Міністерство освіти і науки України

НТУУ «КПІ ім. Ігоря Сікорського»

Навчально-науковий інститут атомної та теплової енергетики

Кафедра цифрових технологій в енергетиці

Лабораторна робота №3 з дисципліни «Операційна система UNIX»

**Робота в оболонці bash, середовище оточення**

**Варіант: 16**

Виконав студент групи ТР-12

Каркушевський Владислав

Перевірила д.т.н., проф. Левченко Л. О.

КИЇВ 2022

**Мета роботи :** набути навичок налаштування програмного середовища Linux.

**Теоретична частина**:

**Командні оболонки ОС Unix**

Всі сучасні ОС сімейства Unix забезпечують підтримку трьох типів інтерфейсів користувача:

* інтерфейс графічної оболонки (X Window),
* інтерфейс командної оболонки (Shell),
* інтерфейс прикладного програмування (API).

Графічна оболонка реалізує модель робочого столу (меню встановлених системних і прикладних програм, панелі управління і відображення станів ОС та її компонентів, віконні та іконографічні інтерфейси процесів ОС і т.п.) і призначена, головним чином, для кінцевих користувачів ОС.

Командна оболонка виконує наступні основні функції:

* організує діалог з користувачем шляхом введення команд;
* є інтерпретатором рядків команд, що складаються з команд ОС, системних
* утиліт і вбудованих в оболонку програмних каналів обміну даними;
* виконує внутрішні команди;
* запускає зовнішні програми;
* виконує командні файл;
* має власну мову сценаріїв, яка істотно розширює її функціональні можливості (зокрема, при вирішенні задач адміністрування ОС).

Інтерфейс прикладного програмування (АРІ) забезпечує доступ додатків, написаних на алгоритмічних мовах високого рівня до бібліотеки системних викликів ядра ОС. В системах Unix використовуються різні командні оболонки (command shells), які називаються також командними процесорами або інтерпретаторами команд (CLI - Command Language Interpreter)).

Командна оболонка забезпечує взаємодію між користувачем та ядром системи. Серед них найбільш відомі і поширені:

* sh (Bourne shell) /bin/sh - оболонка Борна, класична оболонка Unix (не дуже зручна в роботі);
* csh (C-shell) /bin/csh - оболонка С, в якій використовується синтаксис мови С (зручніша у порівнянні з sh, але несумісна з нею по командній мові);
* ksh (Korn shell) /bin/ksh - оболонка Корна (містить потужну командну мову, засновану на мові sh, та розвинені засоби інтерактивної роботи);
* bash (Bourne-Again Shell) /bin/bash - оболонка «Борна», найбільш поширена оболонка сімейства Unix (зручна для інтерактивної роботи, створена на основі sh і багато в чому з нею сумісна);
* Z shell, zsh /bin/zsh - одна з сучасних командних оболонок UNIX, що використовується безпосередньо як інтерактивна оболонка, або як скриптовий інтерпретатор. Zsh є розширеним bourne shell з великою кількістю розширених можливостей підстановки команд, автодоповнення та налаштувань.

Всі команді оболонки, встановлені в операційних системах Linux, прописані у

файлі /etc/shells.

В даний час в ОС сімейства Unix визнано стандартами командну оболонку BASH (Bourne-Again Shell) та інтерфейс прикладного програмування POSIX (Portable Operating System for Unix).

Тип оболонки, як правило, можна визначити за останнім символом запрошення:

* знак долара «$» вказує на sh-сумісну оболонку (sh, bash, ksh),
* знак амперсанта «&» відповідає оболонці csh.

Однак у привілейованого користувача незалежно від командного процесора, який використовується, останнім символом запрошення зазвичай є знак решітки «#».

В графічній оболонці програма «Термінал», яка запускається комбінацією клавіш «Ctrl+Alt+T» або ярлик із загального меню програм , це важливий елемент операційної системи, який дозволяє запускати програми, створювати папки, копіювати і видаляти файли, встановлювати додатки і т.д. Системна утиліта, в яку ви передаєте ці команди, називається Shell або командна оболонка. За замовчуванням в Ubuntu використовується командна оболонка Bash. Файли .bashrc, .bash\_profile та .profile - це файли налаштування нашої оболонки. Вони складаються з команд bash і виконуються перед тим як запустити оболонку або завантажити систему. Різниця цих файлів полягає в тому, коли вони виконуються: .profile і .bash\_profile виконуються один раз при вході користувача в систему (логін користувача). Файл .bashrc виконується кожний раз, коли ви відкриваєте нове вікно терміналу (аналог автозавантаження в Windows).

**Синтаксис команд в Терміналі**

Команди Терміналу, як правило, складаються з назви програми, ключа і значення. В загальному вигляді виглядають так:

назва\_ проrрами [-ключ] [значення].

назва\_проrрами - це ім'я виконуваного файлу з каталогів, записаних у змінну $РАТН (/bin, /sbin, /usr/bin, /usr/sbin, /usr/local/bin, /usr/local/sbin та ін.);

[ключ] - опції програми, які може приймати виконувана програма;

[значення] - даний параметр може приймати в якості аргументу цифри, текст,

спеціальні символи і навіть змінні.

Оболонка має свої налаштування та забезпечує доступ до ресурсів системи, користувач має свої налаштування при роботі з системою та своїми додатками. Для цього використовується середовище оточення.

