

Язык описания станций

Программа состоит из определения лексем, последней должна быть определена лексема с именем goal.

Определение лексемы:

```
имя := тело;
```

Тело goal имеет специальную форму:

```
goal := имя_лексемы || имя_лексемы || имя_лексемы;
```

Имеется также специальная конструкция

```
allowed := "string";
```

Эта конструкция устанавливает допустимые символы с телефонных номерах. По умолчанию разрешены цифры. Проверка производится только при вводе с помощью директивы format.

Эти лексемы будут по очереди применяться для разбора входного потока. Если ни одна лексема не подошла, то из входного потока выбрасывается первый символ (ошибочный) и все повторяется.

Тело лексемы состоит из двух частей (обе могут быть пропущены).

```
<простое-выполнение>  init <инициализация>  
11111111111111111111  22222222222222222222
```

действия, указанные <инициализация>, выполняются каждый раз перед применением любой лексемы из списка goal.

<Простое-выполнение> выполняется при применении лексемы.

Операторы в теле лексемы:

```
{ операторы }
```

блок операторов

```
if условие then оператор-если-да else оператор-если-нет  
? условие : оператор-если-да ^ оператор-если-нет
```

условный оператор

имя-лексемы

применить лексему. Если лексема неприменима,
то прекратить применение текущей лексемы.

имя-лексемы = выражение

оператор присваивания. Задаёт значение лексемы.

имя-лексемыА = имя-лексемыБ()

вызов лексемы с последующим присваиванием.

Сначала вызовется лексема Б, выполнится и ее значение будет присвоено лексеме А.

простая-строка

сравнить входной поток со строкой. Если не совпадают, то прекратить применение текущей лексемы.

format форматная-строка

сравнить входной поток с форматной строкой. Следующие символы имеют специальное значение:

- e - Extension
- l - Trunk
- M - Month
- D - Day
- Y - Year
- H - Hour
- N - Minute
- S - Second
- h - Duration Hours
- m - Duration Minutes
- s - Duration Seconds
- i - Input
- o - Output
- a - Account
- R - Ring Duration Minutes
- r - Ring Duration Seconds
- T - Transferred (если не пробел, то Transferred=1)
- A - AM/PM
- p - Dialed (берутся только символы из allowed)
- t - Type:
 - 'I': incoming
 - 'O': outgoing
 - 'T': transferred
 - 'U': unanswered
- ? - любой символ
- # - кватирование следующего символа

Если при вводе телефонного номера через форматную строку получается пустой номер (после удаления неразрешенных символов) то Incoming=1.

Пробелы внутри чисел (например, в месяце) игнорируются.

debug строка

Вывести отладочную строку

any(выражение)

Пропустить столько символов, сколько указано в <выражении>

spaces

Пропустить пробелы и табуляции

SKIP

Считать входной поток разобранным до текущего положения курсора анализатора.

ERROR

Прервать применение лексемы, т.к. она не соответствует входному потоку.

WRITE

Записать запись в базу данных

NEXT

Передвинуть курсор анализатора на следующий символ.

Операторы

простая-строка

форматная-строка

any()

spaces()

сами передвигают курсор.

//-----

В выражениях можно применять (в порядке увеличения приоритета):

||

&&

== !=

>= <= > <

+ -

* / %

- + ! (все унарные)

Константы и атомы:

@ - символ, на который указывает курсор анализатора

все C-шные константы (включая строки)

thisyear - текущий год

thismonth - текущий месяц

thisday - текущий день

Встроенные функции:

num(N) – разобрать число из входного потока в N цифр.

пробелы не допускаются. Если N==0, то число

продолжается пока встречаются цифры, но хотя бы одна

цифра должна присутствовать.

num2(N, M)

разобрать число из входного потока от N до M цифр.

пробелы не допускаются.

strlen(S) – длина строки

strcmp(S1, S2) – сравнить строки

Возвращаемое число	Обозначает
-1	S1 < S2
0	S1 = S2
1	S1 > S2

strstr(S1, S2) – 1 если подстрока S2 встречается в S1

strchr(S, C) – индекс символа (1..n) C в S

substr(S,M1,M2) – подстрока S[M1..M2] M1 и M2 меняются (1..n)

можно указать $M2=-1$, тогда до конца строки
 если $M2 < M1$ то пустая строка
 $SUBSTR(S,1,1)$ - строка из первого символа
 $instr(N)$ – строка из входного потока длиной в N символов.
 $atoi(S)$ – аналогично Сишной функции $atoi()$
 $strchk(S1, S2)$ – удалить из строки $S1$ все символы, отсутствующие
 в $S2$.
 $instruntil(S)$ – строка из входного потока до символа,
 встречающегося в S .

