

Лабораторная работа №12

Презентация

Устинова В. В.

27 апреля 2025

Российский университет дружбы народов, Москва, Россия

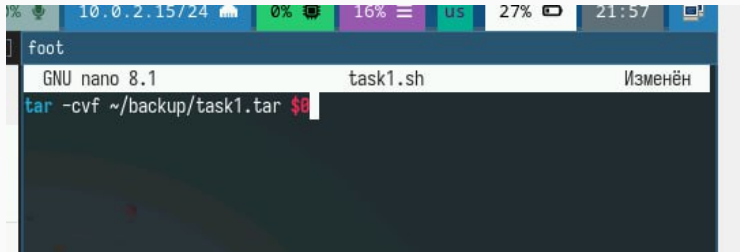
Информация

- Устинова Виктория Вадимовна
- студент НПИбд-01-24
- Российский университет дружбы народов

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 (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 ко-

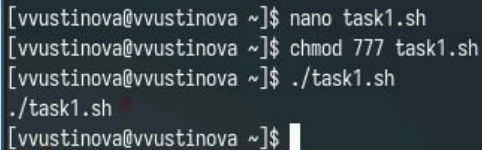
Написать скрипт, который будет делать резервную копию самого себя



The image shows a terminal window with a top status bar displaying network status (10.0.2.15/24), battery level (0%), CPU usage (16%), memory usage (27%), and time (21:57). The terminal title is 'foot'. Inside the terminal, the GNU nano 8.1 editor is open, editing a file named 'task1.sh'. The editor's status bar indicates the file has been modified ('Изменён'). The current line of code in the editor is 'tar -cvf ~/backup/task1.tar \$0', which is a command to create a tar archive of the script itself. The prompt '\$#' is visible at the end of the line.

Рис. 1: Написали скрипт

Запускаем файл

A terminal window with a dark background and light-colored text. It shows a series of commands being entered at a shell prompt. The prompt is '[vvustinova@vvustinova ~]\$'. The commands are: 'nano task1.sh', 'chmod 777 task1.sh', and './task1.sh'. The output of the last command is './task1.sh' followed by a red cursor. The prompt is then shown again with a white cursor.

```
[vvustinova@vvustinova ~]$ nano task1.sh
[vvustinova@vvustinova ~]$ chmod 777 task1.sh
[vvustinova@vvustinova ~]$ ./task1.sh
./task1.sh
[vvustinova@vvustinova ~]$
```

Рис. 2: Вводим команду ./task1.sh

При этом файл должен архивироваться одним из архиваторов на выбор

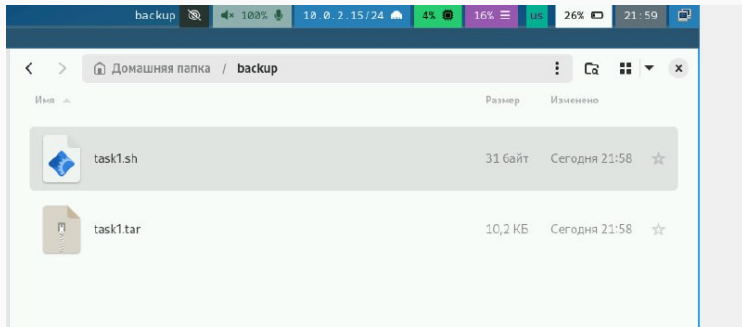
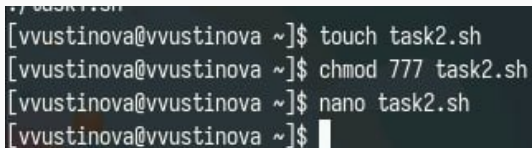


Рис. 3: Перешли с созданную папку backup и разархивировали файл

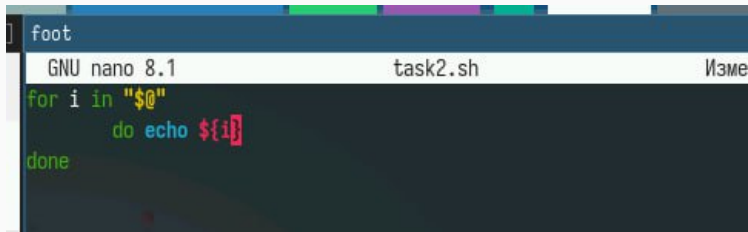
Создаем новый файл для второго задания и открываем редактор

