

## Практична робота № 13. Обробка текстових даних. Користувацьке оточення

---

### Обробка текстових даних

В операційних системах текстові файли використовуються для зберігання даних різного призначення. Як приклади можна виділити файли конфігурації, системні журнали, файли з вихідним кодом програм та ін. Для роботи з цими даними розроблено велику кількість утиліт.

Для об'єднання вмісту декількох файлів і виведення його в стандартний канал виведення або в файл використовується команда **cat**, яка вже була розглянута в минулих лабораторних роботах. Ключ **-n** цієї команди виробляє нумерацію рядків під час виведення.

Серед інших вже знайомих команд обробки тексту слід виділити такі:

- **less** – дозволяє організувати пострінкову роботу з великим набором даних;

- **sort** – виконує сортування даних, що поступають їй на вхід. Використання додаткових опцій дозволяє провести сортування за однією з полів згрупованих даних. Наприклад: **\$ ls -l ~ | sort -n -k 5**. Ця команда виробляє числову (опція **-n**) сортування отриманих від команди **ls** даних за п'ятим стовпчиком (опція **-k**);

- **uniq** – видаляє сусідні повторювані рядки у файлі. Опції команди дозволяють також знайти неунікальні рядки і підрахувати кількість входжень кожного рядка. Цю команду часто використовують спільно з командою **sort**;

- **head** і **tail** – використовуються для відображення обраного числа рядків на початку або в кінці файлу. За замовчуванням число рядків дорівнює 10. Змінити кількість виведених рядків можна за допомогою опції **-n**;

- **wc** – використовується для підрахунку рядків, слів, байт і символів у файлі.

Розглянемо інші, ще незнайомі вам команди обробки текстових даних.

**cut** – виконує фільтрацію тексту за стовпцями. Як опцію команда приймає номер поля (**-f**), роздільник (**-d**) та ін. Наприклад:

```
$ cut -d: -f 1 file1
```

Наведена вище команда з множини стовпців, розділених символом «:» у файлі **file1**, вибирає перший.

**paste** – використовується для об'єднання декількох файлів;

**join** – може розглядатися як команда, споріднена команді **paste**. Ця потужна утиліта дозволяє об'єднувати два файли за загальним полем, що є спрощеною версією реляційної бази даних.

Команда **join** оперує тільки двома файлами і об'єднує тільки ті рядки, які мають загальне поле (зазвичай числове), результат об'єднання виводиться на стандартний потік виведення. Файли, що об'єднуються, повинні бути відсортовані за ключовим полем.

Наприклад, існує 2 файли:

```
File: 1.data
100 Shoes
200 Laces
300 Socks
File: 2.data
100 $40.00
200 $1.00
300 $2.00
```

Виконання команди **join**:

```
$ join 1.data 2.data
100 Shoes $40.00
200 Laces $1.00
300 Socks $2.00
```

### Порівняння файлів

**cmp** – порівнює вміст двох файлів побайтно:

```
$ cmp file1 file2
```

Якщо файли повністю збігаються, ця команда завершує свою роботу, а якщо файли розрізняються, видає номер рядка й номер байта в рядку, де має місце перша розбіжність. Звичайно, інформації, що видається командою **cmp**, обмаль для того, щоб прийняти, наприклад, рішення про те, який із двох файлів для нас важливіший.

**diff** – порівняння двох файлів з одержанням повної інформації про розбіжності у файлах. Для одержання інформації достатньо вказати команді, які саме файли порівнювати:

```
$ diff file1 file2
```

Інформація про виявлені розбіжності буде видана на стандартний потік виведення, але її можна перенаправити у файл:

```
$ diff file1 file2 > diff12
```

### Середовище оточення

Під час роботи з командною оболонкою визначено набір змінних, що описують поточний сеанс взаємодії користувача з системою, що називається **оточенням (environment)**.

Змінні оточення доступні одночасно декільком процесам.

Завантаження змінних оточення з конфігураційних файлів відбувається у результаті запуску командного інтерпретатора. На додаток до змінних оточення, оболонка так само зберігає псевдоніми і функції оболонки.

Список всіх встановлених змінних можна отримати, використовуючи команди **env** або **set** без опцій і аргументів.

Установка нових та зміна значення існуючих змінних середовища оточення здійснюється шляхом експортування (поміщення у середовище):

```
$ export <змінна>=<значення>
```

Дізнатися значення конкретної змінної можна також за допомогою команди:

```
$ echo $<змінна>
```

Щоб видалити змінну, використовується команда **unset**.

Таблиця 1.

