Лабораторна робота №8

Тема: Робота з Redis

Мета: Закріпити розуміння роботи Redis та навчитися використовувати його основні можливості.

Виконав: Бессараб Антон МІТ-31

Хід роботи

1.Встановлення та запуск Redis

Виконав оновлення списку пакетів і встановив Redis:

```
anton-b@DESKTOP-L26HSA4:~$ sudo apt install redis-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libatomic1 libjemalloc2 liblzf1 redis-tools
Suggested packages:
  ruby-redis
```

2.Перевірка Redis після встановлення

Перевірте статус служби:

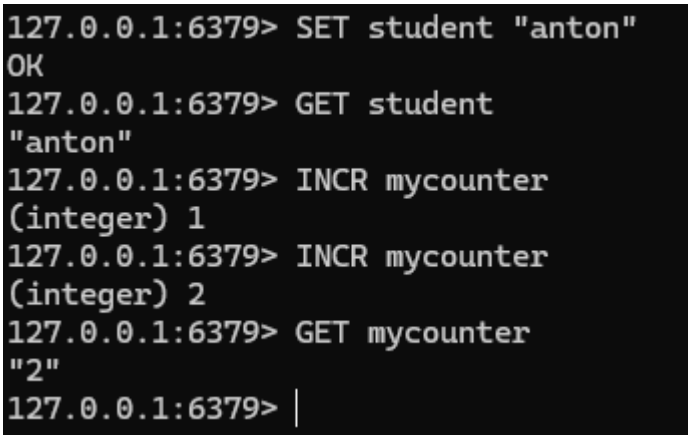
```
anton-b@DESKTOP-L26HSA4:~$ sudo systemctl start redis
anton-b@DESKTOP-L26HSA4:~$ redis-server
942:C 07 May 2025 21:35:50.653 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
942:C 07 May 2025 21:35:50.654 # Redis version=7.0.15, bits=64, commit=00000000, modified=0, pid=942, just started
942:C 07 May 2025 21:35:50.654 # Warning: no config file specified, using the default config. In order to specify a conf
ig file use redis-server /path/to/redis.conf
942:M 07 May 2025 21:35:50.654 * Increased maximum number of open files to 10032 (it was originally set to 1024).
942:M 07 May 2025 21:35:50.654 * monotonic clock: POSIX clock_gettime
942:M 07 May 2025 21:35:50.654 # Warning: Could not create server TCP listening socket *:6379: bind: Address already in
use
942:M 07 May 2025 21:35:50.654 # Failed listening on port 6379 (TCP), aborting.
```

Запускаємо Redis та перевіряємо підключення:

```
anton-b@DESKTOP-L26HSA4:~$ redis-cli
127.0.0.1:6379> PING
PONG
127.0.0.1:6379>
```

3.Виконання операцій з рядками

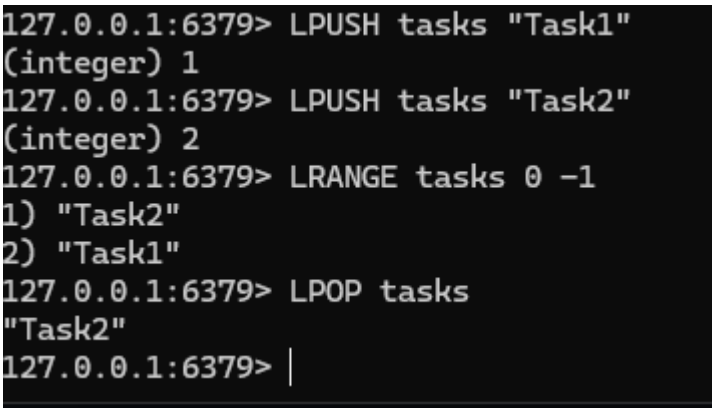
Операції з рядками



Створюємо ключ зі своїм ім’ям: SET student "<anton>".
Отримуємо значення командою GET student.
Виконуємо INCR mycounter (двічі) та перевіряємо його значення.

4.Структури даних

Список (List)



Створюємо чергу задач, додаючи елементи до початку списку (LPUSH) і витягуємо їх по одному (LPOP). Це імітує типову чергу завдань.

