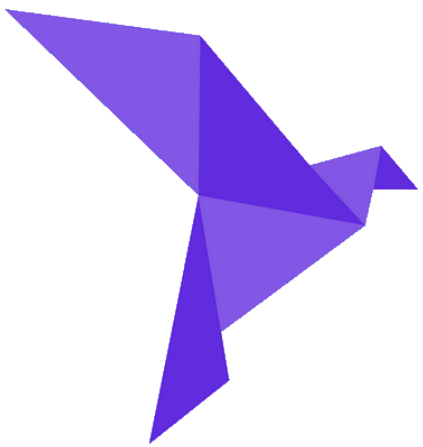
10X faster

taking charge of the compiler backend

Folkert de Vries

Roc

roc-lang.org

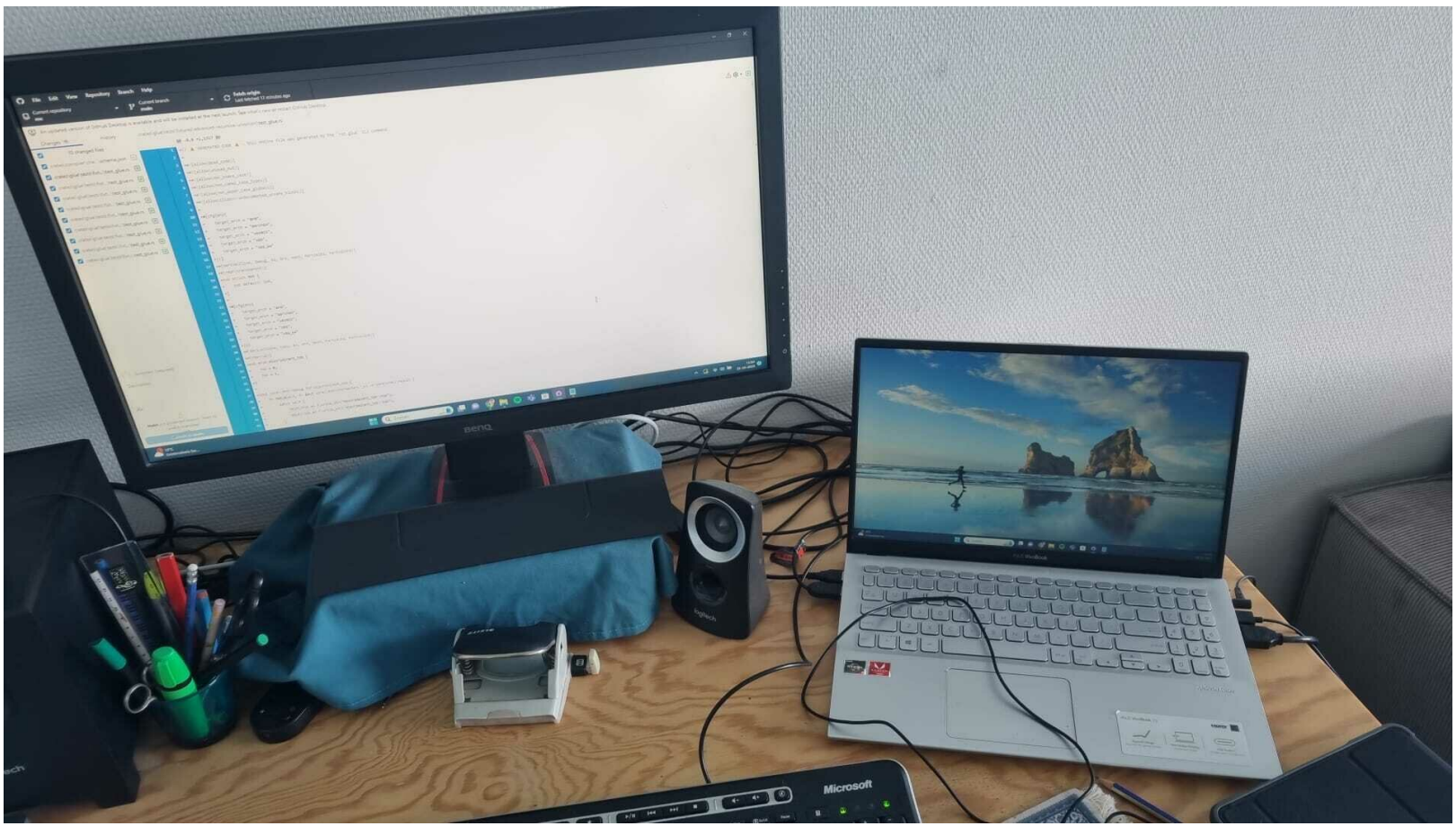


a fast, friendly, functional language

Compiler Feedback is Slow

