

Implementační dokumentace k 1. úloze do IPP 2021/2022

Jméno a příjmení: *Martin Pech*

Login: *xpechm00*

1 Analyzátor kódu

Analyzátor zdrojového kódu je spuštěn skriptem `parse.php`. Po jeho spuštění dojde k volání skriptů `syntax.php` a `scanner.php`. Skript `scanner.php` provádí lexikální analýzu vstupního řetězce. Analýza je prováděna tak, že dojde k načtení řádku textu, který je následně rozporcován na jednotlivá slova. Projdou-li tato slova lexikální analýzou, jsou předány s příslušným tokenem a atributem skriptu `syntax.php`, který provede syntaktickou analýzu.

1.1 Skript `parse.php`

Skript `parse.php` slouží jako hlavní, řídicí a spouštěcí skript. Obsahuje ovladače výstupu na *STDERR*, ovladač na povolení *XML* výstupu, konstanty představující **chybové kódy** a **tokeny**, počítadla výskytu prvků pro rozšíření **STATP** a pole instrukcí, obsahující všechny instrukce rozdělené podle počtu operandů.

1.2 Skript `syntax.php`

Na základě informací získaných z **lexikálního analyzátoru** kontroluje tento skript syntaktickou správnost vstupního jazyka. Na základě tokenu, který přijme, očekává typ následného tokenu. Dostane-li jiný token, skončí chybou. Mimo to zde dochází ke zpracování rozšíření **STATP** a směřování *XML* výstupu do požadovaného souboru.

1.3 Skript `scanner.php`

Po načtení vstupního řetězce rozseká tento řetězec na jednotlivá slova. V případě úspěchu ukládá zpracovaná slova do tokenů realizovaných pomocí *polí*, které předává syntaktickému analyzátoru.

Příklad předávání u instrukce `MOVE GF@var int@2`:

```
Array
(
    [0] => Array                # Index 0 reprezentuje instrukci
    (
        [0] => 4                # Index [0][0] reprezentuje typ tokenu (opcode)
        [1] => 0                # Index [0][1] reprezentuje instrukci (MOVE)
    )

    [1] => Array                # Index 1 reprezentuje první operand
    (
        [0] => 2                # Index [1][0] reprezentuje typ tokenu (variable)
        [1] => GF               # Index [1][1] reprezentuje ramec
        [2] => varName          # Index [1][2] reprezentuje jméno
    )

    [2] => Array                # Index 2 reprezentuje druhý operand
    (
        [0] => 1                # Index [2][0] reprezentuje typ tokenu (constant)
        [1] => string           # Index [2][1] reprezentuje datový typ
        [2] => abc              # Index [2][2] reprezentuje hodnotu
    )
)
```

2 Implementovaná rozšíření

2.1 STATP

Rozšíření **STATP** sbírá při lexikální a syntaktické kontrole statistiky o výskytu jednotlivých instrukcí ve zdrojovém kódu. Zpravidla dochází k detekci sledované instrukce za běhu programu a následnému uložení informace do globální

proměnné.

- Pro sbírání statistik a následné uložení do souboru se používá parametr `--stats=file`, kde *file* je jméno souboru. Za tímto parametrem mohou následovat upřesňující parametry.
- Seznam upřesňujících parametrů a jejich chování je popsán v zadání projektu (sekce 3.2).
- Pokud po parametru `--stats` nenásleduje žádný soubor, dojde k vytvoření prázdného souboru. Naopak přijetí následných parametrů bez předchozího přijetí parametru `--stats` vede na chybu 10.