“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 **КСМ-23А**

**Команда Bald-Hedgehogs**:

**Кошкін І.О, Сарапин Я.О, Кочубей С.С**

Перевірила викладач

Сушанова В.С.

Київ 2024

Робота студентів групи КСМ-23А Команда **Bald-Hedgehogs:** Кошкін І.О, Сарапин Я.О, Кочубей С.С.

**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 та віртуальна машина Virtual Box (Oracle).

3. ОС GNU/Linux (будь-який дистрибутив).

4. Сайт мережевої академії Cisco netacad.com та його онлайн курси по Linux

**Завдання для попередньої підготовки:**

***Готував матеріал студент Кошкін І.О***

1. \*Дайте визначення наступним поняттям:

* Командний інтерпретатор
* Оболонка
* Команда

**Командний інтерпретатор** - програма, яка забезпечує взаємодію користувача з операційною системою.

**Оболонка** - це інтерпретатор командного рядка, який перетворює команди, введені користувачем, у дії, які виконує операційна система. Якщо команда створює вивід, то в терміналі відображається текст. Якщо виникають проблеми з командою, відображається повідомлення про помилку.

**Команда** - це програмне забезпечення, яке під час виконання через інтерфейс командного рядка виконує дію на комп’ютері.

1. Дайте відповіді на наступні питання:

* Яку базову інформацію надає рядок запрошення prompt?
* Для чого команді потрібні параметри та аргументи?
* Яке призначення команд ls, які параметри та аргументи вона може мати? Наведіть 3 приклади.
* Яким чином можна використати історію команд, які переваги це надає?
* Яке призначення команди echo?
* Охарактеризуйте поняття змінної в оболонці Bash, які типи змінних вона підтримує?
* Яке призначення команд env, export та unset?
* Які команди для отримання довідки по командам в терміналі ви знаєте?
  1. Рядок запрошення, як правило, містить інформацію про: Ім'я користувача, ім'я хоста, головний каталог, тип оболонки та додаткову інформацію.
  2. Бо команди дозволяють налаштовувати роботу команди і передавати їй додаткові дані.
  3. Команда **ls** використовується для виведення списку файлів в поточному або вказаному каталозі. Приклади:

Параметри:

ls -l — детальний список файлів у поточній директорії.

ls -a — відображає всі файли, включаючи приховані.

ls -lh /home/user — детальний список файлів у каталозі.

* 1. Історію команд у Bash можна використовувати для повторного виконання раніше введених команд, що значно спрощує роботу. За допомогою команди history можна переглянути список команд, а клавіші стрілки або комбінація Ctrl + R дозволяє швидко знаходити та повторювати їх. Це дозволяє уникнути повторного введення команд, виправляти помилки та аналізувати дії, що покращує ефективність роботи.
  2. Команда **echo** використовується для виведення тексту або значень змінних на екран.
  3. **Змінна** в оболонці Bash – це іменована область пам'яті, яка використовується для зберігання значень.

**Типи змінних** **яка вона підримує:** Локальні змінні, глобальні змінні, позиційні параметри, спеціальні змінні.

* 1. Команда env - виводить список усіх змінних середовища та їхніх значень, команда unset - видаляє змінну зі списку змінних середовища а команда export - додає змінну до списку змінних середовища або змінює значення існуючої змінної.
  2. Для отримання довідки я знаю такі команди: man, info, help, which та --help

**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

(Kochubei S. 9v)

