Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____Комп'ютерних наук_____
(повна назва)

Кафедра _____Інформаційних управляючих систем_____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____другий (магістерський)_____

_Дослідження методів та технологій автоматизації вибору інструментальних__
_____засобів для адміністрування мережевої інфраструктури_____
(тема)

Виконав:
студент _2_ курсу, групи _ІУСТм-22-1_____
___Дмитро СОРОКІН_____
(власне ім'я, ПРІЗВИЩЕ)

Спеціальність ____122 Комп'ютерні_____
_____науки_____
(код і повна назва спеціальності)
Тип програми ____освітньо-професійна_____
(освітньо-професійна або освітньо-наукова)
Освітня програма _Інформаційні управляючі системи та технології_____
(повна назва освітньої програми)

Керівник _доцент каф. ІУС, Олег КОБИЛІН__
(посада, власне ім'я, ПРІЗВИЩЕ)

Допускається до захисту

Зав. кафедри _____ ___Костянтин ПЕТРОВ__
(підпис) (власне ім'я, ПРІЗВИЩЕ)

2024 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Інформаційних управляючих систем

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційні управляючі системи та технології
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
« 20 » листопада 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Сорокіну Дмитру Олексійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів та технологій автоматизації вибору інструментальних засобів для адміністрування мережевої інфраструктури

затверджена наказом університету від 16 листопада 2023 р. № 1359Ст

2. Термін подання студентом роботи до екзаменаційної комісії 14 01 2024 р.

3. Вихідні дані до роботи матеріали звіту з передатестаційної практики; доступ до всіх елементів мережевої інфраструктури, значень конфігураційних параметрів її вузлів; доступ до динамічних та статичних характеристик елементів мережевої інфраструктури.

4. Перелік питань, що потрібно опрацювати в роботі виконати аналіз проблеми адміністрування мережевої інфраструктури, проаналізувати існуючі методи адміністрування мережевої інфраструктури, розглянути основні технології адміністрування мережевої інфраструктури, визначити показники та критерії забезпечення функціональності мережевої інфраструктури, розробити інструментарій адміністратора для автоматизації вибору інструментальних засобів для адміністрування мережевої інфраструктури

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання для кваліфікаційної роботи	20.11.2023	Виконано
2	Аналіз предметної області для кваліфікаційної роботи	21.11.2023 – 03.12.2023	Виконано
3	Формування плану роботи над кваліфікаційною роботою	04.12.2023 – 09.12.2023	Виконано
4	Робота над теоретичними відомостями для кваліфікаційної роботи	10.12.2023 – 15.12.2023	Виконано
5	Аналіз існуючих методів адміністрування мережевої інфраструктури	16.12.2023 – 20.12.2023	Виконано
6	Аналіз технологій опису об’єктів контроля стану	21.12.2023 – 25.12.2023	Виконано
7	Формування методів автоматизації адміністрування мережевої інфраструктури для підтримки інформаційних сервісів	26.12.2023 – 31.12.2023	Виконано
8	Оформлення звіту та висновків	01.01.2024 – 08.01.2024	Виконано
9	Захист кваліфікаційної роботи	16.01.2024	Виконано

Дата видачі завдання __20___ ____листопада___2023 р.

Студент _____
(підпис)

Керівник роботи _____ __доцент каф. ІУС, Олег КОБИЛІН_
(підпис) (посада, власне ім’я, прізвище)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи магістра: 69 стор.,15 рис., 4 табл., 18 джерел, 1 додаток.

АВТОМАТИЗАЦІЯ, АДМІНІСТРУВАННЯ, ІНСТРУМЕНТАЛЬНІ ЗАСОБИ, МЕРЕЖЕВА ІНФРАСТРУКТУРА, СПЕЦІАЛІЗОВАНА БІБЛІОТЕКА.

Об’єкт дослідження – інструментальні засоби для адміністрування мережевої інфраструктури.

Предмет дослідження – дослідження методів та технологій автоматизації вибору інструментальних засобів для адміністрування мережевої інфраструктури.

Мета дослідження – пошук та розробка шляхів підвищення ефективності реалізації процесу вибору інструментальних засобів для мережевої інфраструктури.

ABSTRACT

Explanatory note to the master's thesis: 69 pp, 15 figures, 4 tables, 18 sources, 1 appendix.

ADMINISTRATION, AUTOMATION, NETWORK INFRASTRUCTURE, SPECIALIZED LIBRARY, TOOLS.

The object of the research is instrumental tools for network infrastructure administration.

The subject of research are the methods of dynamic adaptation of recommendations in e-commerce IT projects.

The subject of research are the methods and technologies for automating the selection of tools for network infrastructure administration.

The purpose of the research is to find and develop ways to increase the effectiveness of the process of selecting tools for the network infrastructure.

ЗМІСТ

Скорочення та умовні позначки 7
Вступ 8
1 Аналіз проблем автоматизації адміністрування мережевої інфраструктури для підтримки інформаційних сервісів 10
1.1 Аналіз проблеми адміністрування мережевої інфраструктури 10
1.2 Задачі адміністрування мережевої інфраструктури 12
1.3 Аналіз існуючих методів адміністрування мережевої інфраструктури 14
1.4 Аналіз технологій опису об’єктів контроля стану 18
1.5 Автоматизації контроля стану мережевої інфраструктури 21
1.6 Формування та використання бібліотек скриптів 22
1.7 Постановка задачі дослідження 27
2 Розробка методів автоматизації вибору інструментальних засобів для адміністрування мережевої інфраструктури 28
2.1 Класифікація скриптів, технологій та бібліотек 28
2.2 Формування схем та структур бібліотек 28
2.3 Формування методу вибору програм скриптів із бібліотеки та методу її наповнення 34
2.4 Побудова структури бібліотеки скриптів 36
3 Приклади використання методів та технологій автоматизації вибору інструментальних засобів для адміністрування мережевої інфраструктури із бібліотеки 40
Висновки 44
Перелік джерел посилання 45
Додаток А Графічний матеріал 47

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД – база даних

ПЗ – програмне забезпечення

СУБД – система управління базами даних

ЦП – центральний процесор

API – application programming interface

DDoS – distributed denial-of-service

CIM – common information model

CLI – command line interface

VPN – virtual private network

WMI – windows management instrumentation

ВСТУП

У світі, що стрімко рухається до цифрового майбутнього, мережева інфраструктура відіграє ключову роль у забезпеченні безперервного функціонування різноманітних інформаційних сервісів. Зростання обсягів даних, розширення мереж та змінюючийся характер технологій ставлять перед організаціями виклики в ефективному адмініструванні мережевої інфраструктури.

Автоматизація адміністрування мереж є концепцією, що визначає використання програмних та апаратних засобів для здійснення автоматичних операцій та управління мережевою інфраструктурою без прямого втручання користувача. Цей підхід спрямований на оптимізацію та поліпшення ефективності процесів адміністрування, зменшення ймовірності помилок, а також прискорення реакції на зміни в мережі.

Автоматизація адміністрування мереж може охоплювати широке коло задач - від початкової конфігурації пристроїв до моніторингу продуктивності та оперативного реагування на інциденти. Зокрема, вона дозволяє автоматизувати рутинні операції, такі як розгортання нових пристроїв, оновлення програмного забезпечення, збір статистики тощо.

Основними перевагами впровадження автоматизації є підвищення швидкості виконання задач, зменшення ймовірності людських помилок, можливість швидше реагувати на інциденти, а також оптимізація використання ресурсів мережі. Крім того, автоматизовані системи здатні обробляти великі обсяги даних набагато ефективніше за людину.

Разом з тим, для успішного впровадження автоматизації потрібне комплексне бачення щодо архітектури системи, вибору платформ та інструментів автоматизації, а також навчання персоналу. Необхідно чітко розділити сфери відповідальності між людиною та автоматизованими

системами. Крім того, автоматизація вимагає постійного моніторингу та вдосконалення на основі зворотного зв'язку.

Отже, незважаючи на певні виклики, автоматизація адміністрування мереж є важливим напрямком підвищення ефективності IT-інфраструктури сучасних організацій. Грамотне запровадження дозволить оптимізувати процеси управління, підвищити надійність мережі та швидкість реагування на зміни. Кваліфікаційна робота присвячена дослідженню методів та технологій автоматизації вибору інструментальних засобів для адміністрування мережевої інфраструктури.

Метою роботи є пошук та розробка шляхів підвищення ефективності реалізації процесу вибору інструментальних засобів для адміністрування мережевої інфраструктури.

Користувачем інструментальних засобів є адміністратор мережевої інфрастуктури.

У рамках кваліфікаційної роботи виконано опис об'єкта дослідження – інструментальних засобів для адміністрування мережевої інфраструктури та виконано огляд сучасних інструментальних засобів, класифікація скриптів, технологій та бібліотек, формування технологій автоматизації адміністрування мережевої інфраструктури.

Отримані результати виконання кваліфікаційної роботи оформлені у вигляді пояснювальної записки, перший розділ якої містить: аналіз проблем автоматизації мережевої інфраструктури для підтримки інформаційних сервісів, задачі адміністрування мережевої інфраструктури, аналіз автоматизації контролю стану мережевої інфраструктури, особливості використання скриптів та технологій, формування та використання бібліотек скриптів.

1 АНАЛІЗ ПРОБЛЕМ АВТОМАТИЗАЦІЇ АДМІНІСТРУВАННЯ МЕРЕЖЕВОЇ ІНФРАСТРУКТУРИ ДЛЯ ПІДТРИМКИ ІНФОРМАЦІЙНИХ СЕРВІСІВ

1.1 Аналіз проблеми адміністрування мережевої інфраструктури

Автоматизація адміністрування мереж може включати в себе різноманітні завдання, такі як конфігурація пристроїв, моніторинг стану мережі, управління політиками безпеки, резервне копіювання та відновлення даних, а також виявлення та виправлення неполадок [1]. Зазвичай, цей процес використовує різні інструменти, скрипти, та програмні рішення для автоматизації рутинних завдань, що дозволяє адміністраторам мереж зосередитися на стратегічних аспектах управління мережею та інфраструктурою.

Основною метою автоматизації адміністрування мереж є підвищення продуктивності, забезпечення стабільності та надійності роботи мережі, а також зменшення часу, необхідного для виконання рутинних операцій. Такий підхід дозволяє адаптувати мережеву інфраструктуру до зростаючих потреб бізнесу, швидко реагувати на зміни та забезпечувати ефективне використання ресурсів.

Ефективне адміністрування мережевої інфраструктури є ключовим фактором для забезпечення стабільної роботи сучасних інформаційних систем та сервісів, проте на практиці це складне завдання, що пов’язане з рядом проблем.

По-перше, сучасні корпоративні мережі характеризуються великою кількістю різноманітних мережевих пристроїв, що ускладнює їх централізоване управління. При цьому більшість мереж є гетерогенними, тобто містять обладнання від різних вендорів.

