

# Лабораторная работа №12

Программирование в командном процессоре

---

Кадилова М.Р.

15.04.2023

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Кадирова Мехрубон Рахматжоновна
- студент группы НКАбд-04-22
- Российский университет дружбы народов

## Цель работы

---

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

## 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.



## Теоретическое введение

---

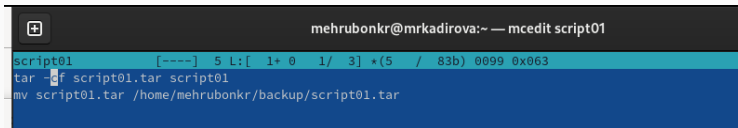
Командный процессор (командная оболочка, интерпретатор команд `shell`) – это программа, позволяющая пользователю взаимодействовать с операционной системой компьютера. В операционных системах типа UNIX/Linux наиболее часто используются следующие реализации командных оболочек: – оболочка Борна (Bourne 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

## Выполнение лабораторной работы

---

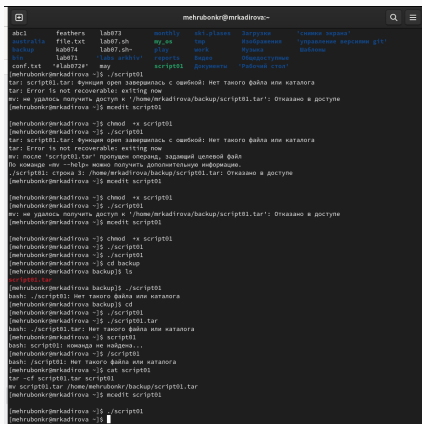
1. Написать скрипт (рис. (fig:001?)), который при запуске будет делать резервную копию самого себя (то есть файла, в котором содержится его исходный код) в другую директорию backup в вашем домашнем каталоге. При этом файл должен архивироваться одним из архиваторов на выбор zip, bzip2 или tar. Способ использования команд архивации необходимо узнать, изучив справку (рис. (fig:002?)).



```
mehrubonkr@mrkadirova:~ — mcedit script01
script01      [----]  5 L:[ 1+ 0  1/  3] *(5  /  83b) 0099 0x063
tar -cf script01.tar script01
mv script01.tar /home/mehrubonkr/backup/script01.tar
```

Рис. 1: Скрипт

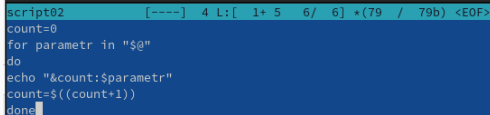
# Выполнение лабораторной работы



```
mehrubonkr@mrkadirova:~  
abc1      feathers  lab073  monthly  ski_places  Загрузка  'снимки экранов'  
australia  file.txt  lab073.sh  my_44    tar         Восстановление  'управление версиями git'  
backup     lab074    lab073.sh  play     work        Измена         'добавление'  
bin        lab071    'labs archive'  reports  backup      Отображение  
conf.txt   'alab0728'  may       script01  документы  'Рабочий стол'  
[mehrubonkr@mrkadirova ~]$ ./script01  
tar: script01.tar: функция open завершилась с ошибкой: Нет такого файла или каталога  
tar: Error is not recoverable: exiting now  
mv: не удалось получить доступ к '/home/mrkadirova/backup/script01.tar': Отказано в доступе  
[mehrubonkr@mrkadirova ~]$ mcedit script01  
[mehrubonkr@mrkadirova ~]$ chmod +x script01  
[mehrubonkr@mrkadirova ~]$ ./script01  
tar: script01.tar: функция open завершилась с ошибкой: Нет такого файла или каталога  
tar: Error is not recoverable: exiting now  
mv: после 'script01.tar' пропущен оператор, задающий целевой файл  
По команде '-mv --help' можно получить дополнительную информацию.  
./script01: строка 3: /home/mrkadirova/backup/script01.tar: Отказано в доступе  
[mehrubonkr@mrkadirova ~]$ mcedit script01  
[mehrubonkr@mrkadirova ~]$ chmod +x script01  
[mehrubonkr@mrkadirova ~]$ ./script01  
mv: не удалось получить доступ к '/home/mrkadirova/backup/script01.tar': Отказано в доступе  
[mehrubonkr@mrkadirova ~]$ mcedit script01  
[mehrubonkr@mrkadirova ~]$ chmod +x script01  
[mehrubonkr@mrkadirova ~]$ ./script01  
[mehrubonkr@mrkadirova ~]$ cd backup  
[mehrubonkr@mrkadirova backup]$ ls  
script01.tar  
[mehrubonkr@mrkadirova backup]$ ./script01  
bash: ./script01: Нет такого файла или каталога  
[mehrubonkr@mrkadirova backup]$ cd  
[mehrubonkr@mrkadirova ~]$ ./script01  
[mehrubonkr@mrkadirova ~]$ ./script01.tar  
bash: ./script01.tar: Нет такого файла или каталога  
[mehrubonkr@mrkadirova ~]$ script01  
bash: script01: команда не найдена...  
[mehrubonkr@mrkadirova ~]$ ./script01  
bash: ./script01: Нет такого файла или каталога  
[mehrubonkr@mrkadirova ~]$ cd script01  
tar -cf script01.tar script01  
mv script01.tar /home/mehrubonkr/backup/script01.tar  
[mehrubonkr@mrkadirova ~]$ mcedit script01  
[mehrubonkr@mrkadirova ~]$ ./script01  
[mehrubonkr@mrkadirova ~]$
```

