**Форма № 17**

Міністерство освіти і науки України

**Національний університет «Запорізька політехніка»**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Кафедра програмних засобів\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(найменування кафедри)

**КУРСОВИЙ ПРОЄКТ**

**(РОБОТА)**

з дисципліни «Операційні системи»

(назва дисципліни)

на тему: «Застосунок аналізу системних ресурсів»

Студента 3 курсу КНТ-122 групи

спеціальності 121 Інженерія   
програмного забезпечення

освітня програма (спеціалізація) інженерія програмного забезпечення

\_\_\_\_\_\_\_\_\_\_\_\_Онишенко О. А.\_\_\_\_\_\_\_

(прізвище та ініціали)

(прізвище та ініціали)

Керівник доцент, к.т.н., Каплієнко Т. І.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_Оцінка: ECTS \_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   Степаненко О. О.

(підпис) (прізвище та ініціали)

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

(підпис) (прізвище та ініціали)

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

(підпис) (прізвище та ініціали

2024 рік

**Форма № 25**

Міністерство освіти і науки України

**Національний університет «Запорізька політехніка»**

( повне найменування закладу вищої освіти)

Інститут, факультет ІІРЕ, ФКНТ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра програмних засобів\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ступінь вищої освіти      бакалавр**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Спеціальність 121 Інженерія програмного забезпечення**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(код і найменування)

Освітня програма (спеціалізація) Інженерія програмного забезпечення**\_\_\_\_\_\_\_**

(назва освітньої програми (спеціалізації)

**ЗАТВЕРДЖУЮ**

**Завідувач кафедри ПЗ, д.т.н, проф.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_С.О. Субботін**

“\_\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_року

З А В Д А Н Н Я

**НА КУРСОВИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА**

  Онишенко О. А

(прізвище, ім’я, по батькові)

1. Тема проєкту (роботи) Застосунок аналізу системних ресурсів

керівник проєкту (роботи)\_Степаненко Олександр Олексійович ,

( прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від

2. Строк подання студентом проєкту (роботи)\_03 грудня 2024 року\_\_\_\_\_\_\_\_\_\_

3. Вихідні дані до проєкту (роботи) створити застосунок аналізу системних ресурсів

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області; 2. Аналіз програмних засобів; 3. Основні рішення з реалізації компонентів системи; 4. Керівництво програміста; 5. Керівництво користувача; 6. Додатки.  
5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

6. Консультанти розділів проєкту (роботи)

|  |  |  |  |
| --- | --- | --- | --- |
| Розділ | Прізвище, ініціали та посада  консультанта | Підпис, дата | |
| завдання видав | прийняв  виконане завдання |
| 1-5 Основна частина | Степаненко О. О. |  |  |
|  |  |  |  |
|  |  |  |  |

7. Дата видачі завдання\_\_16 жовтня 2024 р.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| №  з/п | Назва етапів курсового  проєкту (роботи) | Строк виконання етапів проєкту  ( роботи ) | Примітка |
| 1. | Аналіз індивідуального завдання. | 1 тиждень |  |
| 2. | Аналіз програмних засобів, що будуть використовуватись в роботі. | 2 тиждень |  |
| 3. | Аналіз структур даних, що необхідно використати в курсової роботі. | 3 тиждень |  |
| 4. | Затвердження завдання | 4 тиждень |  |
| 5. | Вивчення можливостей програмної реалізації структур даних та інтерфейсу користувача. | 5-9 тиждень |  |
| 6. | Аналіз вимог до апаратних засобів | 9 тиждень |  |
| 7. | Розробка програмного забезпечення | 9-13 тиждень |  |
| 8. | Проміжний контроль | 10 тиждень | Розділи 1-2 ПЗ |
| 9. | Оформлення, відповідних пунктів пояснювальної записки. | 10-14 тиждень | Розділи 1-5 ПЗ |
| 10. | Захист курсової роботи. | 15 тиждень |  |
|  |  |  |  |
|  |  |  |  |

**Студент \_\_\_\_\_\_\_\_\_ \_\_\_** Онищенко О. А.**\_\_\_\_**

( підпис ) (прізвище та ініціали)

**Керівник проєкту (роботи) \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_** Степаненко О. О.**\_\_\_\_**

( підпис ) (прізвище та ініціали)

реферат

Проєкт «Застосунок аналізу системних ресурсів» є комплексним застосунком, розробленим з використанням мови програмування Python. Застосунок призначений для полегшення процесу отримання актуальних та оновлювальних даних про поточні ресурси системи користувача у цифровому вигляді. Це зручна платформа, яка дозволяє користувачам переглядати інформацію про систему, її поточні характеристики та властивосі, а також показники роботи системи на конкретний момент часу. Застосунок також дозволяє отримати візуальне представлення зміни показників системи, що може полегшити сприйняття користувачем відповідних параметрів системи при аналізі системних ресурсів. Проєкт має на меті спростити процес витягання та взаємодії із ресурсами системи користувача за допомогою візуальних елементів інтерфейсу.

