**РОЗДІЛ 1. ФУНДАМЕНТАЛЬНІ АСПЕКТИ СКРИПТІВ ТА СФЕРИ ЇХНЬОГО ВПРОВАДЖЕННЯ**

**1.1. Сутність та функціональне призначення скриптів**

**Скрипт, термін якого походить від англійського "scenario", являє собою компактну програмну послідовність, призначену для виконання набору схожих завдань одне за одним. Якщо говорити простіше, то скрипт є засобом для автоматизації певних шаблонних дій, наприклад, перетворення в тексті символів дефіса на тире, або стандартних лапок (“”) на кутові («»), а також для підготовки файлової структури вебсайту до розміщення на сервері та для багатьох інших подібних операцій. Створення скрипту можливе з використанням будь-якої мови програмування, що дозволяє формулювати інструкції як логічну послідовність кроків, тобто у формі алгоритмів. Прикладами таких кроків можуть бути: відкриття вебсайту X, активація елемента Y, витримка паузи тривалістю N секунд, після чого слідує закриття сайту та інші аналогічні команди.**

**Різновиди скриптів**

**Скрипти, а також мови, що використовуються для їх написання, можна класифікувати за кількома типами:**

* **Консольні (або командні), прикладами яких є Bash та PowerShell.**
* **Розширювальні (або вбудовувані), до яких належать JavaScript, Lua, VBA.**
* **Макроси, або скрипти для графічних інтерфейсів, специфіка яких визначається конкретною програмою.**
* **Універсального призначення, такі як Python, PHP, JavaScript.**

**Переваги використання скриптів**

**Характерні переваги скриптів роблять їх оптимальним інструментарієм для тих завдань, на які вони орієнтовані. Гнучкість та оперативність розробки є одними з ключових позитивних аспектів роботи зі скриптами. Скрипти дають змогу швидко реалізовувати та модифікувати програмні функції, минаючи етапи компіляції та перекомпіляції коду. Ця особливість робить їх чудовим вибором для оперативного створення прототипів та вирішення завдань невеликого масштабу. До того ж, завдяки своїй відносній простоті та зрозумілій структурі, вони полегшують процеси тестування та налагодження. Важливою перевагою при написанні скриптів є також їхня адаптивність, що полягає у можливості доповнювати скрипт новими функціями в процесі розробки, або ж перепрофілювати його для виконання абсолютно інших завдань.**

**Недоліки скриптів**

**Проте, робота зі скриптами має і певні обмеження. Вони можуть демонструвати нижчу продуктивність та бути менш придатними для масштабування. Окрім цього, скрипти іноді виявляються неефективними з точки зору використання ресурсів, споживаючи їх у значних обсягах, що є небажаним фактором. Також, скрипти можуть бути більш схильними до виникнення помилок, що вимагає пильнішого контролю та ретельної обробки нестандартних ситуацій.**

**Ключові характеристики скриптів:**

* **Інтерпретованість: Скрипти обробляються виконавцем без попереднього перетворення у машинний код (компіляції), що значно прискорює їх тестування та внесення змін.**
* **Адаптивність: Вони здатні функціонувати в різноманітних програмних оточеннях та вирішувати широкий діапазон завдань.**
* **Спрямованість на автоматизацію: Застосування скриптів дозволяє мінімізувати обсяг ручних операцій.**
* **Здатність до взаємодії з іншими програмами: Скрипти можуть ініціювати виконання системних команд, оперувати файлами, взаємодіяти з базами даних та мережевими ресурсами.**

