**оМІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**

**ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ**

**КАФЕДРА ЗАСОБІВ ЗАХИСТУ ІНФОРМАЦІЇ**

**ЗВІТ**

**з лабораторної роботи  № 3 (10)**

**з теми «Archiving and Compression (Архівація та стиск)»**

**з дисципліни «Операційні системи та технології їх захисту»**

Виконав:

студент СЗ-481

Дрик Владислав Олександрович

«24» Жовтня 2023 р.

**КИЇВ 2023**

**Мета роботи:** навчитися працювати с архівними файлами операційної

системи Linux.

**Завдання лабораторної роботи:**

* Створення архівних файлів з використанням tar з стиском та без нього.
* Стиск та розпакування файлів в файл архіву gzip
* Стиск та розпакування файлів в архівний файл bzip2
* Стиск та розпакування файлів в файл архіву xz
* Використовувати zip и unzip для стиску та розпакування архівних

файлів

**Хід роботи**

**10.2 Канали командного рядка і перенаправлення**

Зазвичай, коли ви виконуєте команду, висновок відображається у вікні терміналу. Цей штифт (його ще називають каналом) називається стандартним штифтом і позначається терміном stdout. Номер дескриптора файлу для цього каналу - 1.

Стандартна помилка (stderr) виникає, коли помилка з'являється під час виконання команди; Він має дескриптор файлу 2. Повідомлення про помилки також надсилаються у вікно терміналу за замовчуванням.

У цій лабораторній ви будете використовувати символи, які перенаправляють вихід зі стандартного виводу (stdout) і стандартної помилки (stderr) у файл або на іншу команду замість екрана терміналу.

Стандартний вхід, stdin, зазвичай надається вами при введенні тексту на клавіатурі; Він має дескриптор файлу 0. Однак, перенаправляючи стандартні вхідні дані, файли також можна використовувати як stdin.

10.2.1 Крок 1

Використовуйте символ переспрямування > разом із командою echo, щоб перенаправити вихід зі звичайного виводу stdout (на термінал) у файл. Команду cat можна використовувати для відображення вмісту файлу і в цьому прикладі буде використано для перевірки перенаправлення виводу на файл.

A screenshot of a computer

Description automatically generated

10.2.2 Крок 2

Коли ви використовуєте символ > для перенаправлення stdout, вміст файлу знищується в першу чергу. Введіть наступні команди, щоб побачити демонстрацію:

A screen shot of a computer

Description automatically generated

10.2.3 Крок 3

Можна уникнути перезапису файлу, використовуючи >> замість >. За допомогою >> ви додаєте в файл. Виконайте такі команди, щоб побачити це:

**A screenshot of a computer program

Description automatically generated**

10.2.4 Крок 4

Команда find є хорошою командою для демонстрації того, як працює stderr. Ця дуже гнучка команда дозволяє здійснювати пошук за різними параметрами, такими як ім'я файлу, розмір, дата, тип і роздільна здатність. Команда find здійснить пошук у вказаному каталозі та рекурсивний пошук у всіх підкаталогах.

Команда find здійснить пошук у вказаному каталозі та рекурсивний пошук у всіх підкаталогах. Наприклад, щоб знайти файли, які починаються з домашнього каталогу і містять ім'я bash:

A computer screen with text on it

Description automatically generated

A computer screen with white text

Description automatically generated

10.2.5 Крок 5

Щоб переспрямувати stderr (повідомлення про помилки) до файлу, введіть таку команду:

A screenshot of a computer

Description automatically generated

10.2.6 Крок 6

Ви також можете перенаправити stdout і stderr на два окремих файли.

A computer screen shot of a black screen

Description automatically generated

10.2.7 Крок 7

Щоб перенаправити стандартний вихід (stdout) і стандартну помилку (stderr) на той самий файл, спочатку переспрямуйте stdout на файл, а потім перенаправте stderr на той самий файл за допомогою нотації 2> &1.

A computer screen shot of a black screen

Description automatically generated

10.2.8 Крок 8

Стандартний вхід (stdin) також може бути перенаправлений. Зазвичай stdin поставляється з клавіатури, але іноді потрібно, щоб він походив з файлу. Наприклад, команда tr переводить символи, але приймає дані тільки зі стандартного вводу, а не з імені файлу, вказаного як аргумент. Це чудово, якщо ви хочете зробити щось на зразок великої літери в даних, які ви вводите з клавіатури (примітка: натисніть Control + d, щоб подати сигнал команді tr про припинення обробки стандартного введення):

A computer screen with text on it

Description automatically generated

10.2.9 Крок 9

Команда tr приймає введення з клавіатури (stdin), переводить символи, а потім перенаправляє вихід на stdout. Щоб створити файл, який містить усі символи нижнього регістру, виконайте такі дії:

A screenshot of a computer

Description automatically generated

10.2.10 Крок 10

Виконайте наступні команди, щоб використовувати команду tr, перенаправляючи stdin з файлу:

A computer screen with text

Description automatically generated

10.2.11 Крок 11

Іншою популярною формою переспрямування є отримання результатів однієї команди та надсилання її як вхідні дані іншій команді. Наприклад, результат деяких команд може бути величезним, через що вихідні дані занадто швидко прокручуються з екрана для читання. Запустіть наступну команду, щоб отримати результат команди ls і надіслати її команді more, яка відображає одну сторінку даних за раз:

A screenshot of a computer

Description automatically generated

10.2.12 Крок 12

У наступному прикладі ви будете використовувати команду cut, щоб отримати всі імена користувачів з бази даних з іменем /etc/passwd (файл, який містить інформацію про обліковий запис користувача).

Спочатку спробуйте запустити команду cut окремо:

A screen shot of a computer

Description automatically generated

10.2.13 Крок 13

Вихід в попередньому прикладі вийшов з ладу і прокрутився з екрану. На наступному кроці ви збираєтеся вивести команду cut і надішліть його команді sort, надати певний порядок виведення коштів:

A screenshot of a computer

Description automatically generated

10.2.14 Крок 14

Вихід тепер відсортовано, але він все ще прокручується з екрана. Надішліть результат команди sort команда more, щоб вирішити цю проблему:

A screenshot of a computer

Description automatically generatedA black background with white text

Description automatically generated

**10.3 Перегляд великих текстових файлів**

Хоча великі текстові файли можна переглядати за допомогою кнопки cat, прокручувати назад до початку файлу незручно. Крім того, дійсно великі файли не можуть відображатися таким чином, оскільки вікно терміналу зберігає в пам'яті лише певну кількість вихідних рядків.

Використання more або less команда дозволяє користувачеві переглядати дані «сторінка» або рядок за раз. Ці команди «пейджера» також дозволяють використовувати інші форми навігації та пошуку, які будуть продемонстровані в цьому розділі.

10.3.1 Крок 1

/ Etc / passwd, Можливо, занадто великий, щоб відображатися на екрані без прокручування екрана. Щоб переконатися в цьому, скористайтеся командою cat для відображення всього вмісту файлу / etc / passwdfile:

A computer screen shot of a computer program

Description automatically generated

10.3.2 Крок 2

Використовуйте команду more, Відображення всього вмісту файлу / etc /passwd:

A computer screen shot of a computer

Description automatically generated

10.3.3 Крок 3

Поки ви в команді more, Екран довідки можна переглянути, натиснувши

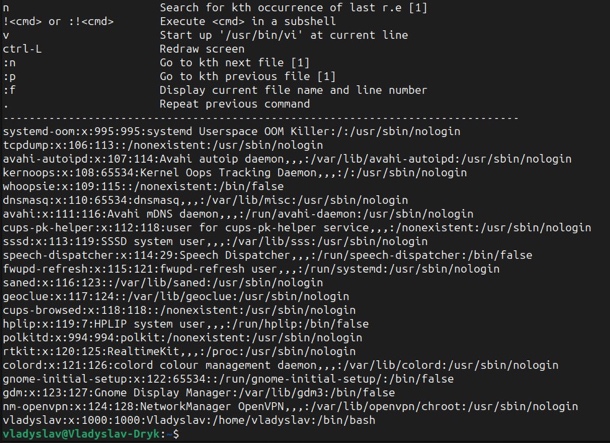
кнопку h:

A screenshot of a computer program

Description automatically generated

10.3.4 Крок 4

Натисніть клавішу пробілу, щоб переглянути решту документа:



10.3.5 Крок 5

Використовуйте команду less, Відображення всього вмісту файлу / etc / passwd. Потім знайдіть слово bin, використання n, рухатися вперед, і N, , щоб повернутися назад. Нарешті, вийдіть із пейджера less, набравши букву q:

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

10.3.6 Крок 6

Можна скористатися командою head , щоб відобразити верхню частинуфайлу. За замовчуванням head Відображає перші десять рядків файлу:

A computer screen with white text

Description automatically generated

10.3.7 Крок 7

Використовуйте команду tail, Відображення останніх десяти рядків файла /etc / passwd:

A computer screen with white text

Description automatically generated

10.3.8 Крок 8

Використовуйте команду head Відображення перших двох рядків файлу /etc / passwd:

A screenshot of a computer code

Description automatically generated

10.3.9 Крок 9

Запустіть наступний командний рядок, щоб передати результат команди ls команда tail, відображення останніх п'яти імен файлів у каталозі / etc:

A black screen with white text

Description automatically generated

10.3.10 Крок 10

Інший спосіб указати, скільки рядків потрібно надрукувати за допомогоюкнопки head, - використовуйте параметр –n -#, де # - Це кількість рядків ізнижньої частини виводу, яку потрібно виключити. Зверніть увагу насимвол мінус - до #. Наприклад, якщо / etc / passwd містить 27 рядків, наступна команда відобразить рядки 1–7, за винятком останніх двадцяти рядків:

A computer screen shot of a black screen

Description automatically generated

**10.4 Пошук тексту за допомогою регулярних виразів**

У цій роботі ви будете використовувати сімейство команд grep з регулярними виразами для пошуку певного рядка символів у потоці даних (наприклад, текстовому файлі).

