

Отчёт по лабораторной работе

Грузинова Елизавета Константиновна; НКНбд-02-21

Программирование в командном
процессоре ОС UNIX. Командные
файлы.

Изучить основы программирования в оболочке ОС UNIX/Linux и 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 командной строки.

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.

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. (рис. 1, 2, 3, 4)

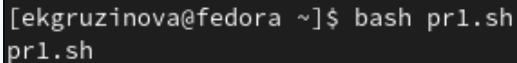
A terminal window with a black background and white text. The prompt character is a dollar sign '\$'. The command 'touch pr1.sh' is entered and displayed on the line.

```
$ touch pr1.sh
```

Figure 1: Создание файла pr1.sh

```
1 #!/bin/bash
2 mkdir backup
3 cp pr1.sh ~/backup
4 tar -cvf ~/backup/backup.tar pr1.sh
```

Figure 2: Программный код файла pr1.sh

A terminal window with a black background and white text. The prompt is [ekgruzinova@fedora ~]\$. The user has entered the command bash pr1.sh, and the terminal has responded with pr1.sh on a new line.

```
[ekgruzinova@fedora ~]$ bash pr1.sh  
pr1.sh
```

Figure 3: Файл в действии

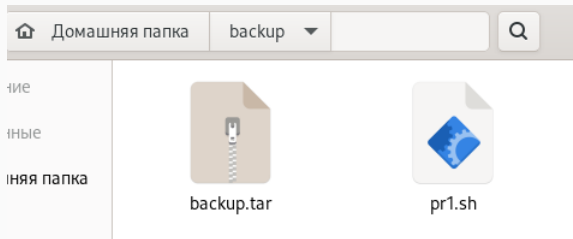


Figure 4: Созданный программой архив файла

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. (рис. 5, 6)

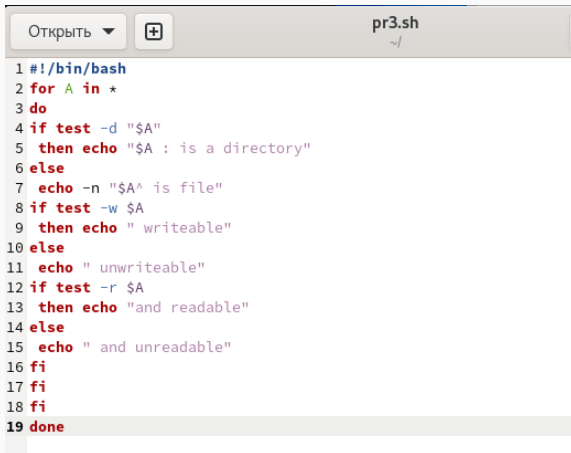
```
#!/bin/bash  
for A in $*  
do echo $A  
done
```

Figure 5: Программный код файла pr2.sh

```
[ekguzinova@fedora ~]$ bash pr2.sh 123 5543 3398 65 2 3  
123  
5543  
3398  
65  
2  
3
```

Figure 6: Скрипт в действии, в данном случае, он записывает необходимые аргументы и распечатывает их обратно

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. (рис. 7, 8)



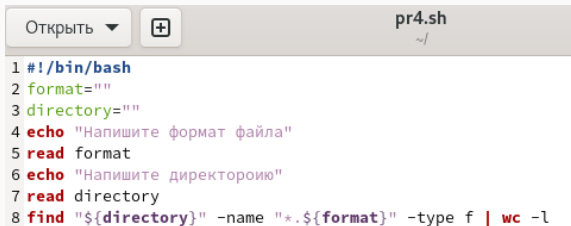
```
1 #!/bin/bash
2 for A in *
3 do
4 if test -d "$A"
5 then echo "$A : is a directory"
6 else
7 echo -n "$A^ is file"
8 if test -w $A
9 then echo " writeable"
10 else
11 echo " unwriteable"
12 if test -r $A
13 then echo "and readable"
14 else
15 echo " and unreadable"
16 fi
17 fi
18 fi
19 done
```

Figure 7: Программный код файла pr3.sh

```
lab07.sh^ is file writeable
lab07.sh~^ is file writeable
lab07.zip^ is file writeable
Lab08.zip^ is file writeable
labo71^ is file writeable
monthly : is a directory
monthly1 : is a directory
my_os^ is file unwriteabl
and readable
play : is a directory
pr1.sh^ is file writeable
pr2.sh^ is file writeable
pr3.sh^ is file writeable
```

Figure 8: Вывод информации о каждом файле или каталоге

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. (рис. 9, 10)



```
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
```

Figure 9: Программный код файла pr4.sh

```
lekgruzinova@fedora ~]$ bash pr4.sh
Напишите формат файла
txt
Напишите директорию
/home/ekgruzinova
ll
  australia      feathers  lab074    monthly1  README.md  Изображения
  backup         file.txt  lab07.sh  my_os     reports    лаба7
  bin            homet    lab07.sh~ play      ski.plases лаба8
```

Figure 10: Вывод количества файлов в /home/gruzinova формата .txt

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

Спасибо за внимание.