“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”**

Виконав(ла/ли) студент(ка/и)

групи КСМ-23А

Bald Hedgehogs:

Кочубей С.С. Кошкін І.О.

Сарапин Я.О.

Перевірила 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:(Koshkin I)**

**1)**

**tar** — архіватор, що об’єднує файли в один архів, але без стиснення.

* Основні параметри:
  + **-c** — створити архів,
  + **-x** — розпакувати архів,
  + **-f [file]** — вказати ім'я файлу архіву,
  + **-v** — відображати процес.
* Приклад: **tar -cvf archive.tar file1 file2**

**xz** — стискає та розпаковує файли за допомогою алгоритму LZMA.

* Основні параметри:
  + **-z** — стиснення файлу,
  + **-d** — розпакування файлу,
  + **-k** — залишити вихідний файл без змін.
* Приклад: **xz -z file.txt** створює **file.txt.xz**.

**zip** — архіватор і стискальний засіб, що створює архіви з розширенням **.zip.**

* Основні параметри:
  + **-r** — рекурсивно додати файли з директорії,
  + **-e** — захист паролем,
  + **-u** — оновити наявний архів.
* Приклад: **zip archive.zip file1 file2**

**bzip2** — стискальний засіб, що використовує алгоритм BWT для створення **.bz2** архівів.

* Основні параметри:
  + **-z** — стиснути файл (стандартна дія),
  + **-d** — розпакувати файл,
  + **-k** — зберегти оригінал.
* Приклад: **bzip2 file.txt** створює **file.txt.bz2.**

**gzip** — стискальний засіб, який використовує алгоритм DEFLATE.

* Основні параметри:
  + **-c** — вивести стиснений вміст до stdout,
  + **-d** — розпакувати файл,
  + **-k** — зберегти оригінал.
* Приклад: **gzip file.txt** створює **file.txt.gz.**

**2)**

**Приклади архівування та стиснення:**

* Архівування та стиснення: **tar -czvf archive.tar.gz file1 file2**
* Стиснення **xz: xz -z file.txt**
* Архівування з паролем у **zip: zip -e secure.zip file1 file2**

**3)**

**cat** — виводить вміст файлу або об’єднує кілька файлів для перегляду.

Основні параметри:

**-n** — нумерація рядків,

**-E** — показує символ кінця рядка **$**.

Приклад: **cat file.txt**

**less** — інструмент для посторінкового перегляду довгих файлів з можливістю прокручування.

Основні параметри:

**-N** — нумерація рядків,

**-S** — вимкнення автоматичного переносу рядків.

Приклад: **less file.txt**

**more** — виводить файли посторінково, працює з основними клавішами для прокручування.

Основні параметри:

**-d** — відображає підказки для навігації,

**-n [num]** — кількість рядків для відображення на екрані.

Приклад: **more file.txt**

**head** — виводить перші рядки файлу.

Основні параметри:

**-n [num]** — кількість рядків для відображення.

Приклад: **head -n 5 file.txt**

**tail** — виводить останні рядки файлу.

Основні параметри:

**-n [num]** — кількість рядків для відображення,

**-f** — безперервний перегляд (корисно для логів).

Приклад: **tail -n 5 file.txt**

**4)**

Командна оболонка в Linux працює з трьома основними типами потоків: стандартним входом (stdin), стандартним виходом (stdout) та стандартним потоком помилок (stderr). Канали (**|**) дозволяють передавати дані з виходу однієї команди в вхід наступної, що дозволяє створювати складні команди з використанням простих утиліт. Фільтри — це команди, які обробляють дані, приймаючи їх з stdin, виконуючи певні дії (наприклад, сортування чи фільтрацію), і виводячи результати на stdout, що робить обробку даних більш гнучкою і потужною.

**5)**

Призначення **grep** шукати рядки в тексті, що відповідають вказаному шаблону, і виводить їх.

**Хід роботи:**

2. (Kochubei S. 9v)

