

LLVM tutorial

How to write a simple LLVM pass?

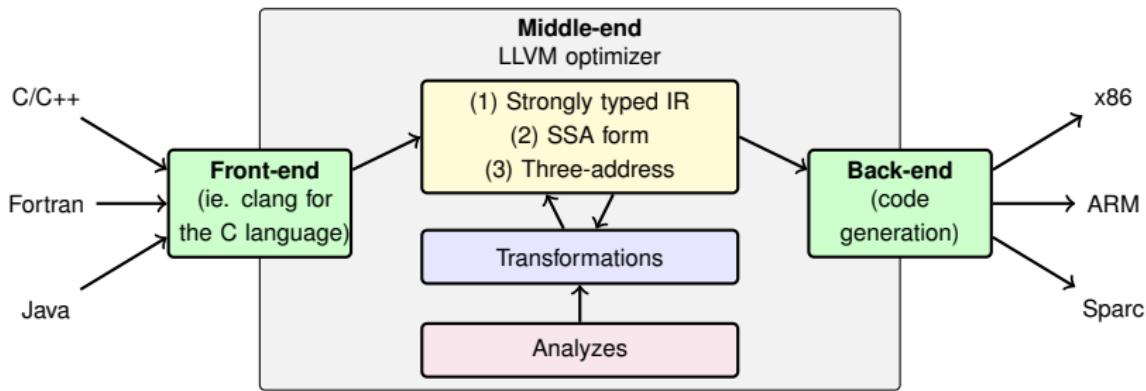
Guillaume Revy

Univ Perpignan Via Domitia, DALI, Perpignan, France
LIRMM, Univ Montpellier, CNRS (UMR 5506), Montpellier, France



LLVM infrastructure

- LLVM = compiler **infrastructure** and **framework**



- LLVM optimizer = **series of "passes"**
 - ▶ analysis and optimization passes, run one by one
- LLVM intermediate form = **Virtual Instruction Set**
 - ▶ language- and target-independent form = same passes for all languages and targets

LLVM IR at a glance

```
ModuleID = 'test.c'
; ...

@.str.1 = private unnamed_addr constant [19 x i8] c"... i est negatif\0A\00",
    align 1
; ...

; Function Attrs: noinline nounwind optnone ssp uwtable
define i32 @main() #0 {
    %1 = alloca i32, align 4
    %2 = alloca i32, align 4
    store i32 0, i32* %1, align 4
    %3 = call i32 (i8*, ...) @scanf(i8* getelementptr inbounds ([3 x i8], [3 x i8]* @.str, i32 0, i32 0), i32* %2)
    %4 = load i32, i32* %2, align 4
    %5 = icmp slt i32 %4, 0
    br i1 %5, label %6, label %8

; <label>:6:                                     ; preds = %0
%7 = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([19 x i8], [19 x i8]* @.str.1, i32 0, i32 0))
br label %10

; <label>:8:                                     ; preds = %0
%9 = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([26 x i8], [26 x i8]* @.str.2, i32 0, i32 0))
br label %10

; <label>:10:                                    ; preds = %8, %6
ret i32 0
}
```

LLVM tutorial

- Different kinds of passes are available: module pass, function pass, loop pass, ...
 - ▶ it executes on each object (module, function, loop) appearing in the input program, independently of all others
- Objective is to write function passes :
 1. to walk through the LLVM IR,
 2. to insert and to remove instructions,
 3. and to insert call to external functions.

<https://gite.lirmm.fr/grevy/llvm-tutorial/>