***Лабораторна робота №3***

***«Перенаправлення стандартних потоків даних. Використання конвейєрів для виконання команд»***

***Стандартні потоки введення-виведення***

З кожною програмою, що запускається з командного рядка Unix, пов'язані три стандартних потоки даних:

* стандартний потік введення (stdin);
* стандартний потік виведення (stdout);
* стандартний потік помилок (stderr).

1. **Стандартний потік введення**

Команди, які потребують вхідних даних, зазвичай читають інформацію зі стандартного потоку введення. Наприклад, команда ***wc*** підраховує кількість рядків, слів та символів у вхідних даних. Якщо запустити цю команду без аргументів, то ***wc*** чекатиме вхідні дані з терміналу (щоб закінчити введення даних, потрібно натиснути комбінацію клавіш ***Ctrl-D***):

***$ wc***

***two words***

***<Ctrl-D>***

1 2 10

В даному прикладі команда ***wc*** прочитала введений користувачем текст зі стандартного потоку введення (куди користувач ввів текст «two words»). За замовчуванням цей потік з'єднаний з терміналом (з клавіатурою) користувача, але допускається його перенаправлення. Щоб зв'язати дані стандартного вхідного потоку з довільним файлом, можна використовувати операцію перенаправлення «***<***», наприклад:

***$ wc < /etc/passwd***

28 37 1052

В даному випадку команда wc вже не вимагає введення з клавіатури, оскільки вона вже отримала вхідні дані з файлу ***/etc/passwd***. Зауважимо, що дана команда може мати практичне застосування – перша цифра означає кількість рядків у файлі ***/etc/passwd***, що відповідає кількості користувачів, зареєстрованих в системі.

1. **Стандартний потік виведення**

Стандартний потік виведення - це потік, куди програми записують вихідні дані. У попередньому прикладі команда ***wc*** виводила результат (три числа) саме в цей потік. Так само працюють і більшість інших неінтерактивних команд (включаючи ***echo***, ***pwd*** і ***ls***).

Подібно стандартному потоку введення вихідний потік спочатку пов'язаний з терміналом і також допускає перенаправлення. Для зв'язування стандартного потоку виведення з файлом використовується операція «***>***», наприклад:

***$ ls > filelist.txt***

У цьому прикладі команда ***ls***, замість того, щоб вивести список файлів на екран, записала його в файл з ім'ям «***filelist.txt***». При цьому, якщо файл з таким ім'ям не існував, він буде створений, а інакше його старий вміст буде втрачений.

Існує й інша можливість перенаправлення виведення, коли нові вихідні дані будуть дописані в кінець існуючого файлу. Для цього використовується операція «***>>***». У наступному прикладі поточні дата і час будуть дописані в кінець файлу з ім'ям «***dates.txt***»:

***$ date >> dates.txt***

1. **Стандартний потік помилок**

Повідомлення про помилки виводяться в стандартний потік помилок. Наприклад, нехай виконується спроба отримати список файлів в каталозі без відповідних прав доступу:

***$ ls -l /home/ftp/bin/***

ls: /home/ftp/bin/: Access denied

В даному випадку команда ***ls*** вивела повідомлення у стандартний потік помилок.

Перенаправлення стандартного потоку помилок здійснюється не так просто, як стандартного виведення. Щоб перенаправити стандартний потік помилок, потрібно вказати його дескриптор файлу. Програма може здійснення виведення в будь-який з кількох нумерованих файлових потоків. Перші три з них ми згадали як стандартні потоки введення, виведення і помилок. Командна оболонка посилається на них як на файлові дескриптори 0, 1 і 2 відповідно.

Командна оболонка підтримує синтаксис перенаправлення файлів з використанням номерів файлових дескрипторів. Так як стандартному потоку помилок відповідає файловий дескриптор 2, ми можемо перенаправити його, як показано нижче:

***$ ls -l /home/ftp/bin/ 2> last-error.txt***

Операції перенаправлення введення-виведення можна комбінувати, наприклад:

***$ wc < /etc/passwd 2> errors.txt > result.txt***

У даному випадку стандартний потік помилок буде перенаправлений у файл ***errors.txt***, а стандартний потік виведення – у файл ***result.txt***.

**Перенаправлення стандартних потоків виведення і помилок в один файл**

Іноді необхідно зберегти все виведення команди в один файл. Для цього потрібно перенаправити відразу два потоки: виведення і помилок. Зробити це можна двома способами.

Перший (традиційний) працює в старих версіях командної оболонки:

***ls -l /etc/passwd > output.txt 2>&1***

Тут виконується два перенаправлення. Спочатку - перенаправлення стандартного потоку виведення у файл ***output.txt***, а потім, з використанням нотації ***2>&1***, - перенаправлення файлового дескриптора 2 (стандартний потік помилок) у файловий дескриптор 1 (стандартний потік виведення).

Сучасні версії bash підтримують другий, більш простий метод виконання перенаправлення цього виду:

***$ ls -l / etc/passwd &> output.txt***

В даному прикладі використовується єдиний оператор ***&>***, що перенаправляє стандартний потік виведення і стандартний потік помилок у файл ***output.txt***.

