

From: Ivan Krylov

Organization: Lomonosov Moscow State University

Subject: Please correct to safely retain
your package on CRAN

Conference: useR!2026

Date: Wed, 08 Jul 2026 13:40:00 +0200



Dear maintainer,

In memory of
Tomáš Kalibera

Content[Weekly Edition](#)[Archives](#)[Search](#)[Kernel](#)[Security](#)[Events calendar](#)[Unread comments](#)[LWN FAQ](#)[Write for us](#)**Edition**[Return to the Front page](#)

Fortran, though, likes to know the sizes of the strings it is dealing with. When strings are passed as arguments to functions or subroutines, the [GCC Fortran argument-passing conventions](#) state that the length of each string is to be appended to the list of arguments. Consider a Fortran subroutine defined something like this:

```
subroutine foo(i, s)
  integer i
  character s
  ...
```

When that subroutine is called from other Fortran code, the length of `s` will be added by the compiler as a third, "hidden" argument. The C compiler will do no such thing, though, so the proper call to that function from C would look like:

```
int i;
char *s = "bar";

foo(&i, s, strlen(s));
```

From C, the length of `s` must be passed explicitly the end of the list of arguments.

At some distant point in the past, though, somebody decided that the hidden length argument should be omitted for single-character strings — those that are declared `"character *1"` in the called function, for example. It is not clear that any Fortran compiler anywhere ever implemented that behavior, but developers writing calls from C developed the habit of leaving out the length in that situation. As long as the called code knew that it was getting a single-character string, it would not need to check the (missing) hidden length parameter; everything worked, even though the calling standards were being violated. Various LAPACK subroutines expect single-character strings, and packages like [CBLAS](#) and [LAPACKe](#) duly leave out the length argument when calling them.

Once again, this is not how these functions are supposed to be called, but things worked anyway. At least, until they broke. [It seems that the problem was originally worked out by Thomas Kalibera in the R language community](#): a fix for [an unrelated ABI issue](#) caused crashes with some LAPACK calls. After, seemingly, a great deal of analysis work, Kalibera figured out where things go wrong. A subroutine taking a single-character string would call another just prior to returning, passing the same string. The compiler would optimize that call into a tail call (or more properly a "sibling call" using the same parameters); prior to making the jump, it would helpfully store the string length at the end of the argument list. But that length wasn't there to begin with,



kalibera / rchk Public

Notifications

Fork 11

Star 104

<> Code Issues 7 Pull requests 2 Actions Projects Wiki Security and quality Insights

master ▾

5 Branches 1 Tag

Go to file

<> Code ▾



kalibera Bump bioc version.

bd34d5d · 6 months ago

473 Commits



doc

Use --rm in docker run examples. Thanks to Kirill Müller.

2 years ago



examples

Docs.

11 years ago



image

Bump bioc version.

6 months ago



scripts

Bump bioc version.

6 months ago



src

Assume that ALTLIST methods are non-allocating

3 years ago



README.md

Use --rm in docker run examples. Thanks to Kirill Müller.

2 years ago



README



This project consists of several bug-finding tools that look for memory protection errors in C source code using R API, that is in the source code of [R](#) itself and packages. The tools perform whole-program static

About

No description, website, or topics provided.

Readme

Activity

104 stars

6 watching

11 forks

Report repository

Releases

1 tags

Packages

No packages published

Contributors 6



kalibera / mfuzz Public

Notifications

Fork 0

Star 2

<> Code Issues Pull requests Actions Projects Security and quality Insights

main ▾

1 Branch 0 Tags

Go to file

<> Code ▾

kalibera Initial version 417ee98 · 6 months ago 1 Commit

README.md	Initial version	6 months ago
genwrappers.r	Initial version	6 months ago
mfuzz.c	Initial version	6 months ago
mfuzz.h	Initial version	6 months ago
wrappers.c	Initial version	6 months ago

README

mfuzz

This is a tool for detecting over-sensitivity issues in programs on Unix (only tested on Linux) to accuracy of the

About

Fuzzing tool for C Math library on Unix

Readme

Activity

2 stars

0 watching

0 forks

Report repository

Releases

No releases published

Packages

No packages published

Contributors 1

clang-ASAN, gcc-UBSAN error (redatam)

```
> read_redatam(paste(dout, "cg15.dic", sep = "/"))
Opening dictionary file...
/usr/bin/../../lib/gcc/x86_64-linux-gnu/14/../../include/c++/
14/bits/basic_string.h:1272:9: runtime error: addition of unsigned
offset to 0x7f38f12bcc30 overflowed to 0x7f38f12bcc2f
    #0 0x7f38e859a15f in std::__cxx11::basic_string<char,
std::char_traits<char>, std::allocator<char>>::operator[](unsigned
long) .../include/c++/14/bits/basic_string.h:1272:9
    #1 0x7f38e859a15f in std::__cxx11::basic_string<char,
std::char_traits<char>, std::allocator<char>>::back()
.../include/c++/14/bits/basic_string.h:1353:9
    #2 0x7f38e859a15f in
RedatamLib::ListExporter::ListExporter(std::__cxx11::basic_string<c
har, std::char_traits<char>, std::allocator<char>> const&)
/root/redatam.Rcheck/00_pkg_src/redatam/src/redatamlib/exporters/RL
istExporter.cpp:16:21
```

the year of the



Linux desktop

[GALLERY](#)[SUPPORT](#)[CHANGES](#)[DOWNLOAD](#)

