**“Київський коледж зв’язку”**

**Циклова комісія Комп’ютерної інженерії**

**ЗВІТ ПО ВИКОНАННЮ**

**ЛАБОРАТОРНОЇ РОБОТИ №6**

**з дисципліни: «Операційні системи»**

**Тема: “Знайомство з базовими командами CLI-режиму в Linux для роботи з**

**тектосвими файлами та написання найпростіших скриптових сценаріїв”**

**Виконали:**

**студенти**

**групи КСМ-83б**

**Ярина Р. та Марчук Р.**

**Перевірив викладач:**

**Повхліб В.С.**

**Київ 2020**

**Мета роботи:**

1. Отримання практичних навиків роботи з командною оболонкою Bash.

2. Знайомство з базовими діями при роботі з довідкою.

3. Знайомство з базовими діями при роботі з файлами та каталогами.

**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 LinuxВиконання роботи:

1. **Словничок:**
2. The **cat** command, short for concatenate, is a simple but useful command whose functions include creating and displaying text files, as well as combining copies of text files. One of the most popular uses of cat is to display the content of text files.

Команда **cat**, скорочена від concatenate, - це проста, але корисна команда, до функцій якої входить створення та відображення текстових файлів, а також комбінування копій текстових файлів. Одне з найпопулярніших видів використання котів – показ вміст текстових файлів.

1. **Head and Tail.**The head and tail commands are used to display only the first few or last few lines of a file, respectively.

Команди **head і tail** використовуються для відображення лише перших кількох або останніх декількох рядків файлу відповідно.

1. **Pipe.** The symbol can be used to send the output of one command to another.

**Pipe** Символ може використовуватися для надсилання результатів однієї команди іншій.

1. **Standard input (STDIN)** is information that the user typically enters using the keyboard.

**Стандартний ввід (STDIN)** - це інформація, яку користувач зазвичай вводить за допомогою клавіатури.

1. Standard output, or **STDOUT**, is the normal output of commands.

Стандартний вихід, або **STDOUT**, - це звичайний вихід команд.

1. Standard error, or **STDERR**, is error messages generated by commands.

Стандартна помилка, або **STDERR**, - це повідомлення про помилки, що генеруються командами.

**2.**

2.1. Команда **cat**, скорочена від concatenate, - це проста, але корисна команда, до функцій якої входить створення та відображення текстових файлів, а також комбінування копій текстових файлів. Одне з найпопулярніших видів використання котів – показ вміст текстових файлів.

Команда **less** забезпечує дуже розширену можливість пошуку сторінок. Зазвичай це пейджер за замовчуванням.

**More** команда існує з перших днів існування UNIX. Хоча вона має менше функцій, ніж **less**

команда, однак, команда "l**less**" не входить до всіх дистрибутивів Linux. Команда завжди

доступна.

Команди **head і tail** використовуються для відображення лише перших кількох або останніх декількох рядків файлу відповідно.

2.2 Перенаправлення вводу / виводу (I / O) дозволяє передавати інформацію командного рядка в різні потоки. Раніше

обговорюючи переспрямування, важливо розуміти стандартні потоки.

СТДІН

Стандартний ввід (STDIN) - це інформація, яку користувач зазвичай вводить за допомогою клавіатури. Коли команда

підказує оболонці дані, оболонка надає користувачеві можливість вводити команди, які, в свою чергу, надсилаються до

команда як STDIN.

ЗНИЖЕННЯ

Стандартний вихід, або STDOUT, - це звичайний вихід команд. Коли команда працює правильно

(без помилок) вихідний результат, який він виробляє, називається STDOUT. За замовчуванням STDOUT відображається у вікні терміналу

де виконується команда. STDOUT також відомий як потік або канал # 1.

STDERR

Стандартна помилка, або STDERR, - це повідомлення про помилки, що генеруються командами. За замовчуванням STDERR відображається у

вікно терміналу, в якому виконується команда. STDERR також відомий як потік або канал №2.

Перенаправлення вводу-виводу дозволяє користувачеві перенаправляти STDIN, щоб дані надходили з файлу, а STDOUT / STDERR

цей вихід надходить у файл. Перенаправлення досягається за допомогою стрілки & lt; & gt; символів.

