“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»

Виконала

студентка

групи БІКС-13

Лось Злата Володимирівна

Перевірив викладач

Сушанова В.С.

Київ 2024

**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:**

1. \*Прочитайте короткі теоретичні відомості до лабораторної роботи та зробіть невеликий словник базових англійських термінів з питань призначення команд та їх параметрів.

Lossy and Lossless Compression - у той час як lossless compression зберігає вихідні дані без втрат, lossy compression забезпечує вищий ступінь стиснення за рахунок втрати частини інформації.

gzip (скорочення від GNU zip) — утиліта стиснення і відновлення (декомпресії) файлів, що використовує алгоритм Deflate.

bzip2 — безкоштовна вільна утиліта командного рядка (а також алгоритм) з відкритим початковим кодом для стиснення даних.

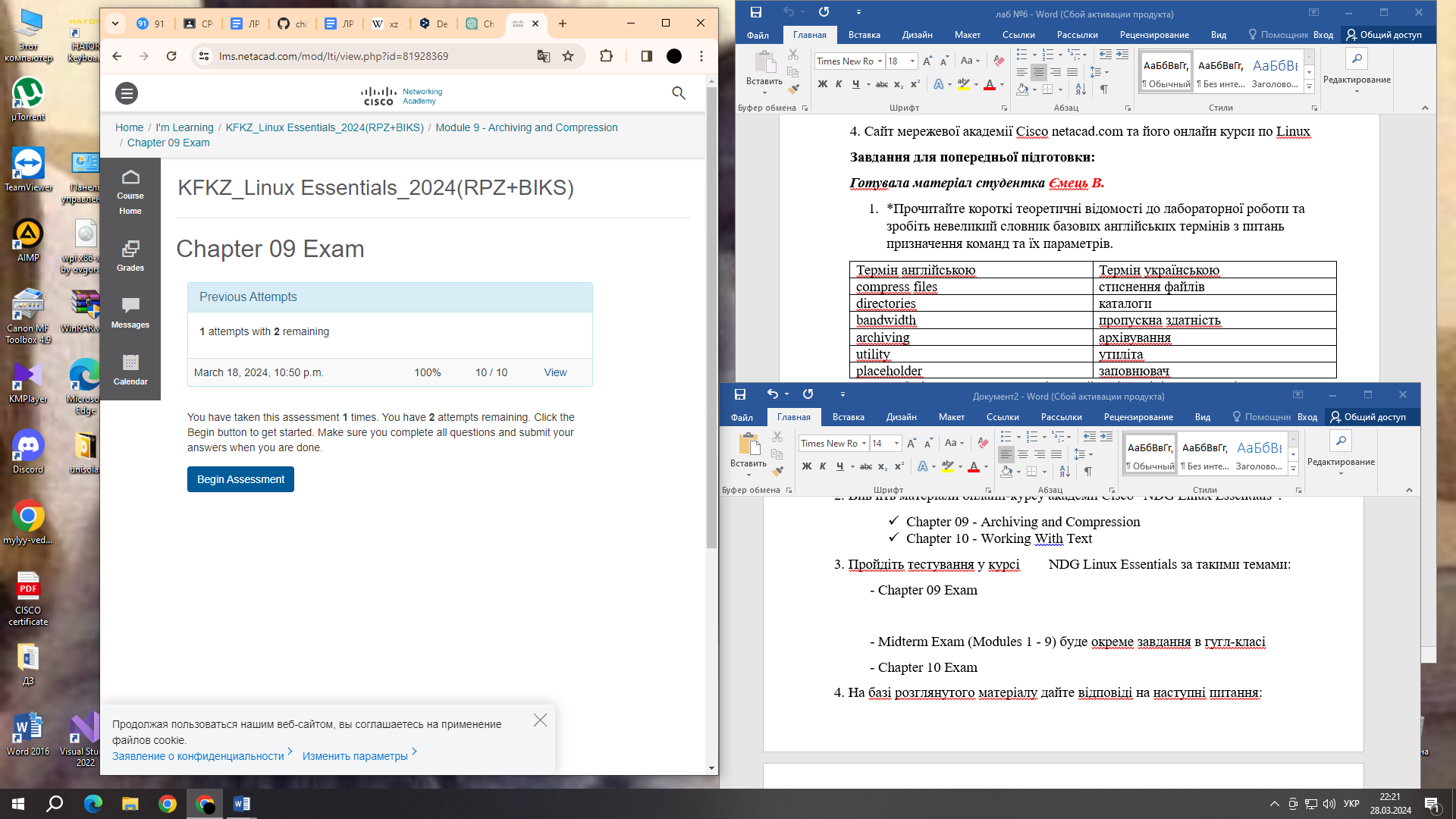
xz — комп'ютерна програма стиснення без втрат, що використовує алгоритм LZMA2, а також відповідний формат файлу.

