Лабораторная работа №10

Простейший вариант выполнения лабораторной работы

Атанесов Александр Николаевич

Содержание

# 1 Цель работы

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

# 2 Задание

1.Взаимодействрвать с ОС через терминал посредством команд;

# 3 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы.

Например, в табл. [[1](#tbl:std-dir)] приведено краткое описание стандартных каталогов Unix.

Table 1: Описание некоторых каталогов файловой системы GNU Linux

| Имя каталога | Описание каталога |
| --- | --- |
| / | Корневая директория, содержащая всю файловую |
| /bin | Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям |
| /etc | Общесистемные конфигурационные файлы и файлы конфигурации установленных программ |
| /home | Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя |
| /media | Точки монтирования для сменных носителей |
| /root | Домашняя директория пользователя root |
| /tmp | Временные файлы |
| /usr | Вторичная иерархия для данных пользователя |

Более подробно об Unix см. в [1–6].

# 4 Выполнение лабораторной работы

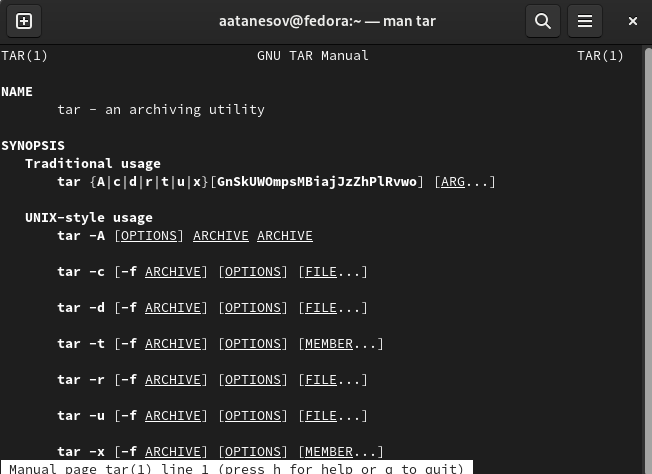
1. Ввожу команду man tar. (рис. [??])

Использую команду man

Использую команду man

## 4.1

1. Изучаю информацию о команде tar. (рис. [??])



Использую команду terminal

## 4.2

1. Создаю файл script.sh. (рис. [??])

Использую команду touch

Использую команду touch

## 4.3

1. Делаю файл script.sh исполняемым . (рис. [??])

Использую команду chmod

Использую команду chmod

## 4.4

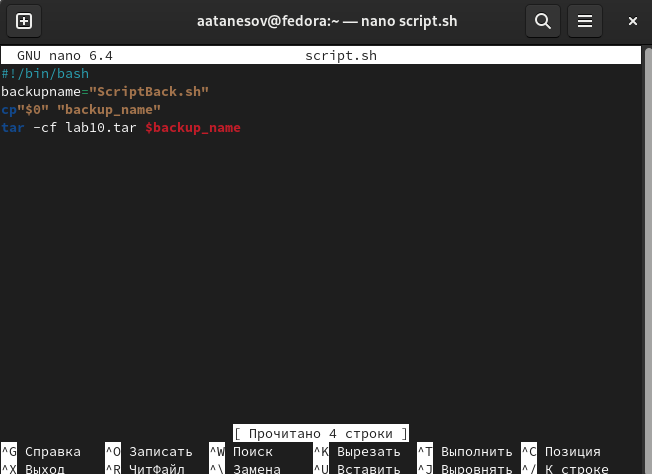
1. Открываю файл через nano. (рис. [??])

Использую команду nano

Использую команду nano

## 4.5

1. Ввожу необходимый код ,для выполнения условия. (рис. [??])



Использую редактор nano

## 4.6

1. Создаю файл script2.sh . (рис. [??])

Использую команду touch

Использую команду touch

## 4.7

1. Делаю файл исполняемым . (рис. [??])

Использую команду chmod

Использую команду chmod

## 4.8

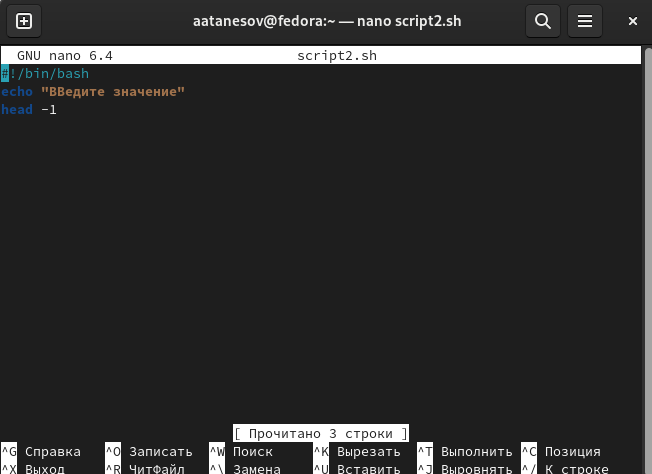
1. Открываю файл script2.sh . (рис. [??])

