

ПРОГРАМНИЙ ЗАСІБ МОНІТОРИНГУ СИСТЕМНИХ РЕСУРСІВ ТА АВТОМАТИЗОВАНОЇ ОЧИСТКИ ФАЙЛОВОЇ СИСТЕМИ

Вінницькій національний технічний університет

Анотація

У роботі досліджено методи моніторингу системних ресурсів та методів автоматизованої очистки файлової системи й оперативної пам'яті. Розроблено десктопний застосунок SysMonitor, що забезпечує збір телеметрії у реальному часі та автоматичне реагування на критичні навантаження. Обґрунтовано вибір архітектури та технологічного стеку, проведено оцінку ефективності роботи програми.

Ключові слова: моніторинг ресурсів, автоматизована очистка, Python, CustomTkinter, телеметрія, системна оптимізація.

Abstract

The paper investigates approaches to system resource monitoring and methods for automated file system and RAM cleaning. The desktop application SysMonitor was developed to provide real-time telemetry collection and automatic response to critical loads. The architecture and technology stack choice are justified, and the performance of the system is evaluated.

Keywords: system monitoring, automated cleaning, Python, CustomTkinter, telemetry, system optimization.

Вступ

У сучасних умовах постійного зростання вимог програмного забезпечення до апаратних потужностей актуальною є проблема контролю стану ресурсів. Накопичення тимчасових файлів та фрагментація даних у пам'яті призводять до зниження швидкодії операційної системи.

Метою роботи є розробка десктопного програмного застосунку для моніторингу апаратних показників і автоматизованого очищення файлової системи.

Для досягнення мети визначено такі завдання:

- проаналізувати існуючі засоби моніторингу системних ресурсів;
- розробити архітектуру та алгоритми збору системної телеметрії;
- реалізувати модуль безпечного очищення тимчасових файлів та оптимізації пам'яті;
- оцінити практичну ефективність застосунку.

Основна частина

Запропонований програмний засіб базується на методі комплексного моніторингу, який поєднує відстеження базових метрик (CPU, RAM) із деталізованою картою навантаження логічних ядер процесора та мережевою статистикою. Такий метод дозволяє виявляти приховані фонові процеси та ідентифікувати аномальну активність конкретних потоків обробки даних.

На першому етапі роботи системи здійснюється безперервний збір показників продуктивності за допомогою бібліотеки psutil. Отримані метрики порівнюються з пороговими значеннями, і при перевищенні критичної межі (завантаження CPU понад 90% або RAM понад 85% протягом 15 секунд) система генерує тригер критичного стану.

Другий етап передбачає використання алгоритму розумного автоочищення (Smart Auto-Purge). Він інтегрує системні виклики Windows API (EmptyWorkingSet через бібліотеку ctypes) для вивільнення робочих наборів пам'яті, активацію збирача сміття (gc) та очищення DNS-кешу. Перевагою такого методу є примусове переміщення неактивних сторінок пам'яті у файл підкачки без необхідності закриття робочих програм користувача.

Архітектура системи реалізована мовою Python з використанням бібліотеки CustomTkinter за модульним принципом, що забезпечує мінімальне споживання ресурсів самою програмою.

З метою оцінювання ефективності запропонованого методу було проведено експериментальне тестування, у якому порівнювалися показники системи до та після застосування модулів оптимізації. Отримані результати наведено в таблиці 1.

Таблиця 1 – Порівняння показників ефективності використання системних ресурсів

Показник	До оптимізації	Після оптимізації
Споживання оперативної пам'яті, %	84	74,8
Вільний дисковий простір (дослідний розділ), ГБ	32,1	32,2
Статус системи при тривалому навантаженні (>15с)	CRITICAL	OPTIMAL

Отримані результати демонструють, що використання запропонованого методу дозволяє оперативно звільнити близько 10% зарезервованої пам'яті при високих навантаженнях. Алгоритм безпечного очищення диска забезпечує швидке видалення залишкових даних (понад 100 МБ) без негативного впливу на поточні процеси ОС.

Висновки

У результаті проведеного дослідження розроблено програмний засіб SysMonitor, що поєднує глибоку телеметрію та інструменти превентивної оптимізації у єдиному легкому середовищі. Експериментальні результати підтвердили доцільність використання інтегрованого циклу превентивного обслуговування для підтримки швидкодії системи. Встановлено, що розроблений метод забезпечує точне виявлення аномальної активності та ефективно вивільняє системні ресурси.

Реалізація системи з використанням низькорівневих системних викликів Windows API та адаптивного графічного інтерфейсу дозволяє значно автоматизувати рутинні процеси системного адміністрування.

Таким чином, програмний застосунок є ефективним та актуальним інструментом для підвищення стабільності роботи комп'ютерних систем у середовищі Windows.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Python Official Documentation — Python.org. [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.python.org/3>

2. Windows Internals: Monitoring System Resources — Microsoft Docs. [Електронний ресурс] – Режим доступу до ресурсу: <https://learn.microsoft.com/windows/monitoring>

Кобенда Денис Сергійович – студент групи ЗПІ-22б, Факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: kobendadenis@gmail.com

Коваленко Олена Олексіївна – к.т.н., доцент кафедри програмного забезпечення, Вінницький національний технічний університет, e-mail: ok@vntu.edu.ua

Науковий керівник: **Коваленко Олена Олексіївна** – к.т.н., доцент кафедри програмного забезпечення, Вінницький національний технічний університет, e-mail: ok@vntu.edu.ua

Kobenda Denys Serhiyovich – student of the group ЗПІ-22b, Faculty of Information Technology and Computer Engineering, Vinnytsia National Technical University, Vinnytsia.

Kovalenko Olena Oleksiivna – Candidate of Technical Sciences, Associate Professor of the Department of Software, Vinnytsia National Technical University, Vinnytsia.

Supervisor: ***Kovalenko Olena Oleksiivna*** – Candidate of Technical Sciences, Associate Professor of the Department of Software, Vinnytsia National Technical University, Vinnytsia.