2.3 Команда **grep** може бути використана для фільтрації рядків у файлі або виводу іншої команди, яка відповідає

вказаний шаблон. Цей шаблон може бути таким самим простим, як точний текст, якому ви хочете відповідати, або може бути набагато більшим

вдосконалений завдяки використанню регулярних виразів.

2.4

* *Охарактеризуйте поняття скриптового сценарію у командній оболонці.*

**Командна оболонка в UNIX** є інтерфейсом командного рядка в Unix-подібних операційних системах, тобто виконує команди, які подає користувач, або які читаються з файлів. Такі файли з *командами оболонки* називаються сценаріями (скриптами, програмами) оболонки. Ці сценарії не компілюються, а інтерпретуються оболонкою. Це означає, що оболонка прочитує сценарій від початку до кінця, рядок за рядком, шукаючи зазначені там команди й виконуючи їх; на відміну від цього підходу, компілятор перетворює цілу програму до вигляду, придатного до виконання машиною — потім файл з таким кодом можна використати в сценарії оболонки. Характерна особливість мови оболонки — багато операцій, які в традиційних мовах програмування є вбудованими, виконуються з допомогою виклику зовнішніх програм.

Одним з завдань оболонки є забезпечення користувацького середовища, яке можна налаштувати за допомогою конфігураційних файлів.

* *Яким чином створюються та редагуються скрипти, що треба зробити щоб запустити скрипт?*

Shell - це один з командних інтерпретаторів, що поставляються разом з ОС unix. Програма на мові Shell називається скриптом. Shell скрпітам вказують розширення \* .sh і працюють вони в Unix системах.

Припустимо у вас є скрипт hello.sh що складається з однієї команди.

echo "Hello World"

Щоб його запустити, треба зайти в каталог, де розташований скрипт, набрати назву інтерпретатора sh і першим параметрів вказати ваш файл hello.sh.

sh hello.sh

Щоб кожен раз не вказувати інтерпретатор в терміналі, можна зробити скрипт виконуваним. Для цього необхідно:

Вказати інтерпретатор всередині файлу.

На першій сходинці після #! прописуючи шлях до bash-інтерпретатора, по-замовчуванню це / bin / bash (подивитися можна командою where is bash). Содержміое скрипта hello.sh виходить таким:

**#!/bin/bash**

echo "Hello World"

Зробити наш файл виконуваним. Для цього використовується команда chmod + x і ім'я файлу скрипта:

chmod +x hello.sh

Тепер можна запускати:

./hello.sh

Перед назвою скрипта треба ставити крапку і слеш, зроблено для ускладнення роботи вірусів. Точка означає "поточний каталог", а слеш "/" роздільник між ім'ям каталогу та ім'ям скрипта.

