

Statická analýza parametrů modulů jádra Linuxu

Nikolas Patrik - xpatri00

Fakulta informačních technologií - VUT

2019

- ▶ Čo je LVVM?
 - ▶ LLVM-IR
 - ▶ Priechody
- ▶ LVVM priechod na získanie východzích hodnôt globálnych premenných
 - ▶ Preklad mimo zdrojových súborov LLVM
 - ▶ Trieda Constant

Čo je LLVM?

LLVM(Low level virtual machine) je infraštruktúra pre prekladač ktorá umožňuje optimalizácie kódu počas prekladu, linkovania alebo počas behu programu pre rôzne programovacie jazyky.

LLVM skladá z nasledujúcich komponent:

- ▶ Front-end

Čo je LLVM?

LLVM(Low level virtual machine) je infraštruktúra pre prekladač ktorá umožňuje optimalizácie kódu počas prekladu, linkovania alebo počas behu programu pre rôzne programovacie jazyky.

LLVM skladá z nasledujúcich komponent:

- ▶ Front-end
- ▶ LLVM-IR

Čo je LLVM?

LLVM(Low level virtual machine) je infraštruktúra pre prekladač ktorá umožňuje optimalizácie kódu počas prekladu, linkovania alebo počas behu programu pre rôzne programovacie jazyky.

LLVM skladá z nasledujúcich komponent:

- ▶ Front-end
- ▶ LLVM-IR
- ▶ Priechody(anglicky passes)

Čo je LLVM?

LLVM(Low level virtual machine) je infraštruktúra pre prekladač ktorá umožňuje optimalizácie kódu počas prekladu, linkovania alebo počas behu programu pre rôzne programovacie jazyky.

LLVM skladá z nasledujúcich komponent:

- ▶ Front-end
- ▶ LLVM-IR
- ▶ Priechody(anglicky passes)
- ▶ Back-end

LLVM-IR(celý názov LLVM intermediate representation) je názov pre nízkoúrovňový jazyk podobný assembleru. Vlastnosti tohoto jazyka:

- ▶ RISC

LLVM-IR(celý názov LLVM intermediate representation) je názov pre nízkoúrovňový jazyk podobný assembleru. Vlastnosti tohoto jazyka:

- ▶ RISC
- ▶ silne-typovaný

LLVM-IR(celý názov LLVM intermediate representation) je názov pre nízkoúrovňový jazyk podobný assembleru. Vlastnosti tohoto jazyka:

- ▶ RISC
- ▶ silne-typovaný
- ▶ nemá konečný počet registrov

LLVM-IR(celý názov LLVM intermediate representation) je názov pre nízkoúrovňový jazyk podobný assembleru. Vlastnosti tohoto jazyka:

- ▶ RISC
- ▶ silne-typovaný
- ▶ nemá konečný počet registrov

LLVM poskytuje tri izometrické (funkcionálne rovnaké) formy IR: pseudo-assembler formát, C++ formát a bitkód.

Priechody

Priechody sú programi ktoré transformujú alebo analyzujú kód v internej reprezentácii. Podľa toho ako chceme aby náš priechod pracoval využívame jednu z nasledujúcich tried:

ModulePass, CallGraphSCCPass, FunctionPass, LoopPass, RegionPass alebo BasicBlockPass

LVVM priechod na získanie východzích hodnôt globálnych premenných

Trieda ModulePass – hľadáme globálne premenné v celku, nepotrebujeme konkretizovať oblasti kódu ako u ostatných tried.

Trieda Module – poskytuje metódy na prácu s globálnymi premennými.

Pozn. LLVM registruje ako globálne premenné aj tie ktoré nie sú inicializované. Potreba ošetriť.

Preklad mimo zdrojových súborov LLVM

LLVM využíva na preklad svojich zdrojových súborov CMake. CMake je nástroj na automatizáciu prekladu a jeho hlavnou prednosťou je jednoduchosť. Na použitie LLVM mimo zdrojových súborov využijeme nasledujúcu direktívu v CMake:

```
find_package(LLVM REQUIRED CONFIG)
```

Trieda Constant

Trieda Constant je obalový typ ktorý uchováva v sebe hodnotu globálnych premenných. Pre naše potreby túto hodnotu musíme prekonvertovať do človeku čitateľnej formy. Toho dosiahneme pretypovaním na jednu z nasledujúcich podtried:

- ▶ ConstantInt
- ▶ ConstantFP
- ▶ ConstantArray
- ▶ ConstantDataArray
- ▶ ConstantStruct
- ▶ ConstantAggregateZero
- ▶ ConstantPointerNull
- ▶ ConstantExpr

Koniec