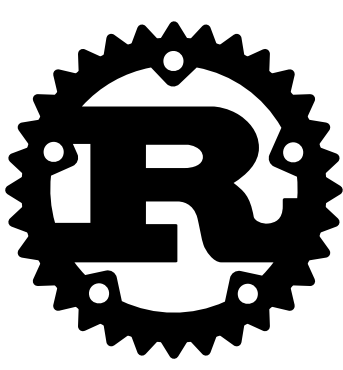


# Rust for HelenOS

Matěj Volf



## BACKGROUND



**Rust** is a modern programming language with a rich standard library and modern tooling. It has novel memory safety features, is suited for high performance code and is used for both systems and application programming. The Rust toolchain provides support for easy cross-compilation.



**HelenOS** is a multiserver microkernel operating system written in C. It draws significant inspiration from the Unix and POSIX standards, but many APIs are revised and implemented using a more modern approach. HelenOS was originally created as a student project at the Faculty of Mathematics and Physics of Charles University.

## THESIS GOALS

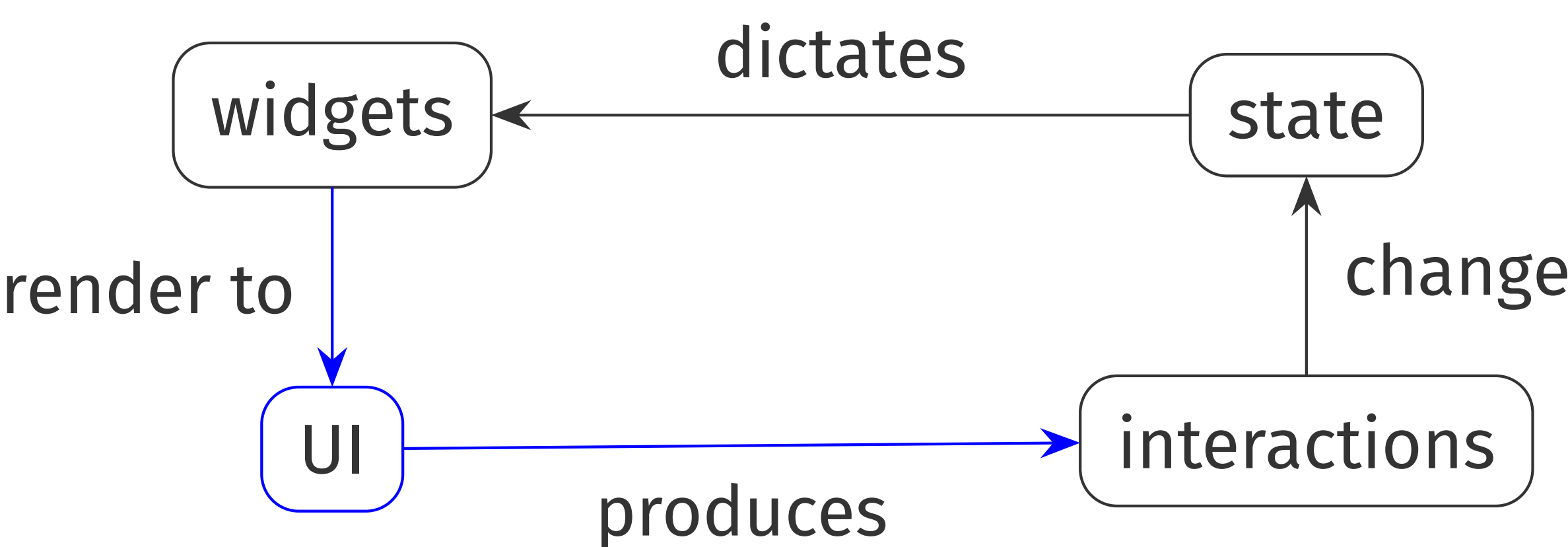
The high-level goal is to add support for running Rust programs on HelenOS. We divide this problem into the following key steps:

- Configure the Rust compiler toolchain to produce binaries compatible with HelenOS.
- Implement the necessary support in the Rust standard library using bindings to HelenOS C library.
- Port the whole toolchain to multiple CPU architectures supported by HelenOS.
- Add support for some paradigm of writing GUI programs.
- Demonstrate our toolchain by porting a few existing Rust applications.

## CHOICE OF UI LIBRARY



Although HelenOS has a native GUI library, writing a good Rust abstraction for the C interface proved very difficult. Instead, we selected the Iced library, which allows easy integration of a custom system runtime. Additionally, unlike many libraries, Iced does not require GPU support and provides a pure-Rust SVG-based renderer. The architecture of the Iced runtime is seen in the following diagram.

