***Лабораторна робота №7***

***«Пошук та архівація файлів»***

***Пошук файлів***

Незважаючи на те, що існують угоди з організації файлової системи для Unix-подібних операційних систем, пошук необхідного файлу може бути досить трудомістким.

У даній лабораторній роботі ми розглянемо кілька утиліт призначених для пошуку файлів в системі.

Команда ***find*** здійснює рекурсивний обхід дерева файлової системи, і здійснює пошук файлу, грунтуючись на одному або декількох атрибутах.

Табл. 1. Приклади опцій команди ***find***

|  |  |
| --- | --- |
| **Опція** | **Опис** |
| ***-name*** | ім'я файлу |
| ***-type*** | тип файлу |
| ***-newer*** | файли з меншою датою модифікації ніж у заданого файлу |
| ***-mtime*** | дата модифікації файлу |
| ***-size*** | розмір файлу |
| ***-exec*** | виконання над знайденими файлами зазначеної команди |
| ***-delete*** | видалення знайдених файлів |

Деякі опції можуть приймати аргумент ***n*** зі знаком. ***+n*** для значень більше ніж n, ***-n*** для значень менше ніж n.

***$ find ~ -name "\*.txt" -type f -mtime -3***

Наведена вище команда зробить пошук всіх звичайних файлів в домашньому каталозі користувача, з розширенням txt і датою модифікації менше 3-х днів.

Команда ***find*** також підтримує об'єднання шаблонів пошуку в логічні вираження за допомогою опцій-операторів ***-or***, ***-and*** і ***-not***.

***$ find ~ \( -size + 100M \) -and \( -not -type d \)***

Наведена вище команда дозволяє отримати список файлів, які не є директорією розміром більше 100 Мегабайт.

Альтернативним способом знайти Вас файл на ім'я є використання команди ***locate***.

***$ locate <ім’я\_файлу>***

Пошук в цьому випадку відбувається не по дереву файлової системи, а по спеціальній базі імен файлів, яка періодично оновлюється. Для оновлення бази використовується команда ***updatedb***. За замовчуванням ***locate*** не перевіряє чи існують файли, знайдені в базі імен на даний момент.

Команда ***whereis*** дозволяє виконати пошук розташування виконуваних файлів, файлів з вихідним кодом і файлів довідкових сторінок для обраної команди. Форматом виведення команди можна керувати за допомогою спеціальних опцій.

***$ whereis <команда>***

Команда ***which*** виконує пошук шляху до виконуваного файлу для певної команди на основі вмісту змінної PATH.

***$ which <команда>***

Для пошуку рядків у файлі містять певний шаблон можна використовувати команду ***grep***. Якщо команді не передається список вхідних файлів, то пошук здійснюється в стандартному вхідному потоці.

***$ cat /var/log/mylog | grep 'warning'***

Табл. 2. Приклади опцій команди ***grep***

|  |  |
| --- | --- |
| **Опція** | **Опис** |
| ***-n*** | виведення номеру рядка, що містить шаблон |
| ***-c*** | виведення кількості рядків, що містять зразок |
| ***-i*** | ігнорувати регістр символів |
| ***-v*** | виведення всіх рядків, що не містять шаблон |

Утиліту ***grep*** часто використовують в зв'язці з іншими командами, передаючи їй дані на стандартний потік введення. Можливе застосування регулярних виразів при написанні шаблону для пошуку.

Команда ***xargs*** перетворює рядки, що надходять їй на вхід, в аргументи для заданої команди.

***$ find ~/mydir -type f -name filename | xargs ls –l***

***Архівація***

Для скорочення обсягу, займаного призначеними для довготривалого зберігання або передачі файлами, застосовують різні утиліти стиснення.

Утиліта ***gzip*** призначена для стиснення одного або декількох файлів. При упаковці оригінальні файли замінюються файлами з розширенням ***.gz***. Перенаправити закодовану інформацію на стандартне виведення зі збереженням вихідного файлу можна за допомогою опції ***-c***.

***$ gzip file1 file2 file3***

Розпакування файлів може здійснюватися c використанням опції ***-d*** або утилітою ***gunzip***. Отримати інформацію про ступінь стиснення файлу можна за допомогою опції ***-l***. За допомогою числових опцій - [***1-9***] можна регулювати ступінь стиснення.

