Kyiv College of Communications

PERFORMANCE REPORT

Laboratory work No. 6

Discipline: "Operating Systems"

Topic: Команди Linux для архівування та стиснення даних. Робота з текстом

Performed by students of Group CSN-33 *(Computer Systems and Networks)*.:

Kuznetsov Artur Serhiyovych

Finchuk Alina Oleksiivna

Checked by:

Sushanova V.S.

Kyiv – 2025

**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.

|  |  |  |
| --- | --- | --- |
| English term | Ukrainian translation | Пояснення своїми словами |
| Compression | стиснення | Зменшення розміру файлу, щоб він займав менше місця. |
| Archiving | архівування | Об’єднання кількох файлів у один для зручного зберігання або передачі. |
| Lossless compression | безвтратне стиснення | Тип стиснення, де можна повністю відновити оригінальний файл (наприклад, gzip). |
| Lossy compression | стиснення з втратами | Частина даних втрачається, тому файл відновити повністю не можна (наприклад, mp3, jpeg). |
| tar | створює архів | Команда, що об’єднує файли/папки в один архів (.tar). Може поєднуватися з іншими стисненнями (gzip, bzip2, xz). |
| gzip | стискання файлів | Швидкий метод стиснення файлів (.gz). Використовує алгоритм DEFLATE. |
| bzip2 | стискання файлів | Стиснення з кращим коефіцієнтом, але повільніше. Створює файли з розширенням .bz2. |
| xz | стискання файлів | Новіший формат із сильним стисненням, але повільний. Використовує алгоритм LZMA2. |
| tar czvf | створити gzip-архів | c — створити, z — gzip, v — показати процес, f — вказати ім’я файлу. |
| tar xzvf | розпакувати gzip-архів | x — витягнути, інші прапори аналогічні. |
| -r flag | параметр -r | Означає “recursive” — діяти рекурсивно (тобто у всіх вкладених папках). |
| -d flag | параметр -d | “decompress” — розпакувати архів або стиснений файл. |
| \*\*pipe ( | )\*\* | конвеєр |
| redirect (> , >> , <) | перенаправлення | Збереження або читання даних із файлу замість екрана. |
| standard output (stdout) | стандартний вихід | Дані, які команда виводить у термінал. |
| standard input (stdin) | стандартне введення | Дані, які команда отримує (з клавіатури або з файлу). |
| standard error (stderr) | стандартна помилка | Потік, де виводяться повідомлення про помилки. |
| grep | пошук у тексті | Команда для пошуку рядків за шаблоном у файлах. |
| cat | показати вміст | Виводить текст із файлів у терміналі. |
| less / more | перегляд тексту | Дозволяє читати великий текст частинами у вікні терміналу. |
| head / tail | початок / кінець файлу | Показують перші або останні рядки файлу. |
| stdout redirection | перенаправлення виводу | Наприклад, cmd > file — записує результат у файл. |
| stderr redirection | перенаправлення помилок | Наприклад, cmd 2> file — записує помилки у файл. |
| /dev/null | чорна діра | Спеціальний файл, куди можна “викинути” непотрібний вивід (він просто зникає). |

1. Додаткові матеріали для вивчення:

* [Як архівувати файли в Linux](https://ittutorials.co.ua/2024/10/29/%D1%8F%D0%BA-%D0%B0%D1%80%D1%85%D1%96%D0%B2%D1%83%D0%B2%D0%B0%D1%82%D0%B8-%D1%84%D0%B0%D0%B9%D0%BB%D0%B8-%D0%B2-linux/)
* [Команда tar](https://docs.rockylinux.org/10/uk/guides/backup/tar/)
* [Стандартні потоки в Linux](https://ittutorials.co.ua/2024/08/07/%D1%81%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82%D0%BD%D1%96-%D0%BF%D0%BE%D1%82%D0%BE%D0%BA%D0%B8-%D0%B2-linux/)
* [Потоки введення / виведення в Bash](https://docs.google.com/document/d/1KFgPMczSDduN6ETikkTAeSmQqCTjPrQrWhzeJVlvtlU/edit?usp=sharing)

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. \*\*Наведіть три приклади реалізації архівування та стискання даних різними командами.
   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
   4. \*\*Поясніть принципи роботи командної оболонки з каналами, потоками та фільтрами
   5. \*Яке призначення команди grep?

