PANIC! AT THE DISK OH!

You *will* get a headache

```
panic!()
```

```
panic!()
unreachable!()
unimplemented!()
todo!()
```

```
panic!()
unreachable!()
unimplemented!()
todo!()
assert!()
assert_eq!()
assert_ne!()
debug_assert!()
debug_assert_eq!()
debug_assert_ne!()
```

```
thread 'main' panicked at src/main.rs:2:9:
explicit panic
stack backtrace:
   0: __rustc::rust_begin_unwind
             at /rustc/29483883eed69d5fb4db01964cdf2af4d86e9cb2/library/std/src/
panicking.rs:697:5
   1: core::panicking::panic_fmt
             at /rustc/29483883eed69d5fb4db01964cdf2af4d86e9cb2/library/core/src/
panicking.rs:75:14
   2: core::panicking::panic_display
             at /rustc/29483883eed69d5fb4db01964cdf2af4d86e9cb2/library/core/src/
panicking.rs:268:5
   3: core::panicking::panic_explicit
             at /rustc/29483883eed69d5fb4db01964cdf2af4d86e9cb2/library/core/src/
panicking.rs:241:5
   4: playground::main::panic_cold_explicit
             at ./.rustup/toolchains/stable-x86_64-unknown-linux-gnu/lib/rustlib/src/rust/
library/core/src/panic.rs:88:13
   5: playground::main
             at ./src/main.rs:2:9
   6: core::ops::function::FnOnce::call_once
             at ./.rustup/toolchains/stable-x86_64-unknown-linux-gnu/lib/rustlib/src/rust/
library/core/src/ops/function.rs:250:5
note: Some details are omitted, run with `RUST_BACKTRACE=full` for a verbose backtrace.
```

```rust
std::panic::catch_unwind(|| {});
```

```rust
std::panic::catch_unwind(|| {
    panic!("foo");
});
```

```rust
let result = std::panic::catch_unwind(|| {
    panic!("foo");
});
```

Execution

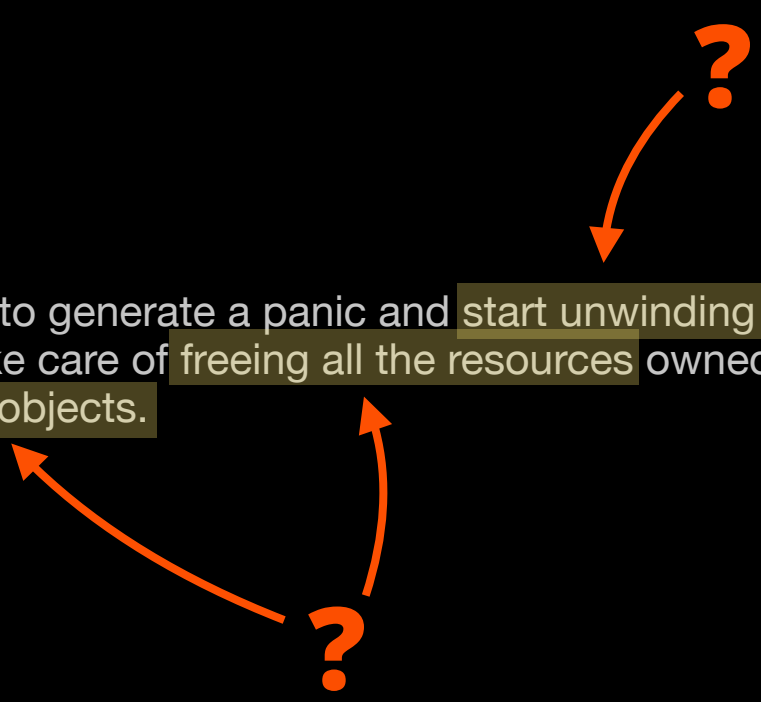
```
   Compiling playground v0.0.1 (/playground)
    Finished `dev` profile [unoptimized + debuginfo] target(s) in 0.52s
     Running `target/debug/playground`

thread 'main' panicked at src/main.rs:3:13:
foo
note: run with `RUST_BACKTRACE=1` environment variable to display a backtrace
```

Standard Output

```
Err(Any { .. })
```

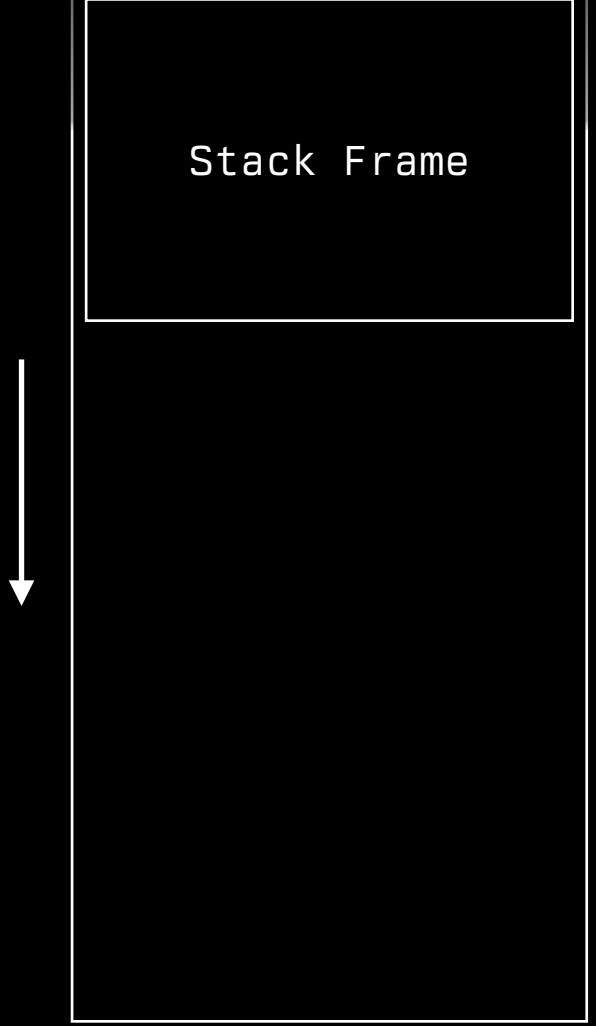
panic!

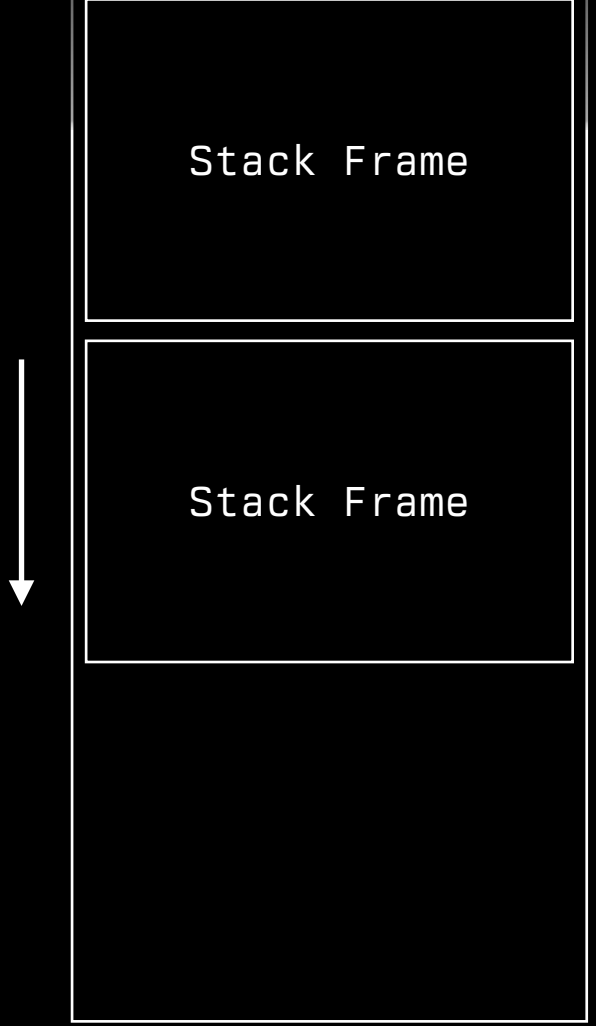
The panic! macro can be used to generate a panic and start unwinding its stack. While unwinding, the runtime will take care of freeing all the resources owned by the thread by calling the destructor of all its objects.

