NAME

TIMER − Directly measure the processor time used by program segments.

SYNOPSIS

COMMON /EXPT/ TIMING,NONALG,NBIN,TOPBIN,NTMEAS,TOH,BINCPU,SPEED
CALL TIMER(TCC,STA)

name	type	meaning
TIMING	LOGICAL*4	T => timing is enabled (input)
NONALG	INTEGER*4	bin where non-algorithm time is accumulated
NBIN	INTEGER*4	bin where time is being accumulated
TOPBIN	INTEGER*4	highest bin number used so far
NTMEAS	INTEGER*4	number of timing measurements made so far
TOH	INTEGER*4	per-call TIMER overhead correction
BINCPU(2,22)	INTEGER*4	timing bin counts in [sec,nsec] format
TCC	INTEGER*4	timer control code (input); see table below
STA	INTEGER*4	station number of the CALL statement (input)
SPEED	REAL*8	processor speed in MHz

TCC	meaning
−1	reset all bin counts to zero
0	update counts without changing bin
>0	switch timing to bin number TCC

WARNING

This routine uses processor cycle counts to estimate CPU time. It can be used only on a computer having an Intel Pentium processor, and for its measurements to be repeatable the program under test must be run on a system that is otherwise empty and idle.

DESCRIPTION

This routine implements a model of CPU timing in which CPU time is thought of as continuously flowing and the function of TIMER is merely to control what bin the time is flowing into. If timing is not enabled, the routine returns immediately. If timing is enabled the first call must be with TCC=−1, which initializes all bin counts to 0 and sets NBIN to NONALG so that on return CPU time will begin accumulating in that bin. On each subsequent call, the CPU time elapsed since the previous call is added to the bin that was specified on the previous call, and, if TCC > 0, NBIN is switched to the new value of TCC. Also, the time spent in TIMER is added to bin NONALG+1. If TCC = 0, the bin counts are updated but NBIN retains its most recent positive value.

This version of TIMER uses the C stub routine GETCYC to read the processor's cycle counter, so its resolution is one clock cycle. To convert cycle counts to CPU time it uses the processor speed, as obtained via the routine GETMHZ from the Linux file /proc/cpuinfo. The resulting estimates of the CPU time accumulated in each timing bin are returned as two-part values with units of seconds and nanoseconds.

SEE ALSO

GETCYC, GETMHZ

DIAGNOSTICS

If successive calls to TIMER specify the same value of TCC (i.e, the same bin), this is assumed to be due to a mistake, an error message is written, and the program stops. If an update call (TCC = 0) is made but the previous call was an initialization (TCC = −1), this is assumed to be a mistake, an error message is written, and the program stops. If STA > 0, the value of STA is included in error messages about illegal values of TCC as an aid to debugging. If GETMHZ cannot find the processor speed, an error message is written and the program stops.

LINKAGE

```
gfortran   source.f   −L${HOME}/lib   −lmisc
```

AUTHOR

Michael Kupferschmid

EXAMPLE

```
      COMMON /EXPT/ TIMING,NONALG,NBIN,TOPBIN,NTMEAS,TOH,BINCPU
      LOGICAL*4 TIMING
      INTEGER*4 TOPBIN,TOH,BINCPU(2,22)
      TIMING=.TRUE.
      CALL TIMER(-1,1)
      CALL TIMER( 1,2)
      X=2.
      DO 1 I=1,10000000
           X=SQRT(3.-X)
    1 CONTINUE
      CALL TIMER( 0,3)
      CALL TIMER(NONALG,4)
      WRITE(6,901) 1.D+09*DFLOAT(BINCPU(1,1))+DFLOAT(BINCPU(2,1))
  901 FORMAT('elapsed CPU time =',1PD13.6,' nanoseconds')
      STOP
      END
```

This example produced the following output:

```
unix[1] a.out
elapsed CPU time = 1.034395D+09 nanoseconds
unix[2]
```