“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»**

Виконавли студенти

групи РПЗ-03

Команда: Губенко Є.О.,

Заїка С.В. та Кресан Р.А.

Перевірив викладач

Сушанова В.С.

Київ 2022

**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 (Windows 7).

3. Віртуальна машина – Virtual Box (Oracle).

4. Операційна система GNU/Linux – CentOS.

5. 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.
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

***Готував матеріал студент Заїка С.В.***

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.
2. *Команда "tar" використовується для створення та розпакування архівів у форматі tar. Основні параметри:*

*c: створити новий архів*

*x: розпакувати архів*

*v: вивести список файлів під час операції*

*f: вказати назву файлу, в який буде записано архів*

*Для встановлення tar на більшості Linux-системах виконайте команду:*



1. Команда "xz" використовується для компресії файлів у форматі xz. Основні параметри:

z: скомпресувати файл

d: розпакувати файл

Для встановлення xz на більшості Linux-системах виконайте команду:



1. Команда "zip" використовується для створення та розпакування архівів у форматі zip. Основні параметри:

r: додати файли до архіву рекурсивно

e: скомпресувати файли зі шифруванням

d: вилучити файли з архіву

Для встановлення zip на більшості Linux-системах виконайте команду:



4)Команда "bzip" використовується для компресії файлів у форматі bzip. Основні параметри:

z: скомпресувати файл

d: розпакувати файл

Для встановлення bzip на більшості Linux-системах виконайте команду:



5)Команда "gzip" використовується для компресії файлів у форматі gzip. Основні параметри:

z: скомпресувати файл

d: розпакувати файл

Для встановлення gzip на більшості Linux-системах виконайте команду:



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

1. *Створення tar-архіву та його стиснення gzip-ом:*



*Ця команда створює архів з назвою "archive.tar.gz", який містить файли та директорії, розташовані за адресою "/path/to/directory", та стискає його за допомогою gzip.*

1. *Стиснення файлу за допомогою bzip:*



*Ця команда стискає файл з назвою "file.txt" за допомогою bzip2 та зберігає його як "file.txt.bz2".*

1. *Створення zip-архіву:*



*Ця команда створює zip-архів з назвою "archive.zip", який містить файли та директорії, розташовані за адресою "/path/to/directory".*

* 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?

1. *cat: ця команда призначена для виводу вмісту файлів на екран.*

*Основний синтаксис команди cat:*



Основні параметри:

-n: виводить номери рядків;

-b: виводить номери лише для непорожніх рядків.

1. *less: ця команда призначена для виводу вмісту файлів поступово з можливістю прокручування в обидва напрямки.*

*Основний синтаксис команди less:*



*Основні параметри:*

*q: вихід з програми;*

*f: прокручування вперед на один екран;*

*b: прокручування назад на один екран;*

*/шаблон: пошук заданого шаблону в файлі.*

1. *more: ця команда також призначена для виводу вмісту файлів поступово з можливістю прокручування в обидва напрямки, але менш функціональна ніж less.*

*Основний синтаксис команди more:*

**

*Основні параметри:*

*q: вихід з програми;*

*f: прокручування вперед на один екран;*

*b: прокручування назад на один екран.*

1. *head: ця команда призначена для виводу перших декількох рядків файлу.*

*Основний синтаксис команди head:*

**

*Основні параметри:*

*-n [кількість]: виводить перші n рядків файлу.*

1. *tail: ця команда призначена для виводу останніх декількох рядків файлу.*

*Основний синтаксис команди tail:*

**

*Основні параметри:*

*-n [кількість]: виводить останні n рядків файлу.*

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

*Командна оболонка в Linux має ряд можливостей для роботи з каналами, потоками та фільтрами, що значно полегшує роботу з даними та забезпечує їх оптимальну обробку.*

*Канали (pipes) дозволяють передавати вивід однієї команди як вхідні дані для іншої команди без необхідності зберігати результат першої команди в файл. Синтаксис для каналів складається з символу вертикальної риски |, що позначає передачу виводу лівої команди як вхідні дані правій команді. Наприклад, команда ls | grep file поверне список файлів з поточного каталогу, що містять слово "file".*

*Потоки (streams) в Linux можуть бути введені з клавіатури (stdin), виведені на екран (stdout) або відправлені до файлу (stderr). Для роботи з потоками використовуються спеціальні символи. Наприклад, > символ використовується для перенаправлення виводу команди в файл, >> символ для доповнення існуючого файлу, < символ для використання файлу як вхідних даних для команди.*

*Фільтри - це команди, що зчитують вхідні дані зі стандартного вводу (stdin), оброблюють їх та виводять результат на стандартний вивід (stdout). Фільтри можуть бути поєднані з каналами та перенаправленнями для отримання потрібного результату. Деякі з найпоширеніших фільтрів в Linux включають grep для пошуку рядків у файлі, sed для редагування та заміни тексту, awk для обробки та форматування тексту.*

*Загалом, командна оболонка в Linux дозволяє легко та ефективно працювати з потоками даних, використовуючи канали та фільтри, що допомагає полегшити роботу з файлами та даними.*

* 1. Яке призначення команди grep?

