Київський фаховий коледж зв’язку

**ЗВІТ ПО ВИКОНАННЮ**

**ЛАБОРАТОРНОЇ РОБОТИ №3**

з дисципліни: «Операційні системи»

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

Виконали: студенти 3 курсу,

групи КСМ-13А

Засенко Олександр      
Дзюбенко Дмитро    
Сторожук Костянтин

Київ 2023

**Мета роботи:**

1. Знайомство з базовими командами CLI-режиму в Linux.

2. Знайомство з базовими текстовими командами в термінальному режимі роботи в різних ОС.

**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.

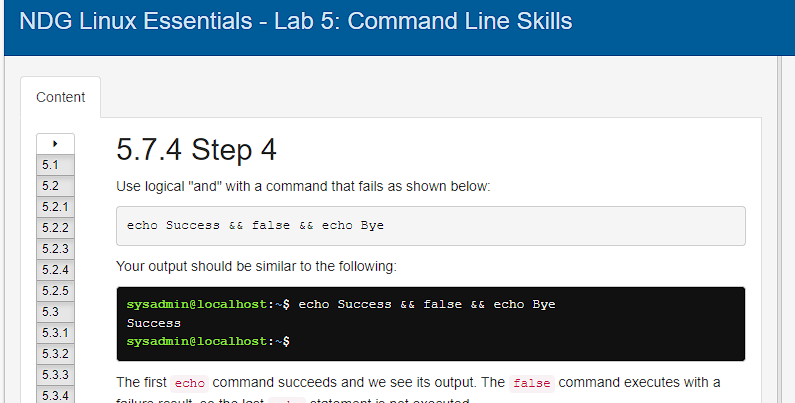
**Завдання для попередньої підготовки**

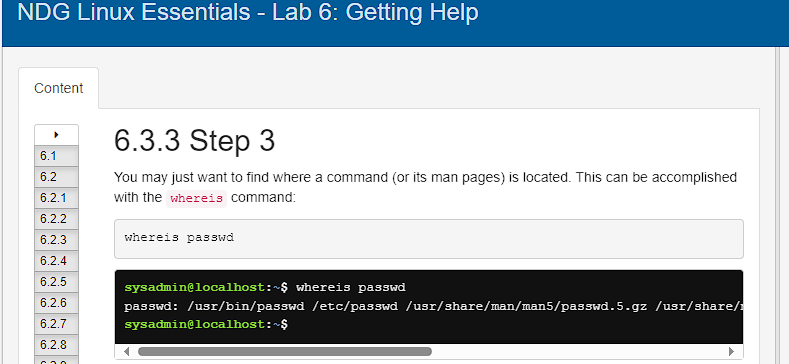
***Готував матеріал студент Zasenko***

Невеликий словник базових англійських термінів з питань призначення команд та їх параметрів.

|  |  |
| --- | --- |
| **Команди** | **Пояснення українською** |
| **ls** | **Вивести список файлів і каталогів у поточному каталозі.** |
| cd | **Змінити поточний каталог.** |
| pwd | **Показати поточний каталог (шлях)** |
| mkdir | **Створити новий каталог.** |
| rmdir | **Видалити порожній каталог.** |
| touch | **Створити новий порожній файл.** |
| cp | **Копіювати файли або каталоги.** |
| **mv** | Перемістити (перейменувати) файли або каталоги. |
| **rm** | Видалити файли або каталоги. |
| **cat** | Вивести вміст файлу на екран |

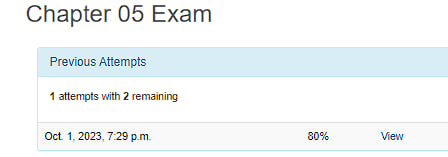
**Вивчить матеріали онлайн-курсу академії Cisco “NDG Linux Essentials”:**



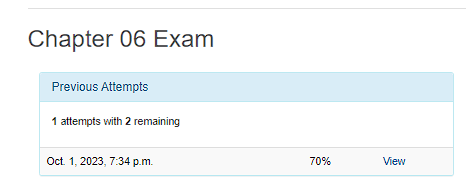


**Пройдіть тестування у курсі NDG Linux Essentials за такими темами:**

**- Chapter 05 Exam**



**- Chapter 06 Exam**



**Дайте визначення наступним поняттям:**

***Готував матеріал Zasenko***

**Командний інтерпретатор (shell)** **-** це програма в операційній системі, що дозволяє вам взаємодіяти з комп'ютером через текстовий інтерфейс. Він виконує команди, працює з файловою системою, управляє процесами та дозволяє створювати скрипти для автоматизації завдань. Bash є стандартним інтерпретатором для багатьох Linux-систем.