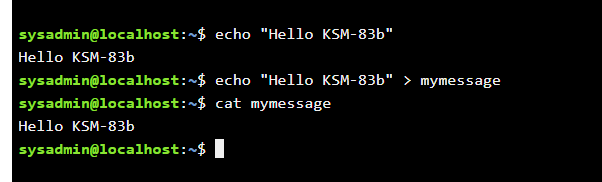
**Хід роботи**

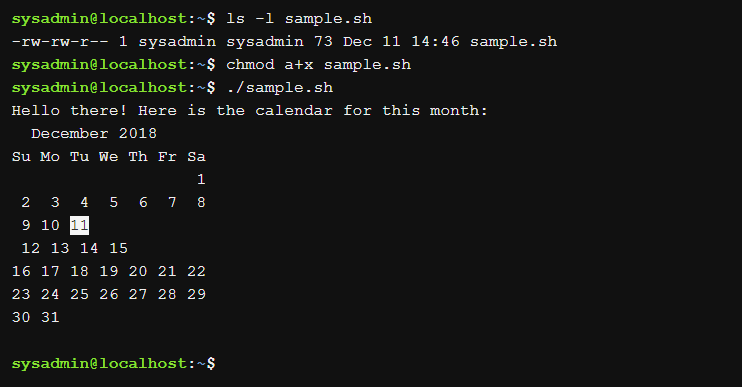
**3.**

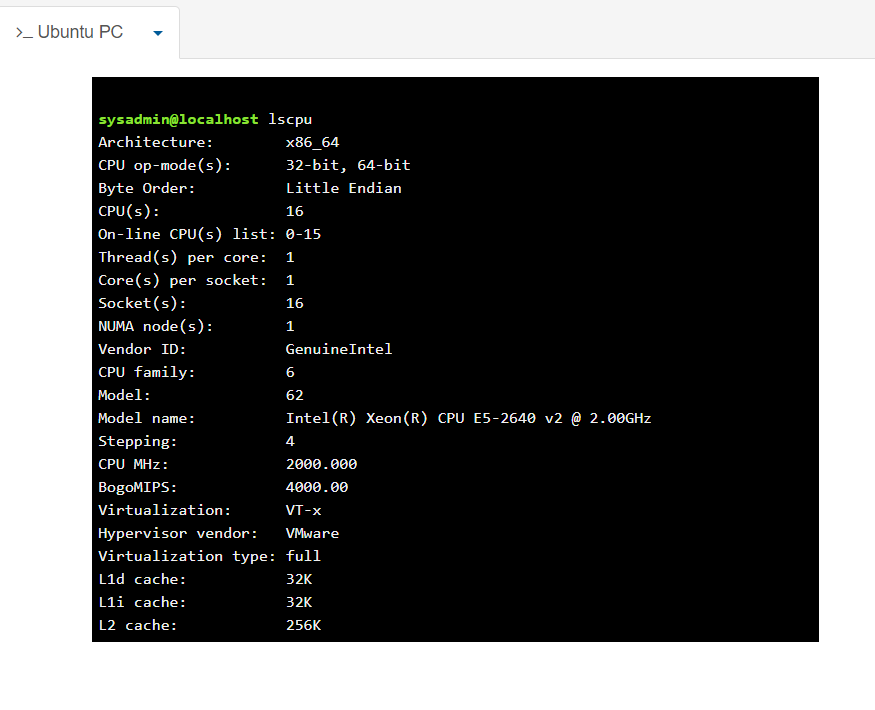
|  |  |
| --- | --- |
| **find-** | Команда find- хороша команда для демонстрації того, як stderr працює. Ця дуже гнучка команда дозволяє виконувати пошук за допомогою безлічі параметрів, таких як ім'я файлу, розмір, дата, тип і дозвіл. |
| **tr** | Команда tr приймає введення з клавіатури (stdin), переводить символи і потім перенаправляє висновок в stdout. |
| **cut** | Команда cut корисна для вилучення полів з файлів, які або розділені символом, наприклад двокрапкою в / etc / passwd, або мають фіксовану ширину. |
| **more або less** | Використання команд more або less дозволяє користувачеві переглядати дані «по сторінці» або по рядку за раз. |
| **cat** | Використовуйте cat команду для відображення всього вмісту / etc / passwdfile: |
| **--More** | Значок --More - (92%) означає, що ви перебуваєте в moreкоманде і на 92% використовуєте поточні дані. |
| **head** | За замовчуванням head команда відобразить перші десять рядків файлу: глава / etc / passwd |
| **tail** | Використовуйте tail команду для відображення останніх десяти рядків / etc / passwd файлу |
| **Head** | Використовуйте head команду для відображення перших двох рядків / etc / passwd файлу |
| **grep** | Команда grep використовує базові регулярні вирази, спеціальні символи, такі як символи узагальнення, які відповідають шаблонах в даних. Команда grep повертає всю рядок, що містить відповідний шаблон. |
| **fgrep** | Команда fgrep використовується для зіставлення буквальних символів, ігноруючи особливе значення символів регулярного виразу. |
| **|** | Символ вертикальної риси | або «оператор чергування» діє як оператор «або». |