2. Вивчіть матеріали онлайн-курсу академії Cisco “NDG Linux Essentials”:

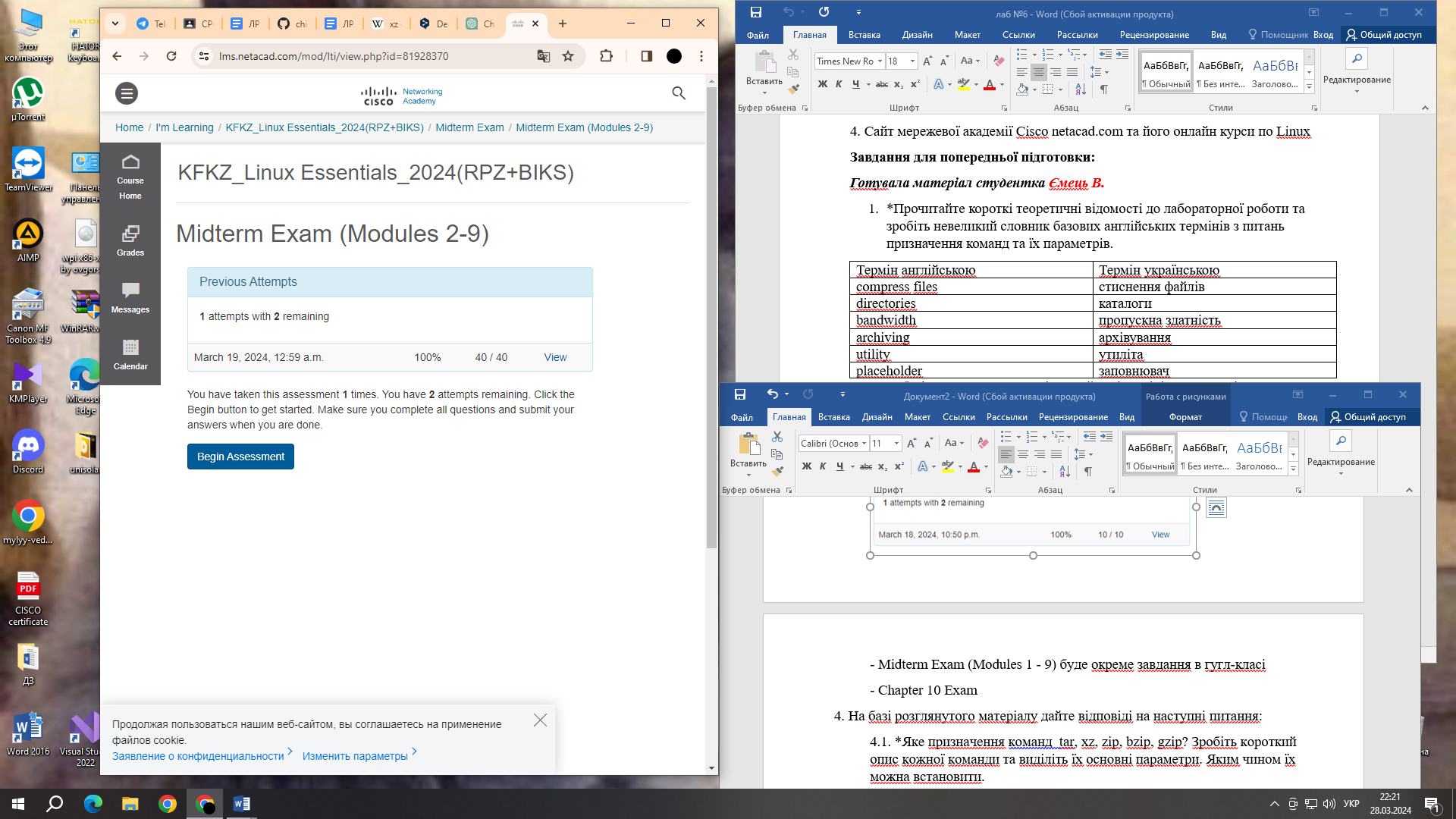
* Chapter 09 - Archiving and Compression
* Chapter 10 - Working With Text

