

Лабораторная работа №12

Операционные системы

Краснова К. Г.

02 мая 2025

Российский университет дружбы народов, Москва, Россия

Цель данной лабораторной работы: изучить основы программирования в оболочке ОС UNIX/Linux. Научиться писать небольшие командные файлы.

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

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.

Выполнение лабораторной работы

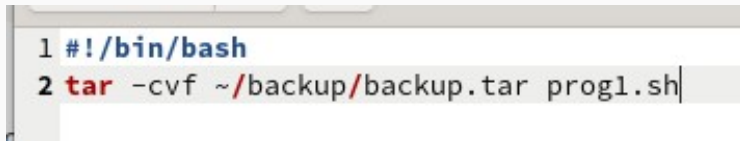
Создаю файл prog1.sh в котором буду писать программу, а также необходимую нам папку.
Запускаю программу (рис. 1).



```
kamilla@fedora:~$ touch prog1.sh
kamilla@fedora:~$ chmod +x prog1.sh
kamilla@fedora:~$ bash prog1.sh
kamilla@fedora:~$ gedit prog1.sh
kamilla@fedora:~$ mkdir backup
kamilla@fedora:~$ bash prog1.sh
prog1.sh
kamilla@fedora:~$
```

Рис. 1: Резервная копия

Код программы, которая делает резервную копию файла (рис. 2).

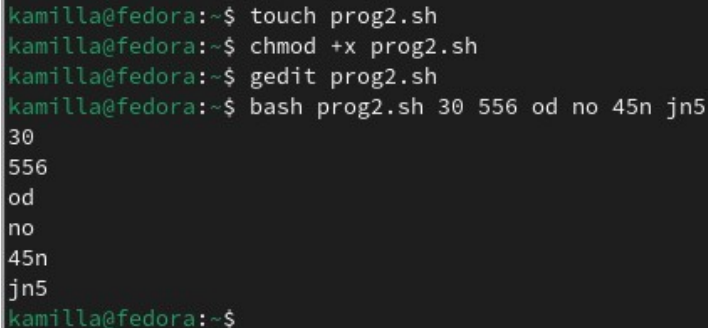
A screenshot of a terminal window with a light gray background. It shows two lines of code. The first line is '1 #!/bin/bash' with the second part in blue. The second line is '2 tar -cvf ~/backup/backup.tar prog1.sh|' with 'tar' in red and a vertical cursor at the end of the line.

```
1 #!/bin/bash
2 tar -cvf ~/backup/backup.tar prog1.sh|
```

Рис. 2: Резервная копия

Выполнение лабораторной работы

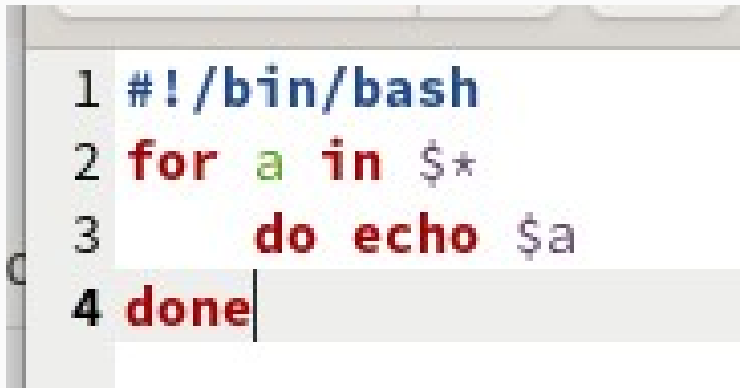
Создаю файл prog2.sh, в котором будет записана программа, обрабатывающая любое произвольное число (рис. 3).

A terminal window with a dark background and green text. The user 'kamilla' is at a 'fedora' machine. They create a file 'prog2.sh' using 'touch', set permissions to '+x' using 'chmod', and open it with 'gedit'. Then they run 'bash prog2.sh' with arguments '30 556 od no 45n jn5'. The script outputs each argument on a new line. The prompt returns to the user.

```
kamilla@fedora:~$ touch prog2.sh
kamilla@fedora:~$ chmod +x prog2.sh
kamilla@fedora:~$ gedit prog2.sh
kamilla@fedora:~$ bash prog2.sh 30 556 od no 45n jn5
30
556
od
no
45n
jn5
kamilla@fedora:~$
```

Рис. 3: Обработка чисел

Пример программы (рис. 4).

A screenshot of a terminal window with a light gray background. It displays a four-line shell script. Line 1 is the shebang `#!/bin/bash`. Line 2 is `for a in $*`. Line 3 is `do echo $a`. Line 4 is `done`. The script is color-coded: `#!/bin/bash` is blue, `for` is red, `a` is green, `in` is red, `$*` is black, `do` is red, `echo` is red, `$a` is black, and `done` is red. A vertical cursor is positioned at the end of the fourth line.

```
1 #!/bin/bash
2 for a in $*
3     do echo $a
4 done
```

Рис. 4: Обработка чисел

Выполнение лабораторной работы

Далее создаю файл prog3.sh, добавляю в него права и записываю код, далее запускаю его (рис. 5).

```
kamilla@fedora:~$ touch prog3.sh
kamilla@fedora:~$ chmod +x prog3.sh
kamilla@fedora:~$ gedit prog3.sh
kamilla@fedora:~$ bash prog3.sh backup/
abc1: is a file and writeable
readable
Arseny^ is a directory
australia^ is a directory
backup^ is a directory
bin^ is a directory
conf.txt: is a file and writeable
```

Выполнение лабораторной работы

Пример кода - аналога ls (рис. 6).

```
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 or writeable"
17            fi
18        fi
19    fi
20 done
```

Так же создаю файл prog4.sh, принимающий в качестве аргумента формат файла (рис. 7).

```
kamilla@fedora:~$ touch prog4.sh
kamilla@fedora:~$ chmod +x prog4.sh
kamilla@fedora:~$ gedit prog4.sh
kamilla@fedora:~$ bash prog4.sh
Напишите формат файла
txt
Напишите директорию
/home/kamilla
40
abc1      doklad      image.zip   my_os       play
Arseny    Downloads  _index.md   pandoc-2.18 prog1.s
australia feathers    '#lab07.sh#' pandoc-2.18-linux-amd64.tar.gz prog2.s
backup    file.txt    lab07.sh    pandoc-crossref prog3.s
bin        fun         lab07.sh~   pandoc-crossref.1 prog4.s
conf.txt   git-extended LICENSE     pandoc-crossref-Linux.tar.xz reports
Documents image      monthly     Pictures     ski.pla
kamilla@fedora:~$
```

Рис. 7: Чтение формата файла

Код программы, считывающей формат файла (рис. 8).

```
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|
```

Рис. 8: чтение формата файла

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