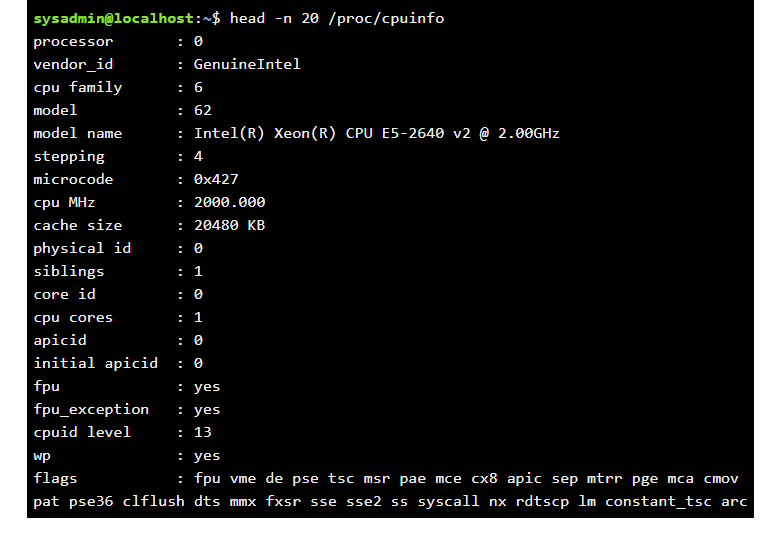
**4***.*

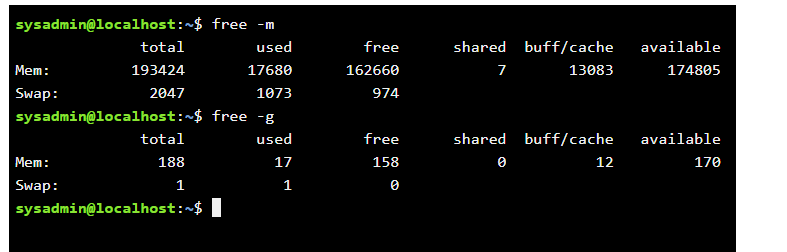
* *сценарій має виводити привітання до поточного користувача вказуючи поточну дату та інформаціюпро поточну систему;*

