**Київський національний університет імені Тараса Шевченка**

**Факультет комп’ютерних наук та кібернетики**

**Звіт**

з дисципліни «Системне програмування»

по п’ятій лабораторній роботі

на тему: «**lex/yacc-parser**»

студента 3 курсу, групи МІ-31

спеціальності 122 «Комп’ютерні науки»

Рубан Д.С.

Київ –2023

Постановка задачі:

На базі Lex/YACC (або аналога) розробити синтаксичний аналізатор (з генерацією коду або обчисленнями - опціонально) для С або іншої сучасної імперативної (ООП, процедурної, тощо) мови програмування, або для арифметичних виразів (спрощений варіант).

#### Додаткові завдання:

* додавання семантики до AST (реалізацію обчислень)
* генерацію коду на цільовій мові в результаті трансляції
* візуалізацію AST

Виконаємо усі додаткові завдання для арифметичних виразів.

Теоретичні відомості:

#### LR parser:

* Висхідний синтаксичний аналізатор
* LR(k)
* L: зліва направо
* R: будує крайню праву похідну у зворотному порядку
* k: кількість вхідних символів передбачення, що використовуються для прийняття рішень при синтаксичному аналізі, коли (k) опущено, вважається, що воно дорівнює 1.
* Табличний
* Синтаксичні аналізатори LR можуть бути побудовані так, щоб розпізнавати всі конструкції мови програмування, для яких можна написати контекстно-вільну граматику
* Найбільш загальний метод синтаксичного аналізу без бектрекінгу

LALR це Lookahead LR

Lex і Yacc (або їх еквіваленти Flex і Bison) - це інструменти для створення лексичних аналізаторів (сканерів) і синтаксичних аналізаторів. Lex генерує лексичні аналізатори, а Yacc - синтаксичні аналізатори. LALR розшифровується як Look-Ahead Left-Right, і це тип алгоритму синтаксичного аналізу, який використовується в Yacc/Bison.

Ось як працюють синтаксичні аналізатори Lex/Yacc:

Lex генерує лексичний аналізатор, який розбиває вхідний потік на токени. Токени - це найменші одиниці значення в мові. Наприклад, у мові програмування токенами можуть бути ключові слова, ідентифікатори, оператори та літерали.

Yacc генерує синтаксичний аналізатор, який аналізує синтаксичну структуру вхідних даних на основі заданої граматики. Синтаксичний аналіз LALR передбачає перегляд вхідних даних наперед, щоб визначити відповідне правило, яке слід застосувати. Зазвичай це один лексема. LALR – висхідний парсер.

Ви визначаєте граматику вашої мови за допомогою формальної нотації. Ця граматика вказується у вхідних файлах для Yacc/Bison. Зазвичай граматика визначається в термінах продукцій, які описують, як можуть бути побудовані різні частини мови.

Yacc читає специфікацію граматики і генерує синтаксичний аналізатор на C. Згенерований синтаксичний аналізатор використовує таблицю синтаксичного аналізу для визначення дій, які потрібно виконати на основі поточного стану синтаксичного аналізатора і наступної лексеми на вході.

Структура проекту:

Файли:

* **build**

Містить інструкції по запуску програми;

* **my\_calc.h**

містить структури в яких зберігаються лексеми а також «пам’ять» ;

* **my\_calc.l**

містить інструкції щодо лексичного аналізу вводу;

* **my\_calc.y**

містить інструкції для синтаксичного аналізу вводу

* **my\_calc\_solve.c**

видає результат обчислення арифметичного виразу

* **my\_calc\_draw.c**

малює AST

* **my\_calc\_translate.c**

видає переклад на імітацію мови асемблер

Тест



