VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Dokumentácia k projektu z IFJ

Interpret jazyka IFJ20

TÍM 62, VARIANTA II

9. DECEMBRA 2020

ZOZNAM AUTOROV:
MATÚŠ NOSKO (XNOSKO06) 37,5% - VEDÚCI
LUKÁŠ CHMELO (XCHMEL33) 37,5%
TOMÁŠ ČECHVALA (XCECHV03) 25%

1 Úvod

Cieľom projektu bolo vytvoriť program v jazyku C. Tento program načíta zdrojový kód napísaný v zdrojovom jazyku IFJ20, ktorý je zjednodušenou podmnožinou jazyka Go a preloží ho do cieľového jazyka IFJcode20 (mezikód). Hlavnou úlohou projektu bola implementácia lexikálneho analyzátora, parsera (syntaktická a sémantická analýza) a generátora inštrukcií.

2 Lexikálna analýza

2.1 Štruktúra Tokenu

2.2 Spracovanie reťazcov

3 Syntaktická analýza

3.1 LL gramatika

1. <program> -> package main <body> EOF
2. <body> -> FUNC ID (<arguments>) (<datatypes>) (BLOCK_BEGIN EOL <statements> EOL BLOCK_END
3. <body> -> ε
4. <statements> -> <statement> EOL <statement>
5. <statement> -> IF <expression> !EOL BLOCK_BEGIN EOL <statements> EOL BLOCK_END !EOL ELSE !EOL BLOCK_BEGIN EOL <statements> EOL BLOCK_END EOL
6. <statement> -> FOR <init> SEMICOLON <expression> SEMICOLON <assign> BLOCK_BEGIN EOL <statements> EOL BLOCK_END
7. <statement> -> <function> EOL
8. <statement> -> <assign>
9. <statement> -> <init>
10. <statement> -> RETURN <value>
11. <statement> -> RETURN <expression>
12. <statement> -> RETURN <function> ?
13. <statement> -> ε
14. <assign> -> <identif> ASSIGN <function>
15. <assign> -> <identif> ASSIGN <value>
16. <assign> -> <identif> ASSIGN <expression>
17. <assign> -> ε
18. <init> -> ID INIT <value>
19. <init> -> ID INIT <function>
20. <init> -> ID INIT <expression>
21. <init> -> ε
22. <function> -> ID (<identif>)
23. <function> -> inputs (<identif>)
24. <function> -> inputi (<identif>)
25. <function> -> inputf (<identif>)
26. <function> -> print (<identif>)

27. <function> -> int2float (<identif>)
28. <function> -> float2int (<identif>)
29. <function> -> len (<identif>)
30. <function> -> substr (<identif>)
31. <function> -> ord (<identif>)
32. <function> -> chr (<identif>)
33. <arguments> -> <identif> <datatype> <argument>
34. <argument> -> , <identif> <datatype>
35. <argument> -> ε
36. <identif> -> ID <identifs>
37. <identifs> -> , ID <identifs>
38. <identifs> -> ε
39. <expression> -> ID COMPARE||ADD||SUB||MUL||DIV <value>
40. <expression> -> ID COMPARE||ADD||SUB||MUL||DIV <expression>
41. <expression> -> ID COMPARE||ADD||SUB||MUL||DIV <function> ?
42. <value> -> INTEGER
43. <value> -> DECIMAL
44. <value> -> STRING
45. <value> -> ID
46. <datatypes> -> <datatype>
47. <datatypes> -> <datatype>, <datatype>
48. <datatypes> -> ε
49. <datatype> -> int
50. <datatype> -> string
51. <datatype> -> float64
52. <datatype> -> ε