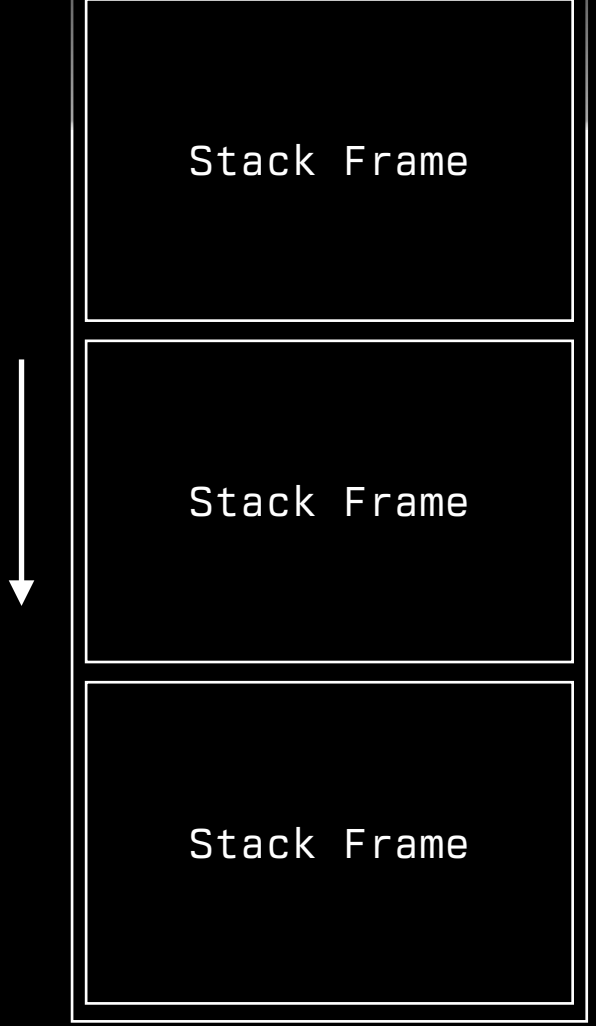


https://doc.rust-lang.org/rust-by-example/std/panic.html

The stack


Stack Frame

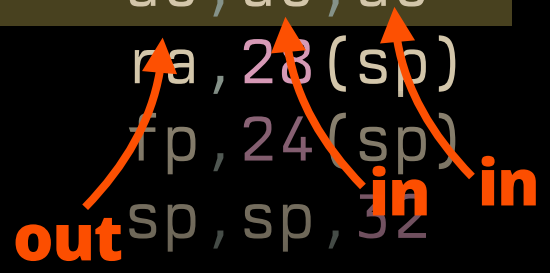

Stack Frame
Stack Frame


Stack Frame
Stack Frame
Stack Frame

```
pub fn square(num: u32) → u32 {
    num * num
}
```

```
pub fn square(num: u32)
  ⟶ u32 {
    num * num
  }
```

```
square:
    addi     sp,sp,-32
    sw       ra,28(sp)
    sw       fp,24(sp)
    addi     fp,sp,32
    mul      a0,a0,a0
    lw       ra,28(sp)
    lw       fp,24(sp)
    addi     sp,sp,32
    jr       ra
```



```
pub fn square(num: u32)
  ⟶ u32 {
    num * num
  }
```

```
square:
    addi     sp,sp,-32
    sw       ra,28(sp)
    sw       fp,24(sp)
    addi     fp,sp,32
    mul      a0,a0,a0
    lw       ra,28(sp)
    lw       fp,24(sp)
    addi     sp,sp,32
    jr       ra
```

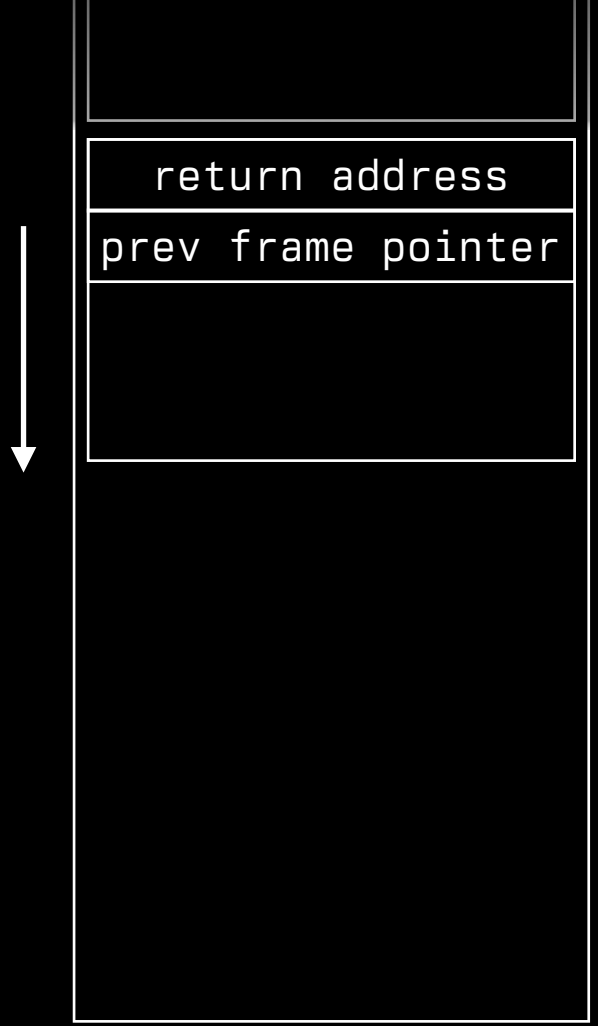
Prologue

stack frame setup

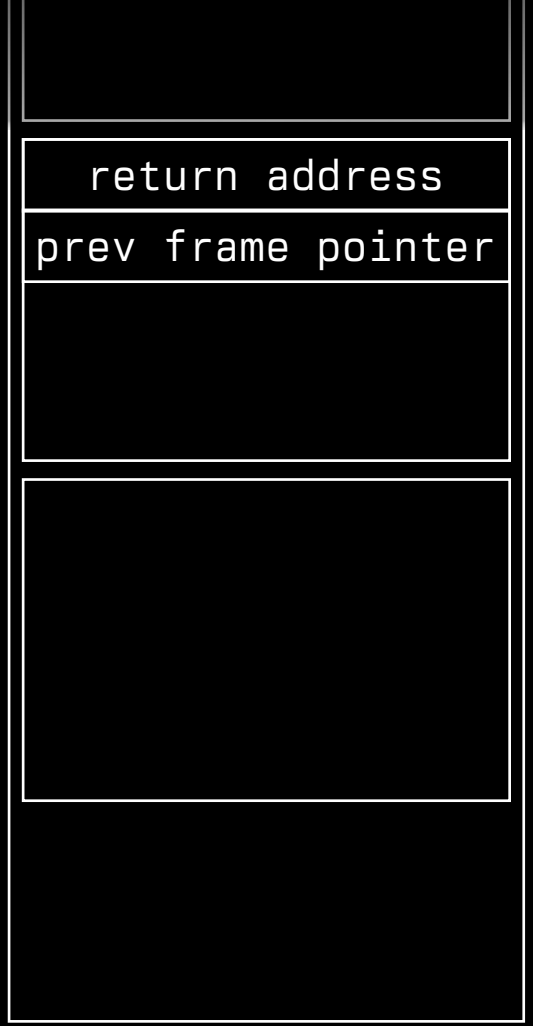
Epilogue

stack frame cleanup

```
square:
    addi    sp,sp,-32
    sw      ra,28(sp)
    sw      fp,24(sp)
    addi    fp,sp,32
    mul     a0,a0,a0
    lw      ra,28(sp)
    lw      fp,24(sp)
    addi    sp,sp,32
    jr      ra
```

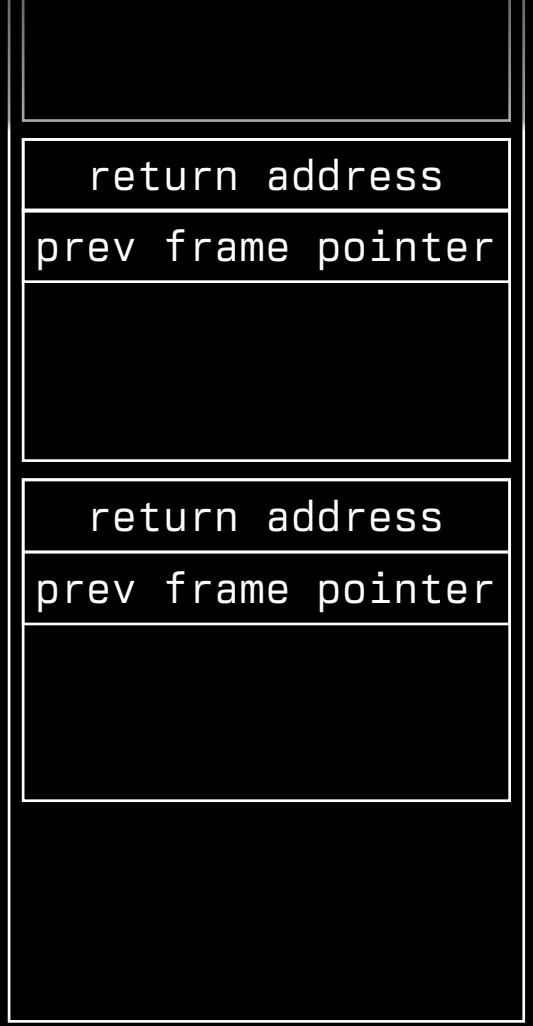



```
square:
    addi    sp,sp,-32
    sw      ra,28(sp)
    sw      fp,24(sp)
    addi    fp,sp,32
    mul     a0,a0,a0
    lw      ra,28(sp)
    lw      fp,24(sp)
    addi    sp,sp,32
    jr      ra
```