an anecdote from the raspberry pi

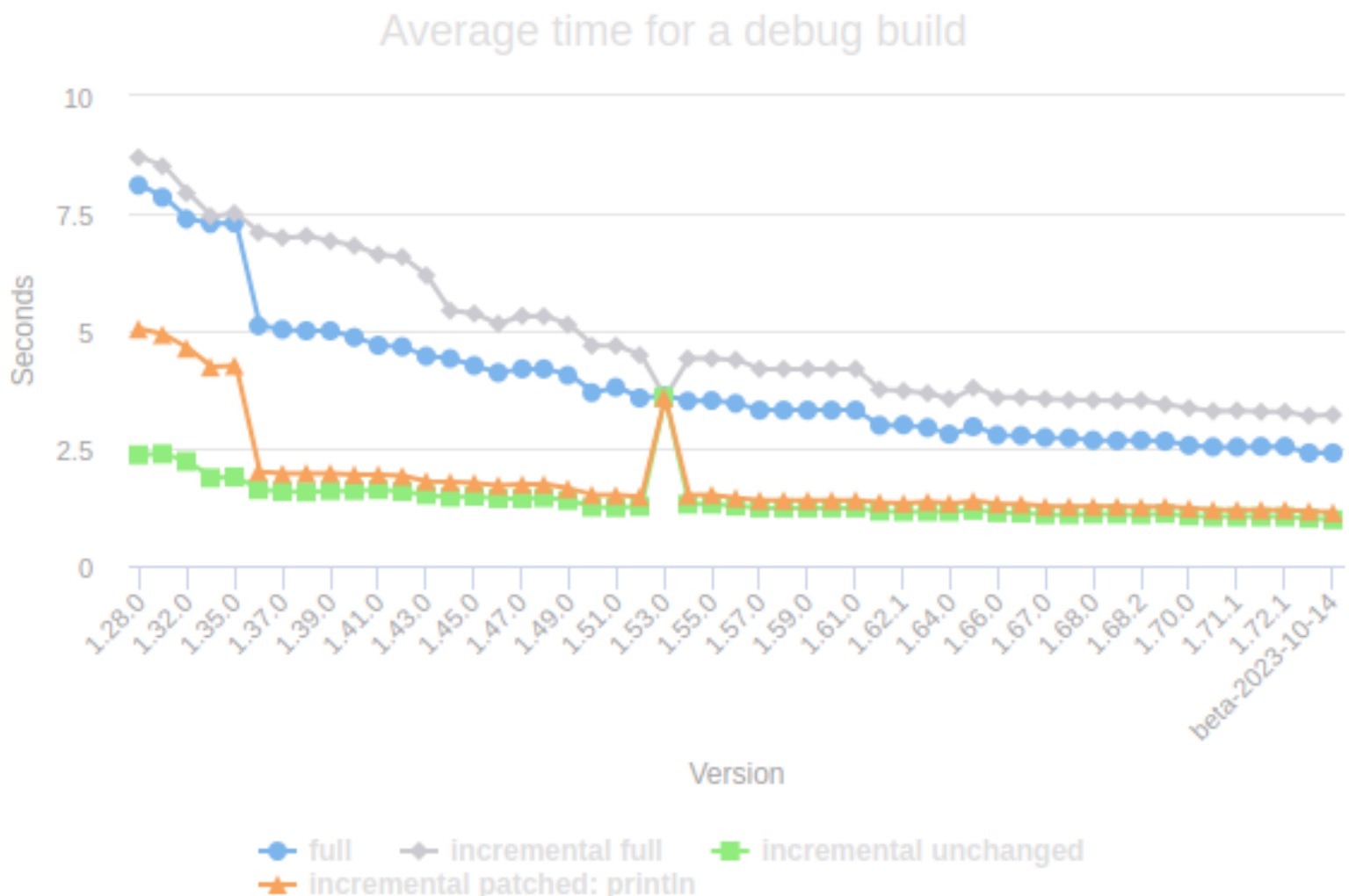
Compiler Feedback is Slow



2X sounds like a lot

but does not change my experience

"rust got faster, though"



"oh sweet summer child"

we don't accept that argumentation elsewhere

