

NEW A few hours ago, @rustlang 1.64.0 was released!

Just like every six weeks, at every new release, here's a thread with some of the highlights.

1/15

blog.rust-lang.org
Announcing Rust 1.64.0 | Rust Blog
Empowering everyone to build reliable and efficient software.

4:46 PM · Sep 22, 2022 · Typefully

93 Retweets 3 Quote Tweets 414 Likes



Mara Bos @m\_ou\_se · 1h
Replying to @m\_ou\_se
Rust now has a new async-related trait: IntoFuture.
The .await syntax be used on anything that implements IntoFuture. (Similar to how, with a for loop, you can iterate over anything that implements Intolterator.)
This allows types to provide easier async interfaces.

2/15
2 8 73

Mara Bos @m\_ou\_se · 1h
Today's Rust release also comes with two more async-related tools:
The std::future::poll\_fn function allows you to easily create a future from a closure (like iter::from\_fn for iterators).
The std::task::ready!() macro extracts a Poll::Ready, or returns early on Pending.

3/15
2 4 45

Mara Bos @m\_ou\_se · 1h
The Rust's NonZero types got new methods for (unsigned) addition, multiplication, exponentiation, absolute numbers, and next-power-of-two calculation, that all preserve "non-zero-ness": they return a NonZero type, because we know the result is never zero.

4/15
1 1 41

Mara Bos @m\_ou\_se · 1h
The types for interfacing with C, such as std::os::raw::c\_int and c\_void, have been moved to core::ffi. They are still available in the old (and new) location in `std`, but are now also available in `core`, for #[no\_std] programs.

5/15
1 2 43

Mara Bos @m\_ou\_se · 1h
The OsString type now implements std::fmt::Write, which means you can write to it using the write!() and writeln!() macros.

6/15
1 1 41

Mara Bos @m\_ou\_se · 1h
The IP and socket address types, Ipv4Addr, Ipv6Addr, SocketAddrV4 and SocketAddrV6 now use a minimal, trivial internal representation, instead the libc types.
This clears the way for moving these types to `core` (no\_std), and using these in const fns, in the future.

7/15
1 3 60

Mara Bos @m\_ou\_se · 1h
The atomic compare\_exchange functions no longer require the success memory ordering to be at least as strong as the failure ordering.
(The same restriction was dropped from C++ as part of C++17.)
For example, compare\_exchange(a, b, Release, Acquire) is now accepted.

8/15
1 2 25

Mara Bos @m\_ou\_se · 1h
The std::slice::from\_raw\_parts function is now a const fn, allowing for even more code crimes at compile time. ☺

9/15
1 5 57

Mara Bos @m\_ou\_se · 1h
To allow for sound (unsafe) implementations of reference-counted allocated objects, it's no longer undefined behavior to keep a reference to something that's de-allocated, as long as you don't use it, and the object's bytes all reside in an UnsafeCell (e.g. an atomic).

10/15
1 3 36

Mara Bos @m\_ou\_se · 1h
Rust 1.64 is the last release that ships with the deprecated old language server, RLS. Instead, starting with this release, the rustup distribution now includes @rust\_analyzer.
(Nothing changes for most users. E.g. the VS Code extension will keep handling updates itself.)

11/15
1 6 70

Mara Bos @m\_ou\_se · 1h
Cargo now makes it possible for a Cargo.toml file in a workspace to inherit dependencies and various package properties from the root Cargo.toml file. This makes it easier to handle large workspaces.

12/15
2 7 77

Mara Bos @m\_ou\_se · 1h
Cargo now also supports multiple --target flags, to build for more than one target at once.

13/15
1 2 61

Mara Bos @m\_ou\_se · 1h
Initial support for the Nintendo Switch has been added: aarch64-nintendo-switch-freestanding as a "tier 3" target.
This doesn't include the Rust standard library, but makes it easier to compile no\_std Rust programs for this platform.

14/15

rust-lang/rust #88991 Add Nintendo Switch as tier 3 target
59 comments 11 reviews 8 files +160 -2
jam1garner • September 15, 2021 10 commits
github.com
Add Nintendo Switch as tier 3 target by jam1garner · Pull Request #88991 · rust-lang/rust
Relevant Zulip Discussion This is the first step towards working on incrementally adding support for the Nintendo Switch. After this lands @leo60228 and I will work on ensuring further work is clea...

2 16 76

Mara Bos @m\_ou\_se · 1h
And that's everything for today's thread!
For a more complete list of changes in Rust 1.64, check the release notes:
Rust: github.com/rust-lang/rust
Cargo: github.com/rust-lang/carg
Clippy: github.com/rust-lang/rust

Enjoy!
15/15
1 52