

# **Лабораторная работа №10**

Доберштейн Алина

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>5</b>	<b>Выводы</b>	<b>14</b>

## Список иллюстраций

4.1	Скрипт . . . . .	8
4.2	Справка . . . . .	9
4.3	Скрипт . . . . .	10
4.4	Пример работы скрипта . . . . .	10
4.5	Скрипт . . . . .	11
4.6	Пример работы скрипта . . . . .	11
4.7	Скрипт . . . . .	12
4.8	Пример работы скрипта . . . . .	12
4.9	Пример работы скрипта . . . . .	13

## Список таблиц

# 1 Цель работы

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

## 2 Задание

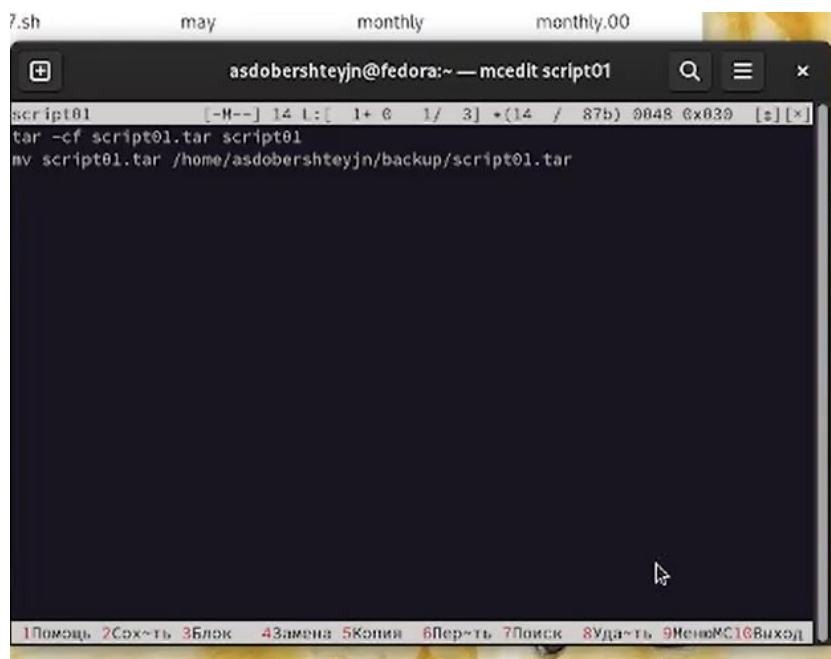
1. Написать скрипт, который при запуске будет делать резервную копию самого себя (то есть файла, в котором содержится его исходный код) в другую директорию backup в вашем домашнем каталоге. При этом файл должен архивироваться одним из архиваторов на выбор zip, bzip2 или tar. Способ использования команд 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 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; – C-оболочка (или csh) — надстройка на оболочке Борна, использующая C-подобный синтаксис команд с возможностью сохранения истории выполнения команд; – оболочка Корна (или ksh) — напоминает оболочку C, но операторы управления программой совместимы с операторами оболочки Борна; – BASH — сокращение от Bourne Again Shell (опять оболочка Борна), в основе своей совмещает свойства оболочек C 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 Корна. Рассмотрим основные элементы программирования в оболочке bash. В других оболочках большинство команд будет совпадать с описанными ниже.

## 4 Выполнение лабораторной работы

1. Написала 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. (рис. 4.1).

