Отчёт по лабораторной работе №12

*дисциплина: Операционные системы*

Максим Александрович Мишонков

Содержание

# 1 Цель работы

Целью данной лабораторной работы является изучение основ программирования в оболочке OC UNIХ, приобретение навыков написания более сложных командных файлов с использованием логических управляющих конструкций и циклов.

# 2 Задание

Изучить основы программирования в оболочке OC UNIХ, научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

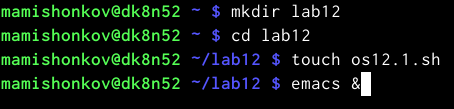
# 3 Теоретическое введение

Преимущества и недостатки Bash:

Многие языки программирования намного удобнее и понятнее для пользователя. Например, Python более быстр, так как компилируется байтами. Однако главное преимущество Bash - его повсеместное распространение. Более того, Bash позволяет очень легко работать с файловой системой без лишних конструкций. Но относительно такие bash очень сжат. То есть, например, C имеет гораздо более широкие возможности для разработчика.

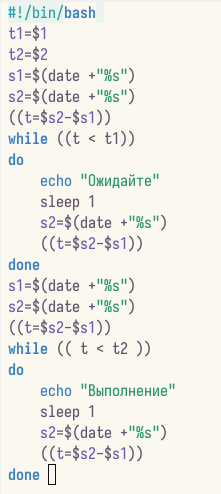
# 4 Выполнение лабораторной работы

1. Создал рабочее пространство для скриптов данной лабораторной работы, создал файл для скрипта 1. (рис. [??])



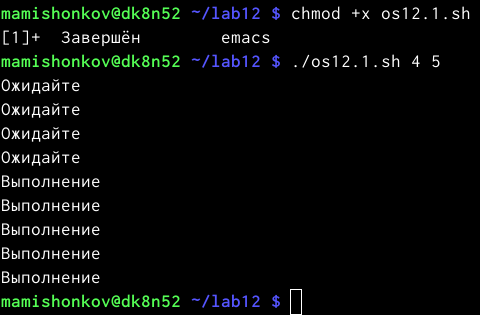
Создание каталога и файла

1. Написал скрипт 1. (рис. [??])



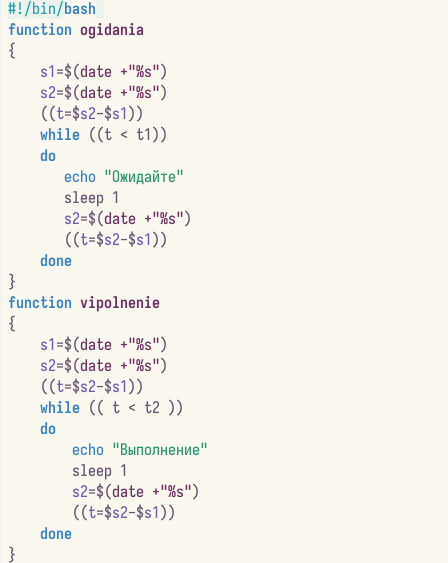
Скрипт 1

1. Проверил работу скрипта 1. (рис. [??])



Проверка работы скрипта 1

1. Изменил скрипт 1 так, чтобы его можно было выполнять в нескольких терминалах. (рис. [??],[??])

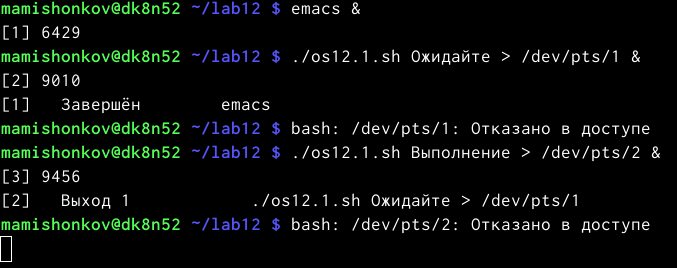


Изменённый скрипт 1



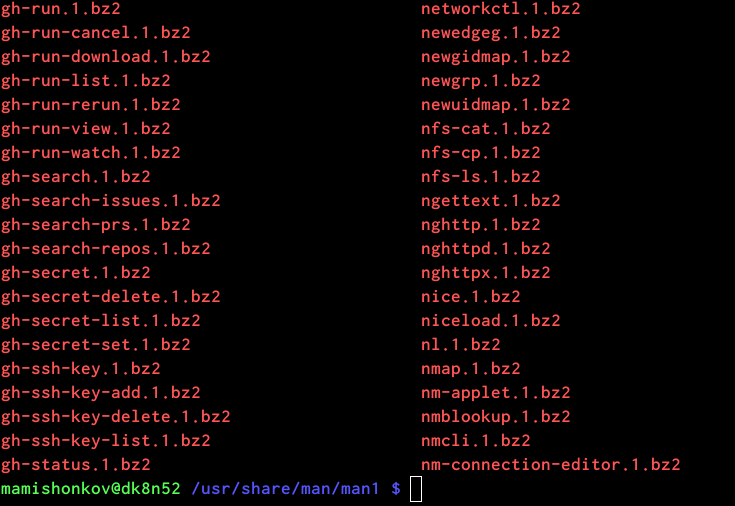
Изменённый скрипт 1

1. Проверил работу изменённого скрипта 1. (рис. [??])



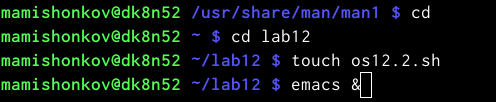
Проверка работы изменённого скрипта 1

1. Изучил содержимое файла “usr/share/man/man1”. (рис. [??])