Система, ресурси системи, Python, курсова робота, курсовий проєкт, Windows застосунок, застосунок аналізу системних ресурсів.

зміст

[реферат 4](#_Toc184211812)

[зміст 5](#_Toc184211813)

[перелік умовних позначень, символів, одиниць, скорочень і термінів 7](#_Toc184211814)

[вступ 8](#_Toc184211815)

[1 Аналіз предметної області 9](#_Toc184211816)

[1.1 Аналіз витягання системних ресурсів, як основи предметної області 9](#_Toc184211817)

[1.2 Огляд існуючих методів вирішення завдання 9](#_Toc184211818)

[1.2.1 Передмова 9](#_Toc184211819)

[1.2.2 Система «Cpu-Z» 9](#_Toc184211820)

[1.2.2 Система «Open Hardware Monitor» 12](#_Toc184211821)

[1.3 Висновки з першого розділу та постановка задачі 15](#_Toc184211822)

[2 Аналіз програмних засобів 16](#_Toc184211823)

[2.1 Огляд особливостей мови програмування 16](#_Toc184211824)

[2.2 Огляд особливостей обраного компілятору 16](#_Toc184211825)

[2.3 Огляд класової ієрархії 16](#_Toc184211826)

[2.4 Висновки з розділу 16](#_Toc184211827)

[3 основні рішення з реалізації компонентів системи 17](#_Toc184211828)

[3.1 Основні рішення щодо розроблених класів 17](#_Toc184211829)

[3.1.1 Клас ….. 17](#_Toc184211830)

[3.1.2 Клас …. 18](#_Toc184211831)

[3.2 Основні розроблені алгоритми 18](#_Toc184211832)

[3.3 Основні рішення щодо розробки інтерфейсу 18](#_Toc184211833)

[3.4 Основні рішення щодо роботи з даними 18](#_Toc184211834)

[3.5 Обробка виключних ситуацій 18](#_Toc184211835)

[4 керівництво програміста 20](#_Toc184211836)

[4.1 Призначення та умови застосування програми 20](#_Toc184211837)

[4.1.1 Призначення програми 20](#_Toc184211838)

[4.1.2 Функції програми 20](#_Toc184211839)

[4.1.3 Умови застосування програми 20](#_Toc184211840)

[4.2 Характеристика програми 20](#_Toc184211841)

[4.2.1 Структура програми 21](#_Toc184211842)

[4.3 Звертання до програми 21](#_Toc184211843)

[4.4 Вхідні та вихідні дані 21](#_Toc184211844)

[4.4.1 Вхідні дані 21](#_Toc184211845)

[4.4.2 Вихідні дані 21](#_Toc184211846)

[4.5 Повідомлення 21](#_Toc184211847)

[5 керівництво користувача 22](#_Toc184211848)

[5.1 Призначення програми 22](#_Toc184211849)

[5.2 Умови виконання програми 22](#_Toc184211850)

[5.2.1 Апаратні вимоги програми 22](#_Toc184211851)

[5.2.2 Вимоги до користувача 22](#_Toc184211852)

[5.3 Виконання програми 22](#_Toc184211853)

[5.3.1 Запуск програми 22](#_Toc184211854)

[5.3.2 Виконання роботи з програмою 23](#_Toc184211855)

[5.4 Повідомлення користувачу 23](#_Toc184211856)

[5.5 Довідка програми 23](#_Toc184211857)

[Висновки 24](#_Toc184211858)

[ПЕРЕЛІК ПОСИЛАНЬ 25](#_Toc184211859)

[Додаток A – Код програми 26](#_Toc184211860)

[А1 – run.py 26](#_Toc184211861)

перелік умовних позначень, символів, одиниць, скорочень і термінів

ОС – операційна система,

Python – мова програмування

NiceGUI – бібліотека Python для розробки графічних інтерфейсів

psutil – бібліотека Python для доступу до ресурсів системи

pywin32 або win32api – бібліотека Python для доступу до ресурсів системи

вступ

Застосунок аналізу системних ресурсів є цифровою платформою надання коректних поточних даних про характеристики користувацької системи та поточних показників конкретних компонентів. Оскільки наявність комп’ютера є корисною можливістю, аналіз ресурсів системи для системних адміністраторів або користувачів, які дбають про систему самостійно може бути корисним інструментом. Такий застосунок може полегшити отримання та опрацювання інформації про поточні характеристики системи та її показники на конкретний момент часу.

