

Лабораторная работа 10

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

Паласиос Фелипе

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	13

Список иллюстраций

3.1	Создание	7
3.2	Запуск	8
3.3	backup.tar	8
3.4	Создание	9
3.5	Файл	9
3.6	Запуск	10
3.7	Результат	10
3.8	Создание	11
3.9	Работа файла	11
3.10	Создание	12
3.11	Работа файла	12

Список таблиц

1 Цель работы

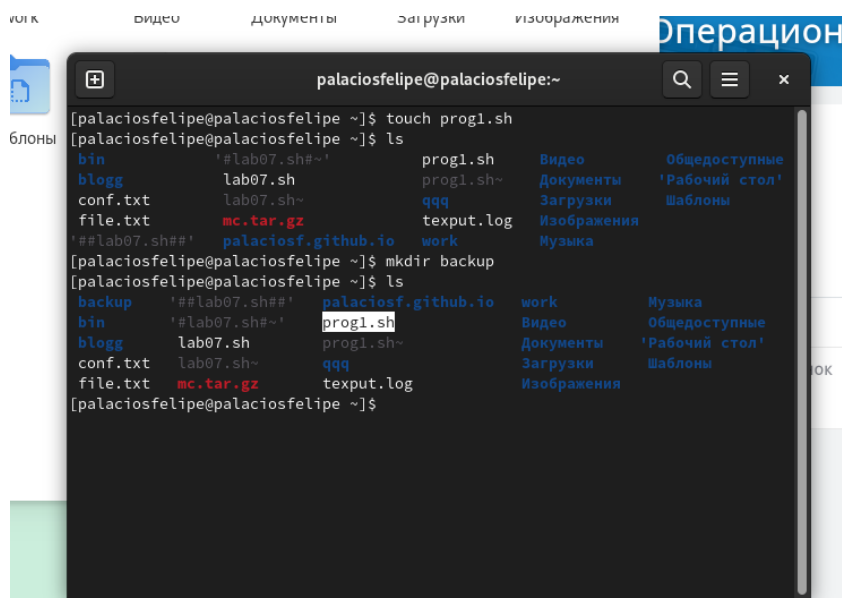
Изучить основы программирования в оболочке ОС UNIX/Linux. Научиться писать небольшие командные файлы.

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 Выполнение лабораторной работы

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. 3.1), (рис. 3.2), (рис. 3.3)



```
palaciosfelipe@palaciosfelipe:~$ touch prog1.sh
palaciosfelipe@palaciosfelipe ~]$ ls
bin          prog1.sh     Видео      Общедоступные
blogg        lab07.sh    Документы  'Рабочий стол'
conf.txt     lab07.sh~   Загрузки   Шаблоны
file.txt     mc.tar.gz   Изображения
palaciosf.github.io  work       Музыка
palaciosfelipe@palaciosfelipe ~]$ mkdir backup
palaciosfelipe@palaciosfelipe ~]$ ls
backup      palaciosf.github.io  work       Музыка
bin          prog1.sh             Видео      Общедоступные
blogg        lab07.sh             Документы  'Рабочий стол'
conf.txt     lab07.sh~            Загрузки   Шаблоны
file.txt     mc.tar.gz            Изображения
palaciosfelipe@palaciosfelipe ~]$
```

Рис. 3.1: Создание

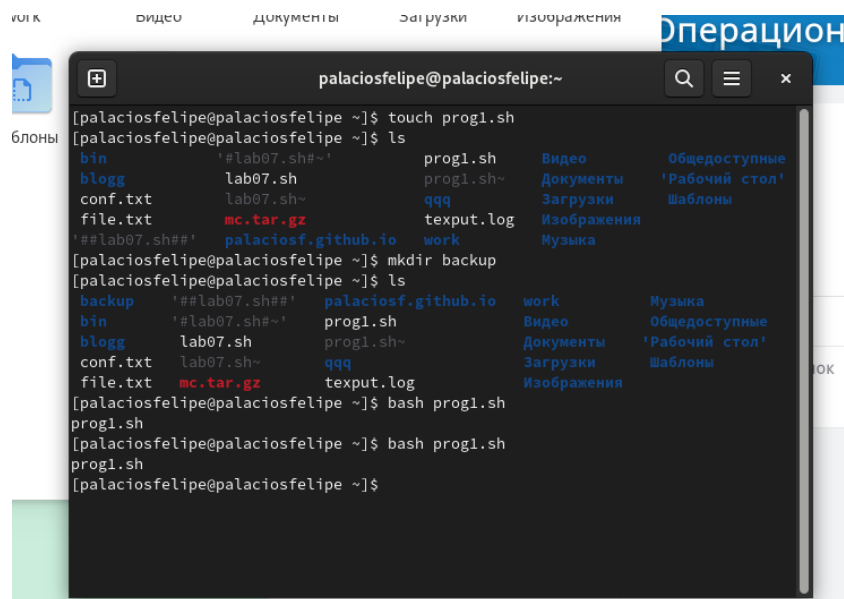


Рис. 3.2: Запуск

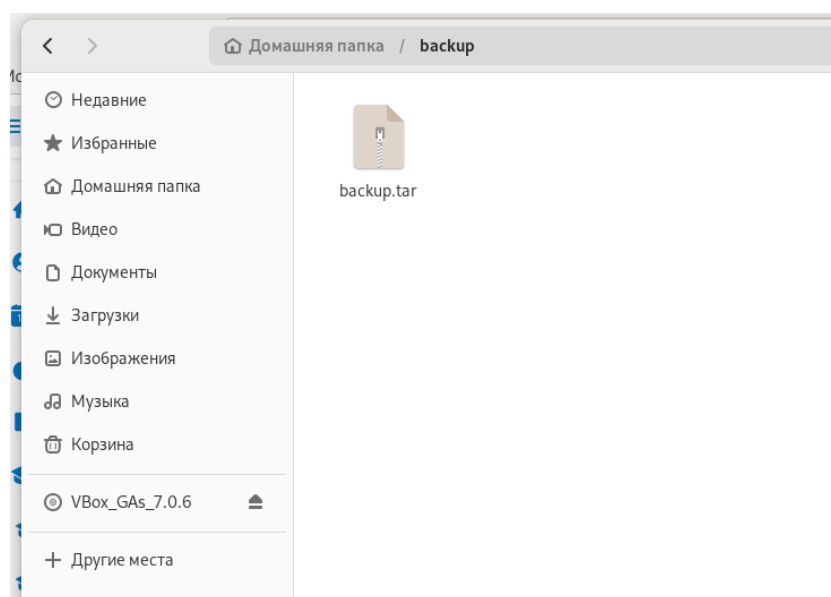
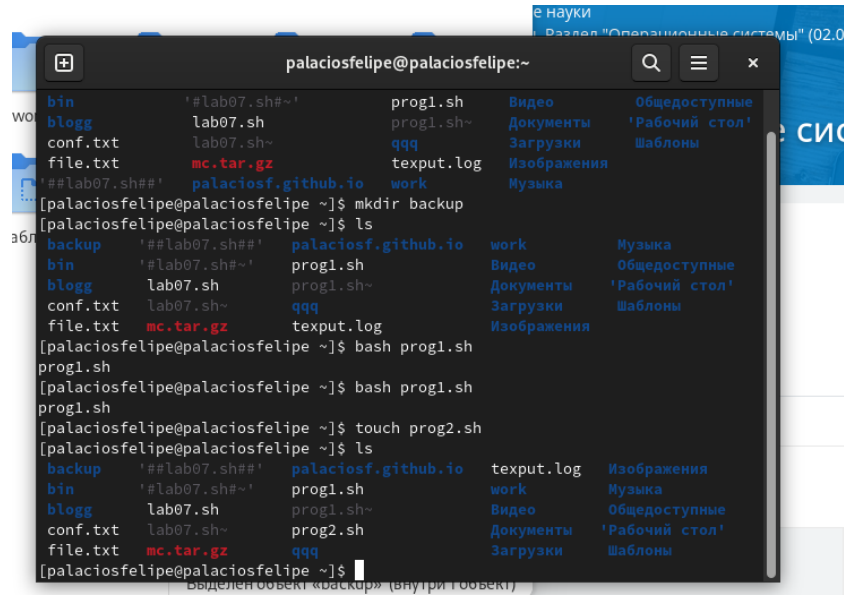