error messages package manager memory safety

The competition is Python

why is python popular?

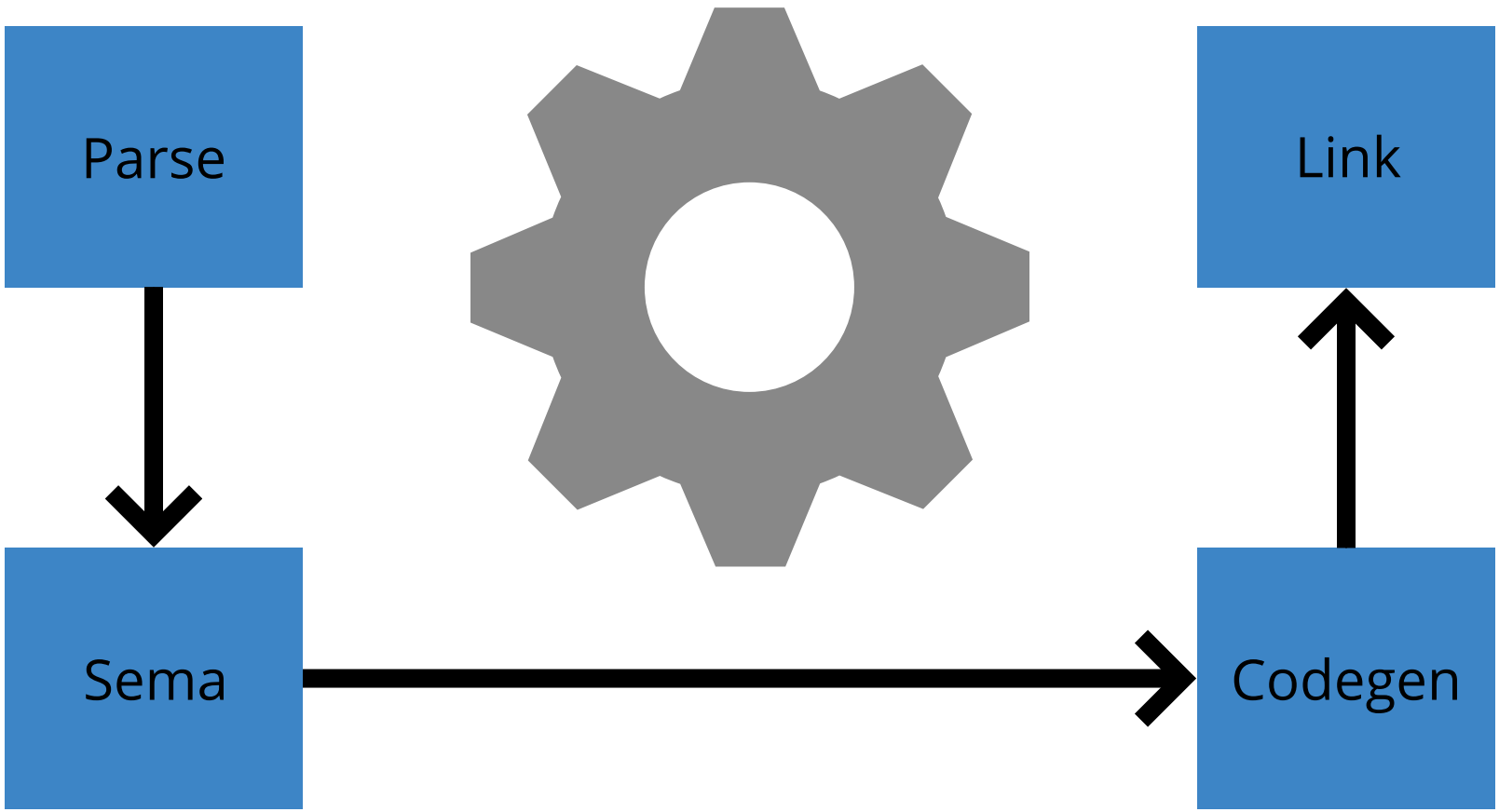
```
>>> a = float("nan")
>>> a == a
False
>>> [a] == [a]
True
>>> [float("nan")] == [float("nan")]
False
>>>
```

