

## 1. Zadání

Vytvořit program (skript) v jazyce PHP5 pro analýzu a výpis statistik zdrojových souborů jazyka C.

## 2. Implementace

### 2.1. Zpracování parametrů

Parametry jsou zpracovávány klasicky pomocí proměnných `$argv`, `$argc` a cyklu. Před samotným zpracováním se vymaže první prvek proměnné `$argv` čímž se zbavím názvu programu, dále následuje cyklus, který dané pole prochází. V případě chyby (nedovolená kombinace, chybějící část parametru, opakování parametru...) se vypíše chybové hlášení a program se ukončí s daným návratovým kódem.

### 2.2. Práce se soubory

Program může pracovat buď s jedním vstupním souborem nebo s adresářem. V případě, že bude v parametrech zadán adresář, začnou se od dané složky rekurzivně vyhledávat soubory s koncovkou `.c` nebo `.h`. Jména souborů, která odpovídají formátu se uloží do pole. Pokud vstupní parametr chybí, pracuje se s aktuálním adresářem, uživatel také může omezit prohledávání pouze na aktuální složku.

Výstupní soubor může být také zadán parametrem a v případě, že daný parametr chybí, tak se data vypíšou na standardní výstup.

### 2.3. Podpůrné funkce pro analýzu

Jednotlivé módy pro analýzu potřebují pro svou správnou funkčnost pozměněný text (odstraněné komentáře, makra atd.) proto se při každém módu volají i potřebné podpůrné funkce. Před samotným zpracováním se vždy daný soubor otevře, a celý jeho obsah se uloží do proměnné, následně se soubor uzavře a již se z ním nepracuje.

#### 2.3.1. Odstranění maker

Tuto operaci provede funkce `Destroy_MACRO()`. V této funkci se nejdříve převedou záměny `\r\n` konce řádku na `\n`. Dále se v cyklu prochází jednotlivé řádky a pomocí stavového automatu se rozhoduje, zda daný řádek uložit do výsledného textu nebo ne. Podporují se i makra jenž pomocí `\` pokračují na dalším řádku.

#### 2.3.2. Odstranění komentářů

Funkce `Destroy_COMMENT()` odstraní veškeré komentáře, ale zachová při tom řetězce, které mohou jako komentáře vypadat. Text se nejdříve rozdělí na jednotlivá písmena a pak se pomocí cyklu a stavového automatu rozhoduje, zda dané písmeno vložit do výsledného textu nebo ne. Řádkové komentáře `//` jenž pomocí `\` pokračují na dalším řádku jsou taktéž podporovány.

#### 2.3.3. Odstranění řetězců

Provádí funkce `Destroy_STRING()`. Odstraní řetězce a znakové literály, zachovává ale komentáře i pokud se v nich objeví znaky řetězců (jsou to pouze komentáře). V této funkci se taktéž pomocí stavového automatu rozhoduje, zda daný znak bude ve výsledném textu či nikoliv.

#### 2.3.4. Odstranění klíčových slov

Poslední z podpůrných funkcí – `Destroy_KEYWORDS()` odstraní z textu klíčová slova. Děje se tak na základě regulárního výrazu a funkce `preg_replace()` jenž z textu odstraní všechna slova z pole `$patterns` a nahradí je mezerou. Aby se nevymazala i část slov, která se z klíčových slov skládají, je každý prvek pole `$patterns` omezen `(\bKL_SLOVO\b)`. `\b` má zajistit aby vymazala slova pouze přesného formátu.

### 2.4. Analýza a jednotlivé módy

#### 2.4.1. Mód -k KLÍČOVÁ SLOVA

Tento mód realizuje funkce `Find_KEYWORDS()`, která funguje podobným principem jako odstraňování klíčových slov, ovšem místo mazání se pouze počítá výskyt jednotlivých klíčových slov. Před voláním této funkce se z textu odstraní veškeré definice maker, komentáře a řetězce – v těch klíčová slova nepočítáme.

#### 2.4.2. Mód -i IDENTIFIKÁTORY

Funkce `Find_IDENT()`, najde pomocí jednoduchého regulárního výrazu, výskyt veškerých identifikátorů. Před jejím voláním odstraníme definice maker, řetězce, komentáře i klíčová slova (pokud bychom klíčová slova neodstranili, započítaly by se jako identifikátory).

#### 2.4.3. Mód -c KOMENTÁŘE

Komentáře spočítá funkce `Find_COMMENT()`, jednotlivými znaky se prochází pomocí stavového automatu, který rozhoduje, zda znak patří do komentáře či nikoli. Podporuje se i zalomený řádkový komentář. Před jejím zavoláním se odstraní makra preprocesoru (v nich komentáře neuvažujeme) a pro bezpečnost i řetězce (řetězec, co vypadá jako komentář, není komentář).

#### 2.4.4. Mód -w

Tento mód využívá funkci `Find_PATTERN()`. Funkce pomocí primitivního regulárního výrazu a funkce `preg_match_all()` najde nepřekrývající se řetězce. Hledá se v celém textu, nic se předem neodstraňuje.

#### 2.4.5. Mód -o OPERÁTORY

V tomto módu se před zavoláním funkce `Find_OPERATORS()` vymažou veškeré komentáře, makra a řetězce, dodatečně se v této funkci vymaže i vše co by bylo možné považovat za deklaraci ukazatele (to se nepovažuje za operátor), děje se tak na základě trochu složitějšího regulárního výrazu. Dále se, také na základě regulárních výrazů, vyhledávají výskyty jednotlivých operátorů.

### 2.5. Formátování výstupu

Po úspěšné analýze lexikograficky seřadíme soubory pro které jsme analýzu prováděli. Jelikož je formát pevně dán a názvy souborů jsou zarovnány nalevo a spočítané hodnoty napravo, musíme spočítat počet mezer, které mají toto formátování udržet. Počet mezer se spočítá z rozdílu mezi nejdelší délkou souboru a délkou daného souboru a rozdílu nejdelšího čísla a spočítaného čísla pro daný soubor.

