

Лабораторная работа №12

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

Буллер Татьяна Александровна

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
2.1	Резервное копирование	5
2.2	Последовательная печать аргументов	6
2.3	Аналог ls	7
2.4	Счетчик-фильтр	8
3	Выводы	10

Список иллюстраций

2.1	Скрипт резервного копирования	5
2.2	Результат выполнения скрипта	5
2.3	Скрипт печати аргументов	6
2.4	Результат выполнения скрипта	6
2.5	Скрипт распечатки файлов	7
2.6	Результат выполнения скрипта	8
2.7	Скрипт счетчика	8
2.8	Результат выполнения скрипта	9

1 Цель работы

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

2 Выполнение лабораторной работы

2.1 Резервное копирование

Скрипт для резервного копирования можно написать в одну строку: для этого используем команду архиватора, которой укажем создать файл в заранее созданной директории backup, и подадим в качестве второго аргумента имя исполняемого файла (\$0). Для того, чтобы все резервные копии различались между собой, пропишем первому аргументу сохранять файл не просто со статичным названием, а добавляя дату создания в конец (переменная дата в формате год+месяц+число-час+минута+секунда).

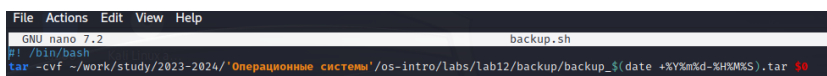
A screenshot of a terminal window showing the nano text editor. The editor is open to a file named 'backup.sh'. The content of the file is a single line of code: `tar -cvf ~/work/study/2023-2024/'Операционные системы'/os-intro/labs/lab12/backup/backup_$(date +%Y%m%d-%H%M%S).tar $0`. The terminal window has a menu bar with 'File', 'Actions', 'Edit', 'View', and 'Help'. The status bar at the bottom shows 'GNU nano 7.2' and 'backup.sh'.

Рис. 2.1: Скрипт резервного копирования

После исполнения скрипта проверим директорию backup командой ls: видим, что создался файл архива резервной копии, в названии которого прописана дата его создания.

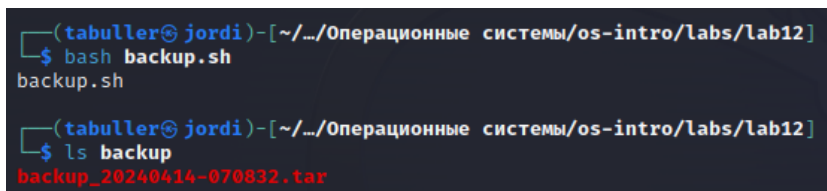
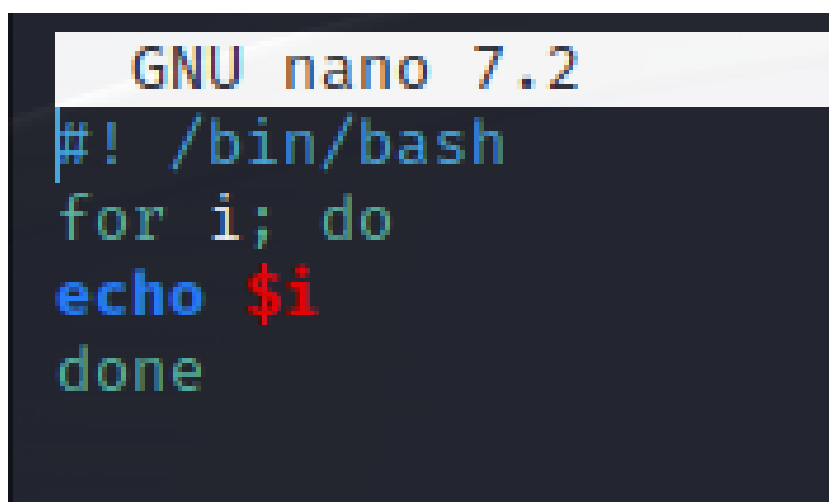
A screenshot of a terminal window showing the execution of the backup script. The prompt is `(tabuller@jordi)-[~/../Операционные системы/os-intro/labs/lab12]`. The user enters `$ bash backup.sh`, and the prompt changes to `backup.sh`. Then the user enters `$ ls backup`, and the output is `backup_20240414-070832.tar`.

Рис. 2.2: Результат выполнения скрипта

2.2 Последовательная печать аргументов

Мы уже обрабатывали аргумент \$0 в предыдущем скрипте. Теперь обработаем все, что будет подано вместе с командой запуска скрипта: проще всего для этого использовать цикл. Пока значение *i* существует, будет печататься аргумент с соответствующим номером.

A screenshot of a terminal window showing the GNU nano 7.2 text editor. The editor contains a shell script with the following code:

```
#!/bin/bash
for i; do
echo $i
done
```

Рис. 2.3: Скрипт печати аргументов

Подадим на печать сколько угодно много аргументов и проверим, справляется ли скрипт с поставленной задачей:

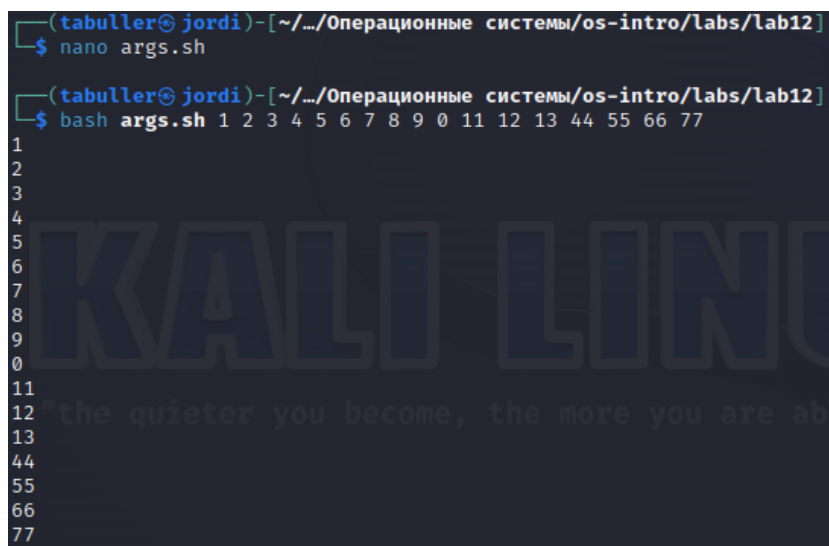
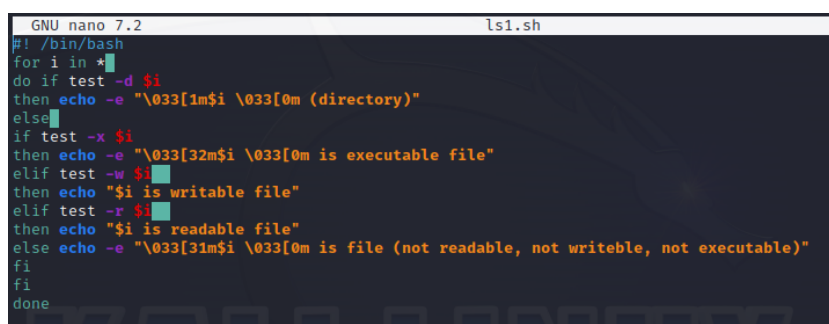
A screenshot of a terminal window showing the execution of the script. The prompt is (tabuller@jordi)-[~/.../Операционные системы/os-intro/labs/lab12]. The user enters 'nano args.sh' and then 'bash args.sh 1 2 3 4 5 6 7 8 9 0 11 12 13 44 55 66 77'. The output shows the numbers 1 through 77 printed on separate lines. A large 'KALI LINUX' watermark is visible in the background. The prompt is (tabuller@jordi)-[~/.../Операционные системы/os-intro/labs/lab12].

Рис. 2.4: Результат выполнения скрипта

2.3 Аналог ls

Работая с этим скриптом будем использовать цикл для всех файлов текущего каталога. Сперва проверим, является ли файл директорией (test -d) и если да, то выведем информацию об этом, выделив название файла жирным шрифтом. Все остальные файлы, очевидно, являются файлами. Проверим их на возможность исполнения, записи или чтения (-x -w -r) и выведем соответствующие сообщения. Если файл с разрешением на запись - выделим его зеленым, если ни одного из разрешений у файла нет - выделим красным.



```
GNU nano 7.2 ls1.sh
#!/bin/bash
for i in *
do if test -d $i
then echo -e "\033[1m$i \033[0m (directory)"
else
if test -x $i
then echo -e "\033[32m$i \033[0m is executable file"
elif test -w $i
then echo -e "$i is writable file"
elif test -r $i
then echo -e "$i is readable file"
else echo -e "\033[31m$i \033[0m is file (not readable, not writeble, not executable)"
fi
fi
done
```

Рис. 2.5: Скрипт распечатки файлов

Проверим: сам скрипт выполняется. Для того, чтобы проверить корректность отображения цветов, изменим разрешения у пары файлов и снова посмотрим на вывод.

```

(tabuller@jordi)-[~/../Операционные системы/os-intro/labs/lab12]
$ bash ls1.sh
args.sh is writable file
backup (directory)
backup.sh is writable file
ls1.sh is writable file
presentation (directory)
report (directory)

(tabuller@jordi)-[~/../Операционные системы/os-intro/labs/lab12]
$ chmod +x args.sh

(tabuller@jordi)-[~/../Операционные системы/os-intro/labs/lab12]
$ chmod -rwx backup.sh

(tabuller@jordi)-[~/../Операционные системы/os-intro/labs/lab12]
$ bash ls1.sh
args.sh is executable file
backup (directory)
backup.sh is file (not readable, not writeble, not executable)
ls1.sh is writable file
presentation (directory)
report (directory)

```

Рис. 2.6: Результат выполнения скрипта

2.4 Счетчик-фильтр

Снова работаем с аргументами на входе. Первым аргументом принимаем формат, вторым - зону поиска. Для реализации скрипта сохраним в массив вывод команды `grep` по результату `ls` нужного каталога и выведем на экран количество элементов массива.

```

GNU nano 7.2
#!/bin/bash
x=( $(ls $2 | grep "${1}$") )
echo ${#x[@]}

```

Рис. 2.7: Скрипт счетчика

Проверим для рабочей директории (не подавая второго аргумента) и для случайно выбранной директории. В обоих случаях скрипт справляется с поставленной задачей корректно.


```

(tabuller@jordi)-[~/../Операционные системы/os-intro/labs/lab12]
$ ls ~/Desktop/lab11/report/image
10.png 12.png 14.png 1.png 3.png 5.png 7.png 9.png
11.png 13.png 15.png 2.png 4.png 6.png 8.png

(tabuller@jordi)-[~/../Операционные системы/os-intro/labs/lab12]
$ bash count.sh .png ~/Desktop/lab11/report/image
15

(tabuller@jordi)-[~/../Операционные системы/os-intro/labs/lab12]
$ ls
args.sh backup backup.sh count.sh ls1.sh presentation report

(tabuller@jordi)-[~/../Операционные системы/os-intro/labs/lab12]
$ bash count.sh .sh
4

```

Рис. 2.8: Результат выполнения скрипта

3 Выводы

Изучены основы программирования в оболочке ОС UNIX/Linux. Приобретены навыки написания небольших командных файлов.