
An implementation shall allow specifying both the task dispatching policy as Round_Robin_Within_Priorities and the locking policy (see D.3) as Ceiling_Locking for a single partition.

!ACATS test
There is no test for the deleted rule (D.2.2(5)). ACATS tests CXD2001.A (and 7 others) test (as a side-effect) that specifying both is allowed. No further tests are needed.