

Лабораторная работа 12

Программирование в командном процессоре ОС UNIX. Командные файлы

Чигладзе М.В.

29 мая 2003

Российский университет дружбы народов, Москва, Россия

Информация

- Чигладзе Майя Владиславовна
- студентка РУДН направления Прикладная информатика
- заместитель ОСК профсоюза РУДН
- волонтер университета и Москвы
- [1132239399@pfur.ru]
- <https://github.com/LaMeru>



Вводная часть

- Программирование в командной оболочке ОС UNIX является актуальной темой, поскольку командная оболочка является основным инструментом взаимодействия пользователя с операционной системой.

- Объект исследования: Командные файлы и язык программирования Bash
- Предмет исследования: Возможности и особенности программирования в командной оболочке UNIX.

Цель работы: Изучение возможностей и особенностей программирования в командной оболочке UNIX, а также получение практических навыков работы с командными файлами

Задачи работы:

- Изучение основ языка программирования Bash;
- Разработка простых и сложных командных файлов;
- Использование переменных, циклов и условных операторов в командных файлах;
- Работа с ошибками и отладка командных файлов;
- Применение функций и переменных для упрощения и структурирования кода.

Анализ литературы, изучение документации, написание и тестирование командных файлов

Выполнение лабораторной работы

Задание 1.

Задача: 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.

```
/home/mvchigladze/Рабочий стол/study_2022-2023_os-intro/Labs/lab11/backup/first.sh
#!/bin/bash

##### переменные #####

### Установите правильное местоположение источника и места назначения
SOURCE_DIR="/home/mvchigladze/Рабочий стол/study_2022-2023_os-intro/Labs/lab11"
DEST_DIR="/home/mvchigladze/Рабочий стол/study_2022-2023_os-intro/Labs/lab11/backup"
TMP_FILE=/tmp/copyfileslist.txt

### Установите имя пользователя и имя группы, чтобы установить права на скопированные файлы

### Установите для CHANGE_OWNERSHIP значение 1, чтобы изменить владельца, или 0, чтобы не изменять его.
CHANGE_OWNERSHIP=1
USER='root'
GROUP='root'

##### Не редактируйте, пока не потребуется #####

### Проверка, существует ли исходный каталог

### Скрипт остановится, если источника не существует

if [ -d "${SOURCE_DIR}" ]; then
echo "Source directory found"
else
echo "Source directory not found. Please check above variables are set correctly"
echo "script exited"
exit 1
fi

### Проверка, существует ли каталог назначения
### Скрипт создаст каталог назначения, если он не существует.
### Если не удалось создать каталог, скрипт будет прерван

if [ -d "${DEST_DIR}" ]; then
echo "Destination directory found, all ok"
else
echo "Destination directory not found, creating now"
mkdir -p "${DEST_DIR}"
if [ $? -eq 0 ]; then
echo "Successfully created destination directory."
else
echo "Failed to create destination directory. Script exited"
exit 1
fi
fi
```

```
root@mvchigladze:/home/mvchigladze/Рабочий стол/study_2022-2023_os-intro/Labs/lab11# chmod +x first.sh
root@mvchigladze:/home/mvchigladze/Рабочий стол/study_2022-2023_os-intro/Labs/lab11# ./first.sh
Source directory found
Destination directory found, all ok
File ./first.sh successfully copied.
```

Рис. 2: Результат 1

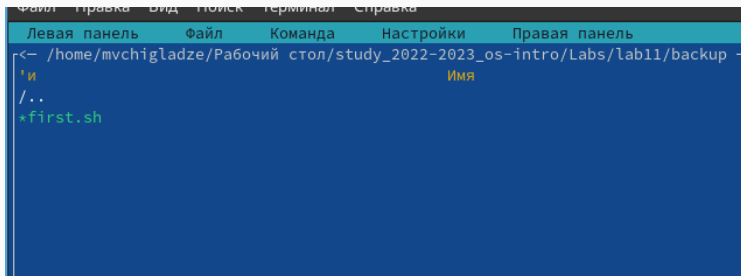


Рис. 3: Результат 2

Задание 2.

Задача: 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

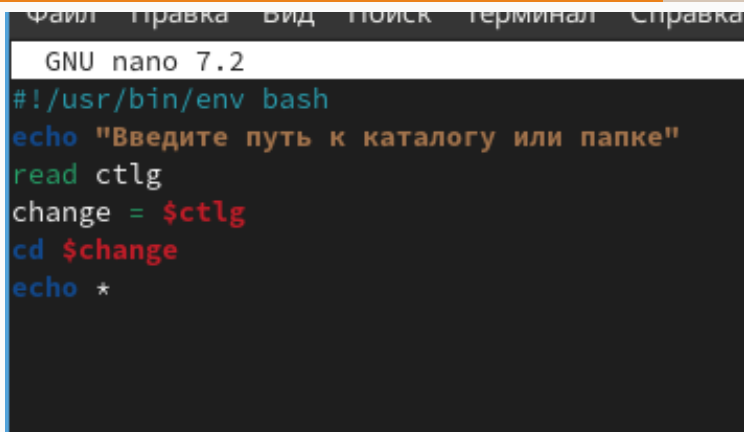
```
/home/mvchigladze/Рабочий с
#!/usr/bin/env bash
i=1
for a in "$@"; do
    echo "$i: $a"
    ((i++))
done
```

```
root@mvchigladze:/home/mvchigladze/Рабочий стол/study_2022-2023_os-intro/Labs/lab11# chmod +x second.sh
root@mvchigladze:/home/mvchigladze/Рабочий стол/study_2022-2023_os-intro/Labs/lab11# ./second.sh 64 8 5 5 5 5
1: 64
2: 8
3: 5
4: 5
5: 5
6: 5
7: 5
```

Рис. 5: Результат

Задание 3.

Задача: 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.



```
Файл  Правка  Вид  Поиск  Терминал  Справка
GNU nano 7.2
#!/usr/bin/env bash
echo "Введите путь к каталогу или папке"
read ctlg
change = $ctlg
cd $change
echo *
```

Рис. 6: Код

```
root@mvchigladze:/home/mvchigladze/Рабочий стол/study_2022-2023_os-intro/Labs/lab11# chmod +x third.sh
root@mvchigladze:/home/mvchigladze/Рабочий стол/study_2022-2023_os-intro/Labs/lab11# ./third.sh
Введите путь к каталогу или папке
/tmp
./third.sh: строка 4: change: команда не найдена
anaconda-ks.cfg mypublic.key public.pgp texput.log
```

Рис. 7: Результат

Задание 4.

Задача: 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 командной строки

```
GNU nano 7.2
#!/bin/bash

if [ -z "$1" ] || [ -z "$2" ]; then
    echo "Usage: $0 directory file_extension"
    exit 1
fi

count=$(ls -A "$1" | grep "^\. $2$" | wc -l)
echo "$1 contains $count files with extension $2."
```

Рис. 8: Код

```
root@mvchigladze:/home/mvchigladze/Рабочий стол/study_2022-2023_os-intro/Labs/lab11# chmod +x fourth.sh
root@mvchigladze:/home/mvchigladze/Рабочий стол/study_2022-2023_os-intro/Labs/lab11# ./fourth.sh /tmp txt
/tmp contains 0 files with extension txt.
root@mvchigladze:/home/mvchigladze/Рабочий стол/study_2022-2023_os-intro/Labs/lab11# nano fourth.sh
```

Рис. 9: Результат

Результаты

В результате данной работы я изучила возможности и особенностей программирования в командной оболочке UNIX, а также получила практические навыки работы с командными файлами