

Zeitmessung in Fortran 90

(manchmal auch schon in Fortran 77)

- **Datum und Uhrzeit** (als Zeichenkette und Integer-Feld)

Die angegebenen Vereinbarungen sind jeweils als Mindestlänge zu verstehen.

```
character date*8, time*10, zone*5
integer values(8)
...
call date_and_time(date,time,zone,values)
```

Die zurückgegebenen Werte der Parameter haben das folgende Format:

<code>date</code>	JJJJMMTT	<code>values(1)</code>	Jahr	<code>values(5)</code>	Stunde
<code>time</code>	hhmmss.sss	<code>values(2)</code>	Monat	<code>values(6)</code>	Minute
<code>zone</code>	±hhmm	<code>values(3)</code>	Tag	<code>values(7)</code>	Sekunde
		<code>values(4)</code>	Differenz zur Weltzeit	<code>values(8)</code>	Millisekunde

- **Systemzeit** (Abfrage des rechnerinternen Zeitgebers)

```
integer count, count_rate, count_max
...
call system_clock(count, count_rate, count_max)
```

Bedeutung der Rückgabewerte:

<code>count</code>	Rechnerabhängiger Zählerwert
<code>count_rate</code>	Wie oft wird <code>count</code> pro Sekunde erhöht
<code>count_max</code>	nach Erreichen dieses Wertes beginnt <code>count</code> wieder bei 0

Beispiel:

```
integer start, ende, rate, cmax
real time
call system_clock(start,rate,cmax)
... ! Anweisungen, deren Dauer zu messen ist
call system_clock(ende)
time=float(ende-start)/float(rate) ! evtl. cmax beachten!
write(*,*) "Zeit in Sekunden: ",time
```

- **CPU-Zeit** (die „reine“ Rechenzeit)

```
real time
...
call cpu_time(time)
```

Rückgabewert des Parameters `time` ist die bisher (seit Start des Programms) verbrauchte CPU-Zeit mit der Maßeinheit „Sekunde“.

Ein negatives Ergebnis deutet an, dass der Rechner keine CPU-Zeiten messen kann.

Bemerkung: Mit der Messung der CPU-Zeit erhält man die reine Hardwareleistung, aber für die Nutzung eines Programms ist es oft interessanter, wie lange man auf das Ergebnis warten muss (also die „Echtzeit“-Messung).