

# **Лабораторная работа №10**

**Операционные системы**

Серёгина Ирина Андреевна

# Содержание

1	Цель работы	3
2	Задание	4
3	Теоретическое введение	5
4	Выполнение лабораторной работы	6
5	Выводы	9
	Список литературы	10

# 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.

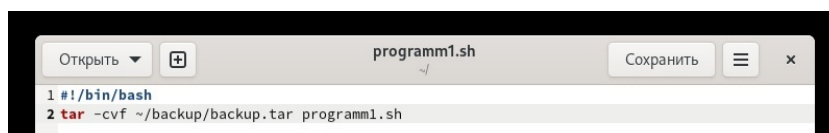
### 3 Теоретическое введение

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.

Командный процессор `bash` обеспечивает возможность использования переменных типа строка символов. Имена переменных могут быть выбраны пользователем. Пользователь имеет возможность присвоить переменной значение некоторой строки символов.

## 4 Выполнение лабораторной работы

Прописываю первую программу (рис. fig. 4.1).



```
1 #!/bin/bash
2 tar -cvf ~/backup/backup.tar programm1.sh
```

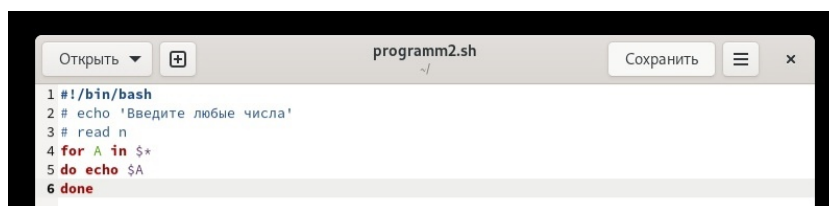
Рис. 4.1: первая программа

Всё работает (рис. fig. 4.2).



Рис. 4.2: выполнение первой программы

Прописываю вторую программу (рис. fig. 4.3.)



```
1 #!/bin/bash
2 # echo 'Введите любые числа'
3 # read n
4 for A in $*
5 do echo $A
6 done
```

Рис. 4.3: вторая программа

Всё работает (рис. fig. 4.4).

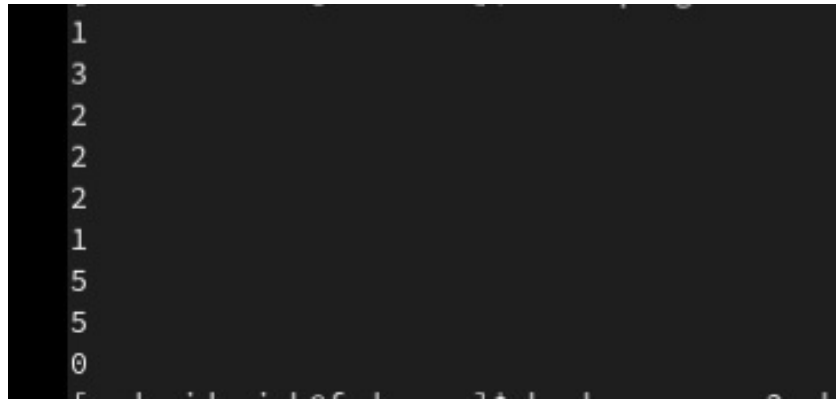


Рис. 4.4: выполнение второй программы

Прописываю третью программу (рис. fig. 4.5).



Рис. 4.5: третья программа

Всё работает (рис. fig. 4.6).

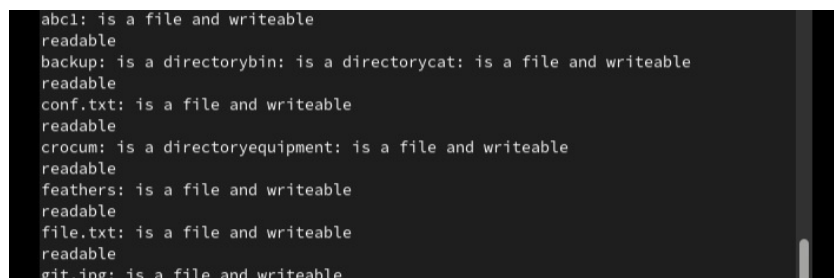


Рис. 4.6: выполнение третьей программы

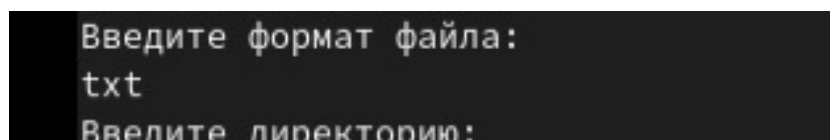
Прописываю четвертую программу (рис. fig. 4.7).



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

Рис. 4.7: четвертая программа

Всё работает (рис. fig. 4.8).



```
Введите формат файла:
txt
Введите директорию:
```

Рис. 4.8: выполнение четвертой программы



## 5 Выводы

Я изучила основы программирования в оболочке ОС UNIX/Linux, научилась писать небольшие командные файлы.

## **Список литературы**