**Оболонка (shell)** - це інтерфейс між користувачем та операційною системою, який дозволяє користувачеві взаємодіяти з комп'ютером через команди і скрипти. Оболонка приймає команди від користувача, виконує їх та повертає результати. Вона також надає доступ до файлової системи, управляє процесами і виконує інші системні завдання.

**Команда** - це інструкція або вираз, який ви вводите в командний рядок або командний інтерпретатор (shell) операційної системи для виконання певної дії або завдання. Команди можуть включати запуск програм, взаємодію з файлами, керування процесами, налаштування системи та багато інших дій.

Наприклад, ось декілька прикладів команд:

1. ls - команда для відображення списку файлів і каталогів у поточному каталозі.
2. cd - команда для зміни поточного каталогу.
3. mkdir - команда для створення нового каталогу.
4. cp - команда для копіювання файлів або каталогів.
5. mv - команда для переміщення або перейменування файлів або каталогів.
6. rm - команда для видалення файлів або каталогів.
7. ps - команда для відображення списку активних процесів.
8. kill - команда для завершення (вбивання) процесів.

**Дайте відповіді на наступні питання:**

1. Рядок запрошення (prompt) в командному інтерпретаторі (shell) надає базову інформацію, яка допомагає користувачеві визначити поточний стан системи та готовність до введення команд. Така інформація включає:

1. **Ім'я користувача**: Рядок запрошення часто містить ім'я користувача, яке вказує, від імені кого ви введете команди.
2. **Ім'я комп'ютера (хоста)**: Він може включати ім'я або адресу комп'ютера, на якому ви виконуєте команди, особливо якщо ви працюєте в мережі або віддалено.
3. **Поточний каталог**: Зазвичай відображається шлях до поточного робочого каталогу, в якому ви працюєте. Це допомагає вам визначити, де ви знаходитеся в файловій системі.
4. **Розділювач команд**: Зазвичай використовується символ $ для користувачів з обмеженими правами та # для користувачів з правами адміністратора (root). Він вказує, що система готова до введення нової команди.

Приклади рядків запрошення:

* user@hostname:~/myfolder$ - звичайний користувач.
* root@localhost:/var/www# - адміністратор (root).

Рядок запрошення може бути налаштований і кастомізований користувачем в залежності від власних вимог і вподобань.

2. .

Команди в операційній системі використовують параметри та аргументи для отримання додаткової інформації та керування їхньою поведінкою. Ось різниця між параметрами та аргументами:

1. Параметри (опції):

* Параметри вказуються в командному рядку після імені команди та зазвичай попереджуються одним або двома дефісами (наприклад, -f або --verbose).
* Вони дозволяють налаштовувати різні аспекти роботи команди, такі як режими роботи, рівень деталізації виводу, шляхи до файлів, тощо.
* Параметри є необов'язковими і можуть бути передані у будь-якому порядку.

Приклад:



У цьому прикладі -l та --all є параметрами команди ls, які встановлюють режим виведення більш деталізованого списку файлів і включають приховані файли.

1. Аргументи:

* Аргументи - це дані або значення, які передаються команді після параметрів.
* Вони вказуються без дефісів та використовуються для конкретних операцій чи обробки.
* Аргументи можуть бути обов'язковими або необов'язковими, в залежності від команди.

Приклад:



У цьому прикладі source.txt і destination/ є аргументами команди cp, які вказують, який файл копіювати та куди його копіювати.

Використання параметрів та аргументів дозволяє командам бути більш гнучкими і спрощує взаємодію користувача з операційною системою, надаючи можливість налаштовувати та керувати їхньою роботою в залежності від потреб.