**Основні напрямки застосування скриптів**

**Найбільш поширеною сферою використання скриптів є веб розробка: деякі з них функціонують на боці клієнта, реагуючи на дії користувача, тоді як інші відповідають за обробку даних, їх передачу на сервер та отримання з нього. Разом з тим, важливо розуміти, що вебсайти не є вичерпною областю застосування скриптів. Скрипти суттєво полегшують автоматизацію робочих процесів у сфері розробки програмного забезпечення. Можна створити спеціалізований код для вирішення конкретного завдання, і такий сценарій буде обробляти інформаційні потоки, генерувати діаграми на основі вхідних даних, взаємодіяти з серверною частиною, створювати анімаційні ефекти для користувацького інтерфейсу або виконувати інші необхідні дії. У контексті роботи з електронними таблицями, скрипти дають змогу автоматизувати обчислення, причому розробити такий сценарій під силу не тільки програмісту, але й рядовому офісному співробітнику.**

**Узагальнюючи, можна виділити такі напрямки використання скриптів:**

* **Автоматизація завдань системного адміністрування (наприклад, конфігурування серверів, моніторинг активних процесів, оновлення версій програмного забезпечення).**
* **Автоматизація бізнес-операцій (зокрема, формування звітності, обробка значних масивів даних).**
* **Автоматизація мережевих завдань (конфігурування мережевих пристроїв, аналіз трафіку).**
* **Автоматизоване створення резервних копій інформації.**

**1.2. Класифікація скриптів та їх специфіка**

**Коли мова заходить про скрипти, найчастіше розглядають дві фундаментальні категорії: ті, що виконуються на стороні клієнта, і ті, що працюють на сервері.**

**Клієнтські скрипти, яскравим представником яких є JavaScript, активуються та працюють безпосередньо у веббраузері кінцевого користувача. Вони надають розробникам інструменти для реалізації широкого спектра інтерактивних компонентів на вебсторінці, включаючи анімаційні ефекти, діалогові вікна, що з'являються поверх основного контенту, форми з валідацією даних до їх відправки, і навіть браузерні ігри. Використання клієнтських скриптів дозволяє користувачам взаємодіяти з вебсайтом без потреби повного перезавантаження сторінки, що суттєво покращує загальний досвід використання ресурсу, роблячи його більш комфортним та динамічним.**

**Серверні скрипти, до яких належать такі мови, як PHP, Python та Ruby, функціонують на стороні сервера, а не в браузері користувача. Їхня основна функція полягає в обробці запитів, що надходять від клієнтських додатків. Наприклад, коли відвідувач сайту заповнює певну форму і натискає кнопку для відправки даних, саме серверний скрипт приймає цю інформацію, обробляє її та може виконати низку подальших дій, таких як збереження отриманих даних у базі даних, відправлення повідомлення електронною поштою або генерація та відображення динамічного контенту на сторінці. Завдяки серверним скриптам, вебсайти набувають розширеної функціональності та здатності адаптуватися до індивідуальних потреб користувачів.**

**1.3. Застосування скриптів для цілей автоматизації**

**Впровадження скриптів для автоматизації різноманітних процесів надає такі переваги:**

* **мінімізація кількості помилок при виконанні одноманітних, рутинних завдань;**
* **оптимізація часових витрат на операції, що регулярно повторюються;**
* **скорочення фінансових витрат на адміністрування та технічну підтримку систем.**

**Скрипти активно використовуються для автоматизації широкого кола завдань. Як приклад, вони можуть бути налаштовані на періодичне створення резервних копій важливих даних, обробку вмісту текстових файлів або управління конфігураціями серверного обладнання. До переваг залучення скриптів в автоматизації належать зростання ефективності, забезпечення відтворюваності результатів та гнучкість у налаштуванні. Вони сприяють швидкому та точному виконанню численних операцій, що, у свою чергу, вивільняє час для концентрації на інших пріоритетних справах та знижує ймовірність допущення помилок.**

**Приклади реалізації автоматизації:**

* **Автоматизоване створення резервних копій файлових даних.**
* **Автоматизований процес оновлення версій програмного забезпечення.**
* **Здійснення моніторингу поточного стану серверів.**
* **Автоматичний запуск визначених додатків під час старту операційної системи.**

