

Front matter

lang: ru-RU title: Программирование в командном процессоре ОС UNIX. Ветвления и циклы. subtitle: ЛР по ОС №12 author: Танрибергенов Эльдар Марсович group: НПИбд-02-20

Отчёт

о выполнении лабораторной работы № 12.

Программирование в командном процессоре ОС UNIX. Ветвления и циклы.

Выполнил:

студент группы НПИбд-02-20

Танрибергенов Эльдар Марсович.

Студ. билет № 1032208074

Москва 2021 г.

Цель работы:

- Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Задание:

- Используя команды `getopts` `grep`, написать командный файл, который анализирует командную строку с ключами: `-- -inputfile` --- прочитать данные из указанного файла; `-- -ooutputfile` --- вывести данные в указанный файл; `-- -rшаблон` --- указать шаблон для поиска; `-- -C` --- различать большие и малые буквы; `-- -n` --- выдавать номера строк. а затем ищет в указанном файле нужные строки, определяемые ключом `-r`.
- Написать на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается с помощью функции `exit(n)`, передавая информацию о коде завершения в оболочку. Командный файл должен вызывать эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено.
- Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N (например `1.tmp`, `2.tmp`, `3.tmp`, `4.tmp` и т.д.). Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (если они существуют).
- Написать командный файл, который с помощью команды `tar` запаковывает в архив все файлы в указанной директории. Модифицировать его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовать команду `find`).

Теоретическое введение

[\[1\]](#)

Командный файл (скрипт или сценарий) - это текстовый файл, состоящий из команд интерпретатора.

При запуске этого файла последовательно выполняются все команды, содержащиеся в нём.

Как файл его можно создать командой `cat` или использовать текстовый редактор, например, `vi`.

Для тех, кто привык работать с текстовым редактором, встроенным в файловый менеджер Midnight Commander (МС - запуск), в котором имеется встроенный текстовый редактор с более привычным интерфейсом.

Краткий справочник команд Linux [\[2\]](#)

Ход работы:

А)

Создал файл `r1` и написал там скрипт.

Командный файл 1.

[https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/c1%20\(1\).png](https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/c1%20(1).png)

[https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/c1%20\(2\).png](https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/c1%20(2).png)

Создал два файла: файл-источник и файл, который будет содержать вывод.

https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/1.png

Источник заполнил отрывком текста из файла лабораторной работы.

Файл-источник.

https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/2.png

Наделил файл `r1` правом на выполнение и начал испытания.

Тест 1.

[https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/t1%20\(1\).png](https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/t1%20(1).png)

Вывод 1.

[https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/r1%20\(1\).png](https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/r1%20(1).png)

Тест 2.

[https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/t1%20\(2\).png](https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/t1%20(2).png)

Вывод 2.

[https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/r1%20\(2\).png](https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/r1%20(2).png)

Тест 3.

[https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/t1%20\(3\).png](https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/t1%20(3).png)

Б)

Создал файл `BMRnul` и в нём написал программу на си.

Программа на си.

[https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/c2%20\(1\).png](https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/c2%20(1).png)

Создал файл p2 и написал там скрипт.

Командный файл 2.

[https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/c2%20\(2\).png](https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/c2%20(2).png)

Испытание:

[https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/r2%20\(1\).png](https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/r2%20(1).png)

[https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/r2%20\(2\).png](https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/r2%20(2).png)

По какой-то неизвестной мне причине отрицательное число выдалось в \$? не правильно.

В)

Создал файл p3, в нём написал скрипт.

Командный файл 3.

https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/c3.png

Испытание:

https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/r3.png

При последнем запуске командного файла с заданным количеством файлов 7, как и нужно было, произошло следующее: все существовавшие файлы от 1 до 5 были удалены, а 6.tmp и 7.tmp были созданы, т.к. не существовали ранее.

Г)

Создал файл p4, написал в нём скрипт.

Командный файл 4.

https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/c4.png

Для испытания создал директорию provdir, которую заполнил 9-ю файлами, созданными при помощи предыдущего командного файла.

Директория provdir.

https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/3.png

Испытание:

[https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/r4%20\(1\).png](https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/r4%20(1).png)

Как видно, появился архив. Просмотрю его содержимое командой less.

Содержимое архива.

[https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/r4%20\(2\).png](https://github.com/emtanribergenov/OS_labs/blob/master/12/screenshots/r4%20(2).png)

Вывод:

В результате лабораторной работы я изучил основы программирования в оболочке ОС UNIX. Научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Ответы на вопросы:

1. Анализирует командную строку и выявляет опции.
2. Метасимволы - символы, имеющие специальное значение для интерпретатора : ... > <пробел> <табуляция> <возврат_каретки>. Однако каждый из этих символов может представлять самого себя, если перед ним стоит \. Все символы, заключенные между кавычками ' и ', представляют самих себя. Между двойными кавычками (") выполняются подстановки команд (см п. 2.2) и параметров (см. п. 2.3), а символы , ` , " и \$ могут экранироваться предшествующим символом \. ... Если ни одно имя файла не удовлетворяет шаблону, то он остается неизменным. Значения указанных символов: * любая строка, включая и пустую.
3. Условные операторы и циклы.
4. Оператор break.
5. **true** и **false** - это значения, которые может принять переменная, объявленная как bool (логическая). По сути true и false эквивалентно да и нет.
6. Если есть файл справки, то...
7. while : проверяет условие, если истинно выполняет "тело"

until : выполняет "тело", а потом проверяет условие