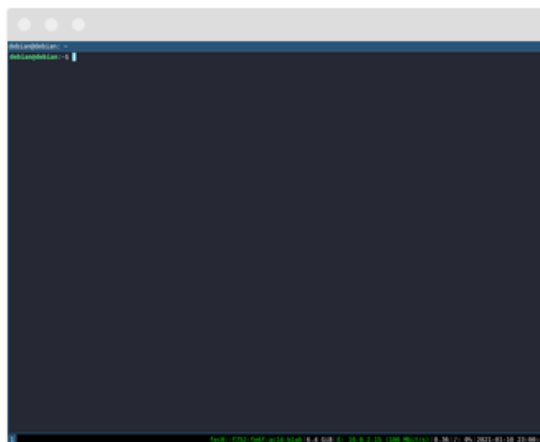
ArchLinux ARM

ARM64



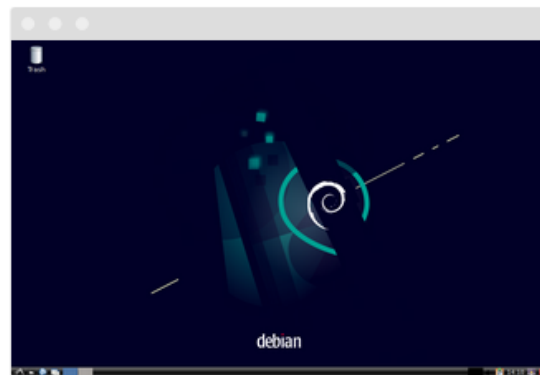
Debian 10.4 (Custom i3)

ARM64



Debian 11 (LXDE)

ARM64



Debian 11 (Xfce)

ARM64



Debian 12 (Rosetta)

ARM64



Debian 12

ARM64



Windows Developer Tools

[Advanced settings](#)
[CLI tools](#)
[Dev Configs](#)
[PowerToys](#)
[WinGet](#)
[WSL](#)
[Terminal](#)
[Setup guides](#)



Find by title

WSL Documentation

> Overview

> Install

Install WSL

Manual install steps for older versions

Install on Windows Server

> Tutorials

> Concepts

> How-to

> Enterprise security

Frequently Asked Questions

Troubleshooting

> Release Notes

Learn > Windows > WSL >


[Ask Learn](#)



How to install Linux on Windows with WSL



Summarize this article for me

Developers can access the power of both Windows and Linux at the same time on a Windows machine. The Windows Subsystem for Linux (WSL) lets developers install a Linux distribution (such as Ubuntu, OpenSUSE, Kali, Debian, Arch Linux, etc) and use Linux applications, utilities, and Bash command-line tools directly on Windows, unmodified, without the overhead of a traditional virtual machine or dualboot setup.

Prerequisites

You must be running Windows 10 version 2004 and higher (Build 19041 and higher) or Windows 11 to use the commands below. If you are on earlier versions please see [the manual install page](#).

In this article

Prerequisites

Install WSL command

Change the default Linux distribution installed

Set up your Linux user info

Set up and best practices

Check which version of WSL you are running

Upgrade version from WSL 1 to WSL 2

Ways to run multiple Linux distributions with WSL

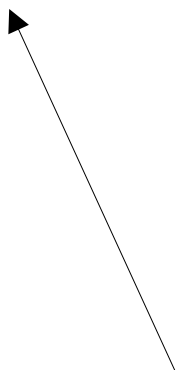
Want to try the latest WSL preview features?

[Show 3 more](#)

Was this page helpful?  

Error location

```
ListExporter::ListExporter(const std::string &outputDirectory)
    : m_path(outputDirectory) {
    if ('/' != m_path.back()) {
        m_path.append("/");
    }
}
```

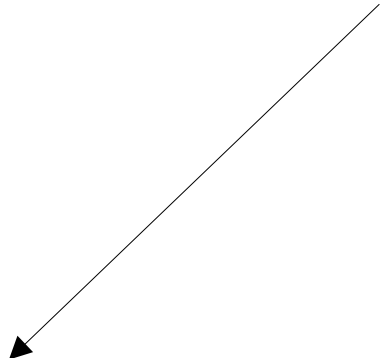


- Equivalent to
 `m_path[m_path.size() - 1]`
- UB if `empty()` (`size() == 0`)
- Just add some size checks?