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

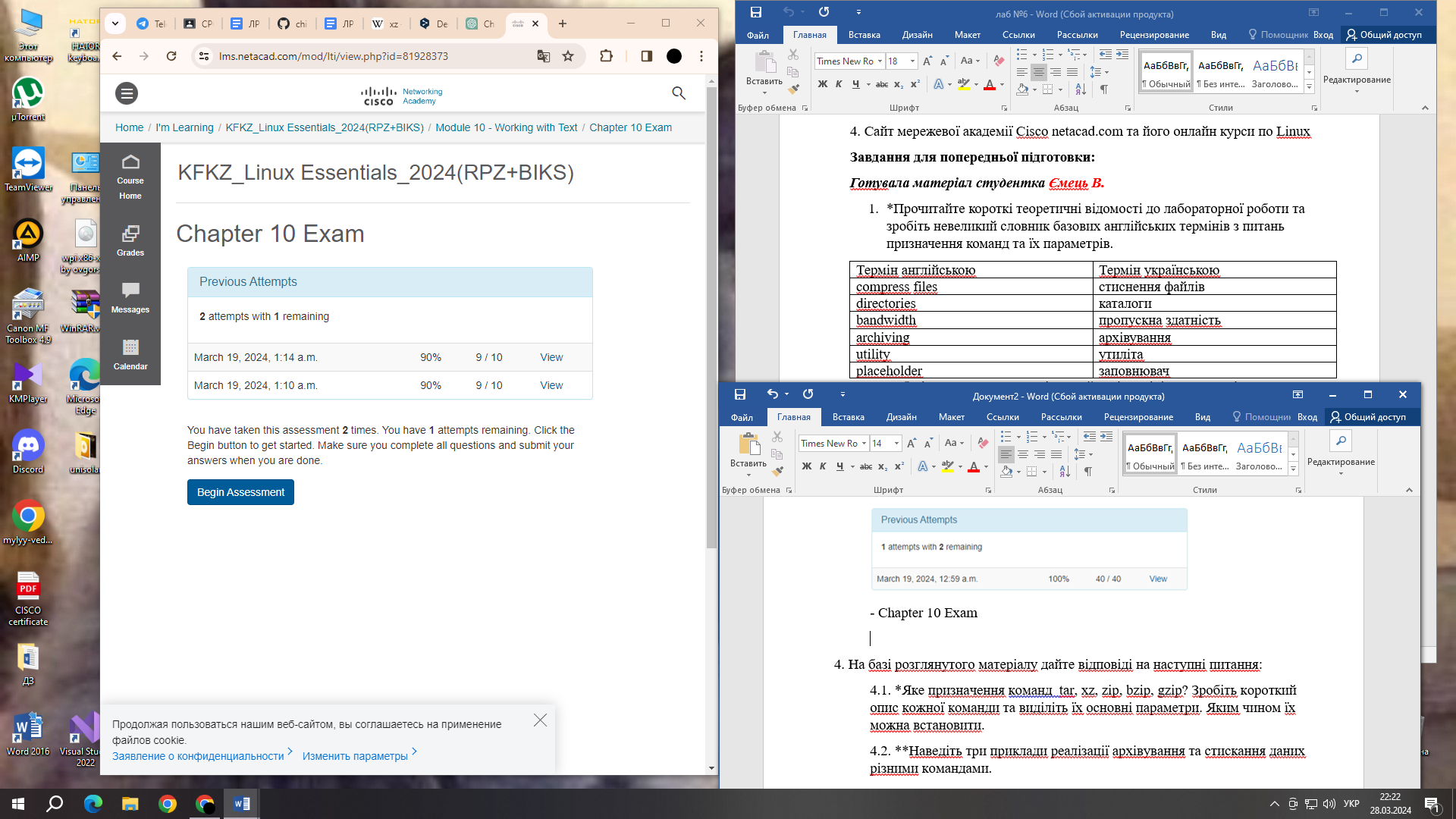
- Chapter 09 Exam



- Midterm Exam (Modules 1 - 9) буде окреме завдання в гугл-класі



- Chapter 10 Exam



4. На базі розглянутого матеріалу дайте відповіді на наступні питання:

4.1. \*Яке призначення команд tar, xz, zip, bzip, gzip? Зробіть короткий опис кожної команди та виділіть їх основні параметри. Яким чином їх можна встановити.

