



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики  
Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота № 2**  
з дисципліни “*Бази даних. Частина 2*”  
на тему “*Практика використання сервера Redis*”

Виконав

студент III курсу

групи КП-81

Подлеснюк Богдан Анатолійович

Зарахована: Петрашенко А. В.

Київ 2021

**Мета роботи:** здобуття практичних навичок створення ефективних програм, орієнтованих на використання сервера Redis за допомогою мови Python.

**Завдання роботи** полягає у наступному:

Реалізувати можливості обміну повідомленнями між користувачами у оффлайн та онлайн режамах із можливістю фільтрації спам-повідомлень.

*Окремі програмні компоненти та вимоги до них*

1. Redis server (RS), що виконує наступні ролі:

1.1. *Сховище*, що містить: дані користувачів, їхні групи (звичайний користувач та адміністратор), а також повідомлення, що пересилаються між ними.

1.2. *Черга повідомлень*, які підлягають перевірці на спам та відправленню адресату.

1.3. Інструмент *Publish/Subscribe* для ведення та розсилання журналу активності користувачів (див. *Список активностей для журналювання*).

2. Інтерфейс користувача (User Interface)

2.1. *Звичайний користувач* має змогу виконувати вхід за ім'ям (без паролю), відправляти та отримувати (переглядати) повідомлення, отримувати дані про кількість **своїх** повідомлень, згрупованих за статусом (див. *Статуси повідомлень*).

2.2. *Адміністратор* має змогу переглядати журнал подій, що відбулись (див. *Список активностей для журналювання*), переглядати список користувачів, які знаходяться online, переглядати статистику (N найбільш активних відправників повідомлень із відповідною кількістю, N найактивніших “спамерів” із відповідною кількістю).

3. *Виконувач (worker)* призначений для:

перегляду черги повідомлень, відбору повідомлення, перевірки його вмісту на наявність спаму (у випадку наявності спаму -- додавання запису

в журнал)

### *Інші вимоги*

1. Проаналізувавши матеріали ресурсів, наведений у пункті “Джерела”, обрати та обґрунтувати вибір структур даних Redis щодо реалізації наведених вище вимог, **обов’язково використати наступні структури даних** та інструменти Redis: List, Hash, Sorted List, Set, Pub/Sub.
2. Забезпечити роботу програмних засобів у режимі емуляції із можливістю генерації повідомлень від різних користувачів, налаштування кількості виконувачів та часу затримки обробки на спам з можливістю підключення адміністратора для перегляду подій, що відбуваються.
3. Перевірку на спам можна проємулювати за допомогою затримки на псевдовипадковий час та генерацію псевдовипадкового результату (Так/Ні).

### *Список активностей для журналювання*

Вхід/вихід користувача, наявність спаму у повідомленні.

### *Статуси повідомлень*

“Створено”, “У черзі”, “Перевіряється на спам”, “Заблоковано через спам”, “Відправлено адресату”, “Доставлено адресату”.

### *Вимоги до інтерфейсу користувача*

Використовувати консольний (текстовий) інтерфейс користувача.

## Обґрунтування вибору структур в Redis

- Хештаблиця - була використана для зберігання повідомлень. Основна причина в тому, що хештаблиця дозволяє зберігати id повідомлення як ключ та ім'я користувача-відправника, користувача-отримувача, текст повідомлення як значення.
- Сет - був використаний для зберігання користувачів, так як як нам важливо, щоб кожен користувач був унікальним. Перевага була надана саме звичайному сету через те, що нам не важливий порядок користувачів, але є важлива швидкість доступу до них.
- Сорований сет - використовується для збереження топ спаммерів, так як нам є важливо задати для спаммерів порядок за кількістю спам повідомлень.
- Список - використовується для черги повідомлень, так як нам є важливий порядок доданих записів, та час додавання та вилучення, який у випадку використання List є константним

## Приклади роботи програми

*Приклад роботи додатку для користувача:*

Реєструємо користувача, авторизуємось та відправляємо повідомлення іншому користувачу. Після цього worker повинен перевірити повідомлення на спам, щоб воно успішно дійшло до іншого користувача

```
(venv) C:\Work\db2\db2-labs\lab2>python -W ignore::DeprecationWarning user_demo.py
***** Main menu *****
1. Register
2. Login
3. Exit
Please, choose action to take: 1
Enter your username:UserTest2
***** Main menu *****
1. Register
2. Login
3. Exit
Please, choose action to take: 2
Enter your login:UserTest2
***** User menu *****
1. Log out
2. Inbox
3. Create message
4. Statistics
Please, choose action to take: 3
Type your message:Hello, UserTest1!
Type the username of the reciever:UserTest1
Sending message...
***** User menu *****
1. Log out
2. Inbox
3. Create message
4. Statistics
Please, choose action to take: 4
In queue: 1
Checking: 0
Blocked: 0
Sent: 0
Delivered: 0
***** User menu *****
1. Log out
2. Inbox
3. Create message
4. Statistics
Please, choose action to take: █
```