Да, чуть не забыл сказать, к строкам можно применять операции присваивания, сравнения
 и сложения (конкатенации)

=====

По умолчанию определены и используются следующие лексемы:

Input := Input=num(0);

Номер входной линии

Output := Output=num(0);

Номер выходной линии

InputTrunk := init InputTrunk=0;

Является ли входная линия trunk'ом
 Обратите внимание, что значение этой лексемы сбрасывается
 в 0 перед каждым анализом

OutputTrunk := init OutputTrunk=0;

Является ли выходная линия trunk'ом
 Обратите внимание, что значение этой лексемы сбрасывается
 в 0 перед каждым анализом

Extension := Extension=num(0);

(значение этой лексемы при формировании записи не используется)
 Номер внутренней линии. Используется только для
 промежуточного хранения данных, перед оператором WRITE необходимо
 перенести эту информацию в переменные Input/Output.
 Для этого можно воспользоваться функцией SetTrunks (см.ниже)

Trunk := Trunk=num(0);

(значение этой лексемы при формировании записи не используется)
 Номер городской линии. Используется только для
 промежуточного хранения данных, необходимо перенести
 эту информацию в переменные Input/Output.
 Для этого можно воспользоваться функцией SetTrunks (см.ниже)

Incoming := ? @ == 'I' : Incoming=1 ^ Incoming=0 NEXT
 SetTrunks;

Тип звонка: если != 0, то входящий.

SetTrunks := ? Incoming
 : { InputTrunk=1 Input=Trunk Output=Extension }

^ { OutputTrunk=1 Output=Trunk Input=Extension };

(значение этой лексемы при формировании записи не используется)
Функция для переноса информации из Extension/Trunk
в Input/Output и InputTrunk/OutputTrunk.

Day := Day = num(0) ? Day > 31 || Day < 1 : ERROR;

Дата звонка: день

Month := Month = num(0) ? Month > 12 || Month < 1 : ERROR;

Дата звонка: месяц

Year := Year = num(0) ? Year < 1900 : Year=Year+1900 init Year=thisyear;

Дата звонка: год.

StartHour := StartHour = num(0) ? StartHour > 24 : ERROR;

Время начала звонка: час

StartMinute := StartMinute = num(0) ? StartMinute > 60 : ERROR;

Время начала звонка: минута
При формировании записи значение этой переменной
нормализуется, т.е. если > 60:
AnswerMinute += AnswerSecond / 60
AnswerSecond += AnswerSecond % 60

StartSecond := StartSecond = num(0) ? StartSecond > 60 : ERROR;

Время начала звонка: секунда
При формировании записи значение этой переменной
нормализуется.

AnswerHour := AnswerHour = num(0) ? AnswerHour > 24 : ERROR;

Время ответа (поднятия трубки): часы

AnswerMinute := AnswerMinute = num(0) ? AnswerMinute > 60 : ERROR;

Время ответа (поднятия трубки): минуты
При формировании записи значение этой переменной
нормализуется.

AnswerSecond := AnswerSecond = num(0) ? AnswerSecond > 60 : ERROR;

Время ответа (поднятия трубки): секунды
При формировании записи значение этой переменной
нормализуется.

EndHour := EndHour = num(0) ? EndHour > 24 : ERROR;

Время окончания звонка: часы

EndMinute := EndMinute = num(0) ? EndMinute > 60 : ERROR;

Время окончания звонка: минуты
При формировании записи значение этой переменной
нормализуется.

EndSecond := EndSecond = num(0) ? EndSecond > 60 : ERROR;

Время окончания звонка: секунды
При формировании записи значение этой переменной
нормализуется.

DurHour := DurHour = num(0) ? DurHour > 24 : ERROR;

Длительность звонка: часы

DurMinute := DurMinute = num(0) ? DurMinute > 60 : ERROR;

Длительность звонка: минуты
При формировании записи значение этой переменной
нормализуется.

DurSecond := DurSecond = num(0) ? DurSecond > 60 : ERROR;

Длительность звонка: секунды
При формировании записи значение этой переменной
нормализуется.

RdurHour := RdurHour = num(0) ? RdurHour > 60 : ERROR;

Длительность звонков телефона до поднятия трубки: часы.

RdurMinute := RdurMinute = num(0) ? RdurMinute > 60 : ERROR;

Длительность звонков телефона до поднятия трубки: минуты
При формировании записи значение этой переменной
нормализуется.

RdurSecond := RdurSecond = num(0) ? RdurSecond > 60 : ERROR;

Длительность звонков телефона до поднятия трубки: минуты
При формировании записи значение этой переменной
нормализуется.

Authorization := Authorization = instr(0);

Код доступа.

Account := Account = instr(0);

Код счёта.

ExtraLong := ? @ != ' ' : ExtraLong=1 ^ ExtraLong=0 NEXT;

Флаг: очень длинный разговор

Redialed := ? @ == '*' : Redialed=1 ^ Redialed=0 NEXT;

Флаг: redial

Transferred := ? strchr("T*", @) : Transferred=1 ^ Transferred=0 NEXT;

Флаг: transferred

Unanswered := ? @ == '*' : Unanswered=1 NEXT;

Флаг: на входящий звонок никто не ответил

Dialed := Dialed = instr(0);

Номер телефона.

CallerId := CallerId = instr(0);

Идентификатор звонящего (АОН).

Cost := Cost = num(0);

Стоимость разговора, подсчитанная оператором связи.

Pulses := Pulses = num(0);

Счётчик тиков для тарификации (Meter Pulse Count).

BytesSent := BytesSent = num(0);

Количество переданных байтов.

BytesReceived := BytesReceived = num(0);

Количество принятых байтов.

SerialNo:= SerialNo = num(0);

Номер записи для объединения фрагментов разговора. Некоторые АТС выдают уникальный номер записи для тех разговоров, которые описываются несколькими записями. Каждая запись, относящаяся к одному разговору, имеет один и тот же SerialNo.

allowed := "0123456789";

Эта лексема содержит символы, разрешенные в номере телефона.
Проверка производится только при вводе с помощью директивы
format.