“Київський фаховий коледж зв’язку”

Циклова комісія Комп’ютерної та програмної 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-03

Команда: Яременко О.

Местецький А. Руда В.

Перевірив викладач

Сушанова В.С.

Київ 2023

**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

**Лабораторна робота №6 Дисципліна: Операційні системи**

**Завдання для попередньої підготовки:**

*Готував матеріал студенти(цифрою позначено хто-які питання готував ):*

*Олексій – 1*

*Вика – 2*

*Антон - 3*

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. 1

Команди tar, xz, zip, bzip та gzip є утилітами для стиснення та архівування файлів.

1. tar - це утиліта для створення або розархівування тар-архівів. Основні параметри:

-c - створення архіву

-x - розархівування архіву

-f - ім'я архіву

-v - виведення деталей процесу архівування/розархівування

2. xz - це утиліта для стиснення та розпакування файлів за допомогою алгоритму стиснення LZMA. Основні параметри:

-z - стиснутий файл буде мати розширення .xz

-d - розпакування файлу

3. zip - це утиліта для створення та розархівування ZIP-архівів. Основні параметри:

-r - рекурсивний прохід папок для додавання у архів

-q - тихий режим

-u - оновлювати файл в архіві, якщо він був змінений

4. bzip - це утиліта для стиснення та розпакування файлів за допомогою алгоритму стиснення BZip2. Основні параметри:

-z - стиснутий файл буде мати розширення .bz2

-d - розпакування файлу

5. gzip - це утиліта для стиснення та розпакування файлів за допомогою алгоритму стиснення GZip. Основні параметри:

-c - стандартне виведення на екран

-d - розпакування файлу

-r - рекурсивний прохід папок для додавання у архів

* 1. Наведіть три приклади реалізації архівування та стискання даних різними командами. 2 Команда gzip: gzip, tar: tar, 7zip: 7zip
  2. Яке призначення команд cat, less, more, head and tail? Зробіть короткий опис кожної команди та виділіть їх основні параметри. Яким чином їх можна встановити 3

\*\*cat\*\* - команда для виведення змісту файлу на екран. Основний параметр - назва файлу. Встановлення проводиться за допомогою пакетного менеджера.

\*\*less\*\* - команда для перегляду файлів і директорій. Основний параметр - назва файлу. Встановлення проводиться за допомогою пакетного менеджера.

\*\*more\*\* - команда для перегляду файлів по сторінкам. Основний параметр - назва файлу. Встановлення проводиться за допомогою пакетного менеджера.

\*\*head\*\* - команда для виведення перших рядків файлу. Основний параметр - назва файлу. Встановлення проводиться за допомогою пакетного менеджера.

\*\*tail\*\* - команда для виведення останніх рядків файлу. Основний параметр - назва файлу. Встановлення проводиться за допомогою пакетного менеджера.

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

Командна оболонка Linux (наприклад, bash) дозволяє створювати і керувати каналами, потоками та фільтрами для обробки виводу програм та команд.

Канали (pipes) - це механізм передачі даних між процесами, який дозволяє використовувати вивід однієї програми як вхід для іншої програми без потреби використовувати проміжний файл. Канал можна створити за допомогою вертикальної риски |. Наприклад, команда ls -l | less передає вивід команди ls -l як вхід для команди less.

Потоки (streams) - це набір даних, що передається між програмами або командами. В Linux існує три основних потоки: вхідний потік (stdin), вихідний потік (stdout) та потік помилок (stderr).

Фільтри - це програми, які приймають вхідні дані зі стандартного вводу, обробляють їх і виводять результат на стандартний вивід. Фільтри можуть бути поєднані між собою за допомогою конвеєрів, щоб обробляти дані послідовно та складніше обробляти потужніше команди.

* 1. Яке призначення команди grep? 2
  2. grep — утиліта інтерфейсу командного рядка, яка знаходить на вводі рядки, що відповідають заданому регулярному виразу, і виводить їх.

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\*\*\*

Каждый по 5 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 –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  ls -l udev.zip | 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 - 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**.