With size checks added

```
try {  
    while (true) {  
        curr = TryGetEntity();  
        if (curr.first) {  
            ret.push_back(std::move(curr.second));  
        } else {  
            m_reader.MovePos(1);  
        }  
    }  
} catch (const out_of_range &) {  
}  
  
if (ret.empty()) {  
    stop("Error: No entities found.");  
}
```

Signals error here



TryGetEntity?

```
std::pair<bool, Entity> FuzzyEntityParser::TryGetEntity() {  
    size_t ogPos = m_reader.GetPos();  
  
    string entityName("");  
    if (!m_reader.TryReadStr(&entityName) || entityName.empty()) {
```

```
bool ByteArrayReader::TryReadStr(string *output, bool filterByContent) {  
    size_t ogPos = m_currPos;
```

```
    try {  
        uint16_t len = ReadInt16LE();
```

Evaluation order
not specified

```
uint16_t ByteArrayReader::ReadInt16LE() {  
    return    static_cast<uint16_t>(ReadByte())  
            | static_cast<uint16_t>(ReadByte()) << 8;  
}
```

ReadInt16LE, sanitized

0000000000267010 <RedatamLib::ByteArrayReader::ReadInt16LE(>:

267028: e8 93 6f f5 ff **call** 1bdfc0 <ByteArrayReader::ReadByte(>
eax <- ReadByte()

26702d: 41 89 c4 **mov** %eax,%r12d
r12 <- ReadByte()

26703d: 41 c1 e4 08 **shl** \$0x8,%r12d
r12 <- r12 << 8

267041: e8 7a 6f f5 ff **call** 1bdfc0 <ByteArrayReader::ReadByte(>
eax <- ReadByte()

26704a: 44 09 e0 **or** %r12d,%eax
eax <- eax / r12

ReadInt16LE, unsanitized

000000000046d30 <RedatamLib::ByteArrayReader::ReadInt16LE(>:

46d39: e8 f2 9c fd ff **call** 20a30 <ByteArrayReader::ReadByte(>
eax <- *ReadByte()*

46d41: 0f b6 d8 **movzbl** %al,%ebx
ebx <- *eax*

46d44: e8 e7 9c fd ff **call** 20a30 <ByteArrayReader::ReadByte(>
eax <- *ReadByte()*

46d4d: c1 e0 08 **shl** \$0x8,%eax
eax <- *eax* << 8

46d50: 09 d8 **or** %ebx,%eax
eax <- *eax* / *ebx*

Bugs in -fsanitize

(s2)

```
g++-15 -fsanitize=address,undefined,bounds-strict ... \  
-c s2-cell-union.cpp -o s2-cell-union.o
```

```
/usr/include/absl/container/internal/raw_hash_set.h:4052:56:   in  
'constexpr' expansion of  
'absl::...::hash_policy_traits<...>::get_hash_slot_fn<...>()'  
/usr/include/absl/container/internal/hash_policy_traits.h:163:54: error:  
'(absl::...::TypeErasedApplyToSlotFn<...> == 0)' is not a constant  
expression  
  163 |         return policy::template get_hash_slot_fn<Hash>() == nullptr  
      |         ~~~~~^~~~~~
```

📁 [r-spatial](#) / [s2](#) Public

🔔 Notifications Fork 18 ☆ Star 79

<> Code Issues 19 Pull requests Actions Projects Security and quality Insights

new CRAN issues w. absl #293

New issue

🔴 Open

 edzer opened on Feb 6

Member ...

We got this message some weeks ago:

► Details

It looks like the `absl` version (system, 20240722) is too new for the `s2geometry` version we bundle. A solution would be to use the vendored (bundled) `absl`, but by default we pick up a system `absl` if it is new enough (≥ 20230802); `absl` however does not aim for backward compatibility and `s2geometry` wants particular versions of it, not $\geq$ some version. I've tried to replace `s2geometry` with 0.12.0 bundled with `absl` 20240116 but get a missing symbol error when linking `s2.so` (some `absl` module missing); I had to uncommend (remove) this line

[s2/src/init.cpp](#)
Line 23 in [ad09e15](#)

```
23     absl::SetFlag(&FLAGS_s2debug, false);
```

to get that running; I also tried `s2geometry` 0.13.1 with the `absl` it requires (20250814) but that led to other problems, e.g. here

[s2/src/s2geography/constructor.h](#)

Assignees

No one assigned

Labels

No labels

Type

No type

Fields

No fields configured for issues without a type.

Projects

No projects

Milestone

No milestone

[Bug 71962](#) - error: '(& x) != 0u' is not a constant expression with fsanitize=undefined**Status:** NEW**Alias:** None**Product:** gcc**Component:** sanitizer ([show other bugs](#))**Version:** 6.1.1**Importance:** P3 normal**Target Milestone:** ---**Assignee:** Not yet assigned to anyone**URL:****Keywords:** rejects-valid**Duplicates (7):** [67762](#) [104426](#) [108617](#) [113826](#) [120825](#) [121731](#) [124468](#) ([view as bug list](#))**Depends on:****Blocks:** [constexpr](#) [110936](#)Show dependency [tree](#) / [graph](#)**Reported:** 2016-07-21 18:04 UTC by Jonathan Wakely**Modified:** 2026-06-25 10:45 UTC ([History](#))**CC List:** 19 users ([show](#))**See Also:** [110493](#)[110828](#)[110936](#)[104426](#)**Host:****Target:****Build:****Known to work:****Known to fail:** 5.3.1, 6.1.1, 7.1.0, 8.0**Last reconfirmed:** 2017-07-21 00:00:00**Attachments**[Add an attachment](#) (proposed patch, testcase, etc.)