How can we do better?

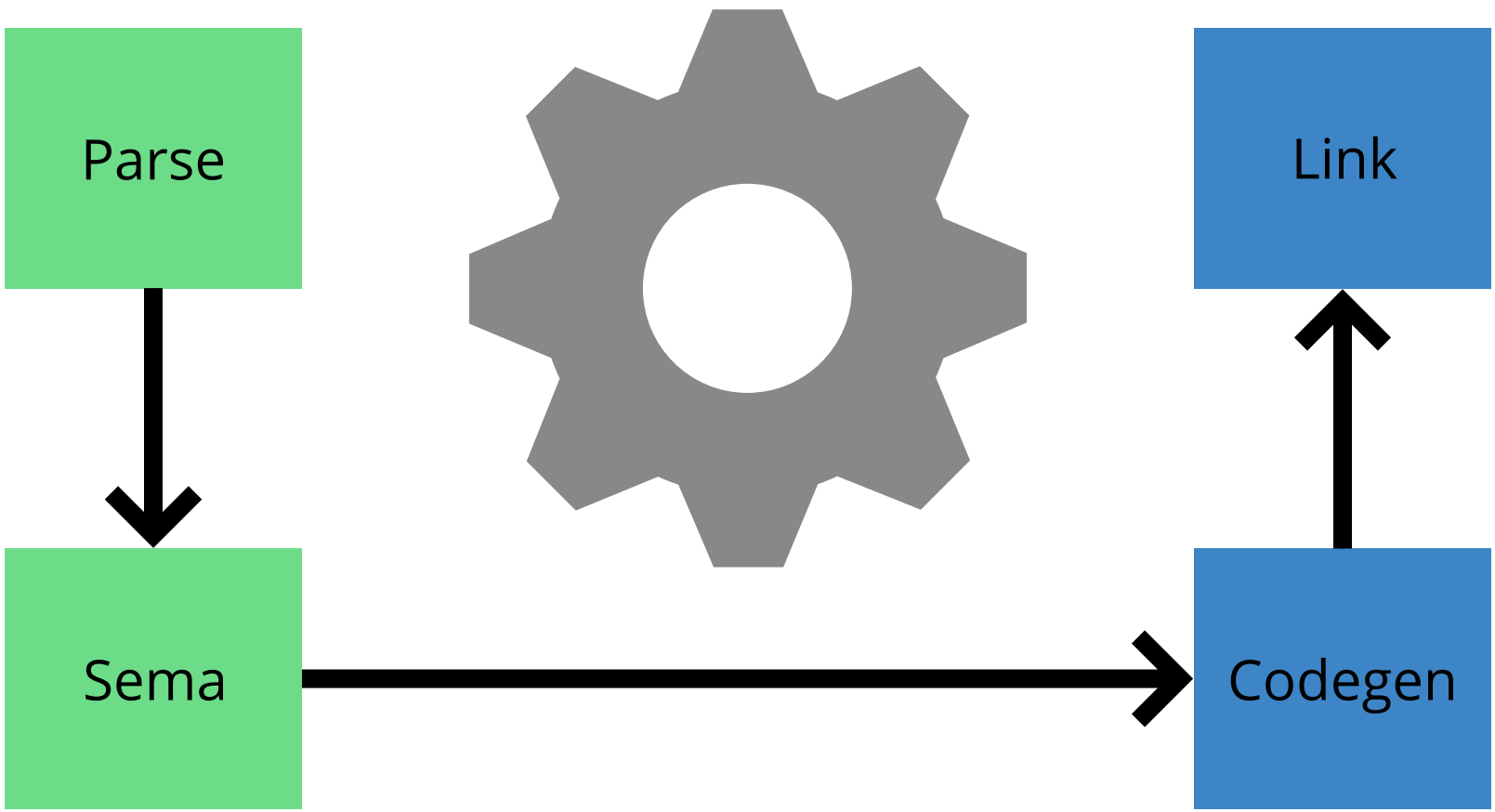
"make it work, then make it fast"
does not work

