

Remember you write logic

@omarsheriffathy



Claims

Music

- *Any instrument can play almost any melody*
- *Each instrument has a sweet spot*
- *Performing as a group expands the effect of each instrument*
- *They all have the same music note*
- *Impossible without a maestro*
- *People only want to hear good music*

Software

- *Any programming language can almost implement any logic*
- *Each programming language has a sweet spot*
- *Using multiple programming languages in a single project gives better control over the implementation of the logic*
- *They all will run on the same machine*
- *Impossible without a compiler*
- *People want to have great tools*

***Melody is the building block of
the music***

***This allows different instruments to play together since they
have the same interface***

Logic must be the building block of a software

*If we can have a single interface to represent logic between
many languages and have this logic portable, then we can
reach more !*

***Is this possible in software
development ?***

Let's explore the different options

The libraries option !

Libraries

Dynamic

- *.dll, .so, .dylib*
- *Linked at compile time or at runtime*
- *The code included in the library is compiled separately and isn't included in the final executable, instead it is loaded on program startup*
- *Easy to modify the library and update the programs which are depending on it*
- *Allow disk space and memory optimizations*
- *It opens the door for security issues*
- *It opens the door for really cool stuff*

Static

- *.lib, .a*
- *Linked at compile time*
- *The code included in the library is compiled, linked and copied to the final executable*
- *Hard to modify the library and update the programs which are depending on it*
- *It's using more disk space and memory*
- *It's generally more secure*
- *It's just boring*

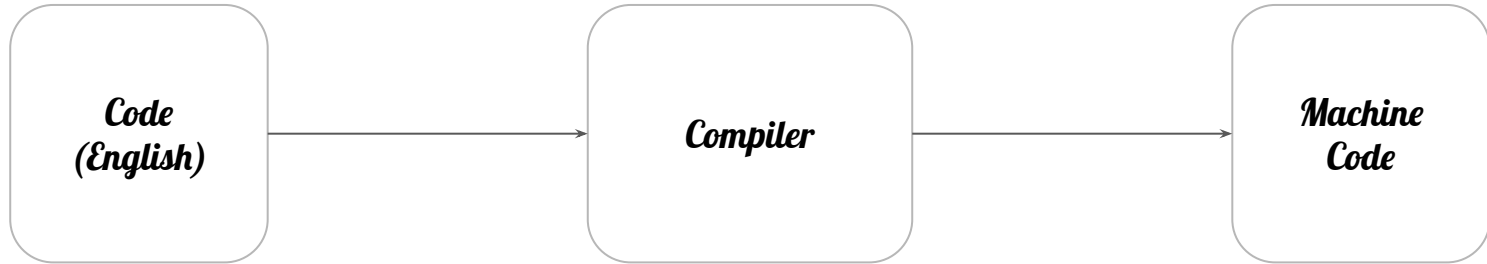
Demo

***Static library and shared library
(libraries_example)***

***We used libraries to prove we
have the concept of common
language or interface***

***What if we dive into the compiler toolchain which is actually
producing these libraries ? we should find another common
language, but on a lower level !***

What Computers understand



LLVM

Code ‘Mixer’ ?

LLVM

- *A collection of modular and reusable compiler toolchain technologies*
- *Started as a research project at the university of illinois*
- *Now is an open source project*
- *Started by Chris Lattner, Apple hired him to bring LLVM to production quality and be used in apple products*

