“Київський фаховий коледж зв’язку”

Циклова комісія Комп’ютерної інженерії

**ЗВІТ ПО ВИКОНАННЮ**

**ЛАБОРАТОРНОЇ РОБОТИ № 6**

з дисципліни: «Операційні системи»

Тема: “ Команди Linux для архівування та стиснення даних. Робота з текстом ”

Виконали студенти

групи БІКС-13

Литвинюк Євген

Меліхов Данило

Бумажний Микола

Перевірив викладач

Сушанова В.С.

Київ 2024

**Робота студентів групи БІКС-13 Команда OCG:Литвинюк Є. Меліхов Д. Бумажний М.**

**Мета роботи:**

1. Отримання практичних навиків роботи з командною оболонкою Bash.
2. Знайомство з базовими командами для архівування та стиснення даних.
3. Знайомство з базовими діями при роботі з текстом у терміналі.

**Матеріальне забезпечення занять:**

1. ЕОМ типу IBM PC.

2. ОС сімейства Windows та віртуальна машина Virtual Box (Oracle).

3. ОС GNU/Linux (будь-який дистрибутив).

4. Сайт мережевої академії Cisco netacad.com та його онлайн курси по Linux

**Завдання для попередньої підготовки:**

**Виконав Литвинюк Євген**

1. \*Прочитайте короткі теоретичні відомості до лабораторної роботи та зробіть невеликий словник базових англійських термінів з питань призначення команд та їх параметрів.
2. Вивчіть матеріали онлайн-курсу академії Cisco “NDG Linux Essentials”:

* Chapter 09 - Archiving and Compression
* Chapter 10 - Working With Text

1. Пройдіть тестування у курсі NDG Linux Essentials за такими темами:

* Chapter 09 Exam
* Midterm Exam (Modules 1 - 9) ***буде окреме завдання в гугл-класі***
* Chapter 10 Exam

1. На базі розглянутого матеріалу дайте відповіді на наступні питання:
   1. \*Яке призначення команд *tar*, *xz*, *zip*, *bzip*, *gzip*? Зробіть короткий опис кожної

команди та виділіть їх основні параметри. Яким чином їх можна встановити.

tar: Команда для архівування файлів і каталогів в один архів. Основний параметр - **-c** для створення архіву. Встановлюється зазвичай разом з основними дистрибутивами Linux.

xz: Команда для стиснення та розархівування файлів у форматі .xz. Основний параметр - **-z** для стиснення файлу. Встановлюється окремо через менеджер пакунків вашої операційної системи.

zip: Команда для архівування та стиснення файлів у форматі .zip. Основні параметри - **-r** для рекурсивного архівування каталогів, -9 для максимального стиснення. Встановлюється окремо через менеджер пакунків вашої операційної системи.

bzip2: Команда для стиснення та розархівування файлів у форматі .bz2. Основний параметр - **-z** для стиснення файлу. Встановлюється окремо через менеджер пакунків вашої операційної системи.

gzip: Команда для стиснення та розархівування файлів у форматі .gz. Основний параметр - **-9** для максимального стиснення. Встановлюється разом з основними дистрибутивами Linux.

* 1. \*\*Наведіть три приклади реалізації архівування та стискання даних різними командами.

1. Архівування та стискання з використанням tar та gzip:

«tar -czvf archive.tar.gz directory»

2. Архівування та стискання з використанням zip:

«zip -r archive.zip directory»

3. Стискання файлу з використанням bzip2:

«bzip2 filename»

* 1. \*Яке призначення команд cat, less, more, head and tail? Зробіть короткий опис кожної команди та виділіть їх основні параметри. Яким чином їх можна встановити.

**cat**: Команда для виведення вмісту файлу у консоль. Основний параметр - вказання файлу. Встановлюється разом з основними дистрибутивами Linux.

**less**: Програма для перегляду вмісту файлів у консольному режимі. Основні параметри - натискання клавіш, таких як стрілки та пробіл, для навігації. Встановлюється разом з основними дистрибутивами Linux.

**more**: Програма для виведення текстових файлів поступово, по сторінках, дозволяючи прокручувати текст. Основний параметр - натискання клавіш, таких як Enter, для переходу до наступної сторінки. Встановлюється разом з основними дистрибутивами Linux.

**head**: Команда для виведення перших рядків файлу. Основний параметр - -n для вказання кількості рядків. Встановлюється разом з основними дистрибутивами Linux.

**tail**: Команда для виведення останніх рядків файлу. Основний параметр - -n для вказання кількості рядків. Встановлюється разом з основними дистрибутивами Linux.

* 1. \*\*Поясніть принципи роботи командної оболонки з каналами, потоками та фільтрами

Канали (Pipes):

Канали дозволяють сполучати вивід однієї команди з вхідним потоком іншої команди без запису результатів на диск.

Синтаксис використання каналів полягає у використанні вертикальної риски | між командами.

Потоки (Streams):

У Linux є три основних потоки: стандартний ввід (stdin), стандартний вивід (stdout) та стандартна помилка (stderr).

Потоки дозволяють передавати дані між процесами і файлами.