1 Аналіз предметної області

Системні ресурси є частиною системи. Вони відображають характеристики системи, її конкретних компонентів, а також надають інформацію про поточні показники компонентів системи.

1.1 Аналіз витягання системних ресурсів, як основи предметної області

Витягання системних ресурсів є основною підзадачею аналізу ресурсів системи користувача. Для витягання ресурсів із системи розробники операційних систем зазвичай надають системні функції та методи для роботи з компонентами системи. Оскільки застосунок розробляється під платформу Windows, програма використовує системні функції та методи, зокрема WMI, для доступу до ресурсів системи, для витягання інформації про компоненти та аналіз їх поточних характеристик.

1.2 Огляд існуючих методів вирішення завдання

1.2.1 Передмова

Оскільки задача аналізу системних ресурсів є необхідною для користувачів які обслуговують системи власноруч, розробники з різних країн зробили подібні системи для витягання та аналізу ресурсів користувацьких систем.

1.2.2 Система «Cpu-Z»

CPU-Z є безкоштовною програмою для Windows, яка збирає інформацію про основні компоненти системи: процесор, материнська плата, пам'ять. CPU-Z повністю підтримується на Windows 11.

Переваги:

* Детальна та точна інформація.

Недоліки:

* Відсутність графічних елементів.

Робота програми наведена нижче.

Зображення, що містить текст, знімок екрана, програмне забезпечення, Комп’ютерна піктограма

Автоматично згенерований опис

Рисунок 1.1 – Робота програми «Cpu-Z»

Зображення, що містить текст, знімок екрана, число, Шрифт

Автоматично згенерований опис

Рисунок 1.2 – Робота програми «Cpu-Z»

Зображення, що містить текст, знімок екрана, монітор, число

Автоматично згенерований опис

Рисунок 1.3 – Робота програми «Cpu-Z»

1.2.2 Система «Open Hardware Monitor»

Open Hardware Monitor є безкоштовним програмним забезпеченням з відкритим вихідним кодом, яке відстежує температурні датчики, швидкість обертання вентиляторів, напругу, навантаження і тактову частоту комп'ютера.

Переваги:

* Детальна інформація;
* Дані чітко структуровано;
* Наявність графічних елементів.

Робота програми наведена нижче.

Зображення, що містить текст, знімок екрана, програмне забезпечення, число

Автоматично згенерований опис

Рисунок 1.4 – Робота програми «Open Hardware Monitor»

Зображення, що містить текст, знімок екрана, монітор, число

Автоматично згенерований опис

Рисунок 1.5 – Робота програми «Open Hardware Monitor»

Зображення, що містить текст, знімок екрана, монітор, програмне забезпечення

Автоматично згенерований опис

Рисунок 1.6 – Робота програми «Open Hardware Monitor»

1.3 Висновки з першого розділу та постановка задачі

Існуючі програмні засоби наводять детальну актуальну та точну інформацію про систему користувача, але не всі мають графічні елементи.

2 Аналіз програмних засобів

В даному розділі розглянуті основні особливості програмних засобів, за допомогою який реалізовано курсовий проєкт.

2.1 Огляд особливостей мови програмування

Python є об'єктно орієнтованою мовою програмування, в якій кожна змінна є об’єктом певного класу. Мова є динамічно типізованою і не вимагає явного визначення типів даних змінних. Ця мова також визначає блоки коду через відступи перед кожним рядком коду. За замовчуванням відступ мови становить 4 пробіли.

Мову було обрано через її простий синтаксис та доступ багатьох бібліотек та засобів для роботи.

2.2 Огляд використаних засобів

Бібліотека визначення системних ресурсів psutil має методи для наступних пунктів системи: CPU, Memory, Disks, Network, Sensors, Other system info.

Опис класів, що використовувались при розробці (стандартних або сторонніх бібліотек, не тих, що ви розробили, а ті, які ви використали)

2.4 Висновки з розділу

У даному розділі було описано мову програмування та її особливості. Також було описано ….

3 основні рішення з реалізації компонентів системи

Текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст.

3.1 Основні рішення щодо розроблених класів

Текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст.