A terminal window with a dark background and light gray text. It shows four lines of commands and their outputs. The first line is a partial command from the previous slide. The second line creates a file named task2.sh. The third line sets permissions of 777 for task2.sh. The fourth line opens task2.sh in the nano text editor. The prompt is [vvustinova@vvustinova ~]\$ and the cursor is at the end of the last line.

```
./task1.sh  
[vvustinova@vvustinova ~]$ touch task2.sh  
[vvustinova@vvustinova ~]$ chmod 777 task2.sh  
[vvustinova@vvustinova ~]$ nano task2.sh  
[vvustinova@vvustinova ~]$
```

Рис. 4: Редактор nano

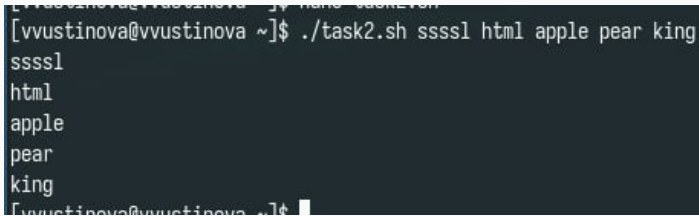
Написать пример командного файла, обрабатывающего любое произвольное число аргументов командной строки



```
foot
GNU nano 8.1 task2.sh
for i in "$@"
do echo ${i}
done
```

Рис. 5: Пишем код

Скрипт может последовательно распечатывать значения всех переданных аргументов

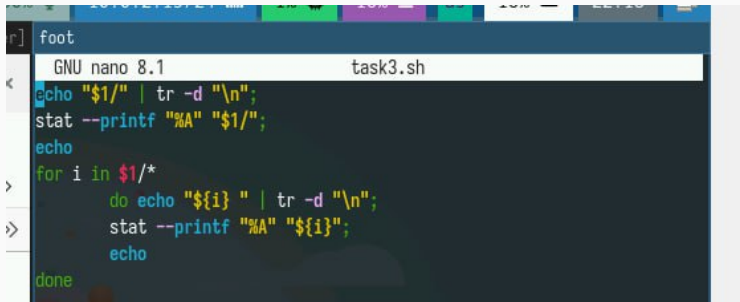
A terminal window with a dark background. The prompt is [vvustinova@vvustinova ~]\$. The command ./task2.sh ssssl html apple pear king is entered. The output is ssssl, html, apple, pear, king, each on a new line. The prompt [vvustinova@vvustinova ~]\$ is visible again at the bottom.

```
[vvustinova@vvustinova ~]$ ./task2.sh ssssl html apple pear king
ssssl
html
apple
pear
king
[vvustinova@vvustinova ~]$
```

Рис. 6: У нас выводит последовательно, как и требуется

Аналог команды ls

Написать командный файл — аналог команды ls (без использования самой этой команды и команды dir)



```
foot
GNU nano 8.1 task3.sh
echo "$1/" | tr -d "\n";
stat --printf "%A" "$1/";
echo
for i in $1/*
do echo "${i} " | tr -d "\n";
stat --printf "%A" "${i}";
echo
done
```

Рис. 7: Пишем код

Требуется, чтобы он выдавал информацию о нужном каталоге и выводил информацию о возможностях доступа к файлам этого каталога

```
[vvustinova@vvustinova ~]$ nano task3.sh
[vvustinova@vvustinova ~]$ nano task3.sh
[vvustinova@vvustinova ~]$ ./task3.sh ~
/home/vvustinova/drwx-----
/home/vvustinova/1 -rw-r--r--
/home/vvustinova/#2# -rw-r--r--
/home/vvustinova/#3# -rw-r--r--
/home/vvustinova/#4# -rw-r--r--
/home/vvustinova/abc1 -rw-rw-r--
/home/vvustinova/australia drwxr--r--
/home/vvustinova/backup drwxr-xr-x
/home/vvustinova/conf.txt -rw-r--r--
/home/vvustinova/dconf.txt drwxr-xr-x
/home/vvustinova/Desktop drwxr-xr-x
/home/vvustinova/Documents drwxr-xr-x
```

Рис. 8: Выводит необходимое

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.



```
root
GNU nano 8.1 task4.sh Изменён
let COUNT=0
for i in $2/*. $1
do let COUNT++
done
echo $COUNT
```

Рис. 9: Пишем код

Проверяем выполнение

```
[vvustinova@vvustinova ~]$ nano task4.sh
[vvustinova@vvustinova ~]$ ./task4.sh txt ~
4
[vvustinova@vvustinova ~]$
```

Рис. 10: Выводит необходимое, я проверила, ровно 4 файла txt находятся у меня в домашней директории

Мы успешно изучили основы программирования в оболочке ОС UNIX/Linux. Научились писать небольшие командные файлы.