

# Package: rcpptimer (via r-universe)

October 23, 2024

**Type** Package

**Title** 'Rcpp' Tic-Toc Timer with 'OpenMP' Support

**Version** 1.2.1

**Date** 2024-09-21

**Description** Provides 'Rcpp' bindings for 'cpptimer', a simple tic-toc timer class for benchmarking 'C++' code [\(<https://github.com/BerriJ/cpptimer>](https://github.com/BerriJ/cpptimer)). It's not just simple, it's blazing fast! This sleek tic-toc timer class supports overlapping timers as well as 'OpenMP' parallelism [\(<https://www.openmp.org/>](https://www.openmp.org/)). It boasts a nanosecond-level time resolution. We did not find any overhead of the timer itself at this resolution. Results (with summary statistics) are automatically passed back to 'R' as a data frame.

**URL** <https://rcpptimer.berrisch.biz>,  
<https://github.com/BerriJ/rcpptimer>

**BugReports** <https://github.com/BerriJ/rcpptimer/issues>

**License** GPL (>= 3)

**Encoding** UTF-8

**Imports** Rcpp

**LinkingTo** Rcpp

**RoxygenNote** 7.3.2

**Suggests** testthat (>= 3.0.0), knitr, rmarkdown

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**Language** en-US

**NeedsCompilation** yes

**Author** Jonathan Berrisch [aut, cre]  
(<https://orcid.org/0000-0002-4944-9074>)

**Maintainer** Jonathan Berrisch <Jonathan@Berrisch.biz>

**Repository** CRAN

**Date/Publication** 2024-09-22 21:40:02 UTC

## Contents

fibonacci . . . . .	2
fibonacci_omp . . . . .	3
print.rcpptimer . . . . .	3
<b>Index</b>	<b>5</b>

---

fibonacci	<i>Simple rcpptimer example</i>
-----------	---------------------------------

---

### Description

Time the computation of Fibonacci numbers

### Usage

```
fibonacci(n)
```

### Arguments

`n` vector giving integers for which to compute the Fibonacci sum

### Details

The function being timed is the following:

```
int fib(int n) { return ((n <= 1) ? n : fib(n - 1) + fib(n - 2)); }
```

Runtime for computations less than  $n = 15$  is nearly unmeasurable.

### Value

vector of integers giving the Fibonacci sum for each element in `n`

### Examples

```
fibonacci(n = rep(20:25, 10))
# this function creates a global environment variable "times"
times
```

---

 fibonacci\_omp

*Simple rcpptimer example using OpenMP*


---

**Description**

Time the multithreaded computation of Fibonacci numbers

**Usage**

```
fibonacci_omp(n)
```

**Arguments**

`n` vector giving integers for which to compute the Fibonacci sum

**Details**

The function being timed is the following:

```
int fib(int n) { return ((n <= 1) ? n : fib(n - 1) + fib(n - 2)); }
```

Runtime for computations less than  $n = 15$  is nearly unmeasurable.

**Value**

vector of integers giving the Fibonacci sum for each element in `n`

**Examples**

```
fibonacci_omp(n = rep(20:25, 10))
# this function creates a global environment variable "times"
times
```

---

 print.rcpptimer

*Print method for rcpptimer output*


---

**Description**

Prints the times object and scales the timings if appropriate. If all timings are smaller than 1 microsecond, the timings are printed in nanoseconds. If the smallest timing is higher than a Millisecond / Seconds / Minutes / Hours, the timings are printed in the unit of that threshold. This behavior can be disabled by setting `scale = FALSE`.

**Usage**

```
## S3 method for class 'rcpptimer'
print(x, scale = TRUE, ...)
```

**Arguments**

x	Object of class rcpptimer
scale	Scale the timings and statistics to a more human readable format
...	further arguments are ignored

# Index

`fibonacci`, 2

`fibonacci_omp`, 3

`print.rcptimer`, 3