

Операційна система Linux

Лекції: Головня Олена Сергіївна

Практичні заняття: Лайчук Артем Андрійович



Теми лекцій

- Лекція 1. Знайомство з Linux

Linux mindset Архітектура Файлова система GUI та CLI Навігація Робота з файлами
Робота у Bash Віртуалізація

- Лекція 2. Процеси у Linux

Процес Ієрархія процесів Стани процесів Виведення відомостей про процеси
Завершення процесів Фонові процеси Встановлення ПЗ

- Лекція 3. Безпека Linux

Безпека ОС Стандартні користувачі та групи Linux Permissions SUID, GUID
Sticky bit

Основний фокус модуля:

інтенсив, орієнтований на практичну роботу з Linux та використання Linux в етичному хакінгу



Олена ГОЛОВНЯ

доцент кафедри КІ та КБ

- Працюю зі студентами 15+ років, 6+ років у Політехніці
- Викладаю дисципліни: “Операційні системи”, “Адміністрування комп’ютерних систем та мереж”, “Технології адміністрування та захисту інформаційних систем” та ін.
- Інструктор курсу NDG Linux Essentials у Мережній академії Cisco
- Користуюся Linux з 2006 року, на регулярній основі 10+ років (з перервами на Windows та macOS)

Контакти: kik_gos@ztu.edu.ua



Трохи про стереотипи



Лекція 1

Знайомство з Linux



Що таке операційна система

- Операційна система (ОС) належить до системного ПЗ
- ОС допомагає користувачу і прикладним програмам взаємодіяти з «залізом»
- ОС керує системними ресурсами комп'ютера (CPU, Mem, I/O, Storage)
- ОС дуже різноманітні (є у серверах, ПК, ноутбуках, телефонах, IoT тощо)
- Ядро ОС (OS kernel) містить найважливіші компоненти



Коротка історія ОС Linux



Фото - Kimmo Mäntylä/REX/Shutterstock.com

- Linux — **Unix-подібна ОС** (схожа на Unix, але не Unix)
- **Лінус Торвальдс** / Linus Torvalds — творець Linux (нині координує розробку ядра Linux)
- Першу Linux Торвальдс розробив у **1991 році** під час навчання в University of Helsinki
- **Linux** (Торвальдс) + **GNU** (Free Software Foundation, Richard Stallman) = **ядро** + **спільнота вмотивованих фахівців**
- Ядро Linux має **відкритий програмний код**
- Існує дуже **багато дистрибутивів** Linux



Дистрибутиви Linux

Приклади дистрибутивів Linux

- RedHat-сумісні (Red Hat Enterprise Linux - RHEL; Fedora; CentOS; Stream; AlmaLinux)
- Debian-сумісні (Debian, Ubuntu, Mint, Kali Linux)
- Slackware, SUSE, openSUSE
- Arch
- Gentoo
- Mageia (раніше - Mandrake)
- Elementary OS
- Rasberian (для Raspberry Pie) та багато інших

Візуалізація походження дистрибутивів Linux:

https://upload.wikimedia.org/wikipedia/commons/1/1b/Linux_Distribution_Timeline.svg

Більше про різні дистрибутиви Linux: <https://distrowatch.com/>



Дистрибутив Kali

- Kali Linux — дистрибутив Linux для тестування на проникнення
- Розроблений на основі Debian Linux. Розробник — OffSec (Offensive Security)
- Теза: будь-яка Linux непогано підходить для етичного хакінгу, але потребує встановлення відповідних інструментів та їхнього налаштування

★ «Фішки» Kali Linux

- Інструменти для тестування на проникнення вже встановлено та сконфігуровано
- Доступна для різних пристроїв та платформ (фізичні та віртуальні машини, віртуальні контейнери, WSL, хмари, мобільні пристрої, Intel/AMD, ARM тощо)
- Iso-образи з широкою кастомізацією
- Розвинена документація

Офіційний сайт: <https://www.kali.org>



Філософія Linux

- **Усе є файлом (Everything is a file)**

Файлами є каталоги, пристрої, потоки даних тощо.

