

# Package: rcpptimer (via r-universe)

September 16, 2024

**Type** Package

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

**Version** 1.1.0

**Date** 2024-03-20

**Description** Provides 'Rcpp' bindings for 'cpptimer', a simple tic-toc timer class for benchmarking 'C++' code <<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/>>. It boasts a microsecond-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>

**License** GPL (>= 3)

**Encoding** UTF-8

**Imports** Rcpp

**LinkingTo** Rcpp

**RoxygenNote** 7.3.1

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

**Config/testthat/edition** 3

**Roxygen** list(markdown = TRUE)

**VignetteBuilder** knitr

**Language** en-US

**Repository** <https://berrij.r-universe.dev>

**RemoteUrl** <https://github.com/berrij/rcpptimer>

**RemoteRef** HEAD

**RemoteSha** 9eb3bd73d386f3b526498c5c3c0639940ce9606b

## Contents

fibonacci . . . . .	2
fibonacci_omp . . . . .	3
<b>Index</b>	<b>4</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(10*(1:4), 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(10*(1:4), 10))  
# this function creates a global environment variable "times"  
times
```

# Index

fibonacci, [2](#)  
fibonacci\_omp, [3](#)