

Лабораторная работа № 10

**Программирование в командном процессоре ОС UNIX. Командные
файлы**

Шулуужук Айраана Вячеславовна НПИбд-02-22

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
3.1	Командные процессоры (оболочки)	7
4	Выполнение лабораторной работы	9
5	Выводы	13

Список иллюстраций

4.1	создание каталога и файла	9
4.2	скрипт lab10_1	9
4.3	результат запуска скрипта 1	10
4.4	скрипт lab10_2.sh	10
4.5	результат запуска скрипта 2	10
4.6	скрипт lab10_3.sh	11
4.7	результат запуска скрипта 3	11
4.8	скрипт lab10_4.sh	12
4.9	результат запуска скрипта 4	12

Список таблиц

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

3.1 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 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.

4 Выполнение лабораторной работы

Создадим каталог backup и файл lab10_1.sh. Отроем редактор emacs (рис. 4.1)

```
[avshuluuzhuk@fedora ~]$ mkdir backup  
[avshuluuzhuk@fedora ~]$ touch lab10_1.sh  
[avshuluuzhuk@fedora ~]$  
[avshuluuzhuk@fedora ~]$ emacs
```

Рис. 4.1: создание каталога и файла

Напишем скрипт, который при запуске будет делать резервную копию самого себя (рис. 4.2)

```
#!/bin/bash  
cp ~/lab10_1.sh backup/lab10_1.sh  
cd backup  
zip backup.zip lab10_1.sh  
rm lab10_1.sh
```

Рис. 4.2: скрипт lab10_1

Присвоим данному файлу права на исполнение и запустим его (рис. 4.3)

```

[avshuluuzhuk@fedora ~]$ chmod u+x lab10_1.sh
[avshuluuzhuk@fedora ~]$ ls -l lab10_1.sh
-rwxr--r--. 1 avshuluuzhuk avshuluuzhuk 96 anp  4 18:41 lab10_1.sh
[avshuluuzhuk@fedora ~]$ ./lab10_1.sh
adding: lab10_1.sh (deflated 41%)
[avshuluuzhuk@fedora ~]$ ls backup
backup.zip
[avshuluuzhuk@fedora ~]$ 

```

Рис. 4.3: результат запуска скрипта 1

Создаем новый файл lab10_2.sh и напишем пример командного файла, обрабатывающее любое число аргументов командной строки (рис. 4.4)

```

#!/bin/bash
count=1
while [ -n "$1" ]
do
    echo "$count: $1"
    count=$((count+1))
    shift
done

```

Рис. 4.4: скрипт lab10_2.sh

Запустим следующий исполняемый файл и введем 10 или более произвольных чисел (рис. 4.5)

```

[avshuluuzhuk@fedora ~]$ ./lab10_2.sh 2 4 1 6 3 7 9 12 56 34
1: 2
2: 4
3: 1
4: 6
5: 3
6: 7
7: 9
8: 12
9: 56
10: 34

```

Рис. 4.5: результат запуска скрипта 2

Создаем новый файл lab10_3.sh для скрипта 3. Напишем командный файл, который воспроизводит аналог команды ls (рис. 4.6)

```
for A in *
do if test -d $A
then echo $A: is a directory
else echo -n $A: "is a file and "
if test -x $A
then echo executable
elif test -w $A
then echo writeable
elif test -r $A
then echo readable
else echo neither readable or writeble
fi
done
```

Рис. 4.6: скрипт lab10_3.sh

Запустим этот файл и посмотрим содержимое домашнего каталога (рис. 4.7)

```
[avshuluuzhuk@fedora ~]$ ./lab10_3.sh backup
backup: is a directory
bin: is a directory
lab10_1.sh: is a file and executable
lab10_1.sh~: is a file and writeable
lab10_2.sh: is a file and executable
lab10_2.sh~: is a file and executable
lab10_3.sh: is a file and executable
lab10_3.sh~: is a file and executable
programs: 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
```

Рис. 4.7: результат запуска скрипта 3

В файле lab10_4 напишем скрипт, который получает в качестве аргумента командной строки формат файла (.txt, .doc, .jpg, .pdf и т.д.) и вычисляет количество таких файлов в каталоге (рис. 4.8)

```

echo Input a directory
read dir
echo Input a file format
read format
find $dir -maxdepth 1 -name "$format" -type f | wc -l

```

Рис. 4.8: скрипт lab10_4.sh

Запустим файл и посмотрим количество файлов формата .txt в домашнем каталоге (рис. 4.9)

```

[avshuluuzhuk@fedora ~]$ ./lab10_4.sh
Input a directory
/home/avshuluuzhuk
Input a file format
.txt
3
[avshuluuzhuk@fedora ~]$ ls
backup      lab10_1.sh~  lab10_3.sh~  programs  Документы
bin         lab10_2.sh  lab10_4.sh  text.txt  Загрузки
conf.txt    lab10_2.sh~  lab10_4.sh~  work      Изображения
lab10_1.sh  lab10_3.sh  new.txt     Видео     Музыка
[avshuluuzhuk@fedora ~]$

```

Рис. 4.9: результат запуска скрипта 4

5 Выводы

В ходе выполнения работы мы изучили основы программирования в оболочке ОС UNIX/Linux и научились писать небольшие командные файлы