

## Індивідуальне практичне завдання

Адміністрування комп'ютерних систем часто вимагає автоматизації певних дій за допомогою сценаріїв. В операційній системі Linux для цього використовуються sh-сумісні інтерпретатори командної мови. Найбільш популярний серед них Bash, як правило, використовується через спеціально написані скрипт-файли. Скрипт-файл — це звичайний текстовий файл, який вміщує послідовність команд Bash, але у властивостях файлу встановлено прапорець на ВИКОНАННЯ.

```
usb_restart.sh
-----
    власник  група   інші
    [гwx]   [г-х]  [г-х]
Мітка ===      ===      ===
```

Для перевірки встановлених дозволів з командного рядка використовується команда

```
ls -l [ім'я_файлу]
```

Наприклад,

```
ls -l usb_restart.sh
```

результат:

```
-гwxг-хг-х 1 vgorbenko vgorbenko 956 лип 2 2023 usb_restart.sh
```

де «х» вказує, що файл дозволено запускати на виконання.

Для зміни дозволів з командного рядка використовується команда:

```
chmod [дозволи] [ім'я файлу]
```

Наприклад,

```
chmod u=rwx,g=rwx,o=rwx usb_restart.sh
```

змінить дозволи на

```
-гwxгwxгwx 1 vgorbenko vgorbenko 956 лип 2 2023 usb_restart.sh
```

а команда (chmod розуміє і цифрове задавання дозволів):

```
chmod 755 usb_restart.sh
```

поверне дозволи у попередній стан

```
-гwxг-хг-х 1 vgorbenko vgorbenko 956 лип 2 2023 usb_restart.sh
```

Таким чином, такий скрипт-файл можна запускати з командного рядка, або планувати його запуск у певний час, або при певній події в системі.

Для вивчення структури та базових принципів скрипт-фалів на Bash існує достатня кількість різноманітних матеріалів в Інтернет. Також, для вивчення пропонується література в модулі ІДЗ.

## Завдання 1

1. Напишіть Bash скрипт-файл, за допомогою якого буде виконано пошук файлів розміром, що задається користувачем у певному каталозі. Діапазон (min - max) розмірів файлів задається користувачем як перший та другий аргумент командного рядка, а ім'я каталогу задається користувачем в якості третього аргументу командного рядка.
2. Напишіть Bash скрипт-файл, в якому використовується цикл for і виводить на консоль розміри та права доступу до усіх файлів у заданому каталозі та усіх його підкаталогах. Ім'я каталогу задається користувачем в якості першого аргументу командного рядка.
3. Напишіть Bash скрипт-файл для пошуку рядка, що задається користувачем, у всіх файлах заданого каталогу і усіх його підкаталогах. Рядок та ім'я каталогу задаються користувачем як перший та другий аргументи командного рядку. На консоль повинні вивестись повний шлях та імена файлів, у вмісті яких є рядок, що задається, та їх розмір.
4. Напишіть Bash скрипт-файл для пошуку однакових по вмісту файлів у двох каталогах, наприклад, Dir1 та Dir2. Користувач задає ім'я Dir1 та Dir2 в якості першого та другого аргументів командного рядка. У результаті роботи скрипту файли, які мають в Dir1, порівнюються з файлами в Dir2 за їх вмістом, а на екран виводиться число файлів, які проглянуто та результати порівняння.
5. Напишіть Bash скрипт-файл, який виконає пошук у заданому каталозі та усіх його підкаталогах усіх файлів, власником яких є вказаний користувач. Ім'я власника та каталог задаються користувачем як перший та другий аргумент командного рядка. Скрипт виводить результати в файл (третій аргумент командного рядка) у вигляді "повний шлях, ім'я файлу, його розмір". На екран виводиться загальне число файлів, що було проглянуто.
6. Напишіть Bash скрипт-файл, який виконає пошук у заданому каталозі та усіх його підкаталогах усі файли заданого розміру, які належать певному користувачу. Діапазон розмірів файлів (min - max) задається користувачем як перший та другий аргумент командного рядку. Ім'я власника та каталог задаються користувачем як третій та четвертий аргументи командного рядку. Результати пошуку повинні заноситись у файл у вигляді "повний шлях, ім'я файлу, його розмір". Ім'я файлу, який буде зберігати результати, задається як п'ятий аргумент командного рядку. На екран виводиться загальне число файлів, які було проглянуто.
7. Напишіть Bash скрипт-файл, який виконує підрахунок загального розміру файлів у заданому каталозі і усіх його підкаталогів. Ім'я каталогу задається користувачем як перший аргумент командного рядка. Результати підрахунку виводяться у файл у вигляді "каталог (повний шлях), загальний розмір файлів, кількість файлів, які було проглянуто". Ім'я файлу з результатами задається користувачем як другий аргумент командного рядку.

