Sustainable Package Development

Tomas Kalibera
R Core, Czech Technical University
Parse data for equal assign

```
parsed <- parse("src.r", keep.source=TRUE)
utils::getParseData(parsed, includeText=TRUE)
```

Bug reported by Barbara Lerner
y <- 5

x = 7
Trivial fix in R: change a single line

equal_assign:  expr EQ_ASSIGN expr_or_assign  { $$ = xxbinary($2,$1,$3); }
Testing change with CRAN/BIOC

“Woken-up” segfault: quickReg

Non-deterministic installation problems:
- ~2 packages, race condition in parallel install

Sophisticated workarounds that broke:
- Styler: 129 LOC (46 true code)
- Linter: 104 LOC (71 true code)

Fixed in R 3.6.0 (R-devel 75386)
Helping to get the packages fixed

Had to debug to identify it is a problem to be fixed in packages (not R)
Provided patch for one package
One package had a new maintainer not yet familiar with the code
One maintainer chose to wait for the commit as he had difficulty building R on Windows
15 emails; committed after 6 weeks, informed CRAN of breakage, packages fixed 6 weeks later

Best practice: File a good bug report first.
Rchk PROTECT bug report

```
for (r_ssize i = 0; i < r_length(elt); ++i) {
    sexp* value = r_list_get(elt, i);
    r_list_poke(out, count, value);

    sexp* name = r_nms_get(names, i);
    if (name != r_string("")) {
        name = KEEP(r_str_unserialise_unicode(name));

        if (out_names == r_null)
            out_names = KEEP_N(init_names(out),
                              n_protect);
        r_chr_poke(out_names, count, name);
    }
    ++count;
}
```

"Unprotected variable out_names while calling allocating function r_nms_get"
for (r_ssize i = 0; i < r_length(elt); ++i) {
    SEXP value = r_list_get(elt, i);
    r_list_poke(out, count, value);

    SEXP name = r_nms_get(names, i);
    if (name != Rf_mkChar(")") {
        name = KEEP(r_str_unserialise_unicode(name));

        if (out_names == Rf_mkChar("")
            out_names = KEEP_N(init_names(out),
                               n_protect);
        r_chr_poke(out_names, count, name);
        FREE(1);
    }
    ++count;
}
Rchk PROTECT bug report

```c
for (r_ssize i = 0; i < r_length(elt); ++i) {
    sexp* value = r_list_get(elt, i);
    r_list_poke(out, count, value);

    sexp* name = r_nms_get(names, i);
    if (name != r_string("")) {
        name = KEEP(r_str_unserialise_unicode(name));

        if (out_names == r_null)
            out_names = KEEP_N(init_names(out),
                                n_protect);
        r_chr_poke(out_names, count, name);
        FREE(1);
    }
    ++count;
}
```

"Unprotected variable out_names while calling allocating function r_nms_get"

out_names PROTECTed

out_names UNPROTECTed

Best practice: use R API directly.
Risks to maintainability: native code

LOW RISK

R
C/Fortran with .C/.Fortran
C with .Call
C++
C++ with .Call

HIGH RISK

C memory corruption
Windows
PROTECT errors
R heap corruption
Constants corruption
Missed destructors
Error handling
Indirect use of APIs

Best practice: only use native code when absolutely necessary.
Maintainability of packages impacts maintainability of R

R changes must be checked against packages
Problems discovered must be debugged
- Requires knowledge of R internals
- Few people willing and able to do this

Debugging memory corruption can take days