

# **Лабораторная работа № 10**

**ОС UNIX. Командные файлы**

**Хамдамова Айжана НКАбд-05-22**

# Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	7
4	Выводы	12
	Список литературы	13

# Список иллюстраций

fignoСоздадим файл и директорию . . . . .	7
fignoКод . . . . .	8
fignoРезультат . . . . .	8
fignoКод в файле . . . . .	9
fignoВызов команды в терминале . . . . .	9
fignoКод . . . . .	10
fignoФайл4 . . . . .	10
fignoВызов файла в терминале . . . . .	11

## Список таблиц

# 1 Цель работы

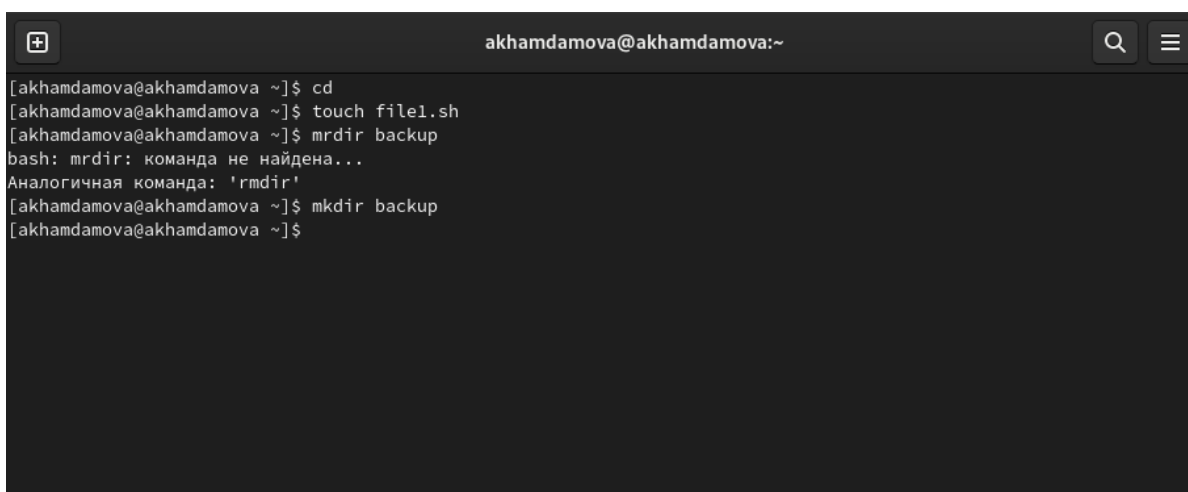
Изучить основы программирования в оболочке ОС UNIX/Linux. Научиться писать небольшие командные файлы.

## 2 Теоретическое введение

Командные процессоры (оболочки) 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 оболочкой 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.

### 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.



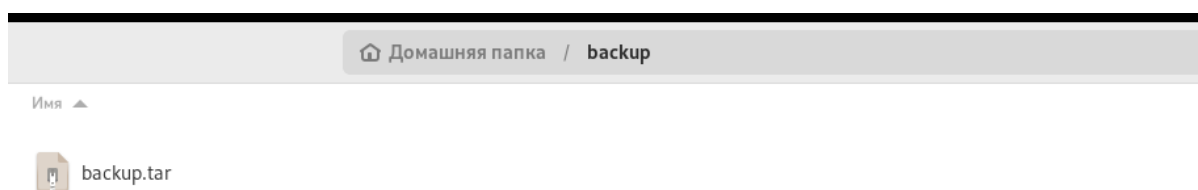
```
akhamdamova@akhamdamova:~  
[akhamdamova@akhamdamova ~]$ cd  
[akhamdamova@akhamdamova ~]$ touch file1.sh  
[akhamdamova@akhamdamova ~]$ mrdir backup  
bash: mrdir: команда не найдена...  
Аналогичная команда: 'rmdir'  
[akhamdamova@akhamdamova ~]$ mkdir backup  
[akhamdamova@akhamdamova ~]$
```

Создадим файл и директорию



The screenshot shows a code editor window with a title bar containing "Открыть", a plus icon, "\*file1.sh", and "Сохранить" with a menu icon. The editor area contains two lines of code: "1 #!/bin/bash" and "2 tar -cvf ~/backup/backup.tar file1|.sh". A dark button labeled "Сохранить текущий файл" is positioned to the right of the code. The background is a light gray.