Середовище оточення - це область, яка містить визначальні властивості системи у вигляді змінних, і це середовище оболонка будує при кожному запуску сесії. Такі змінні можуть містити загальні налаштування системи, параметри графічної або командної оболонки, дані про вподобання користувача, місце розміщення виконуваних файлів в системі, ім'я текстового редактора і багато іншого. Змінні оточення дозволяють простим і надійним способом передавати налаштування відразу для безлічі додатків.

Таким чином, змінні оточення в Linux - це спеціальні змінні, які визначені оболонкою і використовуються програмами під час виконання. Вони можуть визначатися системою і користувачем. Системні змінні оточення Linux визначаються системою і використовуються програмами системного рівня.

Змінні оточення бувають трьох типів:

*1. Локальні змінні оточення*.

Це змінні, які визначені лише для поточної сесії. Вони будуть зразу стерті після завершення сесії, будь то віддалений доступ або емулятор терміналу. Вони не зберігаються ні в яких файлах, а створюються і видаляються за допомогою спеціальних команд.

*2. Змінні користувача в середовищі оболонки.*

Ці змінні оболонки в Linux визначаються для конкретного користувача і завантажуються кожний раз, коли він входить в систему за допомогою локального терміналу, або ж підключається віддалено. Такі змінні зберігаються в файлах конфігурації: .bashrc, .bash\_profile, .bash\_login, .profile або в інших файлах, розміщених в директорії користувача. Ці файли є налаштуванням оболонки користувача. Вони складаються з команд bash і виконуються перед тим як запустити оболонку або завантажити систему.

Різниця цих файлів полягає у тому, коли вони виконуються, а саме:

.profile та .bash\_profile виконуються при вході в систему, тобто один раз- це при введенні логін користувача. Файл .bashrc виконується кожний раз, коли користувач відкриваєте нове вікно терміналу (аналог автозавантаження в Windows). Bash також працює зі скриптами (сценаріями - scripting language), тобто списком команд, які записані у файл. Сценарій – це програма, яка працює з готовими компонентами.

*3. Системні змінні оточення*

Ці змінні доступні у всій системі та для всіх користувачів. Вони завантажуються при старті системи з системних файлів конфігурації: /etc/environment, /etc/profile, /etc/profile.d/\*.sh, /etc/bash.bashrc:

- /etc/environment –системний файл конфігурації, який означає, що він використовується всіма користувачами. Він належить root, тому необхідно бути адміністратором і використовувати його sudo для зміни;

- ~/.profile є одним із сценаріїв ініціалізації особистої оболонки вашого власного користувача. Кожний користувач має один такий файл і може його редагувати.

- /etc/profile та /etc/profile.d/\*.sh є сценаріями глобальної ініціалізації, які є еквівалентними ~/.profile для кожного користувача. Глобальні сценарії виконуються раніше користувацьких сценаріїв; і main /etc/profile виконує усі \*.sh сценаріі з директорії /etc/profile.d/ безпосередньо перед виходом;

- /etc/bash.bashrc - це загальносистемна версія файлу ~/.bashrc. ОС Ubuntu налаштована за замовчуванням для виконання цього файлу, коли користувач вводить оболонку або середовище робочого столу.

Файл /etc/environment зазвичай містить тільки наступний рядок:

PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games"

Він встановлює змінну PATH для всіх користувачів в системі в це значення за замовчуванням, яке не повинно бути змінено основним способом. Принаймні, ви не повинні видалити будь - який з таких важливих шляхів, як /bin, /sbin, /usr/bin та /usr/sbin з нього.

Цей файл читається як один з перших файлів конфігурації кожною оболонкою кожного користувача, це не скрипт оболонки.

Середовище має вигляд рядка, що містить пари виду «ключ-значення». Ключ – це ім’я змінної. Кілька значень, як правило, поділяються символом двокрапки (:). Кожна пара, в цілому, виглядає таким чином:

КЛЮЧ = значення1: значення2: ...

Якщо ж значення містить пробіли, потрібно використовувати подвійні лапки:

КЛЮЧ = "значення з пробілами".

В даному випадку під ключем маються на увазі змінні одного з двох існуючих видів: змінні середовища або змінні оболонки.

Змінні середовища - це змінні, які були визначені для поточної оболонки і успадковуються усіма дочірніми оболонками або процесами. Змінні середовища використовуються для передачі інформації процесам, які запущені з оболонки.

Змінні оболонки - це змінні, які містяться виключно в оболонці, в якій вони були встановлені або визначені. Вони часто використовуються для відстеження поточних даних (наприклад, поточного робочого каталогу). Зазвичай такі змінні позначаються за допомогою великих літер. Це допомагає користувачам розрізняти змінні середовища в інших контекстах.

У сценаріях зазвичай використовуються оголошення виду:

ім'я\_змінної="значення змінної", але конкретний синтаксис залежить від інтерпретатора, який використовується.

Для отримання значення змінної необхідно перед її ім'ям поставити символ долара. Також іноді потрібно її ім'я взяти у дужки (наприклад, в сценаріях утиліти make).

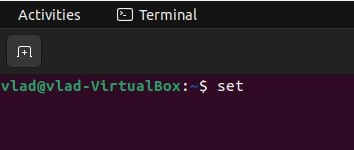
**Завдання**

1. Ознайомитися з теоретичними матеріалом по лабораторній роботі.
2. Опанувати команди по роботі зі змінними оболонки та середовища.
3. Підготувати звіт для викладача про виконання лабораторної роботи і представити його.