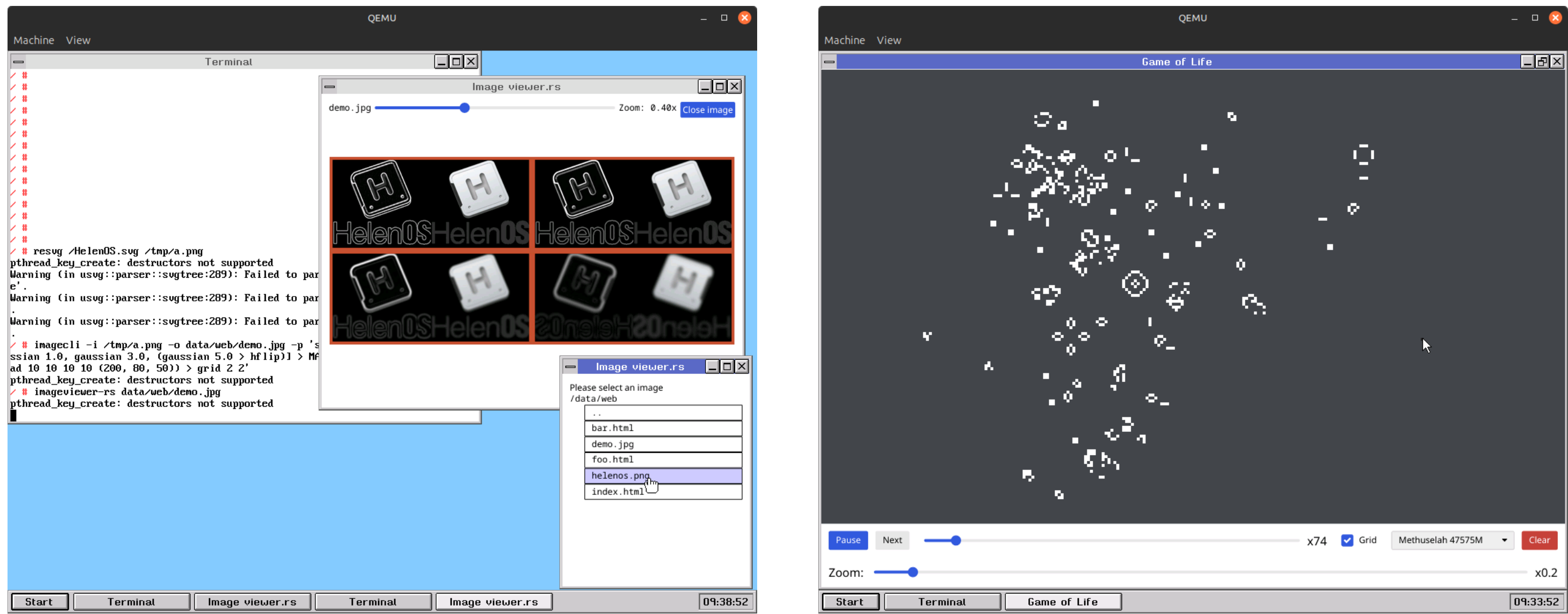


This thesis needed to implement the highlighted part of the runtime lifecycle. This involves opening a native HelenOS window, drawing the UI bitmap onto it and registering callbacks that listen to mouse and keyboard events.

## TLS SUPPORT IN HELENOS

A non-trivial challenge of this thesis was to resolve issues with the HelenOS implementation of **thread-local storage** (TLS). TLS is vital to internals of the Rust standard library runtime for threads and standard IO, and Rust’s usage of this feature triggered bugs that were not noticed by common C programs. A detailed description of the issues is available in the full text of the thesis, and our changes have been accepted into the upstream HelenOS repository.

## RESULT: APPLICATIONS RUNNING IN HELENOS



We have successfully compiled and run multiple existing real-world Rust applications on HelenOS. As can be seen from the screenshots above, we have also succeeded in writing our own image viewer with support for many image file formats, and we have also ported an implementation of Conway’s Game of Life that was present among the examples of the Iced library.

## PLATFORM SUPPORT AND FUTURE WORK

Architecture	Compatible	Tests, CLI tools	GUI apps
x86_64	✓	✓	✓
PowerPC	✓	✓	✓
IA-32	✓	✓	✗
SPARC	✓	✓	✗
ARM 64-bit	✓	✓	-
ARM 32-bit	✓	✗	-
MIPS, RISC-V	✗	HelenOS port does not run	
IA-64	✗	LLVM support removed	

As the support table demonstrates, our toolchain supports majority of the CPU architectures supported by HelenOS. More complete support is impeded by various unresolved bugs. It is unclear whether these originate in HelenOS or our toolchain. Details with descriptions of the specific issues for each platform are available in the full text of the thesis.

## FULL TEXT AND DEMO

The full text of the thesis with attachments, including a demo image of HelenOS with Rust programs preinstalled, can be found at <https://mvolfik.github.io/projects/helenos-rust/>