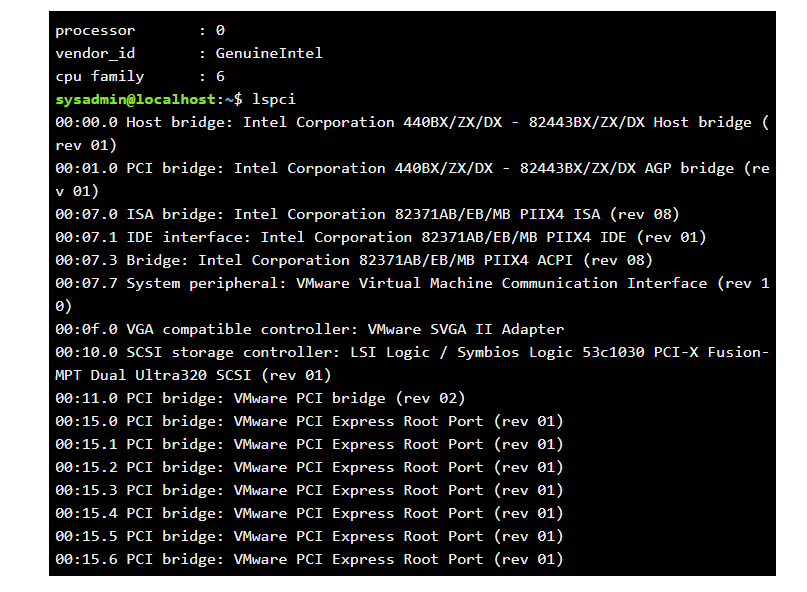
**

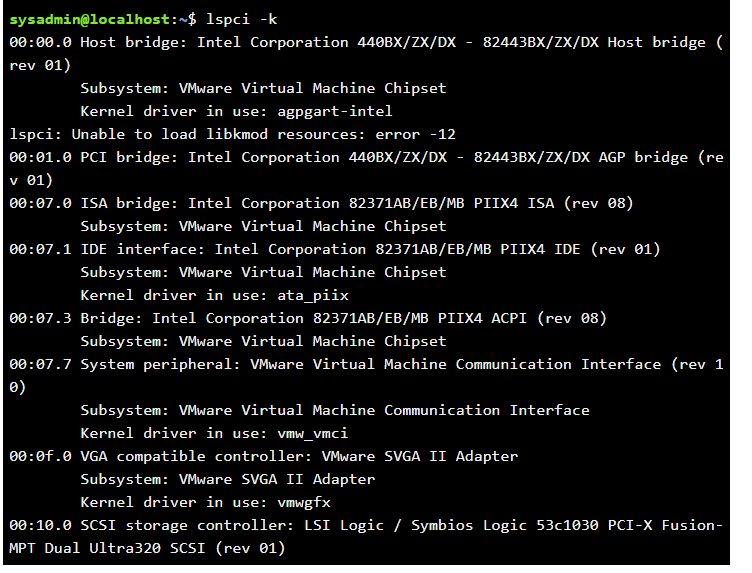
**

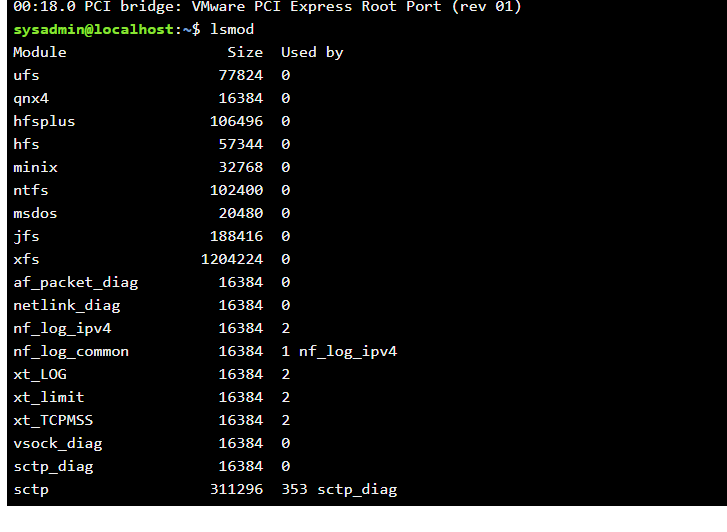
* *сценарій має виводити інформацію про апаратну конфігурацію поточної системи (використовуйтекоманди розглянуті в Lab 12).*****

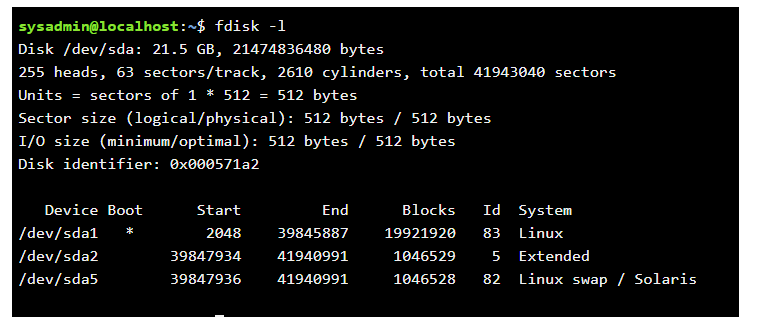
****

****

****

****

****

****

**Відповіді на контрольні запитання:**

**1)**Команди і сценарії можуть отримувати вхідні дані двома способами: з стандартного вхідного потоку (пов'язаний з клавіатурою) або з файлу. Аналогичний поділ існує і при виведенні даних: результати роботи команди або сценарію за замовчуванням направляються на екран терміналу, але ожна перенаправляти їх у файл. Якщо в процесі роботи виникають помилки. повідомлення про них може відображаються на екрані, потік помилок также можна переспрямувати у файл.