|  |  |
| --- | --- |
| Назва команди | Її призначення та функціональність |
| mldir mybackups | Створює нову директорію з назвою "mybackups", що дозволяє організувати файли та резервні копії. |
| tar -cvf mybackups/edev.tar /etc/udev | Створює архів edev.tar, що містить файли з каталогу /etc/udev. |
| ls mybackups | Відображає список файлів у директорії mybackups, що допомагає перевірити, чи архів успішно створений. |
| tar -tvf mybackups/udev.tar | Показує вміст архіву udev.tar, дозволяючи переглянути файли без їх розархівування. |
| tar -zcvf mybackups/udev.tar.gz /etc/udev | Створює стиснений архів udev.tar.gz з вмістом каталогу /etc/udev, використовуючи gzip для зменшення розміру файлу. |
| ls -lh mybackups | Відображає вміст директорії mybackups з детальною інформацією про файли. |
| cd mybackups | Переходить до директорії mybackups, що дозволяє виконувати подальші команди в цій папці. |
| tar -xvf udev.tar.gz | Розархівує файл udev.tar.gz, відновлюючи вміст архіву в поточну директорію. |
| ls etc/udev/rules.d | Відображає вміст каталогу rules.d всередині etc/udev, що дозволяє переглянути специфічні правила. |
| tar -rvf udev.tar /etc/hosts | Додає файл /etc/hosts до архіву udev.tar, що дозволяє оновити архів з новими файлами. |
| tar -tvf udev.tar | Показує список файлів в архіві udev.tar, щоб перевірити, які файли були додані. |
| cp /usr/share/dict/words . | Копіює файл words з системного словника в поточну директорію. |
| ls -l words | Відображає детальну інформацію про файл words, включаючи права доступу, власника та розмір. |
| gzip words | Сжимає файл words, створюючи words.gz, що дозволяє зекономити місце на диску. |
| ls -l words.gz | Показує деталі стисненого файлу words.gz, щоб перевірити, чи стиснення пройшло успішно. |
| gunzip words.gz | Розархівує файл words.gz, відновлюючи оригінальний файл words. |
| bzip2 words | Сжимає файл words в формат words.bz2, що забезпечує високу ступінь стиснення. |
| ls -l words.bz2 | Відображає інформацію про стиснений файл words.bz2. |
| bunzip2 words.bz2 | Розархівує файл words.bz2, повертаючи його до оригінального формату. |
| xz words | Сжимає файл words у формат words.xz, який забезпечує ще більш високу ступінь стиснення. |
| ls -l words.xz | Показує деталі стисненого файлу words.xz. |
| unxz words.xz | Розархівує файл words.xz, відновлюючи оригінальний файл. |
| zip words.zip words | Створює ZIP-архів words.zip, що містить файл words. |
| ls -l words.zip | Відображає деталі ZIP-архіву words.zip. |
| unzip -l words.zip | Показує вміст ZIP-архіву words.zip, щоб перевірити, які файли в ньому містяться. |
| rm -r etc | Видаляє каталог etc та його вміст, що може використовуватися для очищення непотрібних файлів. |
| unzip udev.zip | Розархівує файл udev.zip, відновлюючи його вміст у поточну директорію. |
| echo "Hello World" | Виводить текст "Hello World" на екран, зазвичай використовується для тестування. |
| echo "Hello World" > mymessage | Записує "Hello World" у файл mymessage, перезаписуючи його вміст. |
| cat mymessage | Виводить вміст файлу mymessage на екран. |
| echo "Greetings" > mymessage | Перезаписує файл mymessage новим текстом "Greetings". |
| echo "How are you?" >> mymessage | Додає текст до файлу mymessage, не перезаписуючи попередній вміст. |
| find ~ -name "bash" | Шукає всі файли, що містять "bash" у назві, у домашньому каталозі користувача. |
| find /etc -name hosts | Шукає файл з назвою hosts у каталозі /etc. |
| find /etc -name hosts 2> err.txt | Виконує пошук файлу hosts, записуючи всі помилки в err.txt. |
| cat err.txt | Виводить вміст файлу err.txt, що містить помилки, якщо такі були. |
| find /etc -name hosts > std.out 2> std.err | Записує результати пошуку та помилки в окремі файли для подальшого аналізу. |
| cat std.err | Показує вміст файлу std.err, що містить помилки, якщо вони були. |
| cat std.out | Виводить результати пошуку з файлу std.out. |
| find /etc -name hosts > find.out 2>&1 | Записує результати пошуку та помилки в один файл find.out. |
| cat find.out | Показує вміст файлу find.out, де містяться як результати, так і помилки. |
| tr a-z A-Z | Змінює всі літери з маленьких на великі у вхідних даних. |
| this is interesting | Текст, що подається для перетворення. |
| how do I stop this? | Для зупинки вводу в терміналі використовується комбінація Ctrl+D. |
| tr A-Z a-z > myfile | Перетворює всі великі літери з тексту на малі і записує результат у файл myfile. |
| cat myfile | Показує вміст файлу myfile, що дозволяє перевірити результат перетворення. |
| tr a-z A-Z < myfile | Змінює всі малі літери з файлу myfile на великі. |
| cut -d: -f1 /etc/passwd | Показує вміст каталогу /etc сторінками, що дозволяє переглядати довгий список. |

3)(Kochubei S. 9v)



I entered in terminal -tar command and created file, after that i make it archived with bzip

4)(Koshkin I.)