- **Легкі невеликі програми (Keep it simple, stupid)**

Програма має робити бути сфокусована на своїй задачі і виконувати її добре.

- **Мовчання — золото (Silence is golden)**

Команди не повинні виводити дані, про які її не запитували.

- **Поєднуваність інструментів (Make every program a filter)**

Вивід однієї команди можна одразу скерувати як дані для наступної команди.

- **Конфігурація через файл (Text-based configuration)**

Налаштування зберігаються у відкритих форматах і мають легко читатися.

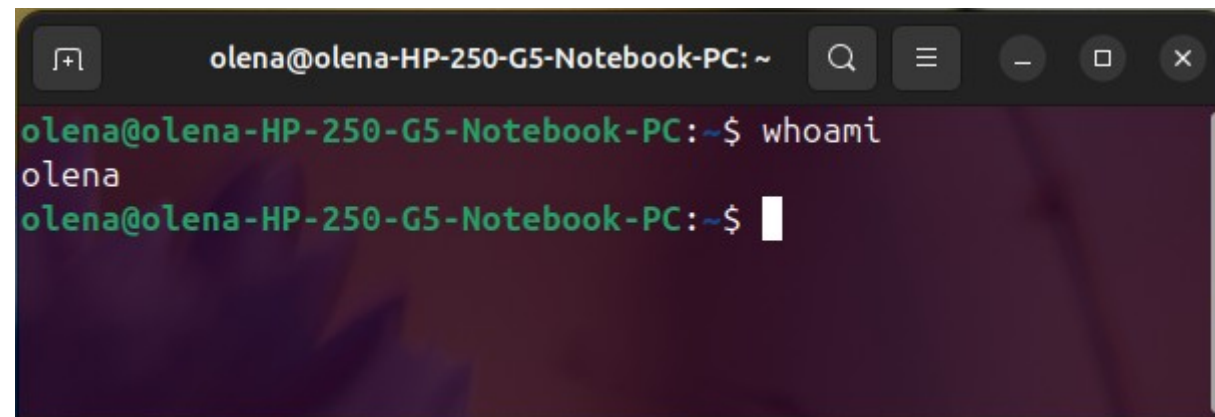
- **Автоматизація (Automation)**

Повторювані завдання автоматизовуються через сценарії командного рядка.



GUI та CLI у Linux

- У Linux доступний графічний інтерфейс (GUI, Graphical User Interface) і / або CLI (Command Line Interface)
- Для CLI є багато командних оболонок (shell), дуже схожих між собою. Найпопулярніша — **Bash**
- GUI також є різноманітні (GNOME, KDE Plasma, Xfce, LXQt, Deepin, Cinnamon, Budgie, MATE та інші)
- Якщо використовується GUI, командний рядок можна використовувати через програму *Термінал*



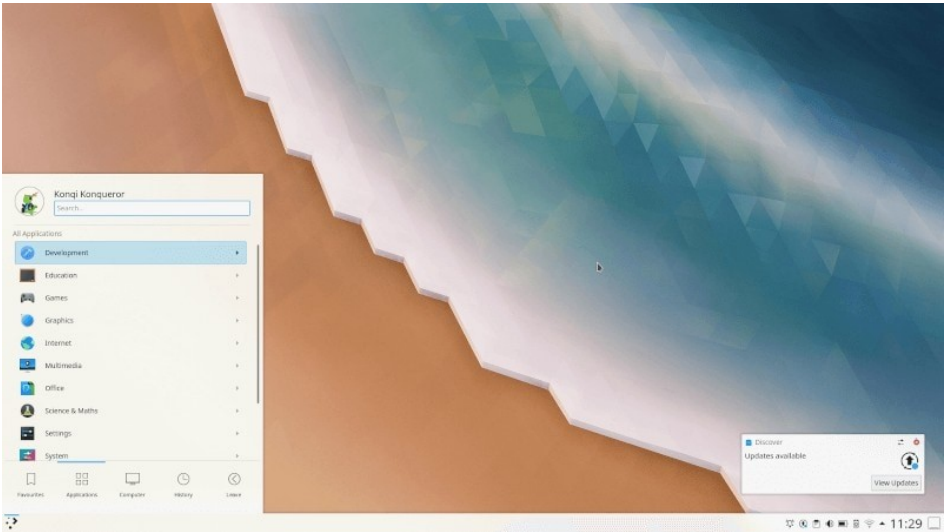
```
olena@olena-HP-250-G5-Notebook-PC: ~  
olena@olena-HP-250-G5-Notebook-PC:~$ whoami  
olena  
olena@olena-HP-250-G5-Notebook-PC:~$
```



