

Отчет по лабораторной работе № 21 по курсу «Практикум программирования»

Студент группы М8О-109Б-22 Концебалов Олег Сергеевич

Контакты: telegram @baronpipistron

Работа выполнена: 07.03.2023

Преподаватель: каф.806 Сысоев Максим Алексеевич

Отчет сдан «7» марта 2023г., итоговая оценка ____

Подпись преподавателя _____

1. Тема: Программирование на интерпретируемых командных языках

2. Цель работы: Написать программу на языке Python и Bash script, которые будут выполнять действия указанные в задании

3. Задание (вариант № 16): Удаление файлов с указанным префиксом и максимальными размерами, пока общий размер остающихся файлов не станет меньше заданного числа

4. Оборудование (студента):

Процессор AMD Ryzen 5 5600H with Radeon Graphics 3.30 GHz, ОП 16,0 Гб, SSD 512 Гб. Монитор 1920x1080 144 Hz

5. Программное обеспечение (студента):

Операционная система семейства Linux, наименование Ubuntu, версия 18.10

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 (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)

Идея заключается в том, что на вход из командной строки подаются два флага – один содержит префикс, второй – число (до какого размера удалять). После этого ищем файлы с таким префиксом по всему компьютеру и заносим их в список. Если таких файлов нет, то завершаем работу программы. После этого сортируем список файлов по размеру и начинаем удалять самые большие файлы, пока общий размер оставшихся файлов не станет меньше указанного числа. Так же ведем подсчет количества удаленных файлов. Выводим, какой файл был удален, а в конце общее количество удаленных файлов.

7. Сценарий выполнения работы (план работы, первоначальный текст программы в черновике [можно на отдельном листе] и тесты, либо соображения по тестированию)

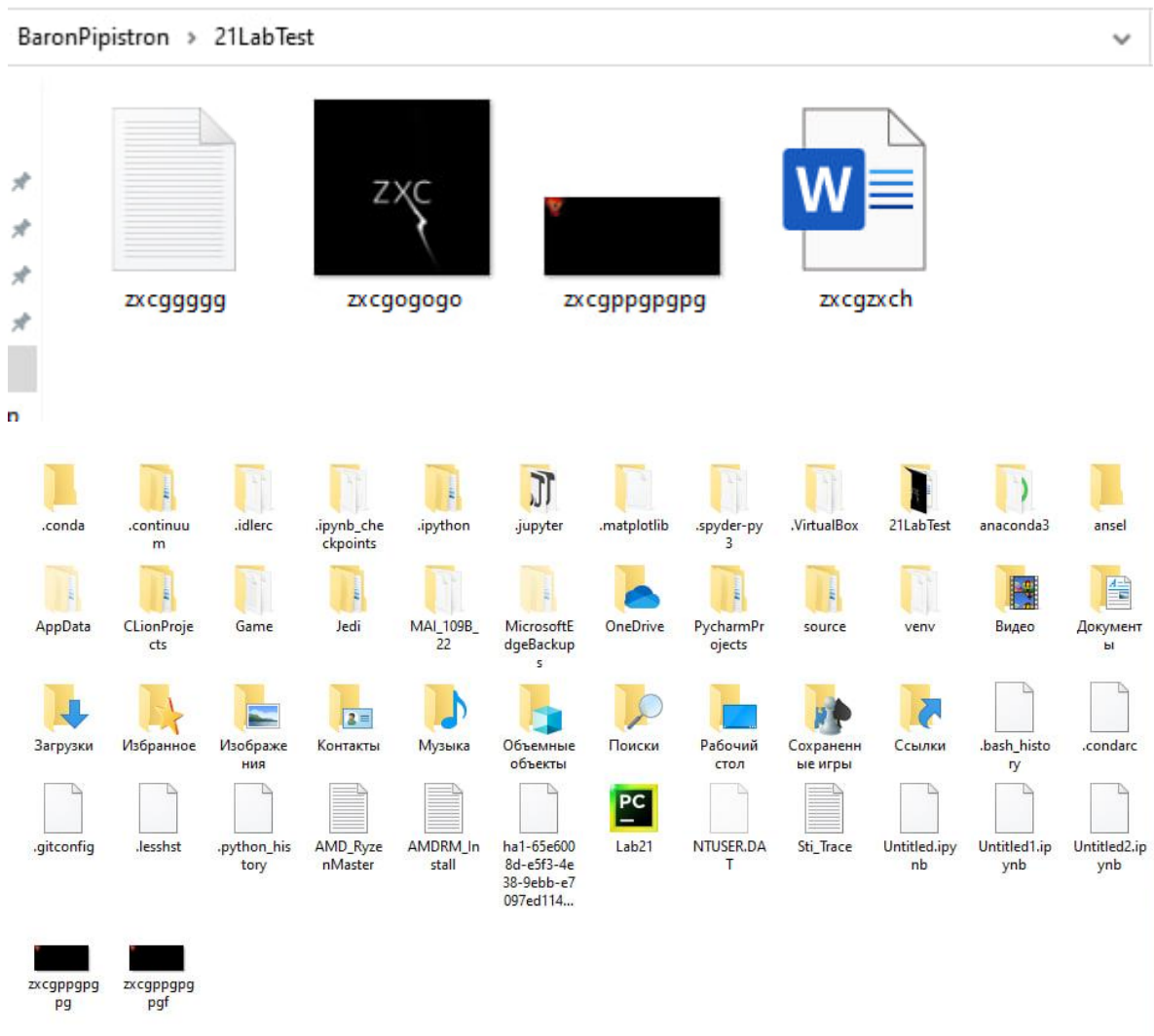
1. Знакомлюсь с библиотеками Python'a, необходимыми для выполнения задания, и их основными функциями
2. Пишу код на Python'e

