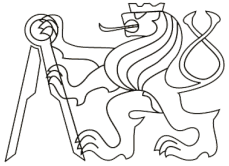
The Invisible Work on R

Tomas Kalibera
Czech Technical University
R Core



CZECH TECHNICAL
UNIVERSITY
IN PRAGUE

What is the core of R?

https://cloud.r-project.org/

Official releases of R source code are most likely what you want if you are looking for the sources of R (both Unix and Windows).

The latest release (2020-06-06, See Things Now): R-4.0.1.tar.gz
Changes to the previous version are documented in the file NEWS (also contained in the sources). Older releases are available here.

Source code

- R interpreter, byte-code compiler
- base, stats, graphics, parallel, tools, utils

Documentation

- ?, R Installation and Administration, Writing R Extensions

Who works on the core of R?

https://www.r-project.org/contributors.html

The current R is the result of a collaborative effort with contributions from all over the world. R was initially written by Robert Gentleman and Ross Ihaka - also known as “R & R” of the Statistics Department of the University of Auckland. Since mid-1997 there has been a core group with write access to the R source, currently consisting of ...

https://svn.r-project.org/R/trunk/doc/THANKS

External contributors

R Core

- 20 current members

Recent activities of R Core

Brian Ripley – CRAN, C-Fortran calls, PCRE2, encodings, documentation

Deepayan Sarkar – command completion, graphics

Kurt Hornik – CRAN, stringsAsFactors, S3 dispatch, bug fixing

Luke Tierney – reference counting, ALTREP, condition handlers, raw strings, bug fixing

Martin Maechler – matrix/array, bug fixing, numerical code, R Foundation

Martin Morgan – Bioconductor

Martyn Plummer – R Foundation

Michael Lawrence – S4, Bioconductor

Paul Murrell – graphics

Peter Dalgaard – release management, numerical code, bug fixing

Simon Urbanek – macOS toolchain and binaries, bugzilla, R Foundation

Tomas Kalibera – PCRE2, parallel/sockets, Windows port, encodings, bug fixing

Uwe Ligges – CRAN, binary packages for Windows, package management

Where to meet R Core?

Conferences on R and statistics

R-Devel mailing list — https://www.r-project.org/mail.html

- Discussing development of core R, technical questions

R Bugzilla — https://www.r-project.org/bugs.html

- Reporting and discussing bugs, wishlist items

R Blog — https://developer.r-project.org/Blog/public

What’s new in R 4.0?

Significant **user-visible** changes

- Major release is time for breaking changes

Reference counting

Migration to PCRE2

New features

Bug Fixes

https://cran.r-project.org/doc/manuals/r-release/NEWS.html

Significant user-visible changes

Strings (not) as factors — R Blog: stringsAsFactors

- Strings no longer converted to factors when creating data frames

Matrices as arrays — R Blog: When you think `class(.) == *`, think again!

- Methods for “array” now dispatch also on matrix objects

Generic plot() moved to base

Raw strings

- New syntax for character literals

```
> r"(c:\Program files\R)"
[1] "c:\\Program files\\R"

> r"(use both "double" and 'single' quotes)"
[1] "use both \"double\" and 'single' quotes"
```

Reference counting

Major change of internals

- references of R objects are counted exactly
- the number can go down, reducing need for copies
- enables future performance improvements

```
> x <- 1:1e6                                                    R 3.6
> x[1] <- 10
> first <- function(x) x[1]
> .Internal(inspect(x))
@7f8f24eb0010 14 REALSXP g0c7 [NAM(1)] (len=1000000, tl=0) 10,2,3,4,5,...
> first(x)
[1] 10
> .Internal(inspect(x))
@7f8f24eb0010 14 REALSXP g0c7 [NAM(7)] (len=1000000, tl=0) 10,2,3,4,5,...
> x[1] <- 100
> .Internal(inspect(x))
@7f8f2470e010 14 REALSXP g0c7 [NAM(1)] (len=1000000, tl=0) 100,2,3,4,5,...
```

Vector “x” is copied

Reference counting

Major change of internals

- references of R objects are counted exactly
- the number can go down, reducing need for copies
- enables future performance improvements