1) tar: Команда tar (Tape ARchive) використовується для створення архівних файлів та пакування директорій у єдиний файл. Основні параметри:

* -c: Створити новий архів
* -x: Розпакувати архів
* -f: Вказати ім'я архіву

2) xz: Команда xz використовується для стиснення файлів. Вона надає дуже ефективне стиснення, але може бути трохи повільною у порівнянні з іншими методами. Основний параметр:

* -z: Стиснути файл

3) zip: Команда zip створює архіви з файлів. Вона є однією з найпоширеніших у Windows, але також доступна у багатьох UNIX-подібних операційних системах. Основні параметри:

* -r: Рекурсивно додати файли та папки
* -q: Тихий режим (без виведення повідомлень)

4) bzip2: Команда bzip2 використовується для стиснення файлів. Вона надає високу ступінь стиснення, але може бути трохи повільною. Основний параметр:

* -z: Стиснути файл

5) gzip: Команда gzip також використовується для стиснення файлів. Вона є дуже швидкою та забезпечує хороший рівень стиснення. Основний параметр:

* -r: Стиснути рекурсивно

У більшості UNIX-подібних операційних систем, ці утиліти встановлені за замовчуванням. Проте, якщо вони відсутні, ви можете встановити їх, скориставшись менеджером пакетів вашої операційної системи. Наприклад, на Debian або Ubuntu це може бути зроблено за допомогою команди ‘sudo apt install <назва\_пакету>’, де ‘<назва\_пакету>’ - це назва пакету, який ви хочете встановити (наприклад, tar, zip, gzip, тощо).

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

1) Використання ‘tar’ та ‘gzip’ для створення архіву та його стиснення:

Спочатку створимо архів за допомогою ‘tar’, який пакує директорію ‘Documents’ у файл ‘documents.tar’: ‘tar -cf documents.tar Documents/’

Тепер застосуємо ‘gzip’, щоб стиснути архів ‘documents.tar’:

‘gzip documents.tar’

Після виконання цих команд у вас буде створений стиснутий архів ‘documents.tar.gz.’

2) Використання ‘zip’ для створення архіву та його стиснення:

Створимо архів ‘documents.zip’, що містить директорію ‘Documents’:

‘zip -r documents.zip Documents/’

З цієї команди ‘zip’ створить архів ‘documents.zip’ з усіма файлами та папками, що містяться в директорії ‘Documents’.

3) Використання ‘tar’ та ‘xz’ для створення архіву та його стиснення:

Спочатку створимо архів за допомогою ‘tar’, який пакує директорію ‘Pictures’ у файл ‘pictures.tar’: ‘tar -cf pictures.tar Pictures/’

Тепер застосуємо ‘xz’, щоб стиснути архів ‘pictures.tar’:

‘xz pictures.tar’

Після виконання цих команд у вас буде створений стиснутий архів ‘pictures.tar.xz’.

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

1) cat: Команда cat (concatenate) використовується для відображення вмісту одного або декількох файлів у стандартний вивід. Основні параметри:

* Не має основних параметрів, але може приймати список імен файлів як аргументи.

2) less: Команда less дозволяє прокручувати текстовий вміст файлу вниз і вгору, щоб зручно переглядати його. Основні параметри:

* Відкрити файл для перегляду: less <ім'я\_файлу>
* Навігація вгору та вниз: Використовуйте клавіші курсора або клавіші прокрутки.

3) more: Команда more також використовується для поступового виведення вмісту файлу на екран, проте менш гнучка за less. Основні параметри:

* Відкрити файл для перегляду: more <ім'я\_файлу>
* Навігація вгору та вниз: Використовуйте клавішу Space для прокрутки на один екран вниз або клавішу Enter для прокрутки на один рядок вниз.

4) head: Команда head виводить перші кілька рядків з файлу. Основні параметри:

* Вивести перші N рядків: head -n N <ім'я\_файлу>

5) tail: Команда tail виводить останні кілька рядків з файлу. Основні параметри:

* Вивести останні N рядків: tail -n N <ім'я\_файлу>

Ці команди є стандартними для більшості UNIX-подібних операційних систем, тому вони, як правило, встановлені за замовчуванням. Якщо вони відсутні, ви можете встановити їх, скориставшись менеджером пакетів вашої операційної системи. Наприклад, на Debian або Ubuntu це може бути зроблено за допомогою команди ‘sudo apt install <назва\_пакету>’, де ‘<назва\_пакету>’ - це назва пакету, який ви хочете встановити (наприклад, ‘coreutils’ для команди ‘cat’, ‘less’ для ‘less’, ‘more’, ‘head’, та ‘tail’).

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

Основні принципи їх роботи:

1) Потоки вводу, виводу та помилок:

* Кожна команда в UNIX має три потоки: стандартний ввід (stdin), стандартний вивід (stdout) і стандартний потік помилок (stderr).
* Потік stdin зазвичай пов'язаний з клавіатурою, а stdout і stderr - з екраном. Проте їх можна перенаправити.