3. Команда ls в операційній системі Linux та Unix використовується для виведення списку файлів і каталогів у поточному каталозі. Вона дозволяє користувачеві переглядати вміст каталогу. Ось деякі з її можливих параметрів та аргументів:

1. **Параметр** -l **(long format)**:

* Параметр -l дозволяє вивести докладний список файлів і каталогів, включаючи інформацію про дозволи, власника, групу, розмір, час останньої модифікації і ім'я файлу або каталогу.

Приклад:



1. Параметр -a (all)**:**

* Параметр -a дозволяє вивести всі файли, включаючи приховані файли, які починаються з крапки ..

Приклад:



1. **Аргументи (шляхи до каталогів або файлів)**:

* Ви можете передати аргументи, які є шляхами до конкретних каталогів або файлів, і ls виведе список вмісту цих каталогів або відомості про вказані файли.

Приклад:



Це лише декілька прикладів використання команди ls. ls має багато інших параметрів і можливостей для налаштування виводу, і ви можете комбінувати їх для досягнення потрібного результату.

4. Використання історії команд - це корисна функція командного інтерпретатора (shell), яка дозволяє користувачам зберігати та використовувати попередні команди, які вони вводили в командному рядку. Для використання історії команд, зазвичай використовується клавішна комбінація Up Arrow (стрілка вгору) та Down Arrow (стрілка вниз) на клавіатурі.

Ось яким чином можна використовувати історію команд і які переваги це надає:

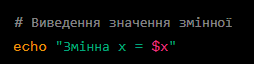
* 1. **Повторення команд**: Натискання Up Arrow дозволяє вам переглядати попередні введені команди. Це дозволяє легко повторювати команди, які ви вже виконували, що зекономлює час і запобігає введенню команди знову з нуля.
  2. **Редагування команд**: Після того як ви відобразите попередню команду, ви можете редагувати її перед виконанням. Це дозволяє виправити помилки чи внести зміни в команду перед її повторним виконанням.
  3. **Пошук команд**: Ви можете швидко знаходити попередні команди за допомогою клавіш Ctrl + R та пошуку за ключовими словами або фразами, що зберігаються в історії. Це особливо корисно при роботі з великим обсягом команд.
  4. **Навчання**: Історія команд також може служити як інструмент для навчання. Ви можете переглядати свої попередні команди, щоб навчитися використовувати команди операційної системи більш ефективно та ефективно.
  5. **Автоматизація**: Ви можете створювати скрипти та командні файли, які використовують історію команд для автоматичного виконання послідовності дій.

В цілому, використання історії команд робить роботу в командному рядку більш зручною, швидкою та продуктивною, спрощуючи доступ до раніше введених команд і забезпечуючи зручний інтерфейс для редагування та повторного виконання команд.

5. Команда echo в операційній системі Linux та Unix використовується для виведення текстового вмісту на екран консолі або у термінал. Основне призначення команди echo - це виведення текстових повідомлень, змінних чи іншого вмісту на екран користувача. Вона може бути корисною в різних сценаріях, включаючи:

* + 1. **Відладка скриптів**: При відладці скриптів або командних файлів команда echo дозволяє виводити значення змінних, проміжні результати та повідомлення для перевірки правильності виконання коду.

Приклад:



* + 1. **Виведення повідомлень для користувача**: Команда echo може використовуватися для виведення інформаційних повідомлень або підказок користувачеві під час виконання скрипту або команди.

Приклад:



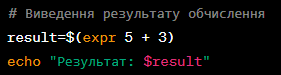
* + 1. **Створення текстових файлів**: Ви можете використовувати команду echo, щоб створювати текстові файли або додавати текст до існуючих файлів.

Приклад:



* + 1. Виведення значень змінних**: Команда** echo **може використовуватися для виведення значень змінних чи інших даних, які вам потрібно відобразити.**

Приклад:



Загалом, команда echo - це важливий інструмент для виведення текстового вмісту в командному рядку, і вона використовується в багатьох скриптах та командах для комунікації з користувачем та передачі інформації.