Рис.1 - Процес створення користувача та відправлення ним повідомлення

```
(venv) C:\Work\db2\db2-labs\lab2>python -W ignore::DeprecationWarning worker_demo.py  
The message with id 35 was checked and sent.  
The message with id 36 was checked and sent.
```

Рис.2 - Процес перевірки повідомлення на спам

*Приклад роботи додатку для адміністратора:*

```
***** Admin menu *****
1. Users online
2. Top senders
3. Top spammers
4. Exit
Please, choose action to take 1
Users online:
UserTest2
UserTest1
cherylclark
***** Admin menu *****
1. Users online
2. Top senders
3. Top spammers
4. Exit
Please, choose action to take 2
Top 10 senders
1 . user:thull - 6 message(s)
2 . user:murphylauren - 6 message(s)
3 . user:lgood - 6 message(s)
4 . user:cherylclark - 6 message(s)
5 . user:cervantesjackson - 6 message(s)
6 . user:UserTest2 - 2 message(s)
7 . user:UserTest1 - 2 message(s)
***** Admin menu *****
1. Users online
2. Top senders
3. Top spammers
4. Exit
Please, choose action to take 3
Top 10 spammers
1 . user:murphylauren - 5 spammed message(s)
2 . user:thull - 3 spammed message(s)
3 . user:lgood - 3 spammed message(s)
4 . user:cherylclark - 3 spammed message(s)
5 . user:cervantesjackson - 3 spammed message(s)
```

Рис.3 - Перевірка можливостей адміну

*Приклад спаму:*

```
***** Main menu *****
1. Register
2. Login
3. Exit
Please, choose action to take: 2
Enter your login:UserTest2
***** User menu *****
1. Log out
2. Inbox
3. Create message
4. Statistics
Please, choose action to take: 3
Type your message:spam
Type the username of the reciever:UserTest1
Sending message...
***** User menu *****
1. Log out
2. Inbox
3. Create message
4. Statistics
Please, choose action to take: █
```

Рис.4 - Процес відправлення спам-повідомлення

```
(venv) C:\Work\db2\db2-labs\lab2>python -W ignore::DeprecationWarning worker_demo.py
The message with id 37 was blocked. It is spam
█
```

Рис.5 - Процес перевірки повідомлення на спам



*Приклад логів, які залишає додаток:*

```
INFO:root:2021-04-11 20:21:15.717517 Actor: UserTest2 Action: register
INFO:root:2021-04-11 20:21:22.633430 Actor: UserTest2 Action: log in
INFO:root:2021-04-11 20:22:11.956183 Actor: UserTest2 Action: sign out
INFO:root:2021-04-11 20:24:37.470608 Actor: UserTest2 Action: log in
INFO:root:2021-04-11 20:30:10.866310 Actor: UserTest2 Action: sign out
INFO:root:2021-04-11 20:30:21.753732 Actor: UserTest2 Action: log in
INFO:root:
2021-04-11 20:30:52.700603 Event: User UserTest2 sent spam message: "spam"
INFO:root:2021-04-11 20:32:02.277064 Actor: UserTest2 Action: sign out
INFO:root:2021-04-11 20:32:03.816021 Actor: UserTest2 Action: sign out
```

Рис.6 - Приклад логів додатку

Основні переваги та недоліки Redis:

- + Висока надійність та відмовостійкість
  - + Легкий в налаштуванні та використанні
  - + Має багато вбудованих структур даних.
  - + Швидкий в роботі
- 
- При підключенні клієнти повинні знати топологію кластера, що збільшує кількість налаштувань на клієнті
  - Збільшення необхідної кількості ресурсів для роботи при масштабуванні
  - Потребує багато ресурсів ОЗУ, так як зберігає все в оперативній пам'яті
  - Погані інструменти моніторингу

**Висновки:** Під час цієї лабораторної роботи я здобув навички створення програм з використанням Redis та мови програмування Python.