Использую команду nano

Использую команду nano

## 4.9

1. Ввожу необходимый код. (рис. [??])



Использую редактор nano

## 4.10

1. Выполняю файл script2.sh. (рис. [??])

Использую команду ./

Использую команду ./

## 4.11

1. Смотрю на вывод. (рис. [??])



Использую терминал

## 4.12

1. Создаю файл file.sh. (рис. [??])

Использую команду touch

Использую команду touch

## 4.13

1. Делаю файл исполняемым. (рис. [??])

Использую команду chmod(image/13.png){#fig:014 width=90%}

## 4.14

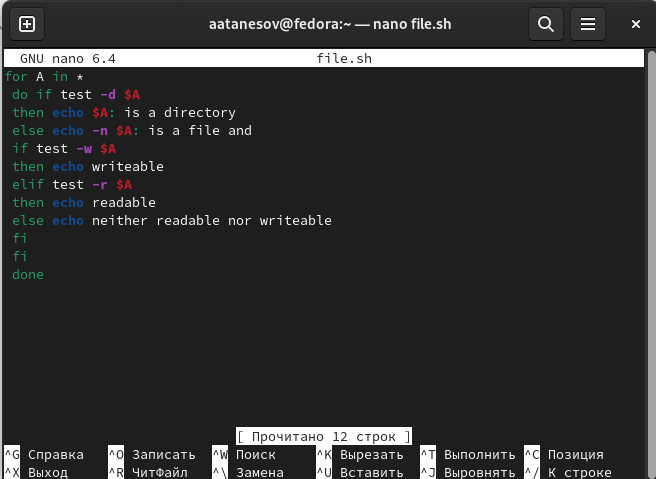
1. Открываю файл через nano. (рис. [??])

Открываю файл через nano

Открываю файл через nano

## 4.15

1. Ввожу необходимый код. (рис. [??])

(image/18.png){#fig:018 width=90%}

## 4.16

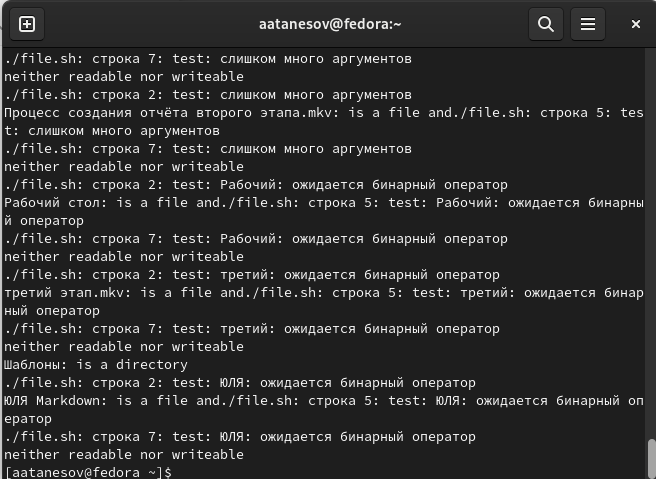
1. Воспроизвожу файл file.sh. (рис. [??])

Использую команду ./

Использую команду ./

## 4.17

1. Смотрю на результат. (рис. [??])



Использую терминал

## 4.18

1. Создаю файл file2.sh. (рис. [??])

Использую команду touch

Использую команду touch

## 4.19

1. Делаю файл исполняемым. (рис. [??])

Использую команду chmod

Использую команду chmod

## 4.20

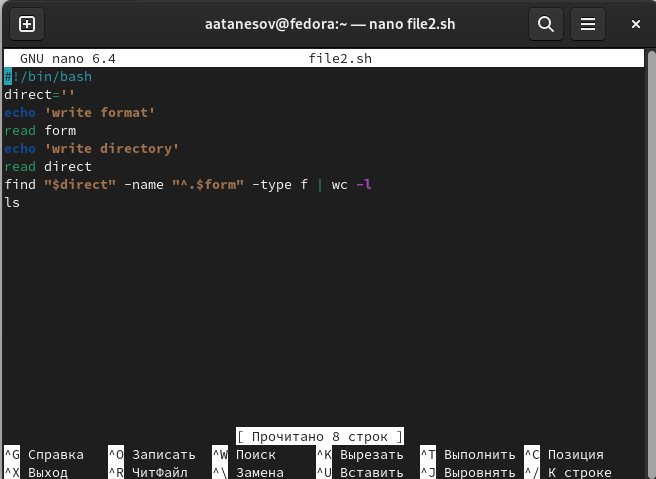
1. Запускаю файл через nano. (рис. [??])

