

Dokumentace k projektu IPP, C Header Analysis

Analýza problému

Úkolem bylo vytvořit skript pro analýzu hlavičkových souborů jazyka C (přípona .h) podle standardu ISO C99, který vytvoří databázi nalezených funkcí v těchto souborech.

Postup řešení

Prvním krokem bylo zpracování argumentů. Pro zpracování argumentů byla použita knihovna *argparse*, což dovolilo ušetřit hodně řádků kodu. Dalším krokem byla práce s adresáři. V závislosti na zadaném parametru `--input` se vyhledávají soubory s příponou .h a následně se ukládají do listu `header_list`. Po zpracování vstupního adresáře a uložení analyzovaného souboru(souborů) `header_list` se postupně prochází cyklem, každý soubor je otevřen pro čtení a rozparsován pomocí stavového automatu. Na konci zpracování všech souborů jak výsledek máme jeden list OUT, ve kterém následují za sebou elementy v takovém pořadí: {dictionary with function description}, [list with types of function parameters]. List OUT je následně procházen cyklem, ve kterém se vytváří výsledný XML dokument.

Zpracování parametrů

Zpracování parametrů zadaných v příkazové řádce se provádí pomocí knihovny *argparse*, která dělá proces zpracování mnohem jednodušší, čitelnější a šetří místo.

Práce s adresářovým systémem

Pro práci s adresářovým systémem byla použita knihovna *os*. Byly použité takové funkce jak: `os.path.abspath`, `isdir`, `isfile`. Velmi zručné bylo použití `os.path.walk`, které provádí rekurzivně procházení adresářovým stromem. O detekci hlavičkových souborů pomocí použití volání `walk` se stará funkce `det_header`.

Stavový automat

Způsobem parsování hlavičkového souboru bylo zvoleno “ručné parsování” pomocí stavového automatu na základě zkušeností s programováním lexikální analýzy v předmětu IFJ. Pro parsování je také použité tři pomocné funkce: `det_rett_and_name`, `detect_params` a `analyze_typedef`. `det_rett_and_name` se stará o rozdělení návratového typu funkce a jména funkce. `detect_params` se stará o zpracování parametrů funkce a `analyze_typedef` provádí detekci nových datových typů definovaných uživatelem a jejich následném uložení do pole s klíčovými slovy `keywords`. Po zpracování všech funkcí a souborů v listu OUT jsou uloženy seznamy pro výpis do výsledného XML souboru.

Tvorba stromu elementů xml

Procházením výsledného seznamu OUT se postupně vytvářejí jednotlivé elementy a argumenty elementů v kořenovém stromě xml. Pro práci s XML soubory byl použit modul xml.dom.minidom.

Zdroje informace

- <https://www.programiz.com/python-programming#learn-python-tutorial>
- <https://docs.python.org/3>
- <https://www.google.cz>