



Rust: Advanced Ownership

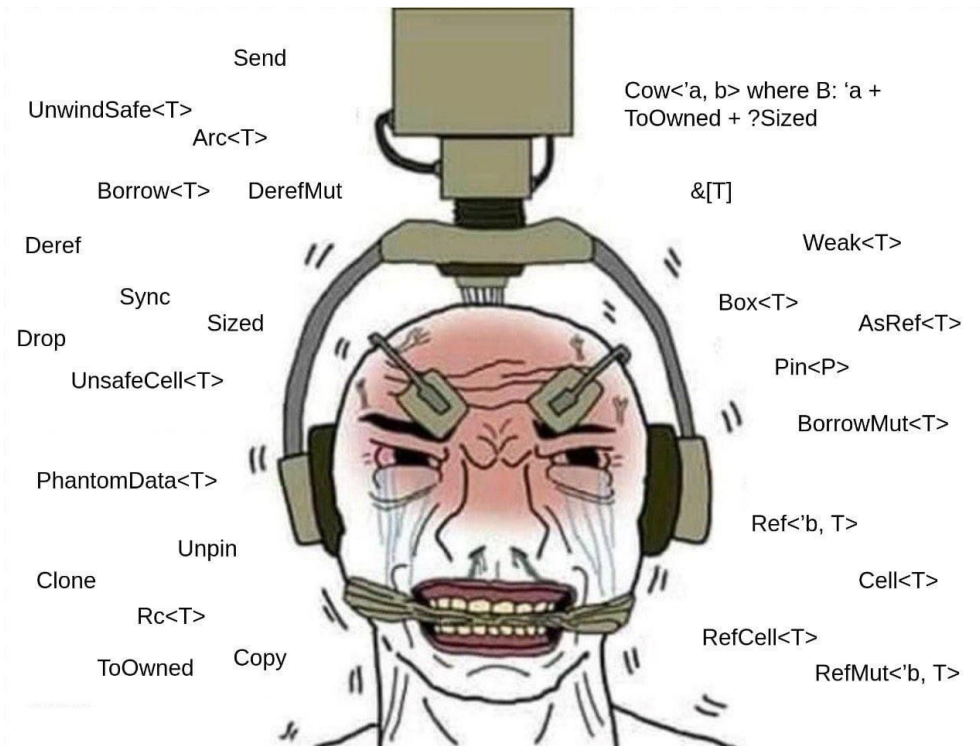
**How I learned to stop worrying and
love the Borrow Checker**

May 2024

Agenda

- 1 Memory and allocations
- 2 Ownership
- 3 Interior mutability
- 4 Multithreaded world

A language empowering everyone to build reliable and efficient software



Tasks and examples

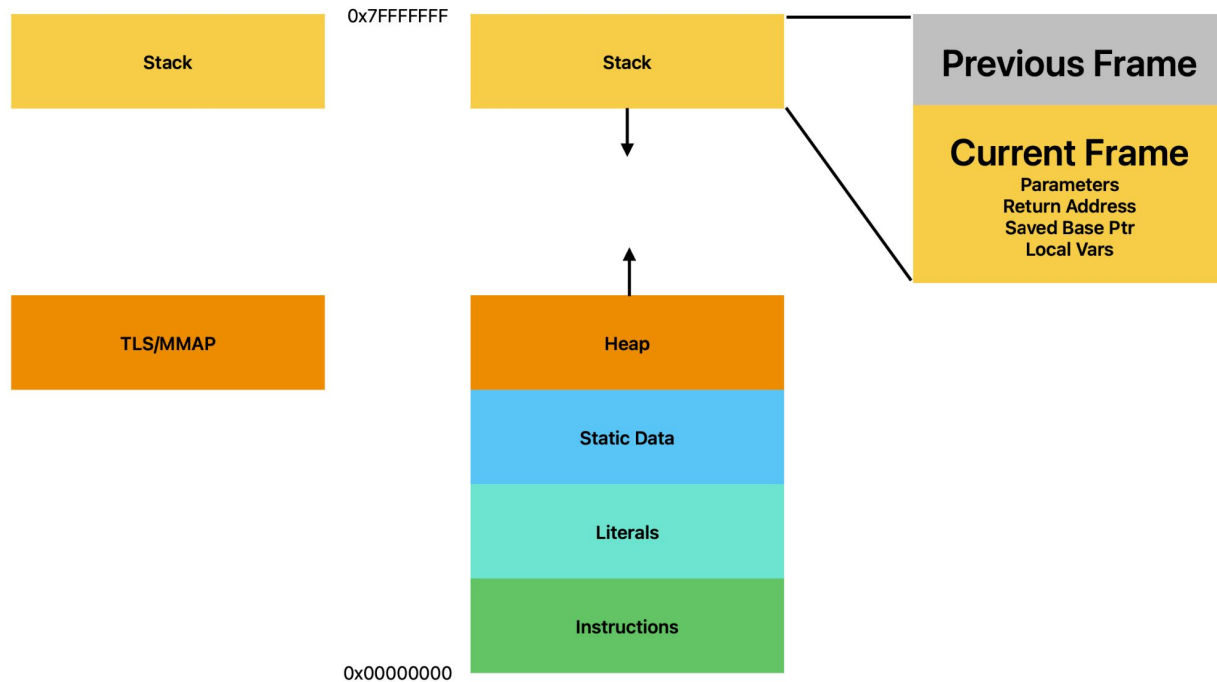
https://github.com/WowVeryLogin/rust_ownership_workshop

