

ТЕХНИЧЕСКИЕ НАУКИ

Новичкова Мария Игоревна

студентка

Трофимов Юрий Александрович

старший преподаватель

ФГБОУ ВПО «Пензенский государственный университет»

г. Пенза, Пензенская область

РАЗРАБОТКА ТРАНСЛЯТОРА ЯЗЫКА ПРОГРАММИРОВАНИЯ ВЫСОКОГО УРОВНЯ

Аннотация: в данной статье дается описание процесса трансляции программы, написанной на одном из языков программирования высокого уровня, в исполняемый код. Рассматривается процесс разработки транслятора компиляторного типа на языке C++ и демонстрируются его основные возможности.

Ключевые слова: транслятор, компилятор, язык программирования, алфавит, грамматика.

Написание программ для ЭВМ прошло большой путь развития от программирования в машинных кодах до использования современных сред программирования. В настоящее время исходный текст программы пишется на одном из языков программирования высокого уровня. Однако процессоры по-прежнему понимают только машинный язык. Поэтому для любой среды разработки необходим транслятор – программа, переводящая команды исходного языка программирования в машинные коды процессора.

Не секрет, что все трансляторы делятся на два типа – компиляторы и интерпретаторы. В своей работе мы остановили свой выбор на трансляторе компиляторного типа, так как предполагаем не только переводить исходную программу в машинный код, но и исполнять ее.

В целом процесс компиляции представляет собой шесть последовательных этапов: лексический анализ, синтаксический анализ, контекстный анализ, генерация промежуточного представления, оптимизация, генерация кода. Обобщенный механизм компиляции представлен на рисунке 1.



Рис. 1. Этапы компиляции

Нами была сделана попытка реализовать транслятор компиляторного типа прямым методом, т.е. исходный текст программы транслировался напрямую в ассемблерный код. Такой подход позволил уже на ранних стадиях разработки получить работающий простой транслятор с минимальным набором функций. В дальнейшем велась работа по усложнению транслятора и расширению его возможностей.

В результате был разработан транслятор компиляторного типа, входным языком которого является специально созданный язык. Алфавит языка состоит из строчных латинских букв, цифр и символов {+, -, *, /, (,), =, #, >, <, &, |, ~, !, ,,}, а основные конструкции языка представлены в таблице 1.

Таблица 1

Основные команды языка

<i>Символ</i>	<i>Описание команды</i>
v	Объявление переменных
p	Вывод на экран
i	Оператор ветвления
l	
w	Цикл с предусловием
e	Окончание блока команд

Компилятор позволяет обрабатывать различные арифметические выражения, состоящие из четырех основных операций – сложение, вычитание, умножение и деление. Также реализована обработка различных логических выражений, содержащих конъюнкцию, дизъюнкцию и отрицание. Кроме того, компилятор обрабатывает основные алгоритмические конструкции: присваивание, ветвление и цикл. Поддерживается обработка вложенных конструкций любого уровня. Таким образом, с помощью созданного языка программирования и разработанного транслятора можно реализовывать алгоритмы достаточной сложности. Эффективность работы транслятора проверена на числовых алгоритмах достаточной сложности.

Для удобства работы с транслятором разработана программная оболочка. Оболочка реализована в виде приложения Windows Forms в среде Microsoft Visual Studio 2010.

Рассмотрим пример работы транслятора при решении следующей задачи: *найти сумму квадратов первых N натуральных чисел*. Результат работы алгоритма в нашей среде представлен на рисунке 2.

