“Київський фаховий коледж зв’язку”

Циклова комісія Комп’ютерної та програмної інженерії

**ЗВІТ ПО ВИКОНАННЮ**

**ЛАБОРАТОРНОЇ РОБОТИ №3**

з дисципліни: «Операційні системи»

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

Виконали студенти

групи БІКС-03

Команда: Яременко О.

Местецький А. Руда В.

Перевірив викладач

Сушанова В.С.

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.

**Лабораторна робота №3 Дисципліна: Операційні системи**

**Завдання для попередньої підготовки:**

*Готував матеріал студенти(цифрою позначено хто-які питання готував ):*

*Антон - 1*

*Вика - 2*

*Олексій – 3*

1. Прочитайте короткі теоретичні відомості до лабораторної роботи та зробіть невеликий словник базових англійських термінів з питань призначення команд та їх параметрів.
2. Вивчіть матеріали онлайн-курсу академії Cisco “NDG Linux Essentials”:

* Chapter 5 - Command Line Skills
* Chapter 6 - Getting Help

1. Пройдіть тестування у курсі NDG Linux Essentials за такими темами:

* Chapter 05 Exam
* Chapter 06 Exam

1. Дайте визначення наступним поняттям:

* Командний інтерпретатор {1}

це програма, яка дозволяє користувачам виконувати команди через текстовий інтерфейс. Командний інтерпретатор аналізує введені команди і виконує їх, забезпечуючи взаємодію з програмою або операційною системою. Зазвичай командний інтерпретатор використовується для автоматизації повторюваних завдань і для доступу до системних ресурсів. Найбільш поширені командні інтерпретатори такі як Bash, PowerShell, і Command Prompt.

* Оболонка {2}
* Оболонка це **програма (комплекс програм), що спрощує роботу з основною програмою**. Наприклад, працювати з операційною системою MS- DOS досить складно — ця система управляється за допомогою команд, і ці команди користувач повинен пам'ятати. Крім того, їх потрібно правильно набирати.
* Команда {3}

1. Дайте відповіді на наступні питання:

* Яку базову інформацію надає рядок запрошення prompt? {1}

Prompt надає користувачеві повідомлення, яке містить базову інформацію, таку як питання, підказку або просто повідомлення. Користувач може ввести текстове значення, яке буде збережене як результат дії prompt.

* Для чого команді потрібні параметри та аргументи? {2}
* Параметри задають інформацію, необхідну виконання команди,а аргумент — це все, що слідує за командою, а параметр (зазвичай, але не завжди) виділяється тире або подвійним тире.
* Яке призначення команд ls, які параметри та аргументи вона може мати? Наведіть 3 приклади. {3}
* Яким чином можна використати історію команд, які переваги це надає? {1}

Історія команд дозволяє користувачам легко повертатися до попередніх команд, виконаних за допомогою поточного інтерфейсу. Це може значно зробити команди більш ефективними, оскільки користувачі не повинні постійно повторювати одні і ті ж команди.

* Яке призначення команди echo? {2}
* Echo — луна) — команда в DOS, OS/2, Microsoft Windows, Unix і Unix-подібних операційних системах, яка **виводить рядок тексту на комп'ютерний термінал**. Це зазвичай використовується в сценаріях оболонок і командних файлах для виводу тексту про стан скрипту на екран або у файл.
* Охарактеризуйте поняття змінної в оболонці Bash, які типи змінних вона підтримує? {3}
* Яке призначення команд env, export та unset? {1}

1. Команда env використовується для перегляду сторінки оточення, в якій працює процес.

2. Команда export використовується для передачі змінної оточення до підпроцесів.

3. Команда unset використовується для видалення змінної оточення.