### **Призначення команд tar, xz, zip, bzip, gzip** **tar** — це програма, яка збирає багато файлів у один архів. Вона сама по собі не стискає, просто “пакує”. Зазвичай використовують разом зі стисненням, наприклад: tar -czvf file.tar.gz folder/ — створить архів і одразу стисне gzip’ом. Головні прапори:

* c — створити,
* x — розпакувати,
* v — показати процес,
* f — вказати назву файлу,
* z, j, J — стискання gzip, bzip2 або xz.  
   Встановлюється стандартно — вже є в Linux.

**gzip** — стискає файли швидко, але не максимально сильно. Дає файли з розширенням .gz.  
 Команда проста:  
 gzip file → стане file.gz.  
 Розпакувати: gzip -d file.gz або gunzip file.gz.  
 Параметри:

* -d — розпакувати,
* -r — рекурсивно по теках,
* -1…9 — рівень стиснення (1 — швидко, 9 — сильно).

**bzip2** — стискає краще, але повільніше. Файли .bz2.  
 bzip2 file → file.bz2,  
 bzip2 -d file.bz2 — розпакувати.  
 Є прапор -s — зменшити споживання пам’яті.

**xz** — ще новіший формат, стискає дуже сильно, але довго. Файли .xz.  
 xz file → file.xz,  
 xz -d file.xz — розпакування.  
 Прапори:

* -l — інформація про архів,
* -0…9 — рівень стиснення,
* -e — “екстремальне” стиснення.

**zip** — класичний архіватор (як у Windows). Може і архівувати, і стискати одразу.  
 zip file.zip file1 file2  
 Розпакувати: unzip file.zip.  
 Якщо нема — встановити через sudo apt install zip unzip.

### **Приклади архівування та стиснення**

1. **Створення gzip-архіву:** tar -czvf backup.tar.gz /home/user/docs  
   (створює архів docs.tar.gz із каталогу /home/user/docs)
2. **Створення bzip2-архіву:** tar -cjvf photo.tar.bz2 /home/user/photos
3. **Просто стиснення файлу без tar:** xz file.txt  
   (отримаємо file.txt.xz)

**Команди cat, less, more, head, tail**  
 **- cat** — показує вміст файлу у терміналі.  
 cat file.txt — просто виведе.  
 Можна з’єднати кілька файлів: cat file1 file2 > all.txt.  
 **- less** — перегляд файлів “посторінково”. Удобно для великих файлів.  
 less file.txt → прокручувати стрілками, вихід — q.  
 **- more** — схоже на less, тільки простіше. Теж показує великий файл частинами.  
 **- head** — показує початок файлу (за замовчуванням 10 рядків).  
 head -n 5 file.txt — 5 перших рядків.  
 **- tail** — показує кінець файлу.  
 tail -n 10 log.txt — останні 10 рядків.  
 Є ще tail -f log.txt — стежить за змінами “вживу” (зручно для логів).  
Усі ці команди вже є в системі, нічого ставити не треба.

**Принцип роботи каналів, потоків і фільтрів**  
У Bash є три “потоки”:  
 - stdin — вхід (що команда читає),  
 - stdout — вихід (що команда показує),  
 - stderr — помилки.  
  
 **Перенаправлення:**

* > — записати у файл,
* >> — дописати в кінець файлу,
* < — читати з файлу,
* 2> — відправити помилки у файл,
* | (пайп) — передати результат однієї команди іншій.

Наприклад: cat file.txt | grep "error" | wc -l  
— показує, скільки рядків містять слово "error".  
 Тобто одна команда передає результат наступній — як ланцюжок.

### **Команда grep** **grep** шукає текст у файлі. grep "слово" file.txt — покаже всі рядки, де є “слово”. Корисні прапори:

* -i — не зважати на регістр,
* -r — шукати у всіх теках,
* -n — показати номер рядка.  
   Приклад: grep -inr "linux" /etc  
  шукає “linux” в усіх файлах теки /etc.