Note

You need to [log in](#) before you can comment on or make changes to this bug.

Bugs in LTO

(isotree)

```
std::string s{" foo \" bar"};
const char * rx = "\\\"";
std::cout
    << std::regex_replace(s, std::regex(rx), "\\\"")
    << std::endl;
```

```
g++-14 -flto=10 -o foo -g -O2 -mtune=native \
    -fsanitize=address,undefined,bounds-strict \
    foo.cpp && ./foo
```

==648==ERROR: AddressSanitizer: **global-buffer-overflow** on address ...

READ of size 1 at 0x556ed780fa02 thread T0

```
#0 0x556ed773151f in ... /usr/include/c++/14/bits/regex_scanner.tcc:98
#1 0x556ed773151f in ... /usr/include/c++/14/bits/regex_scanner.tcc:79
#2 0x556ed7734416 in ... /usr/include/c++/14/bits/regex_compiler.tcc:575
#3 0x556ed7748374 in ... /usr/include/c++/14/bits/regex_compiler.tcc:310
#4 0x556ed7748374 in ... /usr/include/c++/14/bits/regex_compiler.tcc:133
#5 0x556ed7748374 in ... /usr/include/c++/14/bits/regex_compiler.tcc:115
#6 0x556ed7747428 in ... /usr/include/c++/14/bits/regex_compiler.tcc:118
#7 0x556ed7753285 in ... /usr/include/c++/14/bits/regex_compiler.tcc:91
#8 0x556ed77df36e in ... /usr/include/c++/14/bits/regex_compiler.tcc:76
#9 0x556ed77df36e in std::...::M_compile(...) [clone .constprop.0] /usr/...
#10 0x556ed771b8cf in std::...::basic_regex(...) /usr/include/c++/14/bits/...
#11 0x556ed771b8cf in main foo.cpp:7
```

0x556ed780fa02 is located 0 bytes after global variable '***.LC45**' defined in
'./foo.ltrans3.ltrans' (0x556ed780fa00) of size 2

'***.LC45**' is ascii string ''

```

template<typename _CharT>
_Scanner<_CharT>::
    _Scanner(const _CharT* __begin, const _CharT* __end,
        _FlagT __flags, std::locale __loc)
: _ScannerBase(__flags),
    _M_current(__begin), _M_end(__end),
    _M_ctype(std::use_facet<_CtypeT>(__loc)),
    _M_eat_escape(_M_is_ecma()
        ? &_Scanner::M_eat_escape_ecma
        : &_Scanner::M_eat_escape_posix)
{ _M_advance(); }

```

```

template<typename _CharT>
void
_Scanner<_CharT>::
_M_advance()
{
    if (_M_current == _M_end)
    {
        _M_token = _S_token_eof;
        return;
    }

```

```

template<typename _CharT>
void
_Scanner<_CharT>::
_M_scan_normal()
{
    auto __c = *_M_current++; // line 98, ASAN error

```

```

    if (_M_state == _S_state_normal)
        _M_scan_normal();

```

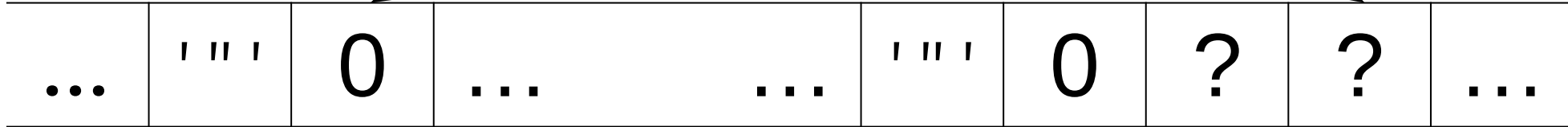


```
(gdb) p _M_current-3  
$12 = 0x55ce00bb4a00 "\
```

`_M_current`

```
(gdb) p _M_end-1  
$13 = 0x55ce00b956a0 "\
```

`_M_end`



C++: "\

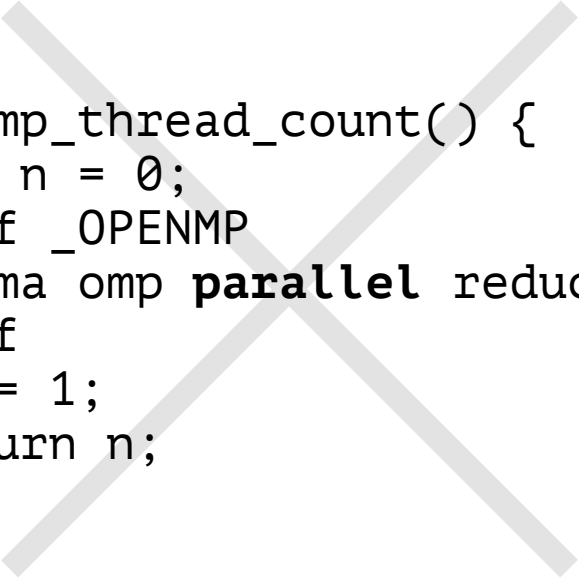
Installation took CPU time x times elapsed time (fawir)