* 1. Ознайомтесь з командою tar та за її допомогою виконати у терміналі наступні дії:
* створити файл з розширенням .tar;
* **.tar** - це розширення файлу, яке вказує на формат архіву, створеного з допомогою утиліти **tar** в середовищі Unix / Linux.
* створити файл з розширенням .tar, що складається з декількох файлів і каталогів одночасно;
* Щоб створити файл з розширенням **.tar**, що складається з декількох файлів і каталогів одночасно, використайте команду **tar** в командному рядку в середовищі Unix / Linux зі списком файлів та каталогів, які необхідно включити до архіву. Наприклад, щоб створити архів **myarchive.tar**, який містить файли **file1.txt**, **file2.txt**, каталог **directory1** та його вміст
* перегляду вмісту файлу;
* В Linux є декілька способів переглянути вміст файлу з терміналу. Ось декілька прикладів:
* Використання команди cat:
* Використання команди less:
* Використання команди head або tail:
* витягти вміст файлу tar;
* Для того, щоб витягнути вміст файлу tar в Linux, можна використовувати команду "tar" з прапорцем "x" (що означає "витягти") за яким слідує ім'я архіву
* створити архівний файл tar, стиснений за допомогою bzip;
* Щоб створити архівний файл tar, стиснений за допомогою bzip, використовуйте опцію **-j** разом з командою **tar**. Опція **-j** вказує, що архів потрібно стиснути за допомогою програми bzip2.
* витягти вміст файлу tar bzip;
* Щоб витягнути вміст файлу tar, який був створений з використанням bzip2-стиснення, необхідно використовувати опцію "j" для tar-команди. Опція "j" вказує на використання програми bzip2 для стиснення або розпакування даних.
* створити архівний tar файл, стисненого за допомогою gzip;
* Щоб створити архівний файл tar, стиснений за допомогою bzip, використовуйте опцію **-j** разом з командою **tar**. Опція **-j** вказує, що архів потрібно стиснути за допомогою програми bzip2.
* витягти вміст файлу tar gzip.
* Для витягнення вмісту файлу tar, який був стиснутий за допомогою gzip, можна використовувати комбіновану команду "tar" та "gzip". Ось приклад команди:
* У цій команді "x" означає розпакування архіву, "z" - розжаття архіву, стиснутого за допомогою gzip, "v" - додатковий вивід інформації про процес розпакування архіву в термінал, "f" - вказує, що ми будемо використовувати вказаний файл як архів, яке слідує після цього прапорця. У нашому випадку файл - "archive.tar.gz".
* Після виконання цієї команди у вашій поточній директорії буде створено директорію з іменем, яке вказано у архіві, і всі файли та директорії, що містяться в архіві, будуть розпаковані до цієї директорії.
  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?** |
| 1 | cmd 1> file | Команда cmd 1> file перенаправляє стандартний вивід команди cmd в файл file, перезаписуючи вміст файлу, якщо він існує, або створюючи новий файл, якщо його не існує. |
| 2 | cmd > file | В Bash команда **cmd > file** використовується для перенаправлення стандартного виводу з команди **cmd** до файлу замість виводу в термінал.  Коли ви вводите цю команду, Bash створить файл з іменем **file** (якщо він не існує) і буде записувати вміст, який виводить команда **cmd** до цього файлу. Якщо файл існує, він буде перезаписано з новим вмістом. |
| 3 | cmd 2> file | перенаправляє стандартний вивід помилок (stderr) від команди cmd у файл з назвою file. |
| 1 | cmd >> file | Команда cmd >> file перенаправляє вивід команди cmd у файл file, дописуючи вміст до кінця файлу замість перезаписування вмісту файлу. |
| 2 | cmd &> file | В Bash команда **cmd &> file** використовується для перенаправлення як стандартного виводу, так і стандартного виводу помилок (stderr) з команди **cmd** до файлу замість виводу в термінал. Коли ви вводите цю команду, Bash створить файл з іменем **file** (якщо він не існує) і буде записувати вміст, який виводить команда **cmd** до цього файлу, включаючи будь-які повідомлення про помилки, які вона генерує. Якщо файл існує, він буде перезаписано з новим вмістом. |
| 3 | cmd > file 2>&1 | перенаправляє як стандартний вивід (stdout), так і стандартний вивід помилок (stderr) від команди cmd у файл з назвою file. |
| 1 | cmd >> file 2>&1 | Команда cmd >> file 2>&1 перенаправляє стандартний вивід та вивід помилок команди cmd в файл file, дописуючи їх до кінця файлу, якщо він існує, або створюючи новий файл, якщо його не існує. |
| 2 | cmd 2>&1 > /dev/null | коли ви вводите цю команду, Bash  відправляє стандартний вивід та  стандартний вивід помилок з команди  **cmd** до **/dev/null**, тобто будь-який  вивід буде ігнорований. |
| 3 | cmd 2> /dev/null | перенаправляє стандартний вивід помилок (stderr) від команди cmd у спеціальний файл /dev/null. |
| 1 | cmd1 | cmd2 | Команда cmd1 | cmd2 дозволяє використовувати вихідні дані однієї команди як вхідні дані для іншої команди, що дозволяє виконувати складні завдання, комбінуючи декілька команд. |
| 2 | cmd1 2>&1 | cmd2 | Отже, коли ви вводите цю команду, Bash відправляє  стандартний вивід та стандартний вивід помилок з команди  **cmd1** до команди **cmd2** через канал. Це означає, що вивід з  команди **cmd1** буде передано в команду **cmd2**, де ви можете  виконати додаткові дії з цим виводом. |