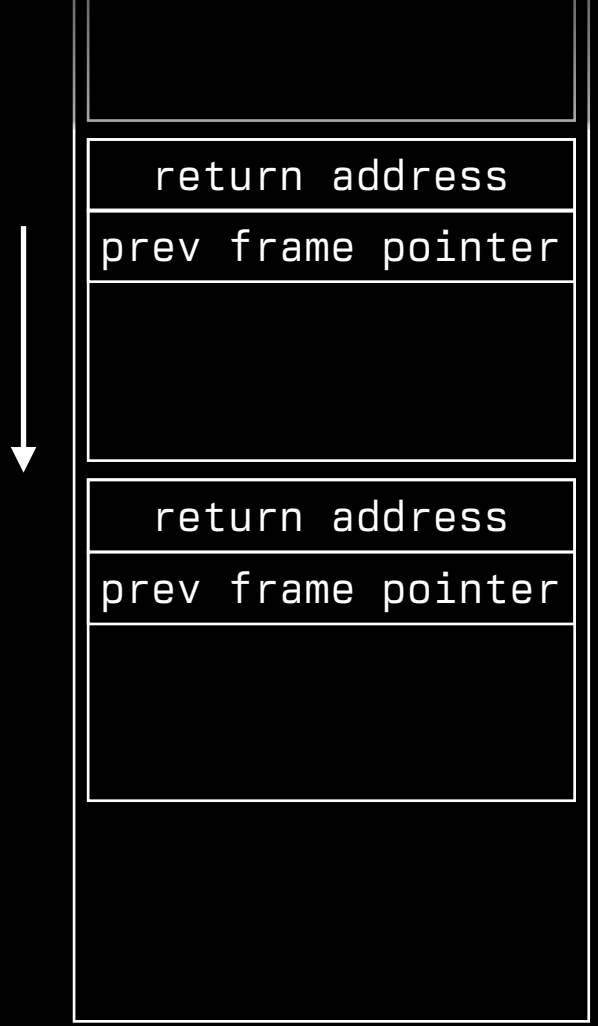



```
square:
    addi    sp,sp,-32
    sw      ra,28(sp)
    sw      fp,24(sp)
    addi    fp,sp,32
    mul     a0,a0,a0
    lw      ra,28(sp)
    lw      fp,24(sp)
    addi    sp,sp,32
    jr      ra
```

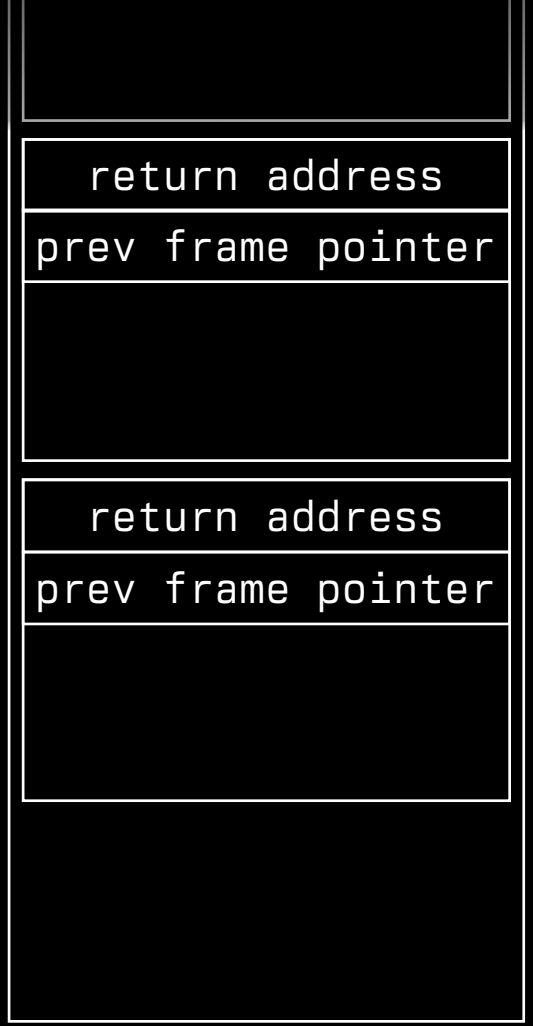

```
square:
    addi    sp,sp,-32
    sw      ra,28(sp)
    sw      fp,24(sp)
    addi    fp,sp,32
    mul     a0,a0,a0
    lw      ra,28(sp)
    lw      fp,24(sp)
    addi    sp,sp,32
    jr      ra
```



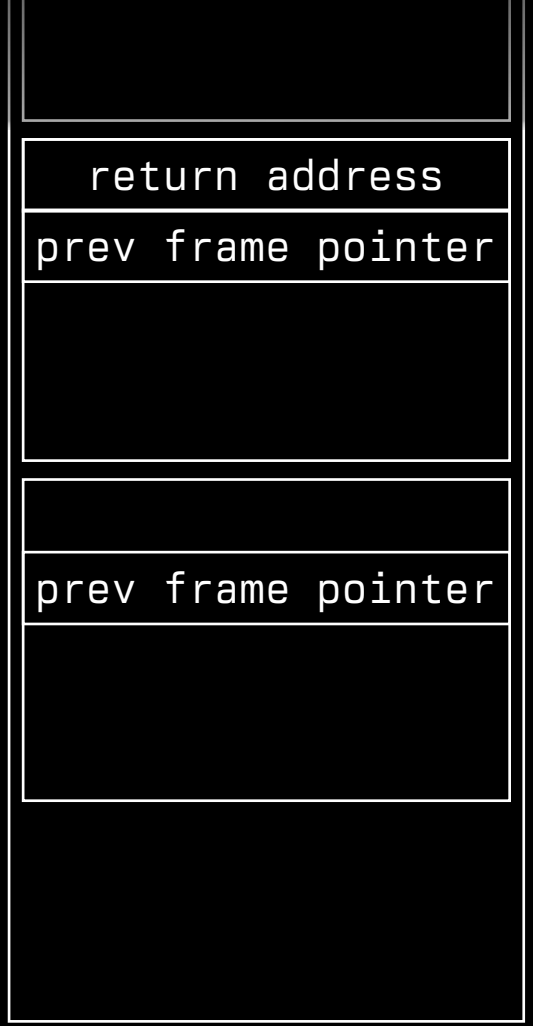

```
square:
        addi    sp,sp,-32
        sw      ra,28(sp)
        sw      fp,24(sp)
        addi    fp,sp,32
        mul     a0,a0,a0
        lw      ra,28(sp)
        lw      fp,24(sp)
        addi    sp,sp,32
        jr      ra
```



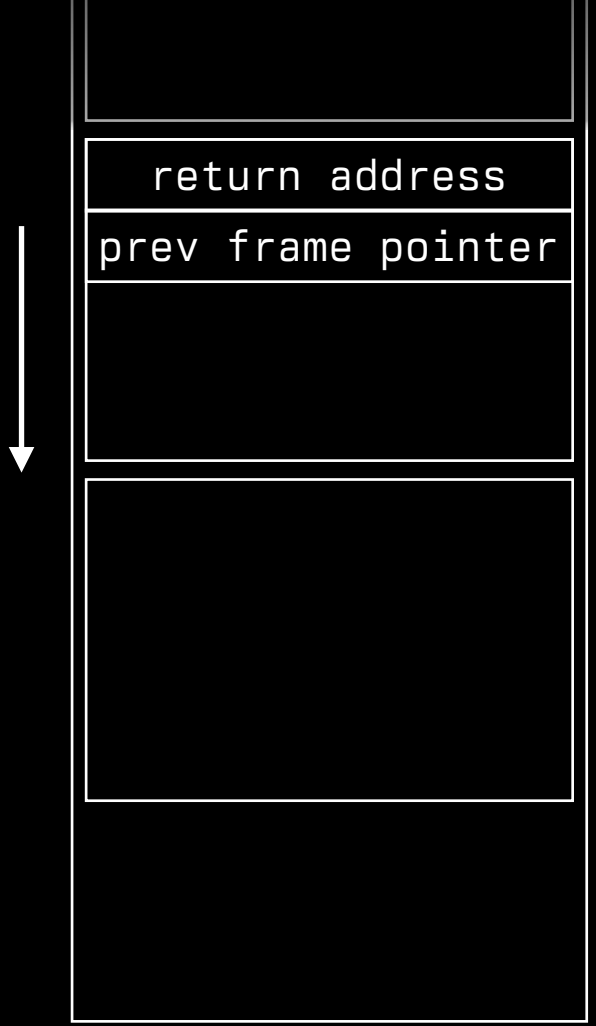

```
square:
        addi    sp,sp,-32
        sw      ra,28(sp)
        sw      fp,24(sp)
        addi    fp,sp,32
        mul     a0,a0,a0
        lw      ra,28(sp)
        lw      fp,24(sp)
        addi    sp,sp,32
        jr      ra
```



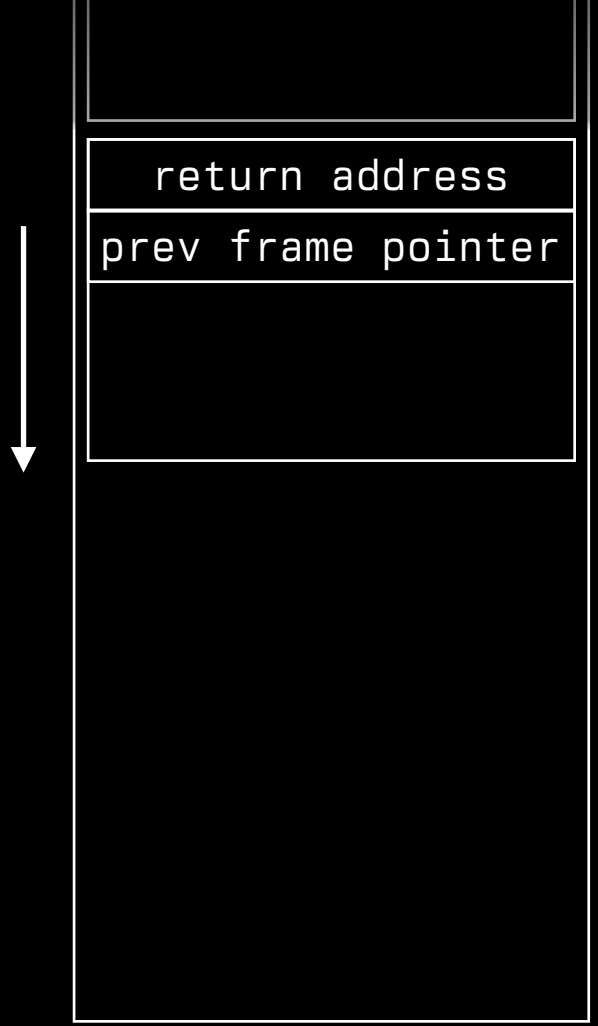

```
square:
        addi    sp,sp,-32
        sw      ra,28(sp)
        sw      fp,24(sp)
        addi    fp,sp,32
        mul     a0,a0,a0
        lw      ra,28(sp)
        lw      fp,24(sp)
        addi    sp,sp,32
        jr      ra
```



