

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №1
дисциплины «Анализ данных»

Выполнил:
Кожуховский Виктор Андреевич
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной
техники и автоматизированных систем
», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 г.

Порядок выполнения работы:

1. Изучил теоретический материал работы.
2. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python.
3. Выполнил клонирование созданного репозитория.
4. Дополнил файл .gitignore необходимыми правилами для работы с IDE PyCharm.
5. Организовал свой репозиторий в соответствии с моделью ветвления git-flow.
6. Создал проект в папке репозитория.
7. Проработал примеры лабораторной работы.



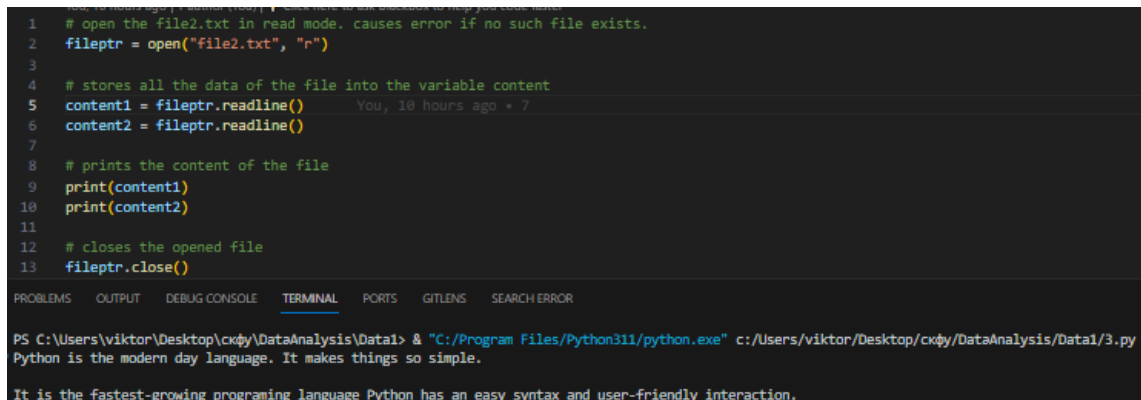
```
1 # open the file2.txt in write mode.
2 fileptr = open("file2.txt", "a")
3     You, 10 hours ago * 7
4 # overwriting the content of the file
5 fileptr.write(" Python has an easy syntax and user-friendly interaction.")
6
7 # closing the opened file
8 fileptr.close()
9
```

file2.txt - Notepad

File Edit Format View Help

Python is the modern day language. It makes things so simple.
It is the fastest-growing programing language Python has an easy syntax and user-friendly interac

Рисунок 1. Результат выполнения кода примера 1

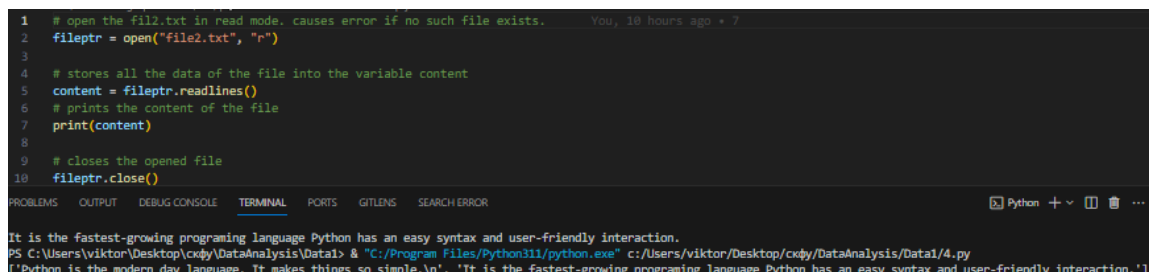


```
1 # open the file2.txt in read mode. causes error if no such file exists.
2 fileptr = open("file2.txt", "r")
3
4 # stores all the data of the file into the variable content
5 content1 = fileptr.readline()     You, 10 hours ago * 7
6 content2 = fileptr.readline()
7
8 # prints the content of the file
9 print(content1)
10 print(content2)
11
12 # closes the opened file
13 fileptr.close()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS SEARCH ERROR

PS C:\Users\viktor\Desktop\сф\DataAnalysis\Data1> & "C:/Program Files/Python311/python.exe" c:/Users/viktor/Desktop/сф/DataAnalysis/Data1/3.py
Python is the modern day language. It makes things so simple.
It is the fastest-growing programing language Python has an easy syntax and user-friendly interaction.

Рисунок 2. Результат выполнения кода примера 2 и 3



