

Лабораторная работа №10

**Программирование в командном процессоре ОС UNIX. Командные
файлы**

Беличева Д.М.; НКНбд-01-21

Содержание

| | | |
|----------|---------------------------------------|-----------|
| 1 | Цель работы | 4 |
| 2 | Задание | 5 |
| 3 | Теоретическое введение | 6 |
| 4 | Выполнение лабораторной работы | 8 |
| 5 | Выводы | 13 |
| | Список литературы | 14 |

Список иллюстраций

| | | |
|------|---|----|
| 4.1 | Текст первой программы | 8 |
| 4.2 | Проверка работы программы | 9 |
| 4.3 | Создание файла для второй программы, проверка содержимого домашнего каталога | 9 |
| 4.4 | Текст второй программы | 10 |
| 4.5 | Проверка работы второй программы | 10 |
| 4.6 | Создание файла для третьей программы | 11 |
| 4.7 | Текст третьей программы | 11 |
| 4.8 | Проверка работы третьей программы | 11 |
| 4.9 | Текст четвертой программы | 12 |
| 4.10 | Проверка работы четвертой программы | 12 |

1 Цель работы

Изучить основы программирования в оболочке ОС UNIX/Linux. Научиться писать небольшие командные файлы.

2 Задание

1. Написать скрипт, который при запуске будет делать резервную копию самого себя (то есть файла, в котором содержится его исходный код) в другую директорию `backup` в вашем домашнем каталоге. При этом файл должен архивироваться одним из архиваторов на выбор `zip`, `bzip2` или `tar`. Способ использования команд архивации необходимо узнать, изучив справку.
2. Написать пример командного файла, обрабатывающего любое произвольное число аргументов командной строки, в том числе превышающее десять. Например, скрипт может последовательно распечатывать значения всех переданных аргументов.
3. Написать командный файл — аналог команды `ls` (без использования самой этой команды и команды `dir`). Требуется, чтобы он выдавал информацию о нужном каталоге и выводил информацию о возможностях доступа к файлам этого каталога.
4. Написать командный файл, который получает в качестве аргумента командной строки формат файла (`.txt`, `.doc`, `.jpg`, `.pdf` и т.д.) и вычисляет количество таких файлов в указанной директории. Путь к директории также передаётся в виде аргумента командной строки.

3 Теоретическое введение

Командный процессор (командная оболочка, интерпретатор команд shell) — это программа, позволяющая пользователю взаимодействовать с операционной системой компьютера. В операционных системах типа UNIX/Linux наиболее часто используются следующие реализации командных оболочек: - оболочка Борна (Bourne shell или sh) — стандартная командная оболочка UNIX/Linux, содержащая базовый, но при этом полный набор функций;

- С-оболочка (или csh) — надстройка на оболочкой Борна, использующая С-подобный синтаксис команд с возможностью сохранения истории выполнения команд;
- оболочка Корна (или ksh) — напоминает оболочку С, но операторы управления программой совместимы с операторами оболочки Борна;
- BASH — сокращение от Bourne Again Shell (опять оболочка Борна), в основе своей совмещает свойства оболочек С и Корна (разработка компании Free Software Foundation).

POSIX (Portable Operating System Interface for Computer Environments) — набор стандартов описания интерфейсов взаимодействия операционной системы и прикладных программ. Стандарты POSIX разработаны комитетом IEEE (Institute of Electrical and Electronics Engineers) для обеспечения совместимости различных UNIX/Linux-подобных операционных систем и переносимости прикладных программ на уровне исходного кода. POSIX-совместимые оболочки разработаны

на базе оболочки Korn. Рассмотрим основные элементы программирования в оболочке `bash`. В других оболочках большинство команд будет совпадать с описанными ниже. [1]

4 Выполнение лабораторной работы

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 4.1)