architecture determines performance

Compilers, how do they work



Engineering Incentives



LLVM

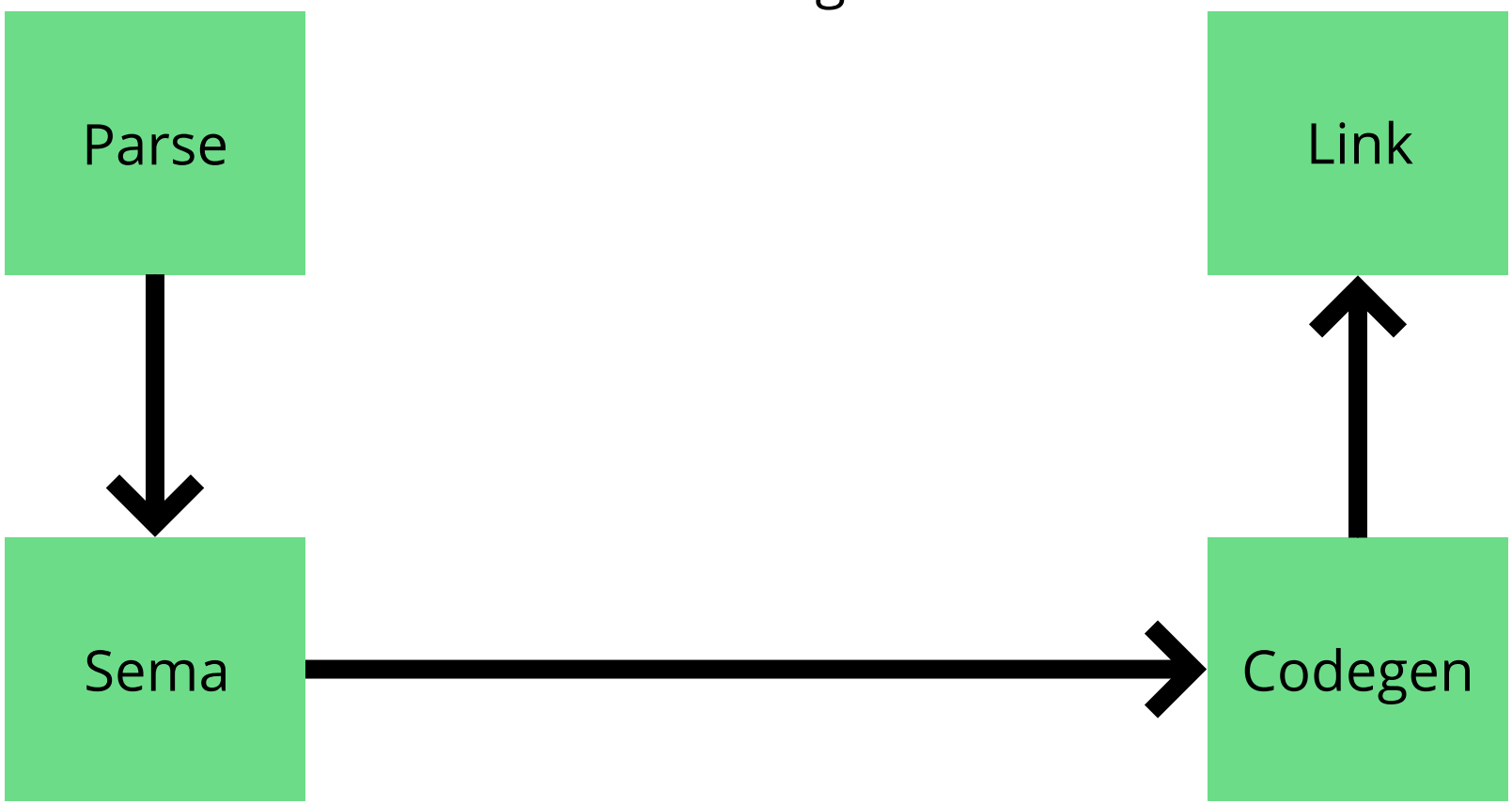
state of the art code generation,
on all the platforms

