

Второй этап – синтаксический анализатор или парсер. Его задача – проверка структуры кода, и создание синтаксического дерева разбора.

Третий этап – семантический анализ. Он необходим для анализа семантики, то есть смыслового значения программы, и проверки кода на соответствие правилам языка.

Во время семантического анализа происходит проверка типов данных, правильности использования операторов и функций, а также обнаружение потенциальных ошибок, связанных с семантикой, преобразование выражений к польской нотации. Для этого анализатор использует информацию, полученную на предыдущих этапах.

Основной отличительной особенностью моего ЯП является использование стековой модели памяти, что означает что все данные хранятся в стеке. Было реализовано соглашение о вызовах `stdcall`.

Стековая модель памяти - это организация памяти компьютерной системы, где данные хранятся в виде стека. Стек представляет собой структуру данных, в которой последний добавленный элемент будет первым удаленным (принцип LIFO – «последним вошел, первым вышел»). Это позволяет эффективно работать с функциями и подпрограммами, сохраняя и восстанавливая контекст выполнения.

На последнем этапе происходит генераторами кода в язык ассемблера MASM. Далее программа на MASM ассемблируется, передается компоновщику, и после этого создается исполняемый файл, который может быть запущен, и код будет выполнен.

УДК 004.021

Студ. Д.И. Подшиваленко

Науч. рук. ст. преп. А.С. Наркевич
(кафедра программной инженерии, БГТУ)

ВЫЧИСЛЕНИЕ ВЫРАЖЕНИЙ В ПОЛИЗ

В математике существует несколько способов представления выражений: инфиксный, постфиксный, префиксный. Самой распространенной формой записи является инфиксная нотация. В этой форме операторы располагаются между операндами. Инфиксная нотация интуитивно понятна и привычна для большинства людей.

В префиксной нотации, операторы располагаются перед операндами, к которым они применяются.

Обратная польская запись, известная как постфиксная нотация, представляет выражение, в котором операторы следуют после своих операндов. Постфиксная форма имеет ряд преимуществ.

1) В ПОЛИЗ не нужны скобки для определения порядка выполнения операций.

2) Упрощенная обработка: ПОЛИЗ облегчает вычисление выражения. Операции выполняются непосредственно над операндами, без необходимости учета приоритета операций.

3) Выражение в обратной польской записи удобно вычислять с помощью стекового автомата.

4) Широко используется для вычислений и обработки выражений в компиляторах.

Для преобразования в ПОЛИЗ используется алгоритм Дейкстры.

Для вычисления выражения в обратной польской нотации используется стековая машина.

В данной работе была реализована функция для построения обратной польской записи для целочисленного выражения в инфиксной форме. Реализована функция для вычисления этого выражения.

Результат работы программы представлен на рисунке ниже.

```
Введите выражение
10 + 30*5 + (15 / 3) - 20 + (10 / (2 + 3))
Верная скобочная последовательность
10 30 5 * + 15 3 / + 20 - 10 2 3 + / +
147
```

Полученный опыт позволил расширить знания в области алгоритмов и структур данных. Эти знания пригодятся в дальнейшем для реализации собственного компилятора.

УДК 004.588

Студ. М.И. Катков, В.Р. Паскин
Науч. рук. ст. преп. А.С. Наркевич
(кафедра программной инженерии, БГТУ)

СОЗДАНИЕ ПРОЕКТА «БПИ-2JUMP»

Проект «БПИ-2jump» был разработан студентами первого курса в рамках изучения предмета «Основы программной инженерии».

Цель: создание проекта на языке C++ используя методологию Agile-Scrum и канбан-доску для эффективного распределения работы.

Задачи проекта:

- выбор темы проекта;
- изучение основ методологии Agile-Scrum;
- определение основных составляющих;
- распределение задач по созданию проекта в команде для каждого участника;