“Київський фаховий коледж зв’язку”

Циклова комісія Комп’ютерної інженерії

**ЗВІТ ПО ВИКОНАННЮ**

**ЛАБОРАТОРНОЇ РОБОТИ №6**

з дисципліни: «Операційні системи»

**Тема:** **“Робота з текстом в CLI-режимі Linux та створення скриптових сценаріїв”**

Виконала студентка

групи РПЗ-93а

Команда 6: Колосюк Д.С., Мількевич В.П.

Перевірив викладач

Сушанова В.С.

Київ 2022

**Мета роботи:**

1. Отримання практичних навиків роботи з командною оболонкою Bash.
2. Знайомство з базовими діями при роботі з текстом у терміналі.
3. Знайомство з базовими діями при роботі зі скриптовими сценаріями.

**Матеріальне забезпечення занять**

1. ЕОМ типу IBM PC.
2. ОС сімейства Windows (Windows 7).
3. Віртуальна машина – Virtual Box (Oracle).
4. Операційна система GNU/Linux – CentOS.
5. Сайт мережевої академії Cisco netacad.com та його онлайн курси по Linux

**Завдання для попередньої підготовки**

***Готувала матеріал студентка Мількевич В.***

1. Прочитайте короткі теоретичні відомості до лабораторної роботи та зробіть невеличкий словник базових англійських термінів з питань призначення команд та їх параметрів.

|  |  |
| --- | --- |
| formatting features | особливості форматування |
| Concatenate | Конкатенувати |
| redirecting file | файл перенаправлення |
| Capability | Здатність |
| Respectively | Відповідно |
| initial command | початкова команда |
| specified pattern | зазначений візерунок |
| Recursively | Рекурсивно |
| delimter | роздільник |

1. На базі розглянутого матеріалу дайте відповіді на наступні питання:

2.1 Яка мета команд кіт, менше, більше, голова і хвіст? Складіть короткий опис кожної команди та виділіть її основні параметри. Як їх встановити.

Cat – дозволяє створювати, об’єднувати та виводити вміст файлів у командному рядку або в іншому файлі.

Утиліта не потребує додаткового встановлення, оскільки вона попередньо встановлена ​​в пакеті coreutils на будь-якій системі на базі Debian або Red Hat.

За допомогою команди cat ви можете швидко створити файл і вставити в нього текст. Для цього скористайтеся оператором > перенаправлення, щоб переспрямувати текст у файл.

Less – утиліта командного рядка, яка відображає вміст файлу або результат команди по одній сторінці. Це схоже на інші, але має більш розширені функції та дозволяє переміщатися вперед і назад через файл. Команда less в основному використовується для відкриття великих файлів.

Утиліта підтримує велику кількість параметрів, і майже всі ці параметри є вузькоспеціалізованими і навряд чи будуть корисні більшості користувачів, а найпопулярнішим з них є параметр -N, що дозволяє відображати номери рядків, які є дуже зручний при читанні програмного коду параметр -X, який дозволяє не очищати вміст терміналу після завершення роботи утиліти, а також параметр -I, що дозволяє здійснювати пошук у тексті без урахування регістру .

Більше – утиліта більше призначена для перегляду файлів у терміналі Linux. Своєю назвою він зобов'язаний напису більше (в російському варіанті - далі), що з'являється внизу кожної сторінки. Команда more linux є однією з найбільш примітивних команд для роботи з текстом.

* 1. Поясніть, як оболонка працює з каналами, потоками та фільтрами.

Канали

Багатопоточність є природним розширенням багатозадачності, так само як віртуальні машини, які дозволяють запускати кілька операційних систем на одному комп’ютері, є логічним розширенням концепції спільного використання ресурсів. У неформальному, але простому визначенні потік — це виконання послідовності машинних інструкцій. Багатопотокова програма має кілька потоків, які працюють одночасно. Деякі автори уникають терміна «потік» і замість нього використовують термін «потік», ймовірно, щоб уникнути плутанини між потоками програми та потоками вводу-виводу. Для позначення послідовного виконання ланцюжка інструкцій я особисто віддаю перевагу терміну «потік», який я буду використовувати. Сподіваюся, що читачі Linux Format не заплутаються в контекстах і, зустрівши слово thread, вони завжди зрозуміють, чи йдеться про програмні потоки, вхідні вихідні потоки чи штормові потоки. Якщо в процесі є кілька потоків, програма (процес) стає багатопоточною. У світі Linux все виглядає інакше. У Linux кожен потік є процесом, і щоб створити новий потік, вам потрібно створити новий процес. У чому ж тоді перевага багатопоточності в Linux перед багатопроцесорною? Багатопотокові програми Linux використовують особливий тип процесу для створення додаткових потоків. Ці процеси є звичайними дочірніми процесами головного процесу, але вони спільно використовують адресний простір, дескриптори файлів і обробники сигналів з основним процесом. Для позначення процесів цього типу використовується спеціальний термін – легкі процеси (lightweight processs). Прикметник «легкий» у назві процесів-потоків цілком виправданий. Оскільки цим процесам не потрібно створювати власну копію адресного простору (та інших ресурсів) батьківського процесу, створення нового легкого процесу набагато дешевше, ніж створення повноцінного дочірнього процесу. Оскільки потоки Linux насправді є процесами, у світі Linux не можна сказати, що один процес містить кілька потоків. Якщо ви скажете це, вас відразу запідозруть у ворожому проникненні!