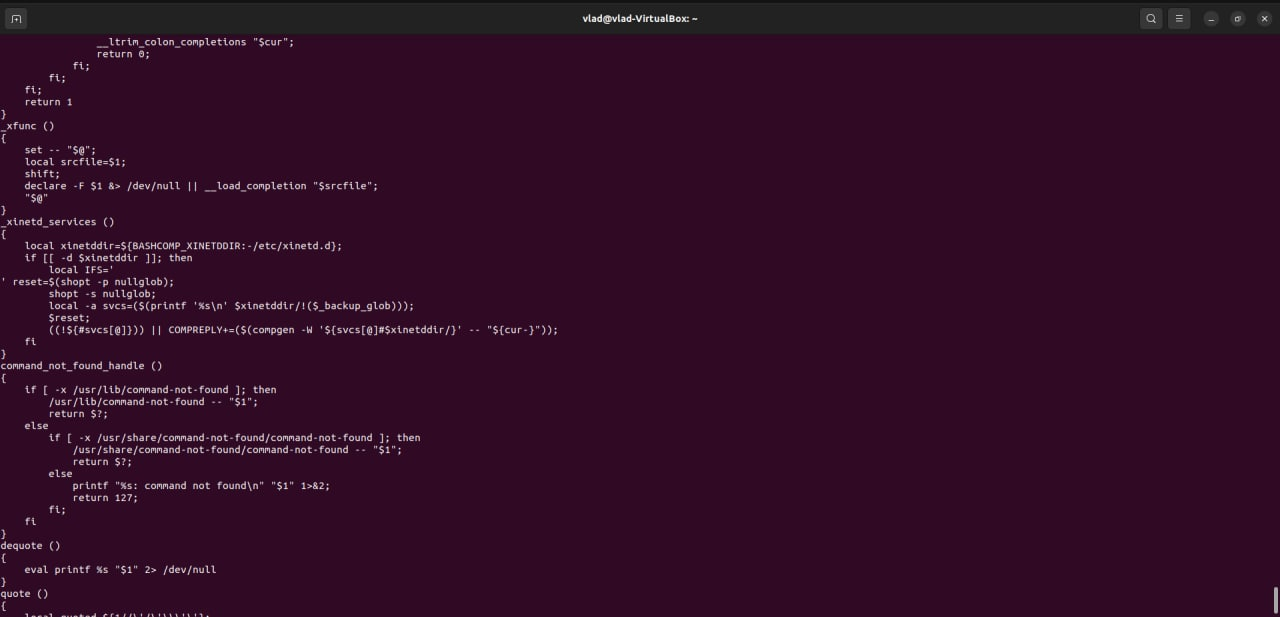
**Результат виконання роботи**

**Вивести змінні оболонки**

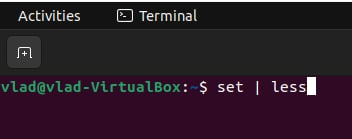
Для того аби вивести змінні оболонки, потрібно ввести команду set.

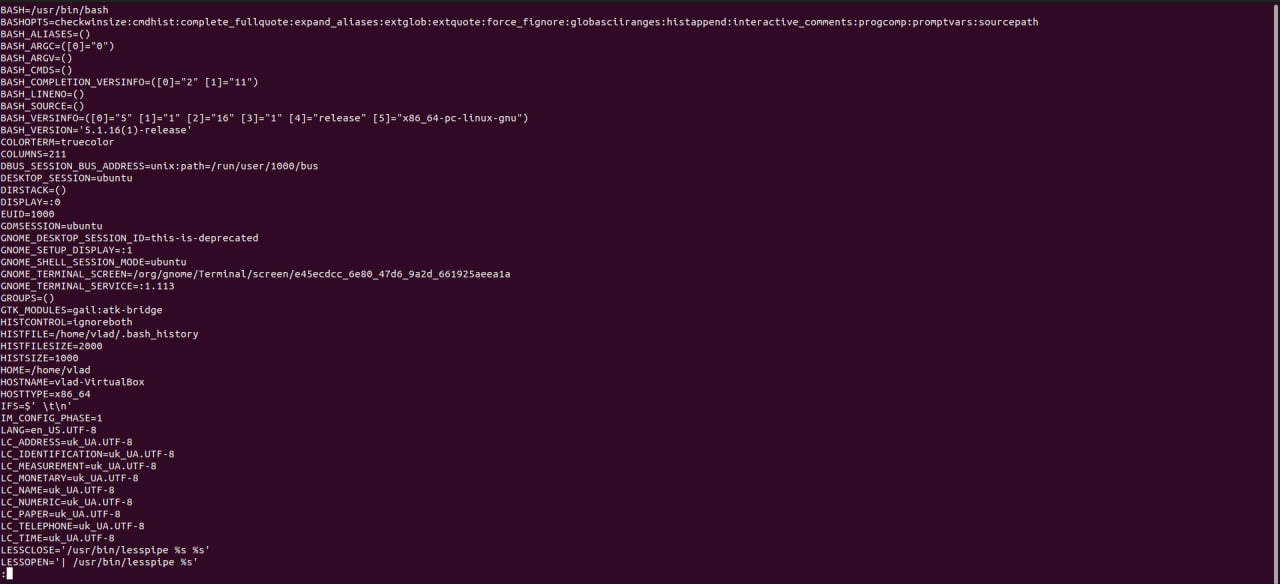


Але вона виводить список всіх змінних оболонки, змінних середовища, локальних змінних і функцій оболонки:



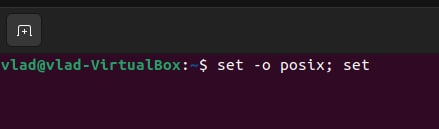
Для виведення списку у більш зручному форматі, вводимо команду set | less, яка дозволяє переглядати інформацію за допомогою програми-пейджера

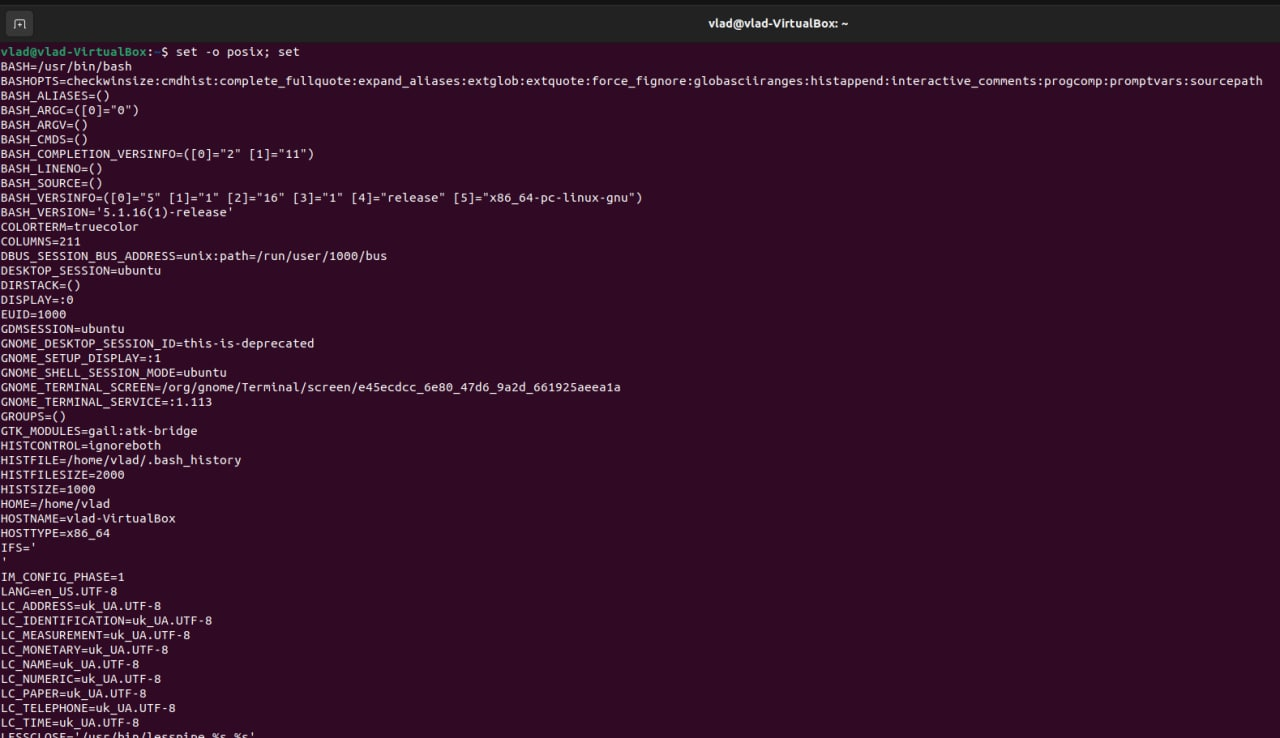




Список містить величезну кількість додаткової інформації, яка в даний момент не потрібна (наприклад, деякі функції bash). Щоб у більш зручному форматі вивести результат, вводимо команду set -o posix; set

Команда виведе усі змінні середовища і оболонки.





**Список основних змінних оболонки:**

**BASHOPTS**: список опцій, які використовуються при виконанні bash.



**BASH\_VERSION**: запущена версія bash в зрозумілій формі.



**BASH\_VERSINFO**: версія bash в машиночитаємому форматі.



**COLUMNS**: визначає ширину виведення у стовбцях.



**DIRSTACK**: стек каталогів, доступних командам pushd і popd.



**HISTFILESIZE**: максимальна кількість рядків, що міститься у файлі історії команд.



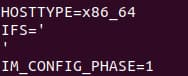
**HISTSIZE**: Кількість команд, які необхідно запам'ятовувати в списку історії.



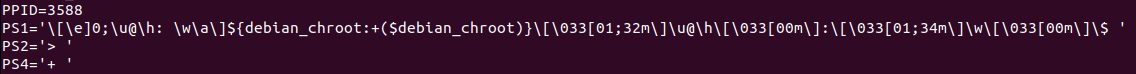
**HOSTNAME**: поточне ім'я хоста.



**IFS**: Внутрішній розділювач полів введення у командному рядку.



**PS1**: визначає рядок первинного запрошення - вид командного рядка при запуску сесії оболонки. Змінна **PS2** встановлює рядок вторинного запрошення, якщо команда займає декілька рядків.