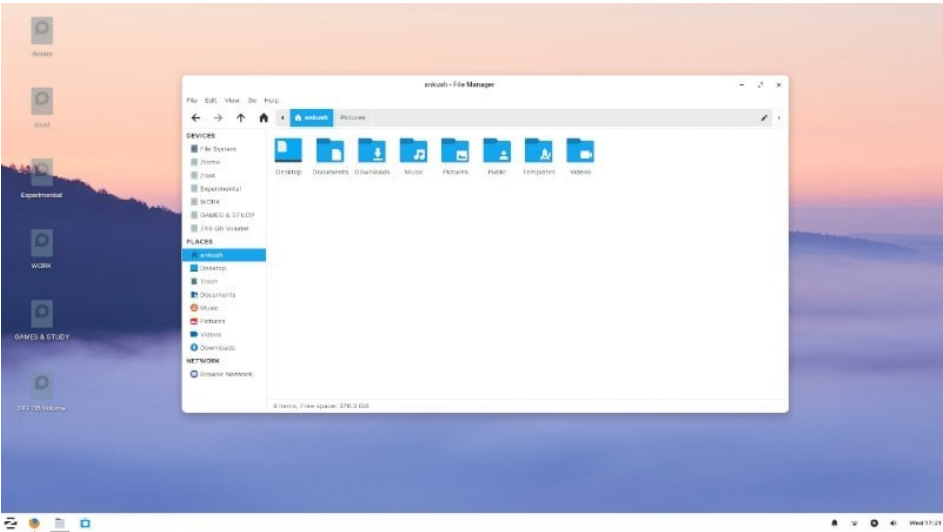
Приклади GUI для Linux



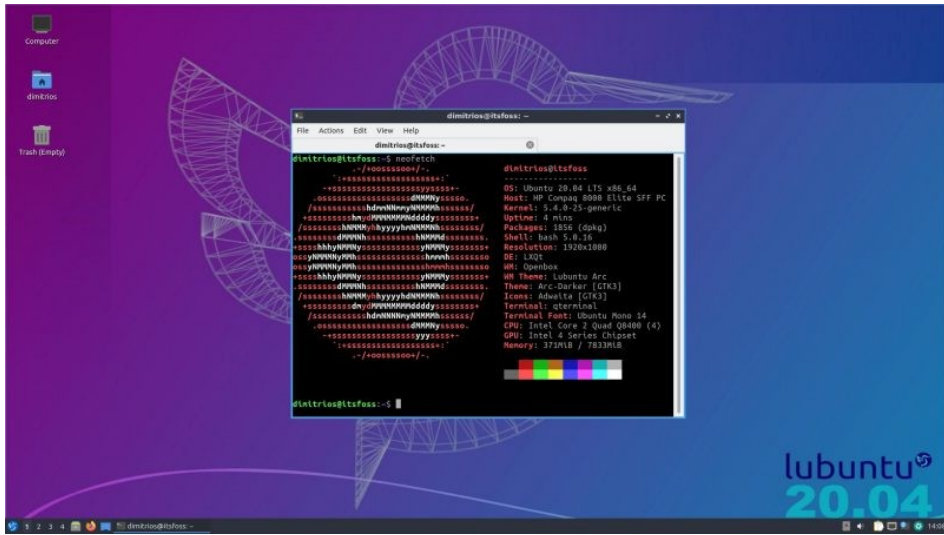
GUI GNOME



GUI KDE Plasma

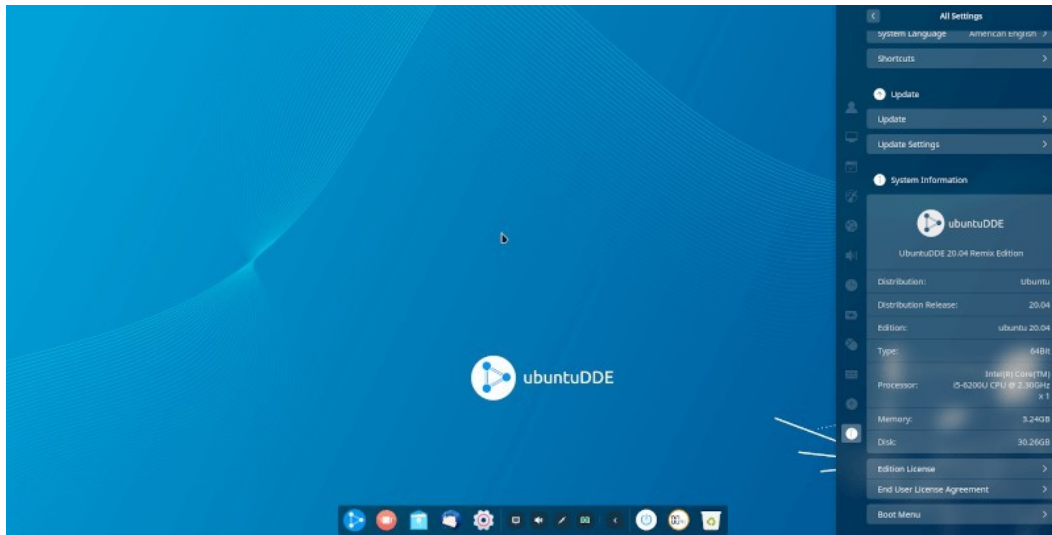


GUI Xfce

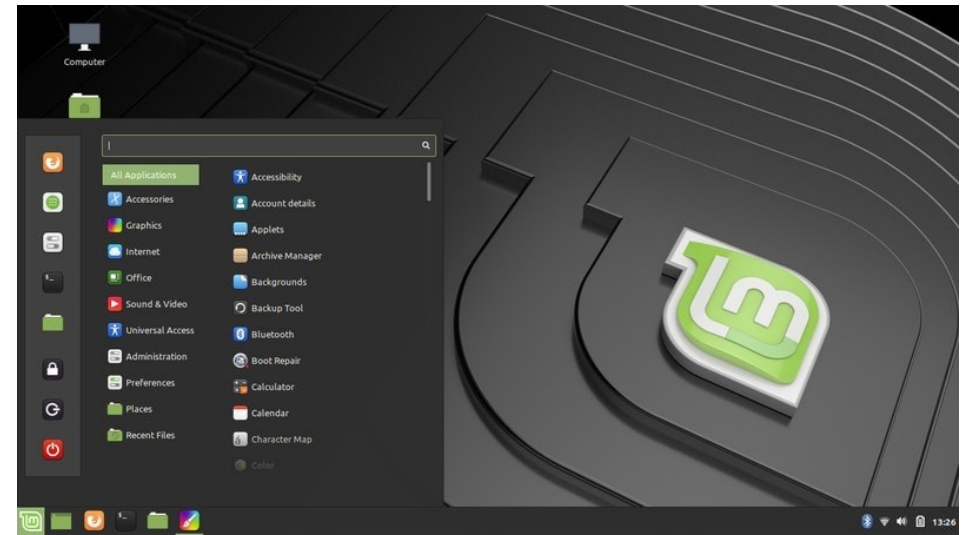


GUI LXQt

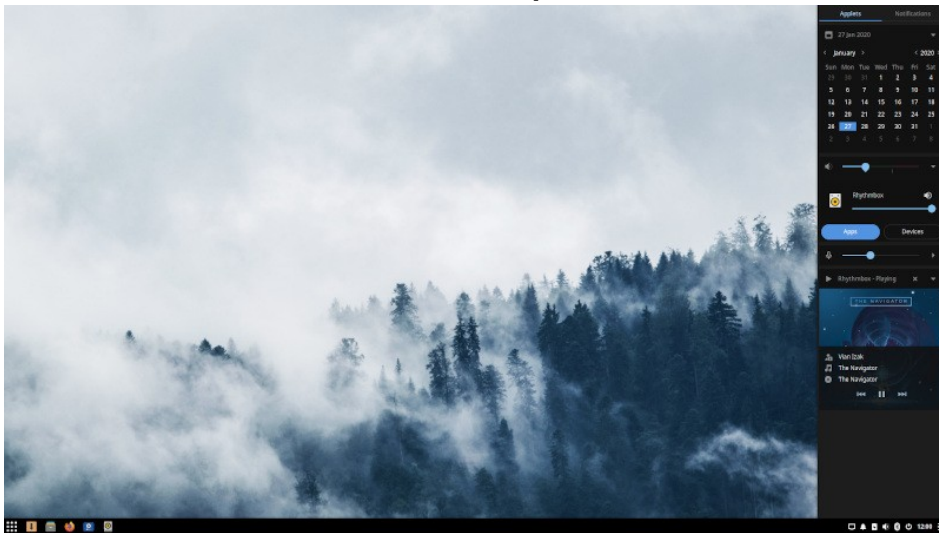
Приклади GUI для Linux



GUI Deepin



GUI Cinnamon (форк GNOME)