- How to consume extra CPU time?
 - threads
 - `pthread_create()`, `clone(CLONE_THREAD)`, `#pragma omp parallel ...`
 - processes
 - `make -j$(nproc)`
 - `fork()`, `system()`, ...
- How to catch that?
 - (gdb) `b pthread_create`


```
$ ltrace -w 10 -f -e pthread_create -- \  
sh -c '~/R-build/bin/R CMD INSTALL fawir_1.0.tar.gz'
```

...

```
[pid 10154] libgomp.so.1->pthread_create(...) = 0  
      omp_fulfill_event (ip = 0x7f13bc0c7391)  
      GOMP_parallel (ip = 0x7f13bc0be0b1)  
      _Z16omp_thread_countv (ip = 0x7f133153cfa1)  
      _rsparse_omp_thread_count (ip = 0x7f133151c17b)  
rsparse:::onLoad(...) { R_doDotCall (ip = 0x7f13bc900712)  
                        bcEval_loop (ip = 0x7f13bc94502c)  
                        bcEval (ip = 0x7f13bc94c902)  
                        Rf_eval (ip = 0x7f13bc94cc7b)  
                        forcePromise.part.0 (ip = 0x7f13bc94d5db)  
                        Rf_eval (ip = 0x7f13bc94cf70)
```



```
int omp_thread_count() {  
    int n = 0;  
#ifdef _OPENMP  
#pragma omp parallel reduction(+ : n)  
#endif  
    n += 1;  
    return n;  
}
```

Wastes CPU time
creating threads



```
int omp_thread_count() {  
    int n = 1;  
#ifdef _OPENMP  
    int max_threads = omp_get_max_threads(),  
        thread_limit = omp_get_thread_limit();  
    n = max_threads < thread_limit  
        ? max_threads : thread_limit;  
#endif  
    return n;  
}
```

Re-building vignettes had CPU time x
times elapsed time
(greta)

- **ltrace** -e pthread_create ...
 - cli
 - OpenBLAS
 - Node (V8) worker threads
 - TensorFlow?
 - ...

perf

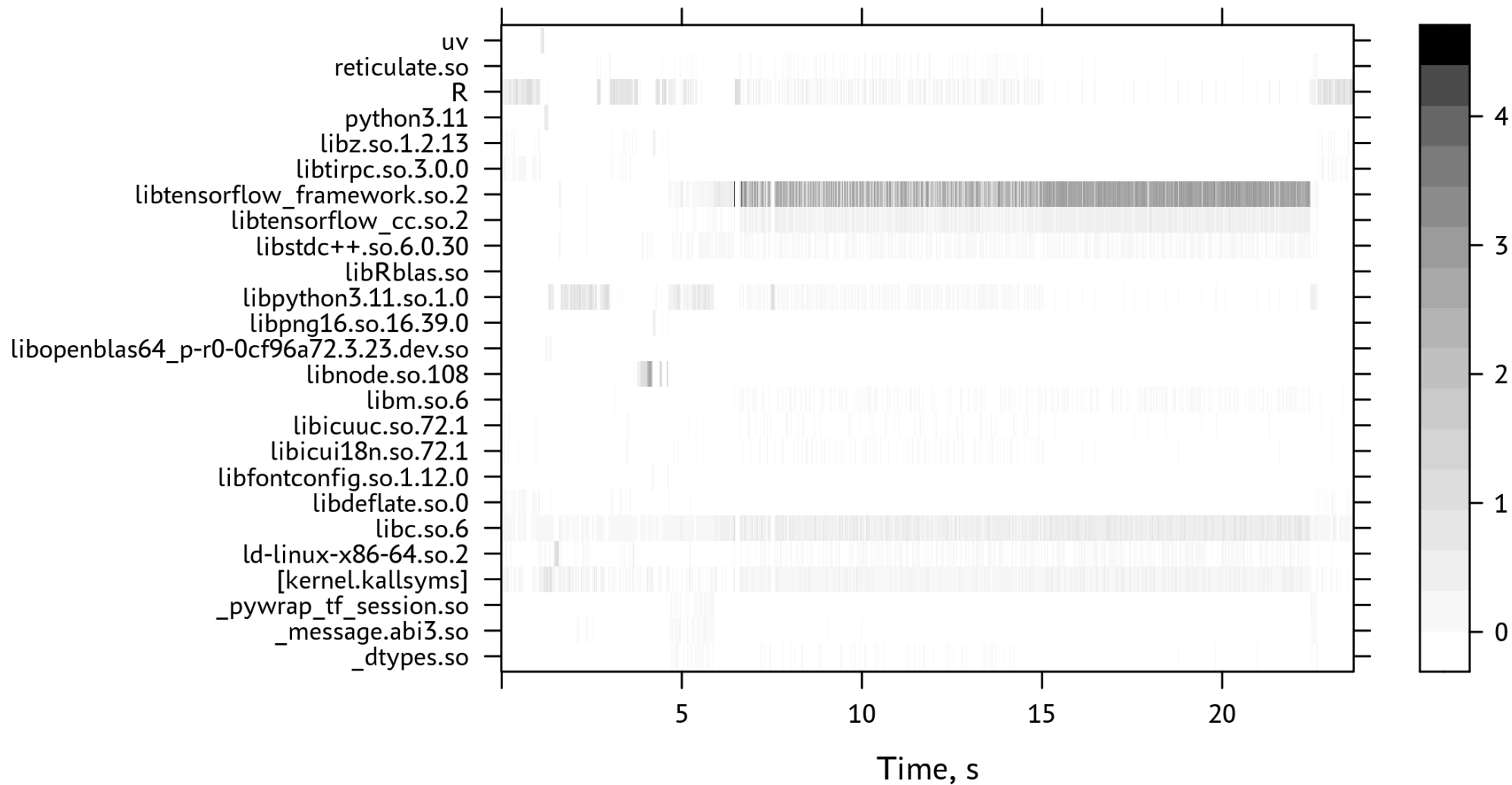
- Profiler built into the Linux kernel
- `task-clock` event $\Leftrightarrow$ “CPU time”
- `perf script` gives ~tabular format
- group by process/DSO,
convert event counts to
CPU usage