**Автоматизація процедури створення резервних копій даних**

**Процедура резервного копіювання передбачає створення дублікатів критично важливих даних, які можуть бути використані для відновлення у випадку їхнього пошкодження або повної втрати. Проте, з огляду на сучасні обсяги інформації, що генеруються та обробляються в організаціях, ручний метод копіювання є малопродуктивним та підвищує ризик помилок. У зв'язку з цим постає гостра потреба в автоматизації процесів, пов'язаних зі створенням резервних копій.**

**Автоматизація резервного копіювання дозволяє суттєво знизити ризики, обумовлені людським фактором, прискорити виконання операцій, гарантувати регулярність створення копій, а також забезпечити централізований контроль над процесом збереження даних. Переважно, автоматизовані підходи базуються на використанні спеціалізованого програмного забезпечення або скриптів, які активуються за попередньо визначеним розкладом і здійснюють копіювання даних у фоновому режимі, не заважаючи поточній роботі.**

**1.4.1. Обґрунтування необхідності резервного копіювання**

**Проблема забезпечення резервного копіювання даних належить до найбільш критичних аспектів інформаційної безпеки. У сучасну цифрову епоху інформація перетворилася на один із найцінніших активів. Її втрата чи пошкодження можуть спричинити надзвичайно серйозні наслідки як для приватних осіб, так і для діяльності підприємств, установ та організацій різного профілю. Процес резервного копіювання (відомий також як "backup") полягає у створенні дубліката даних, який може бути використаний для відновлення оригінального вмісту у випадку його втрати або якогось пошкодження.**

**Нагальність цього процесу обумовлена низкою чинників:**

* **Збої технічного характеру. Компоненти зберігання даних, такі як жорсткі диски, твердотільні накопичувачі, серверне обладнання та інші пристрої, не є абсолютно надійними та мають обмежений термін служби. Відмови можуть траплятися непередбачувано. У ситуації фізичної несправності пристрою, відновлення даних без наявної резервної копії часто стає неможливим.**
* **Помилки на рівні програмного забезпечення. Велика кількість випадків втрати даних є наслідком помилок у функціонуванні програмного забезпечення або некоректного використання системних можливостей. Ненавмисне видалення файлів, пошкодження структур баз даних або неправильно виконані оновлення можуть призвести до втрати важливої інформаційної бази.**
* **Кібернетичні атаки. У сучасному інформаційному просторі постійно зростає рівень загрози кібератак, зокрема поширення шкідливого програмного забезпечення, що здійснює шифрування або видалення даних (наприклад, програми-здирники). За таких обставин, наявність актуальної резервної копії дає змогу відновити працездатність системи без незворотної втрати інформації.**
* **Людський фактор. Найбільш частою причиною втрати інформації є неумисні дії самих користувачів: випадкове видалення файлів, помилкові налаштування, проста неуважність. Впровадження автоматизованого резервного копіювання допомагає зменшити негативний вплив людського фактора.**
* **Природні катаклізми та форс-мажорні обставини. Пожежі, затоплення, значні коливання напруги в електромережі та інші непередбачувані події можуть спричинити фізичне пошкодження технічних засобів, на яких зберігається інформація. У таких критичних ситуаціях збереженою виявиться лише та інформація, копії якої були розміщені на віддалених або спеціально захищених носіях.**
* **Вимоги законодавства та бізнес-стандартів. У багатьох галузях, особливо таких як фінанси, медицина чи юриспруденція, діють нормативні акти, що регламентують обов'язкове зберігання даних протягом певного періоду. Відсутність належним чином організованих резервних копій у таких випадках може розцінюватися як порушення чинного законодавства.**