***Команда cat як універсальна команда для створення, копіювання та об’єднання файлів***

Команда ***cat*** часто використовується для створення файлів (хоча можна скористатися й командою ***touch***). По команді cat на стандартний вивід (тобто на екран) виводиться вміст зазначеного файлу (або декількох файлів, якщо їхні імена послідовно задати як аргументи команди). Якщо вивід команди ***cat*** перенаправляти у файл, то можна одержати копію якогось файлу:

***$ cat file1 > file2***

Первісне призначення команди cat саме й припускало перенаправлення виведення, тому що ця команда створена для конкатенації, тобто об'єднання декількох файлів в один:

***$ cat file1 file2 ... file > new-file***

Саме можливості перенаправлення введення та виведення цієї команди й використовуються для створення нових файлів. Для цього на вхід команди ***cat*** направляють дані з потоку стандартного введення (тобто із клавіатури), а виведення команди – у новий файл:

***$ cat > newfile***

Після того, як ви надрукуєте все, що хочете, натисніть комбінацію клавіш ***<Ctrl>+<D>*** або ***<Ctrl>+<C>***, і все, що ви ввели, буде записано в newfile. Звичайно, у такий спосіб створюються, в основному, короткі текстові файли.

***Конвейєри***

Існує інший корисний спосіб перенаправлення вводу виведення - конвеєри команд. Операція «***|***» дозволяє перенаправити стандартний потік виведення однієї команди на стандартний потік введення іншої команди. Наприклад:

***$ ls -l /etc | less***

У цьому прикладі команда ***ls*** виводить довгий список файлів в каталозі ***/etc***, ці дані потрапляють на вхід програми ***less***, яка дозволяє перегортати текст за допомогою клавіш управління курсором. Так здійснюється об'єднання» двох незалежних команд в один «конвеєр.

Конвеєри часто використовуються для виконання складних операцій з даними. Вони дозволяють об'єднати разом кілька команд. Часто команди, використовувані таким способом, називають ***фільтрами***. Фільтри приймають введення, змінюють його певним чином і виводять результат.

Деякі команди-фільтри:

1. ***$ sort [<файл>]*** – сортування вхідних даних за алфавітом;
2. ***$ uniq [<файл>]*** – пошук або видалення рядків, що повторюються. Використання команди без ключа видалить всі дублікати та виведе рядки без них назалежно від того, чи є дублікати у певних рядків чи ні. Використання команди з ключем ***–d*** навпаки виведе список рядків, що мають дублікати;
3. ***$ wc <файл>***
4. ***$ grep <шаблон> [<файл…>]***- пошук рядків, що відповідають шаблону;
5. ***$ head (tail)*** – виведення перших (останніх рядків файлу).

***Використання фільтрів у конвеєрах***

Розглянемо більш складний приклад використання конвеєрів. Припустимо, що нам потрібно знайти всі файли в списку програм, які мають в своєму імені послідовність символів «***zip***». Результати такого пошуку можуть підказати нам, які програми в системі мають відношення до стискання файлів. Такі програми можуть знаходитись у каталогах ***/bin*** та ***/usr/bin***.

Дану задачу можна виконати за допомогою наступної послідовності дій:

***$ ls /bin /usr/bin | sort | uniq | grep zip***

bunzip2

bzip2

gunzip

gzip

unzip

zip

zipcloak

zipgrep

zipinfo

zipnote

zipsplit

У даному випадку були використані конвеєри за певною послідовністю дій:

1. Виведення вмісту каталогів (команда ***ls***);
2. Сортування списку вмісту каталогів в алфавітному порядку, оскільки при виведенні спочатку знаходився вміст ***/bin***, а потім ***/usr/bin*** (команда ***sort***);
3. Видалення у відсортованому списку записів-дублікатів, оскільки у каталогах ***/bin*** та ***/usr/bin*** можуть знаходитись каталоги або файли з однаковими назвами (команда ***uniq***);
4. Пошук у отриманому списку назв каталогів або файлів, які мають у собі послідовність симолів «***zip***» (команда ***grep***).

***Завдання***

1. Створити порожній файл ***first.txt***.

2. Додати рядок тексту в кінець файлу «***Hello, world***» шляхом перенаправлення виводу.

3. Переглянути вміст файлу ***first.txt***.

4. Скопіювати вміст файлу ***first.txt*** в файл ***1.txt***.

5. Створити новий файл ***second.txt*** та записати у нього декілька рядків на власний розсуд.

6. Створити файл ***home.txt***, записавши у нього назви домашніх каталогів всіх користувачів.

7. Підрахувати кількість користувачів системи, що мають домашній каталог, як кількість рядків файлу ***home.txt***.

8. Об’єднати файли ***first.txt***, ***second.txt*** та ***home.txt*** в один з ім’ям ***big.txt***.

9. Переглянути файл ***big.txt***, переконавшись, що він містить рядки з перерахованих файлів.

10. За допомогою конвеєрів знайдіть у файлі ***home.txt*** назви каталогів користувачів, що є студентами Вашої групи, які мають логіни, що складаються з цифр, та виведіть їх у відсортованому порядку.

***Контрольні питання***

1. Як перенаправити стандартний потік введення?
2. Як перенаправити стандартний потік виведення у файл:

* з його перезаписом;
* з додаванням у кінець існуючого файлу.

1. Як перенаправити стандартний потік помилок у файл?
2. Як перенаправити стандартні потоки виведення та помилок в один файл?
3. Для чого призначена команда ***cat***? Наведіть приклади різних використань команди.
4. Для чого призначені конвеєри та як запустити їх на виконання?
5. Наведіть приклади команд-фільтрів. Для чого вони призначені?