1. Написать скрипт, который при запуске будет делать резервную копию самого себя (то есть файла, в котором содержится его исходный код) в другую директорию backup в вашем домашнем каталоге. При этом файл должен архивироваться одним из архиваторов на выбор zip, bzip2 или tar. Способ использования команд архивации необходимо узнать, изучив справку. рис. (4.1, 4.2, 4.3)

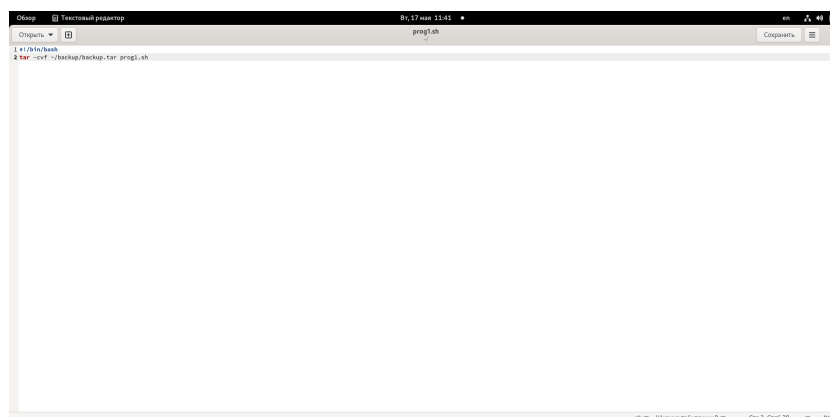


Рис. 4.1: Текст первой программы

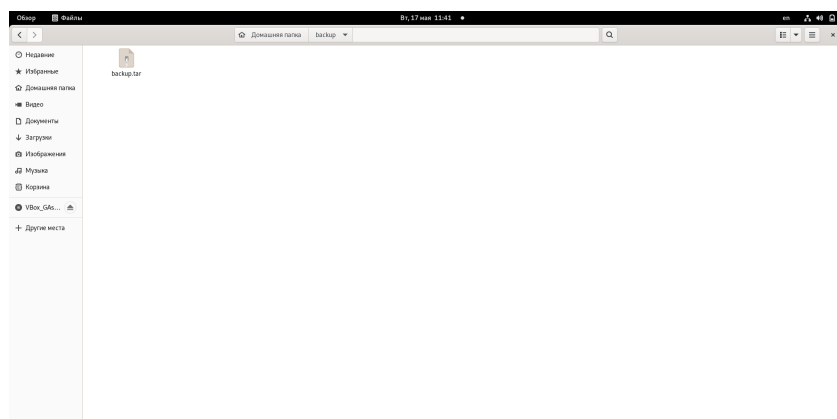


Рис. 4.2: Проверка работы программы

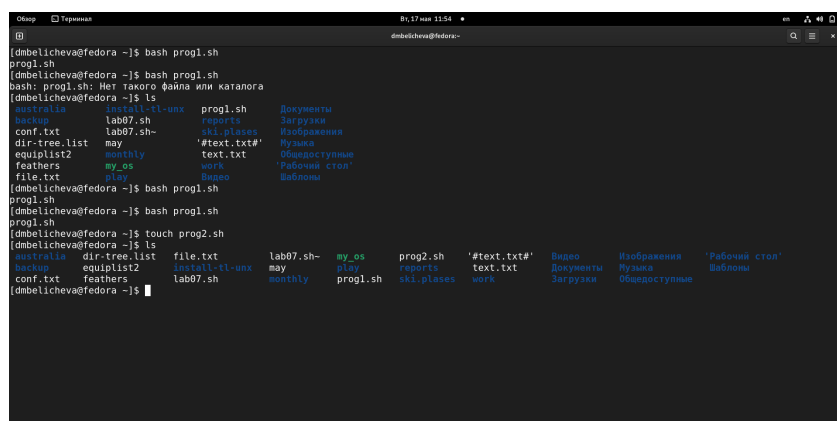


Рис. 4.3: Создание файла для второй программы, проверка содержимого домашнего каталога