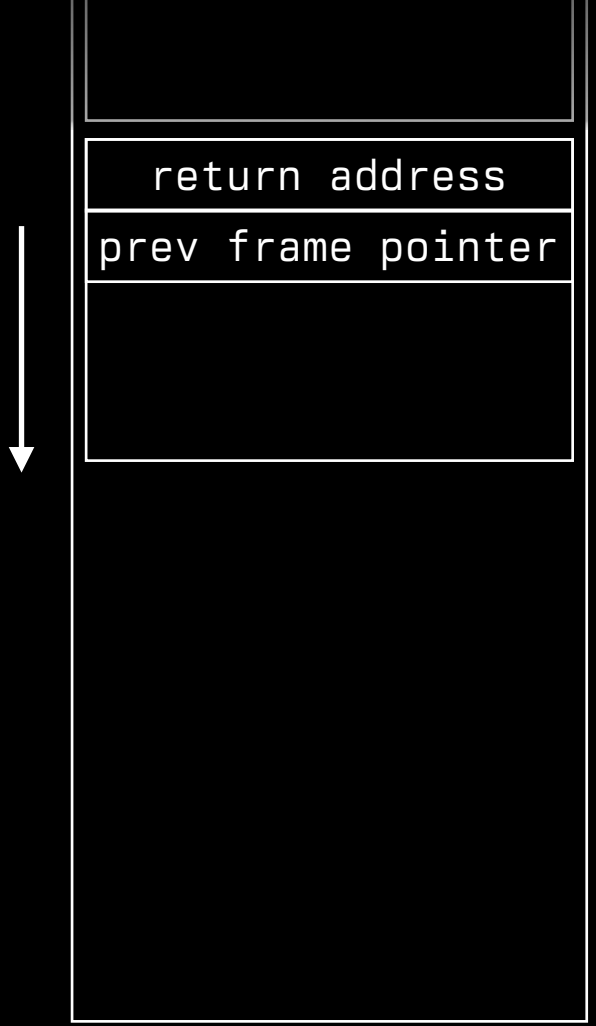

```
square:
        addi    sp,sp,-32
        sw      ra,28(sp)
        sw      fp,24(sp)
        addi    fp,sp,32
        mul     a0,a0,a0
        lw      ra,28(sp)
        lw      fp,24(sp)
        addi    sp,sp,32
        jr      ra
```



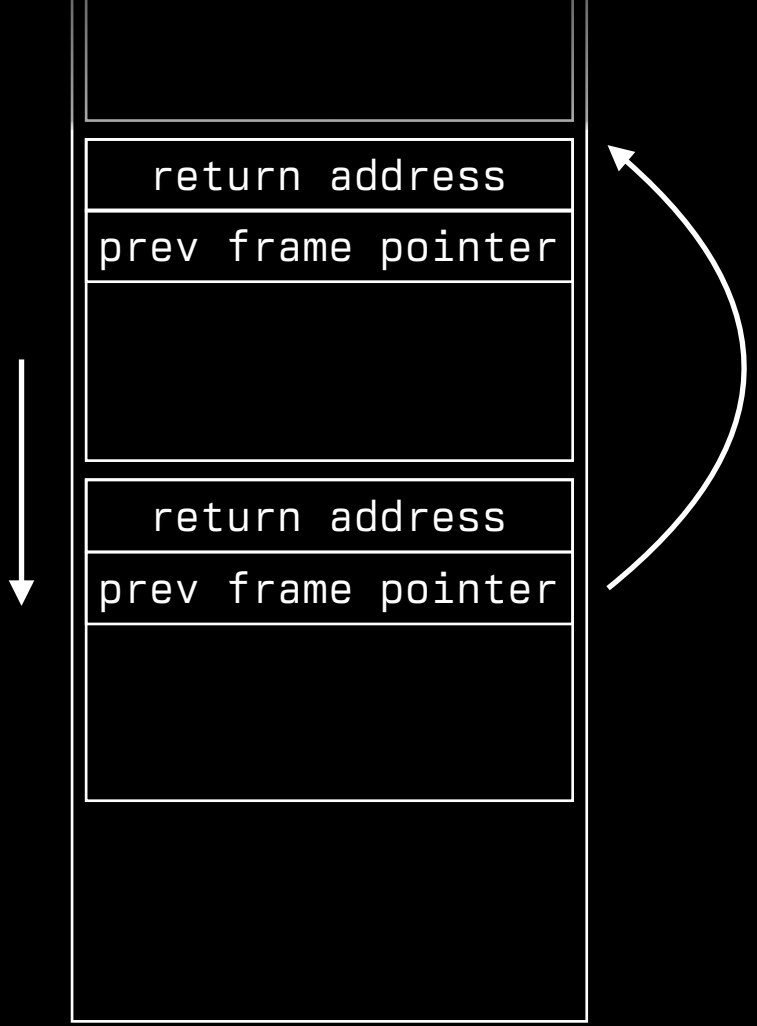

```
square:
    addi    sp,sp,-32
    sw      ra,28(sp)
    sw      fp,24(sp)
    addi    fp,sp,32
    mul     a0,a0,a0
    lw      ra,28(sp)
    lw      fp,24(sp)
    addi    sp,sp,32
    jr      ra
```

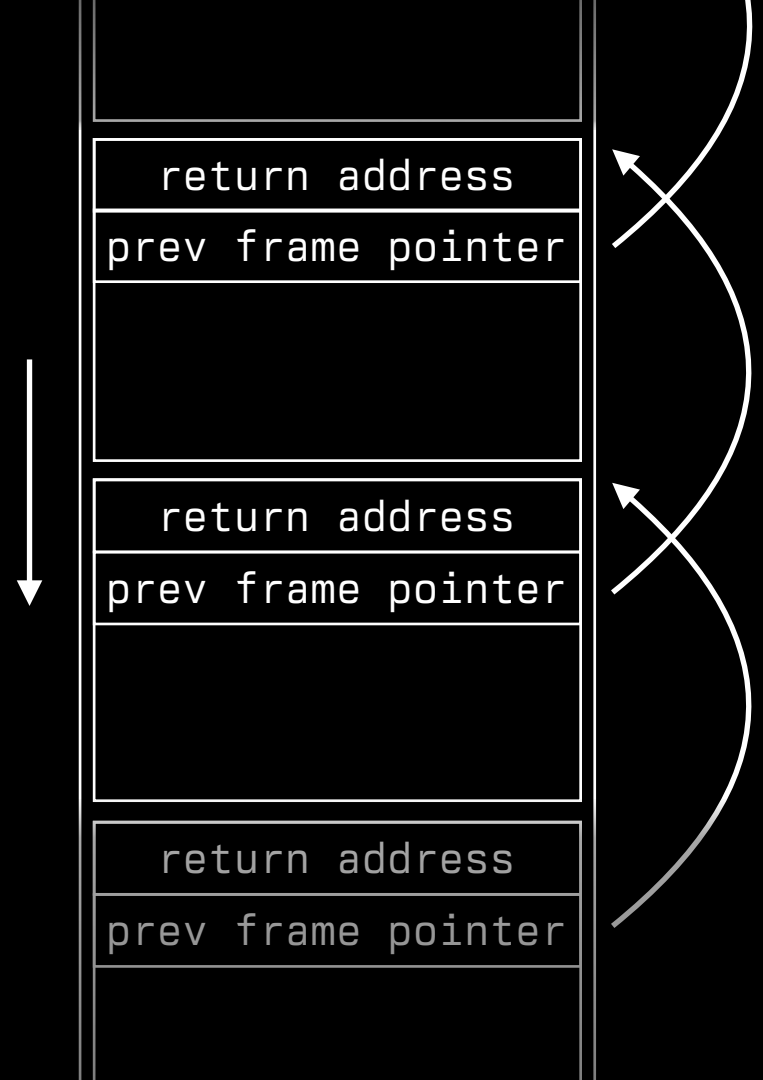



```
square:
    addi    sp,sp,-32
    sw      ra,28(sp)
    sw      fp,24(sp)
    addi    fp,sp,32
    mul     a0,a0,a0
    lw      ra,28(sp)
    lw      fp,24(sp)
    addi    sp,sp,32
    jr      ra
```

