

Лабораторная работа №3.  
**Исследование алгоритмов разбора арифметических выражений.**

**Цель работы:** Исследование структуры арифметических выражений и алгоритмов их разбора и вычислений.

**Содержание работы:** В процессе выполнения лабораторной работы должны быть изучены структуры арифметических выражений и алгоритмы, реализующие синтаксический разбор и вычисление этих выражений.

Арифметические выражения являются важным элементом большинства языков программирования, в том числе и ассемблерного типа. Поэтому их трансляция и вычисление входит в функции компилятора. Были разработаны системы, которые предоставляли возможность пользователям непосредственно записывать свои формулы в удобном для них виде и которые транслировали эти формулы в программу в машинном коде. Наиболее распространенной формой записи арифметических выражений является алгебраическая (скобочная) форма. Такую форму называют инфиксной. Инфиксная форма учитывает старшинство арифметических операций и возможность его изменения с помощью скобок. Например:

$$(a \times b + c \times d) / (a - d) + b \times c \quad (1)$$

В процессе трансляции таких выражений компилятор должен выстроить правильную последовательность выполнения арифметических операций с учетом их старшинства и скобок. Это достигается преобразованием выражения, записанного в инфиксной форме, в так называемую постфиксную или бесскобочную форму. (Польская запись). Так, выражение (1) в постфиксной форме имеет вид:

$$ab \times cd \times + ad - / bc \times + \quad (2)$$

При выполнении преобразования из (1) в (2) используется стек. Исполняющая программа читает постфиксную запись слева направо. При чтении операнда он засылается в стек, при чтении операции она применяется к двум его верхним элементам.

Бауэр и Заммельсон предложили эффективный метод преобразования арифметических выражений из инфиксной в постфиксную форму. Этот метод использует два стека. Один стек используется непосредственно при трансляции выражения, второй стек – во время выполнения. Стек транслятора обозначается Т, стек программы – Е. Транслятор прочитывает выражение один раз слева направо и вырабатывает последовательность команд двух видов:

- $K_I$ , где I – идентификатор, команда выбора числа I и заслать его в стек Е.
- $K_\xi$ , где  $\xi$  - операция, команда выбора двух верхних операндов из стека Е, выполнения над ними операции  $\xi$  и занесения результата в вершину Е.

Когда при просмотре входной строки считываемый символ является операндом I, выдается команда  $K_I$  и транслятор переходит к следующему символу. Когда считываемый символ является операцией  $\xi$ , производится одно из некоторых фиксированных действий. Как правило, либо  $\xi$  засылается в стек Т, либо выдается команда  $K_\xi$ . Действия, которые должен выполнить компилятор, задаются специальной таблицей переходов. В этой таблице каждой операции языка

соответствует строка и столбец. Элементами таблицы являются директивы транслятору. Возможны четыре действия транслятора после прочтения операции  $\xi$ . (Операция в вершине стека  $T$  обозначается как  $\eta$ .)

- I: Заслать  $\xi$  в стек  $T$  и читать следующий символ.
- II: Выдать команду  $K_\eta$ , заслать  $\xi$  в стек  $T$  и читать следующий символ.
- III: Читать из  $T$  и читать следующий символ (используется для удаления скобок).
- IV: Выдать команду  $K_\eta$ , читать из стека  $T$  и повторить с тем же входным символом.

Имеются еще две специальные команды: «Конец процесса» и «Ошибка», которые приказывают транслятору остановиться и выдать соответствующую команду. Транслятор выбирает элементы таблицы, используя в качестве индексов  $\eta$  - верхнюю операцию стека  $T$  и  $\xi$  - операцию, прочитанную последней из входной строки. Для простых арифметических выражений таблица переходов имеет вид:

$\xi$  - входной символ;  
 $\eta$  - вершина стека  $T$ .  $\Lambda$  - пустая строка, или пробел. Если стек  $T$  пуст, или пуста входная последовательность, то считается, что прочитано  $\Lambda$ .

	$\Lambda$	(	+	-	$\times$	/	)
$\Lambda$	Конец	I	I	I	I	I	Ошибка
(	Ошибка	I	I	I	I	I	III
+	IV	I	II	II	I	I	IV
-	IV	I	II	II	I	I	IV
$\times$	IV	I	IV	IV	II	II	IV
/	IV	I	IV	IV	II	II	IV

При преобразовании (1) в (2) использование этой таблицы переходов дает следующую последовательность шагов:

$T$	Прочитанный символ	Действие	Выдать команду
$\Lambda$	(	I	
(	a		$K_a$
(	$\times$	I	
( $\times$	b		$K_b$
( $\times$	+	IV	$K_\times$
(	Повторить	I	
(+	c		$K_c$
(+	$\times$	I	
( $\times$	d		$K_d$
( $\times$	)	IV	$K_\times$
(+	Повторить	IV	$K_+$

(	Повторить	III	
Λ	/	I	
/	(	I	
/(	a		K <sub>a</sub>
/(	–	I	
/(–	d		K <sub>d</sub>
/(–	)	IV	K <sub>–</sub>
/(	Повторить	III	
/	+	IV	K <sub>/</sub>
Λ	Повторить	I	
+	b		K <sub>b</sub>
+	×	I	
+×	c		K <sub>c</sub>
+×	Λ	IV	K <sub>×</sub>
+	Λ	IV	K <sub>+</sub>
Λ	Λ	Конец	

Полученная последовательность команд имеет вид:

K<sub>a</sub>, K<sub>b</sub>, K<sub>×</sub>, K<sub>c</sub>, K<sub>d</sub>, K<sub>×</sub>, K<sub>+</sub>, K<sub>a</sub>, K<sub>d</sub>, K<sub>–</sub>, K<sub>/</sub>, K<sub>b</sub>, K<sub>c</sub>, K<sub>×</sub>, K<sub>+</sub>.

Если опустить все K, то получится постфиксная запись (2).

### **Порядок выполнения лабораторной работы.**

Для выполнения настоящей лабораторной работы необходимо:

1. Дополнить таблицу переходов операцией возведения в степень;
2. Разработать приложение, реализующее вычисление арифметических выражений, задаваемых в виде символьной строки, например:  $(3.14+6.28)/2$ , и т. п. Приложение должно включать в себя блок «сканера» (выделение чисел и операций), и блок, реализующий алгоритм Бауэра – Заммельсона и вычисление преобразованного выражения.

### **Содержание отчета.**

Отчет по лабораторной работе должен включать в себя:

1. Разработанную таблицу переходов;
2. Последовательность шагов преобразования арифметического выражения в постфиксную форму для контрольного примера, заданного преподавателем;
3. Описание структур данных, использованных при разработке приложения;
4. Краткий листинг разработанного приложения с комментариями (В части реализации разработанных алгоритмов).

### **Литература:**

1. Л. Бек. «Введение в системное программирование». М.: «Мир», 1988 г.
2. Т. Пратт. «Языки программирования: разработка и реализация». М.: «Мир», 1979 г.