

# Отчет по лабораторной работе № 21 по курсу “Практикум Программирования”

Студент группы М80-109Б-22 Ефименко Кирилл Игоревич, № по списку 7

Контакты  
recrut5678@gmail.com  
@vivichv9

Работа выполнена: «21» февраля 2023г.

Преподаватель: каф. 806 Сысоев Максим  
Алексеевич

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

Подпись преподавателя  
\_\_\_\_\_

## 1. Тема:

Программирование на интерпретируемых командных языках.

## 2. Цель работы:

Составить программу выполнения заданных действий над файлами на одном из интерпретируемых командных языков ОС UNIX

## 3. Задание:

Вариант 24. Запись имен и размеров всех невыполняемых файлов с указанным суффиксом в заданный файл, размер которых кратен размеру блока

## 4. Оборудование:

Процессор AMD ryzen 7 5800U 8x 3.9GH с ОП 16384 Мб, НМД 512Гб. Монитор 1920x1080.

## 5. Программное обеспечение:

Операционная система семейства: *linux*, наименование: *ubuntu*, версия *18.10 cosmic*  
интерпретатор команд: *bash* версия *4.4.19*.

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 решения задачи:

Считать данные пользователя (имя выходного файла, суффикс). Далее пробежаться по каждому файлу в текущей директории, проверяя заданные в задании условия (отсутствие исполняемости, кратность размеру блока, наличие в имени заданного суффикса) при выполнении всех этих условий записать имя файла и размер файла в выходной файл.

## 7. Сценарий выполнения работы:

- 1) Считать пользовательский ввод
- 2) Запустить цикл for для перебора всех файлов в текущей директории
- 3) Проверить тип файла (чтобы был файлом, а не папкой)
- 4) Проверить файл на права доступа (исполняемость)
- 5) Проверить наличие суффикса
- 6) Проверить кратность размера файла
- 7) Вывести результат работы программы в выходной файл

## 8. Распечатка протокола:

### BASH:

```
1  #!/bin/bash
2
3  while [ -n "$1" ]
4  do
5      case "$1" in
6          -sf) suffix="$2"
7              shift ;;
8          -o) outFile="$2"
9              shift ;;
10         --) shift
11         break ;;
12         *) echo "$1 is not an option";;
13     esac
14     shift
15 done
16
17 path=$(pwd)
18 blockSize=2
19
20 rm $outFile
21
22 for file in "$path"/*
23 do
24     if [[ -f $file ]]
25     then
26         if ! [[ -x $file ]]
27         then
28             if [[ $file == *"$suffix".* ]]
29             then
30                 size=$(wc -c $file | awk '{print $1}')
31                 if [[ $(( $size%$blockSize )) -eq 0 ]]
32                 then
33                     echo "$file $size" | tee -a $outFile
34                 fi
35             fi
36         fi
37     fi
38 done
```

## Python:

```
1 import argparse
2 import datetime
3 import os
4
5
6 def checkSuffix(suffix, file):
7     fileCopy = file
8     fileCopy = fileCopy.split('.')
9     fileCopy = fileCopy[0]
10
11     cutBegin = len(fileCopy) - len(suffix)
12     cutEnd = len(fileCopy)
13
14     fileCopy = fileCopy[cutBegin : cutEnd]
15
16     return fileCopy == suffix
17
18
19 def writeData(suffix, outFile):
20     outFile = os.path.abspath(outFile)
21     fileList = os.listdir()
22     blockSize = 2
23     fileOut = open(outFile, 'w')
24
25     for file in fileList:
26         file = os.path.abspath(file)
27         if os.path.isfile(file):
28
29             if os.access(file, os.X_OK) == False:
30
31                 if checkSuffix(suffix, file):
32                     fileSize = os.stat(file).st_size
33
34                     if fileSize % blockSize == 0:
35                         fileOut.write(f'echo {file} {fileSize}\n')
36
```

```
37
38 def main():
39     now = datetime.datetime.now()
40     nowTime = now.time()
41
42     parser = argparse.ArgumentParser()
43     parser.add_argument("-sf", "--suffix", type=str, help='End of file name', required=True)
44     parser.add_argument("-o", "--outFile", type=str, help='Path to output file', default=f'{nowTime}.txt')
45
46     args = parser.parse_args()
47
48     suffix = args.suffix
49     outFile = args.outFile
50
51     writeData(suffix, outFile)
52
53
54 if __name__ == "__main__":
55     main()
```

## 9. Дневник отладки:

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание
1	лаб	21.02.2023	17:20	Не мог понять, как проверить исполняемость	Загуглил	Мне грустно

## 10. Замечания автора:

Замечаний нет.

## 11. Выводы:

Было очень интересно изучать новый для меня язык Bash, он очень неудобный, но думаю пригождается в некоторых случаях, так как по умолчанию установлен в UNIX. Было очень много информации приходилось многое искать в Интернете, но, как все говорят, для программиста это база. Потраченного времени не жалею, так как узнал много новой и интересной информации.

После написания кода на языке bash, захотелось сменить направления на философию. Ужасный, непонятный, сложный синтаксис. Но зато bash не нужно устанавливать на ПК, и команды в нем аналогичны консольным, что тоже, хоть и немного упрощает написание кода. На этом его преимущества для меня закончились.

Python, после Bash'а это глоток свежего воздуха. Понятный синтаксис, большое количество различных расширений для IDEшек и редакторов кода, множество библиотек, позволяющих писать простой и понятный код. Благодаря всему этому, программа на Python выглядит очень красиво и понятно, хоть и немного длиннее.

Мой выбор очевиден – это Python.

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

Подпись студента

---