По-друге, значна частина задач з адміністрування мережі потребує ручного виконання, що є дуже трудомістким. Багато часу IT-персоналом

витрачається на рутинні операції на кшталт конфігурування пристроїв, моніторингу, оновлення ПЗ.

По-третє, зростає складність забезпечення безпеки мережі. Адміністратори не встигають оперативно реагувати на нові загрози.

По-четверте, складно забезпечити відповідність мережі змінюваним бізнес-вимогам. Час, необхідний для внесення змін вручну, може сягати декількох тижнів.

Тобто основні проблеми полягають у наступному:

– складність підтримки в актуальному стані конфігурацій мережевих пристроїв різних типів та виробників;

– трудомісткість виявлення та усунення несправностей в мережі;

– складність оперативного реагування на інциденти та зміни вимог до мережі;

– необхідність постійного моніторингу працездатності та продуктивності всіх компонентів мережі;

– потреба в регулярному оновленні версій ПЗ та прошивок мережевих пристроїв;

– забезпечення неперервності роботи мережі та інформаційних сервісів;

– дотримання політик інформаційної безпеки при управлінні доступом та розмежуванні прав;

– необхідність планування та управління пропускною спроможністю мережі;

– оптимізація витрат на утримання та розвиток інфраструктури.

Отже, адміністрування мережевої інфраструктури є комплексною проблемою. Для її вирішення необхідно шукати шляхи автоматизації рутинних процесів.

1.2 Задачі адміністрування мережевої інфраструктури

Адміністрування мережевої інфраструктури передбачає виконання комплексу завдань, пов'язаних з усіма етапами життєвого циклу мережі – від проектування до технічної підтримки.

На етапі проектування мережі адміністратор повинен обрати оптимальну топологію, необхідні мережеві компоненти, операційні системи і програмне забезпечення виходячи з вимог до мережі. Після розгортання обладнання настає етап базової конфігурації пристроїв та налаштування всіх необхідних параметрів.

Одним з найважливіших завдань є забезпечення безперервної працездатності всіх сервісів та додатків в мережі. Для цього адміністратор повинен постійно відслідковувати стан компонентів за допомогою моніторингу, оперативно усувати неполадки.

Крім того, регулярно мають виконуватися оновлення ПЗ на мережевому обладнанні для виправлення вразливостей та отримання нового функціоналу. Це потребує ретельного планування та тестування, адже оновлення можуть призвести до збоїв. Адміністратор повинен вчасно реагувати на нові загрози та налаштовувати заходи безпеки (фільтрацію трафіку, VPN тощо).

Основними задачами, які вирішуються в процесі адміністрування мережевої інфраструктури, є[2]:

– первинне проектування та розгортання мережі. Вибір топології, обладнання, програмного забезпечення;

– налаштування параметрів і конфігурацій мережевих пристроїв;

– забезпечення безперебійної роботи всіх сервісів та додатків в мережі;

– моніторинг поточного стану всіх компонентів мережі;

– оперативне реагування на неполадки, збої та відмови;

– аналіз продуктивності мережі, виявлення "вузьких місць";

– регулярне оновлення версій ПЗ на мережевих пристроях;

– управління безпекою - налаштування міжмережевих екранів, VPN, захист від DDoS тощо;

– контроль доступу до мережевих ресурсів на основі ролей та політик;

– планування розвитку інфраструктури з урахуванням майбутніх потреб;

– документування архітектури та конфігурацій мережі.

На рисунку 1.1 представлена мережева інфраструктура, в якій адміністратор вирішує основні завдання з моніторингу та конфігурування обладнання, і забезпечує в ній безпеку та продуктивність її сервісів.

Рисунок 1.1 – Адміністрування мережевої інфраструктури

Мережева інфраструктура – це комплекс технічних засобів і програмного забезпечення, необхідних для функціонування комп'ютерної мережі. Вона включає в себе сервери, комутатори, маршрутизатори, міжмережеві екрани, кабельні системи, системи безперебійного живлення тощо.

З рисунку 1.1 бачимо, що багато завдань адміністратора мереж можна частково або повністю автоматизувати за допомогою скриптів, а також

отримати інформацію про стан інфраструктури. Тобто будемо використовувати скрипти, як інструменти адміністрування мережевої інфраструктури, але перед цим проаналізуємо методи адміністрування мережевої інфраструктури.

1.3 Аналіз існуючих методів адміністрування мережевої інфраструктури

Для автоматизації контролю стану мережевої інфраструктури використовується декілька основних методів, вони представлені в таблиці 1.1.

Таблиця 1.1 – Порівняння методів адміністрування мережевої інфраструктури

Метод	Опис
Метод на основі спеціалізованих пакетів (приклад: System Center Configuration Manager, Tivoli)	Ґрунтується на використанні спеціалізованих пакетів для адміністрування. « + » Ці комплексні рішення охоплюють широкий спектр задач адміністрування та управління IT-інфраструктурою. « - » Досить висока вартість впровадження та підтримки таких систем, через велику функціональність і кількість налаштувань.

Кінець таблиці 1.1

Метод	Опис
Метод на основі спеціальних утиліт (приклад Advanced IP Scanner)	Полягає у використанні спеціалізованих утиліт для виконання окремих задач адміністрування мережевої інфраструктури. « + » Відносна простота - в Інтернеті існує безліч безкоштовних утиліт для виконання певних задач. « - » Відсутність гнучкості та універсальності, утиліта вирішує лише одну вузьку проблему.
Метод на основі спеціалізованих програм	Схожий на метод з використанням утиліт, але з використанням спеціалізованих програм. Спеціалізовані програми мають більший функціонал ніж вузькоспеціалізовані утиліти.
Метод на основі власних програм	Подібний до попереднього і теж базується на використанні спеціалізованих програм, але на відміну від окремих утиліт, які виконують вузькі завдання, тут йдеться про комплексні рішення. Охоплюють більшість аспектів адміністрування, гнучко налаштовувати потрібний функціонал, а також масштабувати систему. Недоліками можуть бути вартість та складність впровадження.
Метод на основі скриптів	Полягає у створенні та використанні скриптів для автоматизації задач. Скрипти пишутся інструментами вбудованими в операційну систему такими як PowerShell та Bash, або мовами програмування, наприклад, Python та іншими. Завдяки можливості інтеграції цих інструментів з операційною системою, можна створювати скрипти для виконання широкого кола функцій адміністрування.

Проаналізувавши таблицю 1.1 робимо висновок, що для завдань автоматизації вибору інструментальних засобів використовуємо метод на основі скриптів за рахунок інтеграції з операційною системою та виконання широкого кола функцій адміністрування.

Скрипти дозволяють регулярно опитувати мережеве обладнання (маршрутизатори, комутатори, сервери) за допомогою SNMP, CLI чи API. Можна збирати дані про завантаження ЦП і пам'яті, стан інтерфейсів, статуси служб. Це дає поточну інформацію про працездатність інфраструктури.

Виявлення аномалій, скрипти аналізують зібрані метрики і виявляють незвичні або порогові значення (наприклад, перевищення трафіку). Дозволяє вчасно вживати заходів і запобігати збоям.

Сповіщення та журналювання, адміністратори можуть отримувати сповіщення про події електронною поштою, Telegram чи SMS. Події також можуть журналюватися в БД.

Автоматичне відновлення, скрипти здатні автоматично перезавантажувати сервіси, віртуальні машини. Спрацьовує за певних умов, визначених в скрипті.

Збір та аналіз статистики. Дані моніторингу можна накопичувати і аналізувати для виявлення тенденцій. Дає можливість краще зрозуміти навантаження і спланувати розвиток інфраструктури.

Масштабованість і повторне використання, скрипти легко масштабуються шляхом додавання нових серверів чи пристроїв. Їх логіку може бути легко перевикористано в інших проектах.

Ефективність та економія часу. Автоматизація економить час адміністраторів і звільняє від рутини. Скрипти працюють швидко і здатні обробляти великі обсяги даних.

Таблиця 1.2 – Порівняльний аналіз методів адміністрування на основі скриптів

Характеристика	Counters	WMI	SNMP
Платформа	Windows	Windows	Мережеве обладнання
Сфери застосування	Моніторинг продуктивності	Інвентаризація, моніторинг стану	Моніторинг роботи пристроїв
Тип даних	Показники продуктивності та ресурсів	Конфігурація, стан компонентів	Дані про роботу мережевих пристроїв
Метод отримання даних	Get-Counter	WMI, CIM	SNMP менеджер
Джерела даних	Лічильники Windows	Класи WMI та CIM	SNMP MIB
Мова запитів	PowerShell	WQL	SNMP Query Language
Опис змінних моніторингу	Так	Так	Ні
Звіти	Так	Так	Так
Моніторинг безпровідного доступу	Так	Так	Так
Моніторинг в реальному часі	Так	Так	Ні
Система безпеки (кодування інформації, що передається)	Так	Так	Ні

Кінець таблиці 1.2

Наявність спеціальних утиліт	Так	Так	Так
Доступ до будь-яких компонентів інформаційної системи	Так	Так	Ні
Використання мережевого трафіку	Ні	Ні	Так

Проаналізувавши таблицю 1.2 зробимо висновок, що для завдань вибору скрипту про конфігурацію та стан мережевої інфраструктури обираємо технологію WMI.

1.4 Аналіз технологій опису об’єктів контроля стану

Для стандартизованого опису об'єктів моніторингу в мережевій інфраструктурі використовуються технології CIM та WMI.

CIM – стандарт опису об'єктів управління в IT [3]. Визначає класи, атрибути, методи.

WMI – технологія управління в ОС Windows [4]. Надає уніфікований доступ до різних параметрів системи.

Переваги стандартизованих моделей:

– спрощує інтеграцію різних систем управління;

– незалежність від конкретних постачальників та платформ;

– можливість централізованого адміністрування на основі політик.

Недоліки:

– складність реалізації, особливо для легасі систем;

– додаткове навантаження на інфраструктуру через уніфікацію моделей;

– потрібна підтримка стандартів виробниками обладнання та ПЗ.

Технології опису об'єктів контролю стану, такі як CIM та WMI, грають важливу роль у забезпеченні ефективного моніторингу та контролю стану мережевої інфраструктури.

CIM є стандартом для опису об'єктів та даних в інформаційних системах та мережах. Використовується для стандартизації способу, яким інформація про систему представляється та обмінюється.

Перевага у тому, що сприяє єдинообразному представленню даних, дозволяючи збільшити сумісність та ефективність обміну інформацією між різними пристроями та програмами.

Використовується для опису конфігураційних даних, ресурсів та подій в мережевих пристроях.

CIM визначає уніфіковані класи для представлення різноманітних концепцій та компонентів IT-інфраструктури: мереж, систем, служб тощо, представлені на рисунку 1.2. Кожен клас містить стандартизований набір властивостей та методів. Наприклад, клас CIM_ComputerSystem описує комп’ютер і включає властивості Name, Description, Status [5]. Це дозволяє уникнути залежності від конкретних рішень вендорів.