6. Змінна в оболонці Bash є іменованим об'єктом, який зберігає значення. Вона може бути використана для зберігання текстової інформації, чисел або інших даних, які ви можете використовувати в скриптах, командах та виразах в оболонці. Основні характеристики змінних в оболонці Bash включають:

1. **Імена змінних**: Імена змінних повинні починатися з літери або підкреслення (\_) і можуть містити літери, цифри та підкреслення. Регістр літер важливий (різниця між регістром і маленькими літерами вважається різними змінними).
2. **Присвоєння значень**: Значення можна присвоїти змінній за допомогою оператора =. Наприклад, x=10 призначає змінній x значення 10.
3. **Читання значень**: Значення змінних можна прочитати, звернувшись до них за іменем, наприклад, echo $x виведе значення змінної x.
4. **Змінні середовища**: Bash також підтримує змінні середовища, які доступні у всіх процесах, запущених з оболонкою. Зазвичай імена змінних середовища написані великими літерами (наприклад, PATH).
5. **Локальні та глобальні змінні**: Змінні можуть бути локальними (доступними лише в межах поточного процесу або скрипта) або глобальними (доступними для всіх процесів).
6. **Спеціальні змінні**: Bash має деякі спеціальні змінні, такі як $0 (ім'я запущеної програми), $1, $2, ... (аргументи командного рядка), $# (кількість аргументів командного рядка), $? (код виходу останньої команди), інші.

Типи змінних в оболонці Bash є динамічними, тобто тип даних змінної не визначається заздалегідь і може змінюватися в процесі виконання. Зазвичай змінні в Bash використовуються для зберігання рядкових даних, але також можуть містити числа і інші дані, якщо вони відповідають використовуваним операціям та контексту. Змінні дуже корисні для передачі даних між командами, керування логікою програми та взаємодії з користувачем.

7. Команди env, export, і unset в оболонці Bash використовуються для керування змінними середовища та змінними в оболонці. Основні завдання цих команд наступні:

1. env: Команда env використовується для виведення змінних середовища або виконання команди в специфікованому середовищі. Вона виводить список всіх змінних середовища, доступних для поточного процесу, а також їхні значення.

Приклад виведення всіх змінних середовища:



Приклад виконання команди в специфікованому середовищі:



1. export: Команда export використовується для створення глобальних (експортованих) змінних, які будуть доступні для всіх процесів, запущених з поточною оболонкою. Вона призначена для експорту локальних змінних у глобальне середовище.

Приклад експорту змінної:



Після виконання цієї команди, змінна VAR буде доступна для всіх наступних процесів, запущених з поточною оболонкою.

1. unset: Команда unset використовується для видалення змінних, які були створені в поточному процесі або глобально. Вона використовується для очищення змінних, щоб вони не були більше доступні.

Приклад видалення змінної:



Ця команда призведе до видалення змінної VAR з поточного середовища.

Загалом, команда env допомагає переглядати та керувати змінними середовища, export використовується для створення глобальних змінних, а unset - для видалення змінних. Використовуючи ці команди, ви можете ефективно керувати змінними в оболонці Bash і управляти поведінкою вашої системи.

8. У терміналі Linux або Unix існують кілька способів отримання довідки по командах. Основні команди для цього:

1. man **(manual)**: Команда man використовується для відображення розділу довідкового посібника для певної команди. Наприклад, щоб отримати довідку по команді ls, ви можете виконати:



Ця команда виведе докладний опис команди ls з відомостями про її опції та використання.

1. --help **або** -h: Багато команд підтримують опції --help або -h, які виводять короткий опис команди та список доступних опцій. Наприклад:



1. info: Команда info надає доступ до документації у форматі Info, яка подібна до довідкових посібників man, але може бути більш докладною і інтерактивною. Для отримання довідки по команді, ви можете виконати:



Наприклад:



Після введення цієї команди ви побачите докладний опис команди ls у форматі Info.

1. whatis: Команда whatis виводить короткий опис команди та її призначення. Наприклад:



1. apropos: Команда apropos використовується для пошуку команд, пов'язаних із введеним ключовим словом або фразою. Наприклад:



Ця команда виведе список команд, пов'язаних з ключовим словом "file".

