Российский университит дружбы народов

Факультет физико-математических и естественных наук

Лабораторная работа № 10 . Программирование в командном процессоре ОС UNIX. Командные файлы

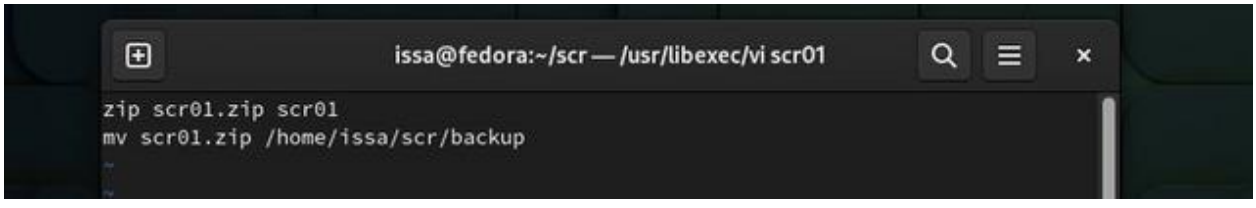
- Имя : исса гадир
- Студенческий билет : 1032218267
- Группа : нфибд-01-21

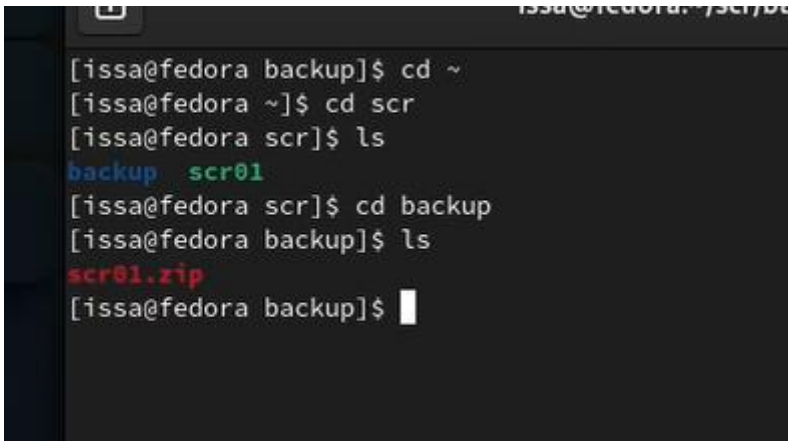
Цель работы

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

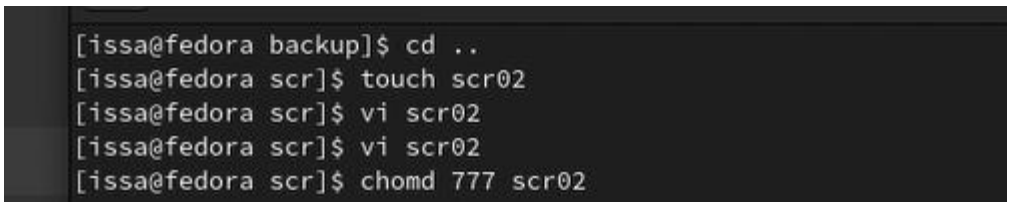
1. Написать скрипт, который при запуске будет делать резервную копию самого себя (то есть файла, в котором содержится его исходный код) в другую директорию backup в вашем домашнем каталоге. При этом файл должен архивироваться одним из архиваторов на выбор zip, bzip2 или tar. Способ использования команд архивации необходимо узнать, изучив справку.

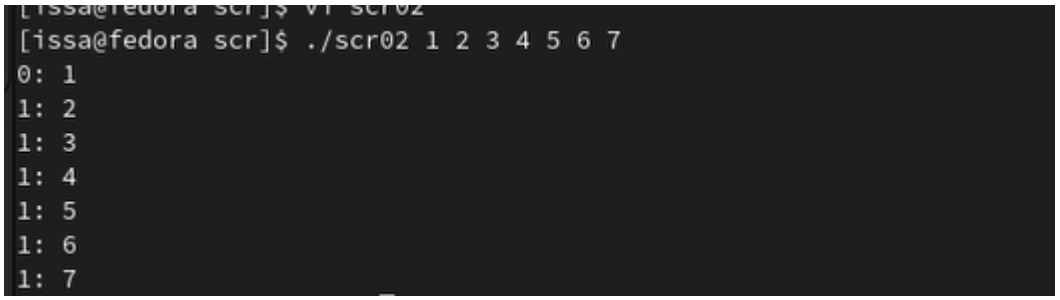
```
[issa@fedora ~]$ mkdir scr
[issa@fedora ~]$ cd scr
[issa@fedora scr]$ touch scr01
```





