

## 1) ÚVOD

Implementační dokumentace popisuje řešení 1. projektu do předmětu IPP. Vybral jsem si zadání CST(C Stats v Perlu). Úkolem byla implementace programu pro analýzu zdrojového kódu v jazyku C. Program počítá statistiky kódu jako počet identifikátorů, operátorů, klíčových slov, komentářů nebo výskytů zadaného řetězce. Program je napsán v jazyku Perl a je ovládán z příkazové řádky.

## 2) POPIS ŘEŠENÍ

### a) Zpracování parametrů příkazové řádky

Funkce pro zpracování parametrů příkazové řádky, je volána ihned na úvod programu, kontroluje správnou kombinaci zadaných parametrů a ukládá je do proměnných. U parametrů, z kterých je potřeba získat nějaká informace jako vstupní soubor apod., je provedeno ořezání a uložení do vhodné proměnné.

### b) Prohledávání souborů od místa uložení hlouběji

Skript musí umět analyzovat soubory s koncovkou .c a .h. ve spuštěném adresáři, tak i v podadresářích. Pro řešení tohoto problému jsem si vytvořil dvě pole, které používám jako seznamy, jedno pro uložení všech adresářů a druhé pro uložení všech analyzovaných souborů včetně cesty. Nejdříve si pole s adresáři nainicializuji prvním adresářem (např. zadaný z přík. řádky). Cyklus prochází pole s adresáři. V těle cyklu otevřu adresář a v něm vyhledám další adresáře (přidám na konec pole s adresáři) a soubory .c a .h (uložím do pole souborů), jakmile projdu všechny prvky pole s adresáři cyklus končí a všechny soubory k analýze jsou uloženy v poli se soubory.

### c) Analýza zdrojového souboru

#### I.) Počet všech klíčových slov

Tuto problematiku řeším konečným automatem. Přeskakuje komentáře a řetězce, jakmile načte celý identifikátor (začíná malým, velkým písmenem nebo podtržítkem a může být následován malými nebo velkými písmeny), zkontroluje zda se jedná o klíčové slovo, pokud ano inkrementuje počítadlo a vrací se do počátečního stavu.

#### II.) Počet operátorů

Počet operátorů se snažím spočítat pomocí konečného automatu. Nadstandardní funkčnost za bonusové body nebyla implementována.

#### III.) Počet identifikátorů včetně klíčových slov

Konečný automat přeskakuje komentáře a řetězce. Identifikátor rozpozná podle počátečního znaku, který může být malé, velké písmeno nebo podtržítko, za ním mohou následovat malá nebo velká písmena, podtržítko nebo číslice.

#### IV.) Počet identifikátorů

Řešeno konečným automatem. Podobně jako část III., ale bez klíčových slov.

#### V.) Počet výskytů hledaného řetězce

Počet jednotlivých výskytů hledám pomocí regulárních výrazů. Hledaný vzor před samotným použitím ještě upravím, před každý metaznak vložím znak, který zruší platnost speciálního znaku.

#### VI.) Počet znaků v komentářích

Řešeno konečným automatem, přeskakuje řetězce. Jakmile narazí na komentář (// nebo /\*), začne počítat jednotlivé znaky, dokud nenasadí na konec komentáře (/n nebo /\*).

### d) Tisk statistik do souboru nebo na standardní výstup

Podle zadaného parametru probíhá tisk do souboru nebo na standardní výstup. Pro uložení údajů používám pole asociativních polí, kde je uložena celá cesta, název souboru a statistika. Podle přepínače dojde k seřazení podle požadovaného klíče a k tisku. Správný odsazení je vypočítáno na základě délky nejdelšího jména souboru respektive absolutní cesty.

## 3) ZÁVĚR

Skript provádí vše podle zadání, byl testován poskytnutými vzorovými testy a také testy vlastními. Výsledky testů prokázali správnost řešení.