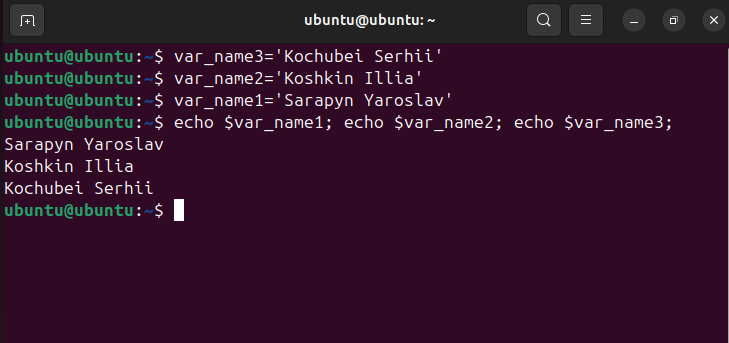
| Назва команди | Призначення та функціональність |
| --- | --- |
| ls | Виводить вміст поточного каталогу. |
| ls -l | Виводить детальний список файлів у поточному каталозі. |
| ls -l /home | Виводить детальний список файлів у каталозі **/home.** |
| whoami | Виводить ім'я користувача, під яким ви увійшли в систему. |
| uname | Виводить інформацію про операційну систему. |
| uname -n | Виводить ім'я хоста (nodename) комп'ютера. |
| uname --nodename | Те ж саме, що й uname **-n** — виводить ім'я хоста. |
| pwd | Виводить шлях до поточного робочого каталогу. |
| echo Hi | Виводить "Hi" на екран. |
| history | Виводить історію виконаних команд. |
| history 5 | Виводить останні 5 команд з історії. |
| !9 | Виконує 9-ту команду з історії. |
| echo Hello Student | Виводить "Hello Student" на екран. |
| echo $PATH | Виводить вміст змінної **PATH**, яка містить шляхи до виконуваних файлів. |
| which date | Показує шлях до виконуваного файлу date. |
| type command | Показує тип команди (чи це вбудована команда оболонки, зовнішня, функція і т. д.). |
| which ls | Виводить шлях до виконуваного файлу **ls**. |
| type vi | Вказує, що **vi** є зовнішньою програмою. |
| cd /bin | Переміщує вас у каталог **/bin.** |
| type vlc | Вказує, чи встановлена програма **vlc**, і її тип. |
| cd | Переміщує в домашній каталог користувача. |
| echo Today is \date`` | Виводить текст "Today is" та поточну дату, виконуючи команду **date** |
| echo Today is $(date) | Виводить текст "Today is" та поточну дату (синтаксис аналогічний попередньому). |
| echo This is the command 'date' | Виводить текст "This is the command" та результат виконання команди date в лапках. |
| echo This is the command \date`` | Аналогічно попередньому — виводить результат команди **date** у рядку. |
| echo This is the command "date" | Виводить текст та результат команди **date** у подвійних лапках. |
| echo D\* | Виводить усі файли, що починаються на D в поточному каталозі. |
| echo "D\*" | Виводить рядок "D\*" (без розширення). |
| echo Hello; echo Linux; echo Student | Послідовно виконує кілька команд, виводить "Hello", "Linux" і "Student". |
| false; echo Not; echo Conditional | Виконує false (несправжня команда) та дві інші команди; всі вони виконаються незалежно. |
| echo Start && echo Going && echo Gone | Виконує ланцюжок команд, якщо попередня команда була успішною (логічне "і"). |
| echo Success && false && echo Bye | Виводить "Success", але через помилку false наступна команда echo Bye не виконується. |
| `false |  |
| `true |  |
| date | Виводить поточну дату та час. |
| man date | Відкриває сторінку довідки для команди date. |
| /file | Використовується в man для пошуку слова "file" у документації (пошук в тексті). |
| man -k password | Шукає сторінки довідки за ключовим словом "password". |
| apropos password | Виводить список команд, які пов'язані з "password", аналогічно до man -k. |
| man -f passwd | Виводить короткий опис команди passwd, аналогічно до whatis. |
| man 5 passwd | Відкриває сторінку довідки для команди passwd, розділ 5 (формат файлу). |
| whatis passwd | Виводить короткий опис команди passwd (для зміни пароля). |
| info date | Виводить детальнішу інформацію про команду date у форматі info. |
| date --help | Виводить коротку довідку з усіма опціями команди date. |
| ls /usr/share/doc | Показує вміст каталогу /usr/share/doc, який містить документацію для встановлених програм. |
| locate crontab | Шукає файли та каталоги з назвою, що містить "crontab". |
| locate -b "\crontab" | Шукає файли з точним ім'ям "crontab". |
| whereis passwd | Виводить місце знаходження виконуваного файлу, сторінок довідки та вихідного коду для passwd. |