```
1 # open the file2.txt in read mode. causes error if no such file exists.     You, 10 hours ago * 7
2 fileptr = open("file2.txt", "r")
3
4 # stores all the data of the file into the variable content
5 content = fileptr.readlines()
6 # prints the content of the file
7 print(content)
8
9 # closes the opened file
10 fileptr.close()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS SEARCH ERROR

Python + - - -

It is the fastest-growing programing language Python has an easy syntax and user-friendly interaction.
PS C:\Users\viktor\Desktop\сф\DataAnalysis\Data1> & "C:/Program Files/Python311/python.exe" c:/Users/viktor/Desktop/сф/DataAnalysis/Data1/4.py
['Python is the modern day language. It makes things so simple.\n', 'It is the fastest-growing programing language Python has an easy syntax and user-friendly interaction.']

Рисунок 3. Результат выполнения кода примера 4

```
1 # open the newfile.txt in read mode. causes error if no such file exists.
2 fileptr = open("newfile.txt", "x")
3 print(fileptr)
4
5 if fileptr:
6     print("File created successfully")
7
8 # closes the opened file
9 fileptr.close()
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS GITLENS SEARCH ERROR

PS C:\Users\viktor\Desktop\сф\DataAnalysis\Data1> & "C:/Program Files/Python311/python
<_io.TextIOWrapper name='newfile.txt' mode='x' encoding='cp1251'>
File created successfully

Рисунок 4. Результат выполнения кода примера 5

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 if __name__ == "__main__":
5     # open the text.txt in append mode. Create a new file if no such file exists.
6     with open("text.txt", "w", encoding="utf-8") as fileptr:
7         # appending the content to the file
8         print(
9             "UTF-8 is a variable-width character encoding used for electronic communication.",
10            file=fileptr
11        )
12        print(
13            "UTF-8 is capable of encoding all 1,112,064 valid character code points.",
14            file=fileptr
15        )
16        print(
17            "In Unicode using one to four one-byte (8-bit) code units.",
18            file=fileptr
19        )
20
```

text.txt - Notepad

File Edit Format View Help

UTF-8 is a variable-width character encoding used for electronic communication.
UTF-8 is capable of encoding all 1,112,064 valid character code points.
In Unicode using one to four one-byte (8-bit) code units.

Рисунок 5. Результат выполнения кода примера 6

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 if __name__ == "__main__":
5     with open("text.txt", "r", encoding="utf-8") as fileptr:
6         sentences = fileptr.readlines()
7
8     # Вывод предложений с запятыми.
9     for sentence in sentences:
10         if "," in sentence:
11             print(sentence)
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS GITLENS SEARCH ERROR

PS C:\Users\viktor\Desktop\сф\DataAnalysis\Data1> & "C:/Program Files/
UTF-8 is capable of encoding all 1,112,064 valid character code points.

Рисунок 6. Результат выполнения кода примера 7

```
1 # open the file file2.txt in read mode
2 with open("file2.txt", "r") as fileptr:
3     # initially the filepointer is at 0
4     print("The filepointer is at byte :", fileptr.tell())
5
6     # changing the file pointer location to 10.
7     fileptr.seek(10)
8
9     # tell() returns the location of the fileptr.
10    print("After reading, the filepointer is at:", fileptr.tell())
11
```