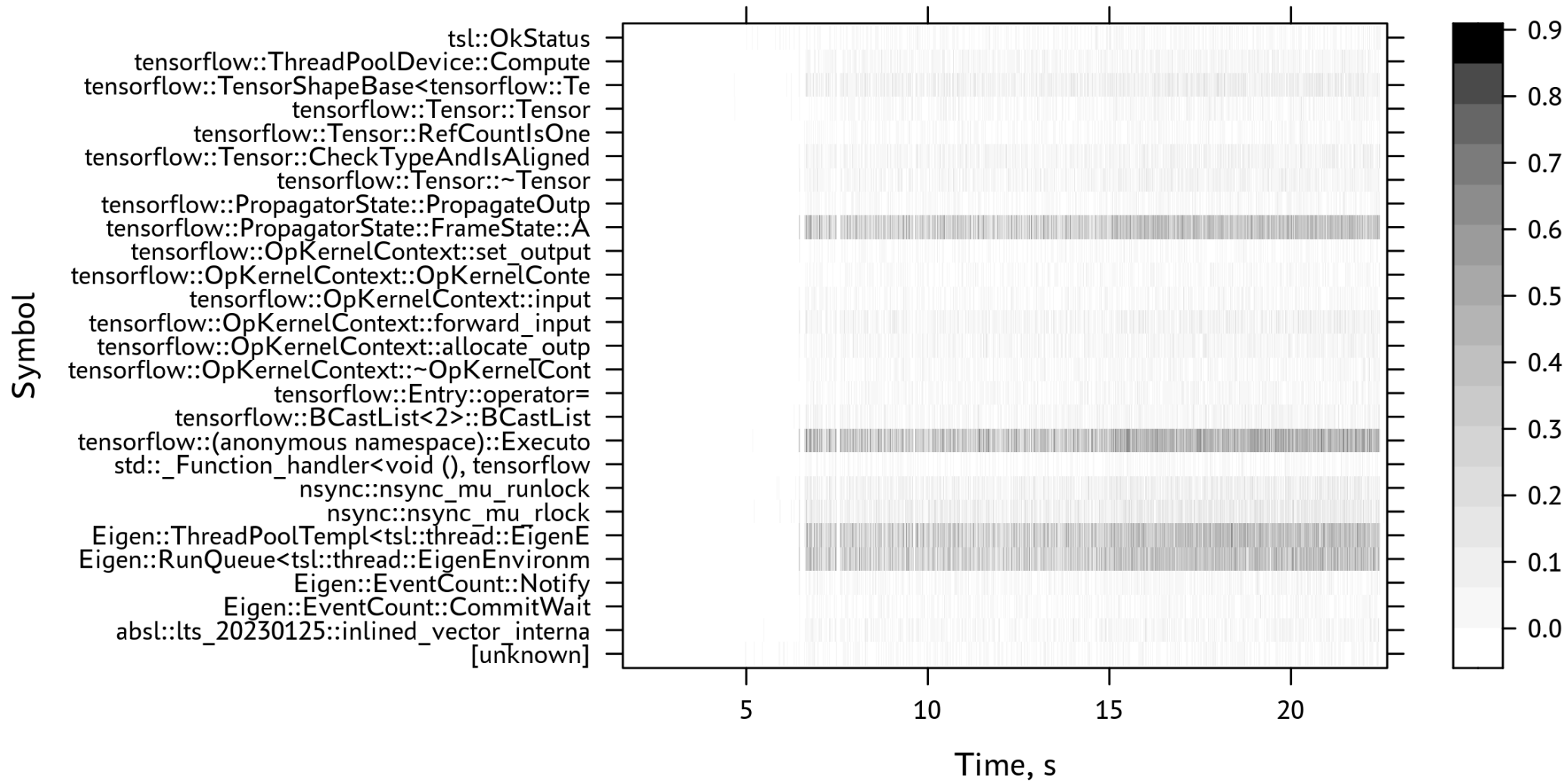


<https://codeberg.org/aitap/perf.cputime>

```
read.perfscript() |> summary(by = 'dso', cutoff = .1) |> plot()
```