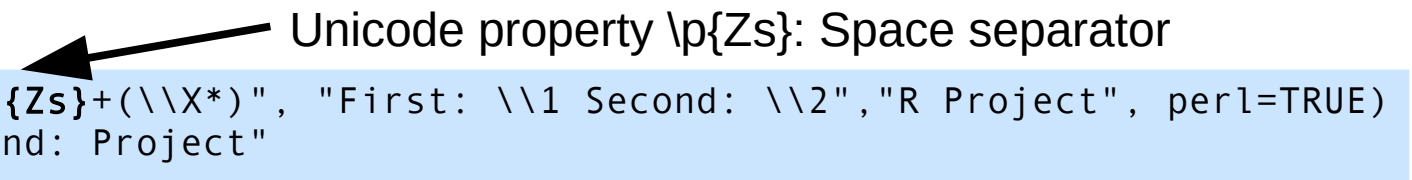
R 4.0

```
> x <- 1:1e6
> x[1] <- 10
> first <- function(x) x[1]
> .Internal(inspect(x))
@7f78e3054010 14 REALSXP g0c7 [REF(1)] (len=1000000, tl=0) 10,2,3,4,5,...
> first(x)
[1] 10
> .Internal(inspect(x))
@7f78e3054010 14 REALSXP g0c7 [REF(1)] (len=1000000, tl=0) 10,2,3,4,5,...
> x[1] <- 100
> .Internal(inspect(x))
@7f78e3054010 14 REALSXP g0c7 [REF(1)] (len=1000000, tl=0) 100,2,3,4,5,...
```

Migration to PCRE2

Maintenance change

- needed to support recent Unicode tables
- required rewrite of R/PCRE layer, now supports both PCRE1 and 2
- Invisible to users, except where PCRE2 is stricter




```
> gsub("^(\\X*)\\p{Zs}+(\\X*)", "First: \\1 Second: \\2","R Project", perl=TRUE)
[1] "First: R Second: Project"
```

```
[^\\w-/\\\\:.]
[\s-.]+
```

No longer accepted, hyphen must be escaped with “\”

```
[^\R]
```

No longer accepted, likely used in error (matched “R”)

Speedup in cluster initialization

R Blog: Socket Connections Update

Performance improvement

- PSOCK cluster is started in parallel
- Improved robustness of R sockets layer
- New API for server socket connections

```
library(parallel); system.time(cl <- makePSOCKcluster(n))
```

	R 3.6	R 4.0
Fedora (64)	14s	0.4s
Ubuntu (40)	6.6s	0.4s
Windows (48)	9.3s	0.5s
Solaris (64)	211s	7s
MacOS (12)	4.2s	0.7s

Speedup in cluster initialization

R Blog: Socket Connections Update

Performance improvement

- PSOCK cluster is started in parallel
- Improved robustness of R sockets layer
- New API for server socket connections

```
serverSocket(port)

socketAccept(socket,
             blocking = FALSE,
             open = "a+",
             encoding = getOption("encoding"),
             timeout = getOption("timeout"))
```

Calling from C to LAPACK/GFortran

20 CRAN packages failing
with unreleased GFortran 8...

R Blog: GFortran Issues with LAPACK
GFortran Issues with LAPACK II

```
void inverse( double A[], double A_inv[], int *p )
{
        int info, dim = *p;
        char uplo = 'U';

        // creating an identity matrix
        #pragma omp parallel for
        for( int i = 0; i < dim; i++ )
                for( int j = 0; j < dim; j++ )
                                                = ( i == j );

BLAS/LAPACK routine 'DPOTRS' gave error code -1

        // LAPACK function: computes solution to A * X = B, where ...
        F77_NAME(dposv)( &uplo, &dim, &dim, A, &dim, A_inv, &dim, &info );
}
```

Calling from C to LAPACK/GFortran

DPOTRS()

```
      upper = lsame( uplo, 'U' )
      IF( .NOT.upper .AND. .NOT.lsame( uplo, 'L' ) ) THEN
         info = -1
      ELSE IF( n.LT.0 ) THEN
```

inverse()

```
void inverse( double A[], double A_inv[], int *p )
{
        int info, dim = *p;
        char uplo = 'U';

        // creating an identity matrix
        #pragma omp parallel for
        for( int i = 0; i < dim; i++ )
                for( int j = 0; j < dim; j++ )
                                        = ( i == j );
```

BLAS/LAPACK routine 'DPOTRS' gave error code -1

```
        // LAPACK function: computes solution to A * X = B, where ...
        F77_NAME(dposv)( &uplo, &dim, &dim, A, &dim, A_inv, &dim, &info );
}
```

inverse() calls DPOSV()
which calls DPOTRS()

Calling from C to LAPACK/GFortran

```
      upper = lsame( uplo, 'U' )
      IF( .NOT.upper .AND. .NOT.lsame( uplo, 'L' ) ) THEN
         info = -1
      ELSE IF( n.LT.0 ) THEN
```

DPOTRS()

```
      IF( info.EQ.0 ) THEN
         CALL dpotrs( uplo, n, nrhs, a, lda, b, ldb, info )
      END IF
      RETURN
```

DPOSV()

```
void inverse( double A[], double A_inv[], int *p )
{
        char uplo = 'U';
                                                  to A * X = B, where ...
        F77_NAME(dposv)( &uplo, &dim, &dim, A, &dim, A_inv, &dim, &info );
}
```

inverse()