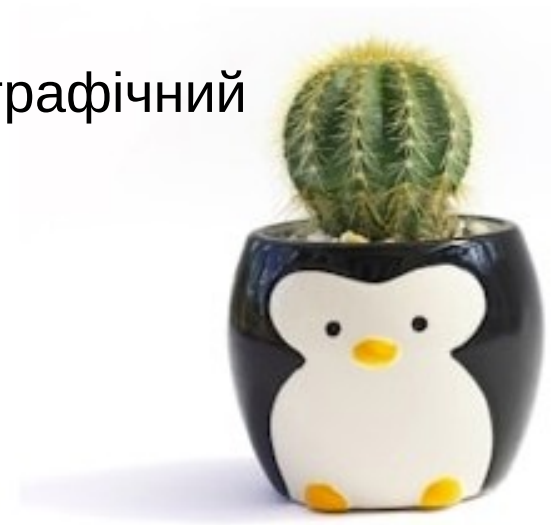
GUI Budgie



GUI MATE (форк GNOME)

Основні компоненти Linux

- **Завантажувач (Bootloader)** — завантажує ОС (найпопулярніший — GRUB)
- **Ядро (Kernel)** — містить найважливіші компоненти ОС
- **Демони (Daemons)** — працюють фоново, забезпечують ключові функції роботи ОС
- **Командна оболонка (Shell)** — командний інтерпретатор, реалізує взаємодію з користувачем незалежно від наявності GUI
- **Графічний сервер (Graphic server)** — забезпечує взаємодію графічних програм з «залізом»
- **Віконна система (Window manager)** — реалізує GUI, спирається на графічний сервер
- **Утиліти (Utilities)** — програми, які реалізують конкретні функції для користувача та інших програм