**SHELLOPTS**: параметри оболонки, які можна встановити за допомогою set.

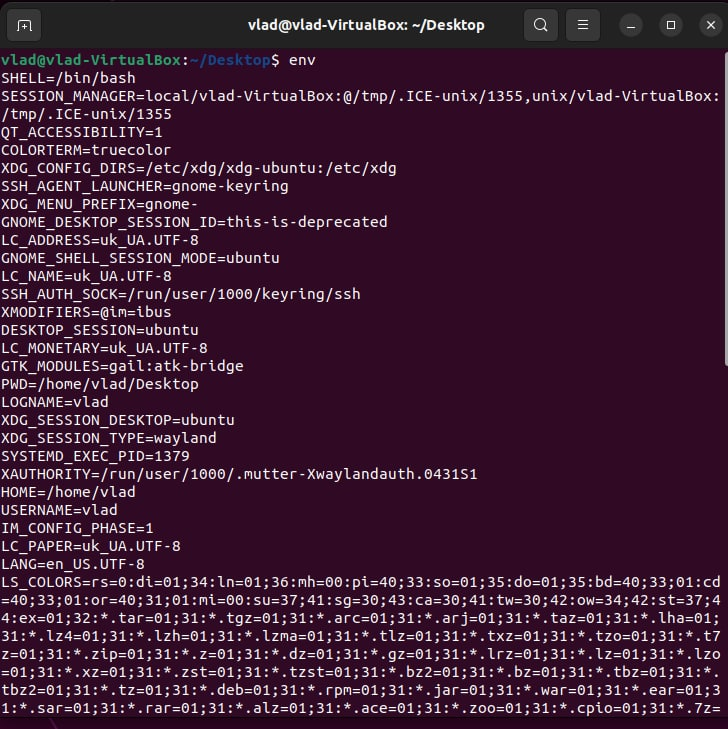


**UID**: унікальний ідентифікатор поточного користувача.



**Вивести змінні середовища.**

Для виведення списка всіх змінних середовища, використовуються команди env або printenv. Вони виводять одну і ту саму інформацію.





Ці команди відрізняються лише кількома індивідуальними функціями. Наприклад, printenv може запитувати значення окремих змінних а команда env дозволяє змінювати середовище, в якій запущені програми, передаючи набір значень змінних в команду.

**Список основних змінних середовища:**

**SHELL** - описує оболонку, яка є інтерпретаторам введених команд. У більшості випадків за замовчуванням встановлена оболонка bash, але це значення можна змінити в разі потреби.



**TERM** - вказує вид терміналу, який емулюється при запуску оболонки. Залежно від операційних вимог можна емулювати різні апаратні термінали.



**LOGNAME** - реєстраційне ім’я користувача.



**USER** - поточний користувач.



**PWD** - поточний робочий каталог.



**LS\_COLORS** - визначає кольорові коди, які використовуються для кольорового виведення результату команди ls. Таке виведення допомагає користувачеві швидше прочитати результат команди (наприклад, швидко розрізнити типи файлів).



**PATH** - список каталогів, до яких звертається система при виконанні команд. Коли користувач запускає команду, система перевіряє ці каталоги у зазначеному порядку в пошуках виконуваного файлу.



**LANG** - поточні налаштування мови та локалізації, в тому числі кодування символів.



**HOME** - ім’я домашнього каталогу поточного користувача.



**\_**: остання виконана команда

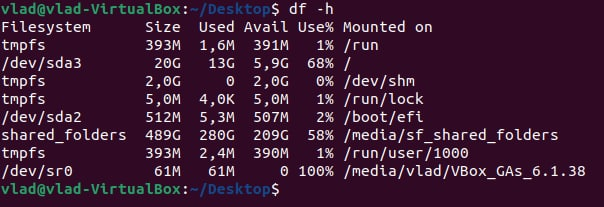


**Вивести поточну дату, обсяг вільного простору на диску, обсяг оперативної пам’яті.**

Для того, аби вивести поточну дату, потрібно у консоль ввести команду date.



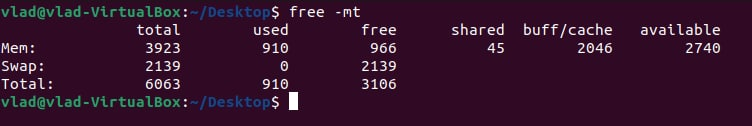
Для того, аби вивести обсяг вільного простору на диску, потрібно ввести у консоль команду df -h. – h показує об'єм диска в кіло-і мегабайтах, а не байтах.



Для того, аби вивести обсяг вільного простору на диску, потрібно ввести у консоль команду free. Також існують такі ключі як:

* -m: показує обсяг в мегабайтах, а не кілобайтах;
* -g: показує обсяг в гігабайтах;
* -t: відображає підсумковий рядок.

Тому введемо команду free -mt аби вивести обсяг у мегабайтах, та вивести підсумковий рядок



**За допомогою команди history виведіть команди, які ви використовували.**

За допомогою команди history виведемо команди, які використовувалися у лабораторній роботі.



**Описати можливості Z shell оболонки.**

**Z shell, zsh** є однією з сучасних командних оболонок UNIX, що використовується безпосередньо як інтерактивна оболонка, або як скриптовий інтерпретатор. Zsh є розширеним *bourne shell* із великою кількістю покращень.

