

Отчет по заданию task 5

Ищенко Ю. А., 209 группа

Постановка задачи

Интерпретатор команд shell — это (интерактивная) программа, воспринимающая и выполняющая команды, вводимые пользователем с терминала или находящиеся в командном файле (т.е. текстовом файле, содержащем последовательность команд).

Команды делятся на «внутренние», которые shell выполняет непосредственно, и «внешние», для выполнения которых создаются отдельные процессы. Имя любого исполняемого файла Unix является «внешней» командой shell. Кроме того, shell позволяет соединять выполняющиеся команды каналами (создавать «конвейер»), перенаправлять ввод-вывод команд в файлы, выполнять команды в асинхронном режиме.

Интерпретатор должен выдавать приглашение на ввод очередной команды, например “=> ” или “\$ ”. При наступлении ситуации «конец файла» (Ctrl-D при вводе с клавиатуры) интерпретатор завершается.

Синтаксис

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

Отличия от bash

Фоновым можно сделать только последний конвейер в строке (& может стоять только в конце, иначе – неправильный формат ввода).

Если при вводе отсутствует закрывающая кавычка, будет сообщение об ошибке с указанием на это.

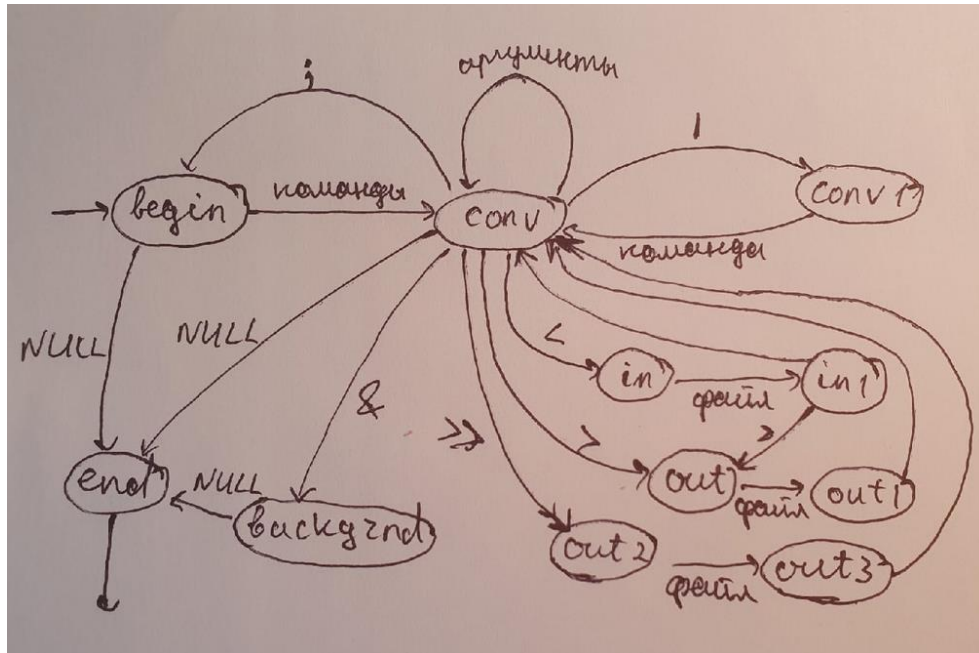
Не реализованы: (), ||, &&

Составляющие программы

Список файлов, которые являются составляющими проекта:

- list.h (объявление структуры Node, функций, нужных для построения дерева)
- list.c (построение, удаление списка слов, преобразование переменных, начинающихся с \$)
- tree.h (объявление структуры cmd_inf, функций, нужных для построения дерева)
- tree.c (построение, удаление, печать дерева, приглашение к вводу)
- exes.h (объявление функций для выполнения команд)
- exes.c (реализация конвейера, фонового режима)
- main.c (вызов функций построения списка (list), преобразования переменных, начинающихся с \$, построение дерева (tree), выполнения команд (exes), удаление списка, дерева)
- Makefile (компиляция и сборка всех объектных файлов, запуск проекта без каких-либо редакторов)

Схема L-графа для построения дерева



Описание структур

Структура данных, хранящая одно слово из строки:

```
typedef struct Node *link1;  
typedef char *elemtype;  
typedef struct Node{  
    elemtype elem; - хранит указатель на слово  
    link1 next; - указатель на следующий элемент  
}node;
```

Структура данных, хранящая информацию для выполнения одной команды:

```
struct cmd_inf {  
    char ** argv; - список из имени команды и аргументов  
    char *infile; - переназначенный файл стандартного ввода  
    char *outfile; - переназначенный файл стандартного вывода  
    int append; - равен 1, если запись в конец файла  
    int backgrnd; - равен 1, если команда подлежит выполнению в фоновом режиме  
    struct cmd_inf* pipe; - следующая команда после "|"   
    struct cmd_inf* next; - следующая после ";"  
};
```

Наборы тестов

Правильное выполнение:

- `cd ..` (переходит в родительскую директорию)
- `echo homedir: $HOME username: $USER`
- `pwd > fpwd` (пишет имя текущей директории в файл `fpwd`)
- `yes | head` (печатает 10 раз букву `y`)
- `pwd | cat >> my_home.txt` (дописывает домашнюю директорию пользователя в файл `my_home.txt`)
- `ls -l > f.out` (записывает список файлов текущего каталога с правами доступа в `f.out`)
- `exit` (завершение `my_shell`)
- `cd` (переходит в домашнюю директорию)
- `yes | yes | yes | yes | sleep 10 | pwd` (печатает текущую директорию, приглашение к вводу появляется через 10 секунд)
- `pwd | exit | date | cat` (печатает текущую дату)
- `cat&` (ничего выводит)
- `pwd | exit | date | cat&` (печатает текущую дату в строке со следующим приглашением ко вводу)
- `echo a b c > f; cat f` (записывает “`a b c`” в `f`, печатает “`a b c`”)
- `sleep 5 &` (печатает приглашения ко вводу, но команды выполняются через 5 секунд)
- `sleep 10; pwd` (выводит имя текущей директории через 10 секунд)
- `pwd;` (выводит имя текущей директории)

Тесты с неправильным форматом ввода: “**ERROR: wrong format**”

- `()`
- `echo << f`
- `cat > > f`
- `echo & cat`
- `echo && cat`
- `echo || cat`
- `;`
- `|cat`
- `cat|`

Если введенной команды не существует: “**ERROR: command '*cmd*' not found**”

- `nocmd`

Если не введена закрывающая кавычка: “**ERROR: no closing quote**”

- `echo “sdsdcs scscs`

Если введена неправильная директория при вызове `cd`: “**ERROR: directory '*dir*' cannot be found**”

- `cd noDir`