2. Написать пример командного файла, обрабатывающего любое произвольное число аргументов командной строки, в том числе превышающее десять. Например, скрипт может последовательно распечатывать значения всех переданных аргументов. (4.4, 4.5)

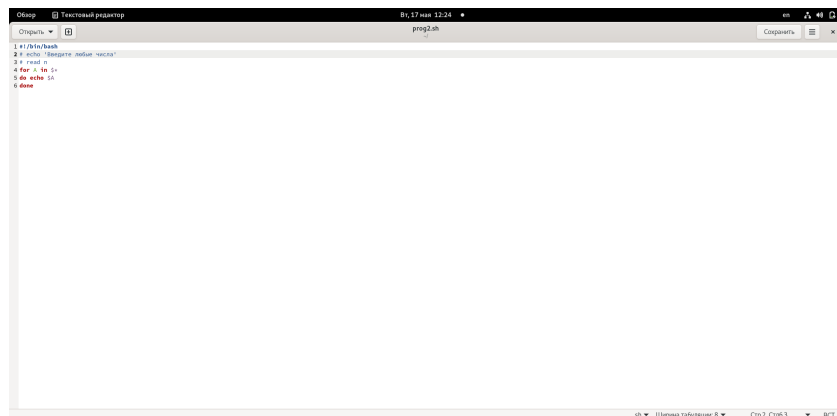


Рис. 4.4: Текст второй программы

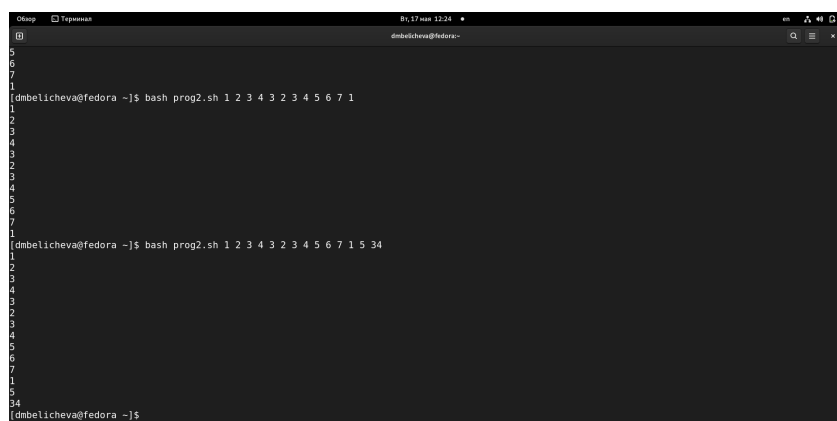


Рис. 4.5: Проверка работы второй программы

3. Написать командный файл — аналог команды ls (без использования самой этой команды и команды dir). Требуется, чтобы он выдавал информацию о нужном каталоге и выводил информацию о возможностях доступа к файлам этого каталога. (4.6, 4.7, 4.8)