*Команда grep є однією з найбільш потужних та популярних команд в Unix-подібних операційних системах, використовується для пошуку вхідних даних на відповідність певному текстовому шаблону. Команда grep часто використовується для пошуку великої кількості тексту, таких як лог-файли та інші тексти, що зберігаються у вигляді потоків даних.*

*Основна синтаксична структура команди grep виглядає наступним чином:*



*Основні параметри команди grep:*

*-i - ігнорувати регістр;*

*-v - повернути рядки, що не відповідають шаблону;*

*-n - повернути номери рядків, що містять відповідність;*

*-r - рекурсивний пошук в підкаталогах.*

*Шаблон може бути звичайним текстом або регулярним виразом для більш точного визначення шуканого тексту.*

*Наприклад, команда grep -i "error" log.txt поверне всі рядки з файлу log.txt, що містять слово "error", ігноруючи регістр. А команда grep -r "pattern" /var/log поверне всі рядки з усіх файлів у каталозі /var/log, що містять шаблон "pattern", включаючи файли в підкаталогах.*

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:
   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.
   2. Запустіть віртуальну машину Ubuntu\_PC (якщо виконуєте завдання ЛР через академію netacad)
   3. Запустіть свою операційну систему сімейства Linux (якщо працюєте на власному ПК та її встановили) та запустіть термінал.

***Готував матеріал студент Заїка С.В.***

1. 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/ude | 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 –tvf mybackups/udev.tar | Відобразити вміст tar-файлу за допомогою доступних параметрів (t = вміст списку, v = докладний, f = назва файлу): |
| cd mybackups ls tar –xvf udev.tar.gz ls  ls etc ls etc/udev ls etc/udev/rules.d | Якщо ви хочете, щоб файли «повернулися» у своє початкове розташування, ви можете спочатку перейти до каталогу /, а потім виконати команду tar. Однак у цьому прикладі вам знадобиться ввійти в систему як адміністратор, оскільки створення файлів у каталозі /etc може виконувати лише адміністратор. |
| tar -rvf udev.tar /etc/hosts tar –tvf udev.tar | Щоб додати файл до існуючого архіву, скористайтеся параметром -r команди tar. |
| ls -l words xz words ls -l words.xz | Використання xz і unxz для стиснення та розпакування файлу |
| zip words.zip words ls -l words.zip | Використовуйте команду zip, щоб стиснути файл word: |
| mv | Команда для переміщення файлів і папок. |
| chmod | Команда для зміни прав доступу до файлів і папок. |
| grep | Команда для пошуку тексту в файлі. |
| df | Команда для відображення інформації про доступні файлові системи. |
| ssh | Команда для встановлення безпечного з'єднання між двома комп'ютерами. |
| ls -l words  bzip2 words  ls -l words.bz2 | ls -l words - виводить детальний список файлів у поточній директорії, включаючи права доступу, власника, розмір та дату створення файлу з іменем words.  bzip2 words - стискання файлу words за допомогою алгоритму Bzip2.  ls -l words.bz2 - виводить детальний список файлів у поточній директорії, включаючи права доступу, власника, розмір та дату створення стислого файлу з іменем words.bz2. |
| ls -l words.bz2  bunzip2 words.bz2  ls -l words | ls -l words.bz2 - виводить детальний список файлів у поточній директорії, включаючи права доступу, власника, розмір та дату створення стислого файлу з іменем words.bz2.  bunzip2 words.bz2 - розпаковує стислий файл words.bz2 за допомогою алгоритму Bzip2. Коли ця команда виконується, вона створює оригінальний файл з іменем words, який містить вміст, що було стиснуто в стислий файл words.bz2.  ls -l words - виводить детальний список файлів у поточній директорії, включаючи права доступу, власника, розмір та дату створення оригінального файлу з іменем words. Цей файл має той самий розмір та вміст, що й перед стисканням командою bzip2, оскільки він був розпакований зі стислого файлу words.bz2. |
| ls -l words.xz  unxz words.xz  ls -l words | ls -l words.xz - виводить детальний список файлів у поточній директорії, включаючи права доступу, власника, розмір та дату створення стислого файлу з іменем words.xz.  unxz words.xz - розпаковує стислий файл words.xz за допомогою алгоритму XZ. Коли ця команда виконується, вона створює оригінальний файл з іменем words, який містить вміст, що було стиснуто в стислий файл words.xz.  ls -l words - виводить детальний список файлів у поточній директорії, включаючи права доступу, власника, розмір та дату створення оригінального файлу з іменем words. Цей файл має той самий розмір та вміст, що й перед стисканням командою xz, оскільки він був розпакований зі стислого файлу words.xz. |
| zip -r udev.zip /etc/udev | Команда zip -r udev.zip /etc/udev створює стислий файл архіву з назвою udev.zip, який містить усі файли та піддиректорії зі шляху /etc/udev. Опція -r вказує на те, що команда має створювати архів з піддиректоріями рекурсивно. |
| ls -l udev.zip | ls -l udev.zip - виводить детальний список файлів у поточній директорії, включаючи права доступу, власника, розмір та дату створення стислого файлу з іменем udev.zip. |
| unzip -l udev.zip | Команда unzip -l udev.zip покаже список файлів, які знаходяться в архіві udev.zip, їхні розміри та дати створення/зміни. Це дозволяє перевірити зміст архіву, не розпаковуючи його повністю. |

