 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Комп’ютерний практикум №1**

з дисципліни «Веб-технології та веб-дизайн»

Варіант №9

**Виконав:**

студент гр. БС-82

Кононов А. В.

**Перевірив:**

ас. каф. БМК

Давидько О.Б.

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ-2020

1) Виберіть одну з мов програмування (C#, Rust, Go, Python) для виконання практикумів, обґрунтуйте свій вибір.

Я хочу обрати мову програмування Python, тому що:

1. Це дуже зручна і практична мова програмування, з широким спектром застосування.
2. Я користувався фреймворком Django для написання бек-енду для сайту.
3. З дисципліни ‘Інтелектуальний аналіз даних’ ми використовуємо Python, тому для глибшого занурення та кращого вивчення мови з різних сторін, я хочу вивчати її і на цій дисципліні.
4. Маючи хороші знання з цієї мови – відкривається багато можливостей її застосування для того чим цікавлюсь я зараз і можливо буду в майбутньому.

2) Виберіть бібліотеку для роботи з мережею, обґрунтуйте свій вибір.

Я хочу обрати бібліотеку Django так як:

1. Я вже мав досвід роботи з нею, тому для мене буде легше згадати вивчене ніж вчити щось нове.
2. Мені подобається, що більшість інструментів для створення аппа є частиною цієї бібліотеки, а не поставляються у вигляді різних бібліотек.

3) Як в вибраній вами мові реалізується управління пам’яттю, та робота з змінюваними(Mutable) об’єктами/структурами.

В Python диспетчер пам'яті відповідає за такі завдання: періодично виконуючи очистку, виділення і управління пам'яттю.  Python управляє об'єктами за допомогою підрахунку посилань (reference count). Це означає, що диспетчер пам'яті відстежує кількість посилань на кожен об'єкт в програмі. Коли лічильник посилань об'єкта падає до нуля, що означає, що об'єкт більше не використовується, збирач сміття (частина диспетчера пам'яті) автоматично звільняє пам'ять від цього конкретного об'єкта.

Змінні об'єкти не постійні. Вони нагадують контейнер для зберігання: контейнер залишається на місці, а от вміст може сильно змінитися. Наприклад, якщо передати список в «чужу» функцію, то його можуть зіпсувати до невпізнання:

def list\_destroyer(l):

l[0] = 'ha'

l[1] = 'hi'

l[2] = 'ho'

a = [1, 2, 3]

print(a) # -> [1, 2, 3]

list\_destroyer(a)

print(a) # -> ['ha', 'hi', 'ho']

4) Як вибрана вами мова виконує паралельні обчислення, як дані обчислення синхронізуються.



Python пропонує два модуля для реалізації потоків в програмах:

* Модуль <thread>
* Модуль <threading>

Модуль <thread> є устарівшим в Python 3 и перейменований в модуль <\_thread>. Основна відмінність між цими двома модулями є те, що модуль <thread> реалізує багатопоточність як функцію. З іншого боку, модуль <threading> пропонує об'єктно-орієнтований підхід для забезпечення можливості створення потоків.

**Використання модуля Thread:**

thread.start\_new\_thread ( function, args[, kwargs] )

Цей метод запускає новий потік і повертає його ідентифікатор. Він викличе функцію, зазначену в якості параметра «function» з переданим списком аргументів. Коли повертається функція, потік мовчки завершує роботу.

Тут args - це кортеж аргументів; використовуйте порожній кортеж для виклику без будь-яких аргументів. Необов'язковий аргумент вказує словник аргументів з ключовими словами.

Якщо функція завершується з необробленим винятком, виводиться трасування стека, а потім потік виходить (це не впливає на інші потоки, вони продовжують працювати).

**Використання модуля Threading:**

Модуль threading надає багаті можливості і велику підтримку потоків, ніж застарілий модуль thread, описаний раніше.

Модуль об'єднує всі методи модуля thread і надає кілька додаткових методів:

* threading.activeCount (): знаходить загальне число активних об'єктів потоку.
* threading.currentThread (): його можна використовувати для визначення кількості об'єктів потоку в елементі управління потоком викликаючої сторони.
* threading.enumerate (): він надасть вам повний список об'єктів потоку, які в даний момент активні.