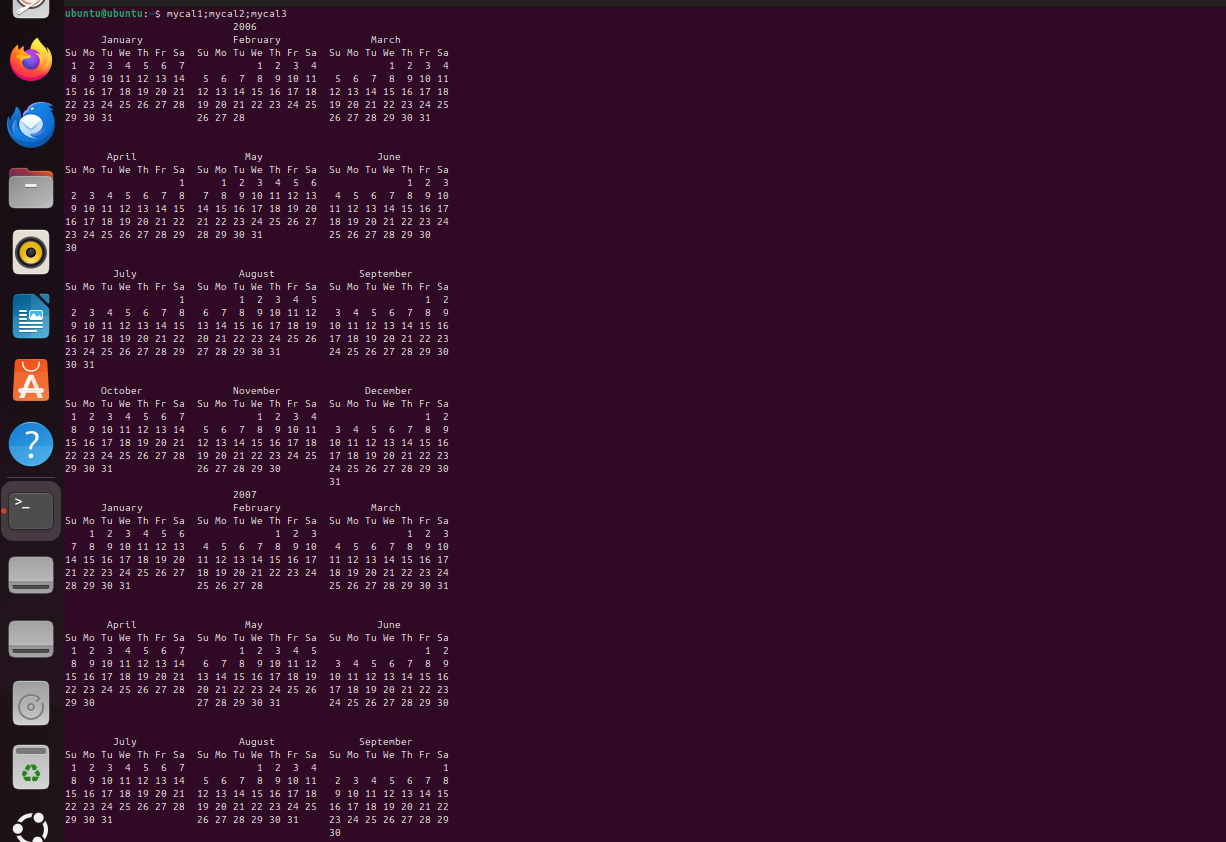
2 Робота в в терміналі (закріплення практичних навичок) (Sarapyn, Kochubei, Koshkin)

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

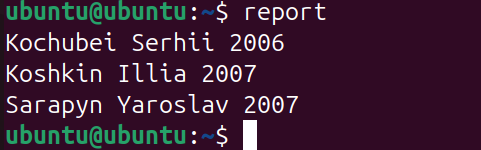
Створіть псевдоніми mycal1, mycal2, mycal3 для команди cal для автоматичного виведення календарю вашого року народження