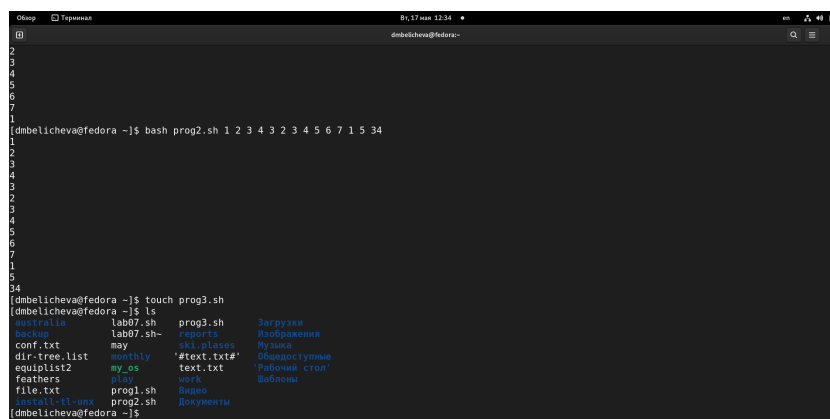


Рис. 4.6: Создание файла для третьей программы

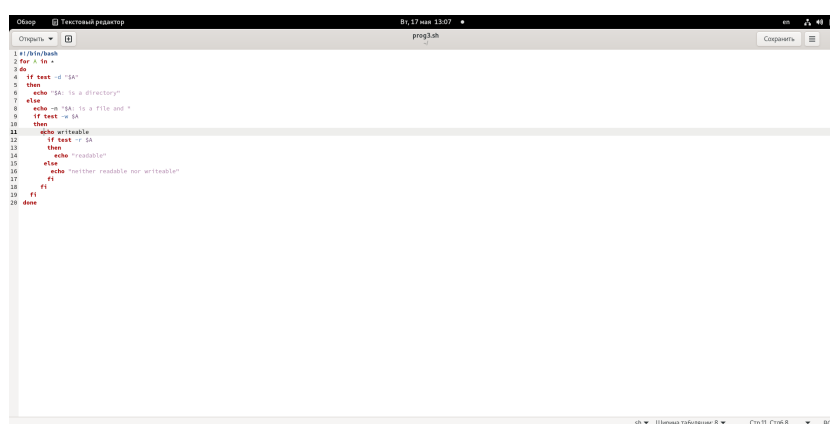


Рис. 4.7: Текст третьей программы

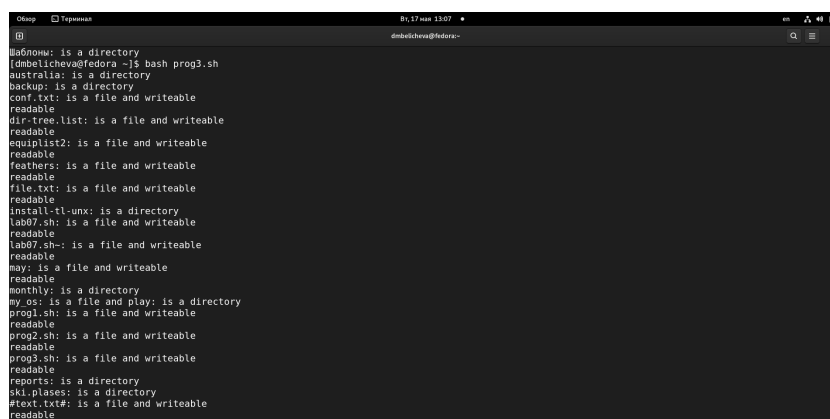
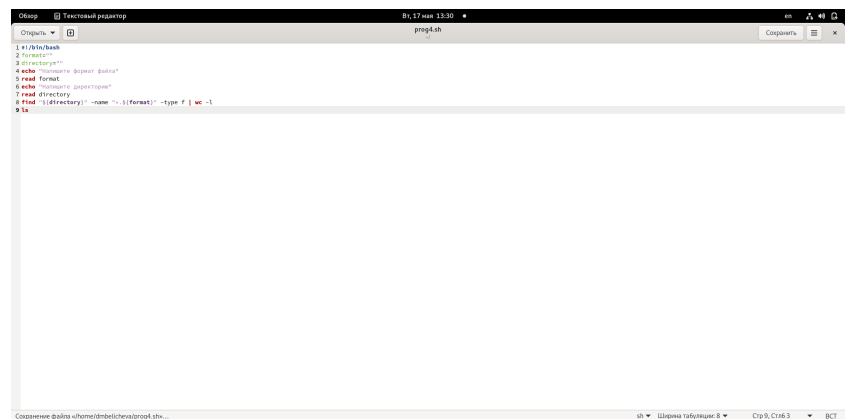