\*\*\*Скріншоти виконання команд в терміналі можна не представляти, достатньо коротко описати команди в таблиці.

***Готував матеріал студент Кресан Руслан***

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

* **створити файл з розширенням** **.tar;**

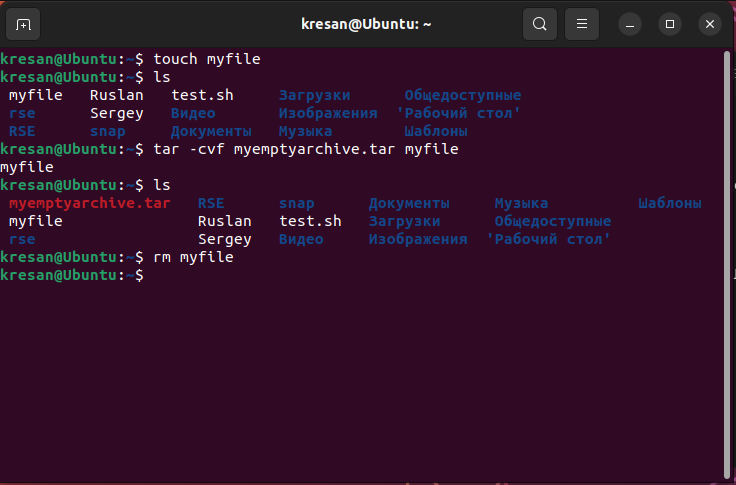
*Для того щоб створити файл з розширенням .tar, потрібно скористатися наступною командою:*

***tar -cvf myemptyarchive.tar***

*У моєму випадку ми одразу додаємо до архіву файл* ***myfile****:*

***tar -cvf myemptyarchive.tar myfile***

*Тому що без додавання файлу консоль видає повідомлення: “tar: Робкий отказ от создания пустого архива”.*

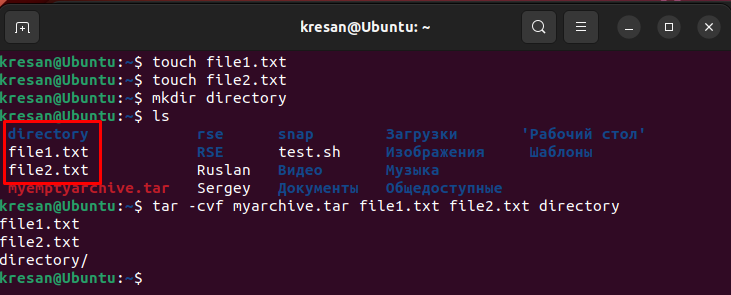


* **створити файл з розширенням .tar, що складається з декількох файлів і каталогів одночасно;**

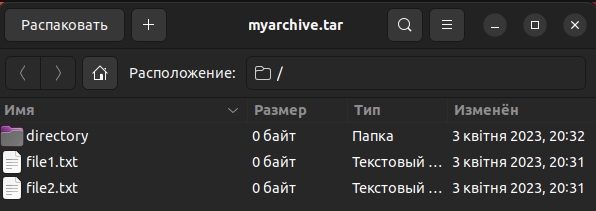
*Для виконання цієї команди ми створили два txt файли та один каталог. Після цього ми використали команду:*

***Tar -cvf myarchive.tar file1.txt file2.txt directory***

*Яка заносить створені нами файли у файл* з розширенням .tar.



