

Лабораторная работа №11. Программирование в командном процессоре ОС UNIX. Командные файлы.

Гекишева Анастасия Дмитриевна, НБИ-01-20,
30 апреля, 2021

¹RUDN University, Moscow, Russian Federation

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;
2. Написать пример командного файла, обрабатывающего любое произвольное число аргументов командной строки, в том числе превышающее десять.
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.

Выполнение 1-го пункта задания

```
adkekisheva@dk8n78 ~ $ man tar
adkekisheva@dk8n78 ~ $ tar --help
Использование: tar [ПАРАМЕТР...] [ФАЙЛ]...
GNU 'tar' saves many files together into a single tape or disk archive, and can
restore individual files from the archive.

Examples:
  tar -cf archive.tar foo bar    # Create archive.tar from files foo and bar.
  tar -tvf archive.tar           # List all files in archive.tar verbosely.
  tar -xf archive.tar            # Extract all files from archive.tar.

Имя выбранного локального файла:
```

Рис. 1: Справка tar

-c, --create	к архиву создание нового архива
-d, --diff, --compare	поиск различий между архивом и файловой системой

Рис. 2: Опция -с - создание

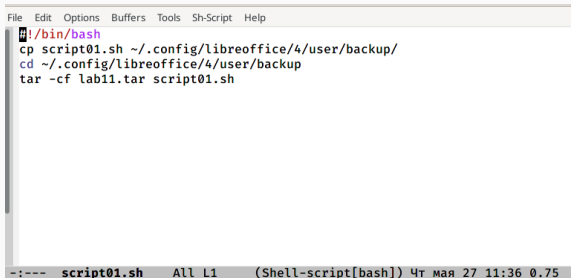
Опция -f - применение файла или устройство АРХИВ

Выбор и переключение устройств:

<code>-f, --file=АРХИВ</code>	использовать файл или устройство АРХИВ
<code>--force-local</code>	файл архива является локальным, даже если содержит двоеточие

Рис. 3: Опция -f - применение файла или устройство АРХИВ

Написание скрипта



The image shows a terminal window with a menu bar at the top containing 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The terminal prompt is `#!/bin/bash`. Below the prompt, three lines of commands are entered: `cp script01.sh ~/.config/libreoffice/4/user/backup/`, `cd ~/.config/libreoffice/4/user/backup`, and `tar -cf lab11.tar script01.sh`. The terminal status bar at the bottom displays `-- script01.sh All L1 (Shell-script[bash]) Чт мая 27 11:36 0.75`.

```
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
cp script01.sh ~/.config/libreoffice/4/user/backup/
cd ~/.config/libreoffice/4/user/backup
tar -cf lab11.tar script01.sh
-- script01.sh All L1 (Shell-script[bash]) Чт мая 27 11:36 0.75
```

Рис. 4: Написание скрипта

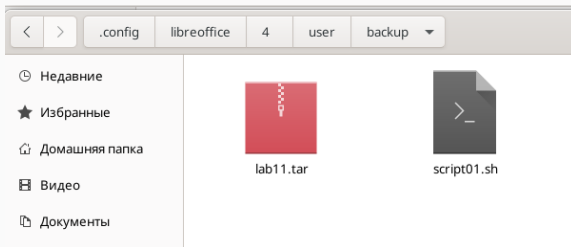
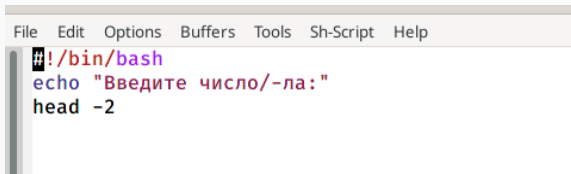


Рис. 5: Результат

Выполнение 2-го пункта задания

1. Воспользоваться head -(количество строк для обработки)

A screenshot of a terminal window with a menu bar at the top containing 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The terminal content shows a shell script starting with a shebang line '#!/bin/bash' in purple and black. The next line is 'echo "Введите число/-ла:"' in black and red, and the final line is 'head -2' in black.

```
#!/bin/bash
echo "Введите число/-ла:"
head -2
```

Рис. 6: Командный файл, обрабатывающий любое произвольное число аргументов командной строки №1

```
adkekisheva@dk8n78 ~ $ touch script02.sh
adkekisheva@dk8n78 ~ $ chmod +x script02.sh
adkekisheva@dk8n78 ~ $ emacs script02.sh
adkekisheva@dk8n78 ~ $ bash script02.sh
Введите число/-ла:
2 3 10 20
2 3 10 20
adkekisheva@dk8n78 ~ $ emacs script02.sh
adkekisheva@dk8n78 ~ $ bash script02.sh
Введите число/-ла:
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
6 30 31 56
6 30 31 56
```

Рис. 7: Результат работы head

2. Ввести `$*`, что означает что вызывая командой файл нам не выйдет предложение ввести символы, их мы будем должны ввести сами в командной строке.