WMI є набором розширень для операційних систем Windows, які надають стандартизований інтерфейс для моніторингу та управління системами.

Перевага у тому, що дозволяє збирати інформацію про конфігурацію, ресурси та події на системах Windows, а також виконувати дії на відстані через мережу.

Використовується для автоматизації збору даних, виявлення проблем та віддаленого управління системами на базі Windows.

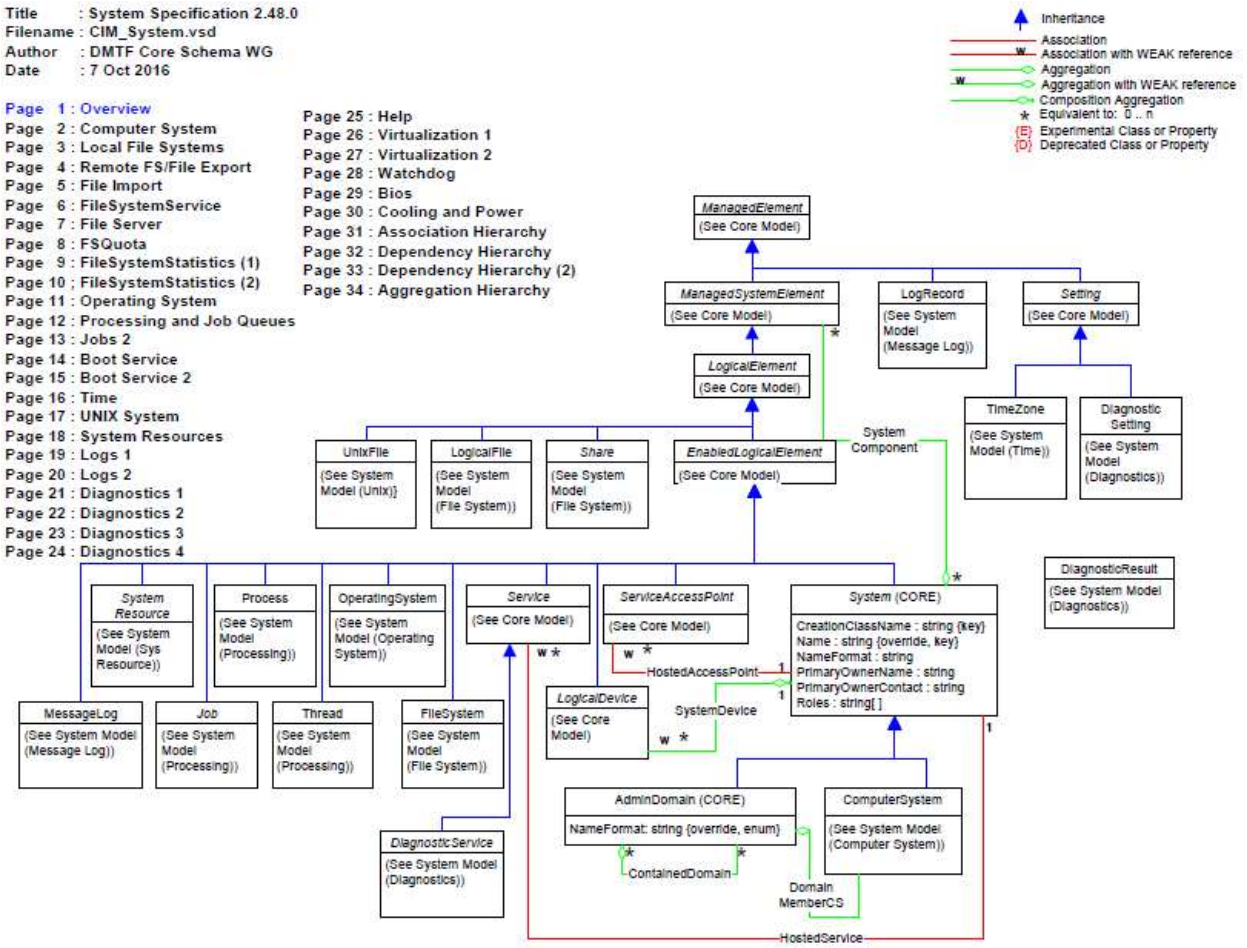


Рисунок 1.2 – Класифікація об’єктів, які використовуються в CIM

WMI надає доступ до об’єктів CIM в операційних системах Windows. За допомогою WMI можна отримувати дані про стан процесорів, пам'яті, мережевих з'єднань на локальних та віддалених хостах.

Технології CIM та WMI надають можливості для стандартизованого опису об’єктів мережевої інфраструктури з метою їх подальшого моніторингу та управління.

Проте існують проблеми інтеграції легасі систем, реалізації підтримки вендорами.

Отже, стандартизовані інформаційні моделі спрощують управління гетерогенним IT-середовищем, але потребують комплексного підходу з урахуванням плюсів та обмежень.

1.5 Автоматизації контроля стану мережевої інфраструктури

Розглянемо можливості автоматизації, зокрема класифікацію скриптів та технологій [6]:

– скрипти для збору даних, можуть бути розроблені для автоматичного збору даних про стан мережевих пристроїв, включаючи інформацію про використання ресурсів, стан з'єднань та виявлення аномалій;

– скрипти для моніторингу інцидентів, використовуються для автоматичного виявлення та реагування на інциденти, такі як атаки або відмови в обслуговуванні, може значно полегшити управління безпекою;

– автоматизовані засоби аналізу пропускної здатності, використовуються для постійного моніторингу та аналізу пропускної здатності може допомогти вчасно виявляти та вирішувати проблеми зі швидкодією мережі;

– системи управління подіями, використовуються технології управління подіями, такі як Syslog або SNMP, для автоматизованого отримання, фільтрації та аналізу подій у мережі;

– скрипти для забезпечення конфігураційної безпеки, автоматизують процеси перевірки та оновлення конфігурацій з метою забезпечення їх відповідності безпековим стандартам.

При автоматизації моніторингу мережі з використанням скриптів та існуючих технологій слід враховувати ряд особливостей:

– скрипти повинні бути максимально простими, надійними та ефективними;

– для збору даних доцільно використовувати готові API та протоколи (SNMP, WMI, netconf тощо) [7];

– варто уникати зайвого навантаження на інфраструктуру частими запитами;

– бажана можливість гнучкого налаштування параметрів моніторингу;

– скрипти мають коректно обробляти помилки та виключні ситуації;

– потрібно передбачити логування подій та результатів виконання;

– дані моніторингу доцільно зберігати в СУБД для подальшого аналізу;

– необхідно виконувати тестування скриптів перед впровадженням у робоче середовище.

Дотримання цих рекомендацій дозволить створити якісну систему автоматизованого моніторингу мережі засобами програмування.

1.6 Формування та використання бібліотек скриптів

Автоматизація контролю стану мережевої інфраструктури включає в себе використання різноманітних скриптів та технологій для ефективного моніторингу та реагування на зміни.

Ефективне управління та безпека мережевої інфраструктури потребує широкого використання спеціалізованих скриптів та програмних технологій.

По-перше, ключовим є розробка скриптів моніторингу для постійного контролю роботи мережі. Вони аналізують такі параметри, як пропускна здатність каналів, використання ресурсів пристроями та їх поточний статус. Завдяки цьому мережеві адміністратори можуть оперативно виявляти аномалії, прогнозувати потенційні збої та реагувати на конкретні несправності.

Крім того, актуальним є впровадження технологій глибокого аналізу мережевих протоколів. Це дозволяє ефективно відслідковувати будь-які відхилення від штатної роботи, а також виявляти спроби несанкціонованого

доступу та атаки. Такі технології значно підвищують рівень захищеності мережі в умовах постійно зростаючих кіберзагроз.

Для швидкого реагування на інциденти розробляються спеціальні скрипти автоматичного виконання відповідних процедур. Їх застосування дозволяє мінімізувати час усунення проблем та управляти кризовими ситуаціями.

Готові рішення доцільно збирати у бібліотеки скриптів для полегшення доступу та обміну досвідом між фахівцями. Це значно спрощує розробку й підтримку комплексних систем мережевої безпеки. Приклад існуючих довідкових бібліотек для PowerShell та Visual Basic представлений відповідно на рисунках 1.3 та 1.4.

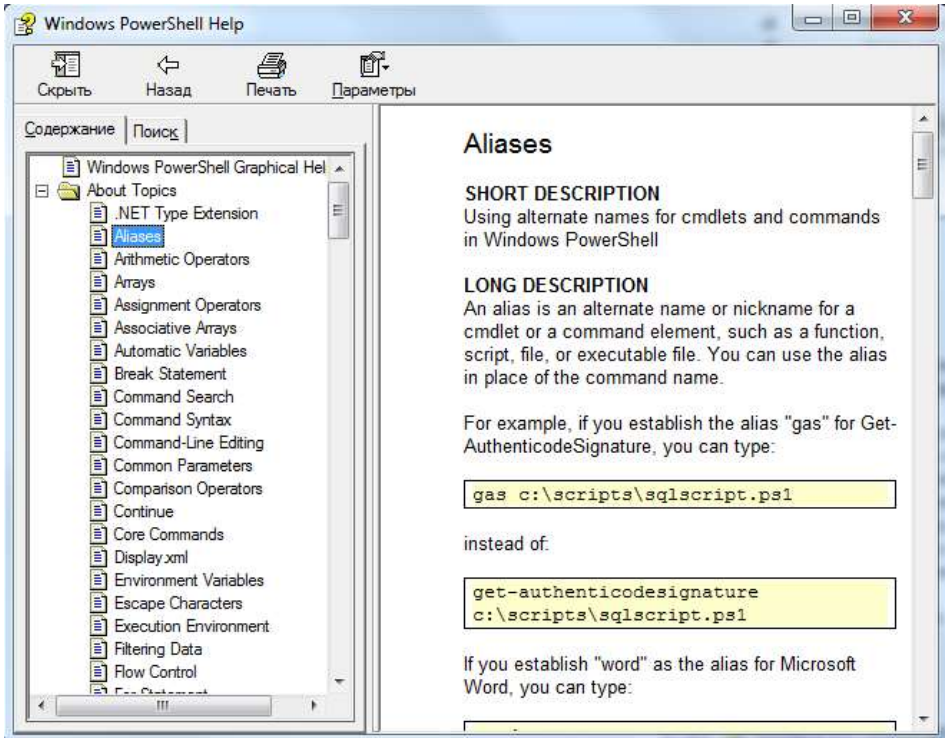


Рисунок 1.3 – Довідкова бібліотека для Powershell

І нарешті, важливу роль відіграє інтеграція з іншими системами через відкриті API. Вона надає розширені можливості збору та аналізу даних з різних джерел для підвищення ефективності моніторингу та захисту мережі.