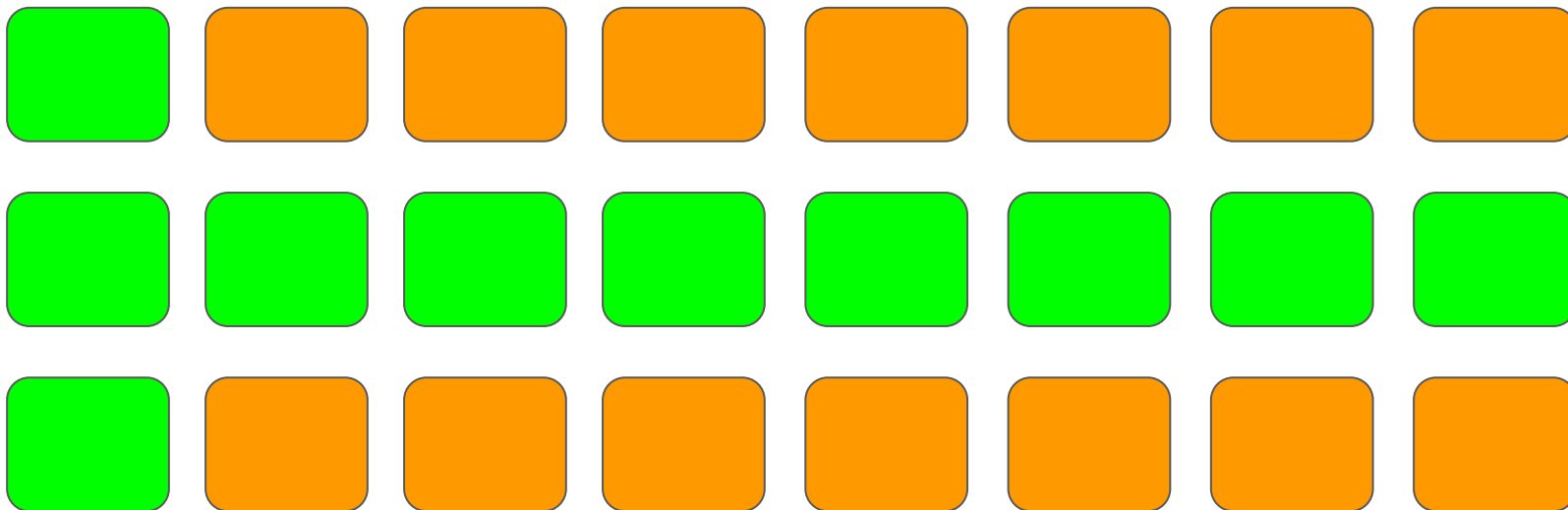
LLVM Design



Alignment

Know the hardware you run your software on

Alignment Explained



LLVM Design

Frontends

- *C, C++*
- *Swift*
- *Python*
- *Rust*
- *Scala*
- *Fortran*
- *Go*
- *CUDA*
- *GLSL*
- *...*

Intermediate

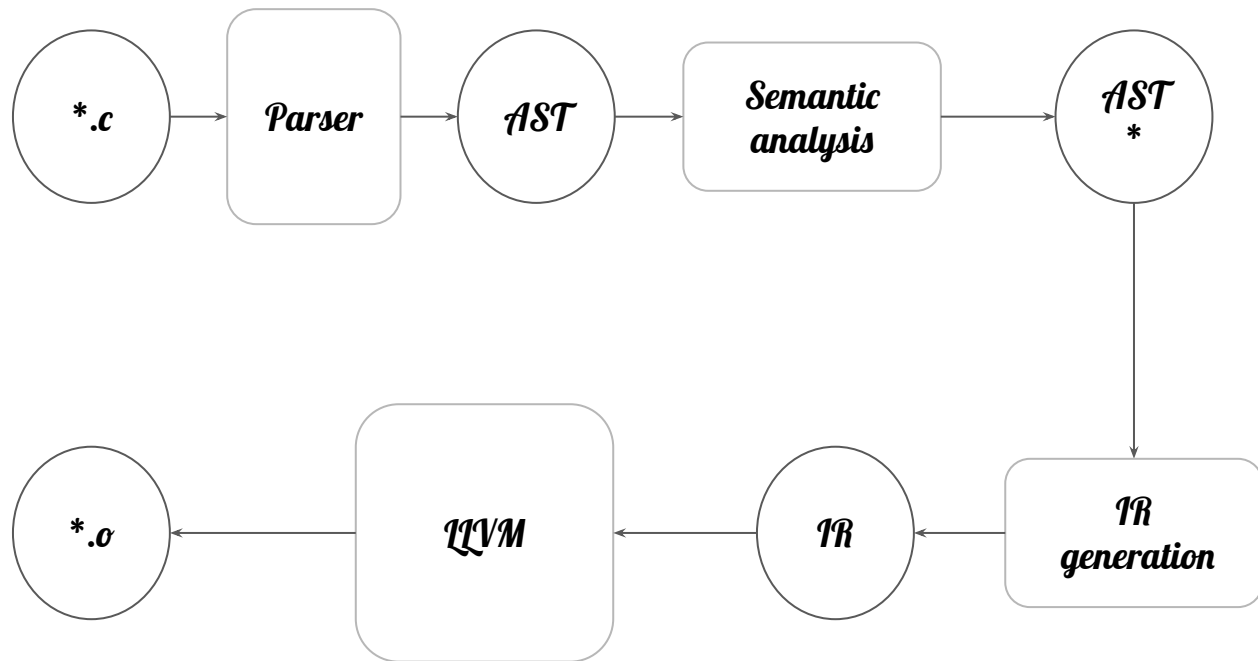
- *LLVM Intermediate representation (IR)*

