Отчет по лабораторной работе №11

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Максим Сергеевич Белов

Содержание

# 1 Цель работы

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

# 2 Задание

1. Используя команды getopts grep, написать командный файл, который анализирует командную строку с ключами:

* -i inputfile - прочитать данные из указанного файла;
* -o outputfile - вывести данные в указанный файл;
* -p шаблон - указать шаблон для поиска;
* -C - различать большие и малые буквы;
* -n - выдавать номера строк. а затем ищет в указанном файле нужные строки, определяемые ключом -p.

1. 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 в о коде завершения в оболочку. Командный файл должен вызывать эту программу и, проанализировав с помощью команды $?, выдать сообщение о том, какое число было введено.
2. 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).
3. 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).

# 3 Теоретическое введение

В табл. 1 приведено краткое описание команд для работы с командными файлами.

Таблица 1: Описание команд для работы с командными файлами

| Команда | Описание |
| --- | --- |
| getopts | Осуществляет синтаксический анализ командной строки, выделяя флаги, и используетсядля объявления переменных. |

# 4 Выполнение лабораторной работы

1. Скрипт 1 (Рис. 1)

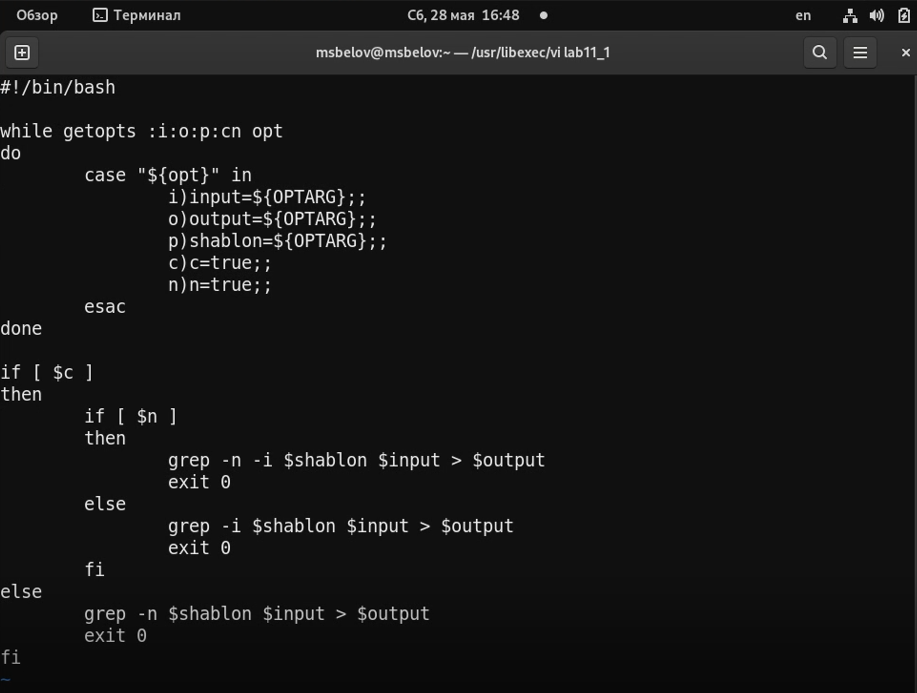


Рис. 1: Скрипт 1

Сделаем файл исполняемым и проверим его работу (Рис. 2)



Рис. 2: Проверка скрипта 1

1. Программа 2.1 (C++) (Рис. 3)



Рис. 3: Программа на си

Командный файл 2.2 (Рис. 4)



Рис. 4: Командный файл 2.2

Проверка работы командного файла (Рис. 5)

