***Лабораторна робота №16***

***«Мова обробки вхідного потоку та построкового розбору gawk у Linux»***

***Теоретичні відомості***

Утиліта ***gawk*** призначена для знаходження інформації в текстових файлах за заданими критеріями вибору. Вона забезпечена потужними засобами обробки тексту в великих текстових файлах. Утиліту ***gawk*** можна віднести до програмованих фільтрів, налаштованих на виконання конкретного завдання.

***gawk*** одночасно є утилітою, програмованим фільтром і середовищем програмування, за допомогою якої можна створювати інші фільтри. Тому вона займає особливе місце в операційних системах UNIX і Linux і є потужним і гнучким інструментом. ***gawk*** є версією утиліти ***awk***, використовуваної в UNIX і розробленої одним із творців мови С Брайеном Керніганом і іншими.

Будь-яке середовище програмування має свою мову програмування, це відноситься і до утиліти ***gawk***. Багато операторів взяті з мови програмування С і мають такий же синтаксис. Фільтри, що створюються за допомогою мови програмування ***gawk***, можуть використовуватися в середовищі будь-якого інтерпретатора UNIX і Linux, наприклад, в **bash**, **shell**, **tcsh**. Створені фільтри зчитують інформацію з файлу або стандартного потоку введення, аналізують, виділяють за певними заданими критеріями або змінюють цю інформацію, зберігають вихідні дані або направляються в стандартний потік виведення.

Таким чином, можна створювати не тільки фільтри, але і власні команди UNIX і Linux, що особливо актуально у великих системах розподілених ресурсів із нестандартною структурою організації. Утиліту ***gawk*** можна використовувати для створення звітів, пошуку заданих текстових фрагментів, виконання обчислень на основі введених даних, для роботи зі структурами, що нагадують бази даних, тобто що складаються з записів, що підрозділяються на поля, розділені спеціальними символами.

Утиліту ***gawk*** можна викликати безпосередньо з командного рядка або з shell-сценарію за допомогою ключового слова ***gawk***. Якщо скрипт включає ***gawk*** як складову, то його можна розглядати як новий фільтр, утиліту або команду.

Синтаксис команди ***gawk*** наступний:

***$ gawk '<Шаблон або умова {дія}>' <імена\_файлів>***

Командою ***gawk*** використовується такий шаблон пошуку, який застосовується у команді фільтрі ***grep*** та її похідних. Шаблон виділяється символами косої риски (знаками слеш).

**Приклад 1.**

***$ gawk '/інтерфейс Gnome/ {print}' /home/info/\****

Тут шукаються всі файли в каталозі ***/home/info***, в яких зустрічається послідовність з слів «***інтерфейс Gnome***» і результати направляються на стандартний пристрій виведення за допомогою дії ***{print}***. У результатах виводяться імена файлів і рядки, що містять шаблон.

Дія перенаправлення на стандартний пристрій виведення ***{print}*** зазвичай задається як дія за замовчуванням, тому команду у даному прикладі можна переписати наступним чином:

***$ gawk '/інтерфейс Gnome/' /home/info/\****

Результат при цьому не зміниться.

**Приклад 2.** Знайти по зазначеному шляху файли вихідних текстів на мові С, що включають слова, що позначають ім'я змінної, наприклад, ***SHELLDATA*** або ***SHDATA***:

***$ gawk '/SHELLDATA|SHDATA/' /home/include/\****

***Змінні і константи***

Утиліта ***gawk*** дозволяє визначати **змінні і константи**. Існує три типи змінних: змінні для позначення полів, спеціальні змінні і змінні. Змінні визначає користувач, інші утиліта визначає автоматично. Існує два типи констант: арифметичні та строкові. Арифметичні константи складаються з цифр, рядкові константи полягають в подвійні лапки і складаються з будь-яких символів, включаючи символи цифр. **Змінні визначаються після їх першого використання.**

***Поля***

Утиліта ***gawk*** може працювати зі змінними, що називаються полями. Поле являє будь-який набір символів, обмежений роздільниками полів (перше і останнє поля можуть мати по одному розділювачу). За замовчуванням як розділювач використовується проміжок або знак табуляції. Утиліта ***gawk*** нумерує поля рядків текстового файлу, починаючи з 1, а також автоматично визначає змінну для кожного поля файлу. Ім'я змінної, що позначає поле, складається з знака ***$*** і номера поля. Наприклад: ***$1***, ***$2***, ***$3*** і т.д. Змінна ***$0*** – спеціальна змінна, що визначає весь рядок.

**Приклад 3.** Припустимо, що у нас є текстовий файл ***/home/tabl/list\_students***, відсортований за алфавітом, кожен рядок якого включає: прізвище, ім'я, факультет, курс, рейтингову оцінку, і складається з наступних рядків-записів:

***Арінин Іван МП 1 4***

***Бетінов Євген ЕКТ 2 5***

***Кошин Леонід МП 1 4***

***Кошкін Володимир ЕКТ 1 5***

***Ліпін Федір МП 2 3***

***Пенов Микола МП 1 4***

***Яшин Петро ЕКТ 2 4***

Потрібно вивести список, що складається з прізвищ, імен та рейтингових оцінок.

