

Implementační dokumentace k 1. úloze do IPP 2018/2019

Jméno a příjmení: Daniel Štěpán

Login: xstepa60

Skript parse.php

Podstatou skriptu `parse.php` (psaného v jazyce PHP 7.3) je načítání zdrojového kódu v jazyce IPPcode19 ze standardního vstupu. Tento skript typu filtr provádí lexikální a syntaktickou analýzu, kde v případě úspěchu vypíše reprezentaci v jazyce XML na standardní výstup. Pro větší přehlednost kódu jsou zde pro 1. úlohu vytvořené 2 soubory. Hlavní skript `parse.php` a dále pomocný skript `parse_functions.php` obsahující funkce vykonávající dílčí úlohy.

Přijímané parametry

Ihned po spuštění skriptu dochází ke kontrole a zpracování parametrů příkazové řádky. K tomuto účelu byla využita vestavěná funkce `getopt()`. Kontrolován je zejména parametr `--help`, který vypíše nápovědu a nesmí být zadán současně s jiným parametrem. Dále se kontrolují parametry `--stats`, `--loc`, `--comments`, `--labels`, `--jumps`, sloužící pro výpis nasbíraných statistik o zdrojovém souboru.

Zpracování zdrojového kódu

K načítání vstupního souboru po řádcích (jelikož pro vstupní zdrojový soubor platí, že na každém řádku se nachází jedna instrukce) se využívá funkce `fgets()` obalená funkcí `trim()` pro odstranění bílých znaků na začátku i na konci. Bezprostředně po načtení prvního řádku programu se kontroluje, zda se zde nachází hlavička `.IPPcode19`, jinak dojde k chybě s návratovým kódem 21. Poté se předpřipraví objekt `$xml` třídy `DOMDocument` s nastavením atributů pro správné formátování, kódování a také se vytvoří první značka *program*, do které se následně budou vkládat XML značky pro jednotlivé *instrukce*.

Stěžejní částí skriptu je cyklus `while`, který má v podmínce funkci `fgets()` načítající text ze standardního vstupu. Od tohoto momentu se vždy načte jedna instrukce i s parametry a kontroluje se jejich syntaktická a lexikální správnost. Po načtení řádku se odstraňují komentáře, zpočátku zde byl problém s komentáři za instrukcí/parametrem bez oddělení bílým znakem. To se podařilo odstranit nahrazením každého znaku na řádku `"#"` za sekvenci znaků `"#"`. Poté se string obsahující řádek rozdělí na prvky pole funkcí `preg_split()`, které se dá jako parametr bílý znak zadaný regulárním výrazem.

Po odstranění komentářů a bílých znaků je tedy stav takový, že máme pole, ve kterém je na nultém indexu uložen operační kód instrukce a na dalších prvcích pole jsou případné parametry. Syntaktická a lexikální kontrola parametrů probíhá s využitím regulárních výrazů a funkce `preg_match()`. Skript ví na kterém místě a indexu v tomto poli má očekávat parametr typu *var*, *symb*, *type* nebo *label*. Tudíž aplikuje konkrétní regulární výraz. Pokud narazí na chybný zápis, vypíše chybové hlášení na standardní chybový výstup a program končí s návratovým kódem 23.

Generování XML a výstup skriptu

Výstupem skriptu je XML reprezentace vstupního kódu, k jehož generování je využita třída `DOMDocument`.

Rozšíření

Základní funkčnost skriptu je rozšířena o rozšíření s identifikátorem STATP. Zadáním parametru `--stats=file` určíme soubor, do kterého se budou zapisovat agregované statistiky. Na každém řádku je uvedena vždy jen jedna hodnota a to v pořadí, v jakém následují další parametry. Parametr `--loc` vypíše počet řádků, na kterých se nacházela nějaká instrukce (úvodní řádek nepočítaje). Parametrem `--labels` se vypíše počet definovaných návěští (`LABEL`) a poslední parametr `--jumps` vypíše počet instrukcí pro podmíněné a nepodmíněné skoky (počítají se instrukce `JUMP`, `JUMPIFEQ`, `JUMPIFEQS`, `JUMPIFNEQ`, `JUMPIFNEQS`).