SYSTEM_CLOCK Fixup

by Craig T. Dedo
May 15, 1997

1. Rationale

Currently, the Fortran standard requires that all three arguments to the SYSTEM_CLOCK intrinsic subroutine be of type default INTEGER. While the INTEGER data type is acceptable for COUNT and COUNT_MAX, it causes serious accuracy problems for the COUNT_RATE argument. This shortcoming unnecessarily restricts the available precision.

The processor clocks of many machines, most notably the clocks of Intel 80x86 PCs, have significant fractional components in their clock rates. On Intel (MS-DOS PC) machines, the clock ticks approximately 18.20648193 (1,193,180 / 65,536) times per second (Norton and Wilton 1988, 143-145) (Norton, Aitken, and Wilton 1993, 174-178). Requiring COUNT_RATE to be INTEGER causes considerable loss of accuracy for such machines.

Interpretation 132 attempted to resolve this problem but it does not resolve the central deficiency. In the Intel case, the interpretation allowed all three of these practices:

1. Increment COUNT by 1 every 0.05495 seconds but return a COUNT_RATE of 18.
2. Increment COUNT by 10 every 0.05495 seconds and return a COUNT_RATE of 182.
3. Increment COUNT by a variable amount every 0.05495 seconds and return a COUNT_RATE of 100.

All three of these practices are, at best, counter-intuitive and not the behavior an ordinary programmer would expect. Most programmers expect COUNT_RATE to be a very close approximation to the actual system clock count rate.

One of the most attractive features of Fortran is that it is generally intuitive and straightforward. There are relatively few traps and "gotchas" which cause problems for programmers who have less than expert level of ability. J3 should eliminate such problems whenever possible.

2. Technical Description

This proposal extends the functionality of the SYSTEM_CLOCK subroutine so that it can accept non-default kinds of integers in all three arguments and can accept a generic real argument for COUNT_RATE.

3. Edits to the Standard

These edits are with respect to the Fortran 95 Committee Draft, X3J3 / 96-007r1.

[270:21-44] Replace section 13.14.106 with the following:

13.14.106 SYSTEM_CLOCK ([COUNT][, COUNT_RATE][, COUNT_MAX])

Description. Returns numeric data from a real-time clock.

Class. Subroutine

Arguments.

COUNT (optional) shall be scalar and of type integer. It is an INTENT (OUT) argument. It is assigned a processor-dependent value based on the current value of the process or clock, or -HUGE (COUNT) if there is no clock. The processor-dependent value is incremented by one for each clock count until the value COUNT_MAX is reached and is reset to zero at the next count. It lies in the range 0 to COUNT_MAX if there is a clock.

COUNT_RATE (optional)
shall be scalar and of type integer or real. It is an INTENT (OUT) argument. It is assigned a processor-dependent approximation to the number of processor clock counts per second, or zero if there is no clock.

COUNT_MAX (optional)
shall be scalar and of type integer. It is an INTENT (OUT) argument. It is assigned the maximum value that COUNT can have, or zero if there is no clock.

Example. Assume that the processor clock is a 24-hour clock that registers time at 1,193,180/65,536 or approximately 18.20648193 ticks per second. At 11:30 A.M. the reference

CALL SYSTEM_CLOCK (COUNT = C, COUNT_RATE = R, COUNT_MAX = M)

defines C = (11 * 3600 + 30 * 60) * 18.20648193 = 753748; R = 18.20648193;
and M = 24 * 3600 * 18.20648193 - 1 = 1573039

4. References

ISO/IEC 1539-1:1991(E) *International Standard Programming Language Fortran 90*, 13.13.104.

Norton, Peter and Richard Wilton. 1988. *The New Peter Norton's Programmer's Guide to the IBM PC and PS/2*. Redmond, WA: Microsoft Press.

Norton, Peter, Peter Aitken, and Richard Wilton. 1993. *The Peter Norton PC Programmer's Bible*. Redmond, WA: Microsoft Press.

X3J3 / 94-160, Results of Corrigendum 2 Ballot

X3J3 / 95-004r1, X3J3 Journal of Requirements (JOR), Item 066

X3J3 / 96-006, Interpretation 000132
[End of 97-160]