Тобто потрібно вивести **1**, **2** та **5** поля. Поля перераховуються через кому і проміжок, а команда виглядатиме наступним чином:

***$ gawk '{print $1, $2, $5}' /home/tabl/list\_students***

В результаті виконання на екран виводиться наступна таблиця, поля якої будуть вирівнені.

***Арінин Іван 4***

***Бетінов Євген 5***

***Кошин Леонід 4***

***Кошкін Володимир 5***

***Ліпін Федір 3***

***Пенов Микола 4***

***Яшин Петро 4***

**Приклад 4.** Тепер роздрукуємо весь файл у вигляді таблиці так, щоб всі поля були вирівняні. Для цього використовуємо змінну ***$0***, інакше виведені дані не вирівняються і будуть погано читабельними:

***$ gawk '{print $0}' /home/tabl/list\_students***

Результат виведення на екран буде наступний:

***Арінин Іван МП 1 4***

***Бетінов Євген ЕКТ 2 5***

***Кошин Леонід МП 1 4***

***Кошкін Володимир ЕКТ 1 5***

***Ліпін Федір МП 2 3***

***Пенов Микола МП 1 4***

***Яшин Петро ЕКТ 2 4***

**Приклад 5.** Виберемо з вихідного файлу ***/home/tabl/list\_students*** студентів з рейтинговою оцінкою 5 і роздрукуємо все значення полів для обраних записів:

***$ gawk '/5/ {print $0}' /home/tabl/list\_students***

***Бетінов Євген ЕКТ 2 5***

***Кошкін Володимир ЕКТ 1 5***

У разі, якщо 5 зустрічається в інших полях, наприклад, курс 5, то умови вибірки недостатні.

***Оператори порівняння і логічні операції***

Для формування логічних виразів, що виконують перевірку умов, використовуються **оператори порівняння**. В останньому прикладі для пошуку потрібно використовувати в шаблоні оператор порівняння, а саме, оператор перевірки на рівність (***==***):

***$ gawk '/$5==5/ {print $0}' /home/tabl/list\_students***

Інший оператор порівняння називається оператор перевірки на нерівність (***!=***). І в якості прикладу може бути використаний для пошуку записів з рейтинговою оцінкою, відмінною від 5:

***$ gawk '/$5!=5/ {print $0}' /home/tabl/list\_students***

Утиліта ***gawk*** використовує оператори порівняння такі ж, як в мові С: ***<***, ***>***, ***>=***, ***<=***, але, на відміну від С, може порівнювати рядкові значення. Порівняння виконується відповідно до алфавіту. Утиліта ***gawk*** використовує логічні оператори:

* ***&&*** - логічне І (AND);
* ***||***  - логічне АБО (OR);
* ***!*** - логічне заперечення (NOT).

Вони дають можливість використовувати складні логічні умови, при цьому умови записуються в дужках.

**Приклад.** Вибрати зі списку студентів 1 курсу факультету ЕКТ:

***$ gawk '($3 == "ЕКТ") && ($4 == 1) {print}' list\_student***

***Функція length***

В умові можна використовувати функцію ***length***, що визначає довжину поля. Наприклад, потрібно вибрати записи студентів факультету ЕКТ. Можна записати так:

***$ gawk '/ЕКТ/ {print}' /home/tabl/list\_students***

Використовуючи функцію ***length*** і помічаючи, що третє поле (факультет) в наших даних складається з 2 або 3 символів, а потрібно вибрати значення, що складається з трьох символів, можна записати так:

***$ gawk '/length ($3 == 3) / {print}' /home/tabl/list\_students***

***Спеціальні змінні мови утиліти gawk***

В утиліті ***gawk*** визначаються спеціальні змінні:

* ***NR*** - зберігає номер переглядається або оброблюваної записи, з її допомогою зручно нумерувати рядки;
* ***NF*** - кількість полів в поточному записі;
* ***$0*** - всі поля поточного запису;
* ***$N*** - номер поля (див. Розділ "Поля");
* ***FS*** - роздільник поля введення;
* ***FILENAME*** - ім'я файлу, в якому працюємо.

Припустимо, нам потрібно вивести дані вихідного файлу в вигляді таблиці і пронумерувати рядки:

***$ gawk '{print NR, $0}' /home/tabl/list\_students***

***1 Арінин Іван МП 1 4***

***2 Бетінов Євген ЕКТ 2 5***

***3 Кошин Леонід МП 1 4***

***4 Кошкін Володимир ЕКТ 1 5***

***5 Ліпін Федір МП 2 3***

***6 Пенов Микола МП 1 4***

***7 Яшин Петро ЕКТ 2 4***

***Використання слів BEGIN і END***

Для опису дій до або після перегляду і обробки всіх записів використовуються ключові слова ***BEGIN*** і ***END*** відповідно.