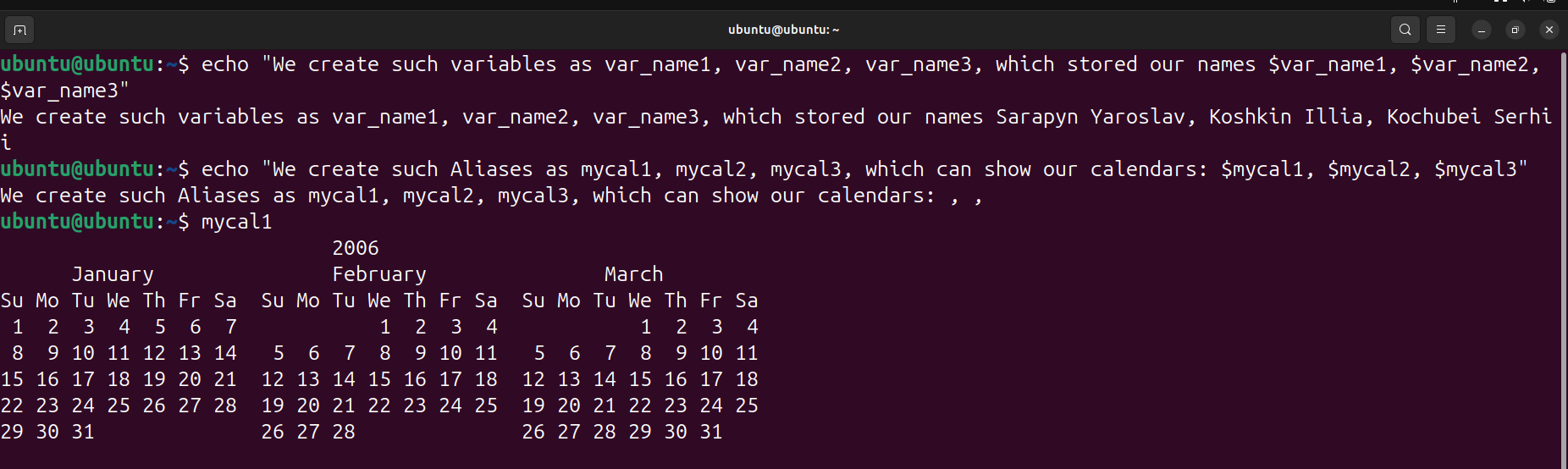
2.2 \*Робота з функціями (Functions) в терміналі:

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



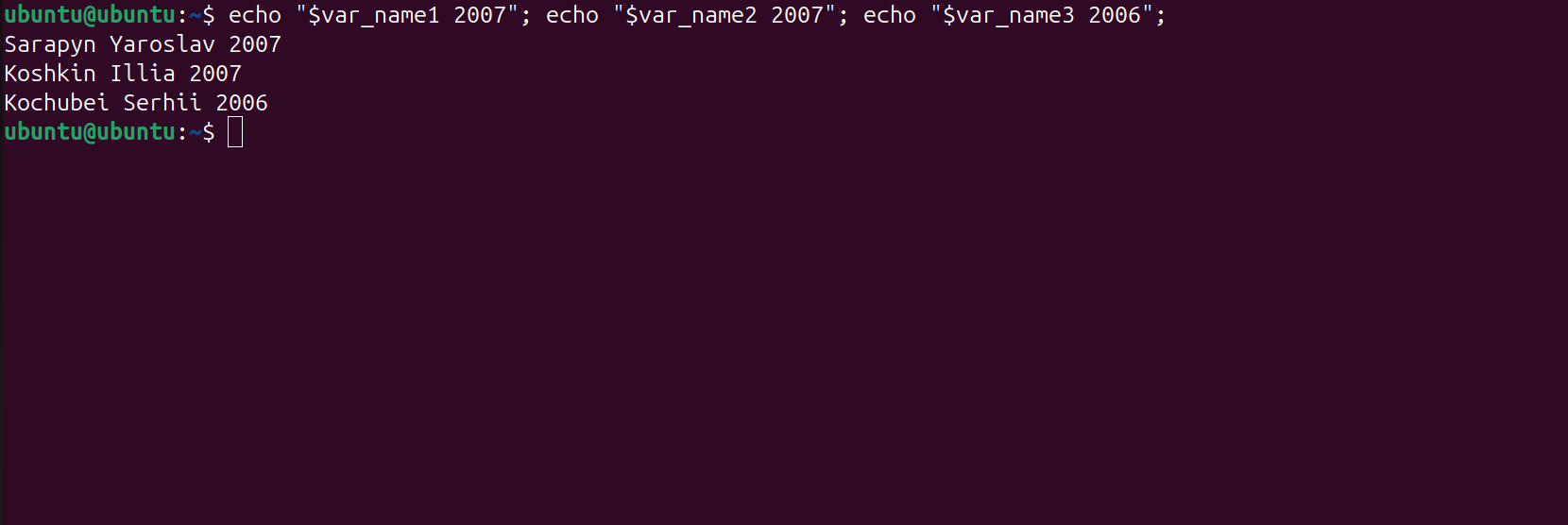
2.3 \*Робота з лапками (Quoting) в терміналі. Виведіть в командному рядку наступні речення:

* “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).



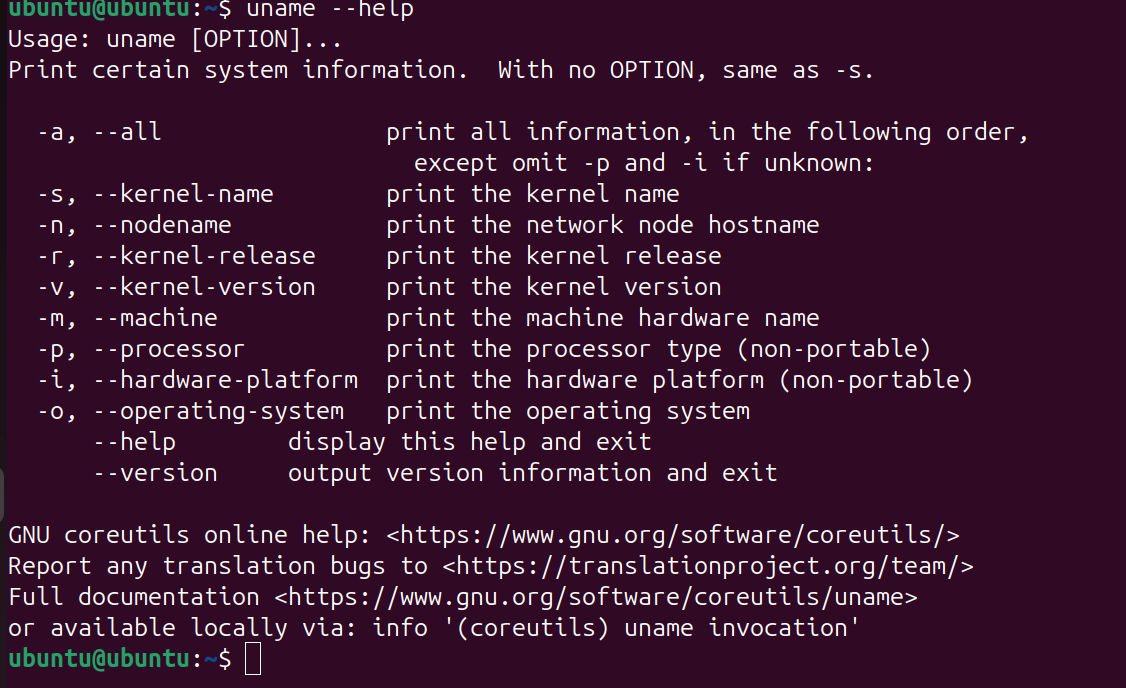
2.4 \*\*Робота з інструкціями керування (Control Statements) в терміналі:.