Рис. 3.3: backup.tar

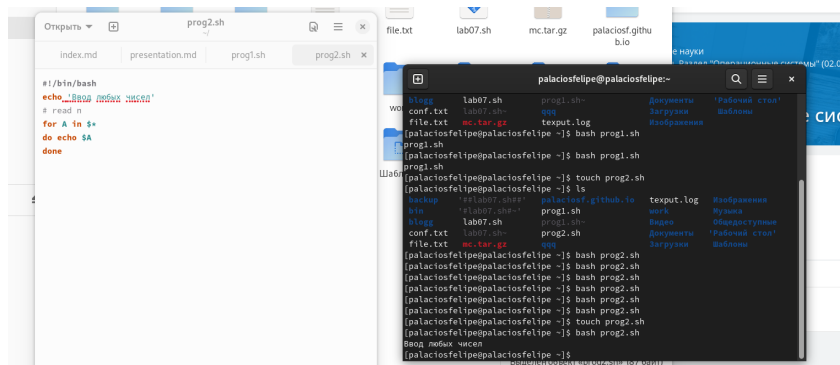
2. Написать пример командного файла, обрабатывающего любое произвольное число аргументов командной строки, в том числе превышающее десять. Например, скрипт может последовательно распечатывать значения всех

переданных аргументов (рис. 3.4), (рис. 3.5), (рис. 3.6), (рис. 3.7)



```
palaciosfelipe@palaciosfelipe:~$ mkdir backup
palaciosfelipe@palaciosfelipe ~$ ls
backup  bin  blogg  conf.txt  file.txt  mc.tar.gz  palaciosf.github.io  prog1.sh  texput.log  work
palaciosfelipe@palaciosfelipe ~$ bash prog1.sh
prog1.sh
palaciosfelipe@palaciosfelipe ~$ bash prog1.sh
prog1.sh
palaciosfelipe@palaciosfelipe ~$ touch prog2.sh
palaciosfelipe@palaciosfelipe ~$ ls
backup  bin  blogg  conf.txt  file.txt  mc.tar.gz  palaciosf.github.io  prog1.sh  prog2.sh  texput.log  work
```

Рис. 3.4: Создание



```
palaciosfelipe@palaciosfelipe:~$ touch prog2.sh
palaciosfelipe@palaciosfelipe ~$ ls
backup  bin  blogg  conf.txt  file.txt  mc.tar.gz  palaciosf.github.io  prog1.sh  prog2.sh  texput.log  work
palaciosfelipe@palaciosfelipe ~$ bash prog1.sh
prog1.sh
palaciosfelipe@palaciosfelipe ~$ bash prog1.sh
prog1.sh
palaciosfelipe@palaciosfelipe ~$ touch prog2.sh
palaciosfelipe@palaciosfelipe ~$ ls
backup  bin  blogg  conf.txt  file.txt  mc.tar.gz  palaciosf.github.io  prog1.sh  prog2.sh  texput.log  work
palaciosfelipe@palaciosfelipe ~$ bash prog2.sh
palaciosfelipe@palaciosfelipe ~$ bash prog2.sh
palaciosfelipe@palaciosfelipe ~$ bash prog2.sh
palaciosfelipe@palaciosfelipe ~$ bash prog2.sh
palaciosfelipe@palaciosfelipe ~$ bash prog2.sh
palaciosfelipe@palaciosfelipe ~$ touch prog2.sh
palaciosfelipe@palaciosfelipe ~$ bash prog2.sh
Ввод двух чисел
palaciosfelipe@palaciosfelipe ~$
```