*Коли ми зробили ці дії ми можемо відкрити наш архів і побачити файли які ми туди додали.*

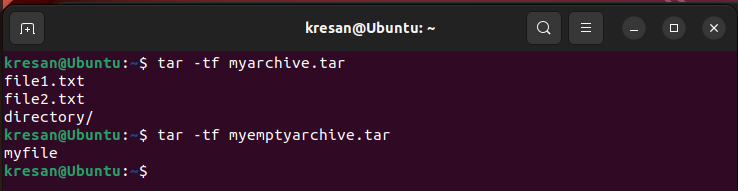


* **перегляду вмісту файлу;**

*Для перегляду вмісту файлу з розширенням* ***tar****, потрібно використати наступну команду:*

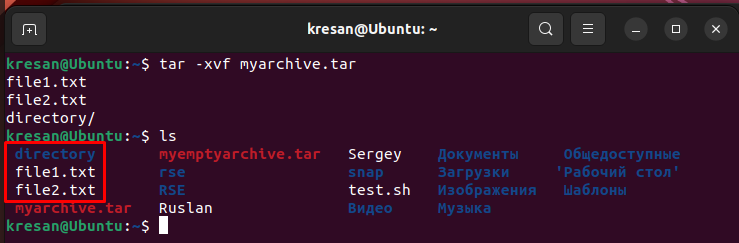
***tar -tf «Назва архіву»***

*Для демонстрації переглянемо вміст створених нами архівів.*



* **витягти вміст файлу tar;**

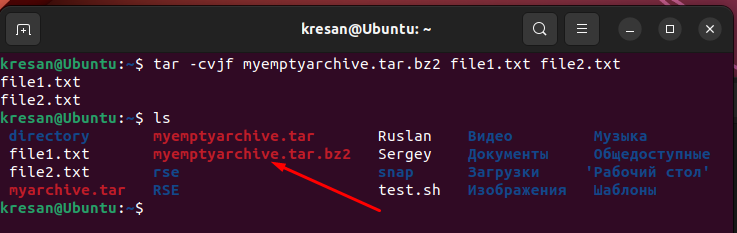
*Щоб витягти вміст файлу з розширенням .tar, ми використали команду tar з опцією -xvf та вказали назву архіву.*



*Після цього файли які ми витягли з архіву з’являться в поточному каталозі.*

* **створити архівний файл tar, стиснений за допомогою bzip;**

*Щоб створити архівний файл з розширенням .tar, стиснутий за допомогою bzip, ми використали команду tar з опцією -cvjf, вказавши назву стиснутого файлу та список файлів, які потрібно додати до архіву.*

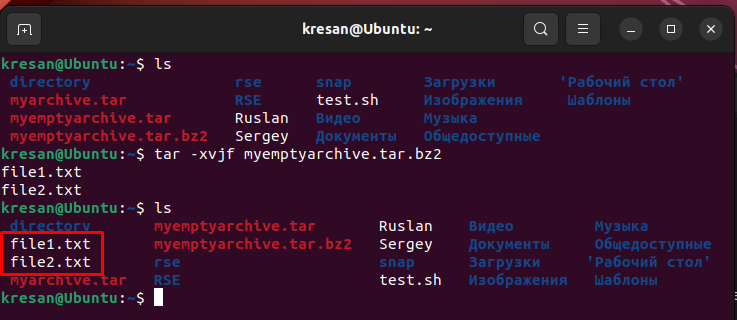


* **витягти вміст файлу tar bzip;**

*Для того щоб витягти вміст файлу* tar. Bzip можна скористатися командою:

***tar -xvjf myemptyarchive.tar.bz2***

*Де* ***myemptyarchive.tar.bz2*** *назва архіву.*

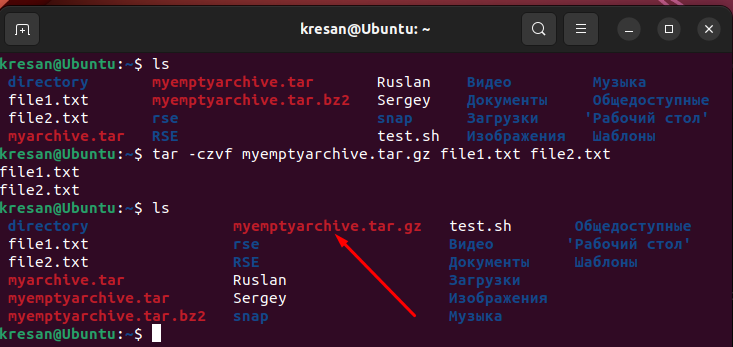


* створити архівний tar файл, стисненого за допомогою gzip;

