“Київський фаховий коледж зв’язку”

Циклова комісія Комп’ютерної та програмної інженерії

# ЗВІТ ПО ВИКОНАННЮ ЛАБОРАТОРНОЇ РОБОТИ №1

з дисципліни: «Операційні системи»

# Тема: «Ознайомлення з робочим середовищем віртуальних машин та операційних систем різних сімейств»

Виконав(ла/ли) студент(ка/и) групи РПЗ-93а(б)

Команда 1: Кулик А.О., Марчук Р.С. та Усенко В.В. Перевірив викладач Сушанова В.С.

Київ 2022

## Мета роботи:

1. Отримання практичних навиків роботи з середовищами віртуальних машин та операційними системами різних типів та сімейств – їх графічною оболонкою, входом і виходом з системи, ознайомлення зі структурою робочого столу, вивчення основних дій та налаштувань при роботі в системі.

## 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.

## Якщо завдання виконано групою студентів, то по тексту має бути вказано, хто який фрагмент робив

**Завдання для попередньої підготовки**

### Готував матеріал студент Кулик А. та Марчук Р. (401)

* 1. Прочитайте короткі теоретичні відомості до лабораторної роботи та зробіть невеличкий словник базових англійських термінів з питань класифікації ОС.

|  |  |
| --- | --- |
| Термін англійською | Термін українською |
| **Operating System** | Операційна система |
|  |  |
| … | Інші терміни |

* 1. Прочитавши матеріал з коротких теоретичних відомостей дайте відповіді на наступні питання:

### Готував матеріал студент Марчук Р.

* + 1. Охарактеризуйте поняття «гіпервізор». Які бувають їх типи?

*Даємо відповідь … Гіпервізор це ...*

## Хід роботи

### Готував матеріал студент Марчук Р.

1. Робота в графічному режимі в ОС сімейства Linux:.
   1. Запустіть віртуальну машину VirtualBox, ознайомтесь з її основними можливостями, прочитайте довідку по роботі з нею.

## Наступні пункти ходу роботи

*Ваші відповіді*

## Відповіді на контрольні запитання

### Готував матеріал студент Топехін Б.

1. Які типи команд існують в оболонці Bash?
   * Вбудовані команди,Зовнішні команди,Псевдокоманди,Команди пайпу (pipe),Команди зворотнього апострофа (backticks)
2. Що таке змінні оточення? Які вони бувають. Як їх можна переглянути в терміналі?
   * Змінні оточення - це змінні, які зберігають інформацію про оточення і конфігурацію операційної системи, доступну для процесів та програм. Вони впливають на роботу програм і служать для зберігання інформації про параметри середовища.
   * Для перегляду значень змінних оточення в терміналі використовуйте команду “echo”.
   * Для перегляду змінних оточення в терміналі використовуйте команду “echo $ЗМІННА”, де “ЗМІННА” - це ім'я змінної оточення, яку ви хочете переглянути. Наприклад, “echo $PATH” виведе значення змінної “PATH”.
3. Опишіть змінну $PS1. Як в терміналі переглянути її вміст?
   * Змінна “$PS1” в оболонці Bash визначає формат та вигляд приголомшливого рядка (примірника) перед командним рядком в терміналі. Щоб переглянути її вміст в терміналі, використовуйте команду “echo $PS1”.
4. 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?

* + Змінна $PS1 використовується в оболонці Bash для налаштування рядка запрошення (prompt), який відображається перед вводом кожної команди. Ви можете змінити значення цієї змінної, щоб налаштувати власний вигляд рядка запрошення. Щоб змінити значення $PS1, введіть наступну команду в терміналі Bash: PS1="нове\_значення"
  + Де "нове\_значення" - це рядок, який ви хочете встановити як новий рядок запрошення. Наприклад, щоб встановити рядок запрошення, який включає поточний каталог і символ "$" як знак команди, ви можете використовувати таку команду: PS1="\w $ "
  + Після введення цієї команди ваш рядок запрошення буде виглядати як щось на зразок: /home/user/Documents $
  + Для того, щоб зберегти це значення $PS1 на постійну, ви можете додати команду в свій файл налаштувань оболонки, такий як "/.bash\_profile".