Одним із завдань системного адміністратора є планування завдань, які потрібно виконувати на регулярній основі. Наприклад, адміністратор може створювати та автоматизувати завдання для резервного копіювання, оновлення системи та виконання багатьох інших дій, що повторюються. Для автоматизації планування періодичних завдань часто використовується засіб **cron**.

Ознайомтесь з навчальним матеріалом щодо використання cron

[https://learning.lpi.org/uk/learning-materials/102-500/107/107.2/107.2\\_01/](https://learning.lpi.org/uk/learning-materials/102-500/107/107.2/107.2_01/)

## Завдання 2

1. Автоматичне резервне копіювання. Створіть завдання cron, яке раз на день у встановлений час виконує резервне копіювання одного каталогу (каталог обирається самостійно). Наприклад, виконується копіювання файлів з каталогу `/home/user/data` до `/home/user/backup`. До імені кожного файлу повинна додаватись дата. В `backup` також повинний бути файл, який містить дату та час копіювання, список файлів, які було скопійовано, та їх об'єм.
2. Моніторинг дискового простору. Налаштуйте завдання cron для перевірки використання дискового простору на сервері. Якщо простір перевищує певний відсоток (наприклад, 80%), cron має надіслати повідомлення на електронну пошту та записати попередження в лог-файл.
3. Очистка тимчасових файлів. Створіть cron-завдання, яке раз на тиждень очищує тимчасові файли в папці `/tmp`, видаляючи файли, старіші за певний період (наприклад, 7 днів). Інформація про видалені файли повинна заноситись до окремого лог-файлу.
4. Синхронізація системного часу. Налаштуйте cron, щоб він регулярно (наприклад, раз на годину) виконував команду синхронізації часу. До окремого файлу повинна заноситись інформація про розбіжність часу в системі з часом на серверах синхронізації. У разі розбіжності часу cron має надіслати повідомлення на електронну пошту адміністратору.
5. Автоматичне оновлення системи. Створіть завдання cron, яке щотижня у встановлений час запускає команду оновлення системи (наприклад, `apt update && apt upgrade -y` для Ubuntu/Debian). Результат оновлення повинний записуватись до файлу журналу та надсилатись на електронну пошту адміністратору.
6. Відправлення системної статистики на електронну пошту. Виконайте налаштування cron так, щоб він щодня у встановлений час (повинно бути декілька позицій) надсилав на електронну пошту адміністратора інформацію про системні ресурси: завантаженість процесора, оперативної пам'яті та використання диска. Відправлена інформація повинна дублюватись до лог-файлу.
7. Перевірка доступності мережевого ресурсу. Виконайте налаштування cron на виконання команди `ping` до певного мережевого ресурсу, наприклад, до веб-сайту або сервера. Команда повинна виконуватись постійно 1 раз на 5 хвилин. Доступність ресурсу (вдале або невдале проходження `ping`) cron має записувати в лог-файл. В разі недоступності ресурсу протягом певного періоду часу (наприклад, годину) повідомлення надсилається на електронну пошту системного адміністратора.