Ці команди допомагають користувачам отримувати докладну довідку та інформацію про доступні команди в системі.

**Хід роботи**

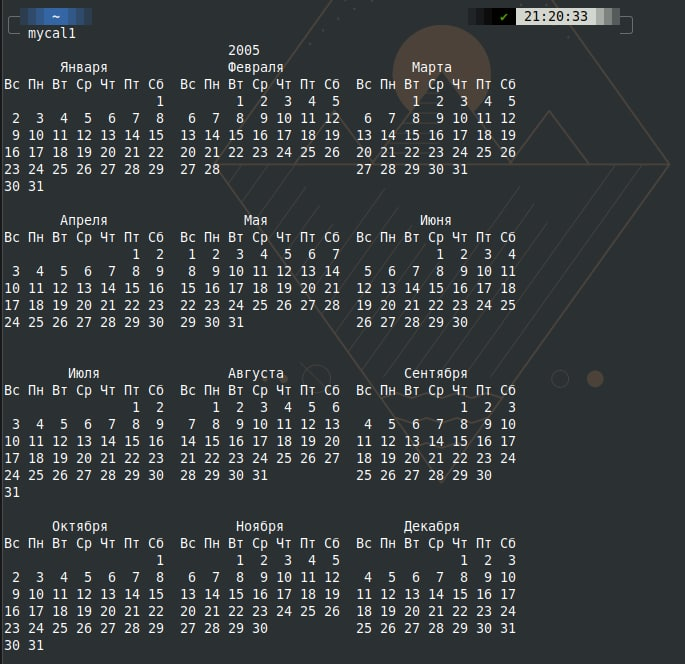
***The material was prepared by a student Dziubenko***

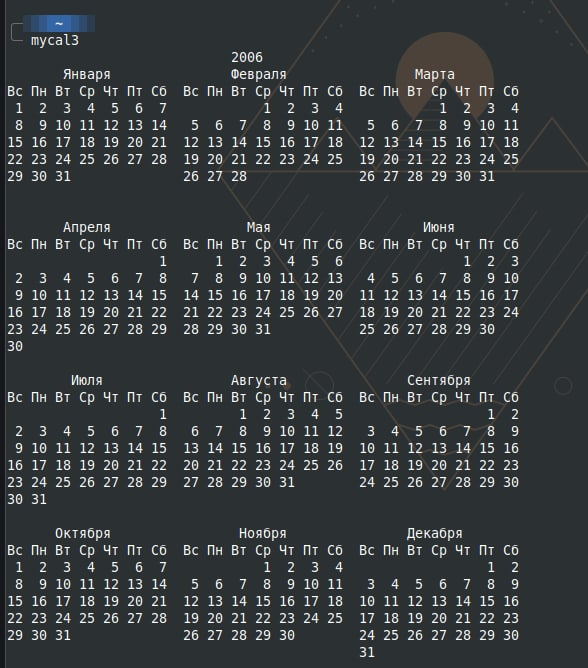
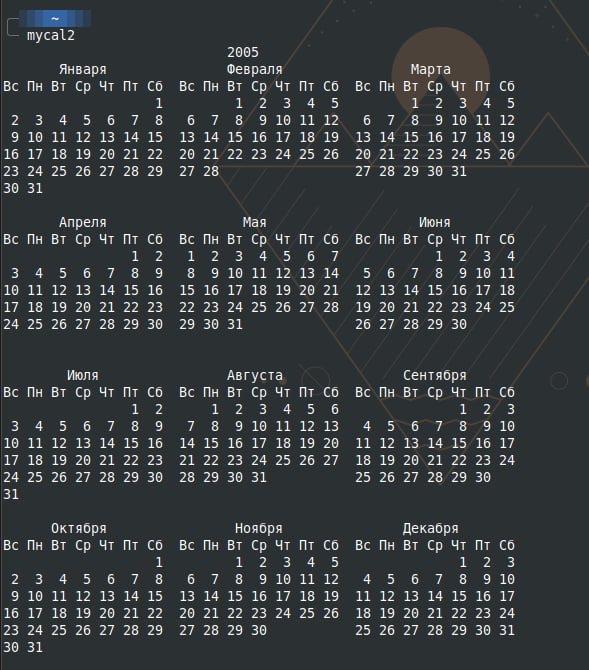
1.