Наприклад, якщо перед друком всіх пронумерованих записів ми хочемо поставити заголовок, то можна написати так (вважаємо, що ми знаходимося в ***/home/tab***):

***$ gawk 'BEGIN {print "student's list"} {print NR, $0}' list\_students***

Спочатку буде виведений заголовок ***student's list***, а далі пронумерований по рядках текст таблиці даних.

***Арифметичні оператори і функції***

***gawk*** підтримує повний набір арифметичних операторів, вони такі ж, як в мові С: ***\****, ***/***, ***+***, ***-***, ***%***.

Арифметичні обчислення в ***gawk*** виконуються над числовими шаблонами. Числовий шаблон являє собою послідовність цифр. Це може бути арифметична константа, змінна або поле, які складаються з цифр. Сюди входять вбудовані числові змінні ***NR*** і ***NF***.

**Приклад 1.** Вивести всі парні записи текстового файлу.

***$ gawk ' (NR % 2) {print NR, $0}' /home/tabl/list\_students***

**Приклад 2.** Порахувати сумарний бал оцінок студентів (знаходимося в ***/home/tabl***). Вивести на екран таблицю і сумарний підсумок.

***$ gawk '{print; sum += $5} END {print "summa"=, sum}' list\_students***

**Приклад 3.** Вивести на екран заголовок таблиці, таблицю, кількість записів і середнє значення. Кількість записів визначається змінною ***NR*** після перегляду всіх записів. В процесі обробки ***NR*** збільшується на 1 при зверненні до чергового запису файлу. Нумерація починається з 1.

***$ gawk 'BEGIN {print "exam results" }{print; sum += $5} END {print NR, "average"=, sum}' list\_students***

**Приклад 4.** Нехай є файл ***/home/plant***, що складається із записів з наступними полями: ім'я підприємства, дохід по **1**, **2**, **3**, **4** кварталах. Знайти підсумковий дохід **d** для кожного підприємства і роздрукувати записи, пронумерувавши їх і вказавши підсумковий дохід останнім полем кожного запису.

***$ gawk '{ d = $1 + $2 + $3 + $4; print NR, $0, d}' /home/plant***

***Пошук за шаблоном***

Для виконання пошуку за шаблоном в полях поточного запису застосовуються спеціальні символи: ***~***, ***!~***. За допомогою операції ***~*** можна перевірити, чи присутній конкретний шаблон в певному полі (шаблоном тут є частина поля або все поле). За допомогою операції ***!~*** Можна перевірити, в яких записах відсутній конкретний шаблон в певному полі.

**Приклад 1.** Чи присутнє прізвище Леонов в списку файлу ***list\_students***?

***$ gawk '($1 ~ /Леонов/) {print}' /home/tabl/list\_students***

**Приклад 2.** Є запис студента Іванова в списку файлу ***list\_students***? Причому ми не впевнені, з великої чи малої літери її могли написати.

***$ gawk '($1 !~ /[Іі]ванов/) {print}' /home/tabl/list\_students***

***Оформлення інструкцій gawk окремим файлом***

Якщо інструкція ***gawk*** велика в написанні, то її можна оформити окремим файлом, наприклад, зазначений вище приклад:

***$ gawk 'BEGIN {print "exam results" }{print; sum += $5} END {print NR, "average"=, sum}' list\_students***

можна записати так. Створюємо окремий файл, назвемо його наприклад ***f\_instr***:

***BEGIN***

***{print "exam results"}***

***{print; sum += $5}***

***END {***

***print NR, "average"=, sum}***

Далі виконуємо команду ***gawk*** з опцією ***-f***, яка дає можливість читати інструкції із зазначеного файлу, а не з командного рядка. Тоді загальна структура команди наступна:

***$ gawk -f <ім’я\_файлу\_інструкції> <ім’я\_оброблюваного\_файлу>***

**Приклад.**

***$ gawk -f f\_instr list\_students***

Іноді файли інструкцій для зручності записують з розширенням ***gawk***, тобто ім'я файлу тоді виглядає так: ***f\_instr.gawk***, а команда так:

***$ gawk -f f\_instr.gawk list\_students***

***Завдання***

1. Вивести в файл список файлів і каталогів поточного каталогу (з опцією ***-l***). Скласти сценарій, який виводить на друк імена файлів або каталогів із зазначенням їх власників.
2. Створити сценарій **bash**, який використовує команду gawk і стандартні команди Linux, який буде показувати 3 користувачів, що використовують найбільший обсяг дискового простору файлової системи /home Спочатку виконайте команди в командному рядку. Потім помістіть їх у сценарій. Сценарій повинен формувати зрозумілий для читання результат.

***Контрольні питання***

1. Для чого призначена програма ***gawk***?
2. Який формат має команда ***gawk***?
3. Як вивести певні поля у тексті за допомогою ***gawk***?
4. Яким чином можна використати декілька команд у програмному сценарії ***gawk***?
5. Як за допомогою ***gawk*** прочитати програму з файлу?
6. Для чого у ***gawk*** призначені слова ***BEGIN*** та ***END***?