RCSL No: 52-AA1094

Edition: February 1982

Author: Jan Bardino

Title:

RC3502 Performance Measurement

RCSL 42-i1905

A/S REGNECENTRALEN
af 1979

Keywords:

Real Time Pascal, RC3502, Performance Measurement.

Abstract:

With revision 5 of the RC3502 system it is possible to make some performance measurements on running REAL TIME PASCAL programs. The result of measuring includes the number of routine activations and the amount of time spent in the routines.

(8 printed pages)

Printed by A/S Regnecentralen af 1979, Copenhagen

RCSL 42-i1905

With revision 5 of the RC3502 system it is possible to make some performance measurements on running Real-Time Pascal programs. The result of measuring includes the number of routine activations and the amount of time spent in the routines. The time indications are real-time, i.e. system overhead, time slicing and time for wait are included.

Measurement is initiated by a call of the procedure "start_measure" with a semaphore as parameter. All succeeding routine calls and returns trigger update of a table. The table is allocated and initialized by start_measure.

The measurement is terminated by a call of stop_measure with the same semaphore parameter as start_measure. Stop_measure signals the table to a system semaphore and a new call of start_measure is possible. Measurements may be started and stopped as long as it is possible to allocate an area of 2574 bytes to be used as measurement area.

The measurement results may be printed on the console by means of the process print_statis(tical_information). The order of results is FIFO. The output process deallocates the measurement areas after printing of the results.

There is an upper limit of 255 different routines per module, i.e. per process including internal processes. If more than 255 different internal and external routines are called the (statically) last routines will use one common table entry, and the output process will put '**********' instead of a name.

The output format may be seen in Appendix A. The number of calls is modulo 64K and the time amounts are given as:

hours.minutes.seconds.hundredths of a second

Since the names are found in the code of the routines the output process must be run before the routines and processes are unloaded.

Use of the measurement tools:

The processes to be measured on must be compiled together with measureenv and with "measure.yes" in the call of the compiler. Measureenv includes external declarations of start_measure and stop_measure.

The necessary routines for measuring is supplied in measurelib, which is required in order to load/link processes compiled with measure.yes.

The output process is supplied as "bprintstat".

A. OUTPUT FORMAT A.

```
RUN TRAPTEST
LIST TRAPTEST
incarnation depth branch level state    size      stack
traptest        1      1      0  wait     500    00c0.bab0
RUN PRINT_STATIS
>print_statis
Performance measurement for traptest

 name          called        time
-------------------------------

traptest          01      0.01.30.40
start_measur      01      0.00.00.05
empty_1        32000      0.00.42.00
empty_2        32000      0.00.42.35
end
Performance measurement for traptest

 name          called        time
-------------------------------

traptest          01      0.01.33.20
start_measur      01      0.00.00.05
empty_1        33000      0.00.43.60
empty_2        33000      0.00.43.90
end
```

```
PROCESS traptest( VAR sy  system_vector );
VAR
  step1, step2 : integer;
  measure_sem : semaphore;

  PROCEDURE empty_1;
  BEGIN END;

  PROCEDURE empty_2;
  BEGIN END;

BEGIN
  start_measure( measure_sem );
  FOR step1 := 1 TO 32 DO
    FOR step2 := 1 TO 1000 DO
      BEGIN
        empty_1;
        empty_2;
      END;
  stop_measure( measure_sem );
  start_measure(measure_sem);
  FOR step1 := 1 TO 33 DO
    FOR step2 := 1 TO 1000 DO
      BEGIN
        empty_2;
        empty_1;
      END;
  stop_measure(measure_sem);
END.
```

```
name          headline beginline endline  appetite(words) default create_size
empty_1             12        12       12 ;          19
empty_2             15        15       15 ;          19
traptest             7        19       35 ;          75
exception       external, called  1 time(s)                    95
_exit_rc        external, called  1 time(s)
trap_code_rc    external, called  1 time(s)
start_measur    external, called  2 time(s)
stop_measure    external, called  2 time(s)

code: 0 . 440 = 440 bytes
```

RETURN LETTER

Title: RC3502 Performance Measurement RCSL No.: 52-AA1094

A/S Regnecentralen af 1979/RC Computer A/S maintains a continual effort to improve the quality and usefulness of its publications. To do this effectively we need user feedback, your critical evaluation of this manual.

Please comment on this manual's completeness, accuracy, organization, usability, and readability:

Do you find errors in this manual? If so, specify by page.

How can this manual be improved?

Other comments?

Name: _____ Title: _____

Company: _____

Address: _____

Date: _____

Thank you

42-i 1288

. . . . . Fold here . . . . .

. . . . . Do not tear - Fold here and staple . . . . .

Affix
postage
here

REGNECENTRALEN
af 1979

Information Department
Lautrupbjerg 1
DK-2750 Ballerup
Denmark