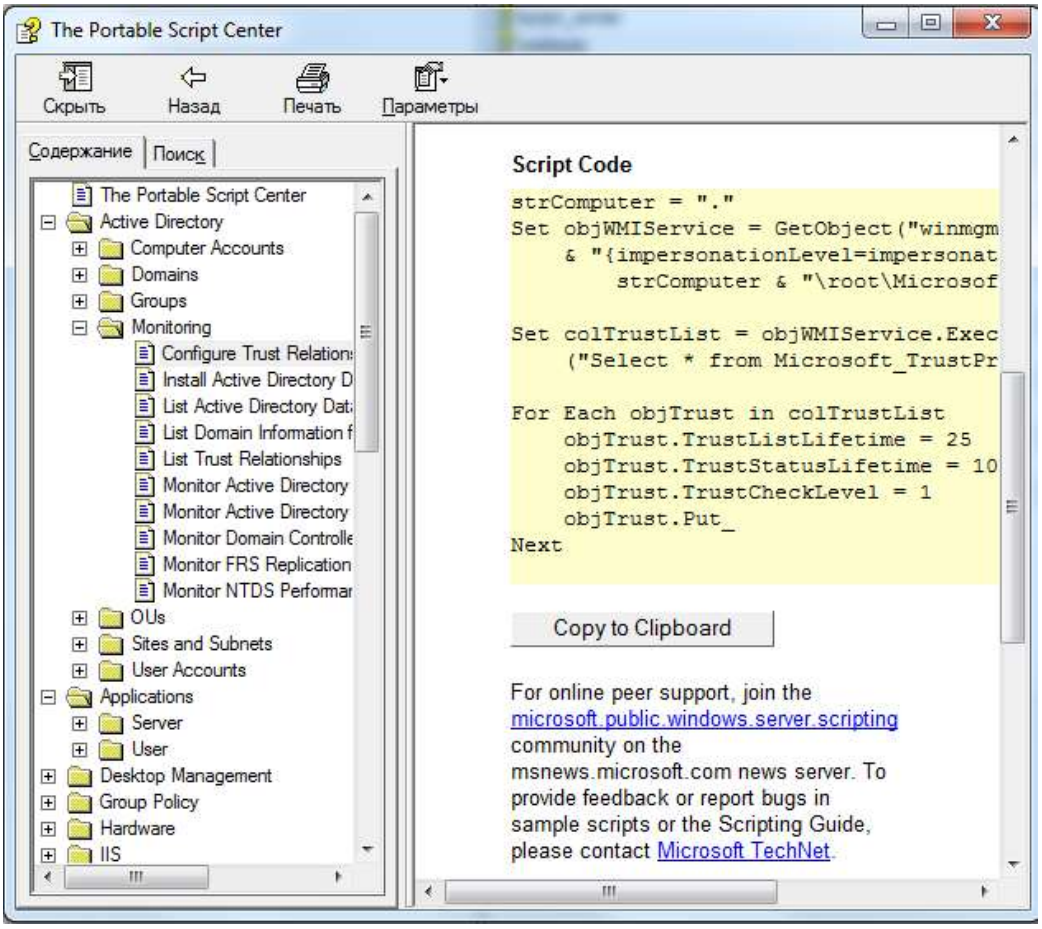


Рисунок 1.4 – Довідкова бібліотека для Visual Basic

Отже, поєднання скриптів автоматизації, спеціалізованих технологій аналізу та інтеграційних рішень є запорукою надійної та безпечної роботи сучасних мережевих інфраструктур.

Для прискорення розробки систем для вирішення завдань адміністрування доцільно використовувати готові бібліотеки скриптів з відкритим кодом.

Paramiko – популярна бібліотека Python для виконання команд та передачі файлів по SSH [8]. Вона надає програмний інтерфейс до протоколів SSHv2 та SFTP і підтримує автентифікацію за паролем і ключем. Завдяки гнучкості Paramiko багато використовується для автоматизації адміністрування Linux/UNIX систем, мережевого обладнання, резервного копіювання, розгортання додатків на серверах тощо. Також ця бібліотека дуже

корисна у скриптах та інструментах тестування проникнення для безпечного взаємодії з віддаленими хостами.

Scrapli – це ще одна бібліотека для виконання команд на мережевому устаткуванні, але через SSH/TELNET за допомогою Netconf та TextFSM [9]. Вона пропонує надійний API для налаштування пристроїв Juniper Junos, Cisco IOS/IOS-XE, Arista EOS. Головна її особливість - це робота на основі драйверів під конкретні ОС, що підвищує стабільність та швидкість взаємодії. Scrapli активно використовується для створення різноманітних систем мережевого моніторингу, збору статистики, аналізу конфігурацій та автоматизованого налаштування устаткування в хмарних мережах.

Scapy – потужний інтерактивний інструмент для маніпуляції мережевим трафіком та аналізу протоколів [10]. Дозволяє перехоплювати пакети, змінювати їх вміст, фальсифікувати трафік, сканувати порти тощо. Завдяки гнучкості Scapy широко використовується для тестування мережевої безпеки, розробки сканерів, засобів моніторингу та інших додатків, що працюють з мережею на низькому рівні.

Ping3 – проста бібліотека для відправки ICMP ping-запитів та вимірювання часу відгуку віддаленого хоста [11]. Вона дозволяє швидко оцінити доступність мережевого ресурсу та його приблизне підключення без занурення в низькорівневі деталі реалізації ICMP-протоколу. Це робить Ping3 зручним інструментом для моніторингу хостів у скриптах.

Psutil – бібліотека для отримання інформації про запущені процеси та системні ресурси в ОС сімейства UNIX [12]. Psutil дозволяє легко отримувати такі дані, як використання ЦП, пам'яті, мережі, завантаження дисків, список активних з'єднань тощо. Це корисно при розробці моніторингу продуктивності серверів, скриптів звітності, профілювання додатків та інших задач системного адміністрування.

PySNMP – реалізація популярних SNMP протоколів на мові Python [13]. Дає можливість легкої та ефективної взаємодії з мережевими пристроями, що підтримують SNMP (маршрутизаторами, комутаторами, системами

безперебійного живлення тощо). PySNMP часто використовується для створення скриптів збору даних, моніторингу обладнання, систем інвентаризації мережі.

Netmiko – бібліотека Python, що полегшує виконання команд та отримання даних з мережевих пристроїв через SSH та Telnet [14]. Netmiko абстрагує користувача від нюансів взаємодії з різними ОС пристроїв, надаючи єдиний інтерфейс для Cisco IOS, Juniper Junos, Arista EOS та ін. Це робить його ідеальним для автоматизації налаштування, збору конфігурацій та моніторингу стану обладнання.

Отже, зазначені бібліотеки є корисними інструментами для широкого спектру задач мережевого адміністрування та безпеки.

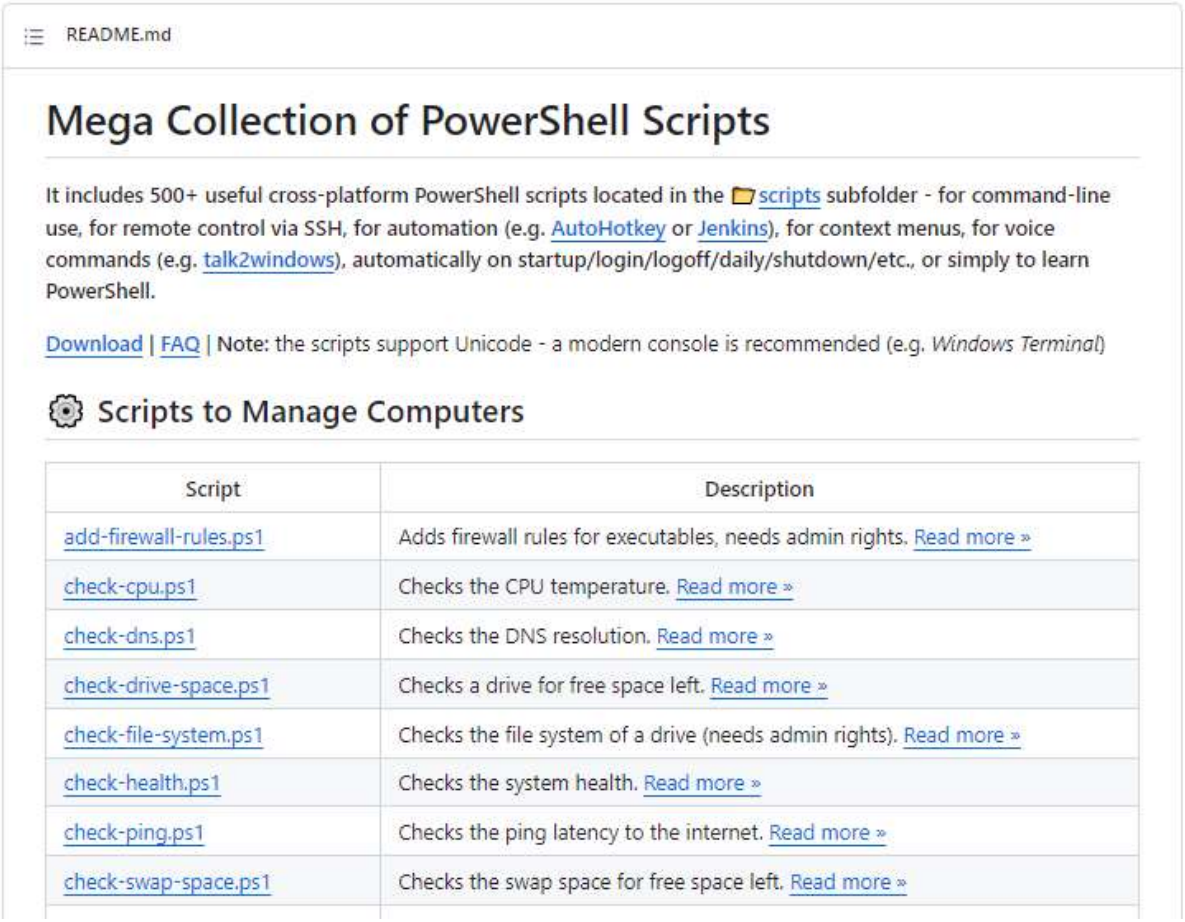
Також можна виділити кілька шарів при розробці бібліотеки скриптів:

– низькорівневі (взаємодія з пристроями, збір даних);
– обробка та агрегування даних;
– візуалізація та повідомлення;
– зберігання даних моніторингу;
– інтеграція з зовнішніми системами.

Така модульна структура спрощує подальше нарощування та розвиток системи моніторингу мережі.

При розробці бібліотек варто дотримуватися принципів модульності, інкапсуляції та низької зв'язаності між компонентами. Це забезпечить гнучкість та можливість легкої модифікації скриптів в майбутньому.