#### Деякі стандартні змінні середовища оточення

Ім'я	Значення
<i>UID</i>	Містить числовий ідентифікатор поточного користувача. Ініціалізується при запуску оболонки
<i>HOME</i>	Домашній каталог поточного користувача
<i>PATH</i>	Список каталогів, розділених двокрапкою, в яких командна оболонка виконує пошук файлу, в разі якщо в команді не заданий його шлях
<i>PS1</i>	Формат рядка-запрошення
<i>PWD</i>	Поточний каталог
<i>TERM</i>	Тип терміналу, що використовується
<i>HOSTNAME</i>	Мережеве ім'я комп'ютера

#### Налаштування зовнішнього вигляду рядка-запрошення

За замовчуванням рядок запрошення до вводу має такий вигляд (у Debian, в інших дистрибутивах він може відрізнятись):

```
<Ім'я_користувача>@<Ім'я_хоста>: <Поточний_робочий_каталог>$
```

Форма запрошення до введення визначається в змінній оточення **PS1** (скорочено від prompt string 1 – рядок запрошення 1). Побачити вміст змінної **PS1** можна за допомогою команди **echo**:

```
$ echo $PS1
|u@|h: |w|$
```

Символи, що екрануються слешем, є спеціальними символами. У табл. 2 наведений неповний список символів, які командна оболонка інтерпретує спеціальним чином у рядку запрошення.

Таблиця 2.

**Екрановані послідовності, що використовуються у рядку запрошення**

<i>Послідовність</i>	<i>Значення, що відображується</i>
<code>\a</code>	дзвінок. Змушує комп'ютер видавати звуковий сигнал
<code>\d</code>	поточна дата у форматі: день тижня, місяць, число; наприклад, «Mon May 26»
<code>\h</code>	ім'я хоста локальної машини мінус ім'я домена
<code>\H</code>	повне ім'я хоста
<code>\j</code>	число завдань, що діють в поточному сеансі
<code>\l</code>	ім'я поточного пристрою терміналу
<code>\n</code>	символ переведення рядка
<code>\r</code>	повернення каретки
<code>\s</code>	ім'я програми командної оболонки
<code>\t</code>	поточний час в 24-годинному форматі
<code>\T</code>	поточний час в 12-годинному форматі
<code>\u</code>	ім'я користувача
<code>\w</code>	ім'я поточного робочого каталогу
<code>\W</code>	остання частина в імені поточного робочого каталогу
<code>\\$</code>	виводить символ \$, якщо користувач не є суперкористувачем, в іншому випадку виводить символ #
<code>!</code>	номер поточної команди в історії
<code>\#</code>	число команд, введених в поточному сеансі командної оболонки

Маючи список спеціальних символів, можна спробувати змінити оформлення запрошення. Для початку можна зберегти вихідне визначення, щоб його можна було відновити пізніше. Для цього скопіюємо значення змінної **PS1** в іншу змінну:

```
$ ps1_old="$PS1"
```

Тут створюється нова змінна з ім'ям `ps1_old`, і їй присвоюється значення змінної `PS1`. Це дозволить вам в будь-який момент відновити вихідне оформлення запрошення, виконавши зворотню процедуру:

```
$ PS1="$ps1_old"
```

Тепер можна спробувати змінити зовнішній вигляд рядка запрошення. Наведемо декілька прикладів:

- `$ PS1=` – порожній рядок запрошення;
- `$ PS1="\$ "` – рядок виводить тільки символ `$` (або `#` для суперкористувача);
- `$ PS1="\a\$ "` – звуковий сигнал під час кожного виведення рядка запрошення та символ `$` (або `#`);
- `$ PS1="\t \h \$ "` – виводить час та ім'я хоста;
- `$ PS1"< \u@ \h \w> \$ "` – виводить ім'я користувача, ім'я хоста, поточний каталог. Все це знаходиться у трикутних дужках. Завершується рядок символом `$` (або `#`).

### Оформлення кольору рядка запрошення

Кольором символів можна управляти, посилаючи емулятору терміналу екрановані послідовності ANSI всередині потоку символів, призначених для виведення на екран. Екрановані послідовності не виводяться на екран; вони інтерпретуються терміналом як інструкції.

Для включення недрукованих символів використовуються послідовності `[\i \j]`. Екрановані послідовності ANSI починаються з вісімкового коду 033 (код, що генерується клавішею **ESC**), за яким іде необов'язковий атрибут символу і інструкція. Наприклад, ось як виглядає код, що визначає текст як простий (атрибут = 0), чорного кольору:

```
\033[0;30m.
```

У табл. 3 перераховані підтримувані кольори тексту.

Таблиця 3.

