Отчёт о выполнении задания shell

Сидоров Леонид

24 декабря 2019 г.

Структура

main.c - основной модуль, отвечает за итеративное выполнение команд.

formlist.c - делит исходный текст на лексемы.

list.c - реализует работу со списком слов.

tree_.c - по списку лексем формирует внутреннее представление команды.

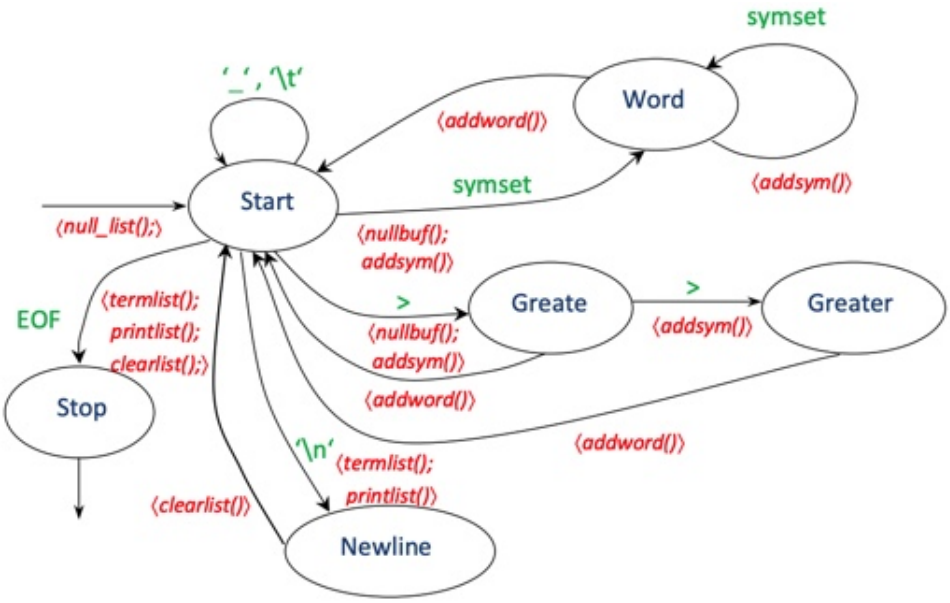
exec.c - выполняет переданную структуру команды.

main.c

Модуль реализует вывод базовой информации об окружении посредством функции *inv* и последовательную обработку команд при помощи обращения к другим модулям. В нём же очищаются все служебные структуры данных.

formlist.c

Является адаптацией *task3* для обработки команд *shell*. Реализован посредством обхода L-графа при помощи указателей на функции.

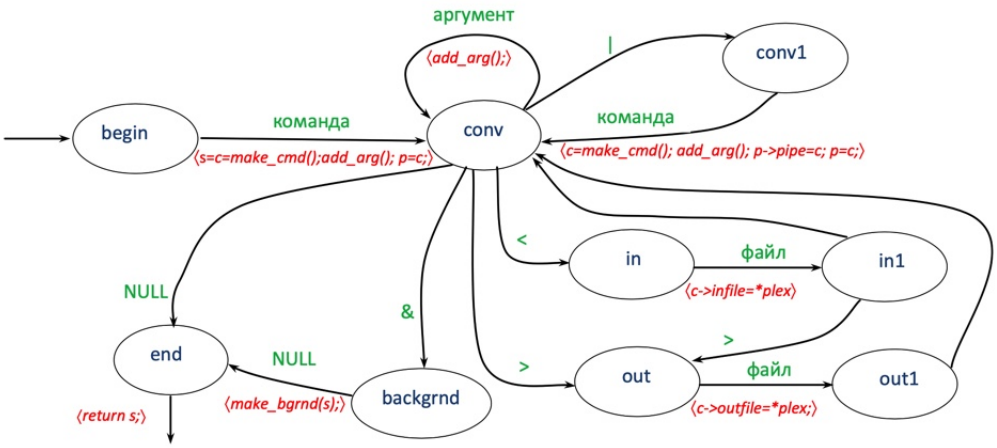


list.c

Является обёрткой над списком слов *lst*, предоставляя для *formlist.c* команды для добавления символа (*addsym*), добавления нового слова в список (*addword*), закрытия списка (*termlist*) и очистки списка (*clearlist*).

tree_.c

Строит структуру типа *tree*, которая содержит всю необходимую информацию о команде и моделирует взаимосвязь различных процессов в рамках одной команды. Модуль реализован с помощью L-графа и указателей на функции.




```
struct cmd_inf {
        list argv; // список из имени команды и аргументов
        int argc;
        char *infile; // переназначенный файл стандартного ввода
        char *outfile; // переназначенный файл стандартного вывода
        int append; // =1, если вывод дописывается в конец файла
        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 ||
};
typedef struct cmd_inf *tree;
```

exec.c

Выполняет команды, преобразованные в структуру *tree*. Команды *exec_simple_com*, *exec_conv*, *exec_com_list* и *exec_com_sh* реализуют соответсвующие сущности из моей бнф, каждая команда отвечает за свой уровень вложенности.

$< simple\_command >::=< file\_name > \{< argument >\} < redirect >$

$< command >::=< simple\_command >$

$< conveyor >::=< command > \{'|' < command >\}$

$< command\_list >::=< conveyor > \{< sep\_character >< conveyor > \} < end\_of\_list >$

$< shell\_command >::=< command\_list >$

Также в этом модуле реализованы функции и структуры для работы со списком фоновых процессов. *add_elem* добавляет элемент в список фоновых процессов. *clear_intlist* очищает элемент списка, если процесс, на который он указывает, уже завершился. Также эта функция убивает зомби-процесс на который указывает элемент списка. *clear_zombie* убирает из списка все звенья, указывающие на зомби-процессы.