Рис. 3.5: Файл

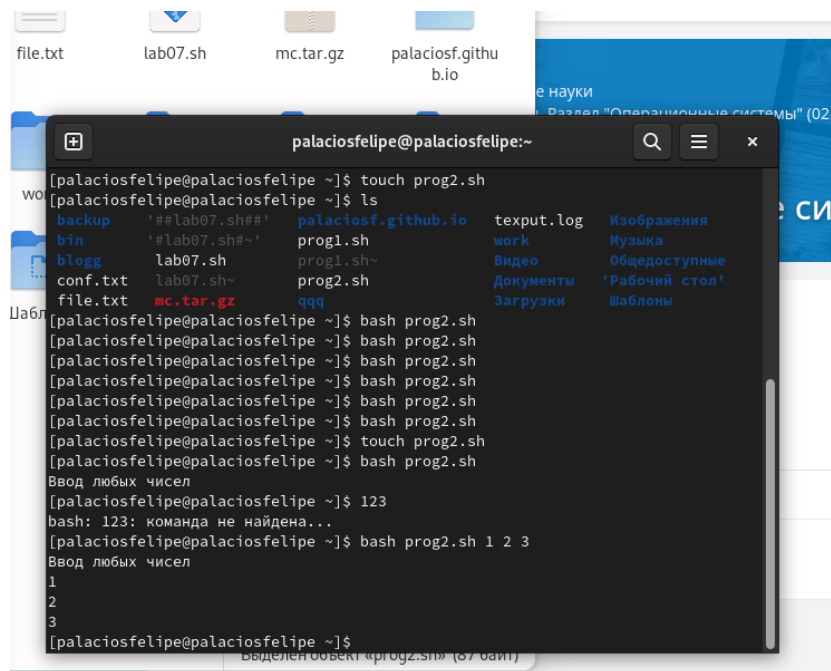


Рис. 3.6: Запуск

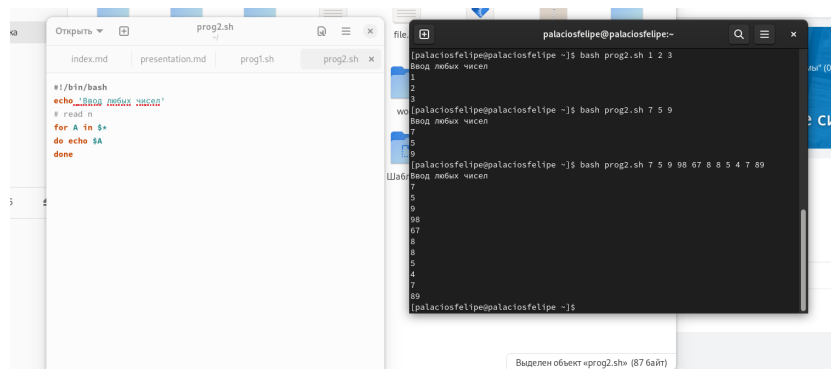


Рис. 3.7: Результат

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 (рис. 3.8),(рис. 3.9)

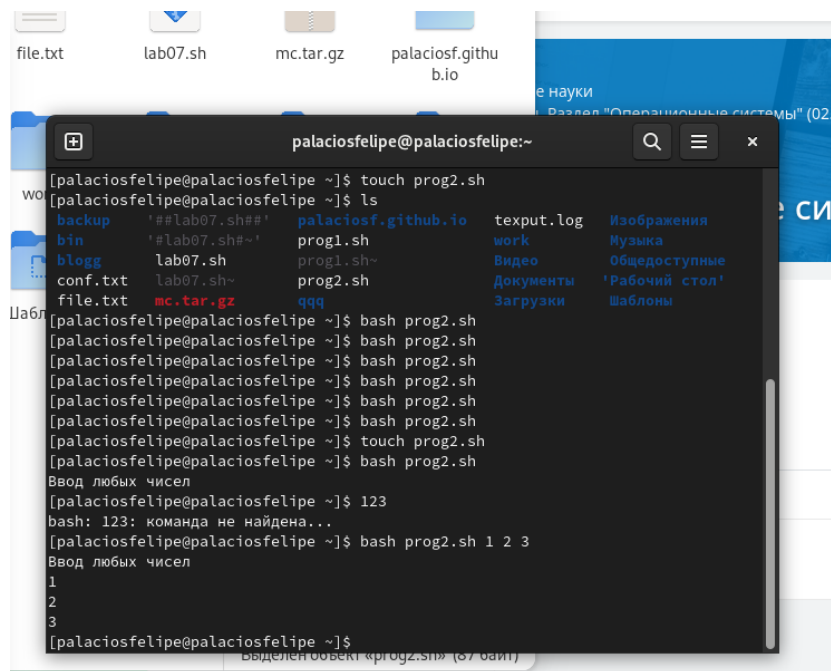


Рис. 3.8: Создание

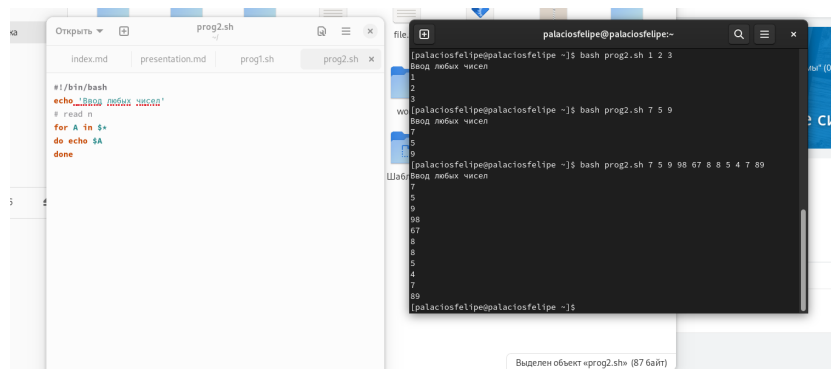


Рис. 3.9: Работа файла

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 (рис. 3.10),(рис. 3.11)

4 Выводы

Изучил основы программирования в оболочке ОС UNIX/Linux. Научился писать небольшие командные файлы.