Skip to main content

Stack Overflow

1. About
2. Products
3. For Teams

1. Stack Internal Implement a knowledge platform layer to power your enterprise and AI tools.
2. Stack Data Licensing Get access to top-class technical expertise with trusted & attributed content.
3. Stack Ads Connect your brand to the world’s most trusted technologist communities.
4. Releases Keep up-to-date on features we add to Stack Overflow and Stack Internal.
5. About the company Visit the blog

Loading…

current community

Stack Overflow
help chat
- Meta Stack Overflow

your communities

Sign up or log in to customize your list.

more stack exchange communities

company blog

3. Log in
4. Sign up

AI Assist is now on Stack Overflow. Start a chat to get instant answers from across the network. Sign up to save and share your chats.

Home
Questions
AI Assist
Tags
Challenges
7. Chat
Articles
Users
Companies
Collectives

Communities for your favorite technologies. Explore all Collectives

Can I use the "null pointer optimization" for my own non-pointer types?

Ask Question

Asked 10 years, 6 months ago

Modified 10 months ago

Viewed 5k times

37

When you have an `Option<&T>`, the compiler knows that `NULL` is never a possible value for `&T`, and encodes the `None` variant as `NULL` instead. This allows for space-saving:

```
use std::mem;

fn main() {
   assert_eq!(mem::size_of::<&u8>(), mem::size_of::<Option<&u8>>());
}
```

However, if you do the same with a non-pointer type, there's no extra bits to store that value in and extra space is required:

```
use std::mem;

fn main() {
   // fails because left is 1 and right is 2
   assert_eq!(mem::size_of::<u8>(), mem::size_of::<Option<u8>>());
}
```

In general, this is correct. However, I'd like to opt-in to the optimization because I know that my type has certain impossible values. As a made-up-example, I might have a player character that has an age. The age may be unknown, but will never be as high as `255`:

```
struct Age(u8);

struct Player {
   age: Option<Age>,
}
```

I'd like to be able to inform the optimizer of this constraint - `Age` can never be `255`, so it's safe to use that bit pattern as `None`. Is this possible?

- optimization
- rust

Share

Improve this question

Follow

edited May 23, 2017 at 12:32

CommunityBot

111 silver badge

asked May 23, 2015 at 14:49

Shepmaster

439k116116 gold badges1.3k1.3k silver badges1.5k1.5k bronze badges

Add a comment |

1 Answer

Sorted by: Reset to default

Highest score (default) Trending (recent votes count more) Date modified (newest first) Date created (oldest first)

25

As of Rust 1.28, you can use `std::num::NonZeroU8` (and friends). This acts as a wrapper that tells the compiler the contents of a number will *never* contain a literal zero. It's also why `Option<Box<T>>` is pointer-sized.

Here's an example showing how to create an `Age` and read its payload.

```
use std::num::NonZeroU8;

struct Age(NonZeroU8);

impl Age {
   pub fn new(age: u8) -> Age {
       let age = NonZeroU8::new(age).expect("Age cannot be zero!");
       Age(age)
   }
   pub fn age(&self) -> u8 {
       self.0.get()
   }
}

struct Player {
   age: Option<Age>,
}

fn main() {
   println!("size: {}", std::mem::size_of::<Player>());
   // Output: size: 1
}
```

Share

Improve this answer

Follow

edited Jan 15 at 18:07

Alex Jasmin

39.5k77 gold badges8080 silver badges6868 bronze badges

answered May 23, 2015 at 14:58

DK.

59.7k77 gold badges210210 silver badges171171 bronze badges

Sign up to request clarification or add additional context in comments.

10 Comments

Add a comment

Shepmaster

Shepmaster Over a year ago

Going by the name, I'm assuming that `NonZero` disallows zero values. What about values other than zero? In my example, zero might be valid, but 255 is not.

2015-05-23T15:00:31.797Z+00:00

6

Reply

- Copy link

Matthieu M.

Matthieu M. Over a year ago

@Shepmaster: It's hardcoded to non-0, I suppose with a trait and associated constants one could extend this... however for now you'll have to settle for mathematics. If 255 is your magic value, then applying a `+1` going to storage and `-1` coming from storage (with wrapping arithmetic) will suffice for the `NonZero` magic to work :)

2015-05-23T15:21:01.333Z+00:00

3

Reply

- Copy link

Shepmaster

Shepmaster Over a year ago

@MatthieuM. certainly, I'll just have to do more profiling to see if it's truly worth it. Using less bytes seems like an obvious win; using less bytes *and* mandatory math everywhere is less sure-fire.

2015-05-23T15:22:48.503Z+00:00

3

Reply

- Copy link

Matthieu M.

