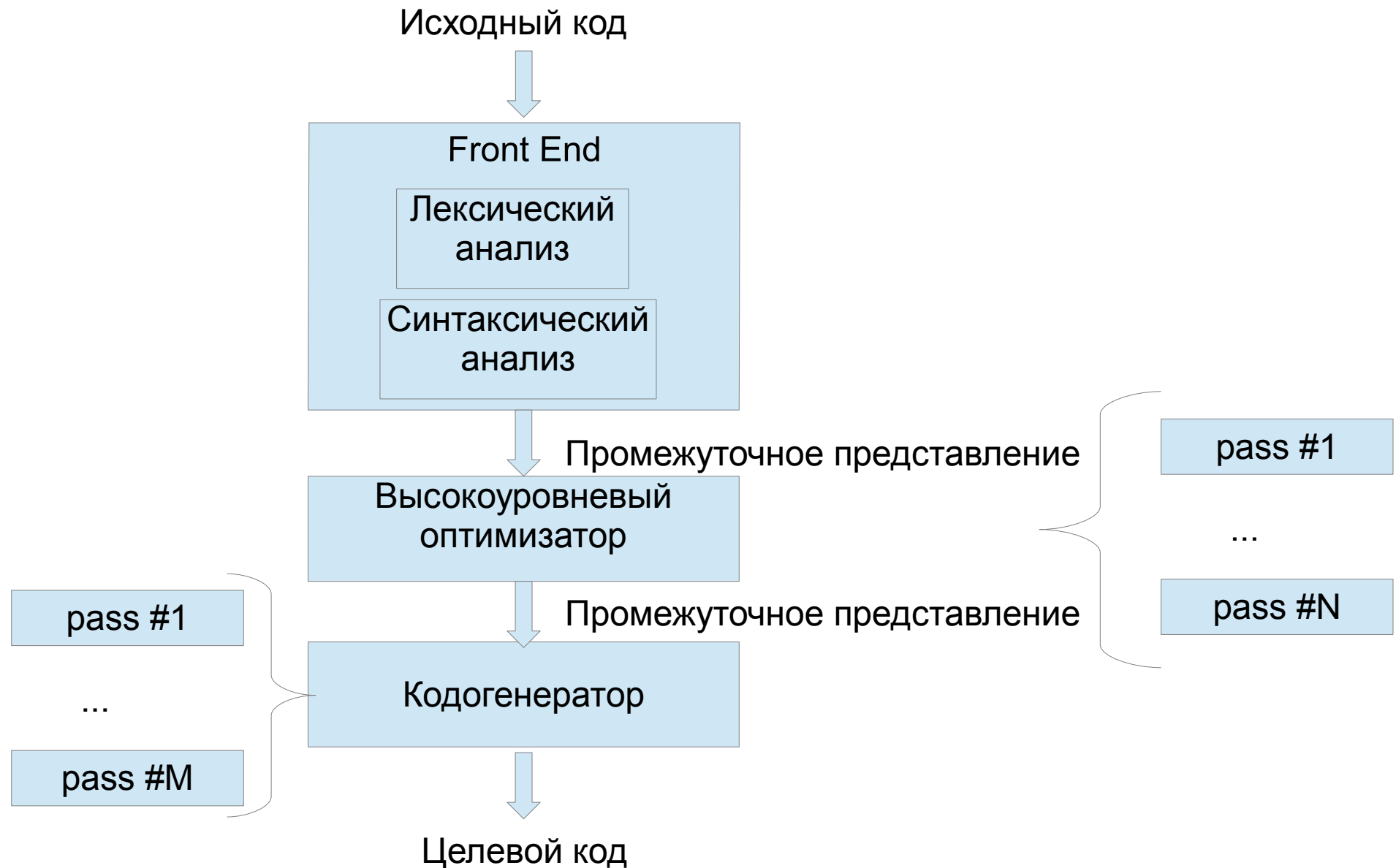


Open source compilers
(примеры для изучения)

GCC (= GNU Compiler Collection)

- C/C++, Objective-C, Fortran, Go, Java, Ada
- x86 (32/64 bit), ARM и т.д.
- статический компилятор
- написан на C
- исходный код довольно сложен для понимания
- хорошие оптимизирующие возможности

GCC (= GNU Compiler Collection)



LLVM (= Low Level Virtual Machine)

- Framework (набор кирпичиков) для написания компиляторов
- x86 (32/64 bit), ARM и т.д.
- статическая компиляция/JIT-компиляция
- написан на C++
- исходный код легче для понимания, чем GCC
- есть примеры простых компиляторов
- возможности по оптимизации уступают GCC, но постоянно развиваются
- рекомендуется для изучения

LLVM (= Low Level Virtual Machine)

