

Список модулів програми

Main

Точка входу в програму. Виконує відкриття файлу з кодом на мові асемблера, зчитування коду та передачі його спочатку до лексичного аналізатора, а потім до генератора лістингу.

Lexical analyser

Містить необхідні процедури та константи для розпізнання лексем та їх розбору.

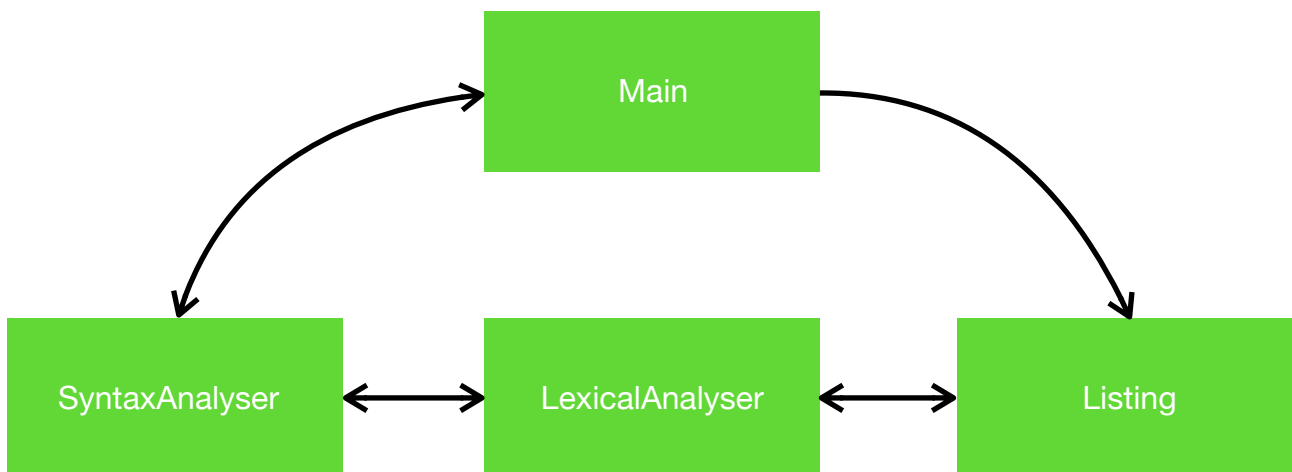
SyntaxAnalyser

Модуль для синтаксичного аналізу строки з кодом. Містить методи для визначення типу строки (директива, команда, визначення змінної, тощо) та поля з результатом лексичного аналізу (операнди та їх типи) для подальшої генерації лістингу.

Listing

Модуль, що отримує на вході результат синтаксичного аналізу та на його основі будує файл лістингу. Містить процедури для коректної обробки директив та генерації зсувів.

Взаємозв'язок між модулями програми



Опис функцій модулів програми

Main

Містить одну функцію для підключення інших модулів, зчитування файла з асемблерним кодом та роботи з ним.

Lexical analyser

strType(what, target)

Перевіряє строку *target* на відповідність типу *what*.

expand(kindOf, target)

Перетворює строку з лексемами на масив.

getType(target)

Визначає тип для окремої лексеми *target* (регістр, константа, адресний вираз, тощо).

isFunction(target)

Визначає, чи є строка асемблерною командою.

SyntaxAnalyser

constructor(lexeme, line, eq)

Конструктор, ініціалізує поля для збереження інформації про тип строки з лексемами, її операндів та їх типів, номер строки.

getType()

Повертає тип строки.

findLexType()

Передає строку з кодом до лексичного аналізатора, намагаючись визначити її тип. Якщо тип було невизначено, то строка помічається як помилкова. Якщо ж тип визначено - викликається відповідний до нього обробник.

parse%type_name%()

Обробник для строки типу %type_name%. Дістає з неї операнди, та визначає їх типи, записує тип строки до окремої змінної для подальшої генерації лістингу.

Listing

constructor(lexemes, equ)

Конструктор, ініціалізує поля для збереження інформації про поточний сегмент, зсув, константи EQU, ініціалізовані змінні та їх розміри, значення і сегменти. Також містить буфер файла лістингу, який по завершенню аналізу записується у файл listing.lst.

getShift()

Повертає поточний зсув у шістнадцятковому форматі з лідуючими нулями.

buildListing()

Проходиться по масиву з лексемами та в залежності від їх типу викликає відповідний звичайну для генерації зміщення якщо це команда, або спеціальний обробник якщо це директива.

process%directive_name%()

Виконує обробку та генерації зсуву (якщо необхідно) для строки з директивою. В залежності від директиви може модифікувати поля з поточною директивою, словник з визначеними змінними та словник з сегментами.

calc%comand_name%()

Виконує підрахунок зсуву для команди %comand_name%, опираючись на її операнди та їх типи. Для команд з операндом - адресним виразом викликається функція для перевірки наявності та обробки префіксу заміни сегмента.

generateSegPrefix(addr, ident, base, index)

Перевіряє наявність префіксу заміни сегменту у адресному виразі *addr* та за його наявності в залежності від сегменту, де була визначена змінна та складових виразу додає 1 байт до поточного зсуву, або ні.