

Лабораторная работа №12

Тойчубекова Асель Нурлановна

Тойчубекова Асель нурлановна

26 апреля 2024

Российский университет дружбы народов, Москва, Россия

Информация

- Тойчубекова Асель Нурлановна
- Студент НПИбд-02-23
- факультет физико- математических и естественных наук
- Российский университет дружбы народов
- 1032235033@rudn.ru
- <https://aseltoichubekova.github.io/ru/>

Целью данной лабораторной работы является изучение основы программирования в оболочке ОС LINUX. Также научиться писать небольшие командные файлы.

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.

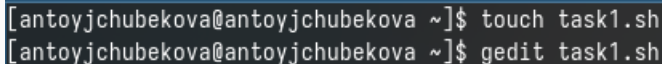
Командный процессор (командная оболочка, интерпретатор команд shell) — это программа, позволяющая пользователю взаимодействовать с операционной системой компьютера. В операционных системах типа UNIX/Linux наиболее часто используются следующие реализации командных оболочек: - оболочка Борна (Bourne shell или sh) — стандартная командная оболочка UNIX/Linux, содержащая базовый, но при этом полный набор функций; - C-оболочка (или csh) — надстройка на оболочкой 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.

Командные файлы, также известные как shell-скрипты, являются текстовыми файлами, содержащими последовательность команд командного процессора. Командные файлы позволяют пользователям автоматизировать задачи, выполняя несколько команд подряд без необходимости вводить их вручную.

Выполнение лабораторной работы

Для начала я создаю файл task1, в котором буду писать программу и открою его в редакторе gedit

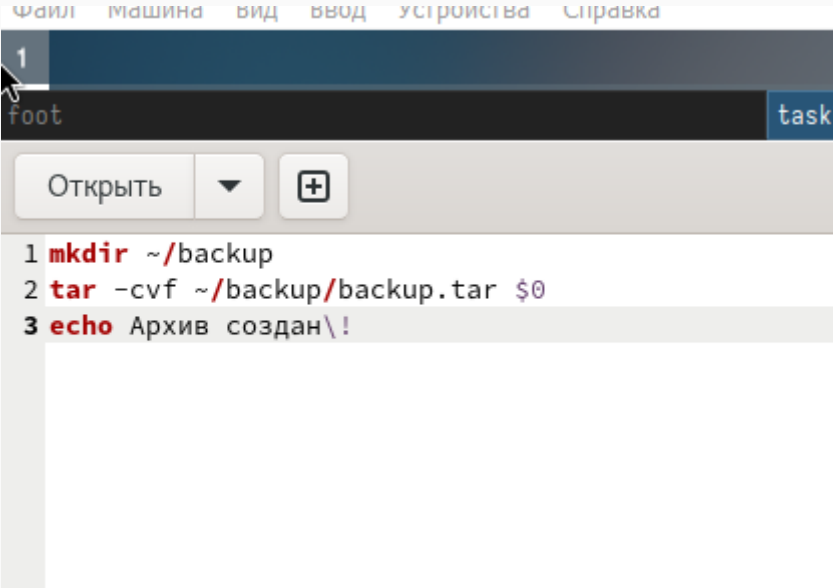
A terminal window with a dark background and light-colored text. It shows two commands being executed in sequence. The first command is 'touch task1.sh' and the second is 'gedit task1.sh'. Both commands are preceded by a prompt indicating the user is 'antoyjchubekova' at a machine named 'antoyjchubekova' in the home directory '~'.

```
[antoyjchubekova@antoyjchubekova ~]$ touch task1.sh  
[antoyjchubekova@antoyjchubekova ~]$ gedit task1.sh
```

Рис. 1: Создание и открытие файла

Редактирую файл, записывая код программы, которая будет делать резервную копию самого себя и при этом файл должен архивироваться. В этом коде я сперва создаю каталог `backup`, где будут храниться резервная копия и архив файла. С помощью команды `tar -cvf` создаю архив из резервной копии. С помощью команды `echo` вывожу сведения о созданном архиве.