**Хід роботи:**

1. Початкова робота в CLI-режимі в Linux ОС сімейства Linux:
2. Запустіть операційну систему Linux Ubuntu. Виконайте вхід в систему та запустіть термінал ***(якщо виконуєте ЛР у 401 ауд.)***.
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 | створює нову папку **mybackups** у поточній директорії (щоб туди зберігати архіви). |
| tar -cvf mybackups/udev.tar /etc/udev | робить архів з папки /etc/udev.  -c — створити архів, -v — показати, що робить, -f — вказати ім’я файлу. |
| tar -tvf mybackups/udev.tar | показує вміст архіву без розпакування.  -t — переглянути файли всередині. |
| tar -xvf mybackups/udev.tar | розпаковує архів у поточну теку.  -x — витягнути файли. |
| gzip udev.tar | стискає архів до файлу udev.tar.gz (менший розмір). |
| gzip -d udev.tar.gz | розпаковує файл, повертає оригінальний udev.tar. |
| bzip2 udev.tar | стискає архів у формат .bz2, більш щільно, але повільніше. |
| bzip2 -d udev.tar.bz2 | розпаковує архів .bz2. |
| xz udev.tar | стискає файл у формат .xz, це найсильніше стиснення, але займає більше часу. |
| xz -d udev.tar.xz | розпаковує .xz архів. |
| tar -czvf backup.tar.gz /etc | одразу створює архів і стискає gzip’ом (-z). |
| tar -cjvf backup.tar.bz2 /etc | архів + bzip2 стискання (-j). |
| tar -cJvf backup.tar.xz /etc | архів + xz стискання (-J). |
| tar -xzvf backup.tar.gz | розпаковує gzip-архів. |
| tar -xjvf backup.tar.bz2 | розпаковує bzip2-архів. |
| tar -xJvf backup.tar.xz | розпаковує xz-архів. |
| zip archive.zip file1 file2 | створює zip-архів з кількох файлів. |
| unzip archive.zip | розпаковує zip-архів. |
| cat file.txt | показує вміст файлу прямо в терміналі. |
| cat file1 file2 > all.txt | об’єднує два файли в один (> створює новий файл). |
| less file.txt | зручно читати великий текст. можна гортати вверх-вниз, вихід — q. |
| more file.txt | простіший варіант less, показує по сторінках. |
| head -n 5 file.txt | показує перші 5 рядків файлу. |
| tail -n 10 file.txt | показує останні 10 рядків. |
| tail -f /var/log/syslog | “слушає” файл у реальному часі — бачиш нові рядки, коли з’являються. |
| sort file.txt | сортує рядки файлу за алфавітом. |
| uniq file.txt | прибирає дублікати рядків (часто після sort). |
| wc -l file.txt | рахує кількість рядків у файлі.  -l — лінії, -w — слова, -c — байти. |
| grep "root" /etc/passwd | шукає рядки, де є слово “root”. |
| grep -inr "error" /home | шукає “error” у всіх файлах папки, незалежно від регістру (-i, -r). |
| find /home -name "\*.txt" | шукає файли з розширенням .txt у папці /home. |
| sort < file1\_unsorted > file2\_sorted | бере дані з одного файлу (<), сортує, і записує в інший (>). |
| `cat file1 | grep "user" |
| echo "Hello" > text.txt | записує фразу в файл (створює або перезаписує). |
| echo "new line" >> text.txt | додає текст у кінець файлу (>>). |
| cmd 2> errors.txt | записує тільки помилки в файл errors.txt. |
| cmd > out.txt 2>&1 | записує і звичайний вивід, і помилки в один файл. |
| `cmd1 | cmd2` |
| /dev/null | “чорна діра” — куди можна відправити непотрібний вивід, щоб не засмічував екран. |

**Примітка:** **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;
* 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.

## Робота з командою tar

### 1. Створити файл з розширенням .tar tar -cvf archive.tar file1.txt - створює архів із файлу file1.txt. -c — створити, -v — показати процес, -f — назва архіву.

