

1 Popis implementace

1.1 Rozpoznání a kontrola parametrů

Zjišťování zadaných parametrů je realizováno cyklem foreach, který prochází pole s argumenty @ARGV a dále v podmínkách rozpoznává jednotlivé parametry.

Jednoduché parametry -i, -k, -o, -c, -ik, -p a --help jsou porovnávány s jejich textovou reprezentací pomocí operátoru eq.

Parametry --input, --output a -w jsou zpracovávány v podmínkách pomocí regulárního výrazu a operátoru =~ a jejich hodnoty získány pomocí funkce split. Hodnoty se ukládají do globálních proměnných.

Chyby a špatné kombinace parametrů jsou ohlášeny na standardní chybový výstup a v takovém případě je návratovým kódem číslo větší než nula.

1.2 Procházení adresářové struktury

Pokud nebyl v parametru --input zadán vstupní soubor nebo adresář, prochází se adresář, odkud byl skript spuštěn.

Pro procházení adresářové struktury je využito pole, které slouží jako zásobník pro dosud neprohledané adresáře, a hash, který slouží pro uložení absolutní cesty souboru jako klíče a jeho hodnota je inicializována na číslo nula.

Když nebyl zadán existující soubor nebo adresář anebo nastala jiná chyba, je vypsáno chybové hlášení na standardní chybový výstup a návratovým kódem skriptu je číslo větší než nula.

1.3 Vyhodnocení a výpis výsledků

K vyhodnocení statistik souboru jsou využívány dvě funkce. Jedna má název retez a realizuje vyhledávání řetězce <pattern> parametru -w. Druhá se jmenuje others a používá se k vyhledání ostatních statistik.

Funkce retez prochází soubor znak po znaku a hledá počáteční znak zadaného hledaného řetězce. Pokud takový znak najde, načte od něj tolik znaků, kolik má vyhledávaný řetězec a tato posloupnost znaků se pomocí operátoru eq porovná s hledaným řetězcem. Když načtená posloupnost souhlasí, pokračuje procházení souboru, v opačném případě se pozice v souboru vrátí za nalezený výskyt prvního písmene a začíná hledání znova.

Funkce others prochází soubor také znak po znaku a obsahuje konečný automat. V konečném automatu se rozpoznávají lexémy pro klíčová slova, operátory, identifikátory, ale komentáře jsou počítány znak po znaku. Klíčová slova a operátory jsou uloženy v poli, ve kterých se nalezený lexém vyhledává. Porovnávání je prováděno v podmínkách pomocí operátoru eq nebo pomocí regulárního výrazu a operátoru =~.

Počty jednotlivých lexémů se ukládají do lokálních proměnných, aby mohly být pro každý soubor vráceny jako návratová hodnota a zároveň se počet přičítá do globálních proměnných pro konečný celkový počet.

Procházení souboru je ukončeno, pokud se narazí na jeho konec.

2 Řešené problémy

2.1 Dva různé konce řádku

Protože Unixové systémy používají pro zakódování konce řádku jeden bajt a Windowsové systémy bajty dva, bylo to nutné ošetřit především v počítání počtu znaků komentáře a pro vypuštění maker preprocesoru ze zpracování ve statistikách. Dvoubajtový konec řádku je započítán jako bajt jeden.

Rozpoznávání konce řádku je realizováno pouhým porovnáním načteného znaku s jeho escape sekvencí pomocí operátoru eq.

2.2 Prefixy a sufixy konstant

Započítání prefixů a sufixů do statistik, především identifikátorů, bylo nežádoucí a muselo být ošetřeno. O jejich vynechání se stará konečný automat, který provádí rozpoznávání lexémů.

3 Metriky kódu

Počet souborů: 1

Počet řádků zdrojového textu skriptu: 778

Velikost skriptu v bajtech: 20 966

4 Použité knihovny

File::Basename

Cwd abs_path