Рис. 2: Запуск скрипта

2. Написать пример командного файла (рис. (fig:003?)), обрабатывающего любое произвольное число аргументов командной строки, в том числе превышающее десять. Например, скрипт может последовательно распечатывать значения всех переданных аргументов(рис. (fig:004?)).



```
script02 [----] 4 L:[ 1+ 5 6/ 6] *(79 / 79b) <EOF>
count=0
for parametr in "$@"
do
echo "&count:$parametr"
count=$((count+1))
done
```

Рис. 3: Скрипт

```
[mehrubonkr@erkadirova ~]$ chmod +x script02
[mehrubonkr@erkadirova ~]$ ./script02 1 2 3 4 5
&count:1
&count:2
&count:3
&count:4
&count:5
[mehrubonkr@erkadirova ~]$ ./count 1 2 3 4 5 6 7 8 9 10 11
bash: ./count: Нет такого файла или каталога
[mehrubonkr@erkadirova ~]$ ./script02 1 2 3 4 5 6 7 8 9 10 11
&count:1
&count:2
&count:3
&count:4
&count:5
&count:6
&count:7
&count:8
&count:9
&count:10
&count:11
[mehrubonkr@erkadirova ~]$
```

Рис. 4: Запуск скрипта



3. Написать командный файл — аналог команды `ls` (рис. (fig:005?))(без использования самой этой ко- манды и команды `dir`). Требуется, чтобы он выдавал информацию о нужном каталоге и выводил информацию о возможностях доступа к файлам этого каталога (рис. (fig:006?)).

```
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. 5: Скрипт

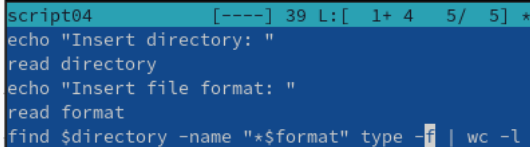
# Выполнение лабораторной работы

```
[mehrubonkr@mrkadirova ~]$ mcedit script03

[mehrubonkr@mrkadirova ~]$ chmod +x script03
[mehrubonkr@mrkadirova ~]$ ./script03
abc1: is a file and writeable
australia: is a directory
backup: is a directory
bin: is a directory
conf.txt: is a file and writeable
feathers: is a file and writeable
file.txt: is a file and writeable
kab074: is a file and writeable
lab071: is a file and writeable
+lab072+: is a file and writeable
lab073: is a file and writeable
lab07.sh: is a file and writeable
lab07.sh-: is a file and writeable
./script03: строка 2: test: labs: ожидается бинарный оператор
labs arkhiv: is a file and ./script03: строка 5: test: labs: ожидается бинарный оператор
./script03: строка 7: test: labs: ожидается бинарный оператор
neither readable nor writeable
may: is a file and writeable
monthly: is a directory
my_os: is a file and readable
play: is a directory
reports: is a directory
script01: is a file and writeable
script02: is a file and writeable
script03: is a file and writeable
ski.places: is a directory
tmp: is a directory
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
./script03: строка 2: test: Рабочий: ожидается бинарный оператор
Рабочий стон: is a file and ./script03: строка 5: test: Рабочий: ожидается бинарный опера
./script03: строка 7: test: Рабочий: ожидается бинарный оператор
neither readable nor writeable
```

Рис. 6: Запуск скрипта

4. Написать командный файл (рис. (fig:007?)), 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. (fig:008?)).



```
script04      [-----] 39 L: [ 1+ 4 5/ 5] *
echo "Insert directory: "
read directory
echo "Insert file format: "
read format
find $directory -name "$format" type -f | wc -l
```

Рис. 7: Скрипт

```
-f, --from      show remote hostname field
-o, --old-style  old style output
-i, --ip-addr   display IP address instead of hostname (if possible)

--help        display this help and exit
-V, --version  output version information and exit

For more details see w(1).
[mehrubonkr@mrkadirova ~]$ mcedit script04

[mehrubonkr@mrkadirova ~]$ chmod +x script04
[mehrubonkr@mrkadirova ~]$ ./script04
Insert directory:
/home/mehrubonkr/
Insert file format:
.txt
find: paths must precede expression: 'type'
0
[mehrubonkr@mrkadirova ~]$
```

Рис. 8: Запуск скрипта

## Выводы

---

Изучили основы программирования в оболочке ОС UNIX/Linux. Научились писать небольшие командные файлы.