* 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?** |
| 3 | $echo "It is a new story." > story | **Тобто, якщо ми перенаправляємо стандартний вивід помилок у файл /dev/null, то будь-які помилки, що виникають при виконанні команди cmd, будуть повністю ігноровані, і ми не побачимо їх на екрані або в будь-якому іншому виводі.** | перенаправляє стандартний вивід помилок (stderr) від команди cmd у спеціальний файл /dev/null.і |
| 1 | $ date > date.txt | Команда $ date > date.txt створює файл з назвою "date.txt" та записує в нього поточну дату та час. | потік перенаправлення > використовується для виведення даних з команди date до файлу date.txt. |
| 2 | $ cat file1 file2 file3 > bigfile | команда **cat file1 file2 file3 > bigfile** використовує перенаправлення стандартного виводу, щоб записати дані з трьох файлів у новий файл **bigfile**. |  |
| 3 | $ls -l >> directory | Символ >> перенаправляє стандартний вивід (stdout) від команди ls -l у файл з назвою directory, додаючи результат в кінець файлу. Якщо файл з назвою directory не існує, то він буде створений автоматично. | додає результат виконання команди ls -l в кінець файлу з назвою directory. |
| 1 | $ sort < file1\_unsorted > file2\_sorted | Команда $ sort < file1\_unsorted > file2\_sorted сортує вміст файлу file1\_unsorted та записує відсортований результат у файл з назвою file2\_sorted. | Перший потік < використовується для введення даних у команду sort, а другий потік > - для виведення результату з команди sort до файлу file2\_sorted. |
| 2 | $ find -name '\*.txt' > file.txt 2> /dev/null | Отже, команда **find -name '\*.txt' > file.txt 2> /dev/null** використовує перенаправлення стандартного виводу та стандартного помилкового виводу (stderr), щоб записати список файлів, які мають розширення **.txt**, до файлу **file.txt** |  |
| 3 | $ cat file1\_unsorted | sort > | Зазвичай, після символу > необхідно вказувати назву файлу, до якого буде перенаправлено виведення команди. Отже, без назви файлу команда буде неповною і виконання команди призведе до помилки. | сортує вміст файлу з назвою file1\_unsorted та записує відсортований список у файл, але вказано назву файлу, куди потрібно записувати вміст не було вказано. |
| 1 | $ cat myfile | grep student | wc -l | Після виконання цієї команди на екран буде виведено кількість рядків у файлі myfile, які містять слово "student". | Цей потік перенаправлення виконує три команди у послідовності |

**Контрольні запитання:**

*Готував матеріал студенти(цифрою позначено хто-які питання готував ):*

*Олексій – 1*

*Вика – 2*

*Антон - 3*

1. Надайте порівняльну характеристику процесам стискання та архівування. 1

Процес стискання та архівування є двома різними методами оптимізації розміру файлів на операційній системі Linux.

Стискання - це процес зменшення розміру файлу шляхом видалення повторюваних або зайвих даних. До прикладів стиснення на Linux відносяться команди gzip, bzip2, xz та compress.

Архівування - це процес зберігання одного або декількох файлів в одному файлі з метою зменшення їхнього розміру та зручності зберігання. Найпоширеніші програми для архівування на Linux - це tar, zip та 7zip.

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. 2

xz: це програма для стискання файлів з використанням алгоритму xz. Вона також може використовуватись разом з командою tar для створення архівів зі стисненими файлами.

zip: це програма для створення та розпакування ZIP-архівів. Вона підтримує різні алгоритми стиснення, такі як deflate, bzip2 та ін.