LLVM

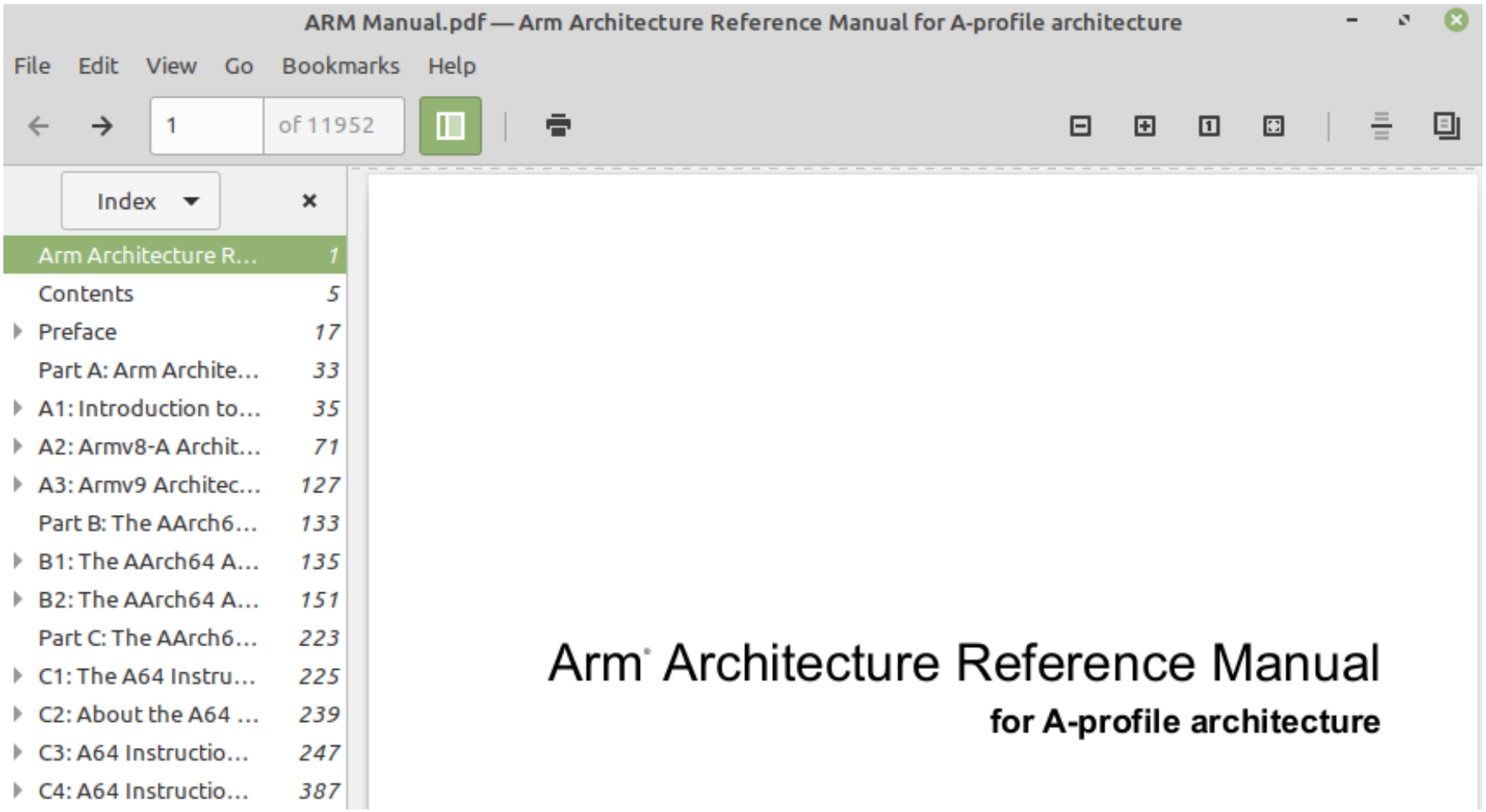
terrible at debug builds

Our Plan

vertical integration

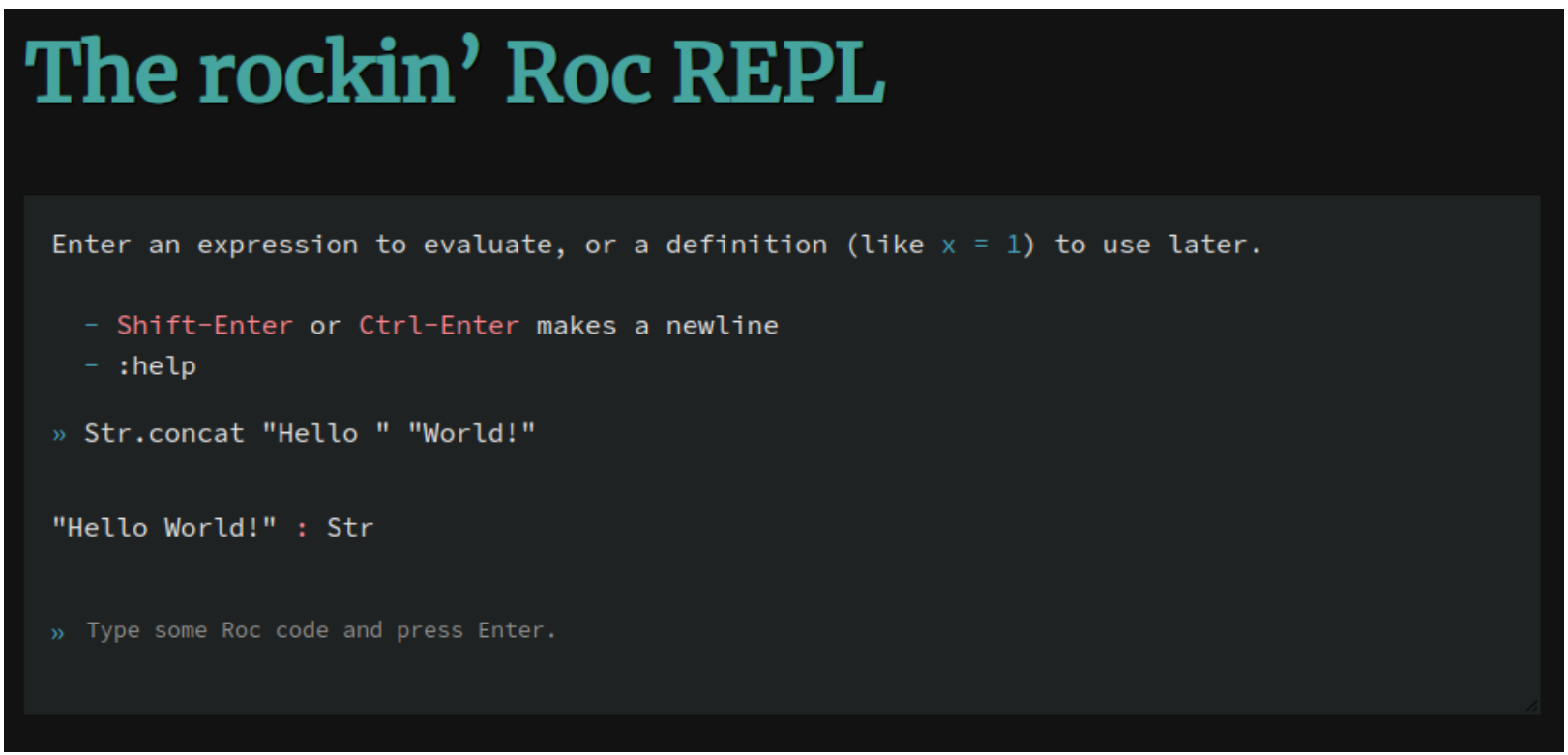


Our ... Challenges

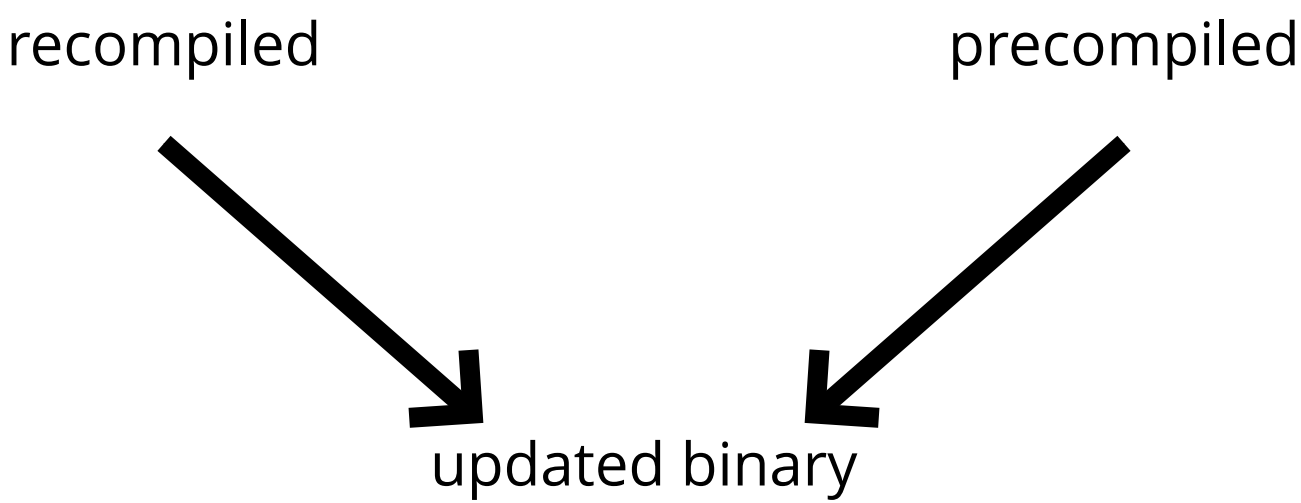


The competition is Python

roc-lang.org/repl



Surgical Linking



The competition is Python

We can compile + run hello world
faster than python can print it

Reflections on Rust

Rust deserves fast (re)compiles

Takeaways

be ambitious

(read 12000 page PDFs)

Takeaways

design for performance

orders of magnitude better

Takeaways

build cool things with friends

Expect more from your Tools

Thanks