

Dokumentace úlohy CST: C Stats v PHP5 do IPP 2014/2015

Jméno a příjmení: Vladimír Ščešňák

Login: xscesn00

Úvod

Tento program se používá k počítání statistik zdrojových kódů v jazyce C. Ovládání programu se zobrazí po zadání argumentu `--help`.

Zpracování parametrů

Ke zpracování parametrů slouží funkce `parse_arguments()`. Na vstupu se očekává jeden nebo více argumentů. Při počtu parametrů, který se rovná jedné, očekává skript parametr `--help` nebo `-k`. `-o`, `-i`, `-w=`, `-c`. Při větším počtu parametrů můžeme zadat parametry `--input=file`, `--output=file`, `--nosubdir`, `-p`, nebo již výše uvedené parametry `-k`, `-o`, `-i`, `-w=`, `-c`, které však mezi sebou nelze kombinovat. Pokud na vstup přijde neočekávaný parametr, program ukončí svou činnost s chybou a návratovým kódem 1. V případě, že byly argumenty správně, program dále postupuje podle zadaných parametrů.

Práce se soubory

Uživatel může argumentem zadat, který soubor nebo složku chce analyzovat. Pro kontrolu souborů slouží funkce `check_files()`. Funkce vyhodnotí zda se jedná o soubor, zkontroluje se, zda zadaný soubor existuje a zda je čitelný, ne-li, program ukončí svou činnost s chybou a návratovým kódem 21. Pokud se jedná o adresář, program otestuje, zda adresář existuje a pokud ano, rekurzivně projde adresářovou strukturu. V případě, že se tam nacházejí soubory v jazyce C, tedy soubory s příponou `.ca` `.h`, funkce je uloží do pole `array_files`. Zadáním argumentu `--nosubdir` adresářová struktura není procházena rekurzivně, ale zůstává jen v kořenovém adresáři.

Pokud není zadán soubor ani složka, program implicitně prochází složku ve které se nachází.

Také je možné zadat soubor, do kterého se statistiky uloží, pokud ale argument není použit, statistiky se vypisují na standardní výstup. Při korektním zadání soubor je soubor zkontrolován, v případě, že do souboru nelze zapisovat program ukončí svou činnost s chybou a návratovým kódem 3.

Rozklad zdrojových kódů

Funkce `exit_status()` slouží na výpis chybové hlášky a ukončení programu s chybovým kódem. Funkce `check_files()` a `parse_arguments()` již byly popsány výše, avšak funkce `parse_arguments()` má na starosti i spouštění jednotlivých počítání statistik.

V případě zadání parametru `-k` tj. počítání klíčových slov, se zavolá funkce `get_keywords()`, pokud zadáme parametr `-o`, chceme tedy statistiku výskytu operátorů, tak se zavolá funkce `get_operators()`. Po zadání parametru `-i` počítáme statistiku identifikátorů a zavolá se funkce `get_identifiers()`. Hledáme nějaký výraz a zadáme parametr `-w= výraz`, zavolá se funkce `get_patterns()`, a v případě statistiky komentářů, na kterou je nutné použít parametr `-c` volá se funkce `get_comments()`.

V každé funkci se dále nachází funkce, která má na starosti spočítání hledaných výrazů, které jsou zadány regulárním výrazem, nebo v případě komentářů konečným automatem.

Konečným automatem je také implementováno odstranění komentářů, odstranění řetězců a odstranění maker, které jsou volané ve funkcích, kde je třeba se těchto částí kódu zbavit. Regulárním výrazem je také vyřešeno odstranění ukazatelů, to má na starosti funkce `del_pointers()`.

Výstup

Formát výstupu je pevně dán v zadání projektu a kromě správného odsazení je potřebné soubory i správně seřadit. Výstup se ukládá do asociativního pole, kde první položka je soubor s celou cestou. V případě, že byl zadán parametr `-p`, tak je uložen pouze název souboru. Další položka je získána statistika ať klíčových slov, identifikátorů, operátorů, hledání přesného výrazu, nebo statistika komentářů. Při výpisu je ještě třeba oba položky správně odsadit, to mají na starosti funkce `spaces()` a `stats_spaces()`. Uživatel také může výstup přesměrovat do souboru a to pomocí parametru `-output=název_souboru`, jinak je výstup zobrazen na STDOUT.

Závěr

Projekt byl implementován is jedním rozšířením a to konkrétně s rozšířením COM a následně byl podroben sadou testů.