Úvod

Tento text popisuje instalaci a použití nástrojů pro kontrolu začátečnických chyb v operačním systému GNU/Linux. Uvedené postupy by měly fungovat i v jiných Unix-like systémech. Pro systém Windows neznám žádný obecný a jednoduchý postup instalace ani použití, tak mi jeho uživatelé jistě prominou.

Instalace

Systém LLVM spolu s novými nástroji můžete nainstalovat z několika zdrojů. Prvním zdrojem je archiv `src.zip` na tomto CD. Druhou možností je získání aktuální verze archivu na adrese `http://kam.mff.cuni.cz/~doucm6am/acc/src.zip`. Poslední možností je ruční instalace zdrojových kódů systému LLVM ze Subversion repozitáře projektu a zdrojových kódů nových nástrojů z Git repozitáře na adrese `http://repo.or.cz/r/clang/acc.git`. Tento postup ale nedoporučuji, protože nové nástroje nemusí fungovat s nejnovější verzí systému LLVM.

Při instalaci ze ZIP archivu postupujte následujícím způsobem:

- Připravte si alespoň 500MB volného místa, pro novější verze může být třeba víc.
- Rozbalte archiv `src.zip`.
- Přejděte do nově vytvořeného adresáře `llvm`.
- Spusťte příkaz `./configure -enable-optimized`. Bez tohoto parametru se v dalším kroku bude kompilovat verze s debug informacemi, která zabírá výrazně více místa (přes 1GB) a je pomalejší.
- Spusťte příkaz `make`. Pozor: kompilace může trvat velmi dlouho. Zkompilované programy najdete v adresáři `Release` (respektive `Debug` při vynechání parametru `-enable-optimized` v předchozím kroku).

Po úspěšné kompilaci ještě doporučuji ověřit funkčnost zkompilovaných programů. Předpokladem je kompilace bez debug informací a postup je následující:

- Rozbalte archiv `test.zip` na stejné místo, kam jste předtím rozbalili archiv `src.zip`.
- Přejděte do nově vytvořeného adresáře `test`.
- Spusťte příkaz `make test`
- Pokud některý test selže (bude označen jako „failed“), odešlete zprávu o problému na email `next_ghost@quick.cz`.

Pokud všechny testy prošly (jsou označeny jako „passed“), nové nástroje jsou připraveny k použití. Zatím není testován jen generátor run-time testů na dereference ukazatelů.

Použití

V souboru `Makefile.example` na tomto CD naleznete příklad řídícího skriptu pro program `make`, který spouští všechny nově implementované nástroje.

Program `clang-cc` nově přijímá následující parametry:

- `-macro-checks` – Při parsování vstupu proběhne kontrola, že v definicích maker jsou všechny expanze argumentů těsně uzavřeny v závorkách.
- `-analyze -side-effects` – Místo kompilace proběhne statická analýza vedlejších efektů. Program ohlásí vícenásobné použití vedlejších efektů na stejný identifikátor mezi dvěma sekvenčními body jako chybu.
- `-analyze -newbie` – Místo kompilace proběhne statická analýza běžných začátečnických chyb. Program ohlásí jako chybu porovnání čísel s plovoucí řádovou čárkou pomocí operátoru ==, přiřazení na místě podmínky, přístup ke globálním proměnným pomocí adresy a bitové operace se znaménkovými celočíselnými typy.
- `-ftrapv` – Při kompilaci bude vygenerován kód zachytávající přetečení celočíselné aritmetiky. Pokud za běhu zkompilovaného programu dojde k přetečení, program vypíše přesné místo přetečení a skončí. K výslednému programu je třeba ještě přilinkovat zkompilovaný soubor `llvm/tools/clang/llvm/rtl/rtl.c`, který zajišťuje výpis chybových hlášení.

Další užitečný nástroj je generátor run-time testů dereferencí ukazatelů. Při kompilaci vkládá kód kontrolující přístup k ukazatelům. Pokud je za běhu zkompilovaného programu dereferencován ukazatel s nulovou bázovou adresou (tedy NULL nebo NULL+index), program vypíše chybové hlášení a skončí. K programu je opět třeba přikompilovat soubor `rtl.c` zajišťující výpis chybových hlášení. Pro správnou funkčnost run-time testů doporučuji použít generátor na všechny kompilované projektu.

Pro použití generátoru je třeba do některého nástroje pro práci s mezikódem LLVM načíst dynamickou knihovnu `ACC.so` a předat parametr `-null`. V souboru `Makefile.example` je příklad použití s nástrojem `opt`. Přestože je možné dynamickou knihovnu `ACC.so` načíst i do programu `llvm-ld`, pokus o vložení run-time testů až při linkování často díky optimalizacím skončí neúspěšně.

Zde je podrobnější příklad použití generátoru:

```
clang-cc -emit-llvm-bc -g -o=foo.bc foo.c
opt -load ACC.so -null -o=bar.bc foo.bc
llvm-ld -native -o=foo bar.c
```

První příkaz zkompiluje soubor *foo.c* do mezijazyka LLVM (parametr `-emit-llvm-bc`) včetně debug informací (parametr `-g`). Druhý příkaz vloží do zkompilovaného kódu run-time testy dereferencí ukazatelů a výsledek uloží do souboru *bar.c*. Třetí příkaz vytvoří ze souboru *bar.c* spustitelný program *foo* v nativní instrukční sadě dané architektury. Debug informace v mezijazyce LLVM umožňují run-time testům vypsat podrobnější chybové hlášení. Proto je při kompilaci do mezijazyka LLVM doporučuji zapnout. Při linkování již nejsou potřeba a je možné je z výsledného programu odstranit. Program `llvm-ld` to udělá automaticky, pokud nedostane parametr `-g`.