

# Разработка анализатора логов

*выполнено командой Radiance 1018*

Разработанный по условию алгоритм находится в файле main.py(выполнен на языке python v3.10), для запуска требуется файл server.log(или любой другой, но с изменением названия в коде алгоритма).

Программа считывает файл с помощью встроенной библиотеки codecs и функций open, read и splitlines. Далее начинается процесс анализа логов в цикле, он предполагает разделение их по открытию и закрытию сессий XIMSS и IMAP. После разделения начинается запись отрезков времени и количество сессий в массив times. Учтено, что есть сессии начатые вчера, при их определении все предыдущие отрезки времени обновляют кол-во сессий. После начинается процесс определения максимального кол-ва сессий в одно время(Учтено, что таких сессий может быть несколько, как и в данном мне примере. Они выводятся последовательно). После представлен вывод ответа в консоль. Код представлен ниже

```
import codecs

def analyze(wa):
    whojoin = []
    wholeft = []
    users = 0
    times = []
    from_time = '00:00:00'
    max_users = 0
    current_time = 0
    nusers = 0
    loggedyes = 0
    for i in wa:
        if 'XIMSS' in i:
            if 'logged' in i:
                whojoin.append(i[i.find('XIMSS'):i.find('(')])
                nusers += 1
            elif 'closed' in i:
                if i[i.find('XIMSS'):i.find('(')] not in whojoin:
                    loggedyes += 1
            else:
                wholeft.append(i[i.find('XIMSS'):i.find('(')])
                nusers -= 1
        current_time = i[:12]
    elif 'IMAP' in i:
        if 'disconnected' in i:
            if i[i.find('IMAP'):i.find('(')] not in whojoin:
                loggedyes += 1
        else:
            nusers -= 1
```

```

        wholeleft.append(i[i.find('IMAP'):i.find('(')])
    elif 'connected' in i:
        nusers += 1
        whojoin.append(i[i.find('IMAP'):i.find('(')])
        current_time = i[:12]
    times.append([from_time, current_time, users])

if loggedyes > 0:
    new_times = []
    for x, y, z in times:
        new_times.append([x, y, z + 1])
    times = new_times
    users += nusers
    nusers = 0
    from_time = current_time
    loggedyes = 0
max_times = []
for i in times:
    if max_users < i[2]:
        max_times = []
        max_users = i[2]
        max_times.append(f"from {i[0]} to {i[1]} - {i[2]} users")
    elif max_users == i[2]:
        max_times.append(f"from {i[0]} to {i[1]} - {i[2]} users")
    print(f"From {i[0]} to {i[1]} - {i[2]} users")
for i in max_times:
    print(f"The maximum number of users was {i}")

```

```

file = codecs.open("server.log", "r", "utf_8_sig" )
log = file.read()
log = log.splitlines()
log = analyze(log)

```