Выполнение лабораторной работы



Файл машина вид ввод устройства справка

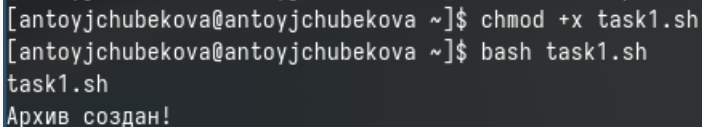
1

foot task

Открыть ▼ +

```
1 mkdir ~/backup
2 tar -cvf ~/backup/backup.tar $0
3 echo Архив создан\!
```

Меняю права доступа, включая права доступа на выполнение. Вызываю командный файл на выполнение и вижу сведения о том, что архив был создан, перейдя в домашний каталог в каталог backup видим, что архив удачно создан.

A terminal window with a dark background and light gray text. The prompt is [antoyjchubekova@antoyjchubekova ~]. The first command is chmod +x task1.sh. The second command is bash task1.sh. The output of the script is task1.sh followed by Архив создан! on the next line.

```
[antoyjchubekova@antoyjchubekova ~]$ chmod +x task1.sh
[antoyjchubekova@antoyjchubekova ~]$ bash task1.sh
task1.sh
Архив создан!
```

Рис. 3: Запуск программы

Выполнение лабораторной работы

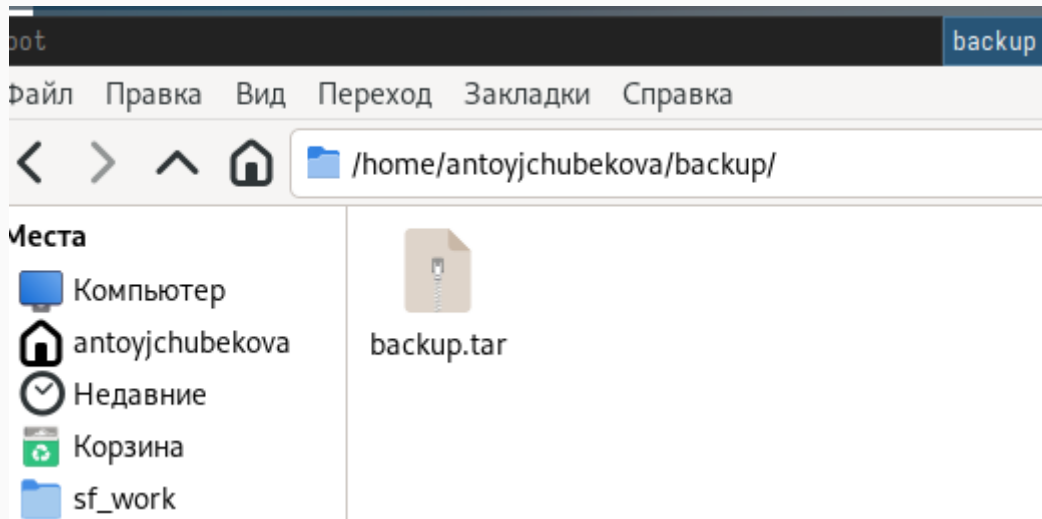
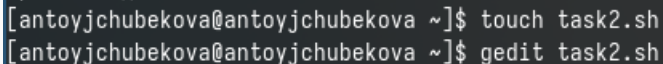


Рис. 4: Архив файла

Создаю файл task2, в котором буду писать программу и открою его в редакторе gedit

A terminal window with a dark background and light gray text. It shows two commands being executed in sequence. The first command is 'touch task2.sh' and the second is 'gedit task2.sh'. Both commands are preceded by the prompt '[antoyjchubekova@antoyjchubekova ~]\$'.

```
[antoyjchubekova@antoyjchubekova ~]$ touch task2.sh  
[antoyjchubekova@antoyjchubekova ~]$ gedit task2.sh
```

Рис. 5: Создание и открытие файла

Редактирую файл, записывая код программы, которая будет обрабатывать любое произвольное число аргументов командной строки. Для этого я создаю цикл который будет проходить по всей командной строке, по всем аргументам и буду выводить каждый из них с помощью команды echo, в конце добавляю done для указания, что конец цикла.