*Щоб створити архівний tar файл стисненого за допомогою gzip потрібно скористатися командою:*

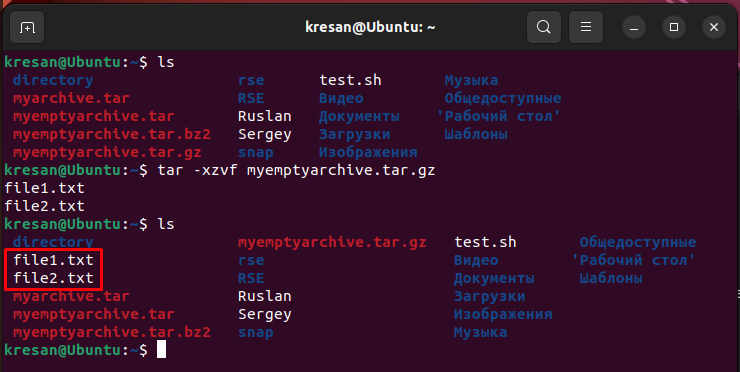
***tar -czvf myarchive.tar.gz file1.txt file2.txt***

*Де* ***myarchive.tar*** *це назва архіву, а* ***file1.txt*** *та* ***file2.txt*** *це файли які буде містити архів.*



* **витягти вміст файлу tar gzip.**

*Щоб витягнути вміст архівного файлу tar, стисненого за допомогою gzip, потрібно використати команду tar з опцією -xzvf і назвою архіву.*



***Готував матеріал студент Кресан Руслан***

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 | Команда перенаправляє виведення стандартного виведення (STDOUT) від команди "cmd" до файлу з назвою "file", перезаписуючи попередній вміст файлу. |
| cmd > file | Команда має ту саму дію, що і попередня, але вона не вказує який саме потік виведення перенаправляється, отже, вона за замовчуванням перенаправляє STDOUT. |
| cmd 2> file | Команда перенаправляє виведення потоку помилок (STDERR) від команди "cmd" до файлу з назвою "file", перезаписуючи попередній вміст файлу. |
| cmd >> file | Команда перенаправляє виведення стандартного виведення від команди "cmd" до файлу з назвою "file", але вона дописує вміст у кінець файлу замість перезаписування. |
| cmd &> file | Команда перенаправляє як STDOUT, так і STDERR від команди "cmd" до файлу з назвою "file", перезаписуючи попередній вміст файлу. |
| cmd > file 2>&1 | Команда перенаправляє STDOUT від команди "cmd" до файлу з назвою "file", а також перенаправляє STDERR до того ж самого файлу, використовуючи комбінований оператор перенаправлення "&>". |
| cmd >> file 2>&1 | Команда перенаправляє STDOUT від команди "cmd" до файлу з назвою "file", а також перенаправляє STDERR до того ж самого файлу, використовуючи комбінований оператор перенаправлення "&>" і дописує вміст у кінець файлу. |
| cmd 2>&1 > /dev/null | Команда перенаправляє STDERR до STDOUT, тобто обидва потоки виведення перенаправляються до одного і того ж місця. Далі STDOUT перенаправляється до /dev/null, що є спеціальним файлом, який відкидає вміст. |
| cmd 2> /dev/null | Команда перенаправляє потік помилок (STDERR) від команди "cmd" до /dev/null, що дозволяє відкинути всі повідомлення про помилки. |
| cmd1 | cmd2 | Команда передає виведення стандартного потоку вводу (STDIN) від команди "cmd1" до команди "cmd2" як стандартний виведення (STDOUT). Тобто команда "cmd2" отримує ввід від "cmd1" і обробляє його. |
| cmd1 2>&1 | cmd2 | Команда перенаправляє потік помилок (STDERR) від команди "cmd1" до STDOUT, після чого передається до команди "cmd2" через стандартний виведення (STDOUT). Тобто команда "cmd2" отримує і стандартний ввід і потік помилок (STDERR) від "cmd1" і обробляє їх. |

