sprof

May 28, 2013

Type Package

Title Profiling, timing and optimization utilitites

Version 0.0-1

Date 2013-05-26

Author G. Sawitzki

Maintainer G. Sawitzki <gsawitzki@r-forge.r-project.org>

Description Support utilitites

License GPL-3

R topics documented:

`sprof-package` *sprof: R profiling*

Description

Profiling, timing and optimization utilitites

Details

Package:	sprof
Type:	Package
Version:	0.0-1
Date:	2013-05-26
License:	GPL-2

readProf reads a profile file and returns a composite structure.

Author(s)

G<c3><bc>nther Sawitzki <gsawitzki@r-forge.r-project.org>

See Also

proftools.

profr.

profr.

---

readProf *Read Rprof Output files and Stack Logs*

Description

Read a log of stack entries, such as the output of the Rprof function, and generate a more accessible representation.

Usage

```
readProf(filename = "Rprof.out", chunksize = 5000,
interval = 0.02, head=c("auto"))
```

Arguments

filename	Name of a file produced by Rprof().
chunksize	Number of lines to read at a time.
interval	real: time interval between samples, in s.
head	c("auto", "none", "Rprofmem") to interpret control information as provided by Rprof or Rprofmem. See details.

Details

This function reads a log file of stacks, one stack snapshot per line, stack entries separated by space.

As profiling output file could be very large, it is read in blocks of chunksize lines. Increasing chunksize will make the function run faster if sufficient memory is available.

The input format is controlled by the head argument. Format "auto" tries to detect control lines as interspersed by Rprof. These lines are not included in the output.

"none" igores all control information and includes these lines as strange stacks.

"Rprofmem" isolates headers as provided by Rprofmem. new page entries are encoded as malloc requests with length 0.

Value

A list with components

`firstline`	A verbatim copy of the first line of the input file. Typically this contains timing or formatting information.
`nodes`	A vector of node names. This may include stray entries from interperspersed lines.
`stacks`	A vector of unique stacks found in input, stored as verbatim copies.
`stacksrenc`	A list of unique stacks in top down order (top first), encoded as vectors or references to `stacks`.
`data`	A vector encoding the data file as referenes to `stacks`.
`timesRLE`	Vector of sampling intervals, in miliseconds. Run-length encoded.
`freq`	A frequency table summarizing `data`.

Author(s)

Günther Sawitzki <gsawitzki@r-forge.r-project.org>, based on the code of `summaryRprof`

References

http://sintro.r-forge.r-project.org/

See Also

`summaryRprof`

`summaryRprof`

`readRprofileData` and `flatProfile` in `library(proftools)`.

`{parse_rprof}` in `library(profr)`.

The chapter on "Tidying and profiling R code" in "Writing R Extensions" (see the 'doc/manual' subdirectory of the R source tree).

`Rprof` is a sampling profiler.

`tracemem` traces copying of an object via the C function `duplicate`.

`Rprofmem` is a non-sampling memory-use profiler.

http://developer.r-project.org/memory-profiling.html

Examples

```
## Not run:
## Rprof() is not available on all platforms
profinterval <- 0.001
simruns <- 100

n <- 10000
x <- runif(n)
y0 <- 2+ 3 * x

sinknull <- textConnection(NULL, "w"); sink(sinknull)
Rprof(tmp <- tempfile(), interval = profinterval)
for (i in 1:simruns) {y <- y0 +  rnorm(n); xxx<- summary(lm(y~x))}
Rprof()
```

```
Rprof_out <- readProf(tmp)

unlink(tmp)
sink(); close(sinknull)

str(Rprof_out)

## End(Not run)
```

---

summary_prof *Summary for profiles*

Description

Print a summary for the output of readProf.

Usage

```
summary_prof(x)
```

Arguments

x	A data structure as returned by readProf.

Value

None.

References

http://sintro.r-forge.r-project.org/

See Also

summaryRprof

Examples

```
## Not run:
## Rprof() is not available on all platforms
profinterval <- 0.001
simruns <- 100

n <- 10000
x <- runif(n)
y0 <- 2+ 3 * x

sinknull <- textConnection(NULL, "w"); sink(sinknull)
Rprof(tmp <- tempfile(), interval = profinterval)
for (i in 1:simruns) {y <- y0 +  rnorm(n); xxx<- summary(lm(y~x))}
Rprof()

Rprof_out <- readProf(tmp)
```

```
unlink(tmp)
sink(); close(sinknull)

summary_prof(Rprof_out)

## End(Not run)
```

Index