Потоки

Ви можете уявити канал як невеликий кільцевий буфер у ядрі операційної системи. З точки зору процесів, пайп виглядає як пара відкритих дескрипторів файлів - один для читання і один для запису (можна більше, але незручно). Ми можемо писати в канал, поки є місце в буфері, якщо місце в буфері закінчиться, процес буде заблоковано під час запису. Ми можемо читати з каналу, поки в буфері є дані, якщо даних немає, процес буде заблоковано при читанні. Якщо закрити дескриптор, відповідальний за запис, тоді спроба читання покаже кінець файлу. Якщо дескриптор, відповідальний за читання, закритий, то спроба запису призведе до доставки сигналу SIGPIPE і помилки EPIPE.

При використанні каналу в програмуванні оболонки блокування читання/запису гарантують, що швидкість виконання двох програм синхронізована і що вони закінчуються одночасно.

Немає концепції позиції читання/запису для каналів, тому запис завжди виконується в хвіст буфера, а читання з голови. Для архітектури i386 розмір буфера, пов’язаного з каналом, кратний розміру сторінки (4096 байт). У Linux версії до 2.6.11 використовували одну сторінку (4 КБ), після - 16 сторінок (65 КБ), з можливістю зміни через fcntl. POSIX визначає значення для PIPE\_BUF, яке визначає максимальний розмір атомарного запису. У Linux PIPE\_BUF становить 4096 байт.

фільтри

Одним із основоположних принципів Linux є те, що кожен елемент повинен виконувати лише одну функцію і що всі елементи можна легко комбінувати один з одним. Ці елементи є будівельними блоками, які можна збирати і таким чином будувати все, що ми хочемо. У цій і наступних статтях ми поговоримо про конкретні будівельні блоки. Перший - це фільтри.

Фільтри – це команди, які певним чином перетворюють передані текстові дані. За допомогою фільтрів ми можемо брати необроблені дані, які зберігаються у файлах, і керувати ними – привести їх у зручну для нас форму.

2.3 Яка мета команди grep?

grep — це утиліта інтерфейсу командного рядка, яка знаходить і виводить рядки, які відповідають заданому регулярному виразу. Ім'я утиліти являє собою послідовність команд для пошуку регулярних виразів в редакторі ed - g / re / p. Цю послідовність команд можна описати англійською фразою «шукати глобально рядки, що відповідають регулярному виразу, і друкувати їх».

Спочатку він був створений для операційної системи UNIX.

Існують модифікації grep:

egrep (з розширеною обробкою регулярних виразів),

fgrep (який інтерпретує символи $ \* [] ^ | () \ буквально),

rgrep (з увімкненим рекурсивним пошуком).

Синтаксис

grep "регулярний вираз" "файл"

або

кіт "файл" | grep "регулярний вираз"

3. На основі розглянутого матеріалу в онлайн-курсах дайте відповіді на наступні питання:

3.1 Опишіть концепцію написання сценаріїв у командній оболонці.

Сценарій оболонки — це список команд у комп’ютерній програмі, яка запускається оболонкою Unix, яка є інтерпретатором командного рядка. Сценарій оболонки зазвичай містить коментарі, які описують кроки. Різними операціями, які виконуються скриптами оболонки, є виконання програми, маніпулювання файлами та друк тексту. Обгортка також є свого роду сценарієм оболонки, який створює середовище програми, запускає програму тощо.

3.2 Як створювати та редагувати скрипти, що мені потрібно зробити, щоб запустити скрипт?

Скрипти створюються шляхом об'єднання окремих блоків: або послідовно, або шляхом розміщення блоку в певному місці іншого блоку (структура керування, функції тощо). Один спрайт може мати кілька сценаріїв, які запускаються незалежно від дій користувача (за допомогою натискання клавіші або кнопки миші), таймера або отримання повідомлення від іншого спрайта. Сценарій складається з стеків.

Стек — це набір різнокольорових графічних блоків, з’єднаних послідовно в межах однієї події. Блок — це мінімальна частина програми в Scratch: змінна, оператор, функція або структура керування.

Контрольні запитання

***Готувала матеріал студентка Колосюк Д.***

1. Як я можу переспрямувати потоки в оболонці? Продемонструйте приклади переспрямування повідомлень про помилки вводу-виводу.
2. Для чого використовуються фільтри команд. Дайте кілька прикладних завдань, де їх використання необхідно.
3. Яке призначення каталогу файлів / dev / null?

**Висновки**

***Готувала матеріал студентка Мількевич В.***

Ми отримали практичні навички роботи з командною оболонкою Bash, вивчили основні дії під час роботи з текстом у терміналі, а також вивчили основні дії під час роботи зі скриптами сценаріїв.