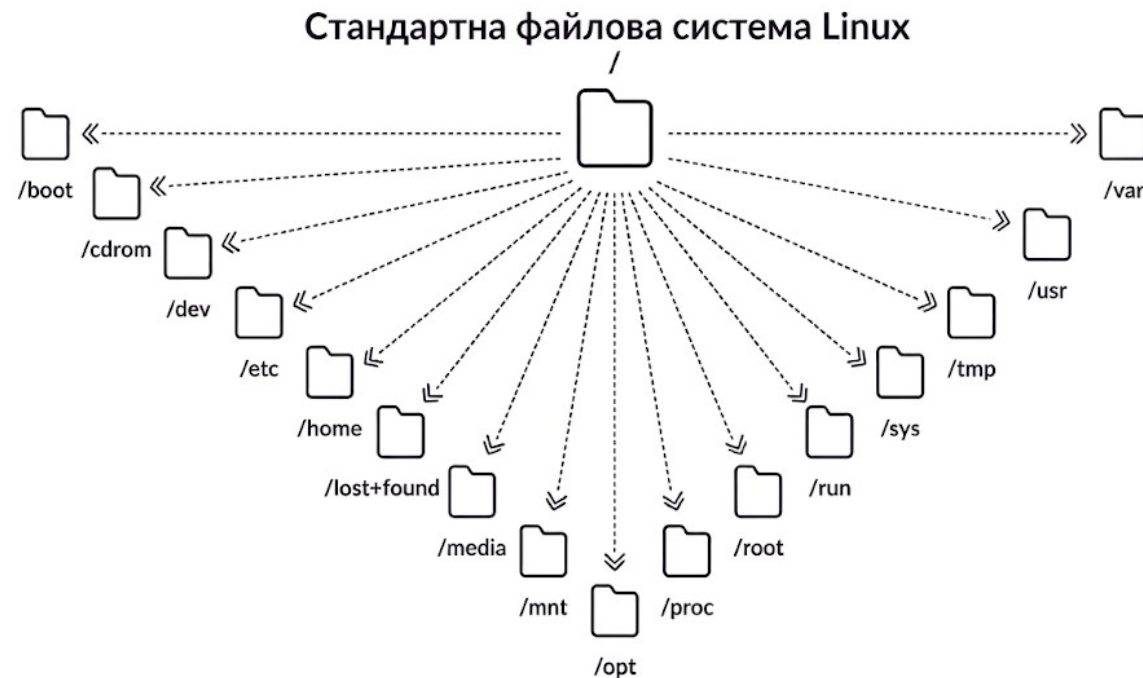
Архітектура Linux

- Апаратне забезпечення (Hardware)
- Ядро (Kernel)
- Командна оболонка (Shell)
- Системні утиліти (System utilities)



Файлова система Linux

- Найпоширеніша файлова система в сучасних Linux — **Ext4**
- Структура файлової системи у Linux може дещо відрізнятися, але **типові риси зберігаються**
- **FHS** — Filesystem Hierarchy Standard — стандарт для файлових систем для Unix-подібних ОС, зокрема Linux



Файлова система Linux

Каталог	Призначення
/bin	Важливі програми, передусім - команди (готові до використання).
/boot	Статичні файли завантажувача, зокрема образ ядра.
/dev	Файли пристроїв.
/etc	Конфігураційні файли (для даного хоста).
/home	Домашні каталоги користувачів.
/lib	Найважливіші спільно використовувані бібліотеки, модулі ядра.
/media	Каталог для монтування змінних носіїв (зазвичай - автоматичного).
/mnt	Каталог для тимчасового монтування файлових систем (зазвичай - ручного).

Файлова система Linux

Каталог	Призначення
/opt	Додаткові програмні пакунки (різне).
/proc	Відомості про наявні у системі процеси, апаратне забезпечення, поточну конфігурацію ядра.
/root	Домашній каталог користувача <i>root</i>
/run	Дані виконуваних процесів.
/sbin	Важливі системні двійкові файли.
/srv	Дані сервісів, які ця система надає іншим.
/tmp	Тимчасові файли.
/usr	Каталог вторинної ієрархії. Програми користувачів, документація тощо. Також містить і важливі системні файли (/usr/bin, /usr/lib, /usr/sbin).
/var	Змінні дані, зокрема <i>log</i> -файли.