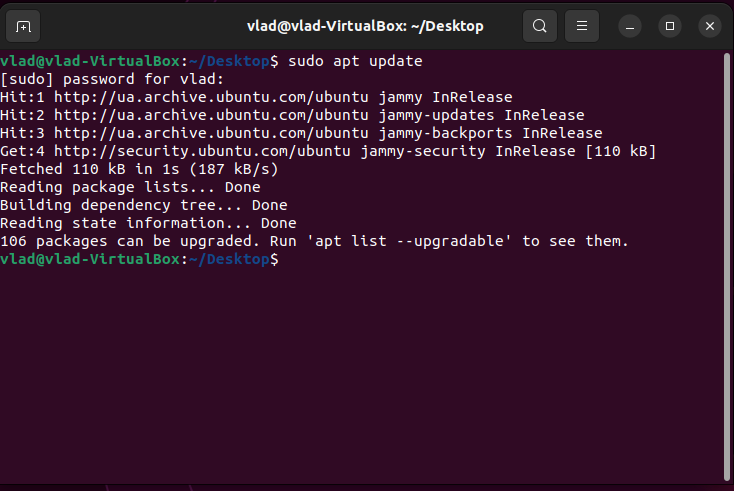
Всі команді оболонки, встановлені в операційних системах Linux, прописані у файлі /etc/shells.

Деякі корисні особливості:

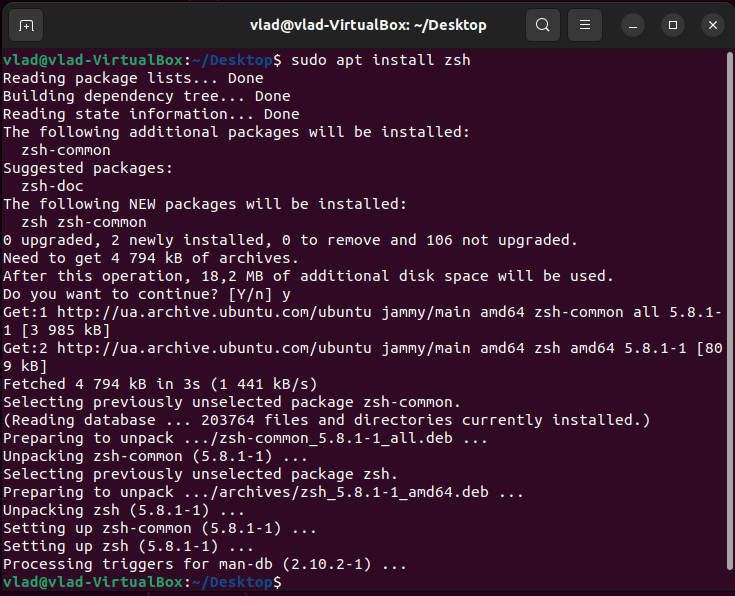
* програмоване автодоповнення, яке допомагає користувачам вводити як команди, так і їх аргументи з вбудованою підтримкою кількох сотень команд;
* поділяє історію команд між усіма запущеними екземплярами оболонки;
* розширене доповнення назв файлів, що дозволяє вказати файл без необхідності запускати зовнішні програми, на зразок find;
* розширена підтримка змінних та масивів;
* редагування багаторядкових команд у єдиному буфері;
* корекція друкарських помилок;
* має різні режими сумісності (тобто ви можете використовувати zsh замість bourne shell при запуску, як /bin/sh);
* модифіковані запрошення (prompts), що включають можливість розташувати запрошення праворуч і налаштувати

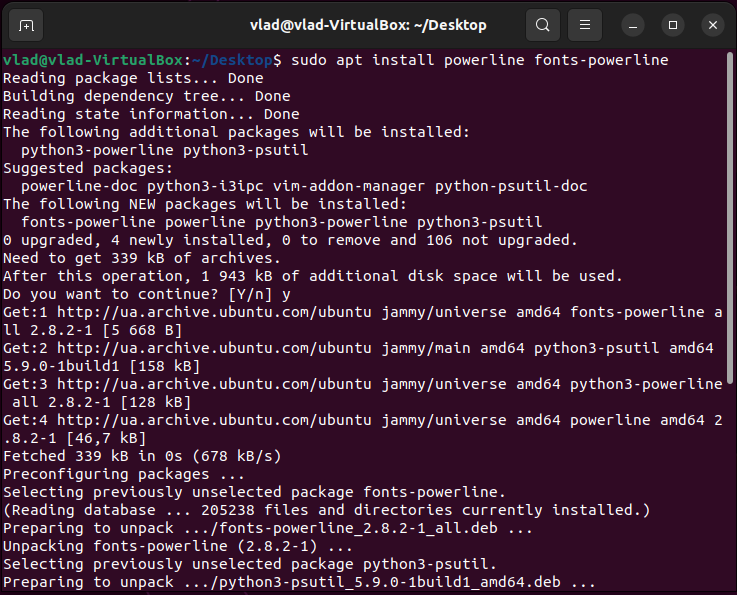
**Встановлення та налаштування Z-shell (Oh My Zsh) в Ubuntu**

Першим ділом оновлюємо пакети за допомогою команди sudo apt update

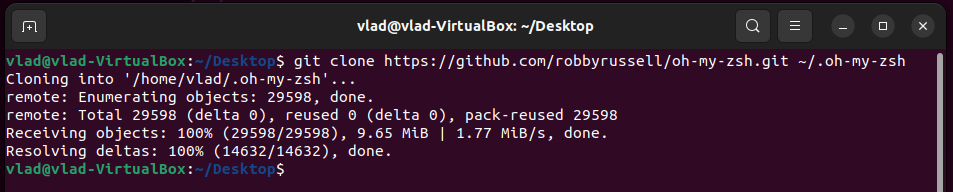


Встановлюємо необхідні пакети (ZSH, powerline та powerline шрифти) за допомогою команд sudo apt install zsh та sudo apt install powerline fonts-powerline