2) Канали (pipes):

* Канали (|) дозволяють перенаправляти вивід однієї команди в якісь інші команди.
* Наприклад, command1 | command2 візьме вивід від command1 і передасть його як вхід в command2.

3) Фільтри:

* Фільтри - це програми, які обробляють або модифікують вхідні дані та виводять результат.
* Такі команди, як grep, sed, awk, sort, uniq, є прикладами фільтрів.
* Вони можуть бути використані для пошуку, заміни, сортування, вилучення тощо.

4.5. \*Яке призначення команди grep?

Основне призначення команди grep - це виявлення рядків, які відповідають вказаному шаблону в текстовому вводі або в файлах. Користувач може вказати шаблон у форматі регулярного виразу для більш гнучкого та точного пошуку.

5. 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:

1.1. 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.

Запустіть віртуальну машину Ubuntu\_PC (якщо виконуєте завдання ЛР 1.2. через академію netacad)

1.3. Запустіть свою операційну систему сімейства Linux (якщо працюєте на власному ПК та її встановили) та запустіть термінал.

2. 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 | Розпакування архіву, який міститься в файлі з назвою udev.tar.gz |
| tar –tvf mybackups/udev.tar | Відображення змісту архіву udev.tar, який знаходиться у директорії mybackups |
| tar -rvf udev.tar /etc/hosts | Додавання (або оновлення) файлу /etc/hosts до існуючого архіву udev.tar |
| tar –tvf udev.tar | Відображення змісту архіву udev.tar, без його розпакування |
| ls -l words | Виведення детальної інформації про файл або директорію з ім'ям words |
| gzip words | Стиснення файлів у формат gzip, що зменшує їх розмір і зменшує обсяг місця на диску, який вони займають |
| gunzip words.gz | Розпакування файлів, стиснених у формат gzip, з відновленням їх початкового вмісту |
| zip -r udev.zip /etc/udev | Створення архівів у форматі ZIP, які містять один або кілька файлів або директорій |
| unzip -l udev.zip | Розпакування файлів, які були стиснуті в архіві формату ZIP |
| rm -r etc | Видалення файлів або директорій з файлової системи |
| cat | Виведення вмісту текстового файлу на стандартний вивід |
| find ~ -name "\*bash\*" | Пошук файлів та каталогів у файловій системі на основі різних критеріїв |
| find /etc -name hosts | Пошук файлів і каталогів відповідно до вказаних критеріїв у заданій директорії та її піддиректоріях |
| ls -l /etc | more | Відображення вмісту директорії (списку файлів та піддиректорій) або властивостей конкретних файлів, залежно від переданих аргументів |
| cut -d: -f1 /etc/passwd | Вибір та відображення певних частин рядків з введеного потоку даних або файлу |
| head | Виведення перших рядків з вказаного файлу або введеного потоку даних |
| tail /etc/passwd | Виведення останніх рядків з вказаного файлу або введеного потоку даних |
| ls /etc | tail -5 | Відображення останніх п'яти рядків виведення команди ls /etc, тобто виведення списку файлів та піддиректорій у директорії /etc, та потім відображення останніх п'яти рядків цього списку |
| grep sshd passwd | Пошук рядків тексту в одному або кількох файлах або в стандартному вводі, які відповідають вказаному шаблону |
| cd /etc | Команда cd змінює поточну робочу директорію на /etc |
| egrep 'no(b|n)' passwd | Є розширеною версією команди grep і використовується для пошуку рядків у введеному потоці тексту або у файлах, використовуючи регулярні вирази для визначення шаблонів пошуку |

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

створити файл з розширенням .tar

* tar -cf archive.tar file1 file2 file3

створити файл з розширенням .tar, що складається з декількох файлів і каталогів одночасно