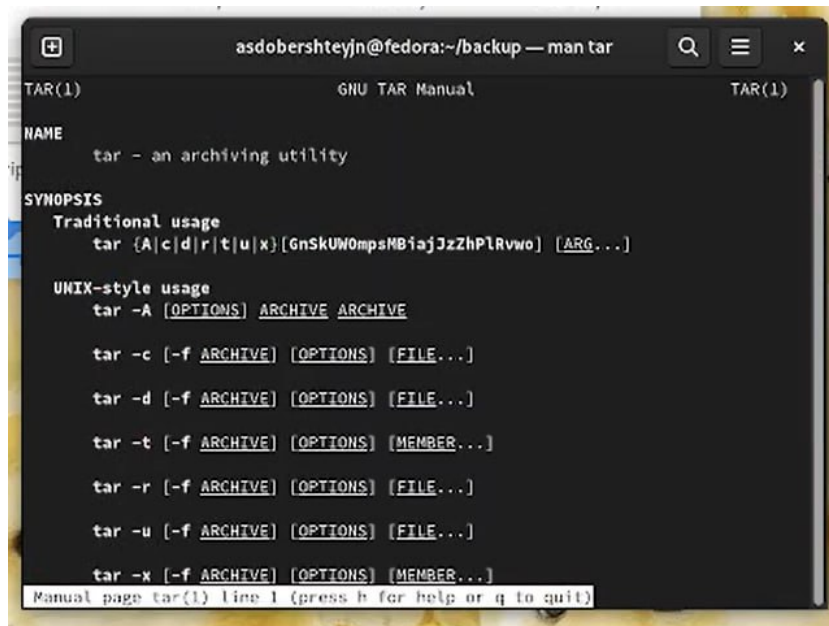


```
script01 [-M--] 14 l: [ 1+ 0 1/ 3] *(14 / 87b) 0048 0x030 [z] [x]  
tar -cf script01.tar script01  
mv script01.tar /home/asdoershteyjn/backup/script01.tar
```

Рис. 4.1: Скрипт

Способ использования команд архивации необходимо узнать, изучив справку. (рис. 4.2).





```
TAR(1)                                GNU TAR Manual                                TAR(1)

NAME
tar - an archiving utility

SYNOPSIS
Traditional usage
tar {A|c|d|r|t|u|x}[GnSkUWompsMBiajJzZhPlRvwo] [ARG...]

UNIX-style usage
tar -A [OPTIONS] ARCHIVE ARCHIVE

tar -c [-f ARCHIVE] [OPTIONS] [FILE...]

tar -d [-f ARCHIVE] [OPTIONS] [FILE...]

tar -t [-f ARCHIVE] [OPTIONS] [MEMBER...]

tar -r [-f ARCHIVE] [OPTIONS] [FILE...]

tar -u [-f ARCHIVE] [OPTIONS] [FILE...]

tar -x [-f ARCHIVE] [OPTIONS] [MEMBER...]

Manual page tar(1) line 1 (press h for help or q to quit)
```

Рис. 4.2: Справка

2. Написала пример командного файла, обрабатывающего любое произвольное число аргументов командной строки, в том числе превышающее десять.(рис. 4.3), (рис. 4.4).

