

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Dokumentácia – IFJ 2018

Tým 40, varianta II

Adam Hostin	xhosti02	25 %
Sabína Gregušová	xgregu02	25 %
Dominik Peza	xpezad00	25 %
Adrián Tulušák	xtulus00	25 %

Obsah

1	Úvod	2
2	Lexikálna analýza	2
2.1	Spracovanie reťazcov	2
3	Práca v tíme	2
3.1	Komunikácia	2
3.2	Verzovanie	2
3.3	Hodnotenie	2

1 Úvod

Naším cieľom je implementovať prekladač imperatívneho jazyka IFJ18 do predmetov IFJ a IAL. Hlavnou náplňou práce bola implementácia: lexikálneho analyzátora, parsera (syntaktická a sématická analýza) a generátora inštrukcií.

2 Lexikálna analýza

Na začiatku sme implementovali lexikálny analyzátor v súbore `lexer.c`, ktorého základom je deterministický konečný automat (ďalej iba DKA). Hlavnou funkciou v tomto súbore je `get_next_token`, ktorá číta jednotlivé znaky a pomocou príkazu `switch` prechádza do nasledujúcich stavov podľa DKA až kým nevyhodnotí lexikálne správny token, inak vracia `ER_LEX`. Pre jednoduchšiu prácu s tokenmi sme použili štruktúru `Token_t`, ktorá obsahovala:

- union `Token_attr`
- struct `Token_type`

2.1 Spracovanie reťazcov

Pre jednoduchšie spracovanie reťazcov sme sa rozhodli implementovať súbor `dynamic_string.c`. Jeho súčasťou je aj štruktúra `string_t`, ktorá obsahuje samotný ukazateľ na dynamický reťazec, súčasnú veľkosť reťazca a celkovú veľkosť bufferu. Na začiatku je alokovaný reťazec s veľkosťou 10 a pri každom pridaní znaku sa kontroluje, či je ešte v reťazci miesto. Keď sa blížime k zaplneniu reťazca, funkcia `check_empty_bites` zväčší veľkosť buffera o 5, čím zaistí adekvátnu veľkosť pre reťazec. Všetky alokácie pamäte sú kontrolované a ich zlyhanie je adekvátne ošetrené vrátením internej chyby `ER_INTERNAL`.

3 Práca v tíme

Náš tím sme si zostavili pomerne skoro.

3.1 Komunikácia

Už na začiatku sme sa dohodli na pravidelných týždenných stretnutiach, kde sme diskutovali o našej ďalšej práci na nadchádzajúci týždeň.

3.2 Verzovanie

3.3 Hodnotenie