Модуль також представляє клас Thread, який ви можете реалізувати в потоках. Це об'єктно-орієнтована варіант багатопоточності в Python.

|  |  |
| --- | --- |
| Методи класу | Опис методу |
| run (): | це функція точки входу для будь-якого потоку. |
| start (): | запускає потік при виклику методу run. |
| join ([time]): | дозволяє програмі очікувати завершення потоків. |
| isAlive (): | перевіряє активний потік. |
| getName (): | витягує ім'я потоку. |
| setName (): | оновлює ім'я потоку. |

**Python Multithreading - синхронізація потоків**

Модуль має вбудовану функціональність для реалізації блокування, яка дозволяє синхронізувати потоки. Блокування необхідне для контролю доступу до загальних ресурсів для запобігання пошкодження або пропущених даних.

Ви можете викликати метод Lock (), щоб застосувати блокування, він повертає новий об'єкт блокування. Потім ви можете викликати метод захоплення (блокування) об'єкта блокування, щоб змусити потоки працювати синхронно.

Необов'язковий параметр блокування вказує, чи очікує потік отримання блокування.

* У разі, якщо блокування встановлена ​​на нуль, потік негайно повертається з нульовим значенням, якщо блокування не може бути отримане, і 1, якщо блокування отримане.
* У разі, якщо для блокування встановлено значення 1, потік блокується і очікує зняття блокування.

Метод release () об'єкта блокування використовується для зняття блокування, коли воно більше не потрібне.

5) Поясніть відмінність між синхронним та асинхронним виконанням програмного коду.

При синхронному виконанні завдання виконуються по черзі, одне після іншого. При асинхронному ж, операції виконуються незалежно одна від іншої. Наприклад, вам слід забронювати білет на літак і повідомити свого боса про вашу відсутність. При синхронному виконанні ви б спочатку набрали туристичне агенство, якщо ж вас поставили на утримування, ви просто чекаєте коли хтось не підійме слухавку. І тільки після бронювання білету ви починаєте писати лист своєму босу. Одна задача після іншої. При асинхронному виконанні ви б набрали агенство, поки ви на утриманні — почали б писати листа, коли хтось зняв би слухавку ви знову перемкнулися на розмову, а мейл дописали б після дзвінка. Ось це асинхронне виконання, одне завдання не блокує інше.

6) Чи може однопоточна програма бути асинхронною, та багатопоточна синхронною, поясніть чому.

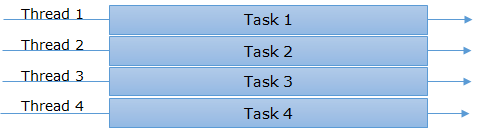
Однопоточні системи обробки подій практично завжди реалізуються за допомогою черги подій або повідомлень. Ідея полягає в тому, що, коли програма нічого не робить, її потік дивиться на наявність нових подій в черзі. Коли відбувається подія, воно додається в чергу і, якщо потік виконання в цей час не робить нічого іншого, він забирає наступну подію з черги і виконує відповідний обробник.

Таким чином, події додаються в чергу і UI-фреймворк представляє диспетчер, який виконується в тому ж потоці, що і обробники, викликаючи їх по мірі потреби. У будь-який момент часу потік знаходиться або в якомусь обробнику події, або в диспетчері, чекаючи наступної події. У операційної системи багато потоків і той код, який дійсно взаємодіє з користувачем, виконується окремо від нашої програми і лише посилає їй повідомлення.

Це приклад однопоточної асинхронної програми, так як ми не знаємо, в якому порядку буде виконуватися код. Обробники подій, з точки зору програми, можуть виконуватися довільно.

image

Багатопоточність - в цьому сценарії, ми використовували багато потоків, які можуть брати завдання і приступати до роботи з ними. У нас є пули потоків і безліч завдань. Отже, потік може працювати ось так:



Тут ми можемо бачити, що у нас є 4 потоку і стільки ж завдань для виконання, і кожен потік починає працювати з ними. Це ідеальний сценарій, але в звичайних умовах використовується більша кількість завдань ніж кількість доступних потоків, таким чином потік який звільнився отримує інше завдання. Створення нового потоку не відбувається кожного разу тому що для цього потрібні системні ресурси такі як процесор, пам'ять і має бути певна початкова кількість потоків.