3.1.1 Клас …..

Текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст.

Дані та методи класу наведені у таблиці 3.1.

Таблиця 3.1 – Опис полів та методів класу …..

|  |  |
| --- | --- |
| Поля та методи класу | Опис |
| **1** | **2** |
| ***public:*** | |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Продовження таблиці 3.1

|  |  |
| --- | --- |
| **1** | **2** |
|  |  |
|  |  |
|  |  |
| ***protected:*** | |
|  |  |
|  |  |
|  |  |

3.1.2 Клас ….

3.2 Основні розроблені алгоритми

Опис розроблених алгоритмів

3.3 Основні рішення щодо розробки інтерфейсу

Опис рішень, які використовувались для реалізації інтерфейсу

3.4 Основні рішення щодо роботи з даними

Текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст.

3.5 Обробка виключних ситуацій

Текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст.

4 керівництво програміста

Текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст.

4.1 Призначення та умови застосування програми

4.1.1 Призначення програми

Текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст.

4.1.2 Функції програми

Функції, що виконує програма:

* …..

4.1.3 Умови застосування програми

Текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст.

4.2 Характеристика програми

Текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст.

4.2.1 Структура програми

Текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст.

4.3 Звертання до програми

Текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст.

4.4 Вхідні та вихідні дані

4.4.1 Вхідні дані

Текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст.

4.4.2 Вихідні дані

Текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст.

4.5 Повідомлення

* ….

5 керівництво користувача

В даному розділі розглянуто призначення програми, умови її виконання, процес виконання програми та повідомлення для користувача.

5.1 Призначення програми

Текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст.

5.2 Умови виконання програми

5.2.1 Апаратні вимоги програми

Текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст.

5.2.2 Вимоги до користувача

Текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст.

5.3 Виконання програми

5.3.1 Запуск програми

Текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст.

5.3.2 Виконання роботи з програмою

Текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст.

5.4 Повідомлення користувачу

….

5.5 Довідка програми

Текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст, текст.

Висновки

Під час виконання курсового проєкту благодаттю Господа нашого Ісуса Христа було розроблено проєкт аналізу системних ресурсів мовою програмування Python із використанням зовнішніх пакетів NiceGUI, psutil, та win32api. Програма має візуальний графічний інтерфейс, оновлює дані щосекунди, має графічні елементи відображення змін параметрів системи та детальний опис інформації системи користувача.

ПЕРЕЛІК ПОСИЛАНЬ

1) How to Get Hardware and System Information in Python [Електронний ресурс]. – Режим доступу: <https://thepythoncode.com/article/get-hardware-system-information-python>

2) psutil documentation [Електронний ресурс]. – Режим доступу: <https://psutil.readthedocs.io/en/latest/?badge=latest#>

3) NiceGUI Documentation [Електронний ресурс]. – Режим доступу: <https://nicegui.io/documentation>

4) How to Make a Network Usage Monitor in Python [Електронний ресурс]. – Режим доступу: <https://thepythoncode.com/article/make-a-network-usage-monitor-in-python>

5) Is it possible to get in Python the CPU/core/processor id that the python program itself is using? - Quora [Електронний ресурс]. – Режим доступу: <https://www.quora.com/Is-it-possible-to-get-in-Python-the-CPU-core-processor-id-that-the-python-program-itself-is-using>

6) Python get cpu info [Електронний ресурс]. – Режим доступу: <https://www.programcreek.com/python/?CodeExample=get%20cpu%20info>

7) How to use the psutil.net\_if\_addrs function in psutil | Snyk [Електронний ресурс]. – Режим доступу: <https://snyk.io/advisor/python/psutil/functions/psutil.net_if_addrs>

Додаток A – Код програми

А1 – run.py

from collections import defaultdict

from datetime import datetime

from nicegui import ui

import win32api

import platform

import psutil

HEADING\_CLASSES='font-bold text-xl'