Також важливою є наявність документації та прикладів використання бібліотек, наприклад фрагмент скриптів з GitHub представлений на рисунку 1.5 [15]. Це полегшить їх освоєння новими розробниками та інтеграторами використовуваних систем.



README.md

Mega Collection of PowerShell Scripts

It includes 500+ useful cross-platform PowerShell scripts located in the scripts subfolder - for command-line use, for remote control via SSH, for automation (e.g. AutoHotkey or Jenkins), for context menus, for voice commands (e.g. talk2windows), automatically on startup/login/logoff/daily/shutdown/etc., or simply to learn PowerShell.

Download | FAQ | Note: the scripts support Unicode - a modern console is recommended (e.g. *Windows Terminal*)

Scripts to Manage Computers

Script	Description
add-firewall-rules.ps1	Adds firewall rules for executables, needs admin rights. Read more »
check-cpu.ps1	Checks the CPU temperature. Read more »
check-dns.ps1	Checks the DNS resolution. Read more »
check-drive-space.ps1	Checks a drive for free space left. Read more »
check-file-system.ps1	Checks the file system of a drive (needs admin rights). Read more »
check-health.ps1	Checks the system health. Read more »
check-ping.ps1	Checks the ping latency to the internet. Read more »
check-swap-space.ps1	Checks the swap space for free space left. Read more »

Рисунок 1.5 – фрагмент скриптів з GitHub

1.7 Постановка задачі дослідження

В результаті проведеного аналізу можна зробити висновок, що одним з ефективних підходів до автоматизації адміністрування мережевої інфраструктури є використання скриптів та готових програмних бібліотек для контролю її поточного стану та оперативного реагування на зміни.

Для вирішення даної задачі пропонується: розглянути основні технології адміністрування мережевої інфраструктури, визначити показники та критерії забезпечення функціональності мережевої інфраструктури, розробити інструментарій адміністратора для автоматизації вибору інструментальних засобів для адміністрування мережевої інфраструктури.

2 РОЗРОБКА МЕТОДІВ АВТОМАТИЗАЦІЇ ВИБОРУ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ДЛЯ АДМІНІСТРУВАННЯ МЕРЕЖЕВОЇ ІНФРАСТРУКТУРИ

2.1 Класифікація скриптів, технологій та бібліотек

Існує декілька підходів до класифікації скриптів, технологій та бібліотек для автоматизації адміністрування мережевої інфраструктури.

Скрипти за мовою програмування: Python, мовою оболонки Bash, мовою оболонки PowerShell та інші.

Скрипти за призначенням, для: моніторингу, резервного копіювання та відновлення, розгортання та налаштування, автоматизації рутинних завдань.

Скрипти за платформою: кросплатформенні, специфічні для Windows, специфічні для Linux, специфічні для мережевого обладнання.

За ліцензією: відкриті бібліотеки з відкритим кодом, пропрієтарні рішення.

За функціоналом: Загального призначення, спеціалізовані під конкретні задачі чи типи обладнання.

При виборі підходу до автоматизації потрібно враховувати всі ці аспекти.

2.2 Формування схем та структур бібліотек

Ефективне управління сучасною мережевою інфраструктурою потребує постійного моніторингу її стану та оперативної аналітичної обробки отримуваних даних. Для автоматизації цих процесів використовуються спеціальні інструментальні засоби.

Рисунок 2.1 – Концептуальне представлення інструментальних засобів адміністратора

Вони містять декілька основних компонентів, концептуальне представлених на рисунку 2.1. Блок отримання (збору) даних відповідає за отримання необхідних вхідних даних з використанням стандартних методів отримання інформації про конфігурацію та стан мережевої інфраструктури.

Блок обробки даних призначений для обробки зібраних даних отриманих на попередньому етапі, які піддаються нормалізації, фільтрації, кореляції, агрегуванню та іншій аналітичній обробці відповідно до поставлених цілей. Наприклад, групування подій за пріоритетами, виявлення аномалій, прогнозування тощо.

Блок збереження даних призначений для надійного зберігання отриманих і опрацьованих даних, які зберігаються у СУБД, хмарних сховищах чи локальних файлових системах для подальшого використання. Останній блок модуля формує на основі даних звіти, графіки, таблиці, сповіщення у зручному для користувача вигляді. Також можлива передача даних в інші інформаційні системи за допомогою відкритих API.

Контроль та керування всіма компонентами традиційно здійснюється централізовано за рахунок використання служб WMI і CIM. Гнучкість описаної архітектури дозволяє налаштувати ефективний моніторинг та аналіз стану мережі для вирішення конкретних завдань адміністрування.

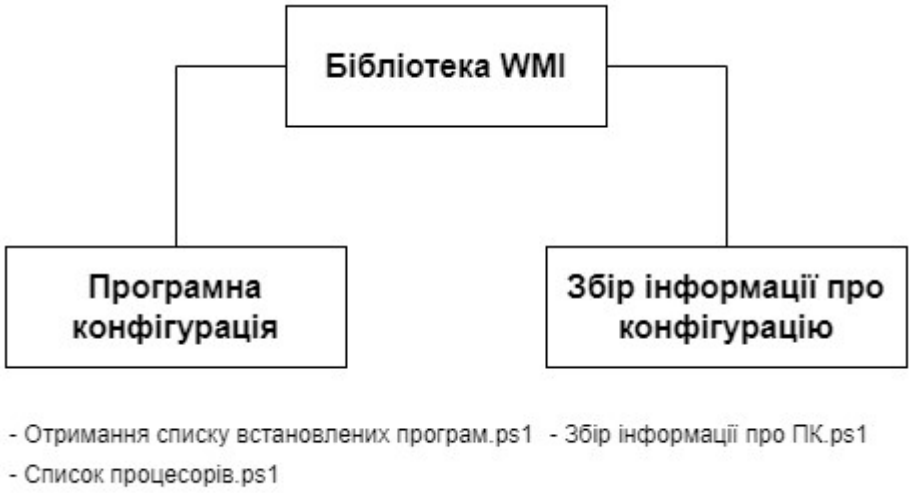


Рисунок 2.2 – Структура бібліотеки скриптів з використанням технології WMI

Щоб спростити використання переваг WMI на практиці, доцільно створювати бібліотеки спеціалізованих скриптів для вирішення конкретних задач адміністрування, структура якої представлена на рисунку 2.2.

Така бібліотека може поділятися на логічні розділи, наприклад, для збору інформації про встановлене на комп’ютерах програмне забезпечення, для отримання детальних даних про апаратне забезпечення чи операційну

систему. Кожен скрипт виконує строго визначену функцію, використовуючи можливості CIM та різноманітних WMI-класів. Завдяки простим і зрозумілим іменам, скрипти можна легко комбінувати між собою для вирішення більш складних задач інвентаризації та моніторингу IT-інфраструктури. Гнучкість WMI дозволяє постійно розширювати функціональність подібної бібліотеки адміністративних скриптів.

Ефективне адміністрування сучасних інформаційних систем потребує використання різноманітних скриптів для автоматизації рутинних задач. Проте з часом накопичується велика кількість готових рішень, написаних мовами Python, PowerShell, Bash та іншими. Це ускладнює їх пошук. Тому виникає необхідність у створенні структурованої колекції чи бібліотеки адміністративних скриптів. На рисунку 2.3 представлена схема об’єднання ієрархій існуючих бібліотек написаних різними мовами.

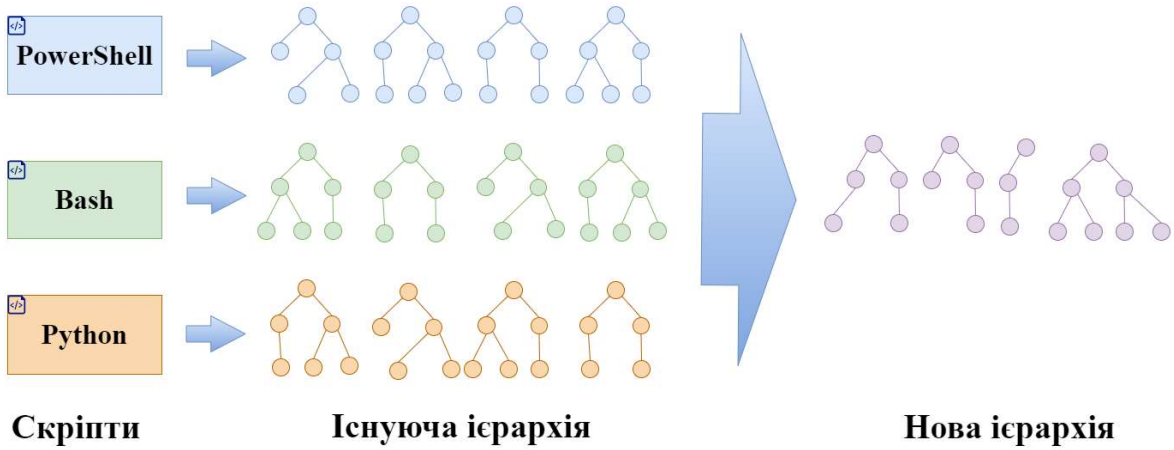


Рисунок 2.3 – Створення колекції з розрізнених бібліотек скриптів

Така колекція об'єднує існуючі розрізнені бібліотекі скриптів в ієрархічну структуру на основі виконуваних ними функцій або типів автоматизованих задач. В кожному розділі згруповані відповідні скрипти з різних мов програмування.

Такий поділ дозволяє швидко знаходити потрібні рішення та визначати можливості для їх подальшого вдосконалення. Крім того, з'являється можливість аналізувати поточне покриття задач автоматизації та виявляти прогалини. Загалом, формування бібліотеки скриптів на основі предметних областей суттєво полегшує повторне використання наявних наробок і пришвидшує вирішення типових задач адміністрування.

Структура типової програми скрипта для автоматизації адміністративних завдань зазвичай містить три основні логічні блоки, кожен з яких виконує певну роль в загальному алгоритмі роботи. Таку структуру визначимо, як схему шаблону програми скрипта, вона представлена на рисунку 2.4.