***Готував матеріал студент Губенко Є.О***

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 назвою "story" за допомогою оператора перенаправлення ">". | Потік, який перенаправляється, є стандартним виведенням (STDOUT), оскільки він є тим, що виводить результат команди "echo". Оператор ">" перенаправляє вивід команди в файл "story", перезаписуючи будь-який попередній вміст файлу. |
| $ date > date.txt | Ця команда записує поточну дату та час у файл з назвою "date.txt" за допомогою оператора перенаправлення ">". | Потік, який перенаправляється, є стандартним виведенням (STDOUT), оскільки команда "date" виводить результат своєї роботи в цей потік. Оператор ">" перенаправляє вивід команди в файл "date.txt", перезаписуючи будь-який попередній вміст файлу, або створюючи новий файл, якщо він не існує. |
| $ cat file1 file2 file3 > bigfile | Ця команда об'єднує вміст трьох файлів "file1", "file2" та "file3" у один файл "bigfile", використовуючи оператор перенаправлення ">". | Потік, який перенаправляється, є стандартним виведенням (STDOUT). Коли команда "cat" виконується без будь-яких аргументів командного рядка, вона очікує введення даних в стандартний ввід (STDIN) терміналу користувача, але в даному випадку, через наявність аргументів - імен файлів, вона зчитує вміст цих файлів і записує їх в стандартний вивід. Оператор ">" перенаправляє вміст стандартного виводу команди "cat" у файл "bigfile", перезаписуючи будь-який попередній вміст файлу, або створюючи новий файл, якщо він не існує. |
| $ls -l >> directory | Ця команда додає вміст виводу команди "ls -l" в кінець файлу з назвою "directory" за допомогою оператора перенаправлення ">>". | Потік, який перенаправляється, є стандартним виведенням (STDOUT), оскільки команда "ls -l" виводить список файлів із детальною інформацією про них у стандартний вивід. Оператор ">>" перенаправляє вивід команди в кінець файлу "directory", доповнюючи його новими даними. Якщо файл не існує, він буде створений. |
| $ sort < file1\_unsorted > file2\_sorted | Ця команда сортує вміст файлу "file1\_unsorted" та записує відсортований результат у файл з назвою "file2\_sorted", використовуючи оператори перенаправлення "<" та ">". | Потік, який перенаправляється, є стандартним введенням (STDIN), оскільки команда "sort" очікує введення даних у цей потік. Оператор "<" перенаправляє вміст файлу "file1\_unsorted" в стандартний ввід команди "sort". Потім, команда "sort" сортує введені дані та виводить результат у стандартний вивід (STDOUT). Оператор ">" перенаправляє вивід команди "sort" у файл "file2\_sorted", перезаписуючи будь-який попередній вміст файлу, або створюючи новий файл, якщо він не існує. |
| $ find -name '\*.txt' > file.txt 2> /dev/null | Ця команда виконує пошук файлів з розширенням ".txt" у поточній директорії та всіх її піддиректоріях за допомогою команди "find", та записує список знайдених файлів у файл з назвою "file.txt", використовуючи оператори перенаправлення "> " та "2> /dev/null". | Потік, який перенаправляється з команди "find", є стандартним виведенням помилок (STDERR), оскільки при виконанні команди можуть виникнути помилки, які будуть виведені у цей потік. Оператор "2>" перенаправляє виведення помилок у спеціальний файловий пристрій "/dev/null", який не зберігає вміст того, що до нього направляється. Це означає, що будь-які помилки, що виникають при виконанні команди "find", будуть ігноруватися. Оператор ">" перенаправляє стандартний вивід команди "find" у файл "file.txt", перезаписуючи будь-який попередній вміст файлу, або створюючи новий файл, якщо він не існує. |
| $ cat file1\_unsorted | sort > file2\_sorted | Ця команда сортує вміст файлу "file1\_unsorted" та записує відсортований результат у файл з назвою "file2\_sorted", використовуючи оператори перенаправлення "| " та ">". | Потік, який перенаправляється, є стандартним виведенням (STDOUT), оскільки команда "cat" виводить вміст файлу "file1\_unsorted" у цей потік, а команда "sort" очікує введення даних у цей потік. Оператор "|" (pipe) перенаправляє виведення команди "cat" у вхід команди "sort". Потім, команда "sort" сортує введені дані та виводить результат у стандартний вивід (STDOUT). Оператор ">" перенаправляє вивід команди "sort" у файл "file2\_sorted", перезаписуючи будь-який попередній вміст файлу, або створюючи новий файл, якщо він не існує. |
| $ cat myfile | grep student | wc -l | Ця команда зчитує вміст файлу "myfile", та застосовує до нього команду "grep" з аргументом "student", яка знаходить всі рядки, що містять слово "student". Результат потім передається до команди "wc -l", яка підраховує кількість рядків у введеному тексті та виводить кількість знайдених рядків у стандартний вивід (STDOUT). | Отже, ця команда виконує пошук кількості рядків, які містять слово "student" у файлі "myfile" та виводить цю кількість у стандартний вивід. Потік, який перенаправляється, є стандартним введенням (STDIN), оскільки команда "cat" виводить вміст файлу "myfile" у цей потік для подальшого оброблення командами "grep" та "wc". |

***Готував матеріал студент Губенко Є.О***

**Відповіді на контрольні запитання**

1. Надайте порівняльну характеристику процесам стискання та архівування.

