VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií

Implementace interpretu imperativního jazyka IFJ16

Tým 021, varianta a/3/I

Vedoucí: Kyzlink Jiří (xkyzli02)
Kubiš Juraj (xkubis15)
Korček Juraj (xkorce01)
Kubica Jan (xkubic39)
Kovařík Viktor (xkovar77)

13. listopadu 2016 Brno

Obsah

1 Úvod

V této dokumentaci naleznete popis a návrh interpretu jazyka IFJ16, který je velmi zjednodušenou podmnožinou jazyka Java SE 8, což je staticky typovaný objektově orientovaný jazyk. Vybrali jsme si variantu varianta a/3/I, kde jsme měli za úkol přidat do interpretu vestavěnou funkci find, která využívala Knuth-Morris-Prattův algoritmus a funkci sort, kterou jsme měli implementovat tak, aby využívala shell sort.

–bude ještě doplněno-

2 Syntaktický analyzátor

3 Sémantický analyzátor

4 Lexikální analyzátor

5 Interpret

6 Vestavěné funkce

6.1 vestavěné funkce v IAL.c

–nevím jestli rozdělovat na podsekce ještě měnší nebo ne–

6.1.1 find

6.1.2 sort

7 Testy

Testovali jsme buď součásti - *unit testy*, kde jsme zkoušeli, zda daná funkce správně reaguje na vstupy. Unit testy si dělal každý sám a podle potřeby. Bylo zvykem v Makefile pro unit test udělat zvláštní target, kde se kromě samotné kompilace ještě prováděl *valgrind* test pro ověření možných *memory leaků*.

Dále jsme dělali ještě systémové testy. To byly vlastně testy samotného interpretu a porovnávání jeho výstupu s výstupem Javy SE 8 s přidanou kompatibilitou s jazykem IFJ16. Test byl vytvořen jako samostatný bash script, který se volal z Makefile. Ve složce test/input/ byly různé programy v jazyce IFJ16 ve formátu *navratovykod_nazevprogramu.ifj16* s možností přidání ještě souboru se stejným formátem, ale koncovkou .in, kde byla možnost dát vstup na *stdin*. Daný skript pak prošel složku, zjistil, zda jsou v ní obsaženy i soubory typu .in pro právě intepretovaný kód. Pokud byla předpokládaná návratová hodnota 0 (chyby ifj16 interpretu nemělo smysl překládat v javě a porovnávat), došlo k interpretaci kódu v javě i ifj16 interpretu s následným porovnání návratových kódů a výstupů. Vše se zapisovalo do logu, který se nacházel ve stejné složce *input* jako interpretovaný kód. Obrázek níže zobrazuje výsledky testování v průběhu raných fází interpretu.

```
MacBook-Pro-3:test viktorkovarik$ ./test
Currently working on 0_faktorial.ifj16:
[ OK ] ... IFJ16 return code is 0, and it was expected.
[FAIL] ... IFJ16 output of 0_faktorial.ifj16 is different, see input/0_faktorial.log.

Currently working on 0_helloworld.ifj16:
[ OK ] ... IFJ16 return code is 0, and it was expected.
[ OK ] ... IFJ16 output of 0_helloworld.ifj16 is correctly intepreted.
```