3. Пишу Bash script
4. Тестирую оба кода

Тестирование

1) Решение на Python'e

Файлы до запуска программы:



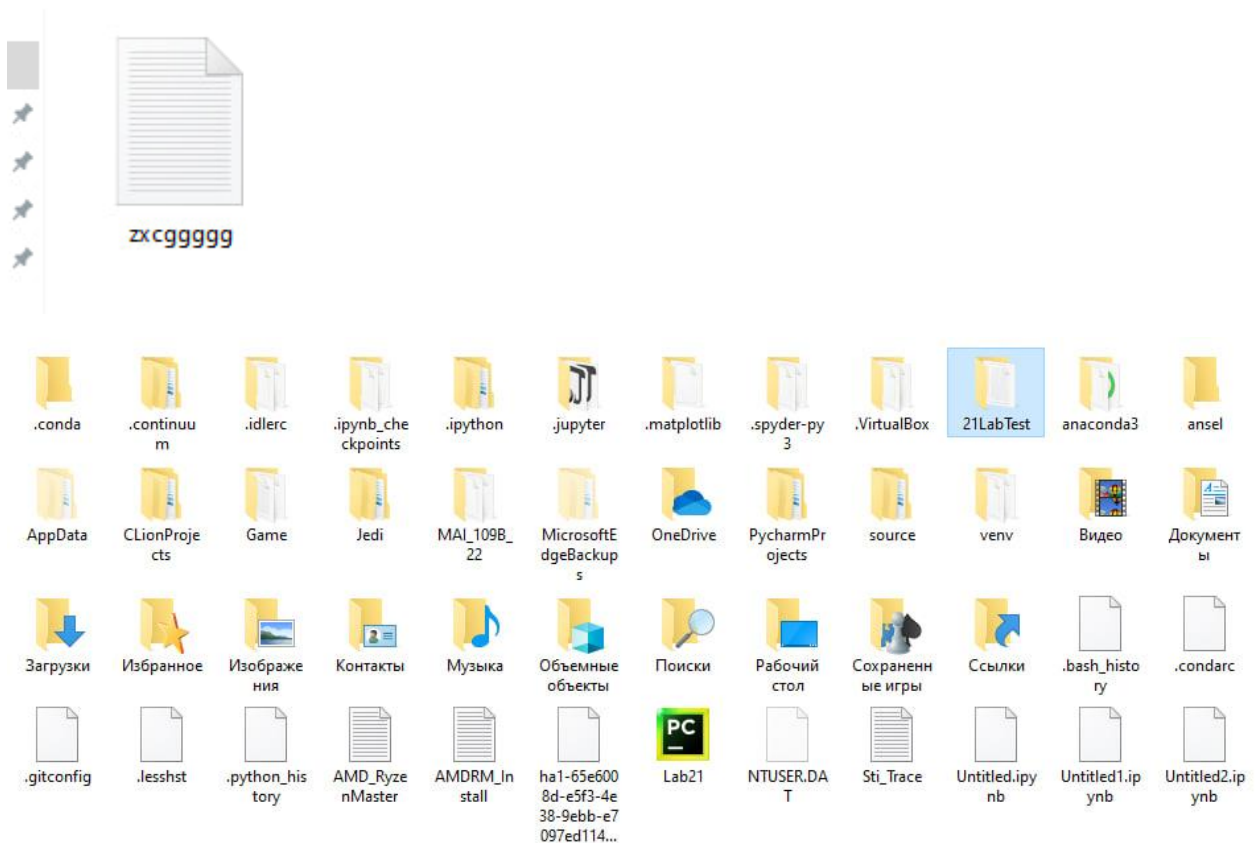
Вывод в терминале после запуска:

```
C:\Users\BaronPipistron>python Lab21.py --prefix=zxcg --size=15
Deleted file: /Users\BaronPipistron\21LabTest\zxcgpppgpgpg.bmp
Deleted file: /Users\BaronPipistron\zxcgpppgpgpgf.bmp
Deleted file: /Users\BaronPipistron\zxcgpppgpgpg.bmp
Deleted file: /Users\BaronPipistron\21LabTest\zxcgzxch.docx
Deleted file: /Users\BaronPipistron\21LabTest\zxcgogogo.jfif
Deleted file: /Users\BaronPipistron\AppData\Roaming\Microsoft\Office\Recent\zxcgzxch.LNK
Deleted file: /Users\BaronPipistron\AppData\Roaming\Microsoft\Windows\Recent\zxcgpppgpgpg.lnk
Deleted file: /Users\BaronPipistron\AppData\Roaming\Microsoft\Windows\Recent\zxcgogogo.lnk
Deleted file: /Users\BaronPipistron\AppData\Roaming\Microsoft\Windows\Recent\zxcgzxch.lnk
Deleted file: /Users\BaronPipistron\AppData\Roaming\Microsoft\Windows\Recent\zxcggggg.lnk

Number of deleted files: 10
```