Навігація. Довідка

>_ Запрошення (prompt):

```
olena@olena-HP-250-G5-Notebook-PC:~$
```

olena — логін користувача;


@ — розділювач;

olena-HP-250-G5-Notebook-PC — ім'я комп'ютера (hostname);

: — розділювач;

~ — поточний каталог (зараз — **/home/olena**);

\$ — знак непривілейованого користувача (користувач **root** матиме знак **#**)

 Хто ви (лише логін):

```
$ whoami
```

 Хто ви (детально, з групами, до яких входите):

```
$ id
```



Навігація. Довідка

📍 Де ви (поточний каталог):

```
$ pwd
```

📍 Зміна поточного каталогу (переходимо в `/etc`)

```
$ cd /etc
```

📍 Зміна поточного каталогу (переходимо в домашній каталог `~`)

```
$ cd
```

📍 Зміна поточного каталогу (переходимо на рівень вище)

```
$ cd ..
```

💻 ⚙️ Відомості про поточну версію системи:

```
$ uname
```



Навігація. Довідка

📄 Вміст каталогу (поточного):

```
$ ls
```

📄 Вміст каталогу `/etc`

```
$ ls /etc
```

📄 Вміст каталогу з відображенням прихованих елементів (`-a`) та атрибутів (`-l`)

```
$ ls -la
```

🌲 Вміст каталогу ієрархічно (у вигляді дерева, два рівні):

```
$ tree -L 2
```

📄 Вміст файлу `some-file.txt` (лише перегляд):

```
$ cat some-file.txt
```



Навігація. Довідка

? Довідка про команду `ls` у форматі MAN Pages:

```
$ man ls
```

? Довідка про команду `cat` у довідковій системі для вбудованих команд (лише базові команди):

```
$ help cat
```

? Швидка довідка про команду `ls`:

```
$ ls --help
```



Робота з файлами та каталогами

👉 Створити порожній файл **new-file**:

```
$ touch new-file
```

📁 Створити каталог **new-directory** у поточному каталозі

```
$ mkdir new-directory
```

📄 Скопіювати файл **file1** з поточного каталогу у каталог **/home/victor/**

```
$ cp file1 /home/victor
```

📁 Скопіювати каталог **dir1** з поточного каталогу у каталог **/home/victor/**

```
$ cp -R dir1 /home/victor
```

🏷️ Перейменувати каталог **summer-school** на **summer-cyberschool**:

```
$ mv summer-school summer-cyberschool
```

✂️ Перемістити каталог **summer-cyberschool** у каталог **/home/victor/**:

```
$ mv summer-cyberschool /home/victor
```



Робота з файлами та каталогами

🔗 Створити символічне посилання на файл `new-file`:

```
$ ln -s new-file /home/victor/new-file.link
```

✗ Видалити файл `new-file`:

```
$ rm new-file
```

✗ Видалити каталог `new-directory`:

```
$ rm -R new-directory
```



Пошук файлів та каталогів

Пошук виконаного файлу, який відповідає програмі **tree**:

```
$ which tree
```

Швидкий пошук у базі даних:

```
$ locate summer-cyberschool
```

Глибинний пошук файлу **summer-cyberschool** по всій файловій системі (/) :

