Хід роботи

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 | Перенаправляє стандартний потік виведення (STDOUT) команди cmd у файл file, створюючи або перезаписуючи файл. |
| cmd > file | Перенаправляє стандартний потік виведення (STDOUT) у файл file, створюючи або перезаписуючи файл. |
| cmd 2> file | Перенаправляє потік помилок (STDERR) команди cmd у файл file, створюючи або перезаписуючи файл. |
| cmd >> file | Перенаправляє стандартний потік виведення (STDOUT) команди cmd у файл file, додаючи вивід до файлу, а не перезаписуючи його. |
| cmd &> file | Перенаправляє обидва потоки (STDOUT і STDERR) у файл file, перезаписуючи файл. |
| cmd > file 2>&1 | Спочатку перенаправляє стандартний потік виведення (STDOUT) у файл file, а потім перенаправляє потік помилок (STDERR) туди ж, куди вже спрямовано STDOUT (тобто у файл file), перезаписуючи файл. |
| cmd >> file 2>&1 | Спочатку перенаправляє стандартний потік виведення (STDOUT) у файл file, додаючи вивід до файлу. Потім перенаправляє потік помилок (STDERR) у той самий файл, додаючи помилки до кінця файлу. |
| cmd 2>&1 > /dev/null | Спочатку перенаправляє потік помилок (STDERR) туди, куди спрямовано стандартний потік виведення (STDOUT), тобто у поточний термінал. Потім перенаправляє стандартний потік виведення (STDOUT) в /dev/null, тим самим відкидаючи його. В результаті помилки все одно будуть виведені в термінал. |
| cmd 2> /dev/null | Перенаправляє потік помилок (STDERR) у /dev/null, відкидаючи помилки. Стандартний потік виведення (STDOUT) залишається без змін. |
| cmd1 | cmd2 | Передає стандартний потік виведення (STDOUT) команди cmd1 як вхідний потік (STDIN) для команди cmd2. |
| cmd1 2>&1 | cmd2 | Перенаправляє обидва потоки (STDOUT і STDERR) команди cmd1 до стандартного вхідного потоку (STDIN) команди cmd2. |

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 | Записує текст у файл, перезаписуючи його. | STDOUT. |
| $ date > date.txt | Записує поточну дату у файл, перезаписуючи його. | STDOUT. |
| $ cat file1 file2 file3 > bigfile | Об'єднує вміст трьох файлів у один файл, перезаписуючи його. | STDOUT. |
| $ls -l >> directory | Додає список файлів у кінець файлу. | STDOUT . |
| $ sort < file1\_unsorted > file2\_sorted | Сортує дані з одного файлу та записує в інший. | STDIN і STDOUT. |
| $ find -name '\*.txt' > file.txt 2> /dev/null | Записує знайдені файли у файл, помилки відкидає. | STDOUT і STDERR. |
| $ cat file1\_unsorted | sort > file2\_sorted | Сортує вміст файлу та записує у новий файл. | STDOUT (через pipe). |
| $ cat myfile | grep student | wc -l | Рахує рядки зі словом "student". | Pipe між командами. |

Контрольні запитання

Виконував студент групи КСМ — 23А Величко Ярослав

1. Порівняльна характеристика процесів стискання та архівування

- Стискання: Це процес зменшення розміру файлів шляхом видалення або оптимізації даних. Стискання працює за допомогою різних алгоритмів, таких як Gzip, Bzip2, XZ тощо. Стиснутий файл може бути відновлений у початкову форму без втрати даних (без втрат).

- Архівування: Це процес об’єднання кількох файлів та каталогів в один файл без зміни розміру даних. У Linux архівування найчастіше виконується за допомогою команди `tar`. Архівування зручне для перенесення та зберігання структур каталогів.

Відмінність: Стискання зменшує розмір окремих файлів, а архівування об'єднує файли та каталоги в один файл, який можна стиснути.

2. Програми для стискання та архівування файлів та каталогів у Linux

Окрім стандартних програм, таких як `tar`, `gzip`, `bzip2`, можна використовувати інші утиліти:

- 7zip (p7zip) — Використовує алгоритм LZMA для високого стискання. Команда `7z` дозволяє стискати, архівувати та витягувати файли.

- Zstd — Забезпечує високу швидкість стискання із задовільним рівнем стиснення. Швидкість його роботи часто вища, ніж у Gzip, а розмір стиснених файлів менший.

- RAR — Пропонує високу ступінь стиснення та багатий функціонал для архівування, включно з розподілом архівів на частини.

3. Порівняння алгоритмів стискання

- Gzip — Використовує алгоритм DEFLATE, забезпечує швидке стискання та декомпресію, але не таке ефективне, як XZ.

- Bzip2 — Застосовує алгоритм Burrows-Wheeler, забезпечує кращий рівень стиснення за Gzip, але працює повільніше.

- XZ — Використовує алгоритм LZMA, забезпечує високе стиснення, однак повільне. Підходить для файлів, які не часто потребують декомпресії.

- Zstd — Забезпечує баланс між швидкістю та рівнем стиснення, тому є одним з найкращих варіантів для швидкого та ефективного стискання.

Найшвидший і найефективніший алгоритм залежить від потреби в балансі між швидкістю та ступенем стиснення: для швидкого стиснення підходить Zstd, для високого стиснення — XZ.

4. Програмні засоби для стискання та архівування на мобільному телефоні

- ZArchiver — Підтримує різні формати архівів (zip, rar, 7z), дозволяє стискати та витягувати файли на мобільних пристроях.

- RAR for Android — Додаток для роботи з архівами RAR, ZIP, 7Z, що дозволяє створювати архіви та розпаковувати їх.

- X-plore File Manager — Багатофункціональний файловий менеджер, що також підтримує роботу з архівами.

5. Програмні засоби для стискання та архівування в Windows

- WinRAR — Підтримує широкий набір форматів архівів, забезпечує високий рівень стиснення та багатий функціонал.

- 7-Zip — Безкоштовний інструмент з високим ступенем стиснення. Підтримує формати 7z, zip, rar.

- WinZip — Відомий архіватор, зручний у використанні для zip-архівів.

Порівняння: 7-Zip забезпечує кращий рівень стиснення, ніж WinRAR, тоді як WinRAR забезпечує багатий функціонал, включаючи розподіл архівів на частини.

6. Стискання та архівування для резервування даних та задач системного адміністрування

Стискання та архівування є корисними для резервування даних, оскільки зменшує обсяг резервних копій, що зручно для зберігання та відновлення.

Інші задачі системного адміністрування:

-Переміщення даних — Архіви зручні для переміщення даних між серверами.

- Захист від видалення — Архівовані копії можуть зберігатися як резервні, навіть якщо вихідні файли випадково видалено.

7. Призначення директорії файлу /dev/null

`/dev/null` — спеціальний файл, який відкидає будь-які дані, які до нього записуються, і повертає "порожній" результат при читанні. Його часто використовують для перенаправлення непотрібного виходу команд, щоб уникнути засмічення консолі.