3.2 LL tabulka

	package	FUNC	ID	EOL	IF	FOR	RETURN	ASSIGN	inputs	inputi	inputf	print	int2float	float2int	len	substr	ord	chr	,	INTEGER	DECIMAL	STRING	int	string	float64	$
<program>	1																									
<body>		2																								3
<statements>				4																						
<statement>				7	5	6	10,11,12																			8,9,13
<assign>								14,15,16																		17
<init>			18,19,20																							21
<function>			22						23	24	25	26	27	28	29	30	31	32								
<arguments>																			34							33
<argument>																										35
<identif>			36																							
<identifs>																			37							38
<expression>			39,40,41																							
<value>			45																	42	43	44				
<datatypes>																			47							46,48
<datatype>																							49	50	51	52

3.3 Precedenčná tabulka

	+	-	*	/	<	>	==	>=	<=	!=	(	)	i	$
+	>	>	<	<	>	>	>	>	>	>	<	>	<	>
-	>	>	<	<	>	>	>	>	>	>	<	>	<	>
*	>	>	>	>	>	>	>	>	>	>	<	>	<	>
/	>	>	>	>	>	>	>	>	>	>	<	>	<	>
<	<	<	<	<	>	>	>	>	>	>	<	>	<	>
>	<	<	<	<	>	>	>	>	>	>	<	>	<	>
==	<	<	<	<	>	>	>	>	>	>	<	>	<	>
>=	<	<	<	<	>	>	>	>	>	>	<	>	<	>
<=	<	<	<	<	>	>	>	>	>	>	<	>	<	>
!=	<	<	<	<	>	>	>	>	>	>	<	>	<	>
(	<	<	<	<	<	<	<	<	<	<	<	=	<	
)	>	>	>	>	>	>	>	>	>	>		>		>
i	>	>	>	>	>	>	>	>	>	>		>		>
$	<	<	<	<	<	<	<	<	<	<	<		<	

4 Sémantická analýza

5 Generátor inštrukcií

6 Práca v tíme

Na projekte sme začali pracovať koncom októbra. Prácu sme delili postupne ako sme pracovali na projekte. Časti, ktoré boli zložitejšie sme riešili spoločne, jednuduchšie časti sme si rozdelili jednotlivo. Dva týždne pred odovzdaním sme začali spájať dokopy všetky časti programu. Pokusných odovzdaní sme sa nezúčastnili, keď že nám vtedy program ešte nefungoval.

6.1 Komunikácia

Komunikáciu v tíme sme riešili pomocou skupiny na messengeri a discord serveru. Stretávali sme sa každý týždeň priebežne a v posledných dvoch týždňoch sme spoločne volali vždy, keď sme pracovali na projekte. Osobné stretnutia neboli z dôvodu pandémie Covid 19.

6.2 Verzovanie

Pre správu súborov sme používali verzovací systém Git a vzdialený repozitár Github. Vďaka tomuto repozitáru sme mohli pracovať na viacerých častiach projektu súčasne. Potom sme jednoducho spojili všetky časti dokopy a každý sme mohli testovať projekt ako celok.

6.3 Rozdelenie práce

- Matúš Nosko 37,5%: vedenie tímu, testovanie, organizovanie stretnutí, sémantická a syntaktická analýza, diagramy
- Lukáš Chmelo 37,5%: lexikálna analýza, testovanie, generátor inštrukcií, syntaktická analýza
- Tomáš Čechvala 25%: lexikálna analýza, testovanie, dokumentácia, diagramy

7 Záver

Projekt bol veľmi náročný pre nás, keďže sme boli len 3. Postupne sme ale prichádzali na to, ako jednotlivé časti projektu riešiť a implementovať, hlavne vďaka prednáškam z IFJ, IAL a materiálom, ktoré sú dostupné on-

line. Jednou z hlavných problémov bol nedostatok času pri dokončovaní projektu. Taktiež nám robilo problém pochopiť niektoré časti zadania, na ktoré sme neskôr prišli. Funkčnosť projektu sme overili pomocou automatických testov. Tento projekt nám všetkým priniesol mnoho nových vedomostí a zlepšil naše schopnosti pracovať v tíme a taktiež vylepšil naše programovacie znalosti.

8 Literatúra