PS C:\Users\viktor\Desktop\скфу\DataAnalysis\Data1> & "C:/Program Files/Python3

The filepointer is at byte : 0

After reading, the filepointer is at: 10

Рисунок 7. Результат выполнения кода примера 8

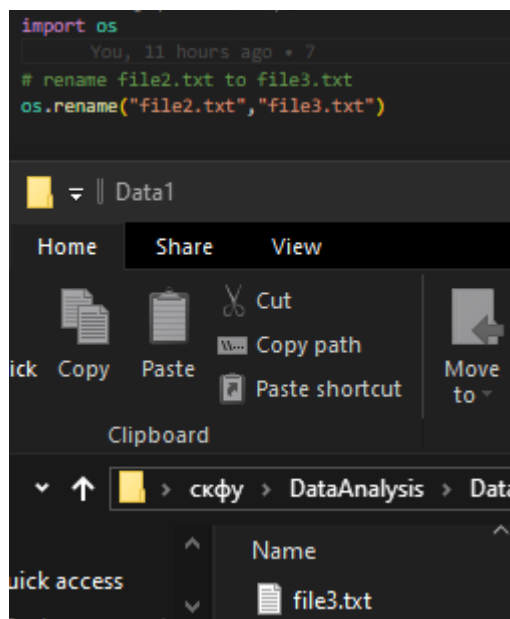


Рисунок 8. Результат выполнения кода примера 9

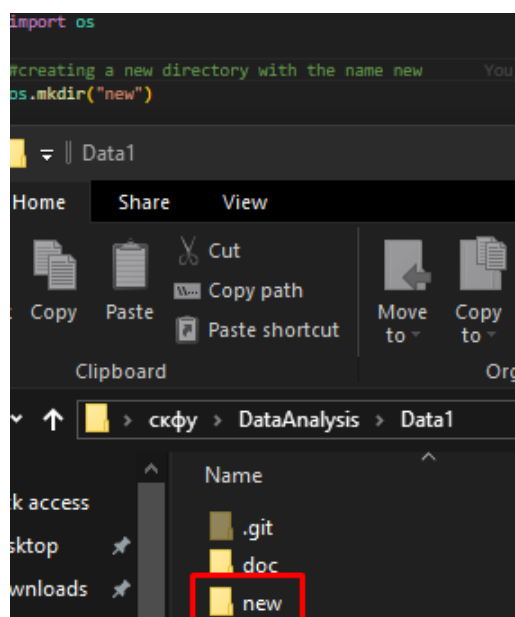


Рисунок 9. Результат выполнения кода примера 11

```
1 import os
2
3 path = os.getcwd()
4 print(path)
5
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS SEARCH ERROR

PS C:\Users\viktor\Desktop\скф\DataAnalysis\Data1> & "C:/Program Files/Python311/python.exe" c:/Users/viktor/Desktop/скф/DataAnalysis/Data1/12.py
C:\Users\viktor\Desktop\скф\DataAnalysis\Data1

Рисунок 10. Результат выполнения кода примера 12

```
1 import os
2
3 # Changing current directory with the new directory
4 os.chdir("C:\\Windows")
5 # It will display the current working directory
6 print(os.getcwd())
7
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS SEARCH ERROR

PS C:\Users\viktor\Desktop\скф\DataAnalysis\Data1> & "C:/Program Files/Python311/python.exe" c:/Users/viktor/Desktop/скф/DataAnalysis/Data1/13.py
C:\Windows

Рисунок 11. Результат выполнения кода примера 13

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 import sys
4
5 if __name__ == "__main__":
6     print("Number of arguments:", len(sys.argv), "arguments")
7     print("Argument List:", str(sys.argv))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS SEARCH ERROR Python

PS C:\Users\viktor\Desktop\скф\DataAnalysis\Data1> & "C:/Program Files/Python311/python.exe" c:/Users/viktor/Desktop/скф/DataAnalysis/Data1/15.py 123 5 23
Number of arguments: 4 arguments
Argument List: ['c:/Users/viktor/Desktop/скф/DataAnalysis/Data1/15.py', '123', '5', '23']

Рисунок 12. Результат выполнения кода примера 15

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import sys
5
6
7 if __name__ == "__main__":
8     for idx, arg in enumerate(sys.argv):
9         print(f"Argument #{idx} is {arg}")
10    print("No. of arguments passed is ", len(sys.argv))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS SEARCH ERROR Python