* Які команди для отримання довідки по командам в терміналі ви знаєте? {2}
* **-h або -help**
* Для отримання довідки по команді використовуйте команду **man**, наприклад: man ls дозволяє отримати довідку на команду ls. apropos “working directory” дозволяє знайти команди, які працюють з поточним каталогом, наприклад знайти команду, яка служить для виведення поточного каталогу.
* Підготувати в електронному вигляді початковий варіант звіту:

1. Підготувати в електронному вигляді початковий варіант звіту:

* Титульний аркуш, тема та мета роботи
* Словник термінів {3}
* Відповіді на п.4 та п.5 з завдань для попередньої підготовки

**Хід роботи:**

* 1. Опрацюйте всі приклади команд, що представлені у лабораторній роботі курсу ***NDG Linux Essentials - Lab 5: Command Line Skills*** та ***Lab 6: Getting Help.*** Створіть таблицю для опису цих команд\*\*\*

|  |  |
| --- | --- |
| Назва команди | Її призначення та функціональність |
| ls | Виводить інформації про каталоги та файли. За замовчуванням без аргументів відображає інформацію для поточного каталогу |
| ls -l | Використанні параметру **-l** в команді **ls** дозволяє відобразити інформацію про файли, розташовані в поточному робочому каталозі, у довгому форматі, який надає більш розширену додаткову інформацію |
| ls -l /tmp | Використання аргументу **/tmp** в поєднанні з параметром **-l** в команді **ls** дозволяєвідобразити детальну інформацію про файли в каталозі /tmp. |
| ls - r | друкує результати в зворотному алфавітному порядку. |
| ls - h | розміри файлів відображатимуться у зручному для читання форматі: |
| history | Якщо потрібна команда є у списку, який генерує команда історії, її можна виконати, ввівши знак оклику! символ, а потім число біля команди, наприклад, щоб виконати команду cal ще раз: |

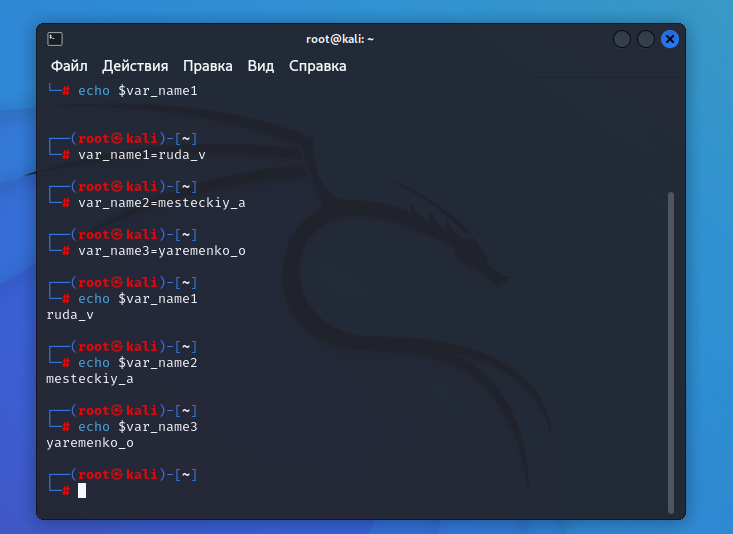
|  |  |
| --- | --- |
| !-n | Щоб виконати n-ту команду знизу списку історії |
| variable=value | Щоб встановити значення змінної, використовуйте наступний вираз присвоєння. Якщо змінна вже існує, значення змінної змінюється. Якщо назва змінної ще не існує, оболонка створює нову локальну змінну та встановлює значення |
| echo | Команда використовується для відображення вихідних даних у терміналі |
| env | Під час запуску без аргументів команда виводить список змінних середовища. |
| type | Команда може використовуватися для визначення інформації про тип команди. |
| cd | Коли користувач вводить команду, оболонка Bash вже виконується та знає, як її інтерпретувати, не вимагаючи запуску додаткових програм. |
| man | переглянути сторінку довідки для команди |
| passwd | Команда є командою користувача, тому відповідна сторінка довідки знаходиться в першому розділі. |

\*\*\***Скріншоти** виконання команд в терміналі можна **не представляти**, достатньо **коротко описати команди в таблиці**.