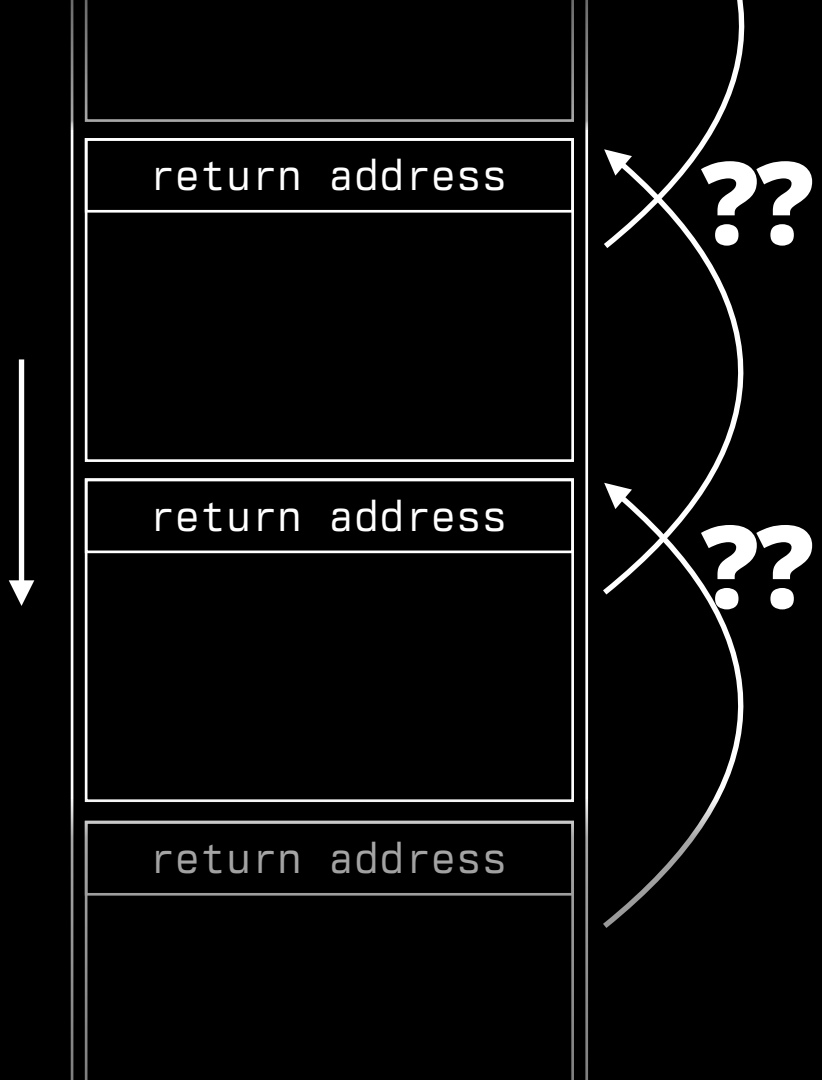



```
square:
        addi    sp,sp,-32
        sw      ra,28(sp)
        sw      fp,24(sp)
        addi    fp,sp,32
        mul     a0,a0,a0
        lw      ra,28(sp)
        lw      fp,24(sp)
        addi    sp,sp,32
        jr      ra
```


return address
prev frame pointer
return address
prev frame pointer



This is a Linked-List!


return address
??
return address
??
return address

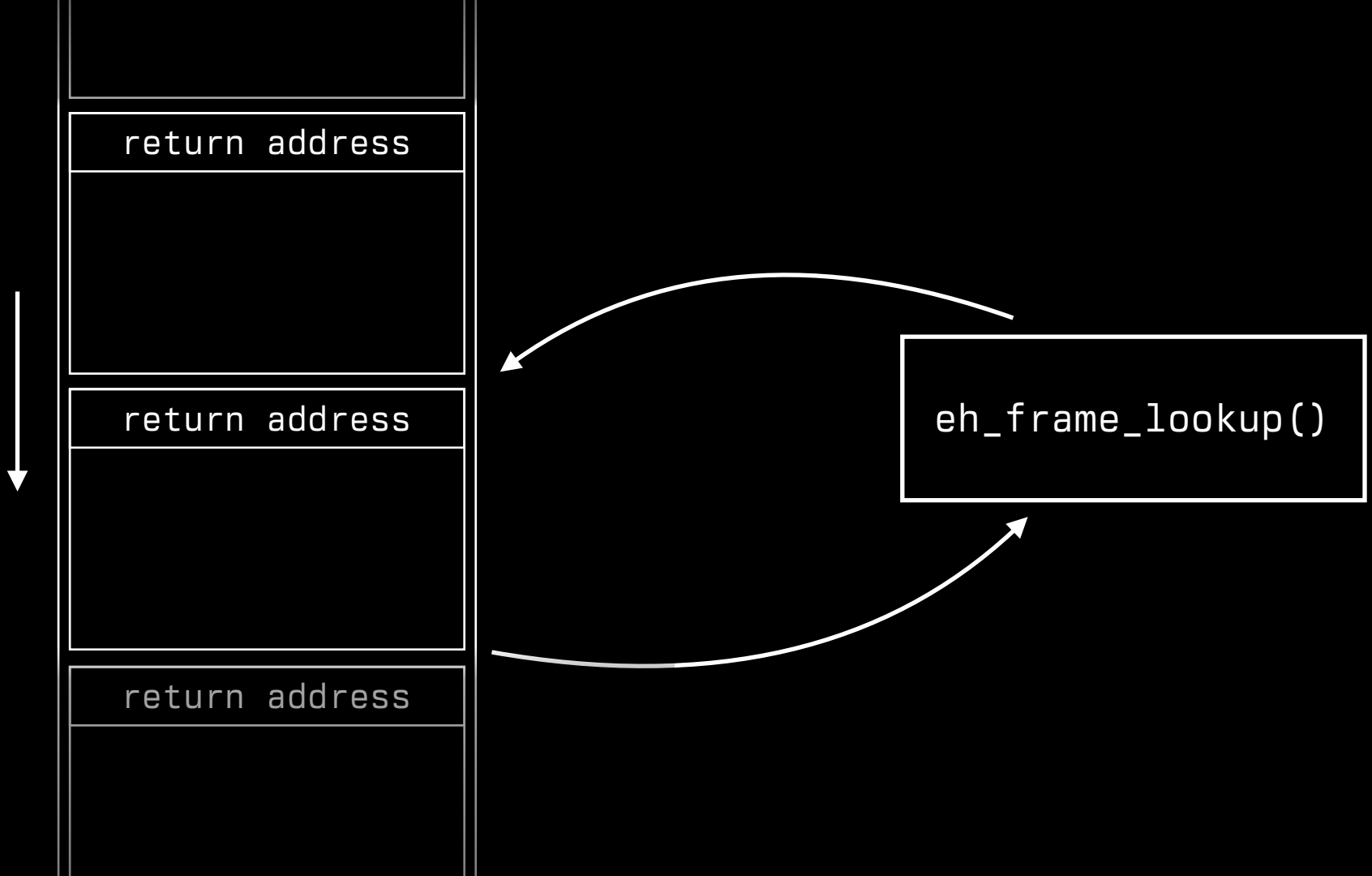
The solution: .eh_frame

```
Sections:
Idx Name              Size     VMA              Type
  0                   00000000 0000000000000000
  1 .dynsym           00000018 00000000000002a8
  2 .gnu.hash         0000001c 00000000000002c0
  3 .hash             00000010 00000000000002dc
  4 .dynstr           00000001 00000000000002ec
  5 .rela.dyn         00089430 00000000000002f0
  6 .rodata           0006e0ae 0000000000089720 DATA
  7 .gcc_except_table 00085edc 00000000000f77d0 DATA
  8 .loader_config    00000008 000000000017d6ac DATA
  9 .eh_frame         001937dc 000000000017d6b8 DATA
 10 .text             00741412 0000000000311ea0 TEXT
 11 .tdata            00040001 0000000000a542b8 DATA
 12 .tbss             00000120 0000000000a942c0 BSS
```

```
├─ FDE  offset=0x00000000'00001274  CIE=0x00000000'00000000
│  ├─  PC range │ 0x00000000'0031b06c..0x00000000'0031b078
│  ├─    Symbol │ .L0  + 0x0
│  ├──[?] DW_CFA_advance_loc(2)          loc += 2      loc = 0x00000000'0031b06e
│  ├──[?] DW_CFA_def_cfa_offset(16)      cfa = %rcx + 16
│  ├──[?] DW_CFA_advance_loc(8)          loc += 8      loc = 0x00000000'0031b076
│  ├──[?] DW_CFA_def_cfa_offset(0)       cfa = %rcx + 0
│  ├──[?] DW_CFA_nop()
│
├─ FDE  offset=0x00000000'0000128c  CIE=0x00000000'00000000
│  ├─  PC range │ 0x00000000'0031b078..0x00000000'0031b084
│  ├─    Symbol │ .L0  + 0x0
│  ├──[?] DW_CFA_advance_loc(2)          loc += 2      loc = 0x00000000'0031b07a
│  ├──[?] DW_CFA_def_cfa_offset(16)      cfa = %rcx + 16
│  ├──[?] DW_CFA_advance_loc(8)          loc += 8      loc = 0x00000000'0031b082
│  ├──[?] DW_CFA_def_cfa_offset(0)       cfa = %rcx + 0
│  ├──[?] DW_CFA_nop()
│
├─ FDE  offset=0x00000000'000012a4  CIE=0x00000000'00000000
│  ├─  PC range │ 0x00000000'0031b084..0x00000000'0031b090
│  ├─    Symbol │ .L0  + 0x0
│  ├──[?] DW_CFA_advance_loc(2)          loc += 2      loc = 0x00000000'0031b086
│  ├──[?] DW_CFA_def_cfa_offset(16)      cfa = %rcx + 16
│  ├──[?] DW_CFA_advance_loc(8)          loc += 8      loc = 0x00000000'0031b08e
│  ├──[?] DW_CFA_def_cfa_offset(0)       cfa = %rcx + 0
│  ├──[?] DW_CFA_nop()
```

```
├─ FDE  offset=0x00000000'00001274  CIE=0x00000000'00000000
│  ├─  PC range │ 0x00000000'0031b06c..0x00000000'0031b078
│  ├─    Symbol │ .L0  + 0x0
│  ├──[?] DW_CFA_advance_loc(2)          loc += 2      loc = 0x00000000'0031b06e
│  ├──[?] DW_CFA_def_cfa_offset(16)      cfa = %rcx + 16
│  ├──[?] DW_CFA_advance_loc(8)          loc += 8      loc = 0x00000000'0031b076
│  ├──[?] DW_CFA_def_cfa_offset(0)       cfa = %rcx + 0
│  ├──[?] DW_CFA_nop()
│
├─ FDE  offset=0x00000000'0000128c  CIE=0x00000000'00000000
│  ├─  PC range │ 0x00000000'0031b078..0x00000000'0031b084
│  ├─    Symbol │ .L0  + 0x0
│  ├──[?] DW_CFA_advance_loc(2)          loc += 2      loc = 0x00000000'0031b07a
│  ├──[?] DW_CFA_def_cfa_offset(16)      cfa = %rcx + 16
│  ├──[?] DW_CFA_advance_loc(8)          loc += 8      loc = 0x00000000'0031b082
│  ├──[?] DW_CFA_def_cfa_offset(0)       cfa = %rcx + 0
│  ├──[?] DW_CFA_nop()
│
├─ FDE  offset=0x00000000'000012a4  CIE=0x00000000'00000000
│  ├─  PC range │ 0x00000000'0031b084..0x00000000'0031b090
│  ├─    Symbol │ .L0  + 0x0
│  ├──[?] DW_CFA_advance_loc(2)          loc += 2      loc = 0x00000000'0031b086
│  ├──[?] DW_CFA_def_cfa_offset(16)      cfa = %rcx + 16
│  ├──[?] DW_CFA_advance_loc(8)          loc += 8      loc = 0x00000000'0031b08e
│  ├──[?] DW_CFA_def_cfa_offset(0)       cfa = %rcx + 0
│  ├──[?] DW_CFA_nop()
```

```
FDE  offset=0x00000000'00001274  CIE=0x00000000'00000000
  PC range | 0x00000000'0031b06c..0x00000000'0031b078
    Symbol | .L0  + 0x0
  [?] DW_CFA_advance_loc(2)          loc += 2      loc = 0x00000000'0031b06e
  [?] DW_CFA_def_cfa_offset(16)      cfa = %rcx + 16
  [?] DW_CFA_advance_loc(8)          loc += 8      loc = 0x00000000'0031b076
  [?] DW_CFA_def_cfa_offset(0)       cfa = %rcx + 0
  [?] DW_CFA_nop()

FDE  offset=0x00000000'0000128c  CIE=0x00000000'00000000
  PC range | 0x00000000'0031b078..0x00000000'0031b084
    Symbol | .L0  + 0x0
  [?] DW_CFA_advance_loc(2)          loc += 2      loc = 0x00000000'0031b07a
  [?] DW_CFA_def_cfa_offset(16)      cfa = %rcx + 16
  [?] DW_CFA_advance_loc(8)          loc += 8      loc = 0x00000000'0031b082
  [?] DW_CFA_def_cfa_offset(0)       cfa = %rcx + 0
  [?] DW_CFA_nop()

FDE  offset=0x00000000'000012a4  CIE=0x00000000'00000000
  PC range | 0x00000000'0031b084..0x00000000'0031b090
    Symbol | .L0  + 0x0
  [?] DW_CFA_advance_loc(2)          loc += 2      loc = 0x00000000'0031b086
  [?] DW_CFA_def_cfa_offset(16)      cfa = %rcx + 16
  [?] DW_CFA_advance_loc(8)          loc += 8      loc = 0x00000000'0031b08e
  [?] DW_CFA_def_cfa_offset(0)       cfa = %rcx + 0
  [?] DW_CFA_nop()
```