Команда grep Використовує основні регулярні вирази, спеціальні символи, такі як символи узагальнення, які відповідають шаблонам даних. Команда grep Повертає весь рядок, який містить шаблон зіставлення.

Параметр -E команди grep може використовуватися для виконання пошуку з використанням розширених регулярних виразів, по суті більш потужних регулярних виразів. Іншим способом використання розширених регулярних виразів є використання команди egrep.

Команда fgrep Використовується для зіставлення буквальних символів, ігноруючи особливе значення символів регулярних виразів.

10.4.1 Крок 1

Використання grep У найпростішій формі це пошук заданого рядка символів, наприклад sshd, в файлі / etc / passwd. Команда grep друкує весь рядок, що містить збіг:

A screenshot of a computer program

Description automatically generated

**Не має збігів**

10.4.2 Крок 2

Регулярні вирази є «жадібними» в тому сенсі, що вони відповідають кожному екземпляру зазначеного шаблону:

A black screen with white text

Description automatically generated

10.4.3 Крок 3

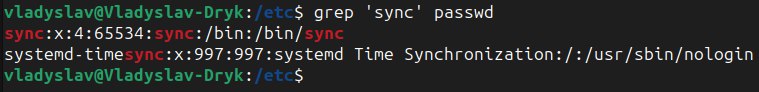
Щоб обмежити вивід, ви можете використовувати регулярні вирази для визначення більш точного шаблону. Наприклад, символ каретки ^ можна використовувати для зіставлення візерунка на початку рядка; Тому під час запуску наступного командного рядка використовуються лише рядки, які починаються з root:

A screen shot of a computer

Description automatically generated

10.4.4 Крок 4

Відповідність шаблону sync в будь-якій точці рядку:



10.4.5 Крок 5

За допомогою символу $, можна співставити sync з шаблоном в кінці рядка:

A computer screen with text

Description automatically generated

10.4.6 Крок 6

Використовуйте символ крапки "." для зіставлення будь-якого окремого символу. Наприклад, запустіть таку команду, щоб відповідати будь-якому символу, за яким слідує 'y':

A screen shot of a computer

Description automatically generated

10.4.7 Крок 7

Вертикальний знак смуги | або «оператор чергування» виступає в ролі оператора «або». Наприклад, виконайте такі дії, щоб спробувати зіставити sshd, root або operator:

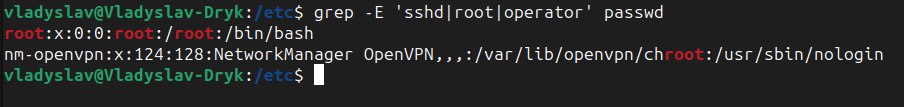


**Не має збігів**

10.4.8 Крок 8

Використання перемикача -E, Щоб дозволити grep Робота в розширеному

режимі для розпізнавання оператора черезрядкової розгортки:



10.4.9 Крок 9

Використовуйте інший розширений регулярний вираз, цього разу з egrep чергуються в групі, щоб відповідати візерунку. Лінія nob і non буде відповідати:

A screen shot of a computer

Description automatically generated

10.4.10 Крок 10

Символи [] також можна використовувати для зіставлення одного символу. Однак, на відміну від символу крапки ".", символи [] використовуються для позначення того, який саме символ потрібно зіставити. Наприклад, якщо потрібно зіставити числовий символ, можна вказати [0-9]. Запустіть таку команду, щоб продемонструвати:

A screen shot of a computer

Description automatically generated

10.4.11 Крок11

Припустимо, ви хочете знайти шаблон, який містить послідовність з трьох цифр. Ви можете використовувати символи {} з числом, щоб висловити бажання повторити шаблон певну кількість разів; Наприклад: {3}. Використання числового обмежувача вимагає розширеного режиму grep:

A computer screen shot of a program

Description automatically generated

ВИСНОВОК Виконавши лабораторну роботу, я навчився перенаправляти текстові потоки, використовувати регулярні вирази та команди для фільтрації тексту.

У цій лабораторній я виконав наступні завдання:

* Дізнайвся, як переспрямовувати стандартні канали вводу, виводу та

помилок.

* Використав регулярні вирази для фільтрації виводу команд або вмісту файлу.
* Переглянув великі файли або виводите команди за допомогою

програмного забезпечення для нумерації сторінок і вибрані частини.