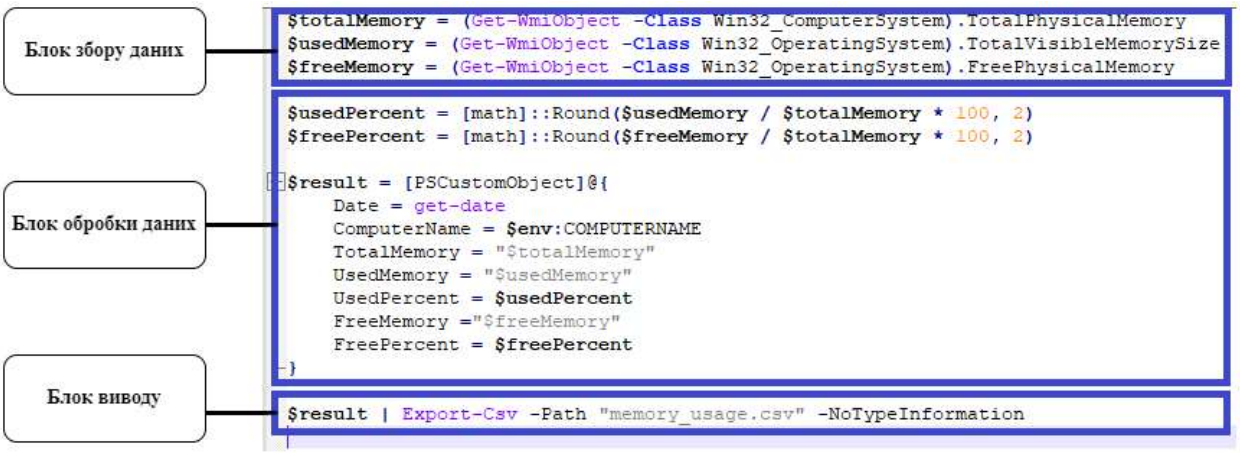


Рисунок 2.4 – Схема шаблону програми скрипта

На етапі введення даних, у блоці збору даних, відбувається збір необхідної вхідної інформації за допомогою технологій CIM/WMI. Зібрані дані зберігаються у змінних і передаються на наступний етап.

У блоці обробки даних ці змінні проходять необхідну обробку згідно з логікою програми. Після завершення обробки, отримані результати передаються в останній блок.

На заключному етапі в блоці виведення відбувається відображення оброблених даних у потрібному форматі. Це може бути збереження

результатів у базі даних, файлі Excel чи CSV, або виведення інформації безпосередньо адміністратору в його робочому середовищі.

Формування комплексної бібліотеки інструментів вимагає впровадження системного підходу до структуризації та каталогізації скриптів, що додаються. Адже саме гнучка та інтуїтивно зрозуміла внутрішня організація контенту значно підвищує зручність та ефективність подальшого використання наявних наробок фахівцями з адміністрування. Логічна схема формування бібліотеки інструментальних засобів адміністратора представлена на рисунку 2.5.

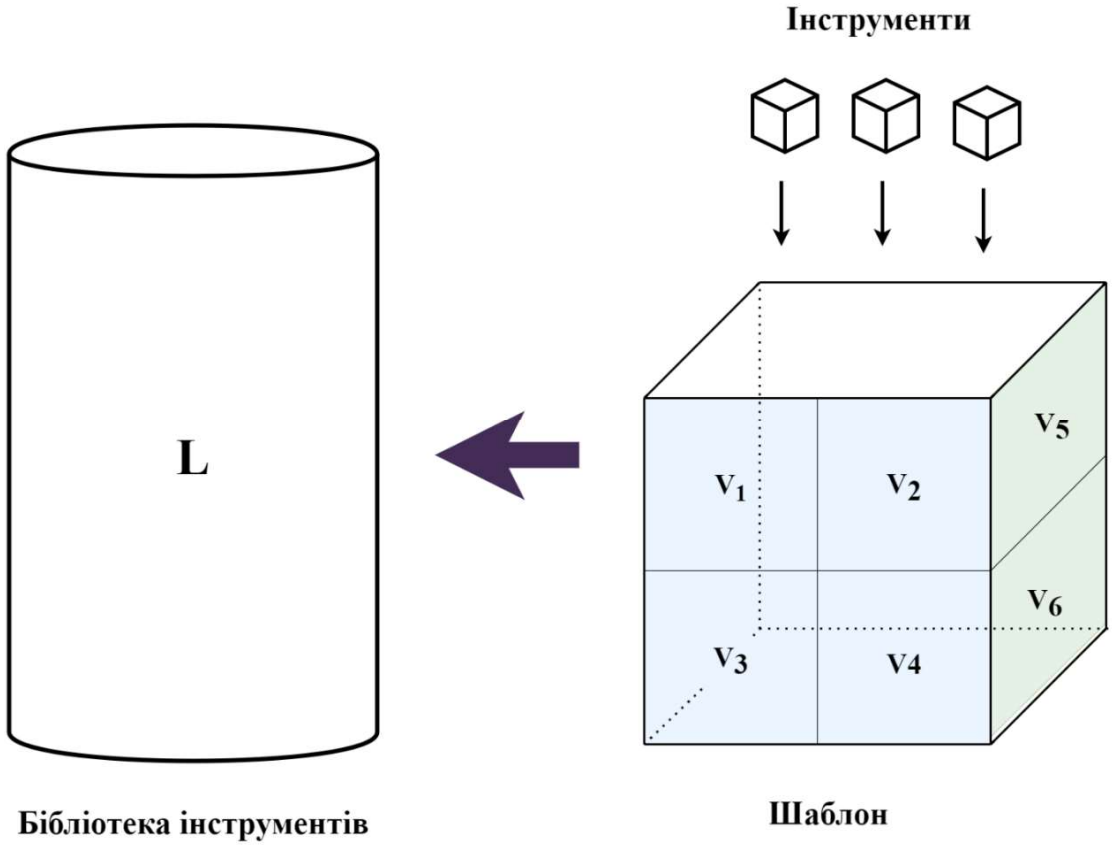


Рисунок 2.5 – Логічна схема формування бібліотеки інструментальних засобів адміністратора

Основою такої схеми виступає шаблон класифікації скриптів, що визначає принципи їх систематизації та структурування. Кожен скрипт набір

ознак, що відображають його функціональне призначення, об’єкти автоматизації, мови програмування тощо.

На базі цих та інших параметрів формується гнучка система категорій та логічних зв’язків, що дозволяє швидко знаходити потрібні інструменти адміністрування серед усього різноманіття наявних скриптів, представлених у бібліотеці.

2.3 Формування методу вибору програм скриптів із бібліотеки та методу її наповнення

Ефективне використання бібліотеки адміністративних скриптів вимагає реалізації зручних механізмів пошуку та відбору потрібних програмних рішень. Коли перед адміністратором постає конкретне завдання з автоматизації, він повинен швидко знаходити відповідні інструменти серед сотень наявних скриптів.

Для цього бібліотека може надавати функції тематичного пошуку, фільтрації за тегами, повнотекстового пошуку в описах скриптів. Знайдені за запитом скрипти виводяться у вигляді списку з коротким описом призначення кожного. Адміністратор аналізує можливості запропонованих варіантів і обирає оптимальний для вирішення свого завдання.

При необхідності створити новий скрипт, спочатку також здійснюється перевірка відсутності аналогів у бібліотеці. Після розробки, програма проходить оцінку якості, отримує необхідні метадані та додається до загального репозиторію для майбутнього використання.

Процес вибору потрібного скрипту для автоматизації з бібліотеки адміністративних включає наступні кроки: S1→ S2→ S3→ S4→ S5

Крок 1. Скласти опис задачі адміністрування.

Крок 2. Визначити Простір A = (v_1, v_2, v_3, v_4, v_5, v_6), де v1=<змінні>, v2=<завдання>, v3=<опис>, v4=<функція>, v5=<мова>, v6=<вид звіту>.

Крок 3. Вибрати бажані властивості, як значення Простору A($v^{(1)}$,$v^{(2)}$ $v^{(3)}$, $v^{(4)}$, $v^{(5)}$, $v^{(6)}$).

Крок 4. Для цих властивостей почати пошук потрібної програми скрипта $L'_k \in L'$ та $L' = \{L_k \mid \forall L_k \exists (i,j),\ v_i = v^{(1)}, v_j = v^{(2)}, v_m = v^{(3)}, v_n = v^{(4)}, v_p = v^{(5)}, v_q = v^{(6)}\}$.

Крок 5. Якщо пошук не дав результатів, розробляємо нову програму використовуючи шаблон, як Простір B = (d1, d2, d3), де d1=<блок збору даних>, d2=<блок обробки даних>, d3=<блок виводу>.

Функціонування бібліотеки корисних скриптів значною мірою залежить від добре налагоджених процесів її наповнення та оновлення. Адже саме актуальний і достатній обсяг практичних рішень у бібліотеці визначає цінність та затребуваність цього ресурсу командами системних адміністраторів.

Внесення скриптів має супроводжуватися належною перевіркою їх якості, унікальності функціоналу та коректності оформлення. Кваліфіковані адміністратори бібліотеки аналізують надані разом зі скриптами метадані, документацію, при необхідності додають додаткове маркування чи зв’язки з іншими скриптами, а також оцінюють корисність скрипту.

Процес наповнення бібліотеки скрипт програмами включає наступні кроки: S1 → S2 → S3 → S4 → S5.

Крок 1. Обрати скрипт програму.

Крок 2. Оцінити корисність скрипту.

Крок 3. Визначити змінні які використовуються у скрипті.

Крок 4. Описати типову задачу адміністрування, для якої використовується цей скрипт.

Крок 5. Створити класифікацію та ідентифікатори для програми скрипта.

Такий підхід дисциплінує та систематизує процес поповнення колекції корисних рішень автоматизації, роблячи її зручнішою для цільової аудиторії – команд системних адміністраторів та IT-фахівців.