Memory and allocations

Memory



Memory



Static

Syntax

StaticItem :

```
static mut? IDENTIFIER : Type ( = Expression )? ;
```

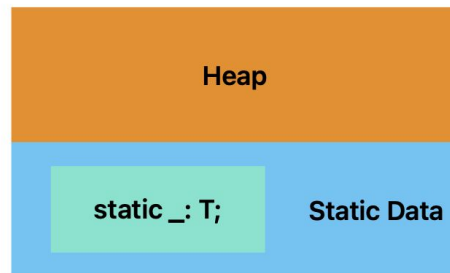
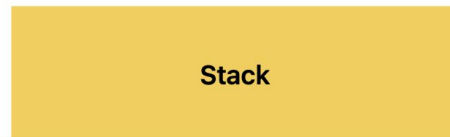
Syntax

ConstantItem :

```
const ( IDENTIFIER | _ ) : Type ( = Expression )? ;
```


Static

```
static a: T = ...;
```

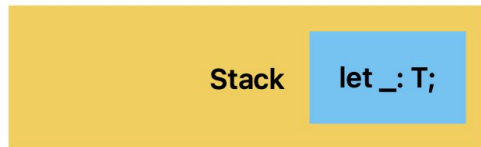


Stack

let a;

let a = ;

let mut a = ;



Heap

Ugly:

malloc/free/new/
delete

Beautiful and safe:

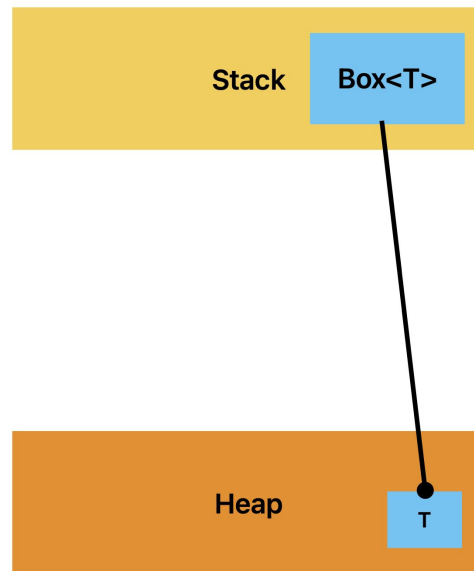
Box<T>

Rc<T>/Arc<T>

Heap

`Box<T>`

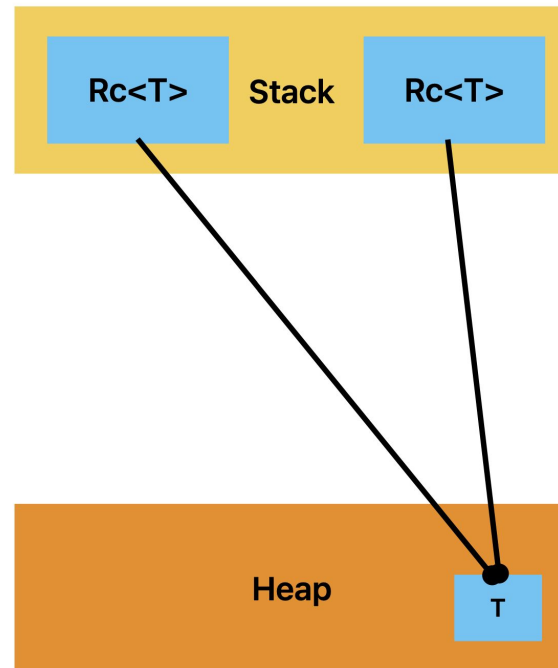
`std::unique_ptr<T>`



Heap

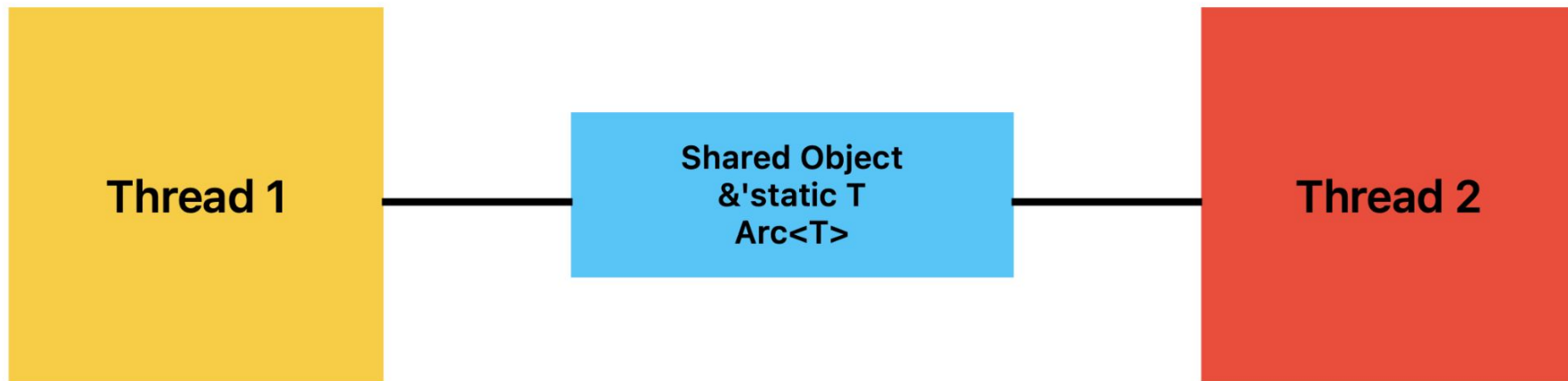
$Rc<T>$ | $Arc<T>$

`std::shared_ptr<T>`



Multithreaded environment

Shared state



Mutex

Mutex<T>

Critical section, exclusive access,
RAII

Atomics

AtomicPtr<T>

load / store

AtomicUsize

compare_exchange

AtomicBool

Memory Ordering

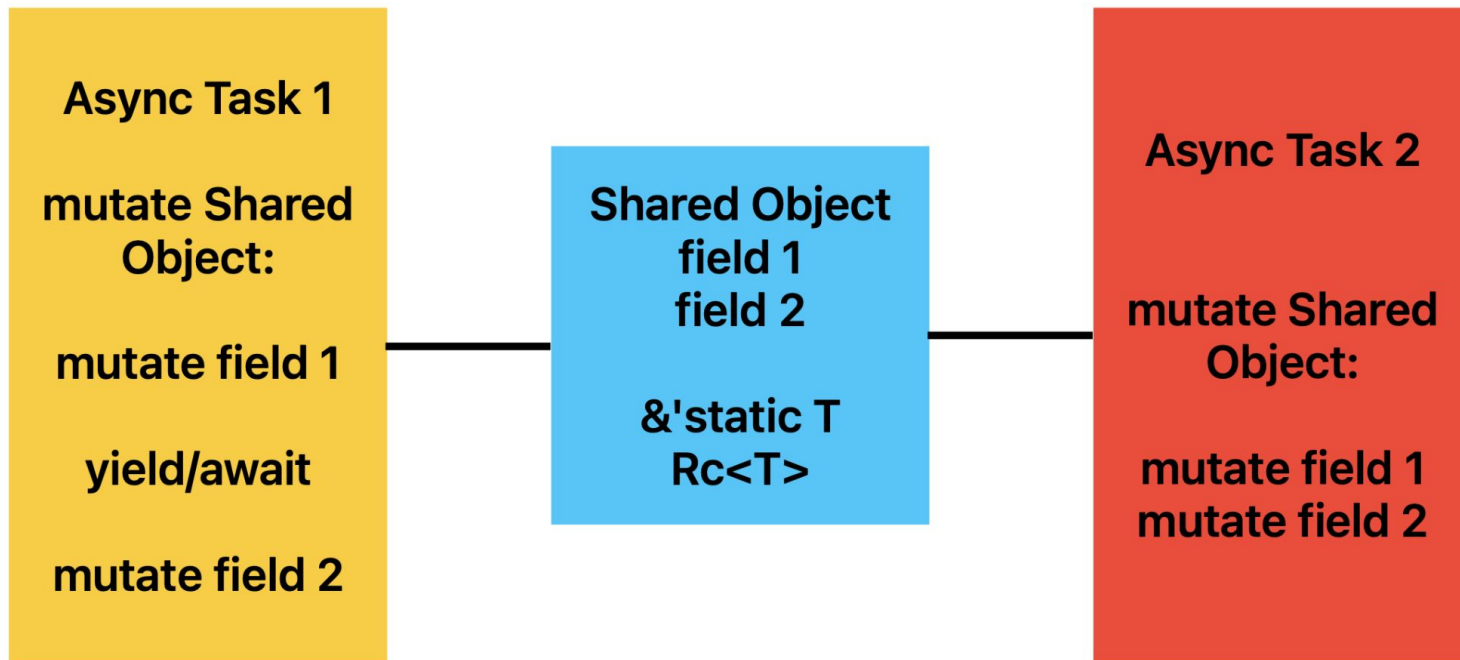


Memory Ordering

- Relaxed
- Store: Release, Load: Acquire
- SeqCst

Interior mutability

Mutate shared



Interior mutability

`Cell<T>` \leftrightarrow `Atomic` `set`, `get`, `swap`, `replace`

`RefCell<T>` \leftrightarrow `Mutex` `borrow`, `borrow_mut`

`Ref<T>`/`RefMut<T>`

About me



Denis Davydov

Cloudflare, FL Team

UK, London



<https://t.me/andrushaTheSlayer>



davidovdd92@mail.ru



<https://github.com/WowVeryLogin>