Файлы после запуска программы:

BaronPipistron > 21LabTest



2) Решение с помощью bash script'a

Вывод в терминале:

```
baronpipistron@baronpipistron-virtual-machine:~$ nano Lab21.sh
baronpipistron@baronpipistron-virtual-machine:~$ bash ./Lab21.sh -p shadowra -s 1
Deleted file: /home/baronpipistron/MAI/shadowrazev.jpg
Deleted file: /home/baronpipistron/zxc/shadowrazex.webp
Deleted file: /home/baronpipistron/shadowrazec.jpeg
Deleted file: /home/baronpipistron/zxc/shadowrazex.html
Number of delted files: 4
baronpipistron@baronpipistron-virtual-machine:~$
```

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

1) Решение задачи на Python'e

```
import argparse
import os

def get_args():
    parser = argparse.ArgumentParser(description='Properties of deleted files')
    parser.add_argument('-p', '--prefix', type=str, help='Input prefix of files')
    parser.add_argument('-s', '--size', type=int, help='Total size of remaining files (in bytes)')
```

```

args = parser.parse_args()

return (args.prefix, args.size)

def remove_files(prefix, size):
    files_list = []

    for root, dirs, files in os.walk('/'):
        for file in files:
            if file.startswith(prefix):
                files_list.append(os.path.join(root, file))

    if not files_list:
        print(f'Files with prefix {prefix} were not found')
        exit(0)

    files_list.sort(key=lambda f: os.path.getsize(os.path.join(root, f)))
    files_size = sum(os.path.getsize(os.path.join(root, file)) for file in
files_list)

    count_delete_files = 0

    while size < files_size:
        file_delete = files_list.pop(-1)
        path_to_file = os.path.join(root, file_delete)
        files_size -= os.path.getsize(path_to_file)
        os.remove(path_to_file)
        count_delete_files += 1
        print(f'Deleted file: {file_delete}')

    print(f'\nNumber of deleted files: {count_delete_files}')

def main():
    prefix, size = get_args()
    remove_files(prefix, size)

if __name__ == '__main__':
    main()

