

# Отчет по лабораторной работе № 2 по курсу “Фундаментальная информатика”

Студент группы М80-109Б-22 Малько Павел Алексеевич, № по списку 10

Контакты e-mail: [malko.pavel07@gmail.com](mailto:malko.pavel07@gmail.com)

Работа выполнена: «25» сентября 2022г.

Преподаватель: каф. 806 Сысоев Максим Алексеевич

Отчет сдан «    » \_\_\_\_\_ 20\_\_ г., итоговая оценка \_\_\_\_\_

Подпись преподавателя \_\_\_\_\_

## 1. Тема: Операционная среда ОС UNIX

2. **Цель работы:** Изучение и освоение программного обеспечения ОС UNIX и приобретение навыков, необходимых для выполнения курсовых и лабораторных работ в среде UNIX.

3. **Задание:** Приобретение основных навыков работы в ОС UNIX

## 4. Оборудование (студента):

Процессор *amd ryzen 5500u* с ОП 7851 Мб, НМД 1024 Гб. Монитор 1920x1080

## 5. Программное обеспечение (студента):

Операционная система семейства: *linux*, наименование: *kali*, версия 2022.3

интерпретатор команд: *bash* версия 0.16.1.

Система программирования -- версия --, редактор текстов *emacs* версия 25.2.2

Утилиты операционной системы --

Прикладные системы и программы --

Местонахождение и имена файлов программ и данных на домашнем компьютере --

## 6. Идея, метод, алгоритм

При просмотре фотографий в файловом хранилище фотоаппарата или смартфона без возможности предварительного просмотра может оказаться полезным автоматически упорядочить фотографии в каталоге по определённому признаку(например, по времени). Автоматически они могут иметь неудобные для ориентировки наименования.

Консоль предоставляет возможности получения информации о дате создания файла и его переименовки. Также стоит предусмотреть случай, когда фотографии созданы одновременно. В скрипте должна быть возможность лёгкого изменения расширения переименованных файлов.

## 7. Сценарий выполнения работы

1. Научиться ориентироваться в файловой системе с консольным интерфейсом

2. Освоить механизмы создания файлов и папок

3. Вникнуть в механизм прав доступа к файлу пользователя, группы и прочих лиц(other) - изменить права доступа к файлу

4. Для создания скрипта на языке Bash освоить работу в текстовом редакторе nano, emacs или vim

5. Ознакомиться с необходимыми источниками в интернете и книгами по Linux для написания скрипта

6. Написать и отладить скрипт, используя полученные знания

Далее - тесты написанного скрипта:

Входные данные	Выходные данные	Описание тестируемого случая
Каталог с изображениями	Каталог с переименованным и в соответствии с датой создания изображениями	Успешная переименовка изображений в общем случае
Каталог с изображениями, где некоторые фотографии созданы одновременно	Каталог с переименованным и изображениями и присвоенными цифрами для совпадающих по времени создания	Переименовка для случая одновременно созданных файлов

**8. Распечатка протокола** (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем).

```
(kali㉿kali)-[~]
└─$ ls
Desktop Documents Downloads hello hello.py Music Pictures Public Templates test Videos

(kali㉿kali)-[~]
└─$ mkdir pops

(kali㉿kali)-[~]
└─$ cd pops

(kali㉿kali)-[~/pops]
└─$ touch program.py

(kali㉿kali)-[~/pops]
└─$ touch writings{1..40}.txt

(kali㉿kali)-[~/pops]
└─$ ls
program.py  writings15.txt writings20.txt writings26.txt writings31.txt writings37.txt writings5.txt
writings10.txt writings16.txt writings21.txt writings27.txt writings32.txt writings38.txt writings6.txt
writings11.txt writings17.txt writings22.txt writings28.txt writings33.txt writings39.txt writings7.txt
writings12.txt writings18.txt writings23.txt writings29.txt writings34.txt writings3.txt writings8.txt
writings13.txt writings19.txt writings24.txt writings2.txt writings35.txt writings40.txt writings9.txt
writings14.txt writings1.txt writings25.txt writings30.txt writings36.txt writings4.txt

(kali㉿kali)-[~/pops]
└─$ pwd
/home/kali/pops

(kali㉿kali)-[~/pops]
└─$ cd ..

(kali㉿kali)-[~]
└─$ mkdir -p uncle/cousin

(kali㉿kali)-[~]
└─$ cd uncle

(kali㉿kali)-[~/uncle]
└─$ cd cousin
```

```
(kali㉿kali)-[~/uncle/cousin]
└─$ cd ..
```

```
(kali㉿kali)-[~/uncle]
└─$ cd ..
```

```
(kali㉿kali)-[~]
└─$ ls -l
total 52
```

```
...
-rw-r--r-- 1 kali kali  11 Sep 12 04:21 hello.py
...
```

```
(kali㉿kali)-[~]
└─$ ls -al
total 180
```

```
...
-rw-r--r-- 1 kali kali  11 Sep 12 04:21 hello.py
...
```

```
(kali㉿kali)-[~]
└─$ chmod u+x hello.py
```

```
(kali㉿kali)-[~]
└─$ ls -l
total 52
```

```
...
-rwxr--r-- 1 kali kali  11 Sep 12 04:21 hello.py
...
```

**9. Дневник отладки** должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание
0	дом	19.09.2020	20:20	Скрипт удаляет все файлы в каталоге при попытке использования	Исправил ошибки в использовании синтаксиса	Ознакомился лучше с переменными в Bash

**10. Замечания автора** замечаний нет

#### **11. Выводы**

В ходе данной работы я ознакомился с базовыми элементами Bash как языка программирования - переменными, циклами, условными ветвлениями. Были найдены и применены встроенные команды языка для получения информации о файле. Полученный в ходе работы скрипт я смогу использовать на изображениях из галереи своего смартфона - теперь я смогу удобно ориентироваться по множеству фото на основе даты создания.

Примечательна творческая составляющая работы - написать свой скрипт на Bash. С одной стороны, было очень интересно этим заняться вместо рутинных занятий. С другой стороны, творчески придумать идею может оказаться сложным для некоторых людей (признаюсь, к таким отношусь и я сам). Или студент может неправильно оценить сложность своей идеи, выбрав либо непосильную для себя задачу, либо слишком лёгкую, которая не понравится проверяющему на защите.

Недочёты при выполнении задания могут быть устранены следующим образом: --

Подпись студента \_\_\_\_\_