return address
return address
return address
eh_frame_lookup()

How do we find & call the destructors??

The solution: personality_routine

```
unsafe fn eh_personality(
    version: c_int,
    actions: UnwindAction,
    _exception_class: u64,
    exception: *mut UnwindException,
    unwind_ctx: &mut UnwindContext<'_>,
) -> UnwindReasonCode {
```

provided by `.eh_frame`!


```
unsafe fn eh_personality(…) → UnwindReasonCode
```


```
unsafe fn eh_personality(…) → UnwindReasonCode
```


```
unsafe fn eh_personality(…) → UnwindReasonCode
```

```
UnwindReasonCode::CONTINUE_UNWIND
UnwindReasonCode::HANDLER_FOUND
UnwindReasonCode::TERMINATE
UnwindReasonCode::END_OF_STACK
…
```

Language Specific Data (LSDA):

- **Rust (LLVM) languages**
- **A big table of** *`Callsite`* **structs**
- **also provided by** `.eh_frame`!

```
struct Callsite {
    start: u64,
    len: u64,
    landing_pad: u64,
    landing_pad_type: LandingPadType,
}

enum LandingPadType {
    Catch,
    Cleanup,
}
```

Catch Landing Pad

```
fn foo() {
    catch_unwind(|| {});
}
```

⇒ *LandingPadType*::*Catch*

Cleanup Landing Pad

```
fn foo() {
    drop(bar);
    drop(baz);
}
```

=> *LandingPadType::Cleanup*

Unwind Phases

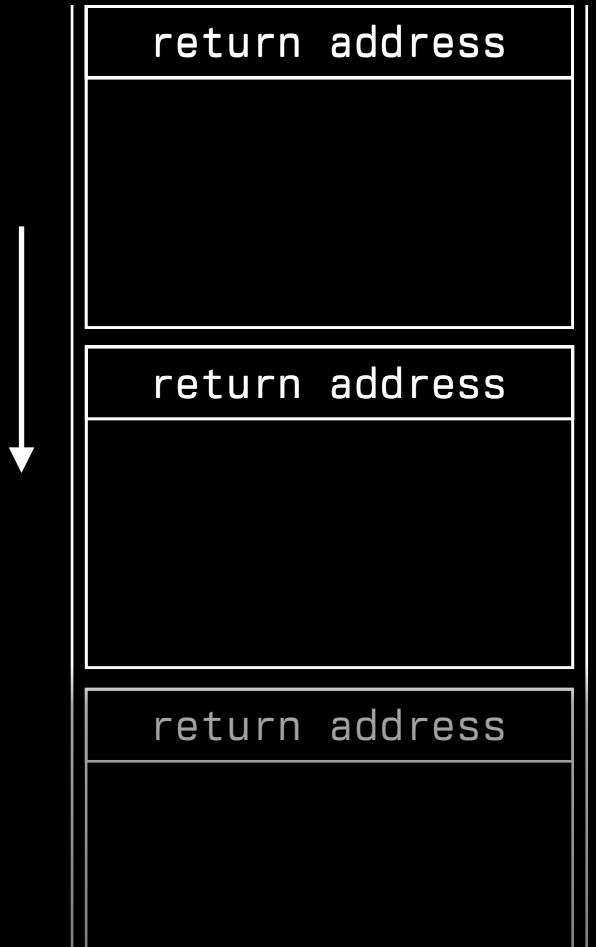
Phase I:

- Search for a `catch` pad
- *ignore* `cleanup` pad

Phase II:

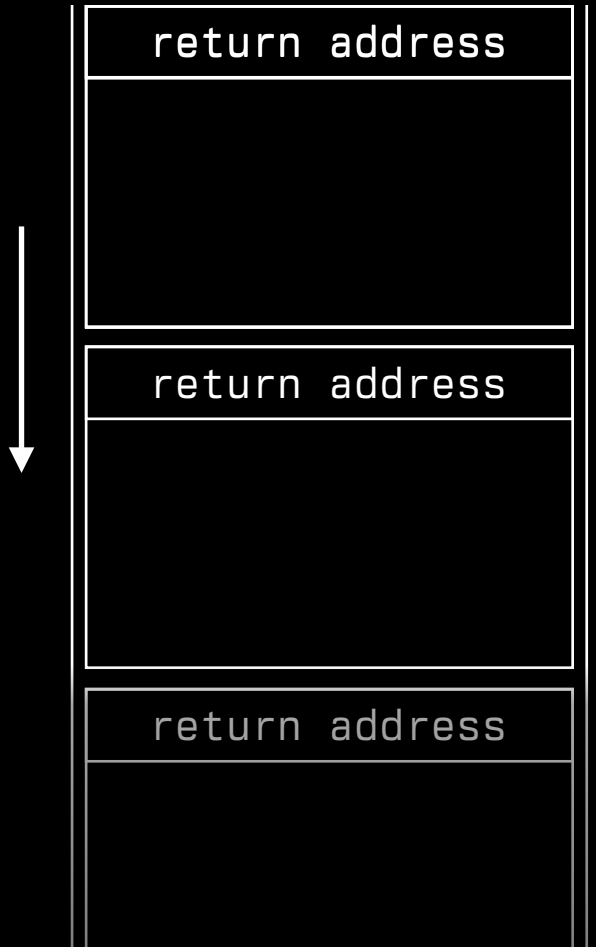
- Execute `cleanup` pad
- Transfer control to `catch` pad

