

В интерпретаторе выделяют три части:

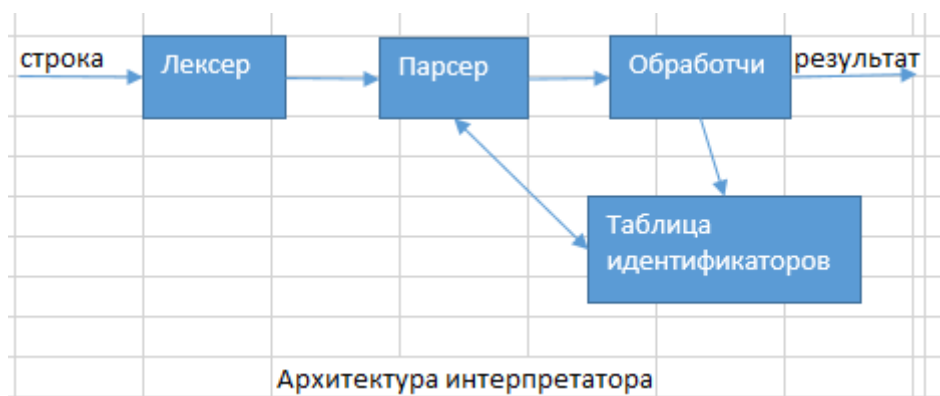
**Лексический анализатор** (лексер, сканер) - просматривает *литеры* исходной программы слева направо и строит токены.

**Синтаксический анализатор** (парсер) - Задача парсера — провести синтаксический разбор выражения. Результатом синтаксического разбора в общем случае является абстрактное синтаксическое дерево (AST). В нашем случае, по-видимому, все проще будет (дерево из одного пути – нужно проверить).

**Обработчик** (виртуальная машина)

Вход: абстрактное синтаксическое дерево (AST).

Выход: результат вычисления



Парсер, когда встречается символ доллара (вне одинарных кавычках) считывает переменную (после знака доллара), осуществляет поиск этой переменной в таблице подстановки, если находит, то заменяет на найденное значение, если не находит, то заменяет на «пусто» (по умолчанию считаем, что не инициализированные переменные имеют значение пусто).

Обработчик, встретив токен, после которого идет знак равно, находит в таблице идентификаторов данный токен (если его нет, то добавляет) и в качестве значения приписывает токен с правой части от знака равенства. **Здесь вопрос: в bash присвоение значения переменной осуществляется только в простом случае, когда строка состоит из одной команды присвоения (если есть пайп, то у переменной остается старое значение).**

На рисунок нужно еще добавить таблицу команд (cat, echo, wc, pwd, exit). В таблице прописано какие параметры имеет данная команда. В программе нужно будет реализовать данные команды (а может и не нужно, вызывать уже готовые).

На рисунок добавить буфер для результата последней выполненной команды.

Обработчик, встретив токен, после которого не знак равно, ищет его в таблице команд, если находит, то смотрит наличие параметров (и наличие параметра в буфере) и выполняет данную команду (при этом, запоминает результат команды в буфер). Если не находит выполняет как внешнюю команду. (здесь считаем, что результат пустой?).

Не совсем понял такой момент в bash

```
>echo 123 | echo
```

Выдает пусто. По-видимому, из-за того, что у данной команды вывод осуществляется только на экран (команда возвращает void) (в отличие от команды "wc"). Т.е. результат вычисляется, если return что-то возвращает.

Алгоритм определения токена (в нашем случае):

Если в строке встретилась кавычка, то программа ожидает ввода пока не встретит такую же кавычку. Т.е. в строке, передаваемой лексеру, должно быть четное количество одинарных и двойных кавычек.

Два подряд идущих пробела заменяем на один пробел (только, если он не в кавычках).

Спецсимволы: «пробел», «=», «|», но только, если они не в кавычках.