2. Разработка компилятора

На данном этапе работы необходимо в соответствии с требованиями, указанными в техническом задании разработать структуру компилятора, а также разработать соответствующий анализатор для каждого из этапов компиляции.

2.1. Разработка структуры компилятора

Для обеспечения трансляции исходного кода в исполняемый модуль разработана структура компилятора, представляющая собой диаграмму последовательности. Диаграмма последовательности представлена на  рисунке 1.

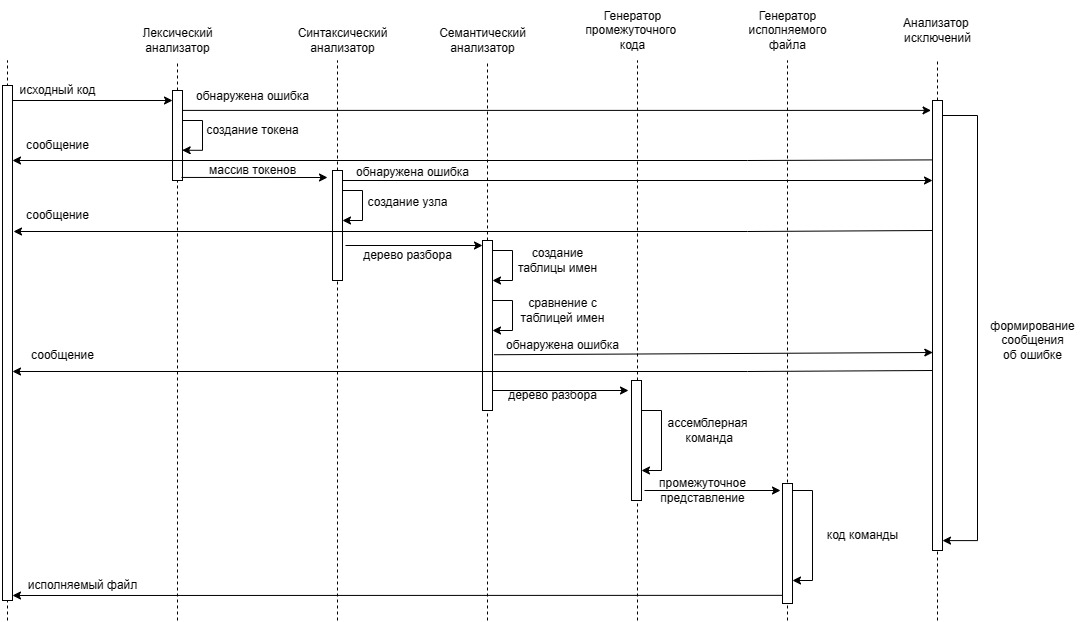


Рисунок 1 – Диаграмма последовательности

Инициализатором трансляции исходного кода является компилятор, передающий управление лексическому анализатору.

В свою очередь лексический анализатор выполняет разбиение текста программы на слова, определение типа введенного слова и создание элемента, содержащего само слово и его тип. Такой элемент называется токеном. Кроме того, анализатор проверяет корректность введенных данных. В случае возникновения некорректных данных управление передается анализатору исключений. Если же введенные данные корректны, то управление передается синтаксическому анализатору, инициализируя при это действие передачи массива токенов.

Анализатор исключений инициализирует действие о формировании сообщения об ошибки и передает управление компилятору.

При получении управления синтаксический анализатор проверяет правильность полученных конструкций при помощи заданной грамматики, последовательно перебирая массив токенов. В результате выполнения анализа формируется дерево разбора. В случае несоответствия полученного токена с тем, который описан в грамматике управление передается анализатору исключения. При отсутствии ошибок управление передается семантическому анализатору, инициализируя при этом действие передачи дерева разбор.

Что касается семантического анализатора, то он проверяет корректность программы, но с той точки зрения были ли объявлены переменны до того, как их начали использовать. Для выполнения этого этапа происходит инициализация действия создания таблицы имен, при котором происходит обход синтаксического дерева и записывания переменных, имеющих определение, в таблицу. Далее инициализируется действие сравнение с таблицей имен. В случае обнаружения неинициализированной переменной управление передается анализатору исключений, иначе – генератору промежуточного кода.

Генератор промежуточного кода, при получении управления, формирует ассемблерные команды, построенные на основе синтаксического дерева, чью инициализацию действия передачи выполнил семантический анализатор. В свою очередь генератор является инициализатором действия передачи промежуточного представления генератору исполняемого файла.

Генератор исполняемого файла инициализирует действие перевода каждой ассемблерной команды в 16-ричный код и запись значений в файл. Далее управление передается компилятору.

Компилятор при повторном получении управлении завершает свою работу.