* tar -cf archive.tar file1 directory1

перегляду вмісту файлу

* tar -tf archive.tar

витягти вміст файлу tar

* tar -xf archive.tar

створити архівний файл tar, стиснений за допомогою bzip

* tar -cjf archive.tar.bz2 file1 file2 directory1

витягти вміст файлу tar bzip

* tar -xjf archive.tar.bz2

створити архівний tar файл, стисненого за допомогою gzip

* tar -czf archive.tar.gz file1 file2 directory1

витягти вміст файлу tar gzip

* tar -xzf archive.tar.gz\

4. \*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 стандартного виводу команди cmd буде перенаправлений у файл file. Помилки (STDERR) залишаться на місці. |
| cmd > file | Те ж саме, що і попереднє. Скорочений варіант запису, **1** необов'язковий. |
| cmd 2> file | Вивід помилок (STDERR) команди **cmd** буде перенаправлений у файл **file**. Стандартний вивід (STDOUT) залишається на місці. |
| cmd >> file | Вивід стандартного виводу команди **cmd** буде додано у кінець файлу **file**. Помилки (STDERR) залишаться на місці. |
| cmd &> file | Якщо ви хочете перенаправити як стандартний вивід, так і помилки у файл **file**, це варіант для вас. Це еквівалентно **cmd > file 2>&1**. |
| cmd > file 2>&1 | Те ж саме, що і **cmd &> file**. |
| cmd >> file 2>&1 | Вивід (STDERR і STDOUT) команди **cmd** буде додано у кінець файлу **file**. |
| cmd 2>&1 > /dev/null | STDOUT та STDERR будуть перенаправлені до одного місця (**/dev/null**, яке відкидає вхід), тобто обидва виводи будуть втрачені. |
| cmd 2> /dev/null | Помилки (STDERR) будуть перенаправлені у **/dev/null**, тобто втрачені, а стандартний вивід залишиться на місці. |
| cmd1 | cmd2 | Вивід **cmd1** буде переданий як вхід до **cmd2**. Це так званий "конвеєр" або "pipe". |
| cmd1 2>&1 | cmd2 | 1. Вивід (STDERR і STDOUT) **cmd1** буде переданий як вхід до **cmd2**Начало формы |

5. \*\*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 використовує команду echo, яка виводить рядок тексту "It is a new story." на стандартний вивід (STDOUT). За допомогою перенаправлення > цей вивід направляється у файл з іменем story. Таким чином, цей рядок буде записаний у файл story. | Це використання перенаправлення виводу (STDOUT). Вивід команди echo ("It is a new story.") буде направлений у файл з ім'ям story. |
| $ date > date.txt | Команда date повертає поточну дату та час. За допомогою перенаправлення > ця дата та час записуються у файл з іменем date.txt. Отже, вміст файлу date.txt буде поточна дата та час. | Також використання перенаправлення виводу (STDOUT). Результат команди date (поточна дата та час) буде записаний у файл з ім'ям date.txt. |
| $ cat file1 file2 file3 > bigfile | Команда cat використовується для виведення вмісту файлів file1, file2 та file3. За допомогою перенаправлення > цей вміст буде записаний у файл bigfile. Отже, вміст файлів file1, file2 та file3 буде об'єднаний у файлі bigfile. | Це також перенаправлення виводу (STDOUT). Вміст файлів file1, file2 та file3 буде об'єднаний і записаний у файл з ім'ям bigfile. |
| $ls -l >> directory | Команда ls -l виводить список файлів у поточному каталозі разом із детальною інформацією про кожен файл. За допомогою перенаправлення >> цей вивід додається у кінець файлу з іменем directory. Отже, інформація про файли буде додана у файл directory. | Це перенаправлення виводу з дописуванням (STDOUT). Результат команди ls -l (список файлів у поточному каталозі з детальною інформацією) буде доданий у кінець файлу з ім'ям directory. |
| $ sort < file1\_unsorted > file2\_sorted | Команда sort використовується для сортування вмісту файлу file1\_unsorted. За допомогою перенаправлення <, вміст цього файлу буде використовуватися як вхід для команди sort. Результат сортування буде записаний у файл file2\_sorted. | Це використання перенаправлення вводу та виводу (STDIN та STDOUT). Вміст файлу file1\_unsorted буде використаний як вхід для команди sort, а відсортований результат буде записаний у файл file2\_sorted. |
| $ find -name '\*.txt' > file.txt 2> /dev/null | Команда find -name '\*.txt' використовується для пошуку файлів з розширенням .txt. За допомогою перенаправлення > список таких файлів буде записаний у файл file.txt. Однак, через перенаправлення 2> /dev/null, всі повідомлення про помилки будуть відкинуті та не виведені. | Це комбіноване використання перенаправлення виводу та помилок (STDOUT та STDERR). Результат команди find -name '\*.txt' (список файлів з розширенням .txt) буде записаний у файл з ім'ям file.txt, а будь-які повідомлення про помилки будуть відкинуті. |
| $ cat file1\_unsorted | sort > file2\_sorted | Ця команда використовує конвеєр (pipe). Вміст файлу file1\_unsorted передається на вхід команді sort, яка сортує його. Результат сортування буде записаний у файл file2\_sorted. | Це використання конвеєра (pipe). Вміст файлу file1\_unsorted буде переданий на вхід команді sort, результат сортування буде записаний у файл file2\_sorted. |
| $ cat myfile | grep student | wc -l | Ця команда також використовує конвеєр. Спочатку вміст файлу myfile передається на вхід команді grep student, яка шукає всі рядки, що містять слово "student". Потім результат цього пошуку подається на вхід команді wc -l, яка підраховує кількість рядків. Таким чином, ця команда підраховує кількість рядків у файлі myfile, які містять слово "student". | Це також використання конвеєра (pipe). Вміст файлу myfile буде переданий на вхід команді grep student, яка відфільтрує рядки, що містять слово "student". Результат фільтрації буде переданий на вхід команді wc -l, яка підрахує кількість рядків. |