2.4 Побудова структури бібліотеки скриптів

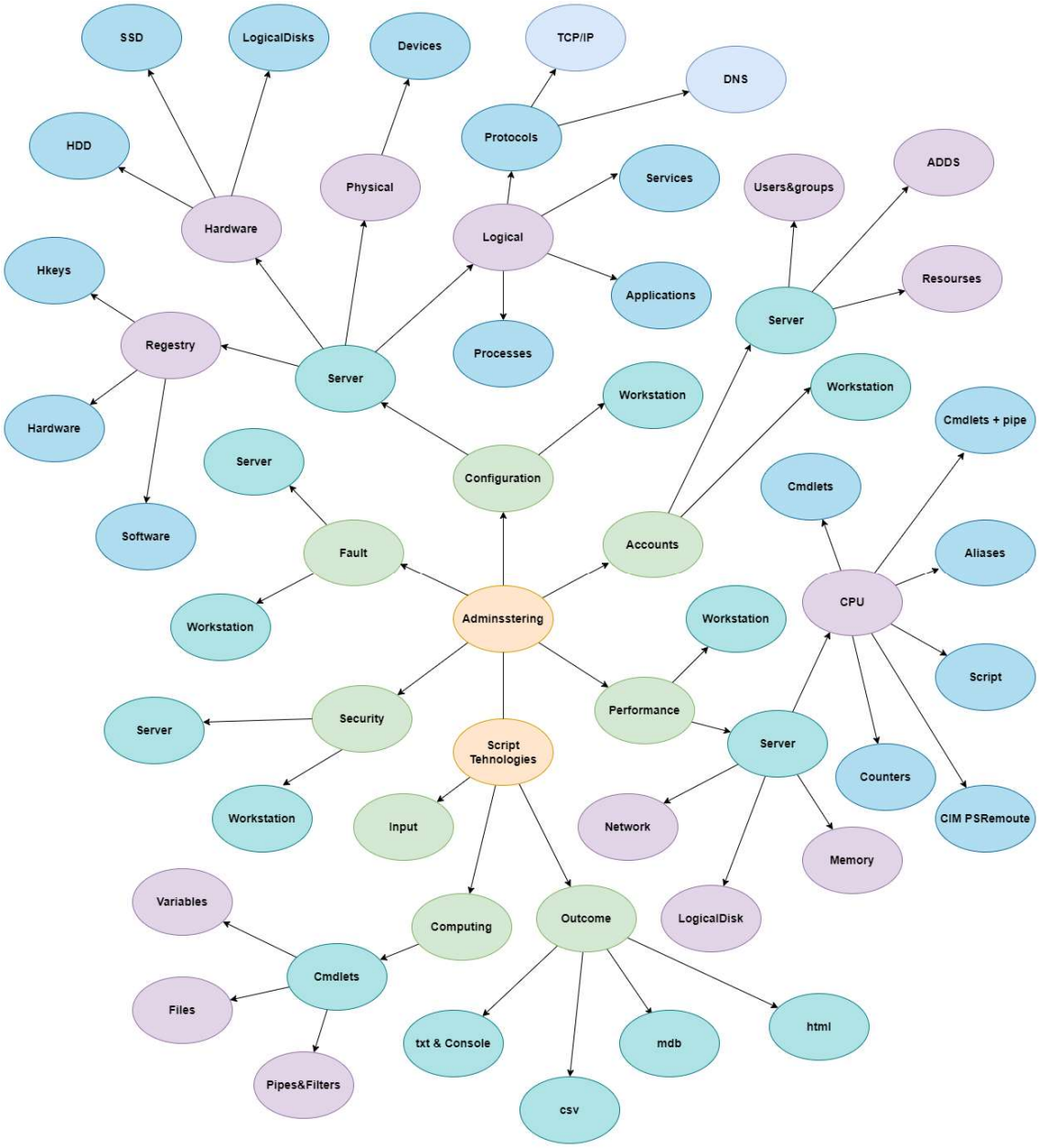


Рисунок 2.6 – Схема класифікації бібліотеки інструментальних засобів

Систематизація скриптів адміністративної автоматизації в уніфікованому репозиторії інструментів системного адміністрування може ґрунтуватися на концептуальній моделі FCAPS (Fault, Configuration, Accounting, Performance, Security). За якою представлена схема на рисунку 2.6. Ця загальновизнана в галузі мережевих технологій схема виділяє п’ять ключових функціональних сфер, які охоплює управління інформаційними системами та інфраструктурою. варто також розглянути їх систематизацію за цільовими об’єктами адміністрування. Адже на практиці скрипти часто призначаються саме для автоматизації завдань щодо конкретних технологій чи компонентів інфраструктури.

Відповідно до цієї моделі кожен скрипт бібліотеки може класифікуватися як такий, що відноситься до однієї або декількох з наступних категорій:

– управління несправностями (Fault), скрипти діагностики та реагування на проблеми;

– управління конфігурацією (Configuration), скрипти для інвентаризації та налаштування компонентів інфраструктури;

– адміністрування облікових записів (Accounting), скрипти для створення та керування обліковими записами та правами доступу;

– управління продуктивністю (Performance), скрипти моніторингу та оптимізації роботи інфраструктури;

– управління безпекою (Security), скрипти перевірки захищеності, фільтрації трафіку тощо.

Доцільно виділити такі категорії об’єктів автоматизації: сервери та серверне обладнання, мережеві пристрої (маршрутизатори, комутатори), робочі станції та пристрої користувачів тощо.

Така додаткова класифікація дає уявлення про спрямованість та сферу застосування того чи іншого скрипту без необхідності заглиблення в його вихідний код чи документацію. А систематизація полегшує навігацію та

пошук в бібліотеці інструментальних скриптів для виконання конкретних оперативних завдань адміністрування.

Зображена структура на рисунку 2.7 демонструє організацію бібліотеки утиліт адміністративної автоматизації у вигляді файлів довідки CHM (Compiled HTML Help).

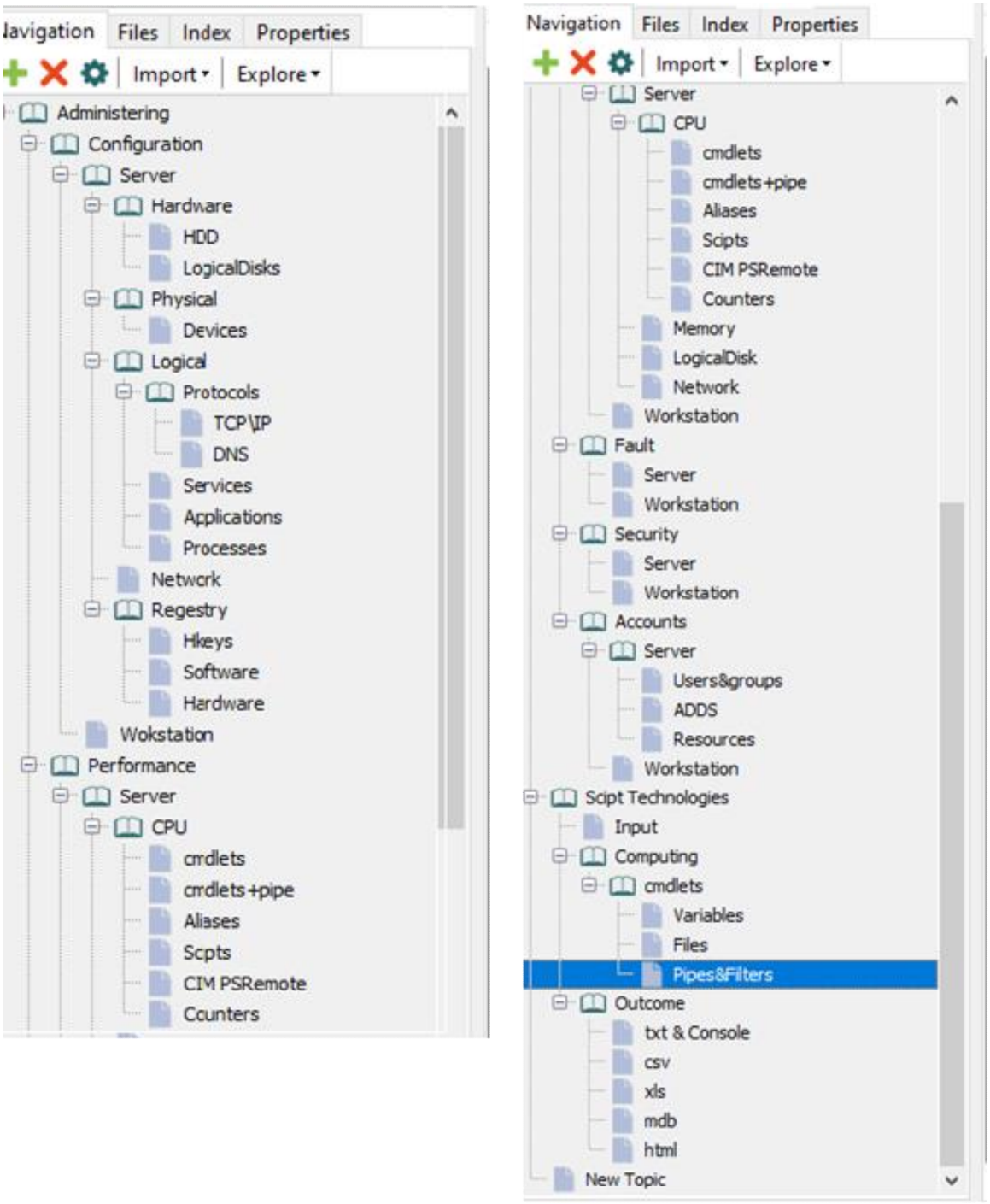


Рисунок 2.7 – Структура створеної бібліотеки скриптів

Цей формат дозволяє консолідувати всю необхідну документацію та описи разом зі скриптами в єдиному інформаційному ресурсі для максимальної зручності використання.

Як бачимо, створена бібліотека у програмі CHM Editor має ієрархічну структуру, поділену на основні предметні розділи. Кожен розділ включає конкретні скрипти, які функціонально пов'язані між собою або відносяться до тієї ж самої предметної області, що ґрунтується на концептуальній моделі FCAPS. Така зручна структурована організація скриптів в єдиному репозиторії спрощує їх пошук і застосування адміністраторами для повсякденної автоматизації рутинних задач.

3 ПРИКЛАДИ ВИКОРИСТАННЯ МЕТОДІВ ТА ТЕХНОЛОГІЙ АВТОМАТИЗАЦІЇ ВИБОРУ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ДЛЯ АДМІНІСТРУВАННЯ МЕРЕЖЕВОЇ ІНФРАСТРУКТУРИ ІЗ БІБЛІОТЕКИ

Зображений в таблиці 3.1 скрипт та результат його виконання (рисунок 3.1) ілюструє типовий процес запуску скрипта після вибору його з бібліотеки інструментів для вирішення завдання отримання інформації про дані конфігурації робочої станції.

Таблиця 3.1 – Приклад виконання скрипту на PowerShell вибраного з бібліотеки

Скрипт	Опис
*$colItems = get-wmiobject -class "**Win32_BaseBoard**" -namespace "root\CIMV2"* *#Set-ExecutionPolicy Unrestricted* *#Set-ExecutionPolicy Restricted* *function toHTML{* *if(!(Test-Path $env:homedrive\11)) {* *mkdir $env:homedrive\11* *}* *$var = "" | select Time, Computer_Name,Device_Name,Description, Manufacturer,Product_Name,Serial_Name, System_ID,Version* *$var.Time = Get-Date -Format T;* *$var.Computer_Name = $strComputer* *$var.Device_Name = $colItems.Caption* *$var.Description = $colItems.Description* *$var.Manufacturer = $colItems.Manufacturer* *$var.Product_Name = $colItems.Product* *$var.Srial_Number = $colItems.SerialNumber* *$var.System_ID = $colItems.Tag* *$var.Version = $colItems.Version* *$var |ConvertTo-Html -head "<title>Details</title>`n<style type=`"text/css`">`nbody { padding: 8px; line-height: 1.33 }`ntable { border-style: ridge }`ntd, th { padding: 10px; border-style: dotted; border-width: 1px }`nth { font-weight: bolder; text-align: center }`n</style>" | Out-File $env:homedrive\11\Win32_BaseBoard.htm* *Invoke-item $env:homedrive\11\Win32_BaseBoard.htm }* *toHTML*	Скрипт показує дані конфігурації, а постачальник WMI надає сховище, яке використовується для опису системи. Клас WMI Win32_ SystemBaseBoard представляє конфігурацію комп'ютерної системи під керуванням Windows. Програма додатково перетворює результати в HTML, який запускається в автономному режимі в Інтернет-браузері.

