

Implementační dokumentace k 1. úloze do IPP 2019/2020

Jméno a příjmení: Tomáš Tlapák

Login: xtlapa00

Arguments

Ke zpracování argumentů využívám funkci `getopt`. Jelikož funkce `getopt` si ukládá jednotlivé argumenty do pole `$options` jako klíče s hodnotou `bool(false)` nebo v případě zadání vícekrát stejného argumentu jako klíč s hodnotou typu pole, které obsahuje takový počet hodnot `bool(false)`, kolikrát byl daný argument zadán, nemohl jsem tedy použít podmínku porovnávající počet uložených klíčů s počtem předaných argumentů do skriptu: `if (count($options) != $argc-1)` (při zadání vícekrát stejného validního argumentu by podmínka havarovala), proto tedy následně ručně iteruji přes pole argumentů a porovnávám pomocí regulárního výrazu, jestli byly zadány pouze validní argumenty. Pokud byl zadán nepovolený argument nebo nepovolená kombinace argumentů, skript končí s návratovou hodnotou 10.

Jelikož podporuji rozšíření STATP, po kontrole validních argumentů si do globální proměnné ukládám hodnoty argumentů, v jakém pořadí byly zadány a v jakém pořadí se mají po úspěšném proběhnutí skriptu vypsát do výstupního souboru.

Parse

Vstup procházím řádek po řádku přes `while ($line = fgets(STDIN))`, každý řádek prvně pomocí funkce `preg_split` zbavím případného komentáře, poté si pomocí funkce `strtok` oddělím první slovo a porovnám, jestli se jedná o validní operační kód, pokud ne, skript končí s návratovou hodnotou 22 značící neznámý operační kód. V opačném případě následně kontroluji pomocí regulárních výrazů, zda daný řádek s operačním kódem má správný počet operandů a zda jsou lexikálně syntakticky správně zapsané. Pokud daný řádek je lexikálně a syntakticky správně, předávám zachycené shody pomocí regulárního výrazu do funkce `XML_generator`, v opačném případě skript končí s návratovou hodnotou 23.

XML_generator

Pro generování využívám knihovnu `XMLWriter`. Pro generování používám obdobnou konstrukci switche jako ve funkci `Parse`, kdy podle operačního kódu rozhoduji, kolik daný XML element má mít argumentů a jakého typu. `XMLWriter` sám převádí znaky `&`, `"`, `<`, `>` na příslušné XML entity. Nepřevádí pouze znak jednoduché uvozovky `'`, používám tedy funkci `str_replace` a nahrazuji `'` za příslušnou XML entitu `'`.

Output

Ve funkci iterativně procházím pole, které jsem si naplnil argumenty ve funkci `Arguments` a porovnávám přes konstrukci switche o jaký argument se jedná. Podle argumentu si následně do proměnné přidávám příslušnou proměnnou reprezentující příslušný čítač dané statistiky. Statistiky se vypíší do souboru zadaného přes argument `--stats=<název_souboru>`, pokud daný soubor neexistuje, je vytvořen, pokud uživatel má příslušná práva, jinak skript končí chybou 12.