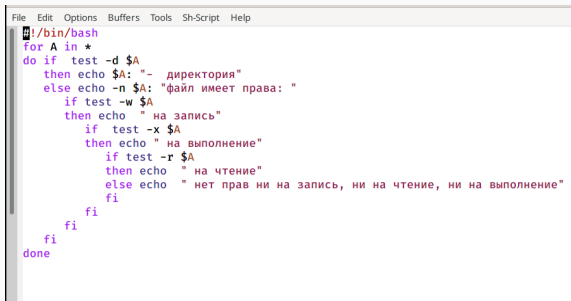
```
#!/bin/bash  
echo "Обработанные числа: $*"
```

```
Cadkeki@dk8n78 ~$ emacs 2.sh  
^Cadkeki@dk8n78 ~$ emacs 2.sh  
^Cadkeki@dk8n78 ~$ bash 2.sh 2 3 10 23 4 5 6 7 8  
Обработанные числа: 2 3 10 23 4 5 6 7 8
```

Рис. 8: Результат работы S*

Выполнение 3-го пункта задания

Командный файл, выводящий информацию о правах доступа



```
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
for A in *
do if test -d $A
then echo $A: "- директория"
else echo -n $A: "файл имеет права: "
if test -w $A
then echo " на запись"
if test -x $A
then echo " на выполнение"
if test -r $A
then echo " на чтение"
else echo " нет прав ни на запись, ни на чтение, ни на выполнение"
fi
fi
fi
done
```

Рис. 9: Командный файл, выводящий информацию о правах доступа

Результат выполнения командного файла

```
adkekisheva@dk8n78:~  
adkekisheva@dk8...  adkekisheva@dk8...  adkekisheva@dk8...  adkekisheva@dk8...  
adkekisheva@dk8n78 ~ $ emacs script03.sh  
adkekisheva@dk8n78 ~ $ bash script03.sh  
1.sh: файл имеет права: на запись  
на выполнение  
на чтение  
2.sh: файл имеет права: на запись  
на выполнение  
на чтение  
2.sh~: файл имеет права: на запись  
на выполнение  
на чтение  
3.sh: файл имеет права: на запись  
на выполнение  
на чтение  
4.sh: файл имеет права: на запись  
на выполнение  
на чтение  
abc1: файл имеет права: на запись  
addition.txt: файл имеет права: на запись  
a.sh: файл имеет права: на запись  
на выполнение  
на чтение  
a.sh~: файл имеет права: на запись  
conf.txt: файл имеет права: на запись  
euler2.sage: файл имеет права: на запись  
euler.sage: файл имеет права: на запись  
feathers: файл имеет права: на запись  
file.txt: файл имеет права: на запись  
GNUstep: - директория  
lab07.sh: файл имеет права: на запись  
lab11.tar: файл имеет права: на запись  
laboratory: - директория  
Makefile: файл имеет права: на запись  
may: файл имеет права: на запись  
my_os: файл имеет права: prog.csr: файл имеет права: на запись  
Programma: файл имеет права: на запись  
на выполнение  
на чтение  
public: - директория  
public.html: - директория
```

Выполнение 4-го пункта задания

Командный файл, который получает в качестве аргумента командной строки формат файла и вычисляет количество таких файлов в указанной директории

A screenshot of a terminal window with a menu bar (File, Edit, Options, Buffers, Tools, Sh-Script, Help) and a scroll bar on the left. The terminal contains a shell script with the following lines:

```
#!/bin/bash
format=' '
directory=' '
echo 'Введите формат'
read format
echo 'Введите директорию'
read directory
cd "$directory"
find "$directory" -name ".*$format" -type f | wc -l
ls
```

Рис. 11: Командный файл, который получает в качестве аргумента командной строки формат файла и вычисляет количество таких файлов в указанной директории

Результат выполнения командного файла

```
adkekiheva@dk8n78 ~ $ bash 4.sh
Введите формат
sh
Введите директорию
/afs/.dk.sci.pfu.edu.ru/home/a/d/adkekiheva
11
1.sh   addition.txt  file.txt    my_os      script01.sh~  ser.sage    Изображения
2.sh   a.sh            GNUstep    prog.ccp   script02.sh~  skr.sh~     Музыка
3.sh~  a.sh~           lab07.sh   Programma  script02.sh~  tmp         Общедоступные
3.sh   conf.txt       lab11.tar  public     script03.sh~  v           ОС
4.sh   euler2.sage   laboratory public_html script03.sh~  Видео       'Рабочий стол'
4.sh~  euler.sage       Makefile   q          script04.sh~  script04.sh~ Документы
abc1   feathers      may        script01.sh script04.sh~  Загрузки
adkekiheva@dk8n78 ~ $
```

Рис. 12: Результат выполнения командного файла

Я изучила основы программирования в оболочке ОС UNIX/Linux, научилась писать командные файлы и скрипты.

1. Ссылка 1
2. Ссылка 2