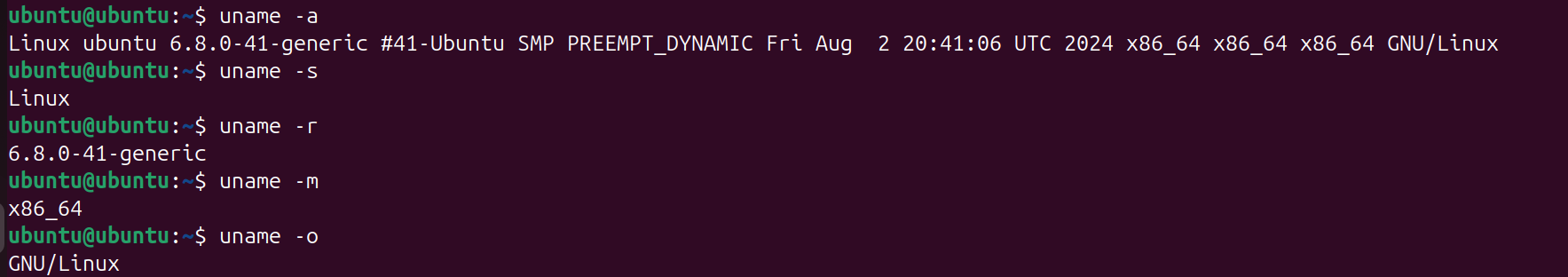
* Чи можна завдання 2.1 та 2.2 ходу роботи виконати через інструкції керування без написання окремої функції, як це буде виглядати?



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)





**Контрольні запитання: (**Сарапин**)**

1. Які типи команд існують в оболонці Bash?
2. Що таке змінні оточення? Які вони бувають. Як їх можна переглянути в терміналі?
3. \*Опишіть змінну $PS1. Як в терміналі переглянути її вміст?
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?
5. \*Для чого використовують лапки в оболонці Bash? .
6. \*\*Для чого використовують інструкції керування, які їх види Ви знаєте?
7. \*\*В чому різниця якщо в кінці рядку запрошення bash стоїть символ $ чи #? Наприклад на екрані ми бачимо наступні записи



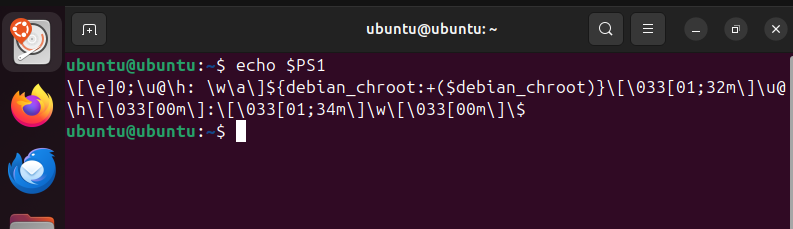


1. \*\*Яке призначення команд whereis та locate? Яка між ними відмінність?
2. There are several different sources of commands within the shell: internal commands, external commands, aliases, functions.
3. Environment variables (global variables) are available system-wide. There are types of this variables: PATH, HOME, and HISTSIZE variables.

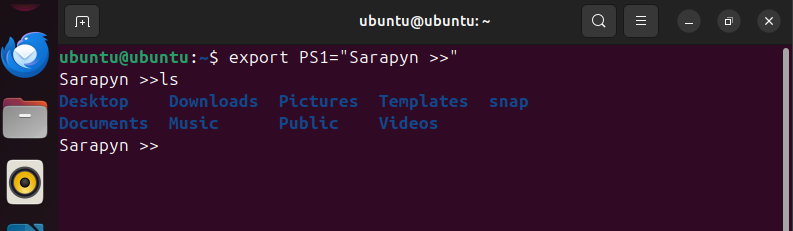
The env command allows to show a list of the environment variables.