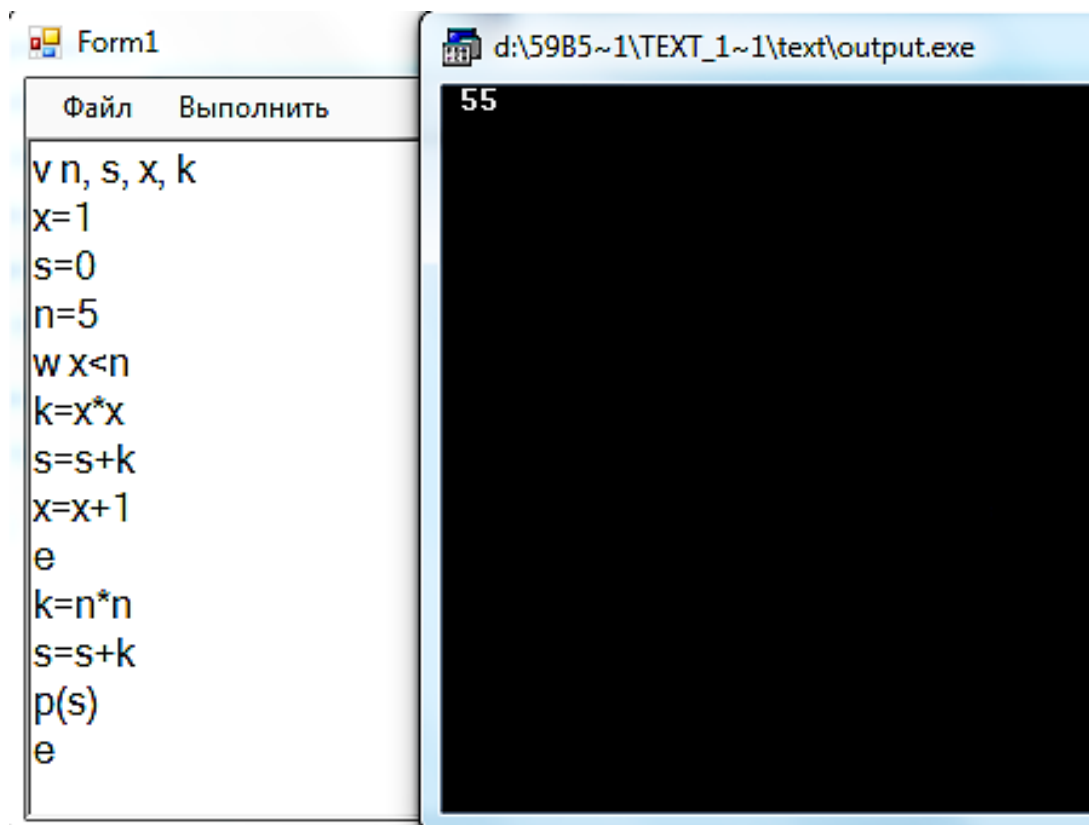


Рис. 2. Пример решения задачи в нашей среде

Для сравнения эта же задача решена с использованием среды Microsoft Visual Studio 2010 и языка C++ (рис. 3).

```
#include "stdafx.h"
#include <iostream>

using namespace std;

int main()
{
    int n=5,s=0;
    for(int i=1;i<=n;i++)
        s+=i*i;
    cout<<s<<endl;
    system("pause");
    return 0;
}
```

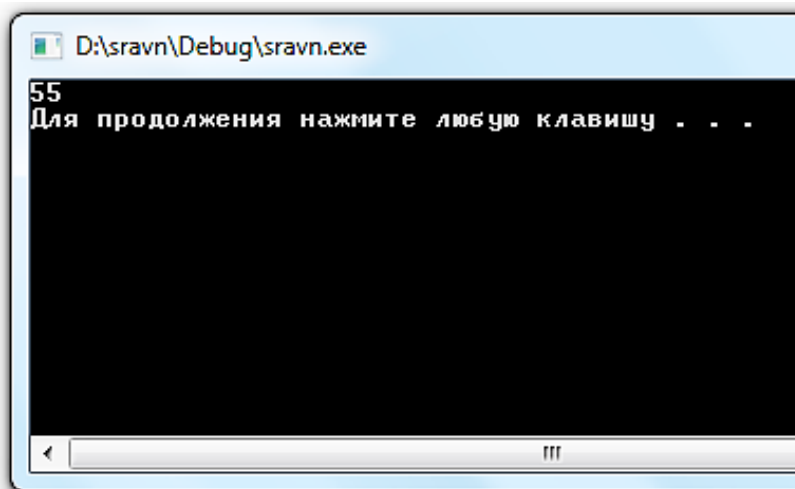


Рис. 3. Решение задачи в среде Microsoft Visual Studio 2010

Разработанная программа может использоваться в процессе изучения студентами технических специальностей курса «Системы программирования», а также для совершенствования навыков программирования. Кроме того, разработка будет полезна при знакомстве студентов с такими понятиями как язык программирования, алфавит языка, лексема, лексический анализ, синтаксический анализ и пр.

Список литературы

1. Ахо, Лам, Сети, Ульман. Компиляторы: принципы, технологии и инструментарий, 2-е изд.: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2008. – 1184 с.: ил.
2. Серебряков В.А., Галочкин М.П. Основы конструирования компиляторов [Электронный ресурс]. – Режим доступа: <http://citforum.ru/programming/theory/serebryakov/>
3. Джек Креншоу. Давайте создадим компилятор [Электронный ресурс]. – Режим доступа: http://programming-lang.com/ru/comp_programming/krenshou/0/