Matthieu M. Over a year ago

@Shepmaster: Certainly; although, with 0 being an oft-used magic value, a good number of branching instructions are specialized for it (`jz` and `jnz` come to mind), so it's unclear whether an associated constant could beat that.

2015-05-23T15:39:27.217Z+00:00

2

Reply

- Copy link

kennytm

kennytm Over a year ago

BTW `NonZeroU*` will be stabilized in 1.28.

2018-06-10T16:56:31.457Z+00:00

2

Reply

- Copy link

Add a comment | Show 5 more comments

Your Answer

Thanks for contributing an answer to Stack Overflow!

- Please be sure to *answer the question*. Provide details and share your research!

But *avoid* …

- Asking for help, clarification, or responding to other answers.
- Making statements based on opinion; back them up with references or personal experience.

To learn more, see our tips on writing great answers.

Sign up or log in

Sign up using Google

Sign up using Email and Password

Submit

Post as a guest

Name

Email

Required, but never shown

Post as a guest

Name

Email

Required, but never shown

Post Your Answer Discard

By clicking “Post Your Answer”, you agree to our terms of service and acknowledge you have read our privacy policy.

Start asking to get answers

Find the answer to your question by asking.

Ask question

Explore related questions

- optimization
- rust

See similar questions with these tags.

- The Overflow Blog
  Introducing Stack Overflow AI Assist—a tool for the modern developer
  Treating your agents like microservices
- Featured on Meta
  Chat room owners can now establish room guidelines
  AI Assist is now available on Stack Overflow
  Policy: Generative AI (e.g., ChatGPT) is banned

Linked

2
How can I use a negative integer sentinel value without incurring a memory penalty?
3
Can I make size_of::<Option<MyType>>() == size_of::<MyType>() by telling the compiler which values are never inhabited by `MyType`?
116
What is the overhead of Rust's Option type?
35
Why does an enum require extra memory size?
10
Is Option compiled to a runtime check or an instruction jump?
4
How does a repr(C) type handle Option?
1
Translate performance critical loop from C to Rust

Related

5
Is Rust able to optimize local heap allocations?
12
Is it possible to match against a NULL pointer in Rust?
9
How to create a null pointer in a struct?
6
Do empty functions get optimized away in Rust?
59
What is the null pointer optimization in Rust?
5
Is it valid to use ptr::NonNull in FFI?
7
Why is `std::ptr::null` not usable with unsized types?
47
Why does Rust allow calling functions via null pointers?
5
Will Rust optimize away unused function arguments?
4
How to inform the optimizer that NonZeroU32::get will never return zero?

Hot Network Questions

- Why does Blender import PLY vertex colors incorrectly (89 → 0.1 instead of 0.349)?
- TikZ word search diagram v2
- Is there a website where we can find the average stipend/salary for the PhD and Postdoc for each country?
- Who is Patrick and why is he referred to at anti-AfD demonstrations?
- What may cause any version of Microsoft Outlook to run slowly?
- Why does black allow white to have a protected passed central pawn here?

- How to avoid an undesired "nil" in defmacro
- What does Felix mean by telling Bond to pick a contact point while "standing up" in Diamonds Are Forever?
- I2C bus problems when using two distance sensors in Arduino
- Can I Automatically List Supported Kernel Parameters in GRUB?
- EEPROM Decoupling Capacitor
- How does supporting "One China" work without also supporting Taiwan-China reunification?
- Is my homebrew Species "Attuned" balanced for D&D 2024?
- PNP BJT transistor for switching and sourcing to IC
- How do Buddhists interpret the Buddha’s explanation of earthquakes in AN 8.70?
- variable name change causes segfault
- How do you fight a Guardian Scout?
- Is zero-point energy a mathematical artifact?
- Where in the world is the chicken?
- Is "We will review application starting from..." a deadline?
- Storm Sphere and Obstruction
- Why Select Within Distance set to 1m selects every waterbody?
- The state of icy worlds during differentiation and accretion, and how they survive with their water
- Looking for a proper formal substitute for "quick fix" for formal letter

more hot questions

Question feed

Subscribe to RSS

Question feed

To subscribe to this RSS feed, copy and paste this URL into your RSS reader.

Stack Overflow

- Questions
- Help
- Chat

Business

- Stack Internal
- Stack Data Licensing
- Stack Ads

Company

- About
- Press
- Work Here
- Legal
- Privacy Policy
- Terms of Service
- Contact Us
- Cookie Settings
- Cookie Policy

Stack Exchange Network

- Technology
- Culture & recreation
- Life & arts
- Science
- Professional
- Business
- API
- Data