Backends

- *x86, x86-64*
- *ARM*
- *MIPS*
- *PowerPC*
- *...*

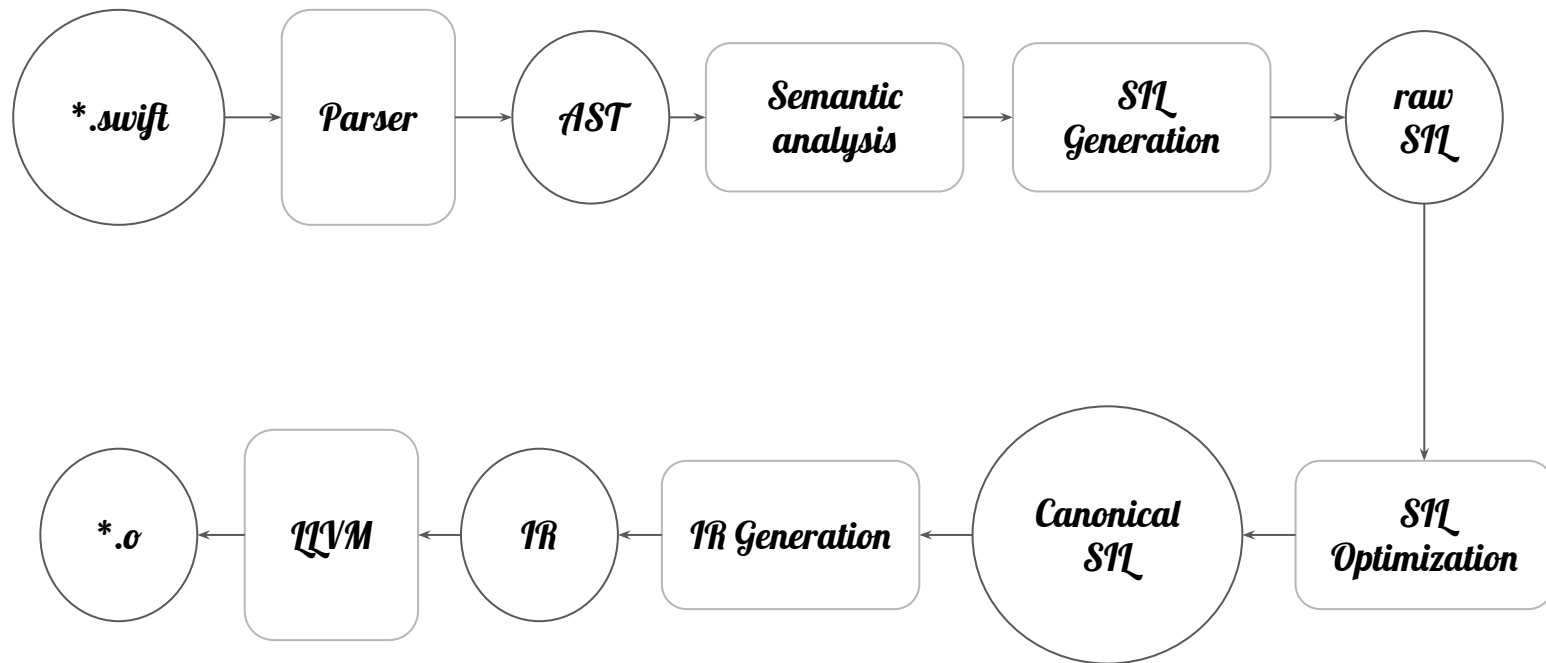
Clang

Clang



Swift

Swift

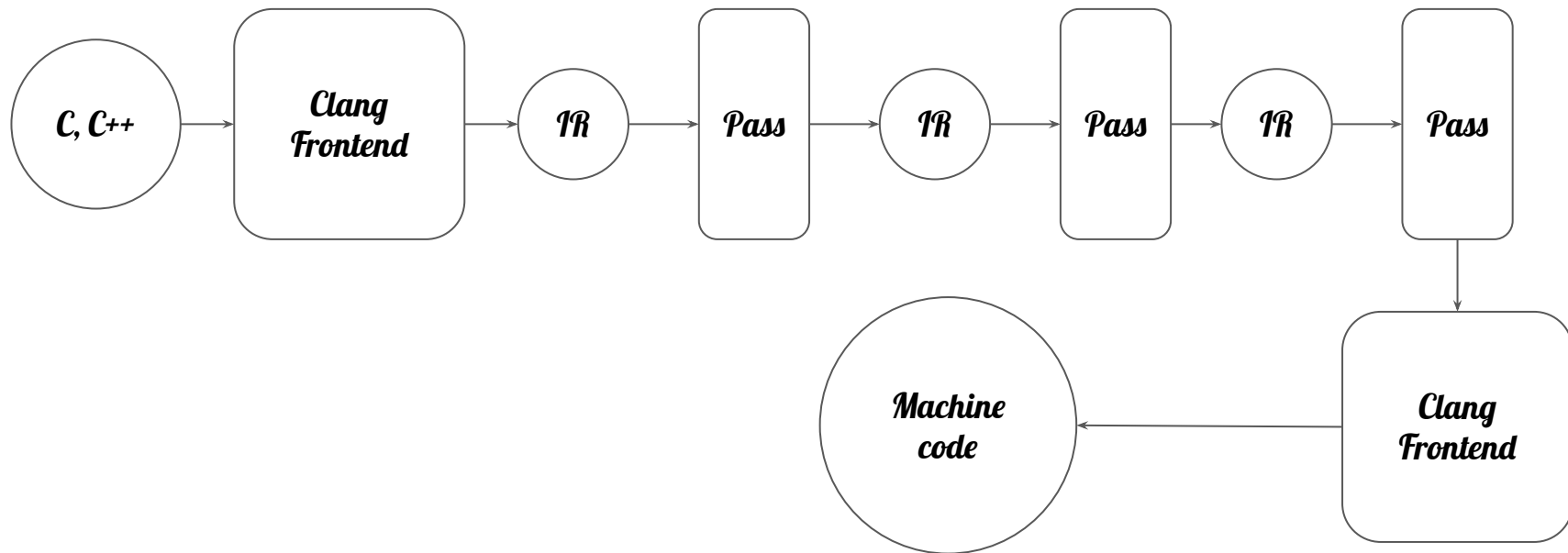


IR Demo

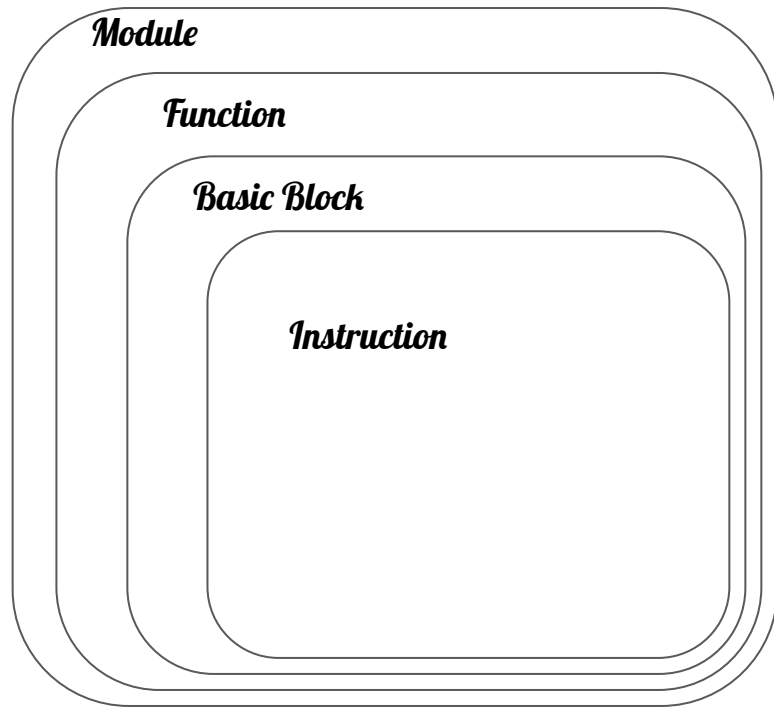