### 2. Створити .tar, що складається з кількох файлів і тек tar -cvf backup.tar file1.txt file2.txt folder1 folder2 - створює один архів із кількох файлів і каталогів. дуже зручно, щоб зібрати все в один .tar.

### 3. Перегляд вмісту архіву без розпакування tar -tvf backup.tar - показує, що всередині архіву. -t — перегляд (table of contents).

### 4. Витягти вміст файлу .tar tar -xvf backup.tar - розпаковує архів у поточну директорію. -x — “extract” (витягнути).

### 5. Створити архівний .tar, стиснений за допомогою bzip2 tar -cjvf archive\_bzip.tar.bz2 folder1 - створює архів і стискає його bzip2. -j — включає bzip2 стиснення.

### 6. Витягти вміст .tar.bz2 tar -xjvf archive\_bzip.tar.bz2 - розпаковує архів, стиснений bzip2.

### 7. Створити архів .tar, стиснений за допомогою gzip tar -czvf archive\_gzip.tar.gz folder1 - створює архів і стискає gzip’ом. -z — gzip.

### 8. Витягти вміст .tar.gz tar -xzvf archive\_gzip.tar.gz - розпаковує gzip-архів.

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 звичайного результату (stdout) записується у файл file. Якщо файл існує — він перезаписується. |
| cmd > file | Те саме, що й вище — стандартний вихід команди записується у файл. |
| cmd 2> file | Помилки (stderr) записуються у файл file, а результат команди показується на екрані. |
| cmd >> file | Результат додається в кінець існуючого файлу, нічого не стирає. |
| cmd &> file | І звичайний вивід, і помилки записуються в один файл. |
| cmd > file 2>&1 | Спочатку перенаправляє stdout у файл, потім до нього додаються помилки — усе в одному файлі. |
| cmd >> file 2>&1 | Те саме, але все додається в кінець файлу, не стираючи старе. |
| cmd 2>&1 > /dev/null | Помилки відображаються, але звичайний вивід зникає (йде в “чорну діру” /dev/null). |
| cmd 2> /dev/null | Помилки просто ігноруються — не показуються і не записуються. |
| `cmd1 | cmd2` |
| `cmd1 2>&1 | 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 | Створює файл story і записує в нього текст *It is a new story.* | > — перенаправлення стандартного виходу у файл. |
| date > date.txt | Записує поточну дату і час у файл date.txt. | > — stdout у файл. |
| cat file1 file2 file3 > bigfile | Об’єднує три файли в один bigfile. | > — стандартний вихід у файл. |
| ls -l >> directory | Додає список файлів у кінець файлу directory, не стираючи попередній вміст. | >> — дописування у файл. |
| sort < file1\_unsorted > file2\_sorted | Читає дані з файлу file1\_unsorted, сортує і записує результат у file2\_sorted. | < — читання з файлу, > — запис у файл. |
| find -name '\*.txt' > file.txt 2> /dev/null | Знаходить усі .txt файли, результат записує у file.txt, а помилки ховає (викидає в “чорну діру”). | > — stdout у файл, 2> /dev/null — stderr у нікуди. |
| `cat file1\_unsorted | sort > file2\_sorted` | Передає вміст файлу у команду sort, сортує і записує результат у file2\_sorted. |
| `cat myfile | grep student | wc -l` |

**Контрольні запитання:**

1. Надайте порівняльну характеристику процесам стискання та архівування.
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.
3. \*Порівняйте алгоритми стискання, що використовуються в командах (програмах), використовуваних в Linux. Які з алгоритмів можна вважати найшвидшим та найефективнішим?
4. \*Опишіть програмні засоби для стискання та архівування, що можуть бути використані у вашому мобільному телефоні.
5. \*Опишіть та порівняйте програмні засоби для стискання та (де)архівування даних у ОС сімейства Windows.
6. \*\*Поясніть яким чином стиснення та архівування даних може бути використано для резервування даних. В яких ще задачах системного адміністрування воно може бути використано.
7. \*\*Яке призначення директорії файлу /dev/null?