Выполнение лабораторной работы

Файл Машина Вид Ввод Устройства Справка

1

foot

*task2.sh (~) -

Открыть



```
1 for A in $*
2 do
3 echo $A
4 done
```

Выполнение лабораторной работы

Меняю права доступа, включая права на выполнение. Вызываю командный файл на выполнения в конце указывая аргументы, мы видим, что программа корректно работает и все аргументы выводятся на экран.

```
[antoyjchubekova@antoyjchubekova ~]$ chmod +x task2.sh
[antoyjchubekova@antoyjchubekova ~]$ bash task2.sh
[antoyjchubekova@antoyjchubekova ~]$ bash task2.sh 1 2 3 4
1
2
3
4
[antoyjchubekova@antoyjchubekova ~]$ bash task2.sh 11 101 456
11
101
456
```

Рис. 7: Запуск программы

Создаю файл task3, в котором буду писать программу и открою его в редакторе gedit

```
[antoyjchubekova@antoyjchubekova ~]$ touch task3.sh  
[antoyjchubekova@antoyjchubekova ~]$ gedit task3.sh
```

Рис. 8: Создание и открытие файла

Редактирую файл, который будет выводить данные о каталоге как команда `ls`, а также выводит информацию о правах доступа. Для этого я с помощью цикла `for` прохожусь по асем у содержанию данного каталога, указывая `*` для этого. Паралельно проверяю если это каталог, исполняемый файл или просто файл с помощью команды `test` и его опций `-d`(для директорий) `-e`(для исполняемого файла) `-f`(для файла) и используя команду `if/elif/else` как и в других высокоуровневых языках программирования. Паралельно с проверкой если условие истина вывожу соответствующее сообщение о том каталог это или файл и о правах доступа.

Выполнение лабораторной работы

Файл Папки Вид Ввод Структура Справка

1

foot

task3.sh (~) - gedit

Открыть



```
1 for A in *
2 do if test -d $A
3 then echo $A: каталог
4 elif test -e $A
5 then echo $A: исполняемый файл
6 else echo -n $A: файл
7 fi
8 if test -w $A
9 then echo Доступен для записи
10 else echo Не доступен для записи
11 fi
12 if test -r $A
13 then echo Доступен для чтения
14 else echo Не доступен для чтения
15 fi
16 done
```

Выполнение лабораторной работы

Меняю права доступа, включая права на выполнение. Вызываю командный файл на выполнение, мы видим, что программа корректно работает и на экран выводится все содержимое домашнего каталога с правами доступа.

```
[antoychubekova@antoychubekova ~]$ chmod +x task3.sh
[antoychubekova@antoychubekova ~]$ bash task3.sh
abc1: исполняемый файл
Доступен для записи
Доступен для чтения
a.sh: исполняемый файл
Доступен для записи
Доступен для чтения
australia: каталог
Доступен для записи
Доступен для чтения
backup: каталог
Доступен для записи
Доступен для чтения
b.sh: исполняемый файл
Доступен для записи
Доступен для чтения
cpr.cpr: исполняемый файл
Доступен для записи
Доступен для чтения
c.sh: исполняемый файл
Доступен для записи
Доступен для чтения
directory: каталог
Доступен для записи
Доступен для чтения
Downloads: каталог
Доступен для записи
```

Создаю файл task4, в котором буду писать программу и открою его в редакторе gedit

```
[antoyjchubekova@antoyjchubekova work]$ touch task4.sh  
[antoyjchubekova@antoyjchubekova work]$ gedit task4.sh  
[antoyjchubekova@antoyjchubekova work]$
```

Рис. 11: Создание и открытие файла

Редактирую файл, который будет получать в качестве аргумента командной строки формат файла и вычислить количество таких файлов в указанной директории. Путь к директории также передается в виде аргумента командной строки. Для этого я сперва с помощью команды `echo` и `read` вывожу на экран запрос пользователю, чтобы он ввел значения пути каталога и тип файла. Далее я проверяю существует ли такой каталог, если нет то вывожу об этом текст. Завожу переменную `x` и с помощью цикла ищу файла соответствующие требуемой, если нахожу прибавляю к `x + 1` который изначально был равен 0. Завершаю цикл с `done` и вывожу на экран `x`.

