

LLVM

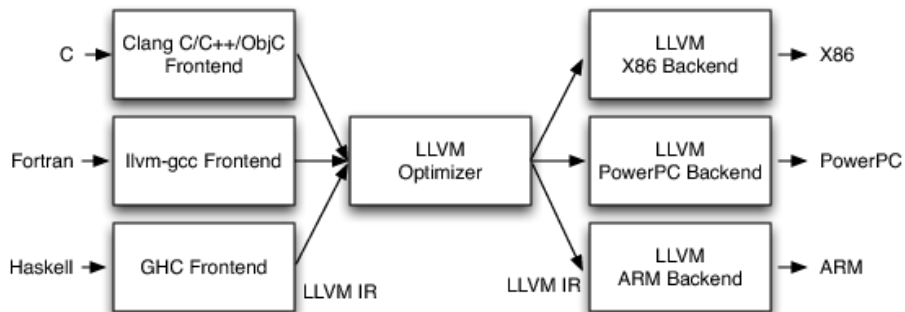
Иван Викторович Михайлов

ИТМО, КТ

imihajlow@gmail.com

25.03.2015

Low level virtual machine.



LLVM IR

LLVM IR (intermediate representation) – типизированный трёхадресный код в SSA-форме.

SSA – static single assignment.

SSA-форма – каждой переменной значение присваивается только единожды.

- Целые:
 - i1
 - i8
 - i32
 - i64
 - и так далее.
- С плавающей точкой:
 - double
 - float
 - x86_fp80
- void

- Указатель – тип^*
- Массив – $[\text{число элементов} \times \text{тип}]$
- Структура – $\{ \text{тип}, \text{тип}, \dots \}$
- Вектор (SIMD) – $\langle \text{число элементов} \times \text{тип} \rangle$

Переменные (регистры)

- %имя – локальная;
- @имя – глобальная.

ret

`ret тип значения`

Возврат из функции.

br

`br i1 условие, label iftrue, label iffalse`

Условный переход.

`%x = add i32 %a, %b`

`%vr = fadd <4 x float >%v1, %v2`

- `add`, `fadd`, `sub`, `fsub`;
- `mul`, `fmul`;
- `udiv` – целочисленное беззнаковое деление;
- `idiv` – целочисленное знаковое деление;
- `fdiv` – деление с плавающей точкой;
- `urem`, `srem`, `frem` – остаток от деления;
- `shl`, `lshr`, `ashr` – сдвиги;
- `and`, `or`, `xor`.

Инструкции сравнения

icmp, fcmp → i1

result = icmp cond type op1, op2

cond:

- eq, ne;
- ugt, sgt;
- uge, sge;
- ult, slt;
- ule, sle;

- extractelement
- insertelement
- shufflevector

Доступ к элементам

- extractvalue
- insertvalue

- `alloca` – выделение памяти на стеке;
- `load` – чтение из памяти;
- `store` – запись в память;
- `fence` – барьер;
- `cmpxchg`;
- `atomicrw` – префикс атомарной инструкции;
- `getelementptr` – адрес элемента структуры или массива.

Базовый блок: от метки до инструкции передачи управления (ret, br и других).

CFG (control flow graph).


```
result = phi type [ var1, label1 ], ...
```

```
LoopHeader:
```

```
    br label %Loop
```

```
Loop:      ; Infinite loop that counts from 0 on up...
```

```
    %indvar = phi i32 [ 0, %LoopHeader ], [ %nextindvar, %Loop ]
```

```
    %nextindvar = add i32 %indvar, 1
```

```
    br label %Loop
```

```
define float @max(float %x, float %y)
{
    %cond = fcmp ogt float %x, %y
    br i1 %cond, label %IfTrue, label %IfFalse
IfTrue:
    ret float %x
IfFalse:
    ret float %y
}
```

```
sudo apt-get install llvm clang
```

```
clang -S -emit-llvm sample.c -o sample.ll
```

```
llc -O3 sample.ll -march=x86-64 -o sample-x86-64.s
```

```
llc -O3 sample.ll -march=arm -o sample-arm.s
```

Что почитать

- <http://habrahabr.ru/post/47878/>
- <http://llvm.org/docs/LangRef.html>

Конец.