|  |  |
| --- | --- |
| **Команда** | **Що виконує команда?** |
| cmd 1> file | Перенаправляє стандартний вихід команди cmd у файл file. Якщо файл існує, його вміст буде перезаписано. |
| cmd > file | Перенаправляє stdout команди cmd у файл file. |
| cmd 2> file | Перенаправляє стандартний вихід помилок команди cmd у файл file. Якщо файл існує, його вміст буде перезаписано. |
| cmd >> file | Додає стандартний вихід команди cmd у кінець файлу file. |
| cmd &> file | Перенаправляє як стандартний вихід, так і стандартний вихід помилок команди cmd у файл file. |
| cmd > file 2>&1 | Перенаправляє стандартний вихід у файл file, а потім перенаправляє стандартний вихід помилок у той же файл. |
| cmd >> file 2>&1 | Додає стандартний вихід у кінець файлу file і перенаправляє стандартний вихід помилок у той же файл. |
| cmd 2>&1 > /dev/null | Спочатку перенаправляє стандартний вихід помилок у stdout, а потім перенаправляє stdout у /dev/null. |
| cmd 2> /dev/null | Перенаправляє стандартний вихід помилок команди cmd у /dev/null, ігноруючи помилки. |
| cmd1 | cmd2 | Команда cmd1|cmd2 використовується для перенаправлення стандартного виходу команди cmd1 на стандартний вхід команди cmd2. |
| cmd1 2>&1 | cmd2 | Команда cmd1 2>&1|cmd2 виконує перенаправлення стандартного виходу помилок команди cmd1 до стандартного виходу, а потім передає весь вивід на вхід команди cmd2. |

5)(Koshkin I.)

|  |  |  |
| --- | --- | --- |
| **Команда** | **Що виконує команда?** | **Який потік перенаправлення?** |
| $echo "It is a new story." > story | Виводить рядок "It is a new story." у файл story, перезаписуючи його вміст, якщо файл існує. | У команді echo "It is a new story." > story використовується перенаправлення стандартного виходу у файл story, перезаписуючи його вміст, якщо файл існує. |
| $ date > date.txt | Виводить рядок "It is a new story." у файл story, перезаписуючи його вміст, якщо файл існує. | > у команді $ date > date.txt перенаправляє стандартний вихідний потік (stdout) команди date у файл date.txt. |
| $ cat file1 file2 file3 > bigfile | Об'єднує вміст файлів file1, file2 та file3 і записує їх у файл bigfile, перезаписуючи його. | Команда $ cat file1 file2 file3 > bigfile з'єднує вміст файлів file1, file2 і file3 та перенаправляє об'єднаний результат у файл bigfile. |
| $ls -l >> directory | Додає вивід команди ls -l у кінець файлу directory, якщо файл існує. | Команда $ ls -l >> directory додає вміст переліку файлів та директорій із поточного каталогу в розширеному форматі (-l) до кінця файлу directory без перезапису |
| $ sort < file1\_unsorted > file2\_sorted | Сортує вміст файлу file1\_unsorted і записує відсортований вміст у файл file2\_sorted, перезаписуючи його. | Команда $ sort < file1\_unsorted > file2\_sorted зчитує вміст файлу file1\_unsorted, сортує його і записує результат у файл file2\_sorted. Тут < перенаправляє **вхідний потік** з файлу file1\_unsorted, а > перенаправляє **вихідний потік** до file2\_sorted. |
| $ find -name '\*.txt' > file.txt 2> /dev/null | Виконує пошук файлів з розширенням .txt та записує результати у файл file.txt, а помилки ігнорує (перенаправляє у /dev/null). | Команда $ find -name '\*.txt' > file.txt 2> /dev/null шукає всі файли з розширенням .txt у поточному каталозі та записує їхній список у файл file.txt. Оператор 2> /dev/null перенаправляє будь-які повідомлення про помилки у "чорний хід" системи, щоб вони не відображалися і не зберігалися. |
| $ cat file1\_unsorted | sort > file2\_sorted | Команда $ cat file1\_unsorted | sort > file2\_sorted читає вміст файлу file1\_unsorted, сортує його і записує результат у файл file2\_sorted, перезаписуючи його вміст. | Об'єднує вміст файлу file1\_unsorted і передає його на вхід команди sort, результат записується у файл file2\_sorted. |
| $ cat myfile | grep student | wc -l | Команда $ cat myfile| grep student|wc -l читає вміст файлу myfile, фільтрує рядки, що містять слово "student", і підраховує кількість таких рядків. | У команді $ cat myfile | grep student | wc -l використовується конвеєрне перенаправлення (|), яке передає стандартний вихідний потік (stdout) однієї команди (cat myfile і grep student) як стандартний вхідний потік (stdin) для наступної команди (grep student і wc -l). |