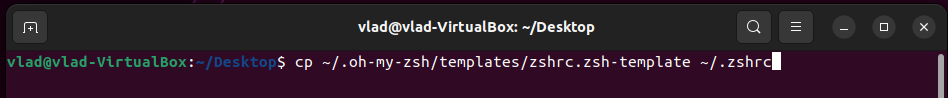




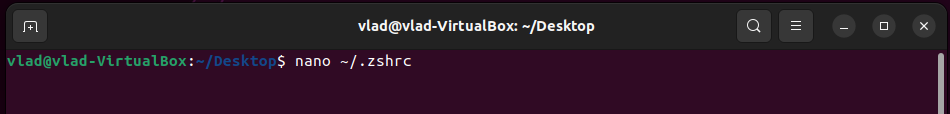
Клонуємо репозиторій Oh My Zsh за допомогою команди git clone https://github.com/robbyrussell/oh-my-zsh.git ~/.oh-my-zsh



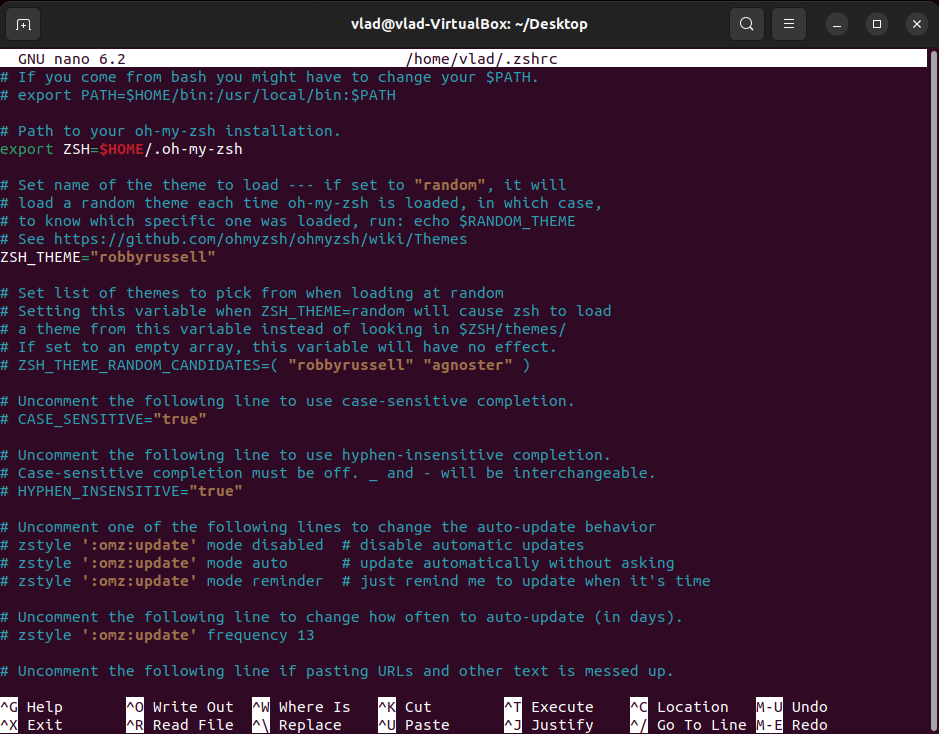
Створюємо новий zsh конфігураційний файл за допомогою команди cp ~/.oh-my-zsh/templates/zshrc.zsh-template ~/.zshrc



За допомогою редактора nano відкриваємо файл .zshrc для того, щоб вибрати тему оформлення для zsh терміналу за допомогою команди nano ~/.zshrc



Після введення цієї команди відкривається ось таке вікно

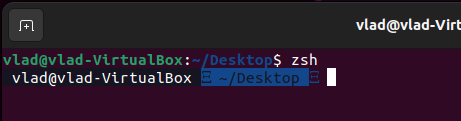


Знаходимо рядок ZSH\_THEME = "robbyrussell" та замінюємо його на тему agnoster.

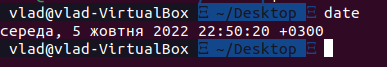


Після чого нажимаємо комбінацію клавіш CTRL + X та ENTER для збереження.

Тепер, ввівши команду **zsh** у консоль можемо запустити командну оболонку. Для того, щоб повернутися в bash вводимо **bash**



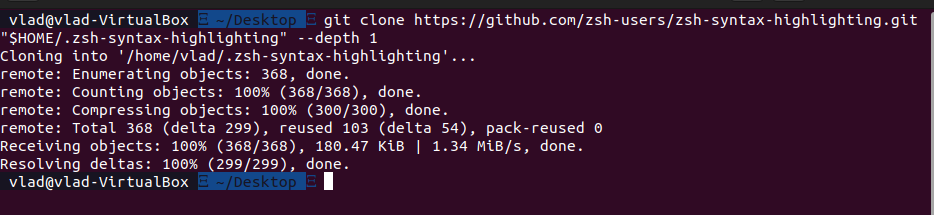
Для перевірки командної оболонки введемо декілька команд, для наглядності