|  |  |
| --- | --- |
| uname | Displays system information. |
| man | Displays manual or help pages. |
| clear | Clears the terminal screen. |
| ls | Lists directory contents. |
| nano | A text editor for the terminal. |
| echo | Prints text or variables to the terminal. |
| bash | Executes the Bash shell. |
| alias | Creates command aliases. |
| source | Executes commands from a file. |
| rm | Removes files or directories. |
| touch | Creates empty files. |

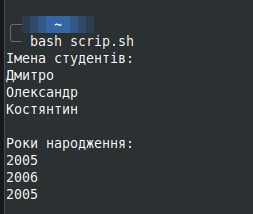
1. Робота в в терміналі



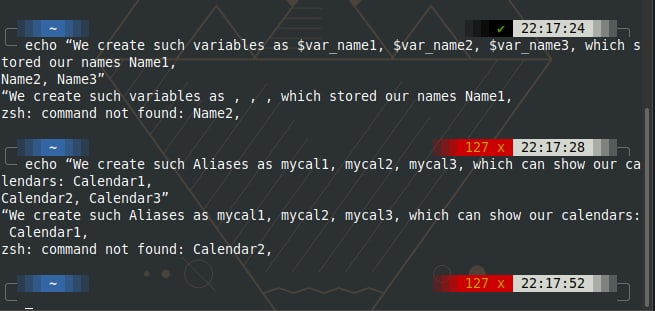




2.2. Робота з функціями в терміналі.



2.3.



2.4. Так, завдання 2.1, 2.2 можуть бути виконані через інструкції керування без написання окремої функції.

Приклад такого коду:

var\_name1="Дмитро"

var\_name2="Костянтин"

var\_name3='Олександр"

echo "$var\_name1"

echo "$var\_name2"

echo "$var\_name3"

alias mycal1="cal 2005"

alias mycal2="cal 2005"

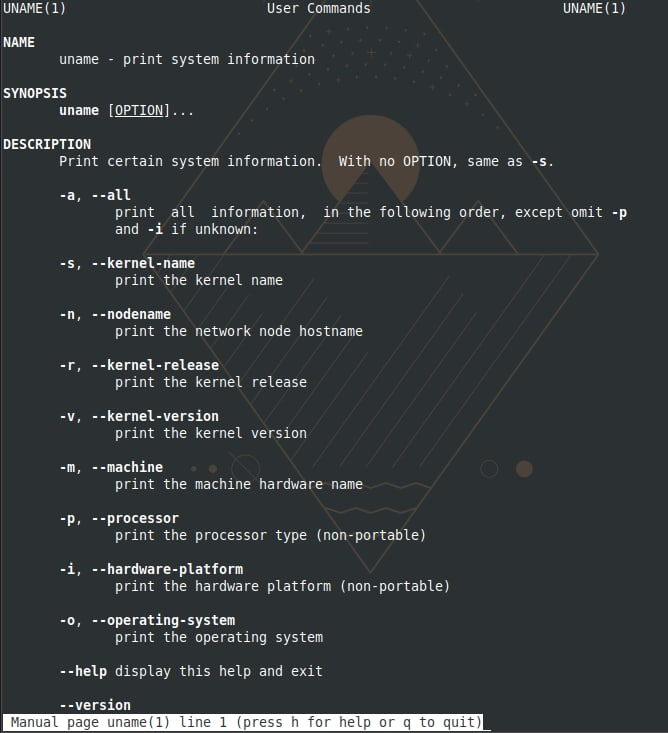
alias mycal3="cal 2006"

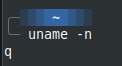
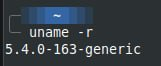
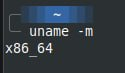
mycal1

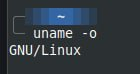
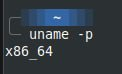
mycal2

mycal3

2.5. Робота з командами довідки (Man Pages) в терміналі.







**Висновки**

***Готував матеріал***