Phase I: Search



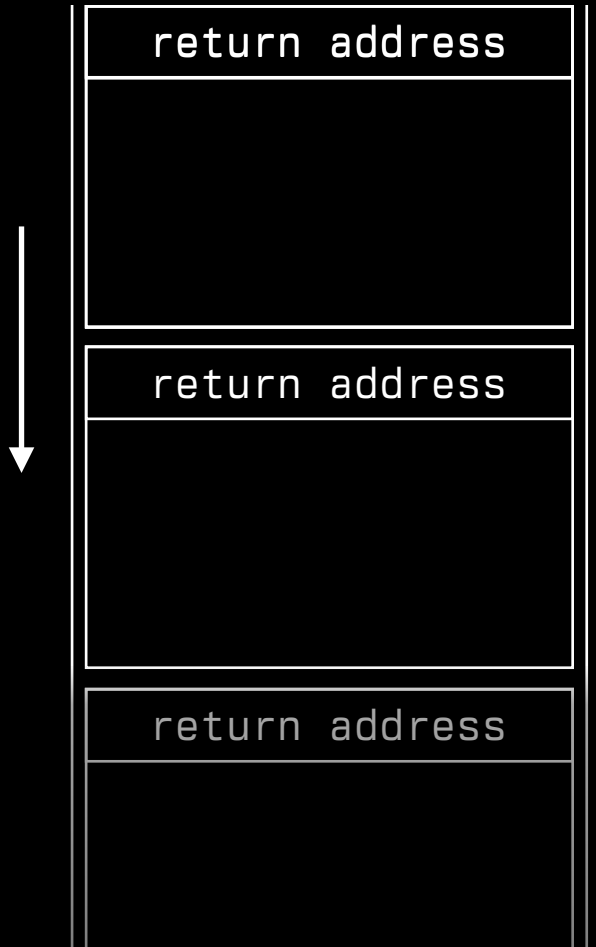

```
unsafe fn eh_personality(…) → UnwindReasonCode
```

Phase I: Search




```
unsafe fn eh_personality(…) → UnwindReasonCode
```

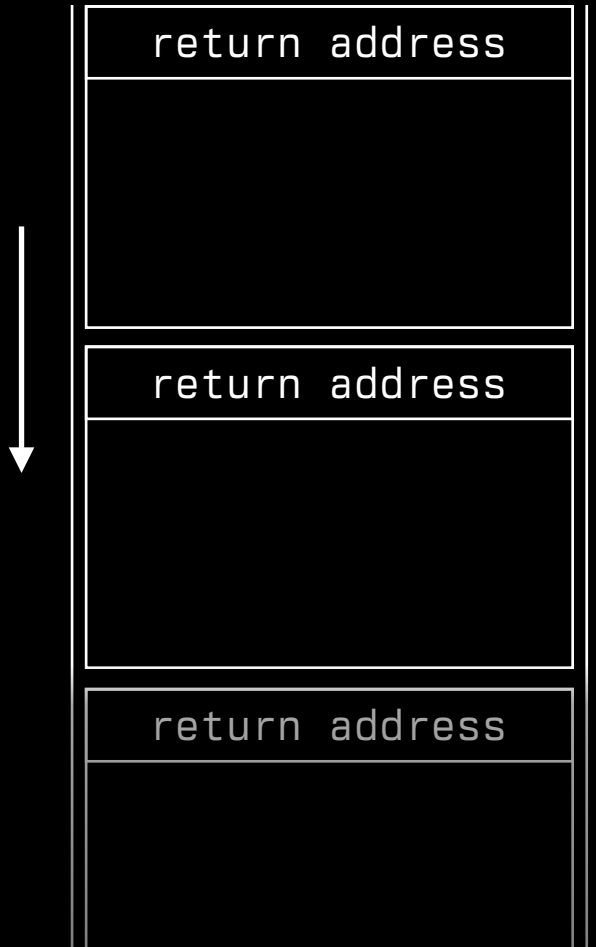
Phase I: Search




```
unsafe fn eh_personality(…) → UnwindReasonCode
```

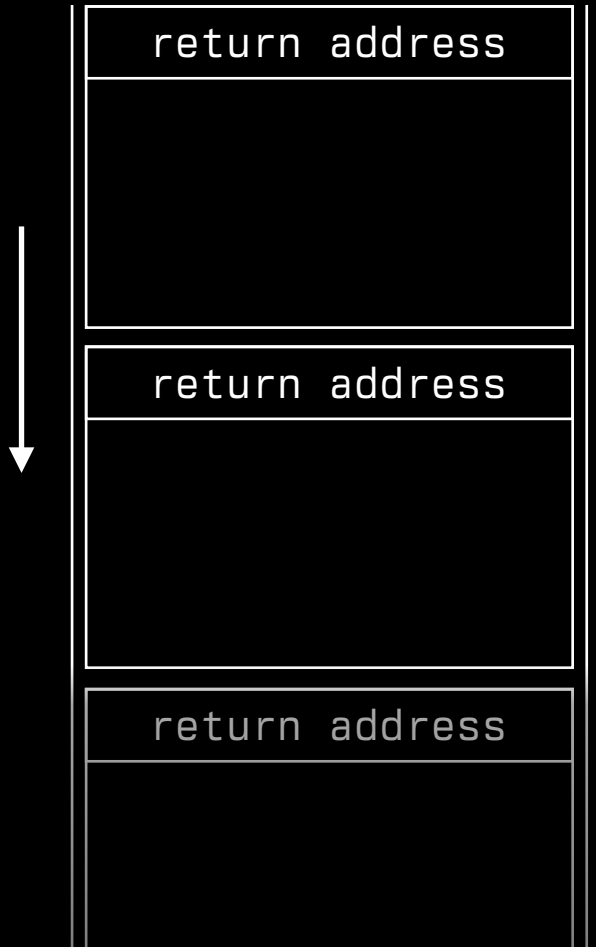
```
UnwindReasonCode :: CONTINUE_UNWIND
```

Phase I: Search




```
unsafe fn eh_personality(…) → UnwindReasonCode
```

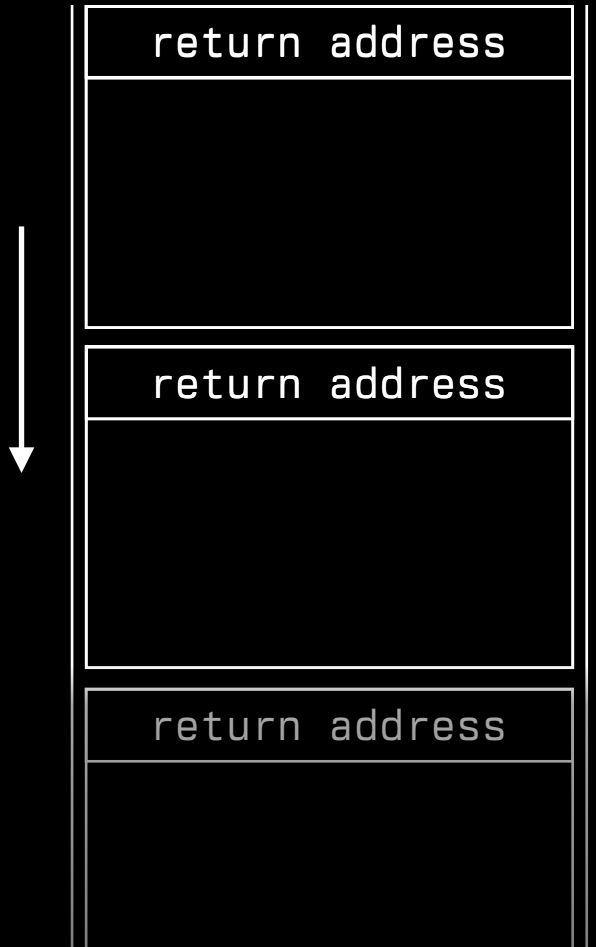
Phase I: Search




```
unsafe fn eh_personality(…) → UnwindReasonCode
```

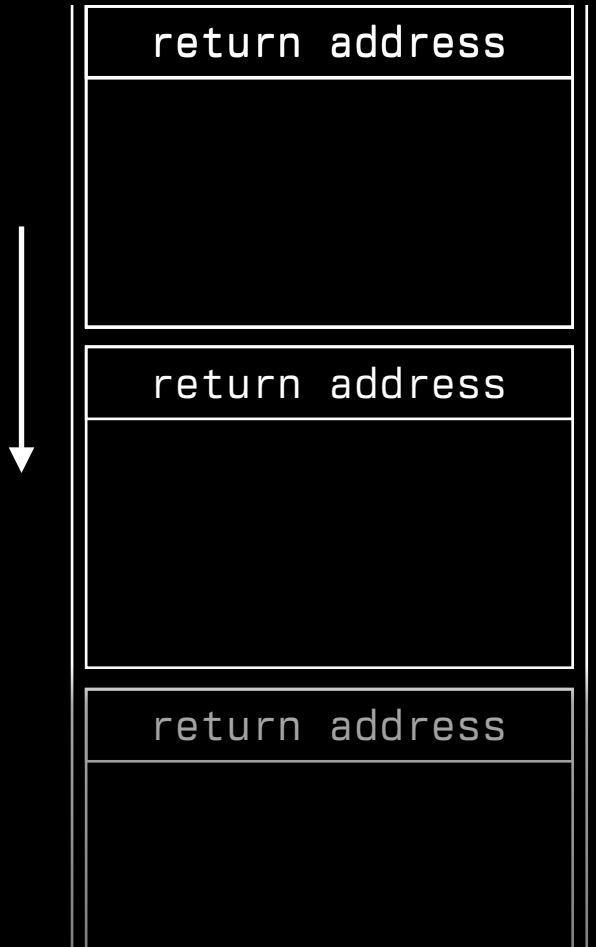
```
UnwindReasonCode :: HANDLER_FOUND
```

