

```

0001: |*****
0002: |***   実経過時間を計測するためのタイマ ルーチン tic - toc (チック・タック)   ***
0003: |*****
0004:
0005: |  使用法 :
0006: |  ○ 単一タイマで間に合う場合           |  ○ 複数タイマを使う場合
0007: |  USE timer_c                             |  USE timer_c
0008: |  CALL tic( )                             |  TYPE(timer_c) :: Stopwatch1, Stopwatch2
0009: |  |                                         |  REAL(DP) :: t(2)
0010: |  PRINT *, "経過時間 = ", toc( )         |  CALL tic(StopWatch2)
0011: |  |                                         |  |
0012: |  PRINT *, "経過時間 = ", toc( )         |  CALL tic(StopWatch1)
0013: |                                         |  |
0014: |                                         |  t(1) = toc(StopWatch1)
0015: |                                         |  |
0016: |                                         |  t(2) = toc(StopWatch2)
0017: |                                         |  PRINT *, "経過時間 = ", t(:)
0018:
0019: |  実経過時間を計測するために使う関数 :
0020: |  逐次実行ならば F90組込みルーチン CPU_TIME ( )
0021: |  Open-MP並列実行ならば、OpenMP関数 GET_OMP_WTIME ( ) (μ秒精度)
0022: |  (分ける理由: Open-MP並列化のとき、CPU_TIME ( ) は全CPUの加算結果を返すので、
0023: |  実経過時間の計測ができない。 )
0024:
0025: |  MODULE timer_class
0026: |
0027: |  USE constants
0028: |  !$ USE OMP_LIB
0029: |  IMPLICIT none
0030: |  PRIVATE
0031:
0032: |  TYPE, PUBLIC :: timer_c
0033: |  PRIVATE
0034: |  REAL(DP) :: time_st = NaN      ! 開始時刻
0035: |  END TYPE
0036:
0037: |  PUBLIC tic
0038: |  PUBLIC toc
0039:
0040: |  REAL(DP), SAVE :: time_st = NaN ! シングルトン (SAVE属性なくともSAVE)
0041:
0042: |  CONTAINS
0043:
0044: |  SUBROUTINE tic ( this_ )
0045: |  ! タイマの初期化
0046: |  TYPE(timer_c), INTENT(inout), OPTIONAL :: this_
0047: |  REAL(DP) :: t
0048: |  CALL CPU_TIME( t )
0049: |  !$ t = OMP_GET_WTIME( )
0050: |  IF ( PRESENT(this_) ) THEN
0051: |    this_%time_st = t
0052: |  ELSE
0053: |    time_st = t
0054: |  END IF
0055: |  END SUBROUTINE tic
0056:
0057:
0058: |  FUNCTION toc ( this_ ) RESULT ( wtime )
0059: |  ! タイマを初期化してからの実経過実時間(秒)を返す
0060: |  TYPE(timer_c), INTENT(inout), OPTIONAL :: this_
0061: |  REAL(DP) :: wtime
0062: |  REAL(DP) :: t
0063: |  CALL CPU_TIME( t )
0064: |  !$ t = OMP_GET_WTIME( )
0065: |  IF ( PRESENT(this_) ) THEN
0066: |    wtime = t - this_%time_st
0067: |  ELSE
0068: |    wtime = t - time_st
0069: |  END IF
0070: |  END FUNCTION toc
0071:
0072: |  END MODULE timer_class

```