PS C:\Users\viktor\Desktop\скф\DataAnalysis\Data1> & "C:/Program Files/Python311/python.exe" c:/Users/viktor/Desktop/скф/DataAnalysis/Data1/16.py asdj e
Argument #0 is c:/Users/viktor/Desktop/скф/DataAnalysis/Data1/16.py
Argument #1 is asdj
Argument #2 is e
No. of arguments passed is 3

Рисунок 13. Результат выполнения кода примера 16

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import os
5 import secrets
6 import string
7 import sys
8
9 if __name__ == "__main__":
10     if len(sys.argv) != 2:
11         print("The password length is not given!", file=sys.stderr)
12         sys.exit(1)
13
14     chars = string.ascii_letters + string.punctuation + string.digits
15     length_pwd = int(sys.argv[1])
16
17     result = []
18     for _ in range(length_pwd):
19         idx = secrets.SystemRandom().randrange(len(chars))
20         result.append(chars[idx])
21
22     print(f"Secret Password: {''.join(result)}")
```

PS C:\Users\viktor\Desktop\cxfy\DataAnalysis\Data1> & "C:/Program Files/Python311/python.exe" c:/Users/viktor/Desktop/cxfy/DataAnalysis/Data1/17.py 150
Secret Password: MC`F"j1cdFte+4ScIU"(F\$.byfl*-l]=1:9'4#84oIRT~8bQ2vc52&I;hKtEN)R#8b]rcS^~:HMf\~y5bu/gzIRCAtbp<s^!my5&.tq-3x7vFpJLlIFGDC9PkDs;p\w,8&<B*5!fXtc\$%8A*yY/GtR

Рисунок 14. Результат выполнения кода примера 17

8. Выполнил индивидуальное задание.

Написать программу, которая считывает текст из файла и выводит его на экран, после каждого предложения добавляя, сколько раз встретилось в нем введенное с клавиатуры слово.

```
1 #!/usr/bin/env python3
2 # coding: utf-8 -*-
3
4 cnt = 0
5
6 if __name__ == "__main__":
7     word = input("Введите слово для поиска: ")
8
9     with open("is1.txt", "r", encoding="utf-8") as f:
10         text = f.read()
11         sentences = text.split('.')
12
13     # Вывод предложений с запятыми.
14     for sentence in sentences:
15         cnt = 0
16         if word in sentence:
17             cnt += 1
18         print(sentence, "Искомое слово встретилось:", cnt, "раз(a).")
```

PS C:\Users\viktor\Desktop\cxfy\Анализ_данных\Data1> & "C:/Program Files/Python311/python.exe" c:/Users/viktor/Desktop/cxfy/Анализ_данных/Data1/Main.py
Введите слово для поиска: с
Диагностическая утилита, предназначенная для определения маршрута до точки назначения с помощью поиска эхо-пакетов протокола ICMP с различными значениями срока жизни (TTL, Time-To-Live) Искомое слово встретилось: 1 раз(a).
При этом требуется, чтобы каждый маршрутизатор на пути следования пакетов уменьшал эту величину по крайней мере на 1 перед дальнейшей пересылкой пакета Искомое слово встретилось: 1 раз(a).
Это делает параметр TTL эффективным счетчиком числа ретрансляций Искомое слово встретилось: 1 раз(a).
Предполагается, что когда параметр TTL становится равен 0, маршрутизатор посылает системе-источнику сообщение ICMP «Time Exceeded» Искомое слово встретилось: 1 раз(a).
Утилита traceroute определяет маршрут путем послки первого эхо-пакета с параметром TTL, равным 1, и с последующим увеличением этого параметра на единицу до тех пор, пока не будет получен ответ из точки назначения или не будет достигнуто максимальное допустимое значение TTL Искомое слово встретилось: 1 раз(a).
Маршрут определяется проверкой сообщений ICMP «Time Exceeded», полученных от промежуточных маршрутизаторов Искомое слово встретилось: 1 раз(a).
Однако некоторые маршрутизаторы сбрасывают пакеты с истекшим временем жизни без отправки соответствующего сообщения Искомое слово встретилось: 1 раз(a).
Эти маршрутизаторы невидимы для утилиты traceroute Искомое слово встретилось: 0 раз(a).
Синтаксис утилиты traceroute: Искомое слово встретилось: 1 раз(a).

Рисунок 15. Код решения индивидуального задания 1 и его выполнение

В операционных системах на базе Unix обычно присутствует утилита с названием head. Она выводит первые десять строк содержимого файла, имя которого передается в качестве аргумента командной строки. Напишите программу на Python, имитирующую поведение этой утилиты. Если файла, указанного пользователем, не существует, или не задан аргумент командной строки, необходимо вывести соответствующее сообщение об ошибке.

