

Лабораторная работа №12

Программирование в командном процессоре ОС UNIX. Командные файлы

Буллер Т. А.

14 апреля 2024

Российский университет дружбы народов, Москва, Россия

Информация

- Буллер Татьяна Александровна
- студент группы НБИбд-01-23
- Российский университет дружбы народов

Вводная часть

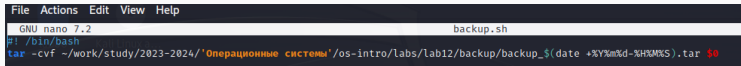
- виртуальная машина Kali Linux
- текстовый редактор nano
- командная оболочка bash

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

- виртуальная машина Kali Linux
- текстовый редактор nano
- командная оболочка bash
- Процессор pandoc для входного формата Markdown
- Результирующие форматы
 - pdf
 - html
- Автоматизация процесса создания: Makefile

Выполнение лабораторной работы

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).



The image shows a terminal window with the nano text editor open. The editor's title bar indicates it is editing a file named 'backup.sh'. The menu bar at the top includes 'File', 'Actions', 'Edit', 'View', and 'Help'. The editor content shows the first line of the script: '#!/bin/bash' followed by a 'tar' command. The 'tar' command is used to create a backup archive named 'backup_\$(date +%Y%m%d-%H%M%S).tar' in the directory '/work/study/2023-2024/Операционные системы/os-intro/labs/lab12/backup/'. The command uses the flags '-cvf' for create, verbose, and file. The prompt '\$0' is visible at the end of the line.

```
File Actions Edit View Help
GNU nano 7.2 backup.sh
#!/bin/bash
tar -cvf ~/work/study/2023-2024/'Операционные системы'/os-intro/labs/lab12/backup/backup_$(date +%Y%m%d-%H%M%S).tar $0
```

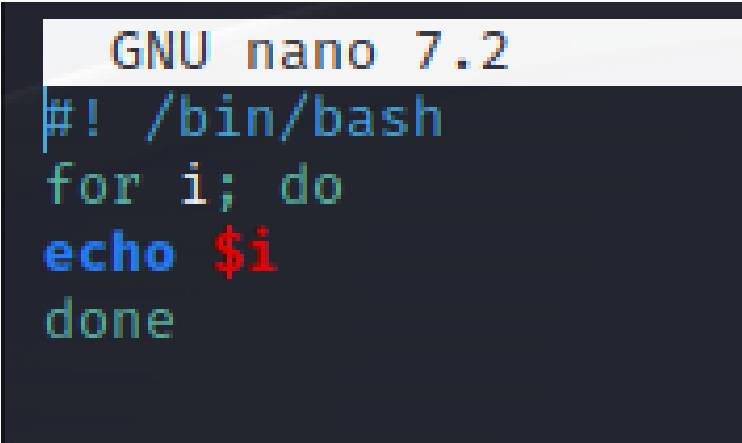
Рис. 1: Скрипт резервного копирования

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

```
(tabuller@jordi)-[~/.../Операционные системы/os-intro/labs/lab12]  
$ bash backup.sh  
backup.sh  
  
(tabuller@jordi)-[~/.../Операционные системы/os-intro/labs/lab12]  
$ ls backup  
backup_20240414-070832.tar
```

Рис. 2: Результат выполнения скрипта

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.



```
GNU nano 7.2
#!/bin/bash
for i; do
echo $i
done
```

A screenshot of a terminal window with a dark background. At the top, a light-colored banner displays 'GNU nano 7.2'. Below this, a shell script is being edited. The script starts with a shebang line '#!/bin/bash', followed by a 'for' loop 'for i; do'. Inside the loop, there is a command 'echo \$i' where '\$i' is highlighted in red. The loop ends with 'done'.

Рис. 3: Скрипт печати аргументов

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

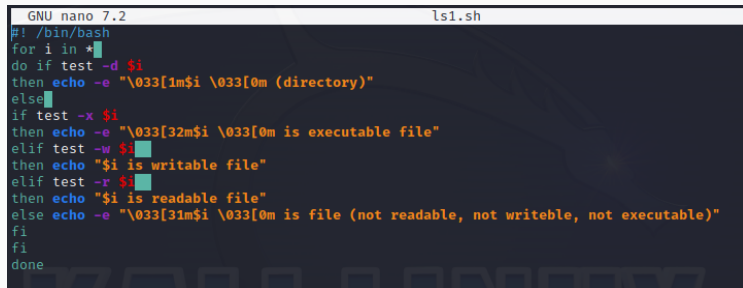
Последовательная печать аргументов

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

(tabuller@jordi)-[~/../Операционные системы/os-intro/labs/lab12]
$ bash args.sh 1 2 3 4 5 6 7 8 9 0 11 12 13 44 55 66 77
1
2
3
4
5
6
7
8
9
0
11
12 "the quieter you become, the more you are able to hear"
13
44
55
66
77
```

Рис. 4: Результат выполнения скрипта

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 $i
then echo "$i is writable file"
elif test -r $i
then echo "$i is readable file"
else echo -e "\033[31m$i \033[0m is file (not readable, not writeble, not executable)"
fi
done
```

Рис. 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. 6: Результат выполнения скрипта

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. 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
"the quieter you become, the more you are able to hear"
(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. 8: Результат выполнения скрипта

Выводы

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