The \textit{rdyncall} package: An improved foreign function interface for R.

Daniel Adler$^{1,\ast}$, Tassilo Philipp$^2$

1. Institut for Statistics and Econometrics, Georg-August University of Göttingen
2. Potion Studios

$\ast$ Contact author: dadler@uni-goettingen.de

Keywords: foreign function call, callbacks, system programming, calling convention

R provides a foreign function interface (.Call(), .C() and .External()) to invoke function calls to precompiled library code. The interface supports a very limited subset of possible argument and return types for a foreign function. For instance, there is no direct support for passing scalar arguments types from R to C functions. Hence, it is often necessary to write C wrapper functions to make a binding work, which can be a cumbersome process.

We present the R package \textit{rdyncall} which provides an enhanced foreign function call interface to precompiled code, support for wrapping R functions into C callback objects, and R helper functions to work with C data structures. It can handle most C argument and return types, and performs automatic type conversions between R and C during calls and callbacks.

The package is implemented using the \textit{dyncall} library that encapsulates architecture-, OS- and compiler-specific function-call semantics. For each class of function-call semantics — the so called calling convention — the library uses a small call kernel written in assembly. It has been ported to several architectures (currently x86, x86\_64, ppc32, arm and mips) and multiple calling conventions (e.g. on x86: ‘cdecl’, ‘stdcall’, and gnu/microsoft ‘fastcall’ and ‘this call’).

A key concept in this package is the use of signature strings; type information is encoded as a compact text string and specifies the full semantics for calls and callbacks. This data format is easy to use, open for extensions and is also very efficient for low-level processing. As a neat side effect, signature strings can be regarded as portable representations for binding information across programming languages.

We show how \textit{rdyncall} can be used to bind R with precompiled code without the need for additional C wrapper code. Examples include bindings to libSDL (a portable multimedia library), to OpenGL (3D graphics rendering) and to the R shared library itself (e.g. access to low-level R memory mangement and in-place sorting of atomic R vectors).

References