COLUMNS=[

{'name':'property','label':'Property','field':'property','align':'left'},

{'name':'value','label':'Value','field':'value','sortable':True},

]

uname=platform.uname()

boot\_time\_stamp=psutil.boot\_time()

boot\_time=datetime.fromtimestamp(boot\_time\_stamp)

cpu\_frequency=psutil.cpu\_freq()

system\_virtual\_memory=psutil.virtual\_memory()

swap=psutil.swap\_memory()

partitions=psutil.disk\_partitions()

disk\_io=psutil.disk\_io\_counters()

if\_addrs=psutil.net\_if\_addrs()

net\_io=psutil.net\_io\_counters()

sent,recv=net\_io.bytes\_sent,net\_io.bytes\_recv

def get\_rows(data:dict):

return [{'property':k.capitalize(),'value':v} for k,v in data.items()]

def update\_ui():

processor\_frequencies\_data={

'min':cpu\_frequency.min,

'max':cpu\_frequency.max,

'current':cpu\_frequency.current,

}

processor\_frequencies\_table.rows=get\_rows(processor\_frequencies\_data)

cpu\_usage=psutil.cpu\_percent(percpu=True)

processor\_usage\_data={

f'Core {i}': usage

for i,usage in enumerate(cpu\_usage)

}

processor\_usage\_table.rows=get\_rows(processor\_usage\_data)

cpu\_usage=psutil.cpu\_percent()

this\_time=datetime.now().timestamp()

processor\_usage\_plot.push([this\_time],[[cpu\_usage]])

system\_virtual\_memory=psutil.virtual\_memory()

virtual\_memory\_data={

'total':get\_formatted\_size(system\_virtual\_memory.total),

'available':get\_formatted\_size(system\_virtual\_memory.available),

'used':get\_formatted\_size(system\_virtual\_memory.used),

'percentage':f'{system\_virtual\_memory.percent}%',

}

virtual\_memory\_table.rows=get\_rows(virtual\_memory\_data)

virtual\_memory\_circle.value=system\_virtual\_memory.percent

memory\_usage\_plot.push([this\_time],[[system\_virtual\_memory.percent]])

swap=psutil.swap\_memory()

swap\_memory\_data={

'total':get\_formatted\_size(swap.total),

'free':get\_formatted\_size(swap.free),

'used':get\_formatted\_size(swap.used),

'percentage':f'{swap.percent}%',

}

swap\_memory\_table.rows=get\_rows(swap\_memory\_data)

swap\_memory\_circle.value=swap.percent

global sent,recv

new\_network\_io=psutil.net\_io\_counters()

us,ds=new\_network\_io.bytes\_sent-sent,new\_network\_io.bytes\_recv-recv

network\_speed\_plot.push([this\_time],[[ds/1],[us/1]])

sent,recv=new\_network\_io.bytes\_sent,new\_network\_io.bytes\_recv

def get\_formatted\_size(bytes,suffix='B'):

factor=1024

for unit in ['','K','M','G','T','P']:

if bytes<factor: return f'{bytes:.2f} {unit}{suffix}'

bytes/=factor

with ui.row().classes('flex gap-3'):

with ui.column():

system\_data={

'type':uname.system,

'user':uname.node,

'release':uname.release,

'version':uname.version,

'machine':uname.machine,

'booted':f'{boot\_time.day}.{boot\_time.month}.{boot\_time.year} {boot\_time.hour:02d}:{boot\_time.minute:02d}:{boot\_time.second:02d}'

}

system\_table=ui.table(columns=COLUMNS,rows=get\_rows(system\_data),row\_key='name',title='System')

with ui.column():

processor\_data={

'name':uname.processor,

'platform':uname.machine,

'cores':psutil.cpu\_count(logical=False),

'threads':psutil.cpu\_count(logical=True),

}

processor\_table=ui.table(columns=COLUMNS,rows=get\_rows(processor\_data),row\_key='name',title='Processor')

with ui.row().classes('flex w-full'):

with ui.column():

processor\_frequencies\_data={

'min':cpu\_frequency.min,

'max':cpu\_frequency.max,

'current':cpu\_frequency.current,

}

processor\_frequencies\_table=ui.table(columns=COLUMNS,rows=get\_rows(processor\_frequencies\_data),row\_key='name',title='Frequencies (MHz)')

ui.label('Processor Frequency')

processor\_frequencies\_circle=ui.circular\_progress(min=cpu\_frequency.min,max=cpu\_frequency.max,value=cpu\_frequency.current)

processor\_usage\_data={

f'Core {i}': usage

for i,usage in enumerate(psutil.cpu\_percent(percpu=True))