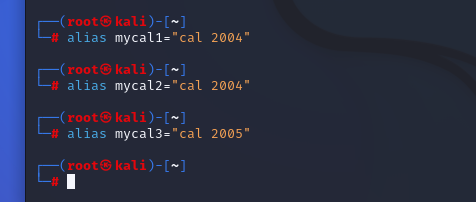
* 1. Робота в в терміналі (закріплення практичних навичок) **обов'язково представити свої скріншоти:**

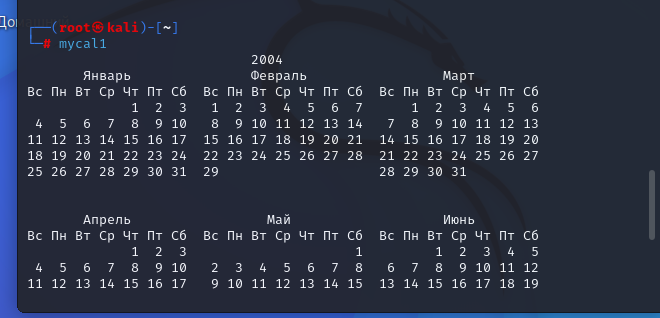
2.1. Робота зі змінними (Variables) та псевдонімами (Aliases) в терміналі:

* Створіть змінні, що будуть містити Ваші імена та прізвища $var\_name1, $var\_name2, $var\_name3
* За допомогою команди echo виведіть імена студентів вашої команди



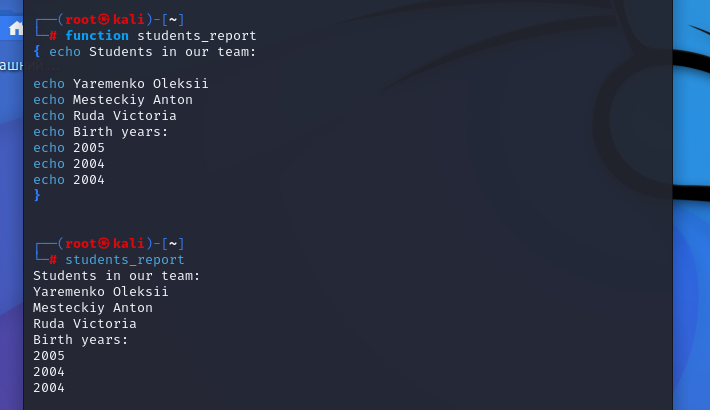
* Створіть псевдоніми mycal1, mycal2, mycal3 для команди cal для автоматичного виведення календарю вашого року народження





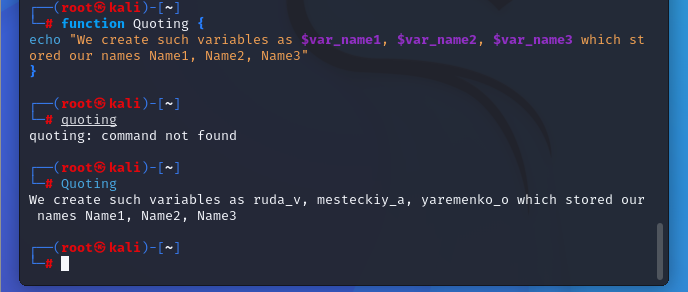
2.2. Робота з функціями (Functions) в терміналі:

* Створіть функцію students\_report, що порядково буде виводити спочатку імена студентів Вашої команди, а потім роки їх народження

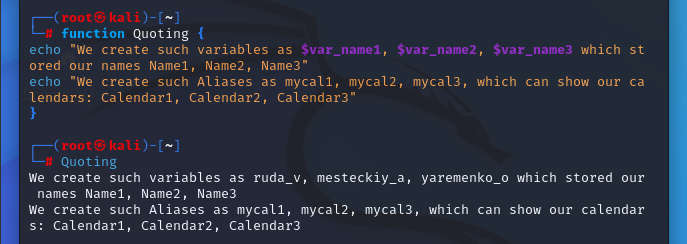


2.3. Робота з лапками (Quoting) в терміналі. Виведіть в командному рядку наступні речення:

* “We create such variables as $var\_name1, $var\_name2, $var\_name3, which stored our names Name1, Name2, Name3” (у реченні спочатку виводимо назви змінних, а потім їх вміст)



* “We create such Aliases as mycal1, mycal2, mycal3, which can show our calendars: Calendar1, Calendar2, Calendar3” (у реченні спочатку виводимо назву команди-псевдонімів, потім вивід цих команд).



2.4. Робота з інструкціями керування (Control Statements) в терміналі:.

* Чи можна завдання 2.1 та 2.2 ходу роботи виконати через інструкції керування без написання окремої функції, як це буде виглядати?