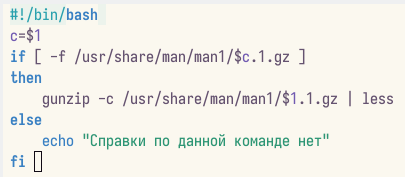
Изучение содержимого файла

1. Создал файл для скрипта 2. (рис. [??])



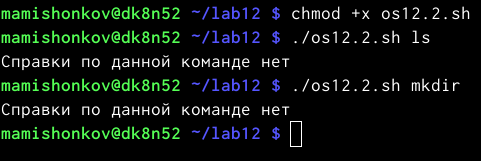
Создание файла

1. Написал скрипт 2. (рис. [??])



Скрипт 2

1. Проверил работу скрипта 2. (рис. [??])



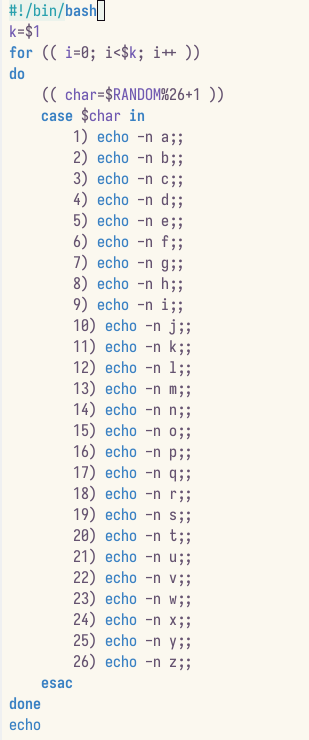
Проверка работы скрипта 2

1. Создал файл для скрипта 3. (рис. [??])

Создание файла

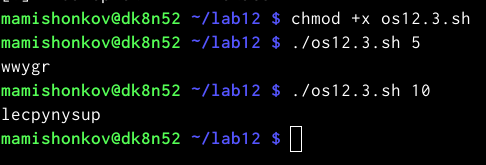
Создание файла

1. Написал скрипт 3. (рис. [??])



Скрипт 3

1. Проверил работу скрипта 3. (рис. [??])



Проверка работы скрипта 3

**Контрольные вопросы:**

1). while [$1 != “exit”]

В данной строчке допущены следующие ошибки:

не хватает пробелов после первой скобки [и перед второй скобкой ]  
  
выражение $1 необходимо взять в “”, потому что эта переменная может содержать пробелы.

Таким образом, правильный вариант должен выглядеть так: while [“$1”!= “exit”]

2). Чтобы объединить несколько строк в одну, можно воспользоваться несколькими способами:

Первый:

VAR1=“Hello,

“VAR2=” World”

VAR3=“VAR2”

echo “$VAR3”

Результат: Hello, World

Второй:

VAR1=“Hello,”

VAR1+=” World”

echo “$VAR1”

Результат: Hello, World

3). Команда seq в Linux используется для генерации чисел от ПЕРВОГО до ПОСЛЕДНЕГО шага INCREMENT.

Параметры:

seq LAST: если задан только один аргумент, он создает числа от 1 до LAST с шагом шага, равным 1. Если LAST меньше 1, значение is не выдает.  
  
seq FIRST LAST: когда заданы два аргумента, он генерирует числа от FIRST до LAST с шагом 1, равным 1. Если LAST меньше FIRST, он не выдает никаких выходных данных.  
  
seq FIRST INCREMENT LAST: когда заданы три аргумента, он генерирует числа от FIRST до LAST на шаге INCREMENT . Если LAST меньше, чем FIRST, он не производит вывод.  
  
seq -f «FORMAT» FIRST INCREMENT LAST: эта команда используется для генерации последовательности в форматированном виде. FIRST и INCREMENT являются необязательными.  
  
seq -s «STRING» ПЕРВЫЙ ВКЛЮЧЕНО: Эта команда используется для STRING для разделения чисел. По умолчанию это значение равно /n. FIRST и INCREMENT являются необязательными.  
  
seq -w FIRST INCREMENT LAST:эта команда используется для выравнивания ширины путем заполнения начальными нулями. FIRST и INCREMENT являются необязательными.

4). Результатом данного выражения $((10/3))будет 3, потому что это целочисленное деление без остатка.

5). Отличия командной оболочки zshот bash:

В zsh более быстрое автодополнение для cdс помощью Тab  
  
В zsh существует калькулятор zcalc, способный выполнять вычисления внутри терминала  
  
В zsh поддерживаются числа с плавающей запятой  
  
В zsh поддерживаются структуры данных «хэш»  
  
В zsh поддерживается раскрытие полного пути на основе неполных данных  
  
В zsh поддерживаетсязаменачастипути  
  
В zsh есть возможность отображать разделенный экран, такой же как разделенный экран vim

6). for((a=1; a<= LIMIT; a++)) синтаксис данной конструкции верен, потому что, используя двойные круглые скобки, можно не писать $ перед переменными ().

7). Преимущества скриптового языка bash:

Один из самых распространенных и ставится по умолчаниюв большинстве дистрибутивах Linux, MacOS  
  
Удобное перенаправление ввода/вывода  
  
Большое количество команд для работы с файловыми системами Linux  
  
Можно писать собственные скрипты, упрощающие работу в Linux

Недостатки скриптового языка bash:

Дополнительные библиотеки других языков позволяют выполнить больше действий  
  
Bash не является языков общего назначения  
  
Утилиты, при выполнении скрипта, запускают свои процессы, которые, в свою очередь, отражаются на быстроте выполнения этого скрипта  
  
Скрипты, написанные на bash, нельзя запустить на других операционных системах без дополнительных действий.

# 5 Выводы

В ходе выполнения данной лабораторной работы я изучил основы программирования в оболочке OC UNIХ, научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.