Использую команду nano

Использую команду nano

## 4.21

1. Ввожу необходимый код. (рис. [??])



Использую редактор nano

## 4.22

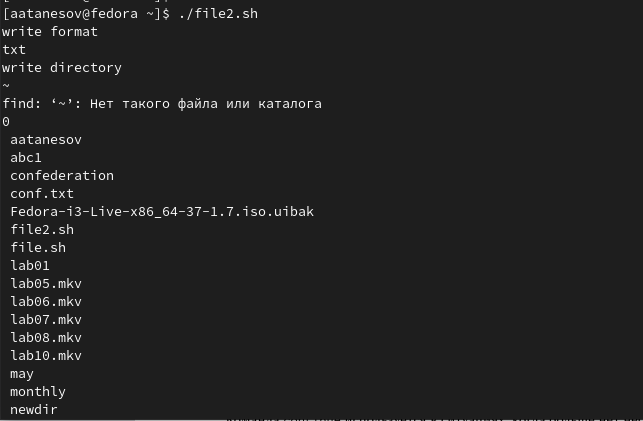
1. Воспроизвожу файл file2.sh . (рис. [??])

Использую команду ./

Использую команду ./

## 4.23

1. Смотрю на результат . (рис. [??])



Использую терминал

# 5 Выводы

* Познал основы программирования в UNIX. # Ответы на контрольные вопросы
  1. Командная оболочка - это программа, которая обеспечивает пользовательский интерфейс для взаимодействия с операционной системой. Она позволяет пользователю вводить команды и запускать программы, а также выполняет подстановку значений переменных и метасимволов. Примеры командных оболочек: bash, zsh, csh, tcsh. Они отличаются синтаксисом и набором доступных функций.
  2. POSIX (Portable Operating System Interface) - это стандарт, описывающий интерфейс между операционной системой и приложением. Он был разработан для обеспечения переносимости приложений между разными операционными системами и включает в себя спецификации для системных вызовов, библиотек, командных интерпретаторов и других компонентов операционной системы.
  3. Переменные в языке программирования bash определяются путем присваивания значения имени переменной. Например: var=“hello”. Массивы определяются с использованием квадратных скобок: array=(1 2 3).
  4. Оператор let используется для выполнения арифметических операций. Пример: let “var = 2 + 2”. Оператор read используется для чтения данных из пользовательского ввода.
  5. В языке программирования bash можно выполнять арифметические операции сложения, вычитания, умножения, деления, взятия остатка, а также использовать скобки для определения порядка операций.
  6. Операция (( )) используется для выполнения арифметических операций, а также для сравнения чисел. Например: (( var1 > var2 )). Она возвращает результат в виде 0 (если условие ложно) или 1 (если условие истинно).
  7. Стандартными именами переменных являются: HOME, USER, PATH, SHELL, TERM и другие. Они определяют различные настройки и параметры системы.
  8. Метасимволы - это символы, которые используются для обозначения шаблонов и задания нескольких символов в одной команде. Например: \*, ?, [], {}.
  9. Метасимволы можно экранировать, добавив перед ними символ обратного слеша (). Например: \*.
  10. Командные файлы создаются с помощью текстового редактора и сохраняются с расширением .sh. Для запуска командного файла необходимо установить ему права на выполнение с помощью команды chmod +x filename.sh, а затем выполнить его с помощью команды ./filename.sh.
  11. Функции в языке программирования bash определяются с помощью ключевого слова function и блока кода, который должен быть выполнен, когда функция вызывается. Например: function hello { echo “Hello, world!”; }
  12. Для определения типа файла используется команда file. Например: file myfile.txt. В ответ будет указан тип файла (текстовый, исполняемый и т.д.).
  13. Команда set используется для установки настроек оболочки, typeset - для определения типа переменных и их свойств, а unset - для удаления переменных.
  14. В командные файлы параметры передаются через аргументы командной строки. Например: ./script.sh arg1 arg2.
  15. Специальные переменные языка bash: $0 - имя скрипта, $1, $2 и т.д. - аргументы командной строки, $# - количество переданных параметров, $\* и $@ - все аргументы командной строки, $$ - идентификатор текущего процесса, $? - статус завершения последней выполненной команды.

# Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016. URL: <https://www.gnu.org/software/bash/manual/>.

2. Newham C. [Learning the bash Shell: Unix Shell Programming](http://www.amazon.com/Learning-bash-Shell-Programming-Nutshell/dp/0596009658). O’Reilly Media, 2005. 354 с.

3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.

4. Robbins A. [Bash Pocket Reference](https://www.ncbi.nlm.nih.gov/pubmed/25246403). O’Reilly Media, 2016. 156 с.

5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.

6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.