***$ gzip -l file1.gz***

Існує схожа на gzip утиліта, яка використовує для стиснення перетворення Барроуза-Уілера, ***bzip2***. Вона має схожий синтаксис і опції. Стислі файли в цьому випадку мають розширення ***.bz2***.

Одночасно зі стисненням, при роботі з файлами застосовується операція об'єднання декількох файлів в один архів. Архівація часто застосовується при створенні резервних копій.

Утиліта ***tar*** призначена для упаковки безлічі файлів в архів, і їх вилучення. В якості аргументів команда приймає файли, що архівуються.

Табл. 3. Приклади опцій команди ***tar***

|  |  |
| --- | --- |
| **Опція** | **Опис** |
| ***-r*** | додавання файлів до архіву |
| ***-c*** | створення нового архіву |
| ***--delete*** | видалення файлів з архіву |
| ***-t*** | виведення вмісту архіву |
| ***-x*** | витяг файлів з архіву |
| ***-f*** | використання архівного файлу |
| ***-v*** | виведення списку оброблюваних файлів |

Створення нового архівного файлу:

***$ tar -cvf myarchive.tar file1 file2 file3***

Перегляд вмісту архівного файлу:

***$ tar -tf myarchive.tar***

Утиліта ***zip*** виконує одночасно функції стиснення та архівації. Підсумковий файл буде мати розширення ***.zip***. Для розміщення в архів директорій разом з вкладеними файлами використовується прапор ***-r***.

***$ zip -r archive.zip <ім’я\_діректорії>***

Для розпаковування архіву використовується утиліта ***unzip***. Для отримання додаткової інформації про видобуті файли використовується опція ***-l***.

***$ unzip -l archive.zip <шлях\_для\_розпакування>***

***Завдання***

1. Ознайомтесь з роботою команд, наведених в тексті лабораторної роботи. Отримайте для них сторінки довідкового керівництва.
2. За допомогою утиліт ***find*** і ***wc*** отримаєте інформацію про кількість файлів в домашньому каталозі користувача.
3. Отримайте імена всіх файлів, які не є символічними посиланнями або каталогами, і помістіть їх в файл ***filelist1.txt***.
4. За допомогою команд ***find***, ***xargs*** і ***ls*** отримайте повну інформацію про атрибути файлів домашнього каталогу розмір яких перевищує 5 Кілобайтів і помістіть результат в файл ***filelist2.txt***.
5. За допомогою команди ***locate*** отримаєте список імен файлів що містять в назві рядок "***bash***".
6. Для команд, які використовуються в попередніх підпунктах, отримаєте розташування файлів довідкових посібників.
7. З файлу ***passwd\_example*** з минулої лабораторної роботи за допомогою утиліти ***grep***, отримайте записи користувачів з домашніми каталогами в папці ***home***, із зазначенням номерів рядків. Помістіть результат в файл ***filelist3.txt***.
8. Стисніть файл ***filelist1.txt*** зі збереженням вихідного файлу, утилітою ***gzip*** з різними ступенями стиснення. Для одержаних файлів дізнайтеся відсоток коефіцієнту стиснення.
9. Стисніть файл ***filelist1.txt*** зі збереженням вихідного файлу, утилітою ***bzip2*** з різними ступенями стиснення.
10. Порівняйте результати для утиліт ***gzip*** і ***bzip2***, подивившись розміри отриманих стиснених файлів.
11. Створіть архів ***tar***, що містить файли ***filelist1.txt***, ***filelist2.txt*** та ***filelist3.txt***.
12. Додайте до створеного архіву файл ***passwd\_example***.
13. Створіть архів ***zip***, що містить файли ***filelist1.txt***, ***filelist2.txt***, ***filelist3.txt*** і ***passwd\_example***.
14. Порівняйте розміри одержані архівів.
15. Розпакуйте архіви, створені командами ***tar*** та ***zip***.

***Контрольні питання***

1. Які утиліти для пошуку файлів ви знаєте?

2. Як дізнатися розташування бінарних файлів певної команди?

3. Де здійснює пошук файлів команда ***locate***?

4. Як отримати номери рядків в файлі, що не містять шуканого шаблона?

5. Як додати файли до архіву ***tar***, отримати список файлів в архіві?

6. Як витягти файли з архіву ***tar***, ***zip***?