*Процес стискання та процес архівування є двома різними методами зменшення обсягу даних для збереження або передачі.*

*Стискання даних використовується для зменшення обсягу файлу шляхом видалення повторюваних даних з вихідного файлу та заміни їх короткими покажчиками, які посилаються на раніше вживані дані. Це дозволяє зберегти більше даних на тому ж обсязі диска або зменшити обсяг передачі даних по мережі. Якщо файл стискається, то для його відновлення потрібна процедура декомпресії. До прикладів програм для стискання відносяться gzip, bzip2, lzma, zstd.*

*З іншого боку, архівування даних використовується для об'єднання кількох файлів в один файл або для збереження даних у вигляді архіву, який може містити кілька файлів у стиснутому вигляді, зберігаючи їх імена та іншу інформацію. Це зручно для збереження або передачі декількох файлів одночасно. При архівуванні зберігається інформація про імена файлів та їхню структуру. Для розпакування файлів із архіву потрібна процедура розархівування. До прикладів програм для архівування відносяться tar, zip, rar, 7zip.*

*Отже, хоча обидва методи використовуються для зменшення обсягу даних, їх використання залежить від конкретної ситуації. Якщо потрібно зменшити обсяг одного файлу, то стискання є кращим варіантом. Архівування зручніше використовувати, якщо потрібно передати кілька файлів або зберегти структуру каталогів.*

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

*У ОС Linux є багато програм для стискання та архівування файлів та каталогів. Деякі з них:*

* ***zip****: програма для створення та розпакування ZIP-архівів. Вона використовує алгоритм стиснення Deflate для зменшення розміру файлів.*
* ***unzip****: програма для розпакування ZIP-архівів.*
* ***rar****: програма для створення та розпакування RAR-архівів. Вона використовує складні алгоритми стиснення для зменшення розміру файлів.*
* ***unrar****: програма для розпакування RAR-архівів.*
* ***tar****: програма для створення та розпакування tar-архівів. Вона не стискає файли, а лише об'єднує їх у один архівний файл.*
* ***gzip****: програма для стискання файлів за допомогою алгоритму gzip. Вона зменшує розмір файлів шляхом заміни повторюваних послідовностей символів на посилання на попередні входження цих послідовностей в файлі.*
* ***gunzip****: програма для розпакування gzip-стиснених файлів.*
* ***bzip2****: програма для стискання файлів за допомогою алгоритму bzip2. Вона використовує більш складний алгоритм стиснення, що забезпечує більший ступінь стиснення, але й вимагає більше ресурсів комп'ютера.*
* ***bunzip2****: програма для розпакування bzip2-стиснених файлів.*

*Ці програми можуть бути корисними при стисканні та архівуванні файлів та каталогів в ОС Linux.*

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

*У Linux доступно кілька алгоритмів стискання, використовуваних в різних програмах, таких як gzip, bzip2, xz, а також програми tar, які можуть використовувати ці алгоритми для створення архівів.*

*Алгоритм gzip використовує DEFLATE-алгоритм стискання, який є дуже швидким і використовує невелику кількість оперативної пам'яті. Проте, він не дає найвищого ступеня стискання, і тому файл, стиснутий за допомогою gzip, може бути трохи більшим за той самий файл, стиснутий за допомогою інших алгоритмів.  
Алгоритм bzip2 використовує більше оперативної пам'яті та ресурсів процесора, ніж gzip, але зазвичай дає кращий ступінь стискання. Тому, якщо метою є максимальне стискання файлів, то краще використовувати bzip2.  
Алгоритм xz використовує ще більше ресурсів процесора та оперативної пам'яті, ніж bzip2, і тому зазвичай дає ще кращий ступінь стискання. Але через високу витрату ресурсів, він може працювати повільніше на менш потужних комп'ютерах.*

*Оскільки gzip є швидшим і використовує менше ресурсів, він зазвичай є найшвидшим варіантом для швидкого стискання файлів, які не потребують максимального ступеня стискання. Однак, якщо потрібно максимально зменшити розмір файлу, то більш ефективним варіантом буде використання bzip2 або xz.*

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

*Більшість сучасних мобільних телефонів мають вбудовані програмні засоби для стискання та архівування файлів. Ось деякі з них:*