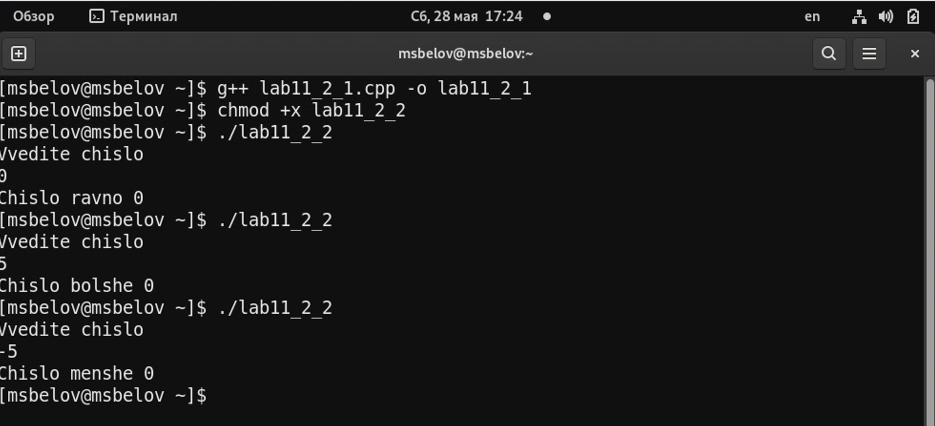


Рис. 5: Проверка скрипта 2

1. Скрипт 3 (Рис. 6)



Рис. 6: Скрипт 3

Сделаем файл исполняемым и проверим его работу

(Рис. 7)

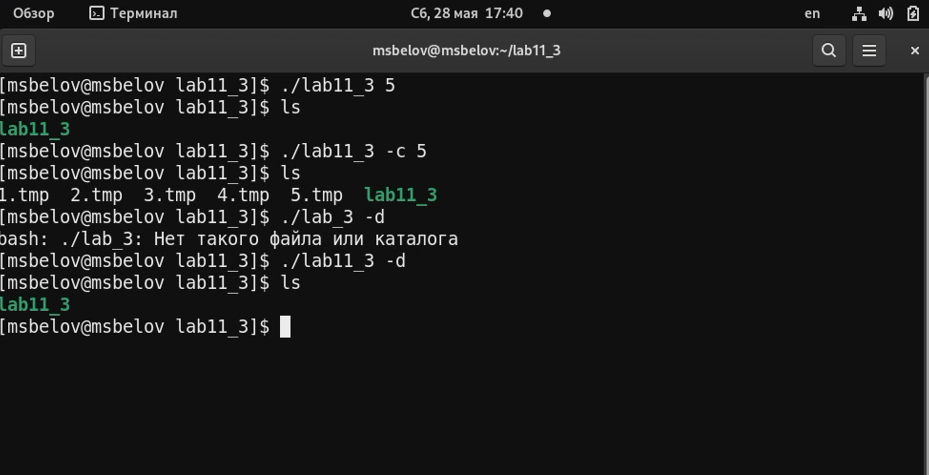


Рис. 7: Проверка скрипта 3

1. Скрипт 4 (Рис. 8)



Рис. 8: Скрипт 4

Сделаем файл исполняемым и проверим его работу (Рис. 9)

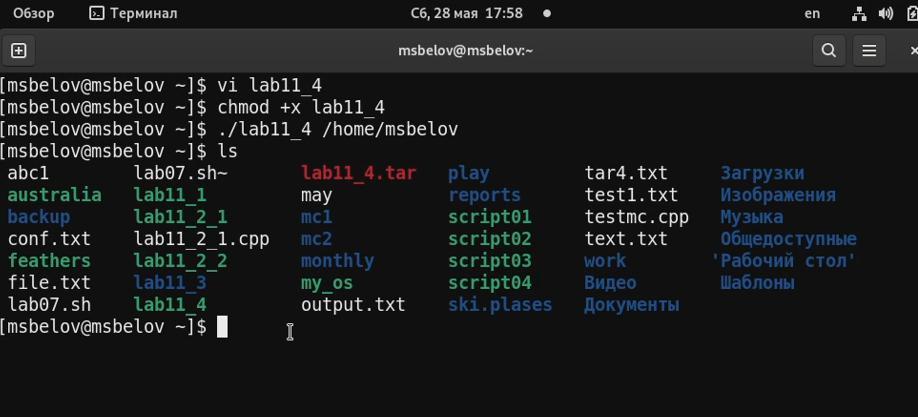


Рис. 9: Проверка скрипта 4

# 5 Выводы

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

# 6 Контрольные вопросы

1. Осуществляет синтаксический анализ командной строки, выделяя флаги, и используется для объявления переменных.
2. При перечислении имен файлов текущего каталога можно использовать некторые символы.
3. for, case, if, while
4. break
5. true,: - всегда возвращает 0 в качестве кода выхода. false - всегда возвращает 1 в качестве кода выхода.
6. Введенная строка означает условие существования файла mani.$s
7. Если речь идет о 2-х параллельных действиях, то это while. когда мы показываем, что сначала делается 1-е действие. потом оно заканчивается при наступлении 2-го действия, применяем until.