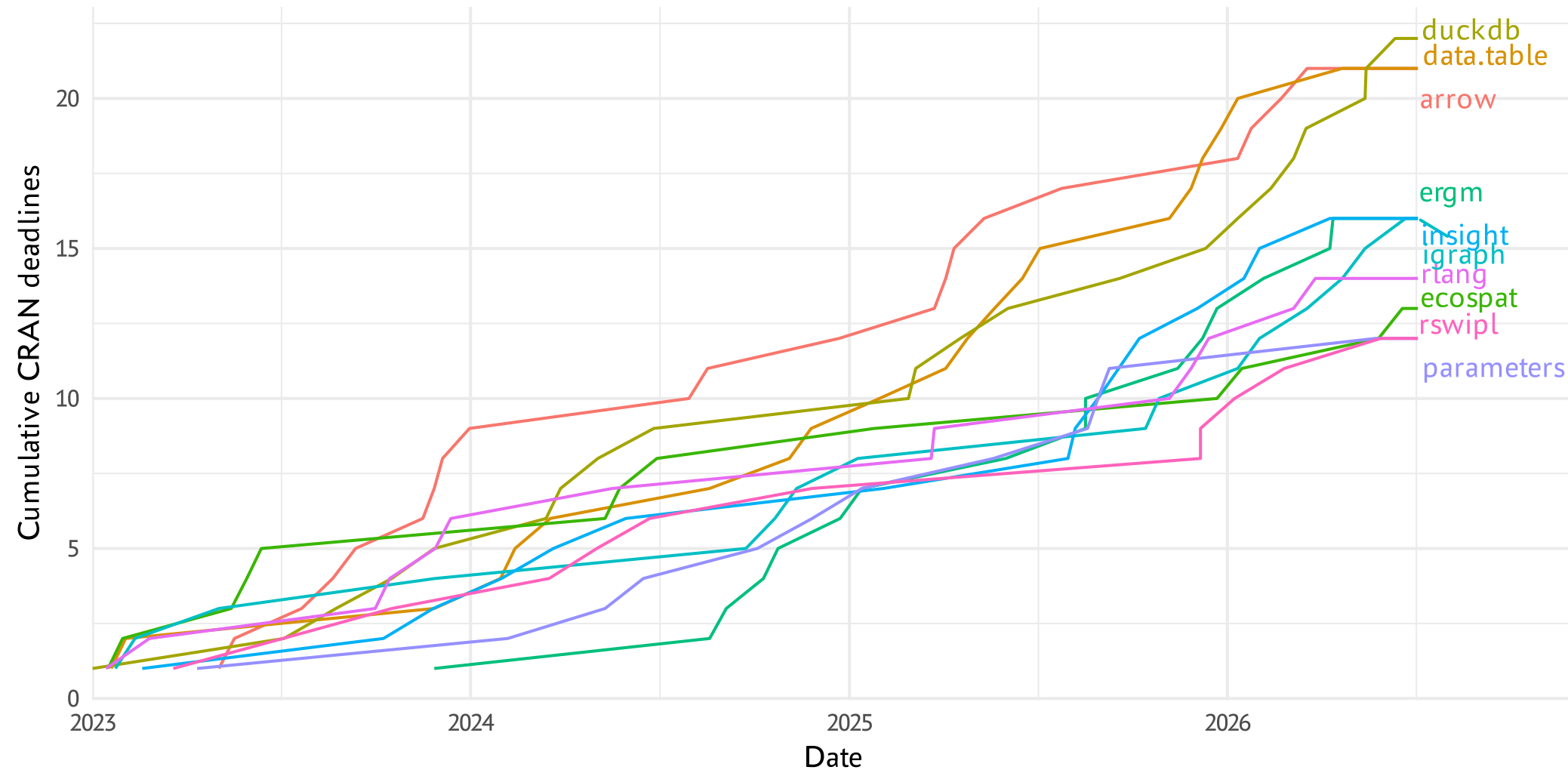
Shared object





Top 10 packages by CRAN deadlines since 2023

Source: tools:::CRAN_package_issues(), captured on 2026-07-04



memcpy(dst, NULL, n) (data.table)

```
> test_check("seqHMM")
forder.c:137:3: runtime error: null pointer passed as argument 2, which
is declared to never be null
    #0 0x7b6289002f38 in flush ../data.table/src/forder.c:137
...
    #5 0x000000742222 in R_doDotCall ../R-devel/src/main/dotcode.c:780
```

```
// forder.c:137
memcpy(gs+gs_n, gs_thread[me], n*sizeof(int));
```