Командна оболонка може перенаправляти потоки вводу та виводу за допомогою спеціальних операторів: < для введення, > для виведення, >> для додавання в кінець файлу, 2> для виведення помилок тощо.

Фільтри:

Фільтри - це програми, які приймають вхідні дані, обробляють їх і видають результат.

Часто вони використовуються разом з каналами для обробки потоків даних.

Найпоширеніші фільтри включають grep, sed, awk, sort, uniq тощо.

* 1. \*Яке призначення команди grep?

Команда grep призначена для пошуку рядків в текстових файлах або виводі інших команд, які відповідають певному шаблону (регулярному виразу). Основне використання grep полягає у фільтрації великих об'ємів текстової інформації за певними критеріями. Наприклад:

**«grep 'pattern' filename»**

Ця команда знайде всі рядки у файлі filename, які містять 'pattern'. Можна також вказати вхідні дані з іншої команди через канал:

**«some\_command | grep 'pattern'»**

Це широко використовується в скриптах і командному рядку для пошуку конкретної інформації в великих об'ємах тексту.

1. Підготувати в електронному вигляді початковий варіант звіту:

* Титульний аркуш, тема та мета роботи
* Словник термінів
* Відповіді на п.4.1 та п.4.5 з завдань для попередньої підготовки

**Хід роботи:**

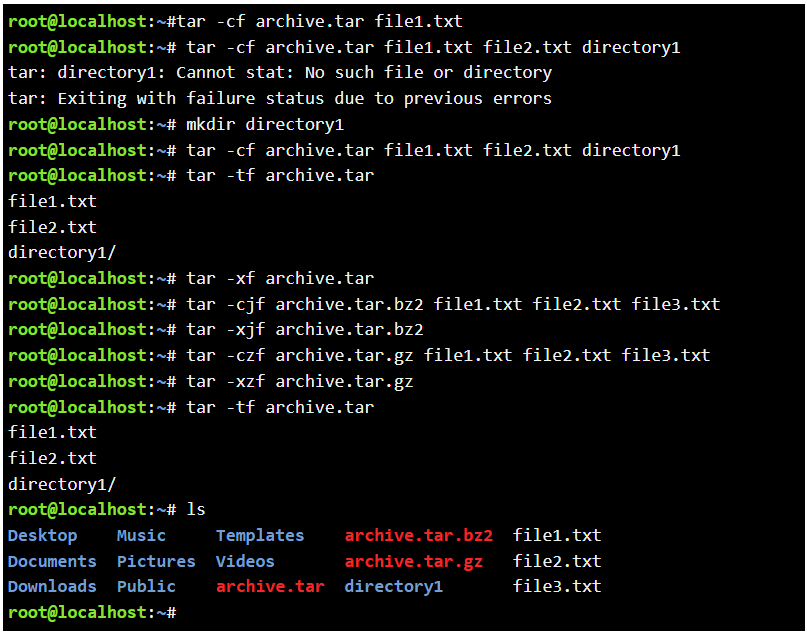
* 1. Початкова робота в CLI-режимі в Linux ОС сімейства Linux:
  2. Запустіть віртуальну машину VirtualBox, оберіть CentOS та запустіть її. Виконайте вхід в систему під користувачем: CentOS, пароль для входу: reverse ***(якщо виконуєте ЛР у 401 ауд.)*** та запустіть термінал.
  3. Запустіть віртуальну машину Ubuntu\_PC ***(якщо виконуєте завдання ЛР через академію netacad)***
  4. Запустіть свою операційну систему сімейства Linux ***(якщо працюєте на власному ПК та її встановили)*** та запустіть термінал.
  5. Опрацюйте всі приклади команд, що представлені у лабораторних роботах курсу ***NDG Linux Essentials - Lab 9: Archiving and Compression*** та ***Lab 10: Working With Text.*** Створіть таблицю для опису цих команд

|  |  |
| --- | --- |
| Назва команди | Її призначення та функціональність |
| mkdir mybackups | Створення нової директорії **mybackups** у домашньому каталозі користувача |
| tar -cvf mybackups/udev.tar /etc/udev | Команда **tar** використовується для об’єднання кількох файлів в один файл. В даному випадку вміст директорії **/etc/udev** буде збережено в архів **udev.tar** у директорії **mybackups**. Параметр **-c** повідомляє команді tar створити файл tar. Параметр **-v** означає "verbose", який наказує команді tar продемонструвати, що вона робить. Параметр **-f** використовується для вказівки назви файлу tar. |
| tar -xvf udev.tar.gz | Розпаковує архівний файл **‘udev.tar.gz’**, розгортаючи його збережені файли та папки. Опції **‘-x’** вказує на розпакування, **‘-v’** на виведення докладних повідомлень під час роботи, ‘-**f’** вказує ім'я архіву, яке йому слід використовувати. |
| ls etc/udev/rules.d | Відображає список файлів та папок, що знаходяться у папці **‘etc/udev/rules.d.’** |
| tar -rvf udev.tar /etc/hosts | Додає файл **/etc/hosts** до існуючого архіву udev.tar. Опція **-r** вказує на додавання файлу до архіву, **-v** на докладне відображення процесу, **-f** вказує ім'я архіву. |

* 1. Ознайомтесь з командою tar та за її допомогою виконати у терміналі наступні дії:

**Виконав Меліхов Данило**

* створити файл з розширенням .tar;
* створити файл з розширенням .tar, що складається з декількох файлів і каталогів одночасно;
* перегляду вмісту файлу;
* витягти вміст файлу tar;
* створити архівний файл tar, стиснений за допомогою bzip;
* витягти вміст файлу tar bzip;
* створити архівний tar файл, стисненого за допомогою gzip;
* витягти вміст файлу tar gzip.



