***Лабораторна робота №6***

***«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*** даної команди виробляє нумерацію рядків при виведенні.

Серед інших вже знайомих команд обробки тексту слід виділити наступні:

* ***less*** – дозволяє організувати посторінкову 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***. Дана команда виробляє числову (опція ***-n***) сортування отриманих від команди ls даних по п'ятому стовпчику (опція ***-k***);
* ***uniq*** – видаляє сусідні повторювані рядки у файлі. Опції команди дозволяють також знайти неунікальні рядки і підрахувати кількість входжень кожного рядка. Дану 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 в файлі.

Розглянемо інші, ще незнайомі вам команди обробки текстових даних.

***cut*** - виконує фільтрацію тексту по стовпцях. В якості опції команда приймає номер поля ***(-f)***, роздільник ***(-d)*** і ін. Наприклад:

***$ сut -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. Об'єднуються файли 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, дана 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***

При роботі 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 при 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***.

Табл. 1. Деякі стандартні змінні середовища оточення

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

***Налаштування зовнішнього вигляду рядка-запрошення***

За замовчуванням рядок запрошення до вводу має наступний 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.

Табл. 2. Екрановані послідовності, що використовуються у рядку запрошення

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

Маючи список спеціальних символів, можна спробувати змінити оформлення запрошення. Для початку можна збереги 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 зворотну процедуру:

***$ PS1="$ps1\_old"***

Тепер можна спробуваити змінити зовнішній вигляд рядка запрошення. Наведемо декалька прикладів:

* ***$ PS1=*** - порожній рядок запрошення;
* ***$ PS1="\$ "*** – рядок виводить тільки символ ***$*** (або ***#*** для суперкористувача);
* ***$ PS1="\a\$ "*** – звуковий сигнал при кожному виведенні рядка запрошення та символ ***$*** (або ***#***);
* ***$ PS1="\t \h \$ "*** – виводить час та ім’я хоста***.***
* ***$ PS1="<\u@\h \w>\$ "*** - виводить ім’я користовача, ім’я хоста, поточний каталог. Все зе 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 ***\[*** і ***\]***. Екрановані послідовності ANSI починаються з вісімкового коду 033 (код, що генерується клавішею **ESC**), за яким слід 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.

Табл 3. Екрановані послідовності, що використовуються для визначення кольору тексту

|  |  |
| --- | --- |
| **Послідовність** | **Колір** |
| ***\033[0;30m*** | чорний |
| ***\033[0;31m*** | червоний |
| ***\033[0;32m*** | зелений |
| ***\033[0;33m*** | коричневий |
| ***\033[0;34m*** | синій |
| ***\033[0;35m*** | пурпурний |
| ***\033[0;36m*** | бірюзовий |
| ***\033[0;37m*** | світло сірий |
| ***\033[1;30m*** | темно-сірий |
| ***\033[1;31m*** | світло-червоний |
| ***\033[1;32m*** | світло-зелений |
| ***\033[1;33m*** | жовтий |
| ***\033[1;34m*** | світло-синій |
| ***\033[1;35m*** | світло-пурпурний |
| ***\033[1;36m*** | світло-бірюзовий |
| ***\033[1;37m*** | білий |

Наприклад, для того, щоб окрасити рядок запрошення у червоний колір, потрібно зробити наступне:

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

Однак в даному випадку червоним кольором буде окрашуватись 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 терміналу, 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.

Табл. 4. Екрановані послідовності, що використовуються для визначення кольору фону

|  |  |
| --- | --- |
| **Послідовність** | **Колір** |
| ***\033[0;40m*** | чорний |
| ***\033[0;41m*** | червоний |
| ***\033[0;42m*** | зелений |
| ***\033[0;43m*** | коричневий |
| ***\033[0;44m*** | синій |
| ***\033[0;45m*** | пурпурний |
| ***\033[0;46m*** | бірюзовий |
| ***\033[0;47m*** | 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 відображуватись зеленим кольором на червоному фоні (першою повинна бути вказана послідовність, що визначає колір фону, а потім послідовність, шо 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 даних 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 в зворотному порядку результат в файл ***cut\_result***.

5. За допомогою текстового редактора **vi** або **nano** змініть імена кількох користувачів в цьому файлі і збережіть результат в 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 в файлі ***cut\_result***.

9. Отримайте 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 в запрошенні 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?