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

* На прикладі команди uname продемонструйте як отримати довідку. На основі отриманої додаткової інформації наведіть 5 різних варіантів виводу результату інформації по даній команді з використанням 5 різних параметрів (Options)
* **Команда uname** з параметром -a застосовується для виведення системної інформації: найменування ядра, його версії, імені хосту, типу процесора та інформації про апаратну платформу.
* Команда uname -o виводить загальне ім’я поточної операційної системи, яке зберігається у змінній сеансу OS. Збереження виведення команди у змінній дуже корисно в сценаріях, завдяки чому можна зберігати та оцінювати дані багатьма різними способами.
* Іноді - особливо під час роботи з автоматизованими сценаріями - вам знадобиться простіше джерело інформації про команду. Команда which, про яку наша попередня команда type надала інформацію, не повертає нічого, крім абсолютного розташування команди. У цьому прикладі розглянуті команди uname та which.
* $ **which uname which**  
  **/**bin/uname  
  /usr/bin/which
* Uname – m - Показати архітектуру ком’ютера

**Контрольні запитання:**

*Готував матеріал студенти(цифрою позначено хто-які питання готував ):*

*Антон - 1*

*Вика - 2*

*Олексій – 3*

1. Які типи команд існують в оболонці Bash? {1}

1) Команди навігації файлової системи: cd, pwd, ls, mkdir, rmdir та ін.

2) Команди пошуку та редагування файлів: grep, sed, awk, find, locate та ін.

3) Команди для запуску та керування процесами: top, ps, kill, nice, nohup та ін.

4) Команди для роботи з мережею: ping, wget, curl, ssh, telnet та ін.

1. Що таке змінні оточення? Які вони бувають. Як їх можна переглянути в терміналі? {2}Змінними оточення в операційних системах на базі ядра Linux називаються ті змінні, які містять текстову інформацію, використовувану іншими програмами під час запуску. Зазвичай вони включають загальні системні параметри як графічної, так і командної оболонки, дані про настройках користувача, розташуванні певних файлів і багато іншого.
2. Опишіть змінну $PS1. Як в терміналі переглянути її вміст? {3}
3. Як можна змінити значення змінної $PS1? Що при цьому відбудеться в рядку запрошенні в bash (рядок запрошення перед початком кожної команди). Як змінити значення цієї змінної не на поточний сеанс, а за замовчуванням?

Значення змінної $PS1 можна змінити, використовуючи команду export. Наприклад, якщо ви хочете змінити значення змінної $PS1 на поточний сеанс, ви можете використовувати команду:

export PS1="\u@\h:\w\$ "

Щоб змінити значення цієї змінної за замовчуванням, ви можете додати цю команду до файлу .bashrc в домашній каталозі користувача. Після цього кожен раз, коли ви відкриваєте новий сеанс bash, значення змінної $PS1 буде змінено на те, що ви вказали в .bashrc. {1}

1. Для чого використовують лапки в оболонці Bash? {2} Лапки використовуються для розмежування рядків, а використання одного або іншого символу залежить від синтаксису, прийнятого на певній мові. В мовах сценаріїв використання лапок і зворотних посилань має інше значення, і Bash в цьому не є винятком.Одиночні лапки використовуються для розмежування рядків символів.Для розмежування рядків використовуються подвійні лапки, однак якщо рядок обмежена цим символом, інтерпретатор виконує так звану «інтерполяцію» і дозволяє значення будь-яких змінних в регулярних виразах Bash в рядку
2. Для чого використовують інструкції керування, які їх види Ви знаєте? {3}
3. В чому різниця якщо в кінці рядку запрошення bash стоїть символ $ чи #? Наприклад на екрані ми бачимо наступні записи {1}





Символ $ використовується, щоб показати, що користувач знаходиться в режимі користувача, а символ # використовується, щоб показати, що користувач знаходиться в режимі адміністратора. Тобто, коли користувач знаходиться в режимі адміністратора, то в кінці рядку запрошення bash стоїть символ #, а коли користувач знаходиться в режимі користувача, то в кінці рядку запрошення bash стоїть символ $.

1. Яке призначення команд whereis та locate? Яка між ними відмінність? {2}

whereis Призначена для пошуку інформації про програми та файли:

Команда locate виконує пошук заданого шаблону у файлі бази даних, згенерованому командою updatedb.

{3} **ВИСНОВОК:**