Рис. 4.8: Проверка работы третьей программы

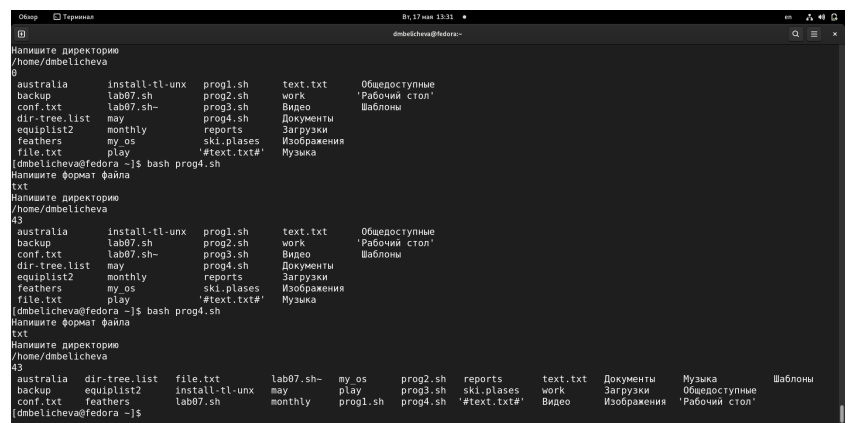
4. Написать командный файл, который получает в качестве аргумента командной строки формат файла (.txt, .doc, .jpg, .pdf и т.д.) и вычисляет количество

таких файлов в указанной директории. Путь к директории также передаётся в виде аргумента командной строки. (4.9, 4.10)



```
1 #!/bin/bash
2 find "$1" -type f | while read file; do
3     echo "Находите формат файла:"
4     read format
5     echo "Находите директорию:"
6     read directory
7     find "$directory" -name "$format" -type f | wc -l
8 done
```

Рис. 4.9: Текст четвертой программы



```
Напишите директорию
/home/dmbelicheva
0
australia      install-1l-unx  prog1.sh  text.txt  Общедоступные
backup         lab07.sh       prog2.sh  work      'Рабочий стол'
conf.txt       lab07.sh       prog3.sh  Видео     Шаблоны
dir-tree.list  may           prog4.sh  Документы
equiplist2     monthly       reports   Загрузки
feathers        my_os         ski.places  Изображения
file.txt       play          '#text.txt#' Музыка
[dmbelicheva@fedora ~]$ bash prog4.sh
Напишите формат файла
txt
Напишите директорию
/home/dmbelicheva
43
australia      install-1l-unx  prog1.sh  text.txt  Общедоступные
backup         lab07.sh       prog2.sh  work      'Рабочий стол'
conf.txt       lab07.sh       prog3.sh  Видео     Шаблоны
dir-tree.list  may           prog4.sh  Документы
equiplist2     monthly       reports   Загрузки
feathers        my_os         ski.places  Изображения
file.txt       play          '#text.txt#' Музыка
[dmbelicheva@fedora ~]$ bash prog4.sh
Напишите формат файла
txt
Напишите директорию
/home/dmbelicheva
43
australia      dir-tree.list  file.txt      lab07.sh~    my_os        prog2.sh     reports      text.txt     Документы  Музыка      Шаблоны
backup         equiplist2     install-1l-unx  may          play         prog3.sh     ski.places   work         Загрузки   Общедоступные
conf.txt       feathers        lab07.sh       monthly      prog1.sh     prog4.sh     '#text.txt#' Видео        Изображения 'Рабочий стол'
```

Рис. 4.10: Проверка работы четвертой программы

5 Выводы

В процессе выполнения лабораторной работы я изучила основы программирования в оболочке ОС UNIX/Linux. Научилась писать небольшие командные файлы.

Список литературы

1. Лабораторная работа № 10. Программирование в командном процессоре ОС UNIX. Командные файлы [Электронный ресурс]. URL: <https://esystem.ru/dn.ru/>.