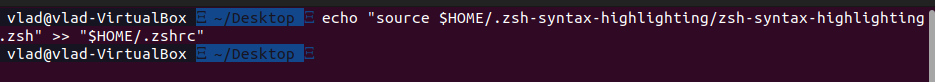
**Налаштування оболонки Zsh**

**1) Встановимо підсвічування синтаксису Oh My Zsh**

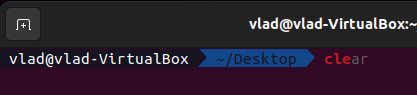
Клонуємо репозиторій ZSH Syntax Highlighting ввівши команду git clone https://github.com/zsh-users/zsh-syntax-highlighting.git "$HOME/.zsh-syntax-highlighting" --depth 1



Додаємо syntax-highlighting у файл конфігурації .zshrc ввівши команду echo "source $HOME/.zsh-syntax-highlighting/zsh-syntax-highlighting.zsh" >> "$HOME/.zshrc"

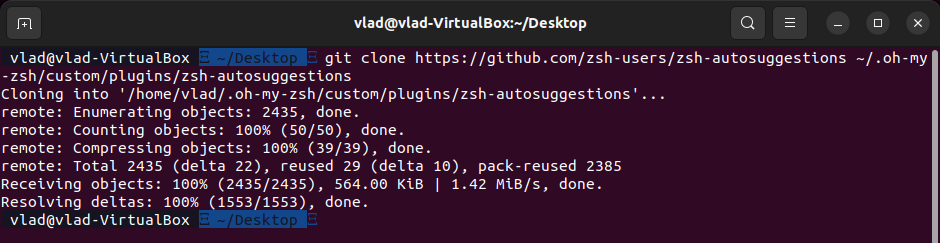


Після встановлення підсвічування консоль виглядає так:

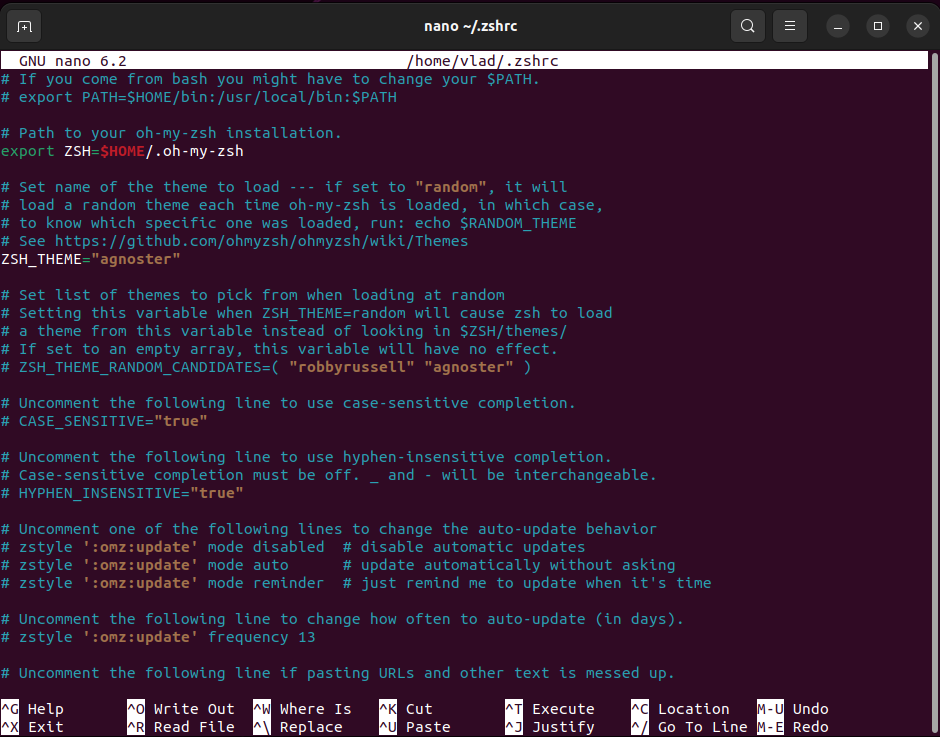


**2) Встановимо плагін автодоповнення для zsh**

Клонуємо репозиторій плагіна zsh-autosuggestions ввівши команду git clone https://github.com/zsh-users/zsh-autosuggestions ~/.oh-my-zsh/custom/plugins/zsh-autosuggestions



Відкриваємо~/.zshrc



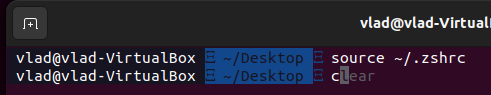
Шукаємо рядок: plugins=(git)і надаємо їй наступного вигляду: plugins=(git zsh-autosuggestions)



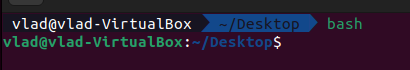
Зберігаємо зміни ввівши комбінацію клавіш CTRL + X та ENTER для збереження. Після збереження оновлюємо конфіг ввівши команду source ~/.zshrc



Після налаштування автодоповнення, починаючи вводити команду, висвічується пропонована команда



Після налаштування оболонки zsh повернемося до оболонки bash ввівши у консоль bash



**Висновок:** На цій лабораторній роботі набуто навички налаштування програмного середовища Linux. Ознайомлено з теоретичним матеріалом по лабораторній роботі. Опанувано команди по роботі зі змінними оболонки та середовища. Вивести змінні оболонки та середовища та описано основні з них. Виведено поточну дату, обсяг вільного простору на диску, обсяг оперативної пам’яті та команди які використовувалися. Описано можливості Z shell оболонки та встановлено її.