Linux надає кілька команд для виведення повідомлень у стандартний потік виводу:

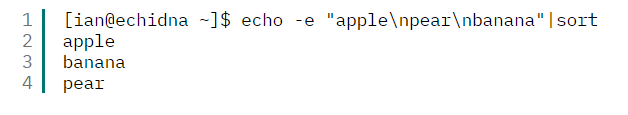
* ***echo*** *- Вивести рядок у стандартний потік виводу.*
* ***printf*** *- Вивести форматований текст у стандартний ппоток виводу.*
* ***yes*** *- Виводити повторюваний текст у стандартний ппоток виводу.*
* ***seq*** *- Показує послідовність чисел у стандартний ппоток виводу*
* ***clear*** *Очистити екран або вікно.*

Перенаправлення потоків введення-виведення здійснюється, подібно DOS (Точніше, синтаксис перенаправлення потоків ОС DOS сприйняла піт UNIX) за допомогою символів:

> - перенаправлення стандартного потоку виводу >> - перенаправлення стандартного потоку виводу у режимі записування < - перенаправлення стандартного потоку вводу << - отримання дані зі стандартного потоку вводу до тих пір, поки не зустрінеться розділювач Однак, на відміну від DOS при створенні програмного каналу між двома процесами пОС UNIX/Linux запускає обидва процесу одночасно і здійснює передачу інформації через системний буфер (без проміжної запису на диск). Таким чином, програмні канали в ОС UNIX/Linux є дуже ефективним способом обміну. У разі ппереполнения системного буфера (наприклад, якщо `передавальна"" програма видає інформацію в канал швидше, ніж її може обробити `приймаюча"" програма) ОС автоматично припиняє той процес, який здійснює запис у канал до звільнення буфера.

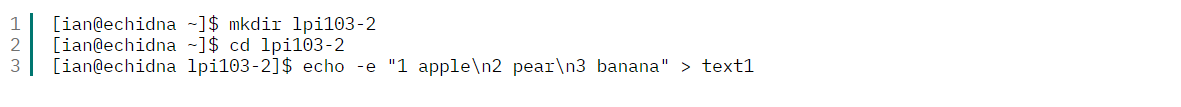
**2)** Фільтрація тексту - це процес отримання вхідного текстового потоку, виконання деяких перетворень над ним і передача змінених даних в вихідний потік. Хоча вхідні або вихідні дані можуть надходити з файлів, в UNIX® і Linux фільтрація зазвичай здійснюється шляхом складання конвеєрів з команд, в яких висновок однієї команди передається з програмного каналу (або перенаправляється) на вхід наступної команди. Більш детально програмні канали та перенаправлення розглядаються в статті "Потоки, програмні канали та перенаправлення" (див. Перелік матеріалів для підготовки до іспитів LPIC-1), а зараз давайте розглянемо програмні канали та прості перенаправлення виведення за допомогою операторів | і>.

Передані командам параметри можуть служити вхідними даними цих команд, а вихідні дані можуть виводитися на ваш термінал. Багато команд обробки тексту (фільтри) можуть отримувати вхідні дані або зі стандартного потоку введення, або з файлу. Щоб передати висновок команди1 на вхід команде2 (яка виступає в якості фільтра), необхідно з'єднати ці дві команди оператором конвейеризации введення / виведення (|). У лістингу 1 показано, як перенаправити висновок команди echo на вхід команді sort, яка сортує отриманий список слів.

Передача виведення команди echo на вхід команди sort

У будь-який з цих команд можуть бути опції або аргументи. За допомогою оператора | можна також перенаправити висновок другої команди на вхід третьої групи і так далі. Побудова довгих конвеєрів з команд, кожна з яких має свій обмежений функціонал - це поширений в Linux і UNIX прийом, використовуваний для вирішення поставлених завдань. Іноді аргументом команди може бути не ім'я файлу, а знак дефіса (-); це означає, що вхідні дані слід приймати зі стандартного пристрою вводу, а не з файлу.

Перенаправлення виводу команди в файл:



**3)** / dev / null - спеціальний файл у системі класу UNIX, що представляє собою так зване «пустое устройство». Запис у нього відбувається успішно, незалежно від обсягу «записаної» інформації. Зчитування з / dev / null еквівалентно зчитуванню кінця файлу (EOF).

У DOS є псевдофайл NUL з аналогічними властивостями.

**Висновок.**

Завдяки цій лабораторній роботі, я отримав практичні навики роботи з командною оболонкою Bash; також ознайомився з базовими діями при роботі з довідкою та при роботі з файлами та каталогами.