Somehow NULL

memcpy(dst, NULL, 0)

- Fix: don't call memcpy for length = 0
 - `if (!n) return;`
- Reproducer: simulate heavy load
 - `for i in {1..16};`
 `do cat /dev/urandom > /dev/null &`
 `done`
 - or:
 `setDTthreads(0)`
 `system(paste("taskset -p 1",`
 `Sys.getpid()))`

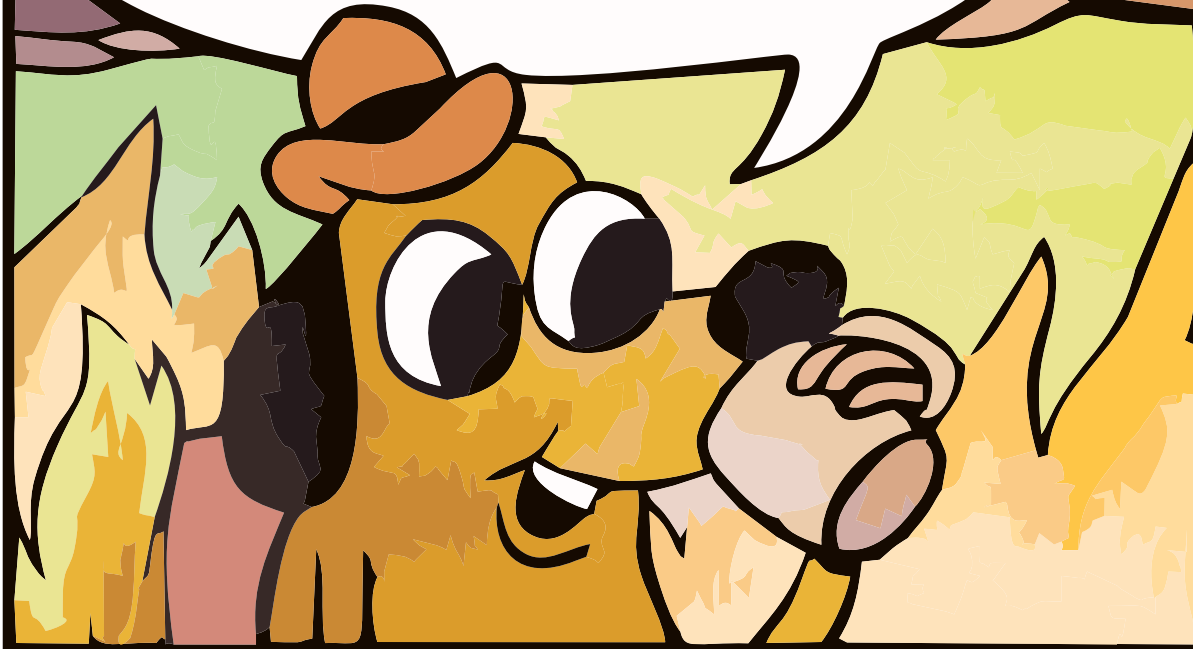
Valgrind, fork(), and threads

(data.table)

Valgrind, fork(), and threads

- Fix: don't combine threads and forks under Valgrind
 - `Sys.setenv(OMP_THREAD_LIMIT =
if (grepl("valgrind",
Sys.getenv("LD_PRELOAD")) "1"
else Sys.getenv("OMP_THREAD_LIMIT",
"2"))`

R CMD check
--as-cran



object of type 'closure'
is not subsettable



TESTS

100%

EXAMPLES

2 3 4
5 6 7

BUILD



0%

COVERAGE

R
data
man
inst

0 / 200
0 / 50
0 / 50
0 / 300



Thanks to

- Patrick Burns
- Mauricio Vargas Sepulveda
- Elias T. Krainski
- David Cortes
- Jouni Helske
- Guillermo Vinue
- Nicholas Tierney
- data.table team

for inspiration in *R Inferno*

for the opportunity to help



Credits

(according to the principles of „dozwolony użytek” / “fair use”)

- The Matrix (1999)
- “xmatrix” screensaver by Jamie Zawinski
- Dinosaur artwork by Shepard Fairey (1998)
- The Hitchhiker's Guide to the Galaxy (2005)
- Yackety Sax by Boots Randolph
- Gunshow #648 “On Fire” by KC Green
- Doom (1993)
- Divine Comedy illustrated by Botticelli, Canto XVIII

The background image is a Renaissance-style painting depicting a scene from Dante's Inferno, specifically the Ninth Circle (Frigidus). The scene is set in a dark, cavernous space with jagged, rocky walls. In the center and foreground, numerous figures are shown in various states of torment, frozen in ice. Some are being crushed by giant hands, while others are being pulled apart. In the upper part of the image, several figures in elaborate robes of purple and red, likely judges or guardians, stand on a raised platform, observing the suffering. The overall tone is somber and dramatic, with a focus on the physical and emotional pain of the damned.

Thanks for watching