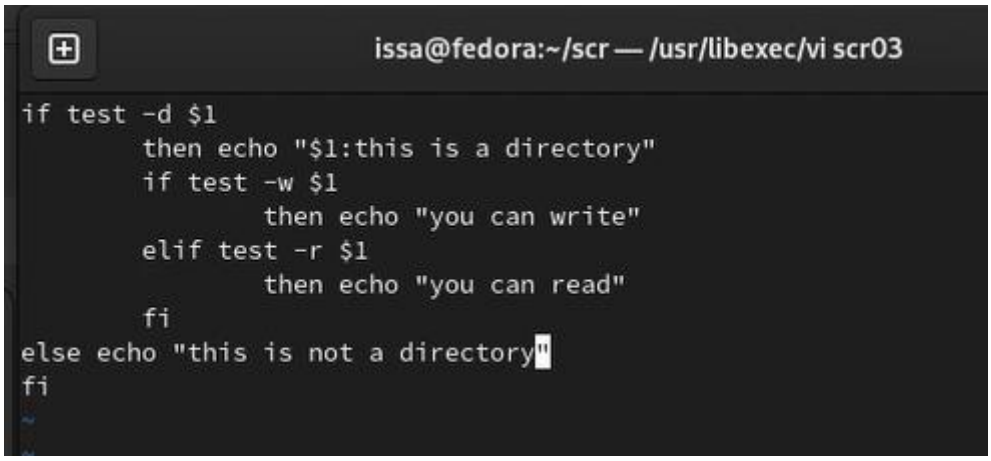
2. Написать пример командного файла, обрабатывающего любое произвольное число аргументов командной строки, в том числе превышающее десять. Например, скрипт может последовательно распечатывать значения всех переданных аргументов.

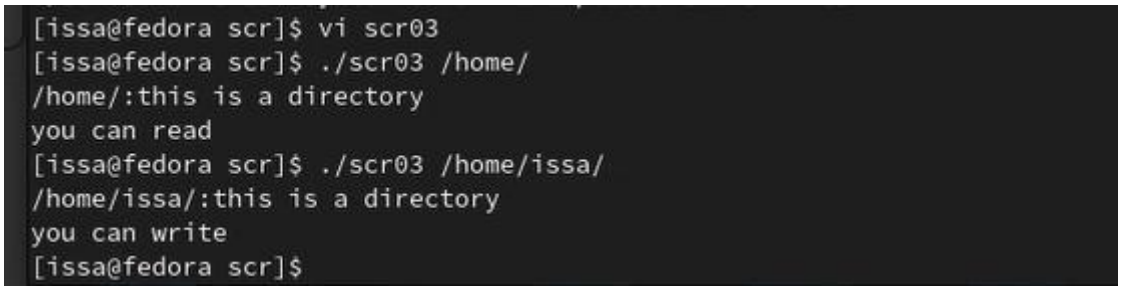




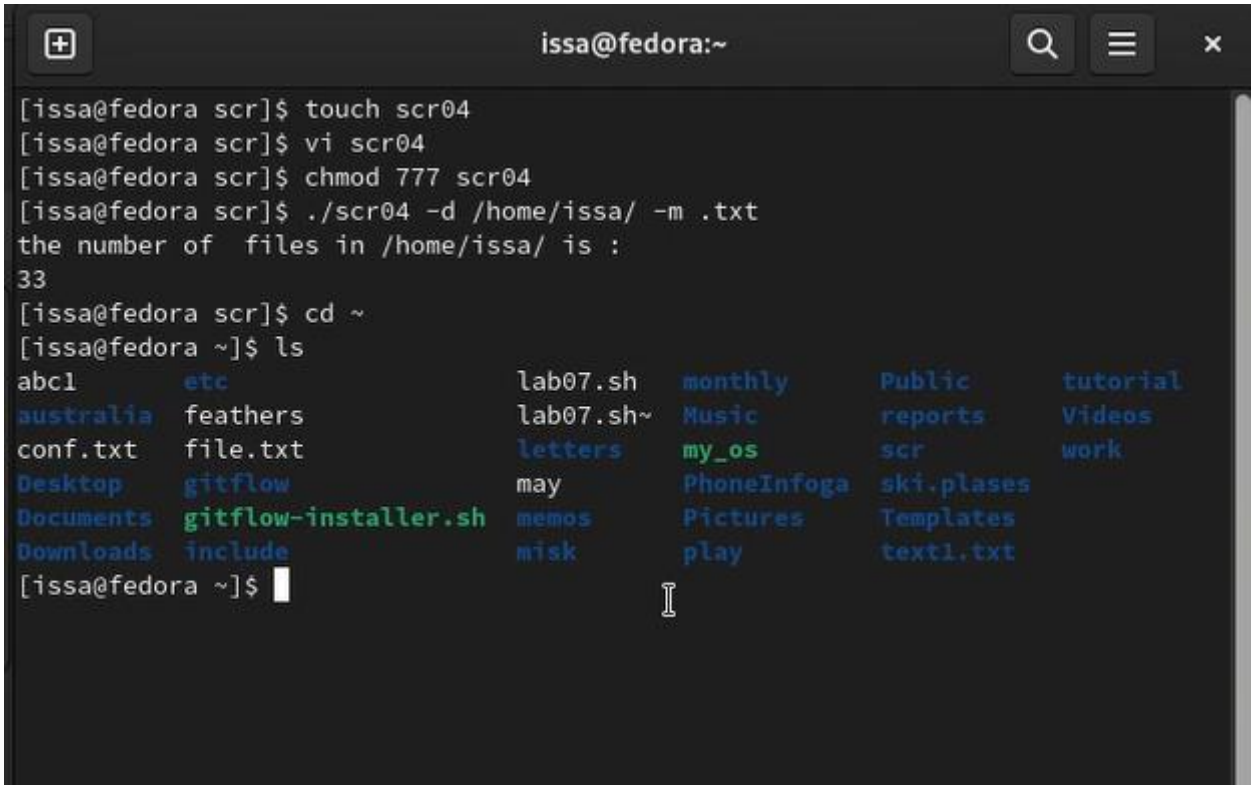
3. Написать командный файл — аналог команды ls (без использования самой этой команды и команды dir). Требуется,

