Сервис CLI: разбор ввода пользователя и построения цепочки команд

Лексический анализ

*Letter* = **A** | … | **Z** | **a** | … | **z**

*Digit* = **0** | … | **9**

*WordCharacter* = *Letter* | *Digit* | **'\_'** | **'-'** | **'.'** | **','**

*Word* = *Letter* {*WordCharacter*}

*LetterChar* = **любой символ кроме '"'**

*String* = **'"'** {*LetterChar |* **'\"'** | **'\\'**} **'"'**

Синтаксический анализ

*Command* = **Word** {**Word** | **String**}

Правила разбора

1) При помощи синтаксического анализа мы разбираем ввод пользователя.

2) Для всех команд, определенных в системе мы строим таблицу для распознавания команд. Каждый элемент этой таблицы представляет список пар: слово, длина префикса, уникально идентифицирующего данное слово (относительно предыдущих слов). Эта таблица сортируется по количеству слов в элементе в порядке убывания (при этом порядок элементов с одинаковым количеством слов не важен).

3) После разбора команды при помощи синтаксического анализа мы ищем для этой команды соответствующий элемент в таблице. При поиске в таблице мы ищем максимально возможное соответствие. Так, например, для ввода пользователя **"interface range XXX"** будет найдено соответствие **"interface range"**, а не **"interface"** (при наличии команд **"interface"** и **"interface range"** в таблице). При этом мы учитываем тот факт, что при сравнении соответствующих слов (из элемента и из команды) достаточно сравнивать их префиксы. Длиной сравниваемых префиксов будет соответствующая длина префикса, уникально идентифицирующего данное слово; эта длина идет вместе с каждым словом для каждого элемента из таблицы для распознавания команд.

4) Остаток команды после ее распознавания считается аргументами команды и передается команде при ее выполнении. Команда может распоряжаться этими аргументами так, как считает нужным.

5) Если для какой-либо команды будет не найден элемент в таблице для распознавания команд, то мы будем генерировать ошибку разбора.

6) После разбора и распознавания команды мы генерируем код для ее вызова. Этот код помимо вызова самой команды с некоторыми аргументами генерирует ряд служебных вызовов, уведомляющих о завершении выполнения команды. Этот код называется контекстом выполнения команды.

Таблица синтаксического анализа

Правила:

1) *Command* = **Word** *Args*

2) *Args* = **Word** *Args* | **String** *Args* | **''**

Множества FIRST и FOLLOW:

FIRST(Command) = {**Word**}

FOLLOW(Command) = {**'$'**}

FIRST(Args) = {**Word**, **String**, **'$'**}

FOLLOW(Args) = {**'$'**}

Таблица:

M[*Command*, **Word**] = 1

M[*Args*, **Word**] = 2(1)

M[*Args*, **String**] = 2(2)

M[*Args*, **'$'**] = 2(3)

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Word** | **String** | **'$'** |
| *Command* | 1 |  |  |
| *Args* | 2(1) | 2(2) | 2(3) |