1. The $PS1 variable in Bash is responsible for the format and appearance of the prompt string.

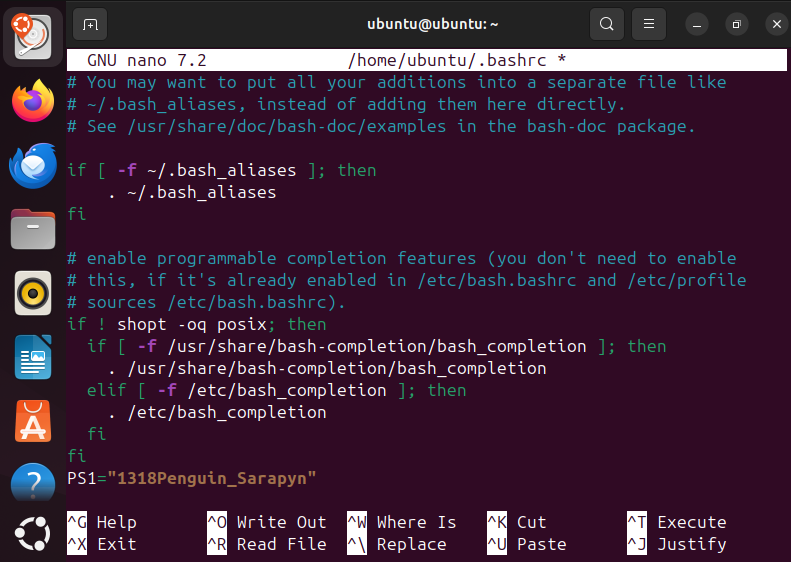
We can write the command echo $PS1 to get the value of a variable:



1. And can write the command export PS1=”new value” to change it temporary



Also we can write the command sudo nano ~/.bashrc and add this string: “PS1=<new value>” and refresh the BASH service (source ~/.bashrc) to change the value permanently:



1. Double quotes:

* Used when you need to nest variables or use shell expansion inside a string.
* All characters within the bounds are interpreted as normal characters, except $ or ', which will be expanded on the shell.

Single quotations- all characters in single quotes are interpreted as a character string.

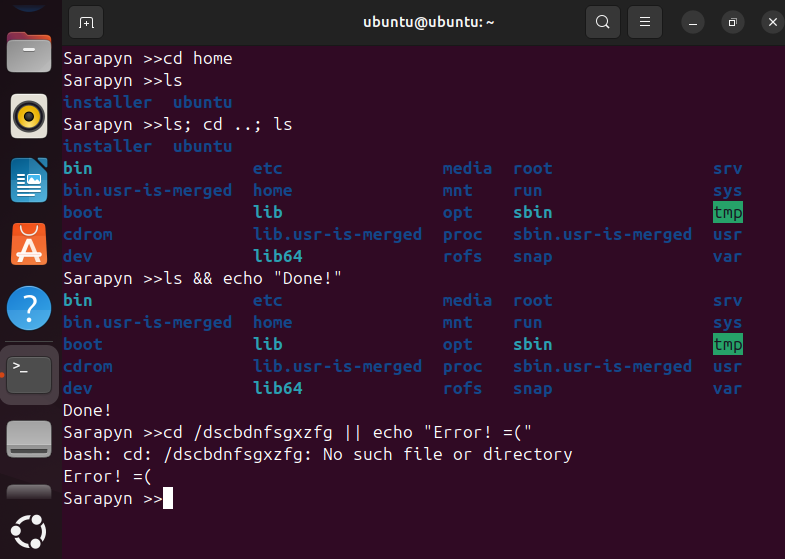
1. Control statements allow you to use multiple commands at once or run additional commands.

The semicolon can be used to run multiple commands, one after the other.

The double ampersand && acts as a logical "and" if the first command is successful, then the second command (to the right of the &&) will also run.

The double pipe || is a logical "or". It works similarly to &&; depending on the result of the first command, the second command will either run or be skipped.

Example:



1. When the symbol $ this means that the user does not have root rights. When the symbol is #, the user can execute all commands on behalf of the superuser (root) and not use the sudo utility to perform actions that require root rights
2. locate is used to scan the whole system quickly for something, and the whereis command simply returns the location of the executables, the man pages and the sources of a program.

**Висновки: (Kochubei S. 9v)**

In this work, we familiarized ourselves with basic CLI commands in Linux and learned essential text-based commands for terminal operations across different operating systems. We explored commands such as ls, cd, pwd, mkdir, rm, man, echo, date, and history, among others. These commands enabled us to navigate the file system, manage files and directories, and access help documentation efficiently. This foundational knowledge is crucial for effective command-line usage and system management. We also played with commands and learned how to interact with functions and aliases.