**Контрольні запитання:**

1. Надайте порівняльну характеристику процесам стискання та архівування.

**Стискання:**

* Мета: Стискання спрямоване на зменшення обсягу файлів шляхом вилучення зайвої інформації або застосування спеціальних алгоритмів стиснення.
* Приклади алгоритмів: Gzip, Bzip2, LZMA, Deflate (використовується у форматах ZIP та gzip).
* Використання: Зазвичай використовується для стискання одного файлу, не обов'язково групи файлів, але може бути застосований до декількох файлів у випадку архівування.
* Переваги: Швидше виконання, видаляє зайву інформацію, дозволяє стиснути файл у більш маленький обсяг.
* Недоліки: Зазвичай застосовується до одного файлу, тому не дозволяє створювати структуровані архіви з багатьма файлами.

**Архівування:**

* Мета: Архівування спрямоване на створення структурованих пакетів або архівів, які можуть містити один або більше файлів та каталогів.
* Приклади форматів архівів: ZIP, TAR (та TAR.GZ, TAR.BZ2), RAR, 7z.
* Використання: Зазвичай використовується для створення архіву, що містить багато файлів та каталогів для зручності зберігання, транспортування або резервного копіювання.
* Переваги: Дозволяє об'єднати багато файлів та каталогів в один архів, зберігає структуру каталогів, може містити метадані та іншу інформацію про файли.
* Недоліки: Зазвичай потребує більше часу для створення архіву порівняно зі стисканням, але цей час може варіюватися залежно від обсягу та кількості файлів.

2. Які програми, окрім наведених в роботі, можуть використовуватись для стискання та архівування файлів та каталогів в ОС Linux? Наведіть приклади та їх короткий опис.

**XZ Utils:**

* XZ Utils надає інструменти для стискання та розпакування файлів за допомогою формату стиснення LZMA2. Він зазвичай має краще стиснення порівняно з іншими алгоритмами стискання.
* Приклад використання: xz, unxz.

**Ark:**

* Ark є інструментом для роботи з архівами у графічному режимі на основі KDE. Він підтримує різні формати архівів і надає можливості для створення, відкриття, перегляду, редагування та витягування архівів.
* Приклад використання: Доступний графічний інтерфейс користувача.

**P7zip:**

* P7zip - це реалізація програми 7-Zip для Unix-подібних операційних систем. Вона підтримує стиснення та розпакування файлів у форматі 7z, який зазвичай надає дуже хороше стиснення.
* Приклад використання: 7z, 7za, 7zr.

3. \*Порівняйте алгоритми стискання, що використовуються в командах (програмах), використовуваних в Linux. Які з алгоритмів можна вважати найшвидшим та найефективнішим?

* **Gzip (DEFLATE):** Gzip використовує алгоритм стискання DEFLATE, який є комбінацією алгоритмів LZ77 та Huffman coding. Він широко використовується через свою ефективність та швидкість стискання.
* **Bzip2:** Bzip2 використовує алгоритм стискання Burrows-Wheeler transform (BWT) у поєднанні з алгоритмом стискання move-to-front transform (MTF) та алгоритмом Huffman coding. Хоча він зазвичай надає краще стиснення, ніж Gzip, але зазвичай працює повільніше.
* **XZ (LZMA/LZMA2):** XZ використовує алгоритм стискання LZMA або його модифікацію LZMA2. Він надає високе стиснення та єдина з програм, яка зазвичай надає краще стиснення, ніж Bzip2. Однак він може бути повільнішим за Gzip або Bzip2.

4. \*Опишіть програмні засоби для стискання та архівування, що можуть бути використані у вашому мобільному телефоні.