Додайте рядок зі зміною $PS1 в цей файл, і вона буде встановлена автоматично при запуску нового сеансу оболонки. Наприклад: echo 'PS1="\w $ "' >> ~/.bashrc

* + Після цього відкрийте новий термінал або виконайте команду source

~/.bashrc, щоб застосувати зміни..Змінити значення $PS1 за замовчуванням можна, внісши зміни в конфігураційний файл оболонки. Для цього:

* + Відкрийте конфігураційний файл оболонки, наприклад, ~/.bashrc для користувача або /etc/bash.bashrc для всіх користувачів.
  + Додайте рядок, що встановлює нове значення $PS1. Наприклад:
  + Для користувача: PS1="\w $ "
  + Для всіх користувачів (потрібні права адміністратора): sudo echo 'PS1="\w

$ "' >> /etc/bash.bashrc

* + Збережіть файл.
  + Відкрийте новий термінал або виконайте команду source ~/.bashrc або source /etc/bash.bashrc (залежно від файлу, який ви редагували), щоб застосувати зміни. Змінене значення $PS1 буде використовуватися за замовчуванням у нових сеансах оболонки.

1. Для чого використовують лапки в оболонці Bash?

Лапки в оболонці Bash використовуються для захисту вмісту від інтерпретації та для визначення рядків. Вони дозволяють:

* + 1. Захищати вміст від інтерпретації змінних: Використання лапок дозволяє вставляти значення змінних в рядки без їхньої автоматичної інтерпретації. Наприклад,

"«Значення $змінної" • виведе рядок зі словом \* "Значення" \* і самою змінною $змінною", а не її значенням.

* + 2. Визначати рядки: Лапки дозволяють визначити значення рядків, включаючи спеціальні символи та змінні. Наприклад, " "Привіт, $імʼя!"" визначає рядок, в якому $імʼя" замінюється на його значення.
  + 3. Створювати командні рядки: У випадку виконання команд в Bash, подвійні лапки

використовуються для визначення командного рядка, який буде виконуваний.

Наприклад, \*"ls -1" • створить командний рядок для виклику команди 'Is з параметром '-1'

* + Загалом, використання лапок допомагає контролювати інтерпретацію змінних, спеціальних символів та створювати рядки з бажаним вмістом в оболонці Bash.

1. Для чого використовують інструкції керування, які їх види Ви знаєте?
   * Інструкції керування в програмуванні використовуються для контролю над виконанням програми та управління потоком виконання. Основні види інструкцій

керування включають:

1. Умовні інструкції (if-else): Використовуються для виконання певного блоку коду, якщо задана умова істинна, та інш§го блоку коду, якщо умова хибна.
2. Цикли (for, while, do-while): Використовуються для повторення блоку коду декілька разів. "for\* використовується для ітерації через

послідовність. while\* i do-while\* використовуються для повторення, поки задана умова виконується.

1. Переривання (break, continue): Використовуються в циклах для виходу із циклу (break) або переходу до наступної ітерації (continue).
2. Перемикачі (switch/case): Використовуються для вибору одного з багатьох варіантів дій, в залежності від значення виразу.
3. Функції та підпрограми: Використовуються для організації коду в окремі функції або процедури, які можуть бути викликані з різних частин програми.
4. Винятки (try/except): Використовуються для обробки виняткових ситуацій (виключень), які можуть виникнути в процесі виконання програми.
5. Заборона памʼяті (goto): Ця інструкція рідко використовується і не рекомендується в сучасному програмуванні, оскільки може призвести до непередбачуваного та

важкочитаного коду.

1. В чому різниця якщо в кінці рядку запрошення bash стоїть символ $ чи #?
   * різниця між символами '$ і # в промпті полягає в рівні привілеїв користувача. $ вказує на звичайного користувача, а # на суперкористувача або адміністратора системи.
2. Яке призначення команд whereis та locate? Яка між ними відмінність?
   * whereis призначений для пошуку виконувих файлів та документації програм, і він швидше, але менш універсальний в порівнянні з locate.
   * locate більш загальний та потужний і може знаходити будь-які файли та директорії у системі, але він вимагає оновленої бази даних і може бути повільним при першому запуску.
   * Якщо вам потрібно швидко знайти бінарний файл або документацію програми, whereis може бути зручнішим. Однак, якщо вам потрібно шукати файли загалом по всій системі, locate є більш ефективним інструментом.

-

### Готував матеріал студент Petrov.

1. Наступні контрольні запитання та відповіді на них

## Висновки

В ході виконання лабораторної роботи мною було досліджено … , більш детально теоретично досліджено питання …. Отримано практичні навики роботи з командами

…, налаштування … ***(Якщо виникли труднощі, то їх описати)***