Множина (Set)

```
127.0.0.1:6379> SADD tech:set "Redis"
(integer) 1
127.0.0.1:6379> SADD tech:set "PostgreSQL"
(integer) 1
127.0.0.1:6379> SADD tech:set "MongoDB"
(integer) 1
127.0.0.1:6379> SMEMBERS tech:set
1) "PostgreSQL"
2) "Redis"
3) "MongoDB"
127.0.0.1:6379> SISMEMBER tech:set "Redis"
(integer) 1
127.0.0.1:6379>
```

Формуємо набір унікальних технологій (SADD), перевіряємо наявність конкретної (SISMEMBER) і отримуємо всі елементи без порядку (SMEMBERS). Це корисно для тегів або категорій.

Хеш (Hash)

```
127.0.0.1:6379> HSET profile name "anton"
(integer) 1
127.0.0.1:6379> HSET profile city "Kyiv"
(integer) 1
127.0.0.1:6379> HGET profile name
"anton"
127.0.0.1:6379> HGETALL profile
1) "name"
2) "anton"
3) "city"
4) "Kyiv"
127.0.0.1:6379>
```

Зберігаємо структуровані дані про профіль користувача у форматі ключ–значення (HSET, HGET, HGETALL). Аналог JSON-об'єкта або словника Python.

Відсортована множина (Sorted Set)

```
127.0.0.1:6379> ZADD scores 85 "Student1"
(integer) 1
127.0.0.1:6379> ZADD scores 92 "Student2"
(integer) 1
127.0.0.1:6379> ZADD scores 74 "Student3"
(integer) 1
127.0.0.1:6379> ZREVRANGE scores 0 -1 WITHSCORES
1) "Student2"
2) "92"
3) "Student1"
4) "85"
5) "Student3"
6) "74"
127.0.0.1:6379> ZRANK scores "Student1"
(integer) 1
127.0.0.1:6379>
```

Зберігаємо студентів з балами (ZADD), отримуємо їх за спаданням (ZREVRANGE) і знаходимо рейтинг окремого студента (ZRANK). Зручно для рейтингів та лідербордів.

5.Робота з TTL та тимчасовими ключами

```
127.0.0.1:6379> SET temp:data "Hello" EX 10
OK
127.0.0.1:6379> GET temp:data
"Hello"
127.0.0.1:6379> TTL temp:data
(integer) 5
127.0.0.1:6379>
```

Створюємо тимчасовий ключ (SET ... EX) зі строком життя 10 секунд, перевіряємо його значення (GET) і скільки часу залишилось до автоматичного видалення (TTL). Це імітує зберігання даних, що мають обмежений термін дії

Запитання для самоперевірки

1. Як встановити Redis у Windows/Linux?

На Linux Redis встановлюється командою sudo apt install redis, а для Windows можна скористатися WSL (Windows Subsystem for Linux) або docker.

2.Які основні команди для роботи з рядками в Redis?

SET створює ключ, GET читає значення, INCR автоматично збільшує числове значення ключа.

3.Як працювати з списками, множинами, хешами та Sorted Set?

- Списки: LPUSH, LRANGE, LPOP — як черга.
- Множини: SADD, SMEMBERS, SISMEMBER — зберігають унікальні значення.
- Хеші: HSET, HGET, HGETALL — як словники.
- Sorted Set: ZADD, ZREVRANGE, ZRANK — зберігають значення з рейтингами (балами).

4.Як встановити час життя ключа в Redis?

За допомогою параметра EX у команді SET можна задати TTL (час життя) для автоматичного видалення ключа.

5. Які можливі сценарії використання Redis у реальних застосунках?

Redis використовується для кешування, управління сесіями, створення черг, лідербордів, обробки даних у реальному часі та зберігання проміжних результатів.

Висновок

У цій лабораторній роботі ми ознайомилися з базовими типами даних Redis, командами для роботи з ними, TTL-ключами, а також реалізували просту інтеграцію з Python. Redis є потужним інструментом для створення високошвидкісних і масштабованих застосунків, особливо коли потрібне кешування або обробка даних у реальному часі.