```

2) Решение задачи с помощью bash script'a

Код скрипта (фото):

```
GNU nano 6.2                               Lab21.sh
#!/bin/bash

while getopts ":p:s:" ARG;
do
    case "$ARG" in
    p) prefix="$OPTARG" ;;
    s) size="$OPTARG" ;;
    esac
done

suitable_files_list=$(sudo find / -path "/run" -prune -o -type f -name "${prefix}*" -printf "%s %p\n"|sort -n -r)
files_size=$(echo "$suitable_files_list"|awk '{sum+=$1} END {print sum}')
number_of_files=$(echo "$suitable_files_list"|wc -l)

if [[ "$number_of_files" -eq 0 ]]; then
    echo "Files with prefix $prefix were not found";
    exit 1;
fi

count_deleted_files=0

while [[ "$files_size" -gt "$size" ]];
do
    biggest_file=$(echo "$suitable_files_list"|head -n 1|awk '{print $2}')

    if [[ -r "$biggest_file" ]]; then
        echo "Deleted file: $biggest_file"
        rm "$biggest_file"

        count_deleted_files=$((count_deleted_files+1))
        suitable_files_list=$(echo "$suitable_files_list"|tail -n +2)
    else
        suitable_files_list=$(echo "$suitable_files_list"|tail -n +2)
    fi

    files_size=$(echo "$suitable_files_list"|awk '{sum+=$1} END {print sum}')
done

echo "Number of delted files: $count_deleted_files"
```

Код скрипта (текстом):

```
#!/bin/bash
```

```
while getopts ":p:s:" ARG;
```

```
do
```

```
    case "$ARG" in
```

```
p) prefix="$OPTARG" ;;
```

```
s) size="$OPTARG" ;;
```

```
    esac
```

```
done
```

```
suitable_files_list=$(sudo find / -path "/run" -prune -o -type f -name "${prefix}*" -printf "%s
%p\n"|sort -n -r)
```

```
files_size=$(echo "$suitable_files_list"|awk '{sum+=$1} END {print sum}')
```

```

number_of_files=$(echo "$suitable_files_list"|wc -l)

if [[ "$number_of_files" -eq 0 ]]; then
    echo "Files with prefix $prefix were not found";
    exit 1;
fi

count_deleted_files=0

while [[ "$files_size" -gt "$size" ]];
do
    biggest_file=$(echo "$suitable_files_list"|head -n 1|awk '{print $2}')

    if [[ -r "$biggest_file" ]]; then
        echo "Deleted file: $biggest_file"
        rm "$biggest_file"

        count_deleted_files=$((count_deleted_files+1))
        suitable_files_list=$(echo "$suitable_files_list"|tail -n +2)
    else
        suitable_files_list=$(echo "$suitable_files_list"|tail -n +2)
    fi

    files_size=$(echo "$suitable_files_list"|awk '{sum+= $1} END {print sum}')
done

echo "Number of deleted files: $count_deleted_files"

```

9. Дневник отладки (дата 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	Примечания
1	Дом	05.03.23	15:14	Программа на Python ищет файлы только в исходной директории, не заходит в поддиректории	Неправильно указал параметр в функции os.walk – поставил . Надо было /	Python кайф
2	Дом	06.03.23	18:38	Bash не запускается	Оказалось, что не поставил пробел в if и while после [и перед]	Слишком требовательный к синтаксису
3	Дом	06.03.23	19:01	Заходит не во все директории	Как и в Python'е поменял . на / в find	
4	Дом	06.03.23	21:30	Выводит отказано в доступе для некоторых файлов. Вывод получается некрасивым, надо фиксировать	Много всего пробовал, в итоге прописал в find команду, которая не заходит в определенную директорию (run)	

10. Замечания автора (по существу работы)

Писать на bash очень больно. Очень....

11. Вывод

В целом лабораторная понравилась. Писать на Python'е было просто прекрасно. Очень приятно вернуться на свой основной язык, особенно после C. А вот когда дело дошло до bash script'а начался ад. Писать его было очень больно, неприятно и неудобно. Bash мне вообще не нравится, надеюсь больше с ним не встречусь. Крайне неприятный язык, с не интуитивным синтаксисом, было слишком много гугла...

Работа на 9/10 (все вытащил Python)

Подпись студента _____