* ***Google Files*** *- це програма для управління файлами, яка також має можливість стискання та архівування файлів. З його допомогою можна стиснути файли в форматах ZIP, 7Z, а також створити ZIP-архів з декількох файлів.*
* ***WinZip*** *- це популярна програма для архівування та стискання файлів. Вона доступна для багатьох мобільних платформ, таких як Android та iOS. WinZip дозволяє створювати ZIP-архіви, розпаковувати архіви та стискати файли в форматах ZIP та RAR.*
* ***RAR*** *- це програма для стискання та архівування файлів, доступна для Android та iOS. З його допомогою можна створювати архіви у форматах RAR та ZIP, розпаковувати архіви та стискати файли в форматах ZIP, RAR та 7Z.*
* ***ZArchiver*** *- це додаток для Android, який дозволяє стискати та розпаковувати файли у форматах ZIP, 7Z, TAR, GZIP та BZIP2. Він також підтримує шифрування файлів та може виконувати функції менеджера файлів.*
* ***iZip*** *- це програма для iOS, яка дозволяє стискати та розпаковувати файли у форматах ZIP, RAR, 7Z, TAR, GZIP та BZIP2. Вона також має можливість створювати ZIP-архіви з декількох файлів та підтримує функції менеджера файлів.*

*Ці програми є дуже корисними для зберігання та передачі файлів, особливо якщо обмін файлами відбувається на мобільному пристрої.*

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

*У ОС сімейства Windows є багато програмних засобів для стискання та архівування даних. Найпоширеніші з них:*

* ***Windows built-in zip****: це вбудований у Windows інструмент для архівації та стискання файлів та каталогів. Він підтримує стиснення у форматах zip, gzip та tar. Windows built-in zip простий у використанні та підтримує шифрування.*
* ***WinZip****: це один з найпопулярніших інструментів для архівації та стискання даних у Windows. WinZip підтримує широкий спектр форматів архівів, включаючи zip, gzip, tar, RAR та інші. Він також має можливість шифрування та захисту паролем.*
* ***7-Zip****: це безкоштовна програма з відкритим кодом, яка підтримує стиснення у форматах 7z, ZIP, GZIP, BZIP2 та TAR. 7-Zip має високу ступінь стискання та швидкодію.*
* ***WinRAR****: це комерційний інструмент для архівації та стискання даних, який підтримує широкий спектр форматів архівів, включаючи RAR, ZIP, CAB, ARJ, LZH, TAR, GZip, UUE, ISO та інші. WinRAR має можливість шифрування та захисту паролем.*

*У порівнянні зі вбудованим в Windows інструментом для архівації, програми WinZip, 7-Zip та WinRAR мають більше можливостей та ширший спектр підтримуваних форматів архівів. Щодо швидкодії та ступеня стискання, 7-Zip зазвичай вважається одним з найшвидших та найефективніших алгоритмів стискання у порівнянні з іншими програмами.*

1. 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 може бути використано для резервного копіювання даних, тобто для збереження копії важливих даних в безпечному місці для подальшого відновлення в разі втрати чи пошкодження основних даних.*

*Для резервного копіювання даних, зазвичай, використовуються програми для архівування та стискання даних, які дозволяють об'єднувати багато файлів та каталогів в один архівний файл, зменшуючи обсяг даних та полегшуючи їх збереження на зовнішніх носіях.*

*Після архівування та стискання даних, архівний файл може бути скопійовано на зовнішній носій, як от USB-флешку, зовнішній жорсткий диск або на хмарне сховище, де він буде зберігатись в безпеці до тих пір, поки не знадобиться його відновлення.  
Крім резервного копіювання, стискання та архівування даних може бути використано для ефективного зберігання великої кількості даних на обмеженому місці зберігання, для передачі даних по мережі з меншим використанням пропускної здатності та для ефективної організації та збереження даних в системах з обмеженими ресурсами.*

1. Яке призначення директорії файлу /dev/null?

*Директорія /dev/null у системі Linux і інших Unix-подібних операційних системах використовується для відправки в неї даних і відкидання їх без збереження. Вона не є звичайною директорією, що містить файли, а є файлом-примарою, що доступний як пристрій у системі. Коли дані записуються у файл /dev/null, вони будуть повністю відкинуті і не збережуться в системі. Це може бути корисно для програм, які генерують вивід, який не потрібен, або для перенаправлення виводу програми в нікуди.*

*Для прикладу, якщо ви хочете виконати команду, але не хочете зберігати результат виведення, ви можете скерувати вивід до /dev/null замість збереження його в файлі:* ***$ some\_command > /dev/null***

*Також /dev/null може використовуватися для знищення конфіденційної інформації, так як дані, що записані в цей файл, не можуть бути відновлені.*

***Готував матеріал студент Губенко Є.О***

**Висновки**

В ході виконання лабораторної роботи ми отримали навики роботи з командною оболонкою Bash та познайомились з базовими командами для архівування та стиснення даних та базовими діями при роботі з текстом у терміналі.