Белоусова Надежда, 215 группа

**Отчет по заданию shell**

1. Описание синтаксиса

<команда\_shell> ::= <список\_команд>

<список\_команд> ::= <конвейер> { [один из &, &&, ||, ;] <конвейер>} [&, ;]

<конвейер> ::= <команда> { | <команда>}

<команда> ::= <имя\_файла> {<аргумент>} [<имя\_файла>] [ [один из >, >>]

<имя\_файла>]

1. Набор тестов
2. Правильная работа

Команда:

**/Users/nadezdabelousova/C/Programming/Studying/2023\_belousova215/Task\_5 >>>**

cat f; date; pwd > zz

Вывод:

abc

Tue Dec 12 16:25:19 MSK 2023

Команда:

**/Users/nadezdabelousova/C/Programming/Studying/2023\_belousova215/Task\_5 >>>**

ls && pwd

Вывод:

Makefile

error.h

file.in

list.o

sh

test.c

tree\_.h

'Пример структуры проекта.zip'

README.md

error.o

file.out

my\_home.txt

shell.c

tree.c

f

zz

'Примеры тестов для shell.pdf'

current\_dir.txt

list.c

print\_tree.c

shell.o

tree.h

'Методические указания по Task5.pdf'

'Структуры данных для реализации shell.pdf'

error.c

file.bin

list.h

prog.c

test tree.o

'Печать каталогов.pdf'

'задание SHELL.pdf'

/Users/nadezdabelousova/C/Programming/Studying/2023\_belousova215/Task\_5

Команда:

**/Users/nadezdabelousova/C/Programming/Studying/2023\_belousova215/Task\_5 >>>**

ls | pwd | wc

Вывод:

1 1 72

1. Ошибочный ввод

Команда:

**/Users/nadezdabelousova/C/Programming/Studying/2023\_belousova215/Task\_5 >>>**

wc < file1.in <file2.in

Вывод:

my\_shell: error two input files

Команда:

**/Users/nadezdabelousova/C/Programming/Studying/2023\_belousova215/Task\_5 >>>**

wrong\_command

Вывод:

my\_shell: incorrect command

Команда:

**/Users/nadezdabelousova/C/Programming/Studying/2023\_belousova215/Task\_5 >>>**

ls | pwd || | wc

Вывод:

my\_shell: error near ||

1. Используемые структуры данных

Для разбиения строки на лексемы используется односвязный список строк, который состоит из узлов соответствующего вида:

typedef char \* Data;

typedef struct Node Node;

struct Node

{

Data data; \\ строка, лежащая в узле

Node \*next; \\ ссылка на следующий узел

};

Для хранения информации о введенной команде используется следующая древовидная структура данных:

enum type\_of\_next

{

NXT,

AND,

OR

};

typedef struct cmd\_inf tree\_node;

struct cmd\_inf

{

int argc; // количество аргументов

char \*\*argv; // список из имени команды и аргументов

char \*infile; // переназначенный файл стандартного ввода

char \*outfile; // переназначенный файл стандартного вывода

int append; // дозапись в файл

int backgrnd; // = 1, если команда подлежит выполнению в фоновом режиме

struct cmd\_inf \*psubcmd; // команды для запуска в дочернем shell

struct cmd\_inf \*pipe; // следующая команда после "|"

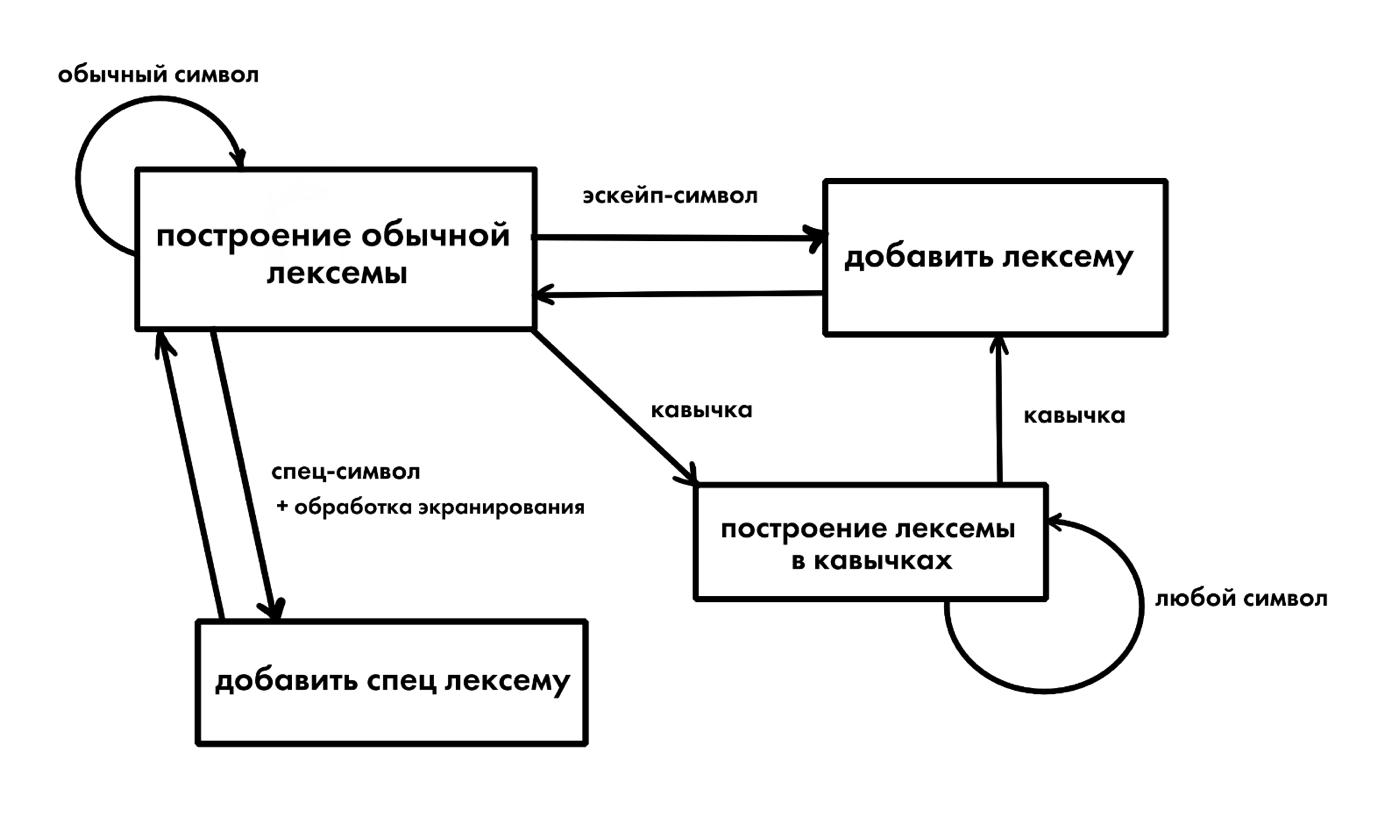
struct cmd\_inf \*next; // следующая после ";" (или после "&")

enum type\_of\_next type; // связь со следующей командой через ; или && или ||

};

1. Построение списка слов и дерева команды

Список слов строится согласно следующему графу:



Дерево команды строится согласно следующему графу:

