



İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA COMPUTER ENGINEERING

FILE ORGANIZATION TERM PROJECT

NAME : MUHAMMET TALHA ODABAŞI
ID : 1306220012
PROGRAMMING LANGUAGE : PYTHON

1. INTRODUCTION

In my project, I carried out all operations with the OS and hashlib library using the Python programming language. The goal of the project is, as stated in the document shared on google classroom, using password lists on SecList to collect and process raw passwords to create an Index file and to facilitate the search process

2. USED TECHNOLOGIES

As I explained in the introduction, I wrote the code using Python 3.11 and libraries I used are:

- I performed file operations using OS library, used functions such as `getcwd()`, `listdir()`
- I calculated MD5, SHA1, SHA256 hash values with the `hashlib` library.
- I calculated how long it took to find the password with the `datetime` library.

3. PROJECT DESIGN

- There are unprocessed password files which are taken from SecLists repository in Unprocessed-Passwords directory.
- Processed directory is used to store password files that are processed. After processing a file it is moved from Unprocessed-Passwords to Processed.
- Index directory holds indexed versions of all passwords. It creates folders with ascii number value of password's first character and stores passwords in a file inside that directory such as 0.txt which holds 10000 records.
- Code directory includes (main.py) program code.

4. INDEXING

```
def create_index_files():
    len_dict = dict()
    pass_dict = get_non_duplicated_passwords()
    os.chdir(os.getcwd().replace("Unprocessed-Passwords", "Index"))
    for key in pass_dict:
        first_char = str(ord(key[0]))
        dir_contents = os.listdir()
        if first_char not in dir_contents:
            if not os.path.exists(first_char):
                os.mkdir(first_char)
                len_dict[first_char] = 1
        index = len_dict[first_char] // 10000
        with open(f"{first_char}/{index}.txt", "a") as open_file:
            open_file.write(f"{key}|{md5(key.encode()).hexdigest()}|{sha1(key.encode()).hexdigest()}|{pass_dict[key]}\n")
            len_dict[first_char] += 1
```

```
def get_non_duplicated_passwords():
    non_duplicated_passwords = dict()
    os.chdir(os.getcwd().replace("Code", "Unprocessed-Passwords"))
    files = os.listdir()

    for file in files:
        with open(file, "r", encoding="utf8") as open_file:
            for line in open_file:
                line = line.strip("\n")
                non_duplicated_passwords[line] = open_file.name

    if len(files) == 0:
        print("Unprocessed-Passwords klasöründe hiç dosya bulunamadı.")
        exit()

    for f in files:
        old_path = os.path.join(os.getcwd(), f)
        new_path = os.path.join(os.getcwd().replace("Unprocessed-Passwords", "Processed"), f)
        os.replace(old_path, new_path)

    return non_duplicated_passwords
```

As seen on above codes, all files inside Unprocessed-Passwords are looped line by line and stored inside a dictionary with key of password and value of source file. It is stored in password to make sure no duplicate passwords are processed.

After that dictionary is looped and all passwords are stored with specified format inside Index folder. Ascii numbers are used as folder names to be able to create folders of special characters. Moreover, len_dict is used to be able to store files up to 10000 lines and then separate.

```
Parolalar okunuyor ve index dosyaları oluşturuluyor...
Geçen süre: 0:09:17.576163
```

This is the time it takes to index all files.

5. SEARCH FUNCTION

```
def find_password():
    input_password = input("Aranacak parolayı giriniz:")
    start_time = datetime.now()
    end_time = 0
    first_char = str(ord(input_password[0]))
    os.chdir(os.getcwd().replace("Code", "Index"))
    os.chdir(os.path.join(os.getcwd(), first_char))
    files = os.listdir()
    for file in files:
        with open(file, "r", encoding="utf8") as open_file:
            for line in open_file:
                line = line.strip("\n").split("|")
                if line[0] == input_password:
                    print()
                    print("PAROLA BULUNDU!\n")
                    print(f"Parola: {line[0]}\nMD5: {line[1]}\nSHA1: {line[2]}\nSHA256: {line[3]}\nKaynak Dosya: {line[4]}\n")
                    end_time = datetime.now()
                    print(f"Geçen süre: {end_time - start_time}")
                    exit()
    print()
    print("Parola bulunamadı.")
    end_time = datetime.now()
    print(f"Geçen süre: {end_time - start_time}")
    add_new_password(input_password)
```

This is the search function. It gets a password input and searches for the password inside Index folder. If password is not found it adds it to Index folder with source as User Input.

```
PS D:\Downloads\1386220012_MuhammetTalhaOdbasi\1386220012_MuhammetTalhaOdbasi\Term Project\Code> python .\main.py
[1] Parolaları oku ve Index dosyalarını oluştur
[2] Parola Arama
[4] Çıkış yap
Seçiminizi yapınız:2
Aranacak parolayı giriniz:test

PAROLA BULUNDU!

Parola: test
MD5: 096f8bcd4621d373cde4e830677b4f6
SHA1: 2b4a8f5c5cb18ba61c4c88734391e987982fbbd3
SHA256: 9f864081884c7d659a2feae0c55ad015a3bf4f1b2b08822cd15d6c15b0f80a08
Kaynak Dosya: worst-passwords-2017-top100-slashdata.txt

Geçen süre: 0:00:00.015625
PS D:\Downloads\1386220012_MuhammetTalhaOdbasi\1386220012_MuhammetTalhaOdbasi\Term Project\Code> python .\main.py
[1] Parolaları oku ve Index dosyalarını oluştur
[2] Parola Arama
[4] Çıkış yap
Seçiminizi yapınız:2
Aranacak parolayı giriniz:Testere1iTalha

Parola bulunamadı.
Geçen süre: 0:00:00.011561
PS D:\Downloads\1386220012_MuhammetTalhaOdbasi\1386220012_MuhammetTalhaOdbasi\Term Project\Code> python .\main.py
[1] Parolaları oku ve Index dosyalarını oluştur
[2] Parola Arama
[4] Çıkış yap
Seçiminizi yapınız:2
Aranacak parolayı giriniz:Testere1iTalha

PAROLA BULUNDU!

Parola: Testere1iTalha
MD5: ef67368e647d2f18a692e759adc28d45
SHA1: f8867cee5c1ca406e4dbda454b9811892a5b5784
SHA256: ab97c6de54445b730798c1f82669d522289f4104f54e86f373e8211bc52cf269
Kaynak Dosya: User Input

Geçen süre: 0:00:00.002327
PS D:\Downloads\1386220012_MuhammetTalhaOdbasi\1386220012_MuhammetTalhaOdbasi\Term Project\Code> python .\main.py
[1] Parolaları oku ve Index dosyalarını oluştur
[2] Parola Arama
[4] Çıkış yap
Seçiminizi yapınız:2
Aranacak parolayı giriniz:monke

PAROLA BULUNDU!

Parola: monke
MD5: 08720b7459c70ae3af3e982d45af5f8e
SHA1: 02e277790eb8469a5ae47c5927074f806a0c5e
SHA256: 671f5cfad9a9af5b43c93b6d527cafc7c135f5998ac11d7e628f2fff91b380b
Kaynak Dosya: 10-million-password-list-top-1000000.txt

Geçen süre: 0:00:00.015437
```

```
PS D:\Downloads\1386220012_MuhammetTalhaOdbasi\1386220012_MuhammetTalhaOdbasi\Term Project\Code> python .\main.py
[1] Parolaları oku ve Index dosyalarını oluştur
[2] Parola Arama
[4] Çıkış yap
Seçiminizi yapınız:2
Aranacak parolayı giriniz:test

PAROLA BULUNDU!

Parola: abcdefg
MD5: 7ac66c9f148de9519b8bd264312c4d64
SHA1: 2f05e1419f8c89240865e7a324f476ec624e8740
SHA256: 7d1e5412792225e2f5b7965f0e0830e1152a44f92b37e23c6527baF665d4da9a
Kaynak Dosya: 10k-most-common.txt

Geçen süre: 0:00:00
PS D:\Downloads\1386220012_MuhammetTalhaOdbasi\1386220012_MuhammetTalhaOdbasi\Term Project\Code> python .\main.py
[1] Parolaları oku ve Index dosyalarını oluştur
[2] Parola Arama
[4] Çıkış yap
Seçiminizi yapınız:2
Aranacak parolayı giriniz:ABCDEF6

PAROLA BULUNDU!

Parola: ABCDEF6
MD5: bb747b3df3130f6ca4fa93fb7d97c9
SHA1: 93be4612c4d23ef1891d4c5f80d535796ff4e3
SHA256: ebd2a2a2e8e53732ac138b31a27607181421c4633c9f41844cca884d482b16
Kaynak Dosya: 10-million-password-list-top-1000000.txt

Geçen süre: 0:00:00
PS D:\Downloads\1386220012_MuhammetTalhaOdbasi\1386220012_MuhammetTalhaOdbasi\Term Project\Code> python .\main.py
[1] Parolaları oku ve Index dosyalarını oluştur
[2] Parola Arama
[4] Çıkış yap
Seçiminizi yapınız:2
Aranacak parolayı giriniz:QWERASD3ASD3CZXCASD3ASD3

Parola bulunamadı.
Geçen süre: 0:00:00.001095
PS D:\Downloads\1386220012_MuhammetTalhaOdbasi\1386220012_MuhammetTalhaOdbasi\Term Project\Code> python .\main.py
[1] Parolaları oku ve Index dosyalarını oluştur
[2] Parola Arama
[4] Çıkış yap
Seçiminizi yapınız:2
Aranacak parolayı giriniz:AAQWERASD3ASD3CZXCASD3ASD3

Parola bulunamadı.
Geçen süre: 0:00:00.001918
PS D:\Downloads\1386220012_MuhammetTalhaOdbasi\1386220012_MuhammetTalhaOdbasi\Term Project\Code> python .\main.py
[1] Parolaları oku ve Index dosyalarını oluştur
[2] Parola Arama
[4] Çıkış yap
Seçiminizi yapınız:2
Aranacak parolayı giriniz:AAQWERASD3ASD3CZXCASD3ASD3

PAROLA BULUNDU!

Parola: AAQWERASD3ASD3CZXCASD3ASD3
MD5: 60ba1f1fca13c8f80e5e18fec46051
SHA1: dcf46e34589f9aaf74234cbb0242bc95a7776bb4
SHA256: a3f8fae6aad58d511372daF71212F8b57542f8d6595d8401d5e7a53ac6cb597
Kaynak Dosya: User Input
```

These are from performance tests. It takes microseconds to find or not find a password. And as you can see, If password is not found on first try it is found on second one.

6. CONCLUSION

As a result of indexing, the search process has visibly accelerated. Considering that this is our purpose I can say that we fulfilled our purpose. Doing the project with C style low level languages would take longer but its indexing and search ability would be faster. Moreover, this project can be used to create mini relational database.

7. SOURCE

I did not use any specific source, I just used stackoverflow and GPT 3.5 to create specific algorithms.