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





4. Написаткомандный файл,который получаетв 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 командной строки.



Контрольные вопросы

1. **Объясните понятие командной оболочки. Приведите примеры командных оболочек.Чем они отличаются?**

- *Shell означает интерпретатор командной строки. Оболочка — это программа, которая обрабатывает команды и выводит результаты. Оболочка — это слой, который находится поверх ядра: 1) Он интерпретирует и обрабатывает команды, введенные пользователем. В отличие от пользователей оболочка имеет доступ к ядру. Пользователи могут получить доступ к ядру, только используя оболочку и вводя команды (то есть запуская программы). Системные вызовы используются программами*

для получения доступа к функциям ядра. Системный API состоит из всех системных вызовов.

2. `Что такое POSIX?`

- *POSIX расшифровывается как Portable Operating System Interface. Это семейство стандартов, определенных IEEE для обеспечения совместимости между операционными системами. Следовательно*

3. `Как определяются переменные и массивы в языке программирования bash?`

- *• Переменная в bash создается путем присвоения значения ее ссылке. Хотя для явного объявления переменной в bash не требуется использовать встроенный оператор declare, эта команда часто используется для более сложных задач управления переменными.*
- *• Массив представляет собой расположение элементов. Таким образом, каждый элемент называется массивом, если он организован таким образом.*

4. `Каково назначение операторов let и read?`

- *The LET statement can be used to assign a constant value to a variable name, a variable to a variable name, or the result of an expression to a variable name.*
- *The READ statement is used to pick up information for input from external sources. These sources could be input from the keyboard, computer file or a magnetic tape.*

5. `Какие арифметические операции можно применять в языке программирования bash?`

Общие варианты использования включают в себя:

- Сложение/вычитание/умножение/деление чисел.
- Округление чисел.
- Увеличение и уменьшение числа.
- Преобразование единиц.
- Вычисления с плавающей запятой.
- Нахождение процентов.
- Работа с различными системами счисления (двоичной, восьмеричной или шестнадцатеричной).

6. `Что означает операция (( ))?`

- *Условия оболочки bash могут быть записаны в двойных скобках*

7. `Какие стандартные имена переменных Вам известны?`
8. `Что такое метасимволы?`

- *Наиболее мощной функцией оболочки Linux Bash является ее способность работать с файлами и эффективно перенаправлять их ввод и вывод. Linux использует специальные символы или символы, известные как метасимволы, которые придают особое значение команде оболочки в отношении поиска файлов и соединения команд.*

9. `Как экранировать метасимволы?`

- *Most special characters can be escaped using the caret(^). Take a look at the following example.*

10. `Как создавать и запускать командные файлы?`

- *Чтобы создать его, используйте расширение .sh, но это не имеет большого значения, но помогает будущим пользователям быстро определить, какой это тип файла. Имя bat в основном используется в Windows, но в Linux расширения имен файлов не имеют большого значения. Это означает, что я могу назвать свой файл, скажем, run.de, и он все равно будет работать в файле bash, но я считаю хорошей практикой называть их с расширением .sh.*

11. `Как определяются функции в языке программирования bash?`

- function_name () { commands; }
- function function_name { commands; }

12. `Каким образом можно выяснить, является файл каталогом или обычным файлом?`

- *Наиболее удобочитаемый вариант проверки существования файла — использовать команду (test) в сочетании с (if , statment). Любой из приведенных ниже фрагментов кода проверит, существует ли файл /etc/resolv.conf:*
- ФАЙЛ=/etc/resolv.conf

13. `Каково назначение команд set, typeset и unset?`

- *Функция обработки*

14. `Как передаются параметры в командные файлы?`

    - *В пакетном сценарии вы можете получить значение любого аргумента, используя его %, за предмет следует конкретное положение в командной строке. Первый элемент, который всегда присутствует, — это %1, а второй элемент — всегда %2 и так далее.*
    - *Если вам нужны все аргументы, вы можете просто %* в пакетном сценарии. %* обозначают все аргументы (например, %1% 2% 3% 4% 5...), но только аргументы от %1 до %9 могут быть обозначены числом.*

15. `Назовите специальные переменные языка bash и их назначение.`

- $$ – display the PID (Process Identifier)
- $? – exit code
- $0 – the command running
- $1, $2, … $* – passing arguments

ВЫВОД :

Я знаком с linux, получил навыки работы с редактором emacs