BLAS/LAPACK routine 'DPOTRS' gave error code -1

inverse() calls DPOSV()
which calls DPOTRS()

Calling from C to LAPACK/GFortran

DPOSV()

```
      CALL DPOTRS( UPLO, N, NRHS, A, LDA, B, LDB, INFO )
1174d4:       48 8b 04 24             mov    (%rsp),%rax <======= rax holds LDB
1174d8:       4c 89 7c 24 68          mov    %r15,0x68(%rsp) <=== save INFO to output param
1174dd:       49 89 d8                mov    %rbx,%r8 <========== pass LDA as LDA
1174e0:       4c 89 e1                mov    %r12,%rcx <========= pass A as A
1174e3:       4c 8b 4c 24 08          mov    0x8(%rsp),%r9 <===== pass B as B
1174e8:       4c 89 ea                mov    %r13,%rdx <========= pass NRHS as NRHS
1174eb:       48 89 ee                mov    %rbp,%rsi <========= pass N as N
1174ee:       4c 89 f7                mov    %r14,%rdi <========= pass UPLO as UPLO
1174f1:       48 c7 44 24 70 01 0(    movq   $0x1,0x70(%rsp) <=== pass 1 hidden arg on stack
1174f8:       00 00
1174fa:       48 89 44 24 60          mov    %rax,0x60(%rsp) <=== pass LDB as LDB (stack)
    END
1174ff:       48 83 c4 28             add    $0x28,%rsp <== remove 5 vars from stack
117503:       5b                      pop    %rbx
117504:       5d                      pop    %rbp
117505:       41 5c                   pop    %r12
117507:       41 5d                   pop    %r13
117509:       41 5e                   pop    %r14
11750b:       41 5f                   pop    %r15
      CALL DPOTRS( UPLO, N, NRHS, A, LDA, B, LDB, INFO )
11750d:       e9 de 56 ef ff          jmpq   cbf0 <dpotrs_@plt> <=== tail call to dpotrs
```

inverse() calls DPOSV()
which calls DPOTRS()

Calling from C to LAPACK/GFortran

DPOSV()

```
        CALL DPOTRS( UPLO, N, NRHS, A, LDA, B, LDB, INFO )
  1174d4:       48 8b 04 24             mov    (%rsp),%rax <======= rax holds LDB
  1174d8:       4c 89 7c 24 68          mov    %r15,0x68(%rsp) <=== save INFO to output param
  1174dd:       49 89 d8                mov    %rbx,%r8 <========== pass LDA as LDA
  1174e0:       4c 89 e1                mov    %r12,%rcx <========= pass A as A
  1174e3:       4c 8b 4c 24 08          mov    0x8(%rsp),%r9 <===== pass B as B
  1174e8:       4c 89 ea                mov    %r13,%rdx <========= pass NRHS as NRHS
  1174eb:       48 89 ee                mov    %rbp,%rsi <========= pass N as N
  1174ee:       4c 89 f7                mov    %r14,%rdi <========= pass UPLO as UPLO
  1174f1:       48 c7 44 24 70 01 00    movq   $0x1,0x70(%rsp) <=== pass 1 hidden arg on stack
  1174f8:       00 00
  1174fa:       48 89 44 24 60          mov    %rax,0x60(%rsp) <=== pass LDB as LDB (stack)
      END
```

```
subroutine dpotrs   (
          character UPLO,
          integer   N,
          integer   NRHS,
          double precision, dimension( lda, * )   A,
          integer   LDA,
          double precision, dimension( ldb, * )   B,
          integer   LDB,
          integer   INFO
     )
```

1 is hidden argument, length of UPLO, passed in the stack slot where that length should have been passed to DPOSV, but was not...

inverse() calls DPOSV()
which calls DPOTRS()

Calling from C to LAPACK/GFortran

Writing R Extensions:
Fortran character strings 6.6.1

R Blog: GFortran Issues with LAPACK
GFortran Issues with LAPACK II

R, R packages, LAPACKE,... call LAPACK incorrectly

- Not allowed by current Fortran standard
- Yet it always worked before **and is widely used**

R has been fixed to call LAPACK correctly

- Macros to do that also available for packages

R uses compile options to prevent tail-optimizations

As a result of this, GFortran has been fixed not to break this code.
A Fortran that could not build LAPACK would be of little use.

R Can use your help with bugs

R Blog: R Can Use Your Help: Reviewing Bug Reports
R Blog: Thanks for Reviewing Bug Reports

Basic skills (and hard work) enough to help

- Find minimal reproducible examples
- Identify invalid reports, bugs already fixed

Technical skills (and hard work) allow for special help

- Debug/analyze confirmed bugs

Special thanks to those who helped most recently:

- Elin Waring, Michael Chirico, Benjamin Tyner, Sebastian Meyer