

Implementační dokumentace k 1. úloze do IPP 2020/2021

Jméno a příjmení: Adam Marhefka

Login: xmarhe01

Moje riešenie skriptu `parse.php` sa začína kontrolou parametrov skriptu, špecificky parameter `--help` a všetky parametre rozšírenia `STATP`. Parametre rozšírenia `STATP` kontrolujem pomocou konštrukcie `switch` v jednotlivých návěštích `case`. Tieto parametre si ukladám do dvojrozmerného poľa reťazcov, kde prvý rozmer reprezentuje súbory (teda jeden prvok poľa za každý výskyt parametru `--stats=file`) a druhý rozmer sú samotné parametre v poradí v ktorom sa budú do súboru vypisovať. Súbory pre výpis štatistík otváram pomocou funkcie `fopen`.

Na načítanie vstupu zo štandardného vstupu používam funkciu `fgets` spolu s cyklom `while`. Ako prvú kontrolujem prítomnosť hlavičky jazyka `IPPCODE21`, pri čom preskakujem všetky prázdne riadky a komentáre. Po úspešnom nájdení hlavičky sa začína syntaktická analýza samotného kódu. Opäť používam funkciu `fgets` a cyklus `while` a načítaný riadok si vždy vo vlastnej funkcii rozdelím podľa bielych znakov na pole reťazcov. Funkcia vždy preskočí všetky komentáre a prázdne riadky. Na nultom indexe poľa navráteného z funkcie musí byť prítomná jedna z inštrukcií jazyka `IPPCODE21`. Či je inštrukcia validná kontrolujem pomocou konštrukcie `switch`, ktorá má prípad pre všetky inštrukcie `IPPCODE21`. Na nasledujúcich indexoch poľa sú potom samotné operandy pre danú inštrukciu. V jednotlivých prípadoch konštrukcie `switch` ďalej kontrolujem či sedí počet operandov pre inštrukciu a pomocou funkcie `preg_match` kontrolujem či tieto operandy odpovedajú regulárnym výrazom povoleným pre danú inštrukciu. Použil som 4 regulárne výrazy tak ako špecifikovalo zadanie (tzn. `var`, `symb`, `label` a `type`).

Na generovanie výstupného xml som použil knižnicu `xmlwriter`. Výhodou tejto knižnice bolo, že problematické znaky pre xml (`&`, `<`, `>`) automaticky prevedie na xml entity (`&`, `<`, `>`). Pri kontrole typu argumentu pre výstupné xml som opäť použil regulárne výrazy a funkciu `preg_match`. Výstupné xml je generované za behu skriptu a vypíše sa na štandardný výstup iba v prípade úspešného ukončenia syntaktickej analýzy vstupného kódu.

Štatistické rozšírenie `STATP` som implementoval pomocou viacerých počítadiel, ktoré sa inkrementujú za behu skriptu no vypíšu sa iba pri uvedení špecifických parametrov. V jednotlivých prípadoch konštrukcie `switch` v ktorej kontrolujem operačné kódy viem ktoré počítadlo mám inkrementovať. Problematickejšie bolo implementovať počítanie rôznych typov skokov. To som nakoniec vyriešil pomocou poľa do ktorého sa ukladajú všetky návěštie a ak bolo pri inštrukcii skoku návěštie nájdené v poli, ide o skok dozadu, inak sa inkrementuje počítadlo skokov na neplatné návěštie a návěštie ktoré nebolo nájdené v poli je uložené do poľa doposiaľ nenájdenných návěští. Vždy pri inštrukcii `label` potom skontrolujem pole doposiaľ nenájdenných návěští a ak sa tam práve definované návěštie nachádza tak sa dekrementuje počítadlo skokov na neplatné návěštie a inkrementuje počítadlo skokov dopredu.