

## 1 Úvod

Tato dokumentace pojednává o skriptu napsaném v jazyce Python. Jeho účelem je analyzovat soubory obsahující zdrojový kód jazyka C podle standardu ISO C99. Typ prováděné analýzy je vždy daný parametrem při spuštění skriptu. Níže v dokumentu naleznete popis implementace těchto typů analýzy či dalších částí skriptu.

## 2 Zpracování parametrů

Ke zpracování parametrů skript využívá standardní knihovnu jazyka Python – `argparse`, díky které stačí nastavit ve skriptu názvy přepínačů a typ ukládané hodnoty.

Použitím této knihovny je zároveň zajištěno částečné ověření parametrů. Například pokud je skript spuštěn s parametrem, který není mezi nastavenými, skript se automaticky ukončí. Jelikož však výchozí metoda obstarávající vypisování chyby parametrů nevyhovuje tím, že při chybě parametrů vrací hodnotu 2, je potřeba předefinovat tu metodu vlastní.

Co však tato knihovna neřeší, je vícenásobné zadání stejného parametru. Tento problém je vyřešený sečtením parametrů a ověřením, zda je daný parametr zadán více než jedenkrát. Stejným způsobem je vyřešeno i zadání více parametrů, které jsou dle zadání nekombinovatelné.

## 3 Analýza souboru

### 3.1 Klíčová slova

Účelem této analýzy je sečíst všechna klíčová slova ze zdrojového souboru, která se nevyskytují v poznámkách, makrech a řetězcích. Prvním krokem je tedy odstranění částí, které nemají být prohledávány, z načteného obsahu souboru. To je řešeno pomocí regulárních výrazů, kterými se vyhledají a odstraní všechny komentáře, řetězce a makra. Poté jsou, opět pomocí regulárních výrazů, vyhledána všechna slova, u kterých se zjišťuje, za jsou jedním z hledaných klíčových slov. Pro tento účel je do skriptu importován modul `keyword.py`, v němž je vytvořený seznam klíčových slov jazyka C, s nimiž se vyhledaná slova porovnávají.

### 3.2 Jednoduché operátory

Při analýze operátorů je potřeba, podobně jako u analýzy klíčových slov, odstranit všechny komentáře, řetězce a makra, a navíc ještě vyhledat a odstranit deklarace ukazatelů, jelikož v tomto případě není `*` považována za operátor. Poté jsou již operátory vyhledány regulárním výrazem a sečteny.

### 3.3 Identifikátory

Implementace analýzy identifikátorů je téměř totožná jako analýza klíčových slov. Jediným rozdílem je, že se sečtou slova, která nevyhovují žádnému ze slov uložených v seznamu v importovaném modulu `keyword.py`.

### 3.4 Textový řetězec `pattern`

Analýza textového řetězce se spouští zadáním parametru `-w=pattern`, kde `pattern` je přesný řetězec, který má být v analyzovaném souboru vyhledán. Počet výskytů se v této analýze zjišťuje metodou `count`.

### 3.5 Komentáře

Počítání komentářů je implementováno jako konečný automat. Obsah zdrojového souboru se prochází znak po znaku, přičemž na základě posledních dvou znaků se nastavují hodnoty booleovských proměnných, které následně reprezentují stavy konečného automatu – proměnná `comment` informuje, že aktuálně načítané znaky jsou součástí komentáře, `oneLineComment` zda se jedná o jednořádkový komentář a proměnná `string` informuje že aktuální znaky jsou součástí řetězce.

Řetězce jsou v konečném automatu rozpoznávány, jelikož je nelze odstranit před samotným počítáním znaků (mohou být součástí komentáře), avšak znaky označující začátek či konec komentáře mohou být zapsané v řetězci. V takovémto případě se však nejedná o komentář a tyto znaky se nemají započítávat.

Tato implementace načítání komentářů díky své konstrukci splňuje bonusové rozšíření, spočívající v započítávání znaků komentáře i v rámci definic makra.

## 4 Výstup

Po analýze jednoho či více souborů je potřeba zpracovaná data vypsát podle volby buď to na standardní výstup, nebo do souboru. Jelikož je však ve výstupu součet výskytu zarovnaný doprava, je potřeba zjistit odsazení pro jednotlivé soubory. Pro odsazení je potřeba nejdříve nalézt nejdelší řetězcovou délku cesty a součtu výskytu hledaných prvků u analyzovaných souborů. Od této hodnoty se poté odečítá řetězcová délka cesty a počtu výskytů pro každý vypisovaný soubor.