```

1  #!/usr/bin/env python3
2  # coding: utf-8 -*-
3
4  import sys
5
6
7  cnt = 0
8
9  if __name__ == "__main__":
10     if len(sys.argv) < 2:
11         print("Ошибка: не указан файл.")
12     else:
13         filename = sys.argv[1]
14         try:
15             with open(sys.argv[1], "r", encoding="utf-8") as file:
16                 lines = file.readlines()[1:10]
17                 for line in lines:
18                     print(line.strip())
19         except FileNotFoundError:
20             print(f"Ошибка: файл '{filename}' не найден.")
21         except:
22             print("Произошла ошибка при чтении файла.")
23

```

PS C:\Users\viktor\Desktop\схф\DataAnalysis\Data1> & "C:/Program Files/Python311/python.exe" c:/Users/viktor/Desktop/схф/DataAnalysis/Data1/Main2.py is2.txt
 1
 2
 3
 4
 5
 6
 78
 9
 8
 7
 PS C:\Users\viktor\Desktop\схф\DataAnalysis\Data1> & "C:/Program Files/Python311/python.exe" c:/Users/viktor/Desktop/схф/DataAnalysis/Data1/Main2.py is2.tt
 Ошибка: файл 'is2.tt' не найден.

Рисунок 16. Код решения индивидуального задания 2 и его выполнение

9. Зафиксировал сделанные изменения в репозитории.

10. Самостоятельно подобрал задачу для работы с изученными функциями модуля os. Привел решение этой задачи.

Используя библиотеку os в python создать в папках от Data2 до Data7 папки doc и скопировать из папки Data1\doc документ "Лр 1 Кожуховский.docx" в эти папки док, заменяя цифру после слова Лр.

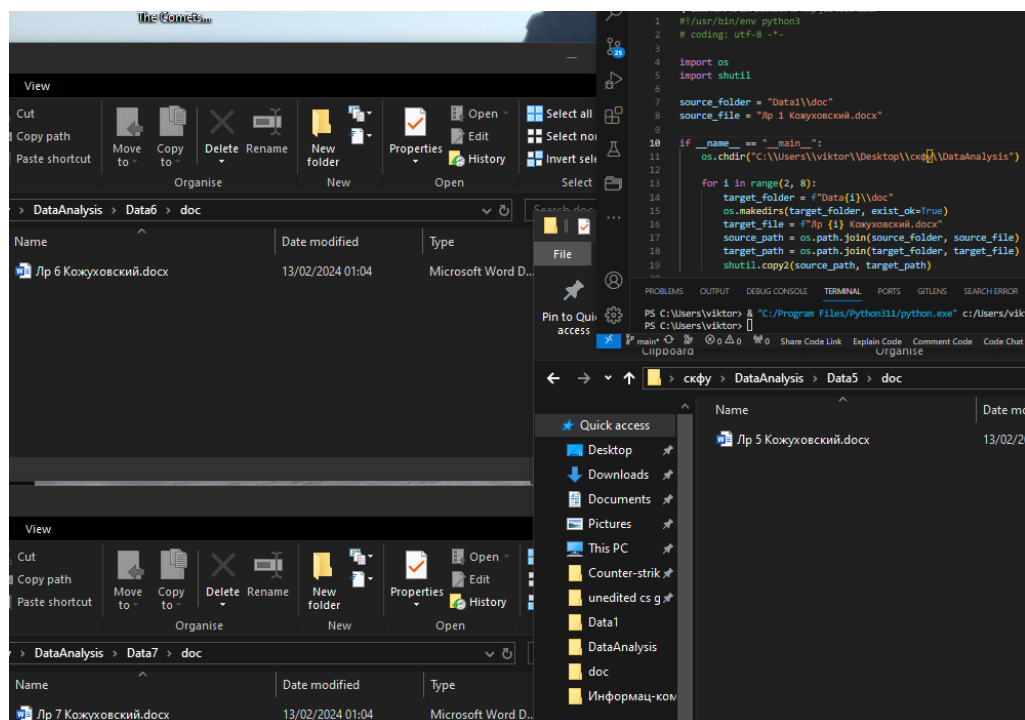


Рисунок 17. Код решения придуманного задания и его выполнение

10. Добавил отчет по лабораторной работе в формате PDF в папку doc репозитория. Зафиксировал изменения.

11. Выполнил слияние ветки для разработки с веткой master/main.

12. Отправил сделанные изменения на сервер GitHub.

Контрольные вопросы:

1. Как открыть файл в языке Python только для чтения?

Чтобы открыть файл для чтения, мы используем режим r или rb. `file = open("text.txt", 'r')`.

2. Как открыть файл в языке Python только для записи?

Чтобы открыть файл для записи, мы используем режим w или wb. `file = open("text.txt", 'w')`. Также можно использовать режим a или ab, чтобы не перезаписывать файл, а дозаписывать в него.

3. Как прочитать данные из файла в языке Python?