Код



Результат

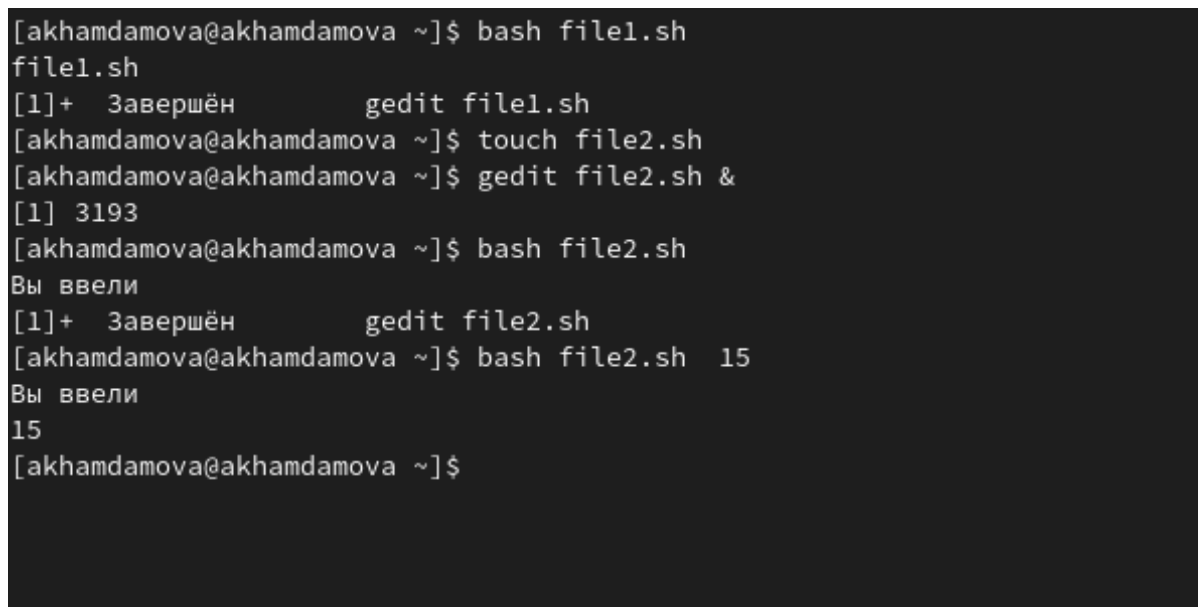
2. . Написать пример командного файла, обрабатывающего любое произвольное число аргументов командной строки, в том числе превышающее десять. Например, скрипт может последовательно распечатывать значения всех переданных аргументов.





```
1 #!/bin/bash
2 echo 'Вы ввели'
3 for A in $*
4 do echo $A
5 done
```

код в файле



```
[akhamdamova@akhamdamova ~]$ bash file1.sh
file1.sh
[1]+  Завершён      gedit file1.sh
[akhamdamova@akhamdamova ~]$ touch file2.sh
[akhamdamova@akhamdamova ~]$ gedit file2.sh &
[1] 3193
[akhamdamova@akhamdamova ~]$ bash file2.sh
Вы ввели
[1]+  Завершён      gedit file2.sh
[akhamdamova@akhamdamova ~]$ bash file2.sh 15
Вы ввели
15
[akhamdamova@akhamdamova ~]$
```

Вызов команды в терминале

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.



```
1 #!/bin/bash
2 for A in *
3 do
4   if test -d "$A"
5   then
6     echo "$A: is a directory"
7   else
8     echo -n "$A: is a file and"
9     if test -w "$A"
10    then
11      echo writeable
12      if test -r "$A"
13      then
14        echo "readable"
15      else
16        echo "neither readable nor writeable"
17      fi
18    fi
19  fi
20 done
```

Код

[Вызов команды в терминале]](image/6.png)

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.



```
1 #!/bin/bash
2 format=""
3 directory=""
4 echo "Напишите формат файла"
5 read format
6 echo "Напишите директорию"
7 read directory
8 find "${directory}" -name ".*${format}" -type f |wc -l
9 ls
```

Файл4

```
[1]+  Завершён      gedit file4.sh
[akhamdamova@akhamdamova ~]$ bash file4.sh
Напишите формат файла
txt
Напишите директорию
/home/akhamdamova
15
backup      file2.sh    lab07.sh~   tutorial    Изображения
bin         file3.sh    lab09.txt   work        Музыка
conf.txt    file4.sh    play        Видео       Общедоступные
feathers     file.txt    ski.plases  Документы  'Рабочий стол'
file1.sh    lab07.sh    text.txt    Загрузки   Шаблоны
[akhamdamova@akhamdamova ~]$
```

Вызов файла в терминале

## 4 Выводы

я изучила основы программирования в оболочке ОС UNIX/Linux. Научилась писать небольшие командные файлы.

## **Список литературы**