**Методи реалізації резервного копіювання**

**Існують різноманітні підходи до автоматизації процесу створення резервних копій:**

* **Локальне резервне копіювання: передбачає створення копії даних на тому ж комп'ютері (на іншому логічному чи фізичному диску) або на підключеному зовнішньому накопичувачі.**
* **Мережеве резервне копіювання: полягає у збереженні резервних копій на спеціалізованих файлових серверах або у мережевих системах зберігання даних (NAS).**
* **Хмарне резервне копіювання: включає автоматичне передавання резервних копій даних до хмарних сервісів зберігання, таких як Google Drive, Dropbox, OneDrive.**

**Здійснення резервного копіювання під час завантаження системи**

**Один із найбільш практичних та результативних методів автоматизації процедури створення резервних копій полягає у їхньому виконанні на етапі запуску операційної системи. Такий підхід забезпечує регулярне виконання резервного копіювання без потреби в ручному запуску відповідних утиліт чи скриптів.**

**У рамках даної дипломної роботи передбачається створення скрипту, який буде автоматично ініціюватися при старті комп'ютера та виконувати процедуру резервного копіювання заздалегідь визначених даних. Це рішення дозволить гарантувати збереження актуальної інформації перед початком активної сесії користувача в системі, тим самим мінімізуючи ризик втрати важливих змін, внесених у дані.**

**Ключові функціональні можливості скрипту**

* **Ідентифікація каталогів, призначених для процедури резервного копіювання.**
* **Автоматизоване перенесення файлів до обраного сховища.**
* **Копіювання даних здійснюватиметься до вказаної користувачем директорії, яка може бути розміщена:** 
  + **на окремому розділі локального диска;**
  + **на зовнішньому пристрої зберігання (USB-флеш, зовнішній HDD/SSD);**
  + **у мережевому сховищі;**
  + **у хмарному сервісі зберігання даних.**
* **Процес копіювання буде відбуватися в автоматичному режимі, не вимагаючи втручання з боку користувача, що сприятиме збереженню звичного ритму роботи без виникнення додаткових затримок.**
* **Оптимізація самого процесу копіювання для ефективного використання ресурсів.**
* **Забезпечення можливостей для подальшого масштабування та гнучкого налаштування функціонала.**
* **У перспективі, розроблене рішення може бути розширене шляхом додавання, наприклад, графічного веб інтерфейсу для конфігурації параметрів, впровадження системи сповіщень електронною поштою про результати виконання копіювання, або інтеграції з існуючими системами моніторингу.**

**Інструменти для технічної реалізації**

**Розробка скрипту буде здійснюватися мовою програмування Python, із залученням стандартних бібліотечних модулів, таких як os, shutil, datetime, а також, за необхідності, додаткових інструментів для ведення журналів виконання (логів) та програмних інтерфейсів (API) для взаємодії з хмарними сервісами зберігання даних.**

**Перелік інформаційних джерел**

* **Foxminded.ua: “Що таке скрипти і навіщо вони потрібні?”**
* **William Shotts. "The Linux Command Line: A Complete Introduction". No Starch Press, 2019.**
* **Don Jones, Jeffrey Hicks. "Learn PowerShell Scripting in a Month of Lunches". Manning Publications, 2018.**
* **Mark Lutz. "Learning Python". O'Reilly Media, 2013.**
* **Офіційна документація Python:** [**https://docs.python.org**](https://docs.python.org/)
* **Офіційна документація Bash:** [**https://www.gnu.org/software/bash/manual/bash.html**](https://www.gnu.org/software/bash/manual/bash.html)
* **Microsoft Docs – PowerShell:** [**https://docs.microsoft.com/en-us/powershell/**](https://docs.microsoft.com/en-us/powershell/)
* **Google Cloud Backup Solutions:** [**https://cloud.google.com/backup**](https://cloud.google.com/backup?authuser=2)
* **Dropbox API Documentation:** [**https://www.dropbox.com/developers**](https://www.dropbox.com/developers)
* **Microsoft OneDrive Documentation:** [**https://learn.microsoft.com/en-us/onedrive/developer/**](https://learn.microsoft.com/en-us/onedrive/developer/)