Чтобы прочитать файл с помощью сценария Python, Python предоставляет метод `read()`. Метод `read()` считывает строку из файла. Он может читать данные как в текстовом, так и в двоичном формате. Также можно использовать цикл `for`: `for i in fileptr: print(i)`; также можно использовать метод `readline()`, который читает строки файла с самого начала, т. е. если мы используем его два раза, мы можем получить первые две строки файла. Python также предоставляет метод `readlines()`, который используется для чтения строк. Возвращает список строк до конца файла (EOF).

4. Как записать данные в файл в языке Python?

Запись данных в файл. Записать данные в файл можно с помощью метода `write()`.

5. Как закрыть файл в языке Python?

После того, как мы открыли файл, и выполнили все нужные операции, нам необходимо его закрыть. Для закрытия файла используется функция `close()`.

6. Изучите самостоятельно работу конструкции `with ... as`. Каково ее назначение в языке?

Конструкция with ... as в языке Python предназначена для обеспечения управления ресурсами с автоматическим освобождением этих ресурсов после завершения блока кода. Она часто используется с объектами, которые поддерживают протокол менеджеров контекста.

7. Изучите самостоятельно документацию Python по работе с файлами. Какие помимо рассмотренных существуют методы записи/чтения информации из файла?

Метод `writelines()`:

Данный метод принимает список строк в качестве аргумента и записывает каждую строку списка в файл.

В Python также существует метод `readinto()`, который читает данные из файла и записывает их в предварительно выделенный буфер. Это может быть полезно, если нужно работать с данными напрямую в буфере, минуя дополнительные копирования в памяти.

8. Какие существуют, помимо рассмотренных, функции модуля `os` для работы с файловой системой?

`os.chmod (path, mode, *, dir_fd=None, follow_symlinks=True)` - смена прав доступа к объекту (`mode` - восьмеричное число).

`os.chown (path, uid, gid, *, dir_fd=None, follow_symlinks=True)` - меняет id владельца и группы (Unix).

`os.link (src, dst, *, src_dir_fd=None, dst_dir_fd=None, follow_symlinks=True)` - создаёт жёсткую ссылку.

`os.listdir (path=".")` - список файлов и директорий в папке.

`os.makedirs (path, mode=0o777, exist_ok=False)` - создаёт директорию, создавая при этом промежуточные директории.