* Функція "Архівування" (Archiving): Вбудована функція архівування файлів у iOS, яка дозволяє створювати ZIP-архіви з файлів та каталогів. Ви можете скомпонувати список файлів та каталогів, які потрібно архівувати, а потім створити ZIP-архів, щоб зберегти їх разом. Ця функція доступна в стандартному файловому менеджері на iPhone або в інших програмах, що підтримують архівування.
* Програма Files by Google: Ця програма доступна для iPhone і надає інструменти для керування файлами на пристрої. Вона також має функцію архівування, яка дозволяє створювати ZIP-архіви з файлів та каталогів прямо на вашому телефоні.

5. \*Опишіть та порівняйте програмні засоби для стискання та (де)архівування даних у ОС сімейства Windows.

**WinZip:**

* WinZip є одним з найпопулярніших програмних засобів для стискання та архівування файлів у середовищі Windows. Він підтримує різні формати архівів, включаючи ZIP, ZIPX, RAR, 7Z, TAR, і багато інших. WinZip також має можливості для захисту архівів паролем та шифрування файлів.
* Переваги: Легкий у використанні, широкий спектр підтримуваних форматів, високий рівень стиснення.
* Недоліки: Платна версія має обмеження, безкоштовна версія має обмежені функціональність та можливості.

**7-Zip:**

* 7-Zip є безкоштовним програмним забезпеченням з відкритим кодом для стискання та архівування файлів у Windows. Він підтримує різні формати архівів, включаючи 7z, ZIP, TAR, GZIP, BZIP2, і багато інших. 7-Zip має високий рівень стиснення та може працювати з файлами дуже великих розмірів.
* Переваги: Безкоштовний, відкритий код, високий рівень стиснення, великий список підтримуваних форматів.
* Недоліки: Менше інтуїтивний користувацький інтерфейс порівняно з комерційними аналогами.

**Windows Built-in ZIP Utility:**

* Операційна система Windows має вбудований інструмент для роботи з ZIP-архівами. Цей інструмент дозволяє створювати, відкривати та редагувати ZIP-архіви без необхідності встановлення додаткового програмного забезпечення.
* Переваги: Вбудований у операційну систему, не потребує додаткового встановлення, простий у використанні.
* Недоліки: Має обмежені функціональність та можливості порівняно з іншими програмами.

6. \*\*Поясніть яким чином стиснення та архівування даних може бути використано для резервування даних. В яких ще задачах системного адміністрування воно може бути використано.

Стиснення та архівування даних є важливими інструментами для забезпечення резервного копіювання даних та ефективного управління ними в системах комп'ютерного та мережевого адміністрування. Ось деякі способи, якими вони можуть бути використані:

**1. Резервне копіювання даних:**

* Стиснення та архівування даних дозволяє зменшити обсяг копій даних, що зберігаються на резервних носіях. Це забезпечує економію місця на резервних носіях і знижує витрати на їх придбання.
* Застосування стиснення також дозволяє прискорити процес резервного копіювання, оскільки менший обсяг даних займає менше часу для передачі та збереження.

**2. Передача даних через мережу:**

* При передачі великих обсягів даних через мережу стиснення та архівування даних може зменшити час передачі та обсяг використаної мережевої пропускної здатності.
* Зменшений обсяг даних також може знизити витрати на передачу даних через платні мережеві канали.

**3. Оптимізація зберігання даних:**

* Стиснення та архівування даних дозволяє оптимізувати простір зберігання на серверах та зберігаючих пристроях. Менший обсяг даних займає менше місця на диску.
* Для системного адміністрування це може бути важливо, оскільки зменшується необхідність у великих накопичувачах та збільшуються ресурси, що можуть бути використані для інших завдань.

**4. Архівування старих даних:**

* Старі дані, які вже не активно використовуються, можуть бути стиснуті та архівовані для зменшення місця, яке вони займають на серверах.
* Це дозволяє зберігати старі дані для можливого відновлення чи аудиту, не займаючи великого обсягу місця на активних системах.

**5. Захист даних:**

* Деякі програми для архівування дозволяють захистити дані паролем або шифруванням. Це допомагає зберегти конфіденційність даних, що архівуються, та захистити їх від несанкціонованого доступу.

7. \*\*Яке призначення директорії файлу /dev/null?

Основне призначення /dev/null полягає в тому, щоб надати механізм для видалення непотрібного виводу або непотрібних даних з системи, коли вони більше не потрібні. Наприклад, якщо ви хочете відправити вивід програми у "чорну дірку", щоб вона не займала дискового простору або не виводила непотрібні дані на екран, ви можете перенаправити вивід у /dev/null.

**Висновок:** Отримала практичні навички роботи з командною оболонкою Bash. Ознайомилась з базовими командами для архівування та стиснення даних. Ознайомилась з базовими діями при роботі з текстом у терміналі.