Практична робота № 14. 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 цій лабораторній роботі ми розглянемо кілька утиліт, призначених для пошуку файлів у системі.

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

Таблиця 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, які більше ніж n, ***-n*** для значень, які менше ніж n.

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

Наведена вище команда зробить пошук всіх звичайних файлів у 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 файл на ім'я є використання команди ***locate***.

```
$ locate <ім'я_файлу>
```

Пошук у цьому випадку відбувається не за деревом файлової системи, а за спеціальною базою імен файлів, яка періодично оновлюється. Для оновлення бази використовується команда ***updatedb***. За замовчуванням ***locate*** не перевіряє, чи існують файли, знайдені в базі імен на цей 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, що 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'
```

Таблиця 2.

Приклади опцій команди *grep*

Опція	*Опис*
-n	виведення номера 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 у зв'язці з іншими командами, передаючи їй дані на стандартний потік введення. Можливе застосування регулярних виразів у 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. У результаті упаковки 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 з 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, під час роботи з файлами застосовується операція об'єднання декількох файлів в один архів. Архівація часто застосовується у створенні резервних копій.

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

Таблиця 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 зі вкладеними файлами використовується прапор ***-r***.

```
$ zip -r archive.zip <ім'я_директорії>
```

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 у тексті лабораторної роботи. Отримайте для них сторінки довідкового керівництва.
2. За допомогою утиліт ***find*** і ***wc*** отримаєте інформацію про кількість файлів у домашньому каталозі користувача.
3. Отримайте імена всіх файлів, які не є символічними посиланнями або каталогами, і помістіть їх у 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 у файл ***filelist2.txt***.
5. За допомогою команди ***locate*** отримаєте список імен файлів, що містять у 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 у 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 у 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***?