Phase II: Cleanup



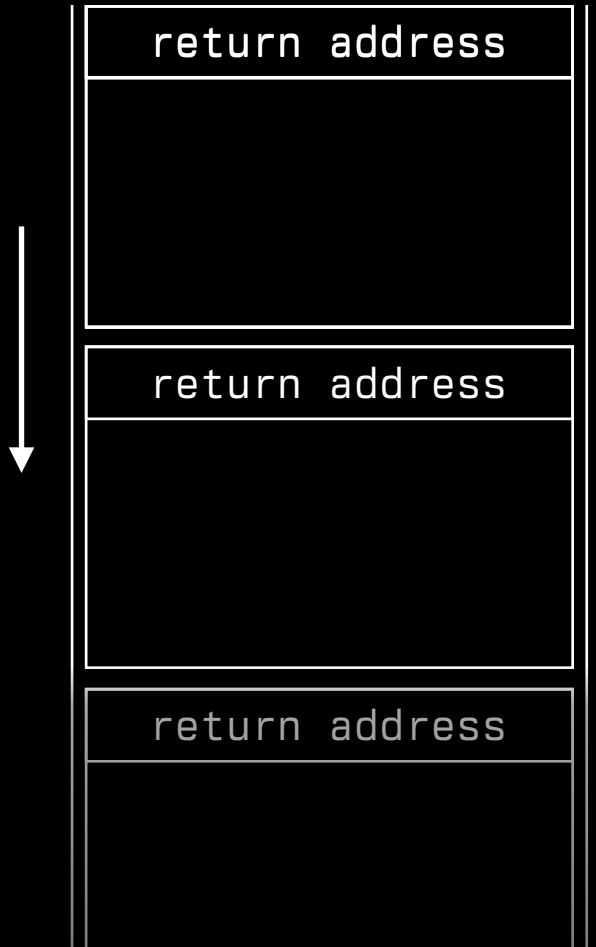

```
unsafe fn eh_personality(…) → UnwindReasonCode
```

Phase II: Cleanup




```
unsafe fn eh_personality(…) → UnwindReasonCode
```

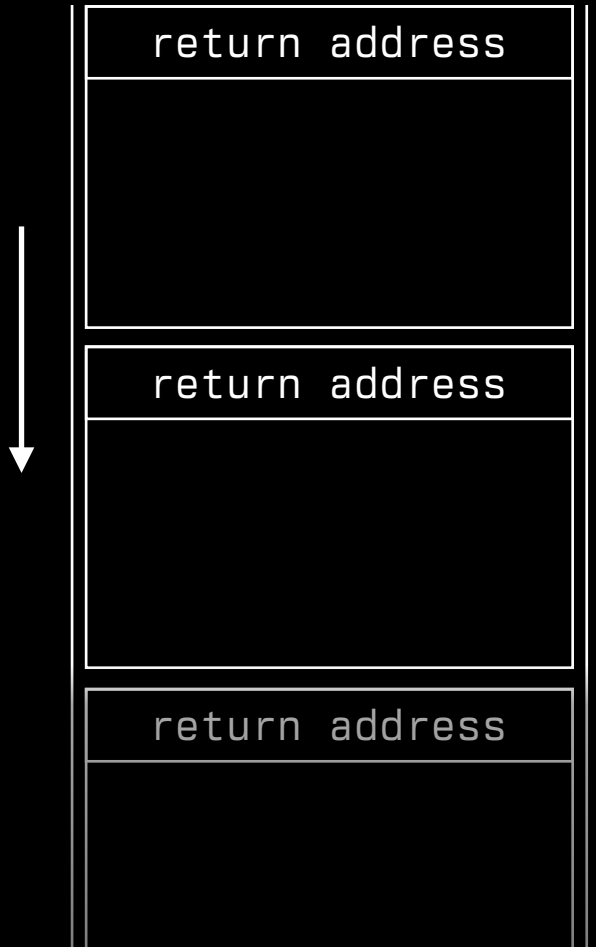
Phase II: Cleanup




```
unsafe fn eh_personality(…) → UnwindReasonCode
```

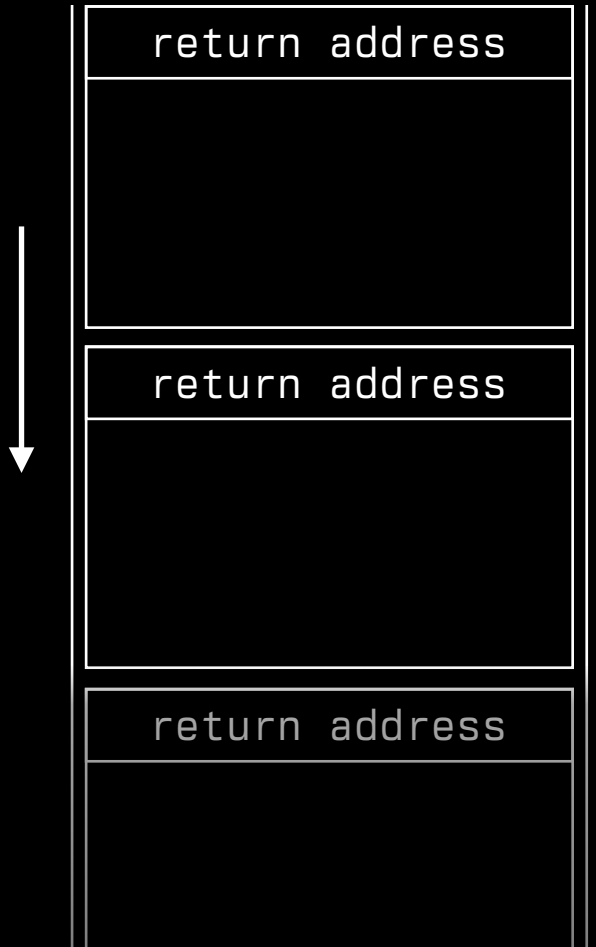
```
UnwindReasonCode :: INSTALL_CONTEXT
```

Phase II: Cleanup




```
unsafe fn eh_personality(…) → UnwindReasonCode
```

Phase II: Cleanup




```
unsafe fn eh_personality(…) → UnwindReasonCode
```

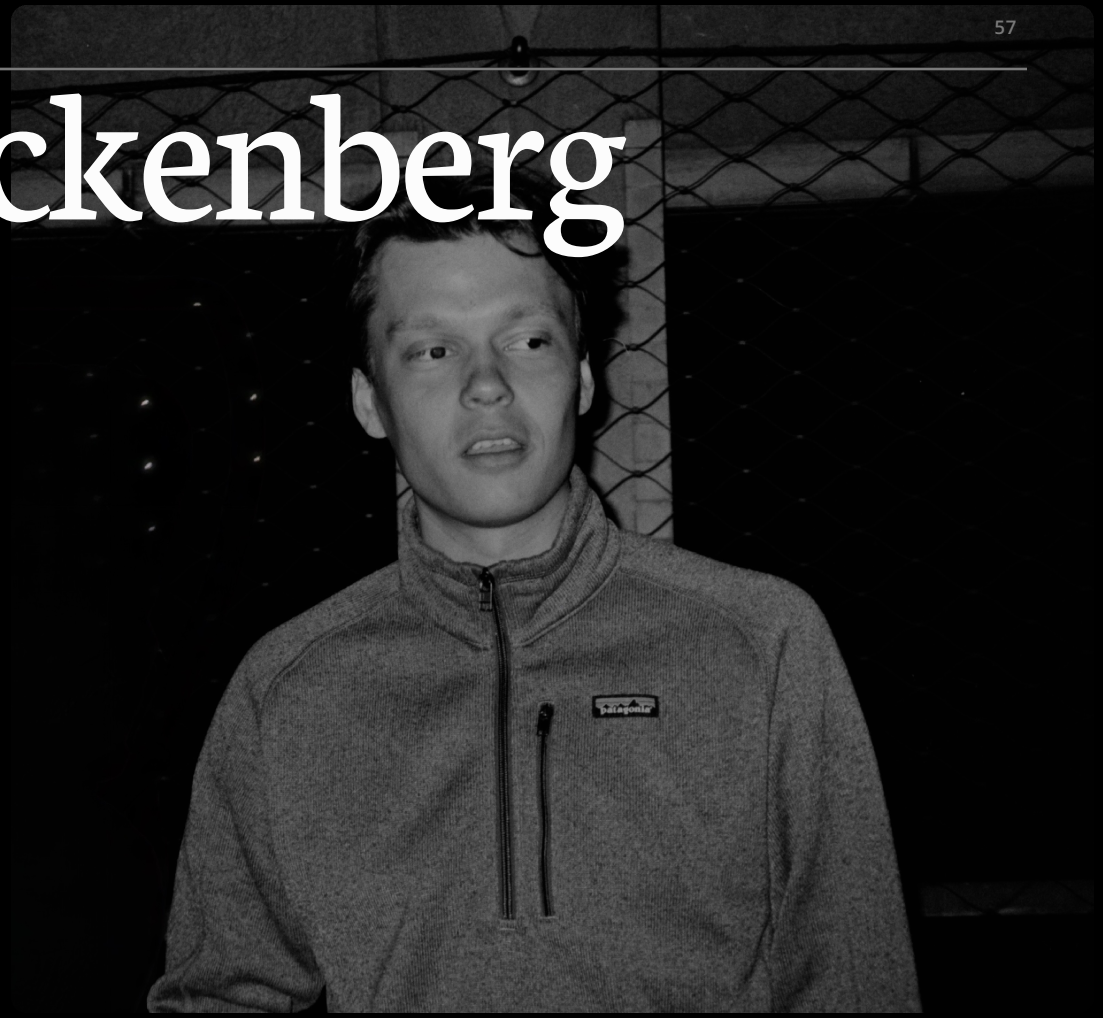
`UnwindReasonCode::HANDLER_FOUND`

This is *not* magic

don’t be afraid

Jonas Kruckenberg

- k23 OS
- Mainmatter
- TC39 Invited Expert



@jonaskruckenberg.de

JonasKruckenberg

@unsafe@webtoo.ls