7zip: це програма для створення та розпакування архівів з використанням алгоритму 7z. Вона підтримує високу ступінь стиснення та підтримує різні алгоритми стиснення, такі як LZMA, LZMA2, bzip2 та ін.

rar: це програма для створення та розпакування RAR-архівів. Вона підтримує високу ступінь стиснення та підтримує різні алгоритми стиснення, такі як RAR, ZIP, CAB та ін.

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

Найшвидшим та найефективнішим алгоритмом стиснення в Linux є LZMA. Це алгоритм, що використовує довготривалу компресію і достатньо високу ступінь стиснення. Це дозволяє зменшити розмір файлів на достатньо високий рівень. Також використовується алгоритм Bzip2, який також має доволі високий рівень стиснення, але його процес компресії потребує більше часу. Інші алгоритми, такі як Gzip і XZ, мають посередньо високий рівень стиснення, але процес компресії потребує більше часу, ніж LZMA.

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

Ось кілька програмних засобів, які можуть бути використані для цієї мети:

ES File Explorer: Це потужний файловий менеджер для Android, який підтримує ряд форматів стискання та архівування файлів, включаючи ZIP, RAR, 7Z, GZ, TAR та багато інших.

ZArchiver: Це ще один потужний файловий менеджер для Android, який підтримує багато форматів стискання та архівування файлів.

WinZip: Цей додаток, який доступний для Android, є одним з найбільш популярних програмних засобів для стискання та архівування файлів.

1. Опишіть та порівняйте програмні засоби для стискання та (де)архівування даних у ОС сімейства Windows. 2

У ОС сімейства Windows є багато програм для стискання та (де)архівування даних, серед яких найбільш популярні наступні:

WinZip: це один з найбільш відомих та популярних програм для стискання та (де)архівування даних у Windows. Вона підтримує різні формати архівів, такі як ZIP, RAR, 7Z, TAR та ін. WinZip також може створювати саморозпаковуючі архіви та захищати їх паролем.

WinRAR: це ще одна дуже популярна програма для стискання та (де)архівування даних у Windows. Вона підтримує формати архівів, такі як RAR, ZIP, CAB, ARJ та ін. WinRAR може стискувати файли з використанням різних алгоритмів стиснення, таких як RAR, ZIP, 7Z та ін.

7-Zip: це вільно поширюваний архіватор, який підтримує формати архівів, такі як 7z, ZIP, RAR, TAR, GZIP та ін. 7-Zip підтримує високу ступінь стиснення з використанням алгоритмів LZMA та LZMA2.

PeaZip: це вільно поширюваний архіватор, який підтримує формати архівів, такі як 7z, ZIP, RAR, TAR, GZIP та ін. PeaZip має інтуїтивно зрозумілий інтерфейс користувача та підтримує різні алгоритми стиснення.Bandizip: це безкоштовний архіватор, який підтримує формати архівів, такі як ZIP, RAR, 7Z, TAR, LZH та ін. Bandizip має дуже швидкий алгоритм стиснення та може створювати саморозпаковуючі архіви.

1. Поясніть яким чином стиснення та архівування даних може бути використано для резервування даних. В яких ще задачах системного адміністрування воно може бути використано. 3

Стиснення та архівування даних може бути використано для резервування даних, щоб забезпечити безпечне збереження інформації в разі виникнення аварійних ситуацій. Архівування даних дозволяє зберігати дані в оптимальному вигляді і зменшувати зайву пам'ять. Стиснення даних дозволяє зменшити розмір файлів, що допомагає заощадити дискове простір та прискорити їх передачу.

1. Яке призначення директорії файлу /dev/null? 1

Директорія /dev/null має спеціальне призначення. Вона представляє собою віртуальний файл, який може бути використаний для відкидання будь-якої виводу з програми або ввідних даних до програми.

Коли будь-яка програма відправляє вивід у файл /dev/null, то цей вивід просто відкидається і програма не отримує жодного підтвердження про успішність відправлення. Таким чином, використання файлу /dev/null дозволяє програмі відправляти вивід у "никуди" і підтримує чистоту виводу в консолі.

2 **ВИСНОВОК:**

В ході виконання лабораторної роботи нами було досліджено командна оболонкою Bash, познайомились з базовими командами для архівування та стиснення даних, з базовими діями при роботі з текстом у терміналі,порівняли алгоритми стискання,дізнались відмінності стиснякання,і архівування,закріпили на практиці багато команд в Linux по роботі з архівуванням,стисканням,копіюванням,створили ахівний файл tar.