Міністерство освіти і науки України

Національний університет "Львівська Політехніка"

Кафедра ЕОМ



**Пояснювальна записка**

до курсового проєкту "Системне програмування"

на тему: “Розробка системних програмних модулів та компонент систем програмування”

Індивідуальне завдання

“Розробка транслятора з вхідної мови програмування”

Виконав студент групи КІ-:

Перевірив:

Львів-2024

Завдання на курсовий проєкт

1. Цільова мова транслятора – мова програмування С або асемблер для 32/64 розрядного процесора.
2. Для отримання виконуваного файлу на виході розробленого транслятора скористатися середовищем Microsoft Visual Studio або будь-яким іншим.
3. Мова розробки транслятора: C/C++.
4. Реалізувати графічну оболонку або інтерфейс з командного рядка.
5. На вхід розробленого транслятора має подаватися текстовий файл, написаний на заданій мові програмування.
6. На виході розробленого транслятора мають створюватись такі файли:

*файл з лексемами;*

*файл з повідомленнями про помилки (або про їх відсутність);*

*файл на мові С або асемблера;*

*об’єктний файл;*

*виконуваний файл.*

1. Назва вхідної мови програмування утворюється від першої букви у прізвищі студента та останніх двох цифр номера його варіанту. Саме таке розширення повинні мати текстові файли, написані на цій мові програмування.

**Деталізація завдання на проектування:**

1. В кожному завданні передбачається блок оголошення змінних; змінні зберігають значення цілих чисел і, в залежності від варіанту, можуть бути 16/32 розрядними. За потребою можна реалізувати логічний тип даних.
2. Необхідно реалізувати арифметичні операції – додавання, віднімання, множення, ділення, залишок від ділення; операції порівняння – перевірка на рівність і нерівність, більше і менше; логічні операції – заперечення, “логічне І” і “логічне АБО”.

Пріоритет операцій наступний – круглі дужки (), логічне заперечення, мультиплікативні (множення, ділення, залишок від ділення), адитивні (додавання, віднімання), відношення (більше, менше), перевірка на рівність і нерівність, логічне І, логічне АБО.

1. За допомогою оператора вводу можна зчитати з клавіатури значення змінної; за допомогою оператора виводу можна вивести на екран значення змінної, виразу чи цілої константи.
2. В кожному завданні обов’язковим є оператор присвоєння за допомогою якого можна реалізувати обчислення виразів з використанням заданих операцій і операції круглі дужки (); у якості операндів можуть бути цілі константи, змінні, а також інші вирази.
3. В кожному завданні обов’язковим є оператор типу “блок” (складений оператор), його вигляд має бути таким, як і блок тіла програми.
4. Необхідно реалізувати задані варіантом оператори, синтаксис операторів наведено у таблиці 1.1. Синтаксис вхідної мови має забезпечити реалізацію обчислень лінійних алгоритмів, алгоритмів з розгалуженням і циклічних алгоритмів. Опис формальної мови студент погоджує з викладачем.
5. Оператори можуть бути довільної вкладеності і в будь-якій послідовності.
6. Для перевірки роботи розробленого транслятора, необхідно написати три тестові програми на вхідній мові програмування.

**Деталізований опис власної мови програмування:**

Анотація

Зміст

Вступ

1. Огляд методів та способів проєктування трансляторів
2. Формальний опис вхідної мови програмування
   1. Деталізований опис вхідної мови в термінах розширеної нотації Бекуса-Наура.
3. Розробка транслятора з вхідної мови програмування
   1. Вибір технології програмування.
   2. Проектування таблиць транслятора та вибір структур даних.
   3. Розробка лексичного аналізатора.
      1. Розробка алгоритму роботи лексичного аналізатора.
      2. Опис програми реалізації лексичного аналізатора.
   4. Розробка синтаксичного та семантичного аналізатора.
      1. Розробка дерева граматичного розбору.
      2. Розробка алгоритму роботи синтаксичного і семантичного аналізатора.
      3. Опис програми реалізації синтаксичного та семантичного аналізатора.
   5. Розробка генератора коду.
      1. Розробка алгоритму роботи генератора коду.
      2. Опис програми реалізації генератора коду.
4. Налагодження та тестування розробленого транслятора
   1. Опис інтерфейсу та інструкції користувачу.
   2. Виявлення лексичних і синтаксичних помилок.
   3. Перевірка роботи транслятора за допомогою тестових задач.

Висновки

Список літературних джерел

Додатки

А. Таблиці лексем для тестових прикладів

Б. С код (або код на асемблері), отриманий на виході транслятора для тестових прикладів;

В. Абстрактне синтаксичне дерево для тестових прикладів (за наявності).

Г. Документований текст програмних модулів (лістинги)