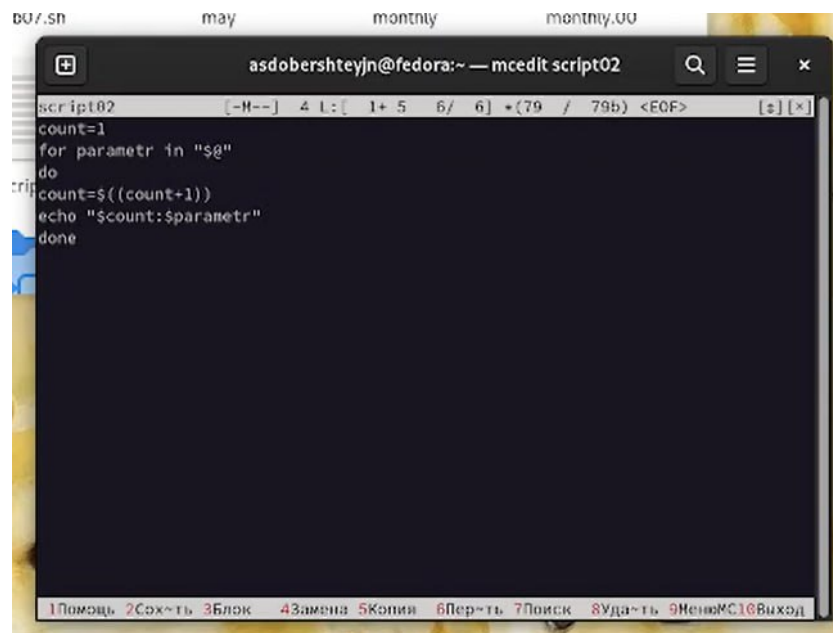


Рис. 4.3: Скрипт

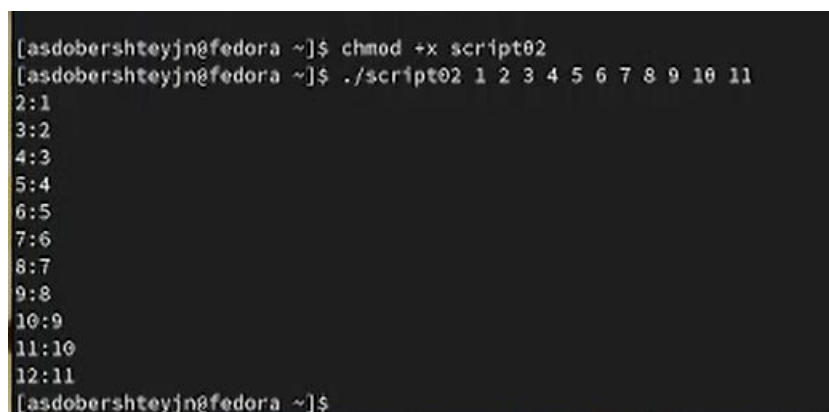


Рис. 4.4: Пример работы скрипта

3. Написала командный файл — аналог команды ls (без использования самой этой команды и команды dir), чтобы он выдавал информацию о нужном каталоге и выводил информацию о возможностях доступа к файлам этого каталога.(рис. 4.5), (рис. 4.6).

```
script03
[ -M-- ] 4 L: [ 1+11 12/ 12] *(230 / 230b) <EOF> [z] [x]
for A in *
do if test -d $A
then echo $A: "is a directory"
else echo -n $A:"is a file and "
<----->if test -w $A
<----->then echo writeable
<----->elif test -r $A
<----->then echo readable
<----->else echo neither readable nor writeable
<----->fi
fi
done
```

Рис. 4.5: Скрипт

```
asdoershteyjn@fedora:~
[asdoershteyjn@fedora ~]$ chmod +x script03
[asdoershteyjn@fedora ~]$ ./script03
abcl: is a file and writeable
april: is a file and writeable
backup: is a directory
bin: is a directory
lab07.sh: is a file and writeable
lab07.sh~: is a file and writeable
may: is a file and writeable
monthly: is a directory
monthly.00: is a directory
script01: is a file and writeable
script02: is a file and writeable
script03: is a file and writeable
text.txt: is a file and writeable
work: is a directory
Видео: is a directory
Документы: is a directory
Загрузки: is a directory
Изображения: is a directory
Музыка: is a directory
Общедоступные: is a directory
```

Рис. 4.6: Пример работы скрипта

4. Написала 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.(рис. 4.7), (рис. 4.8), (рис. 4.9).

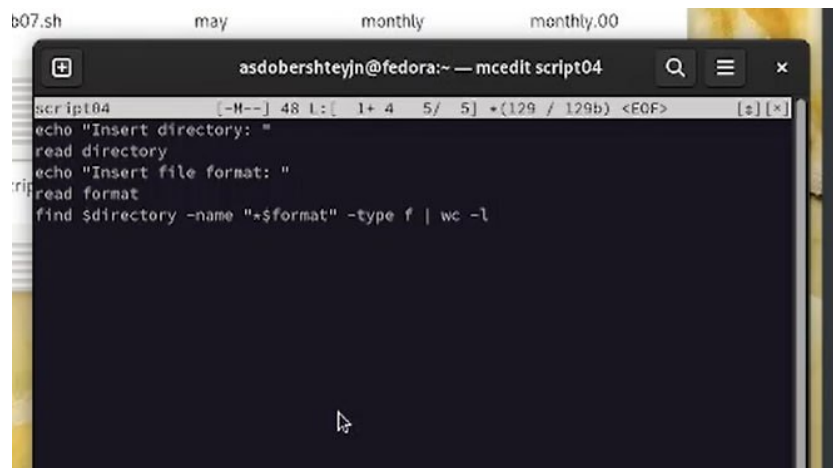


Рис. 4.7: Скрипт



Рис. 4.8: Пример работы скрипта

```
[asdoershteyjn@fedora ~]$ ./script04
Insert directory:
/home/asdoershteyjn
Insert file format:
pdf
23
[asdoershteyjn@fedora ~]$ ./script04
Insert directory:
/home/asdoershteyjn
Insert file format:
.md
97
[asdoershteyjn@fedora ~]$
```

Рис. 4.9: Пример работы скрипта

## 5 Выводы

В ходе выполнения данной лабораторной работы я изучила основы программирования в оболочке ОС UNIX/Linux и научилась писать небольшие командные файлы.