}

processor\_usage\_table=ui.table(columns=COLUMNS,rows=get\_rows(processor\_usage\_data),row\_key='name',title='Usage (%)').classes('flex-1')

with ui.column():

virtual\_memory\_data={

'total':get\_formatted\_size(system\_virtual\_memory.total),

'available':get\_formatted\_size(system\_virtual\_memory.available),

'used':get\_formatted\_size(system\_virtual\_memory.used),

'percentage':f'{system\_virtual\_memory.percent}%',

}

virtual\_memory\_table=ui.table(columns=COLUMNS,rows=get\_rows(virtual\_memory\_data),row\_key='name',title='Virtual Memory').classes('w-full')

ui.label('Virtual Memory Usage')

virtual\_memory\_circle=ui.circular\_progress(value=system\_virtual\_memory.percent,max=100)

swap\_memory\_data={

'total':get\_formatted\_size(swap.total),

'free':get\_formatted\_size(swap.free),

'used':get\_formatted\_size(swap.used),

'percentage':f'{swap.percent}%',

}

swap\_memory\_table=ui.table(columns=COLUMNS,rows=get\_rows(swap\_memory\_data),row\_key='name',title='Swap Memory')

ui.label('Swap Memory Usage')

swap\_memory\_circle=ui.circular\_progress(value=swap.percent,max=100)

with ui.column():

with ui.card():

ui.label('Disks').classes('q-table\_\_title')

disk\_tables=defaultdict(dict)

for partition in partitions:

try: usage\_data=psutil.disk\_usage(partition.mountpoint)

except: continue

partition\_name=win32api.GetVolumeInformation(partition.device)[0]

disk\_data={

'device':partition.device,

'name':partition\_name,

'file system':partition.fstype,

}

space\_data={

'total':get\_formatted\_size(usage\_data.total),

'used':get\_formatted\_size(usage\_data.used),

'free':get\_formatted\_size(usage\_data.free),

'percentage':f'{usage\_data.percent}%',

}

with ui.expansion(partition\_name):

disk\_tables[partition\_name]['disk']=ui.table(columns=COLUMNS,rows=get\_rows(disk\_data),row\_key='name',title=f'{partition\_name} Data')

disk\_tables[partition\_name]['space']=ui.table(columns=COLUMNS,rows=get\_rows(space\_data),row\_key='name',title=f'{partition\_name} Space')

ui.label(f'{partition\_name} Usage')

ui.circular\_progress(value=usage\_data.percent,max=100,min=0)

disks\_data={

'read':get\_formatted\_size(disk\_io.read\_bytes),

'write':get\_formatted\_size(disk\_io.write\_bytes),

}

disks\_table=ui.table(columns=COLUMNS,rows=get\_rows(disks\_data),row\_key='name').classes('w-full')

with ui.column():

with ui.card():

ui.label('Network').classes('q-table\_\_title')

network\_tables=defaultdict(str)

for interface\_name,interface\_addresses in if\_addrs.items():

interface\_addresses=[a for a in interface\_addresses if a.family.name=='AF\_INET' or a.family.name=='AF\_PACKET']

for address in interface\_addresses:

network\_data={

'IP Address' if address.family.name=='AF\_INET' else 'MAC Address':address.address,

'netmask':address.netmask,

'Broadcast IP' if address.family.name=='AF\_INET' else 'Broadcast MAC':address.broadcast,

}

with ui.expansion(interface\_name):

network\_tables[interface\_name]=ui.table(columns=COLUMNS,rows=get\_rows(network\_data),row\_key='name',title=f'{interface\_name} Data')

network\_data={

'sent':get\_formatted\_size(net\_io.bytes\_sent),

'received':get\_formatted\_size(net\_io.bytes\_recv),

}

network\_table=ui.table(columns=COLUMNS,rows=get\_rows(network\_data),row\_key='name').classes('w-full')

with ui.row():

processor\_usage\_plot=ui.line\_plot(n=1,figsize=(4.7,2.47)).with\_legend(['CPU Usage %'],loc='upper center',ncol=1)

processor\_usage\_plot.push([datetime.now().timestamp()],[[0]])

processor\_usage\_plot.push([datetime.now().timestamp()],[[100]])

memory\_usage\_plot=ui.line\_plot(n=1,figsize=(4.7,2.47)).with\_legend(['RAM Usage %'],loc='upper center',ncol=1)

memory\_usage\_plot.push([datetime.now().timestamp()],[[0]])

memory\_usage\_plot.push([datetime.now().timestamp()],[[100]])

network\_speed\_plot=ui.line\_plot(n=2,figsize=(4.7,2.47)).with\_legend(['Download Speed','Upload Speed'],loc='upper center',ncol=2)

network\_speed\_plot.push([datetime.now().timestamp()],[[0],[0]])

network\_speed\_plot.push([datetime.now().timestamp()],[[100],[100]])

ui.timer(1,update\_ui,active=True)

ui.run(title='System Resources Analysis',favicon='💻')