Logic distributed on different programming languages

LLVM, let's open that box

LLVM (Clang) Inner Pieces



LLVM Passes

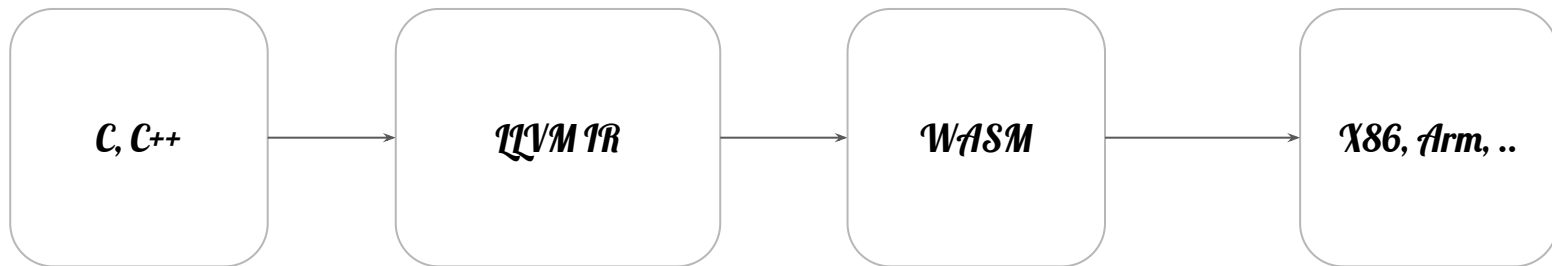


Demo

LLVM Transform Function Pass

LLVM, Emscripten

Emscripten (WASM)



Demo

Wasm

Thank you, Questions ?