### Екрановані послідовності, що використовуються для визначення кольору тексту

Послідовність	Колір
<code>\033[0;30m</code>	чорний
<code>\033[0;31m</code>	червоний
<code>\033[0;32m</code>	зелений
<code>\033[0;33m</code>	коричневий
<code>\033[0;34m</code>	синій

Закінчення таблиці

<code>\033[0;35m</code>	пурпуровий
<code>\033[0;36m</code>	бірюзовий
<code>\033[0;37m</code>	світло-сірий
<code>\033[1;30m</code>	темно-сірий
<code>\033[1;31m</code>	світло-червоний
<code>\033[1;32m</code>	світло-зелений
<code>\033[1;33m</code>	жовтий
<code>\033[1;34m</code>	світло-синій
<code>\033[1;35m</code>	світло-пурпуровий
<code>\033[1;36m</code>	світло-бірюзовий
<code>\033[1;37m</code>	білий

Наприклад, для того, щоб пофарбувати рядок запрошення у червоний колір, потрібно зробити таке:

```
$ PS1="\[\033[0;31m\]<\u@\h \w>\$ "
```

Однак у нашому випадку червоним кольором буде фарбуватись і весь текст, який буде введений з клавіатури. Для усунення цього ефекту потрібно додати ще одну екрановану послідовність в кінець визначення запрошення – цим ми повідомимо емулятору термінала, що той повинен відновити нормальний колір:

```
$ PS1="\[\033[0;31m\]<\u@\h \w>\$\[\033[0m\] "
```

Крім того, існує можливість змінити колір фону, для чого призначені екрановані послідовності, перераховані у табл. 4.

Таблиця 4.

**Екрановані послідовності, що використовуються для визначення кольору фону**

<i>Послідовність</i>	<i>Колір</i>
<code>\033[0;40m</code>	чорний
<code>\033[0;41m</code>	червоний
<code>\033[0;42m</code>	зелений
<code>\033[0;43m</code>	коричневий
<code>\033[0;44m</code>	синій
<code>\033[0;45m</code>	пурпуровий
<code>\033[0;46m</code>	бірюзовий
<code>\033[0;47m</code>	світло-сірий

Наприклад, щоб вивести запрошення на червоному фоні, достатньо змінити першу екрановану послідовність:

```
$ PS1="\[\033[0;41m\]<\u@\h \w>\$\[\033[0m\] "
```

А так запрошення буде відображатись зеленим кольором на червоному фоні (першою повинна бути вказана послідовність, що визначає колір фону, а потім – послідовність, що визначає колір тексту):

```
$ PS1="\[\033[0;41m\033[1;32m\]<\u@\h \w>\$\[\033[0m\] "
```

### Завдання

1. Ознайомтеся з роботою команд, які наведені в теоретичній частині лабораторної роботи. Подивіться для цих команд сторінки довідкового керівництва.
2. Відсортуйте виведення команди `ls -l` за датою зміни вмісту за місяцями.
3. Скопіюйте файл `/etc/passwd` у файл `passwd_example` та помістіть його у своєму домашньому каталозі.
4. З файлу `passwd_example` отримайте імена всіх користувачів, що містяться в першому полі кожного рядка, і помістіть відсортований у зворотному порядку результат у файл `cut_result`.
5. За допомогою текстового редактора `vi` або `nano` змініть імена кількох користувачів у цьому файлі і збережіть результат у новий файл `cut_result2`.
6. Порівняйте вміст файлів з іменами користувачів за допомогою програми `diff`.
7. Додайте до вмісту файлу `cut_result` вміст файлу `cut_result2`.
8. За допомогою команди `uniq` позбудьтеся від дублікатів у файлі `cut_result`.
9. За допомогою однієї команди отримаєте домашній каталог користувача `user` з файлу `passwd_example`.
10. Виведіть на екран значення всіх змінних середовища оточення. Проаналізуйте отримані результати і поясніть значення відомих вам змінних оточення.
11. Визначте тип використовуваного терміналу.
12. Змініть вміст змінної `PS1` так, щоб у запрошенні замість дужок використовувалися символи «<>».
13. Зробіть, щоб запрошення виводилось текстом жовтого кольору на синьому фоні.
14. Поверніть старий вигляд рядка запрошення.

***Контрольні питання***

1. Які утиліти для роботи з текстом ви знаєте?
2. Яка опція використовується для зміни порядку сортування **sort**?
3. За допомогою якої команди можна порівняти файли?
4. Що таке змінні оточення?
5. Як задати значення змінної оточення і як вивести його на екран?
6. Як змінити значення змінних оточення, видалити змінну?
7. Як змінити зовнішній вигляд рядка запрошення до введення команд?