Выполнение лабораторной работы

```
foot task4.sh (~/work)
Открыть ▼ +
1 echo Введите путь к каталогу
2 read directory
3 echo Введите тип файла
4 read filetype
5 if [ ! -d "$directory" ]; then
6 echo Такого каталога нет
7 exit 1
8 fi
9 let x=0
10 for i in $directory/*. $filetype
11 do let x++
12 done
13 echo Количество файлов с расширением $filetype : $x
```

Выполнение лабораторной работы

Меняю права доступа, включая права на выполнение. Вызываю командный файл на выполнение, сперва проверяю каталог по адресу /home/antoyjchubekova/fun на количество файлов с sh и получаю ответ 3, перейдя по данному адресу мы видим, что программа сработала верно.

```
[antoyjchubekova@antoyjchubekova work]$ chmod +x task4.sh
[antoyjchubekova@antoyjchubekova work]$ bash task4.sh
Введите путь к каталогу
/home/antoyjchubekova/fun
Введите тип файла
sh
Количество файлов с расширением sh : 3
[antoyjchubekova@antoyjchubekova work]$
```



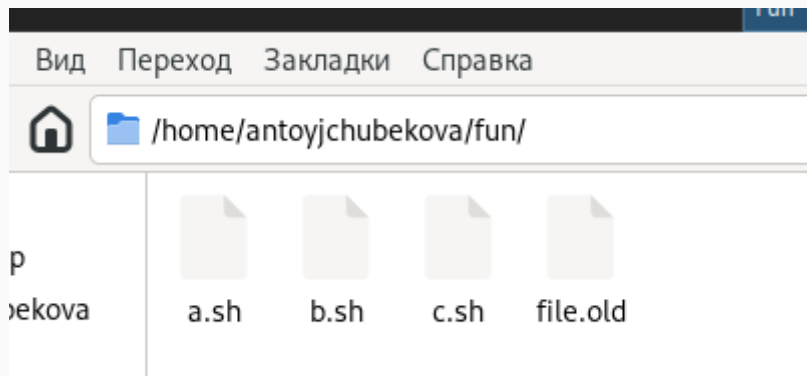
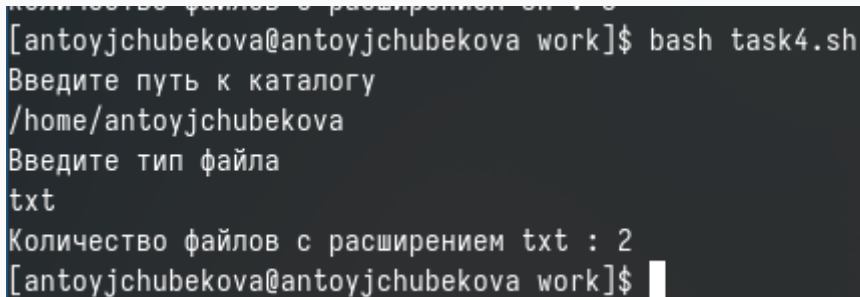


Рис. 14: каталог по адресу

Выполнение лабораторной работы

Далее проверяю каталог по адресу /home/antoyjchubekova/ на количество файлов с txt и получаю ответ 2, перейдя по данному адресу мы видим, что программа сработала верно.



```
[antoyjchubekova@antoyjchubekova work]$ bash task4.sh
Введите путь к каталогу
/home/antoyjchubekova
Введите тип файла
txt
Количество файлов с расширением txt : 2
[antoyjchubekova@antoyjchubekova work]$
```

Рис. 15: Запуск программы

Выполнение лабораторной работы

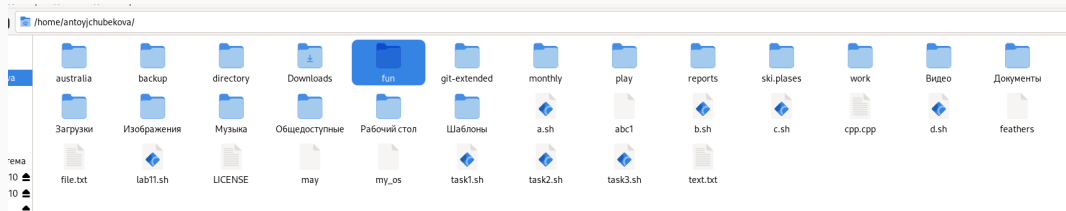


Рис. 16: каталог по адресу

В ходе выполнения лабораторной работы №12 я изучила основы программирования в оболочке ОС LINUX. Также я научилась писать небольшие командные файлы.