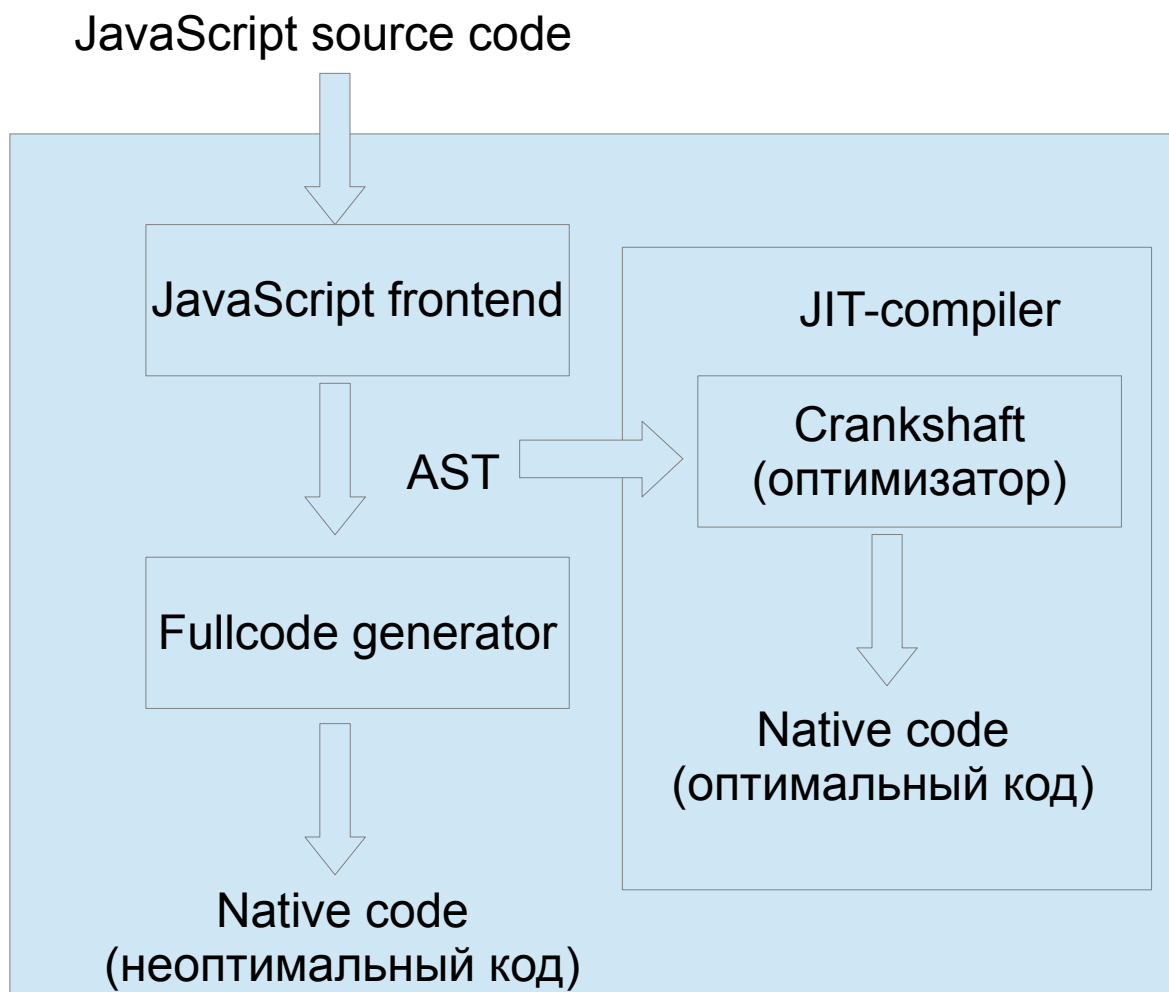
Где используется

- clang (C/C++/Objective-C) — статический компилятор
- JSC (= JavaScriptCore). LLVM используется как JIT-компилятор
- Pyston (= интерпретатор «движок» Python). LLVM используется как JIT-компилятор
- Mono (.net open source реализация). LLVM используется как JIT-компилятор
- CoreCLR. Попытки использования LLVM как JIT

V8 JavaScript engine

- интерпретатор («движок») JavaScript
- генерируется машинный код
- JIT-компилятор (LLVM не используется)
- для понимания исходного кода требуется понимание ECMA 262 (JavaScript)
- встроенный оптимизатор

V8 JavaScript engine



Двухъярусная архитектура

- Fullcode generator (генерирует неоптимальный код, делает это очень быстро)
- Crankshaft оптимизатор

Оптимизируются

- Циклы
- Функции

V8 JavaScript engine

Фрагмент на JavaScript

```
function Fun(a, b)
{
    // это может, например,
    //   * две строки (конкатенация строк)
    //   * два числа (арифметическое сложение)
    return a + b;
}
```

Fullcode generator

Генерируется код для обоих случаев

```
if (a is string && b is string)
{
    do string concat
}
else if (a is number && b is number)
{
    do a + b
}
```

Crankshaft

Например, код может быть такой
(если оптимизатор выяснил, что
всегда имеет место второй
случай)

do a + b

JSC (= Java Script Core)

- интерпретатор («движок») JavaScript
- четыре яруса, а не два (как в V8)
- для понимания исходного кода требуется понимание ECMA 262 (JavaScript)
- встроенный оптимизатор

JSC (= Java Script Core)