Time	Computer_Name	Device_Name	Description	Manufacturer	Product_Name	Serial_Name	System_ID	Version
22:04:32	XXX	Base Board	Base Board	FUJITSU	FJNB265		Base Board	B1

Рисунок 3.2 – Результат виконання скрипту на PowerShell

Спочатку адміністратор формулює потребу в автоматизації певної задачі у сфері його відповідальності. Далі за допомогою вбудованих механізмів навігації та пошуку він знаходить та обирає відповідний за функціоналом скрипт з бібліотеки інструментів адміністратора.

Наступним кроком є безпосереднє застосування скрипту до цільових систем чи компонентів інфраструктури для вирішення поставленого завдання, з можливою додатковою настройкою параметрів.

Такий підхід дозволяє многократно прискорити впровадження типових рішень з автоматизації рутинних процесів за рахунок використання наявного в бібліотеці напрацьованого функціоналу.

Наступний приклад з використанням скрипту на bash, вибраного із бібліотеки. У таблиці 3.2, представлено пояснення результату виконання скрипту. На рисунку 3.2 представлений код скрипту та результат його виконання.

Заключний приклад з використанням мови python. На рисунку 3.3 представлений код скрипту на мові python та результат його виконання.

Таблиця 3.2 – Пояснення результату виконання скрипту на bush

Назва	Значення	Опис
Використання CPU	13.5%	Середнє навантаження процесора за останній час
Використання RAM	1402/7890 МБ (17.5%)	Використана/загальна пам'ять та відсоток використання

Кінець таблиці 3.2

Назва	Значення	Опис
Швидкість мережі	1000 Мбіт/с	Поточна швидкість мережевого інтерфейсу
Втрачено пакетів	1255	Кількість мережевих пакетів, які було відхилено через помилки

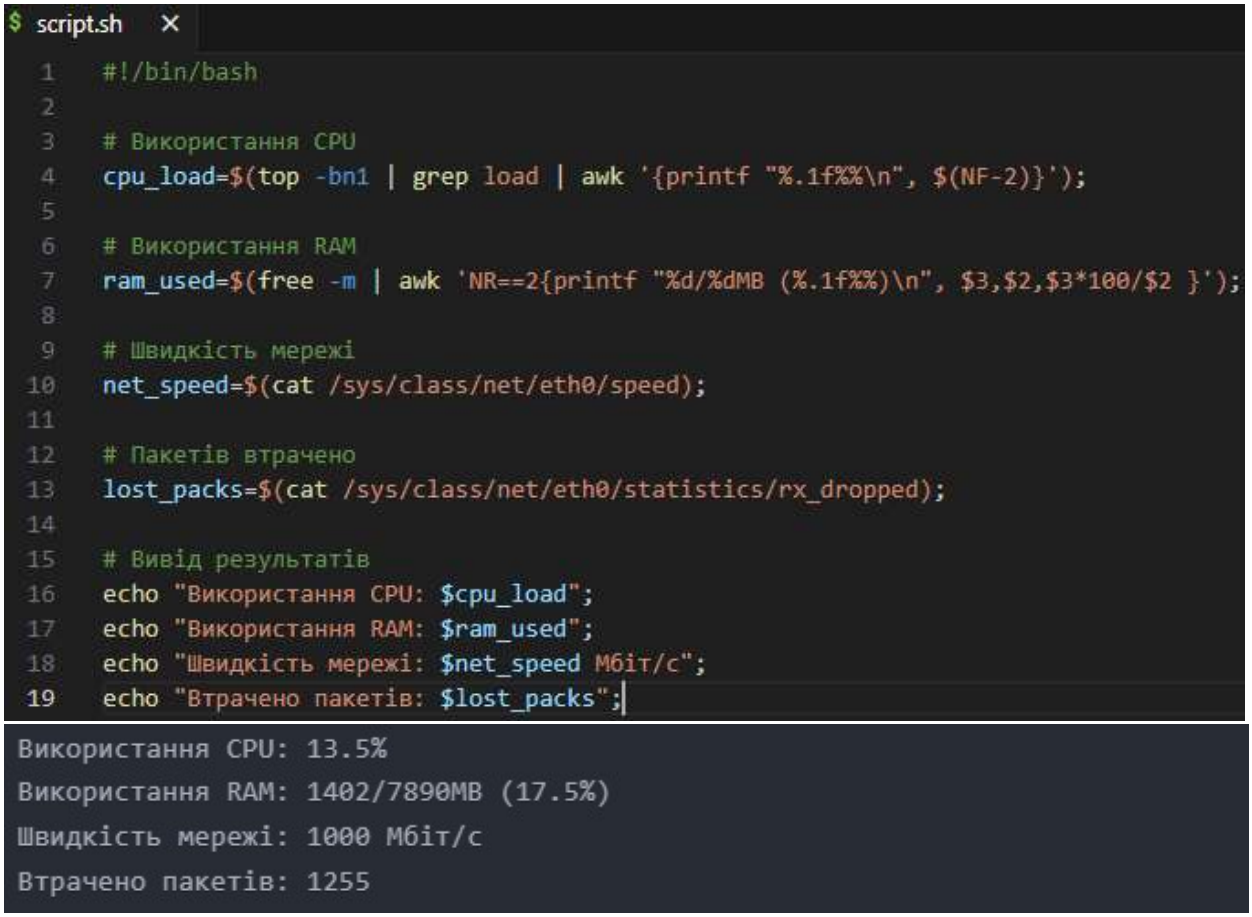


Рисунок 3.2 – Скрипт на bush і результат його виконання

script.py

```python
import psutil
import netifaces

cpu_use = psutil.cpu_percent(1)

mem_stats = psutil.virtual_memory()
mem_free = round(mem_stats.free/1024/1024)

net_speed = netifaces.ifaddresses('eth0')[netifaces.AF_INET][0]['netmask']
net_errors = netifaces.ifaddresses('eth0')[netifaces.AF_INET][0]['broadcast']

print(f'Використання CPU: {cpu_use}%')
print(f'Вільно ОЗП: {mem_free} Гб')
print(f'Швидкість передачі: {net_speed} Кбіт/с')
print(f'Помилок з\'єднання: {net_errors}')
```

```
Використання CPU: 4.3%
Використання RAM: 3.26/15.47 GB (21%)
Швидкість: 95.14 Мбіт/с
Втрачено пакетів: 204
```

Рисунок 3.3 – Скрипт на мові python і результат його виконання

ВИСНОВКИ

Формування ефективної бібліотеки інструментів адміністративної автоматизації як єдиного середовища накопичення корисних напрацювань у цій сфері вимагає системного та комплексного підходу. Ключовим є запровадження гнучкої логічної структуризації та класифікації скриптів за функціональними ознаками та цільовими об’єктами застосування. Це сприяє швидкій навігації та ідентифікації релевантних рішень з автоматизації.

Подальшого розвитку бібліотеки варто досягти реалізацією додаткових інструментів - семантичного пошуку, візуалізації взаємозв’язків, засобів підтримки прийняття рішень. Інтеграція скриптів з метаданими та документацією в єдиному середовищі типу CHM також підвищує зручність та ефективність використання наявних напрацювань.

Загалом, реалізація описаних підходів дозволить створити потужну експертну екосистему для прискорення впровадження автоматизації адміністративних процесів. Подальші дослідження можуть відбуватися в напрямках розширення функціоналу програмного забезпечення та структури інформаційного забезпечення. Для цього необхідно глибше вивчення предметної області.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Matt Oswalt, Christian Adell, Scott Lowe, Jason Edelman. Network Programmability and Automation: Skills for the Next-Generation Network Engineer. – O'Reilly Media, 2023. – 825 p.

2. Jennifer Davis. Modern System Administration: Managing Reliable and Sustainable Systems. – O'Reilly Media, 2022. – 325 p.

3. Common Information Model [Електронний ресурс] – Режим доступу до ресурсу: https://www.dmtf.org/standards/cim/cim_schema_v2540 (дата звернення 12.10.2023)

4. Windows Management Instrumentation [Електронний ресурс] – Режим доступу: https://learn.microsoft.com/en-us/windows/win32/wmisdk/wmi-start-page (дата звернення 12.10.2023)

5. CIM Schema [Електронний ресурс] – Режим доступу до ресурсу: https://www.dmtf.org/standards/cim/cim_schema_v2541 (дата звернення 12.10.2023)

6. Tolga Koca. Python Networking Solutions Guide: Leverage the Power of Python to Automate and Maintain your Network Environment. – BPB Publications, 2023. – 456 p.

7. Tim Peters. Mastering Python Network Automation: Automating Container Orchestration, Configuration, and Networking with Terraform, Calico, HAProxy, and Istio. – GitforGits, 2023. – 307 p.

8. Paramiko [Електронний ресурс] – Режим доступу до ресурсу: https://www.paramiko.org (дата звернення 12.10.2023)

9. Scrapli [Електронний ресурс] – Режим доступу до ресурсу: https://github.com/carlmontanari/scrapli (дата звернення 12.10.2023)

10. Scapy [Електронний ресурс] – Режим доступу до ресурсу: https://scapy.net (дата звернення 12.10.2023)

11. Ping3 [Електронний ресурс] – Режим доступу до ресурсу: https://github.com/kyan001/ping3 (дата звернення 12.10.2023)

12. Psutil [Електронний ресурс] – Режим доступу до ресурсу: https://github.com/giampaolo/psutil (дата звернення 12.10.2023)

13. Pysnmp [Електронний ресурс] – Режим доступу до ресурсу: https://github.com/etingof/pysnmp (дата звернення 12.10.2023)

14. Netmiko [Електронний ресурс] – Режим доступу до ресурсу: https://github.com/ktbyers/netmiko (дата звернення 12.10.2023)

15. Mega Collection of PowerShell Scripts [Електронний ресурс] – Режим доступу до ресурсу: https://github.com/fleschutz/PowerShell (дата звернення 12.10.2023).

16. Методичні вказівки щодо розробки та оформлення кваліфікаційної роботи (для студентів усіх форм навчання другого (магістерського) рівня програми "Інформаційні управляючі системи та технології") / Упоряд.:Петров К.Е., Левикін В.М., Чалий С.Ф., Євланов М.В., Саєнко В.І., Міхнов Д.К., Міхнова А.В., Чала О.В. – Харків: ХНУРЕ, 2021. – 30с.

17. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлювання. – Чинний від 22.06.2015. – Київ: ДП «УкрНДНЦ», 2016. – 31 с.

18. ДСТУ 8302:2015. Інформація та документація. Бібліографічні посилання. Загальні положення та правила складаня. – Чинний від 04.03.2016. – Київ: ДП «УкрНДНЦ», 2016. – 20 с.