* 1. \*Як буде відбуватись перенаправлення потоків виведення в bash для наступних дій з командами (позначено як cmd) та файлами (позначено як file):

|  |  |
| --- | --- |
| **Команда** | **Що виконує команда?** |
| cmd 1> file | Вивід stdout команди **cmd** буде перенаправлений у файл **file**.  Помилки (**stderr**) залишаться в стандартному виведенні. |
| cmd > file | Те ж саме, що і вище: вивід **stdout** буде перенаправлений у файл **file**,  а **stderr** залишиться в стандартному виведенні. |
| cmd 2> file | Помилки (stderr) команди **cmd** будуть перенаправлені у файл **file**.  Вивід stdout залишиться в стандартному виведенні. |
| cmd >> file | Вивід stdout команди **cmd** буде додано до кінця файлу **file**.  Помилки (stderr) залишаться в стандартному виведенні. |
| cmd &> file | Якщо доступна, stdout та stderr команди **cmd** будуть перенаправлені у файл **file**.  Якщо stderr недоступний, ця команда еквівалентна **cmd > file**. |
| cmd > file 2>&1 | Якщо доступний, stdout команди **cmd** буде перенаправлений у файл **file**,  а stderr буде перенаправлений туди ж. Тобто обидва виведення ведуть на  одне й те ж місце. |
| cmd >> file 2>&1 | Аналогічно до попереднього, але вивід stdout буде додано до кінця файлу **file**,  а stderr також буде перенаправлений туди ж. |
| cmd 2>&1 > /dev/null | stdout команди **cmd** буде перенаправлений у **/dev/null** (тобто ігнорується), а stderr буде перенаправлений туди ж, де і stdout. Тобто обидва виведення також ігноруються. |
| cmd 2> /dev/null | stderr команди **cmd** буде перенаправлений у **/dev/null** (тобто ігнорується),  а stdout залишиться в стандартному виведенні. |
| cmd1 | cmd2 | stdout команди **cmd1** буде передано як ввід для команди **cmd2**,  а stderr залишиться в стандартному виведенні. |
| cmd1 2>&1 | cmd2 | stdout та stderr команди **cmd1** будуть передані як ввід для команди **cmd2**. |

* 1. \*\*Розгляньте наведені нижче приклади та поясніть, що виконують дані команди та який тип перенаправлення потоків вони використовують:

|  |  |  |
| --- | --- | --- |
| Команда  (контейнер команд) | Що виконує команда? | Який потік перенаправлення? |
| $echo "It is a new story." > story | Ця команда записує рядок "It is a new story." у файл з ім'ям story. | Вона використовує перенаправлення stdout (>) для виведення вмісту команди у файл. |
| $ date > date.txt | Ця команда записує поточну дату та час у файл з ім'ям **date.txt**. | Вона також використовує перенаправлення stdout для виведення вмісту команди у файл. |
| $ cat file1 file2 file3 > bigfile | Ця команда об'єднує вміст файлів **file1**, **file2** та **file3** в один великий файл з ім'ям **bigfile**. | Вона використовує перенаправлення stdout для запису вмісту великого файлу. |
| $ls -l >> directory | Ця команда додає вивід команди **ls -l** (список файлів та папок з детальною інформацією) в кінець файлу **directory**. | Вона використовує перенаправлення stdout (**>>**) для додавання вмісту у файл. |
| $ sort < file1\_unsorted > file2\_sorted | Ця команда сортує вміст файлу **file1\_unsorted** та записує відсортований вміст у файл **file2\_sorted**. | Вона використовує перенаправлення stdin (**<**) для введення даних та перенаправлення stdout для виведення результату в файл. |
| $ find -name '\*.txt' > file.txt 2> /dev/null | Ця команда знаходить всі файли з розширенням **.txt** та записує їх імена у файл **file.txt**. | При цьому вона використовує перенаправлення stdout для виведення результату, а також перенаправлення stderr (**2> /dev/null**), щоб ігнорувати всі можливі помилки. |
| $ cat file1\_unsorted | sort > file2\_sorted | Ця команда бере вміст файлу **file1\_unsorted**, сортує його та записує відсортований вміст у файл **file2\_sorted**. | Вона використовує канал для передачі вмісту однієї команди у другу. |
| $ cat myfile | grep student | wc -l | Ця команда бере вміст файлу **myfile**, фільтрує рядки, що містять слово "student" за допомогою **grep.** | використовує **wc -l** для підрахунку кількості відфільтрованих рядків. |