

Структура данных

```
struct cmd_inf {
    char ** argv; // список из имени команды и аргументов
    char *infile; // переназначенный файл стандартного ввода
    char *outfile; // переназначенный файл стандартного вывода
    int backgrnd; // =1, если команда подлежит выполнению в фоновом режиме
    cmd_inf* psubcmd; // команды для запуска в дочернем shell
    cmd_inf* pipe; // следующая команда после "|"
    cmd_inf* next; // следующая после ";", (или после "&")
};
```

Синтаксис

```
<команда_shell> ::= <список_команд>
<список_команд> ::= <конвейер> { [ один из &, &&, ||, ; ] <конвейер> } [ &, ; ]
<конвейер> ::= <команда> { | <команда> }
<команда> ::= <простая_команда> |
                ((<список_команд>) [ <имя_файла> ] [ [ один из >, >> ]
                <имя_файла> ]
<простая_команда> ::= <имя_файла> { <аргумент> }
                [ <имя_файла> ] [ [ один из >, >> ]
                <имя_файла> ]
```

Лексические тесты

```
echo Hello, world!           # корректная команда. 2 аргумента команды echo
echo Hello , world !        # корректная команда. 4 аргумента команды echo. не должно быть
                              лишних пробелов.
echo "! Hello, world! !"    # корректная команда. 1 аргумент команды echo. кавычки не выводятся.
                              пробелы между аргументами
echo "! Hello, world! !     # дисбаланс кавычек
echo "! Hello, " world! !"  # дисбаланс кавычек
echo "! Hello, \" world! !" # корректная команда. экранирование внутренней кавычки
echo \\ Hello,\\ world!     # корректная команда. экранирование пробелов => все запишется как 1
                              аргумент команды echo
echo -e $USER\\n $EUID\\n $HOME\\n $SHELL\\n $PATH # корректная команды. замена переменных
                              + экранирование
echo \                      # корректная команда
```

echo \n # корректная команда
 echo \\$USER # корректная команда. вывод: \$USER (за счет экранирования)
 echo "Hello, \$USER!" # корректная команда. замена переменной
 echo Hello, \$USER! #comment # корректная команда. замена переменной + комментарий
 echo Hello, \$USER#nocomment # корректная команда. замена переменной. комментария нет
 echo Hello, \$USER \#nocomment # корректная команда. замена переменной. комментария нет, т.к. экранируется '#'
 echo Hello, \$USER1 # корректная команда. нет замены переменной.
 echo Hello, \$user # корректная команда. нет замены переменной.
 echo Hello, \$USERa # корректная команда. нет замены переменной.
 echo Hello,\$USER! # корректная команда. замена переменной
 echo \$USER\$HOME\$SHELL # корректная команда. замена переменной
 echo \ (Hello\) # корректная команда. 1 аргумент. экранирование скобок

Синтаксические тесты

(pwd) # корректная команда
 (pwd # дисбаланс скобок
 pwd) # дисбаланс скобок
)pwd # дисбаланс скобок
 (((# дисбаланс скобок
))) # дисбаланс скобок
 (pwd() # дисбаланс скобок
 \ (pwd) # дисбаланс скобок. экранирование
 (pwd\) # дисбаланс скобок. экранирование
 pwd(ls) # отсутствует разделитель команд
 (ls)pwd # отсутствует разделитель команд
 (ls(ls)) # отсутствует разделитель команд
 (ls); # корректная команда
 (ls)& # корректная команда
 (ls;) # корректная команда
 (ls&) # корректная команда
 pwd && # отсутствует команда

|| pwd # отсутствует команда
& pwd # отсутствует команда
;;; # отсутствует команда
(pwd) < in.txt > out. txt # корректная команда
pwd >>> out.txt # отсутствует разделитель команд
| pwd | cat -n # отсутствует команда
pwd | # отсутствует команда
pwd & | ls # отсутствует команда
pwd && # отсутствует разделитель команд
pwd && | ls # отсутствует команда

Исполнительные тесты

exit # завершение процесса
exit&
cd # смена каталога на домашний
cd . # смена каталога на текущий
cd .. # смена каталога на родительский
cd / # смена каталога на корневой
cd \$HOME # смена каталога на домашний
cd ~ # смена каталога на домашний
cd noex # noex - несуществующий файл
cd arg1 arg2 <...> # >= 2 аргументов у команды cd
touch info.txt
логика
pwd && ls
pwd || ls
pwd && false || ls
pwd && true || ls
pwd || false && ls
pwd || true && ls
pwd || ls || who;
pwd && ls && who;
pwd || ls || who&

pwd && ls && who&

subshell

(exit)

(pwd)

((pwd))

((pwd|cat -n)|cat -n)

(pwd;exit;ls)

(exit;yes)

(pwd) > info.txt

(echo \$USER) >> info.txt

yes | head

ls | cat -n >> info.txt

pwd | pwd | pwd

pwd | sleep 3

sleep 5& pwd

sleep 3; pwd

sleep 3 && pwd

(false) && pwd

(false&) && pwd

(false;ls) && pwd

(ls;true) && pwd

pwd | false && ls

pwd | true && ls