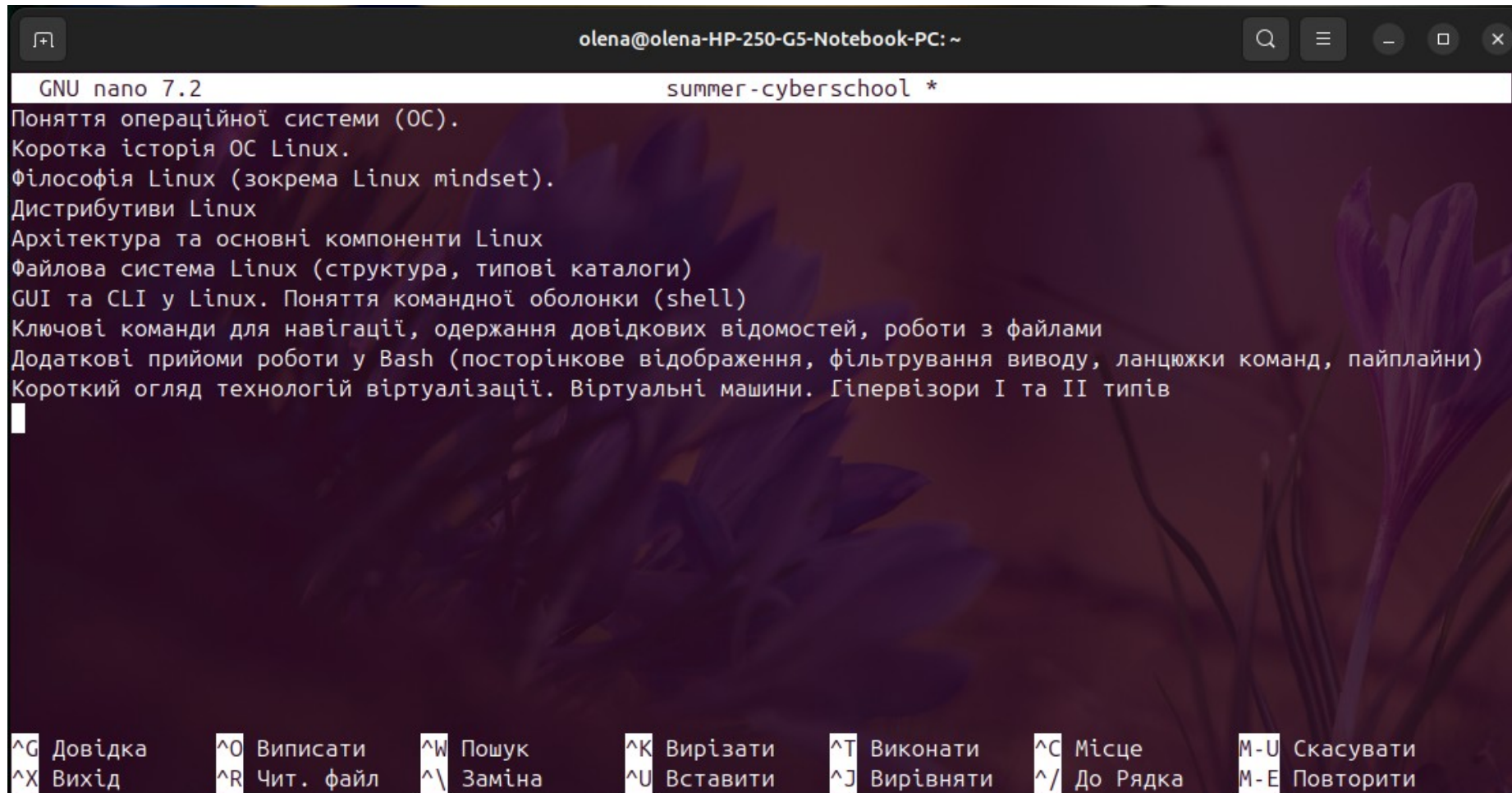
```
$ find / -name summer-cyberschool
```



Редагування файлів у командному рядку

Відкрити файл `summer-cyberschool` текстовим редактором `nano`:

```
$ nano summer-cyberschool
```



The screenshot shows a terminal window with the title bar "olena@olena-HP-250-G5-Notebook-PC: ~". The terminal content is as follows:

```
GNU nano 7.2 summer-cyberschool *
Поняття операційної системи (ОС).
Коротка історія ОС Linux.
Філософія Linux (зокрема Linux mindset).
Дистрибутиви Linux
Архітектура та основні компоненти Linux
Файлова система Linux (структура, типові каталоги)
GUI та CLI у Linux. Поняття командної оболонки (shell)
Ключові команди для навігації, одержання довідкових відомостей, роботи з файлами
Додаткові прийоми роботи у Bash (посторінкове відображення, фільтрування виводу, ланцюжки команд, пайплайни)
Короткий огляд технологій віртуалізації. Віртуальні машини. Гіпервізори I та II типів
|
```

At the bottom of the terminal, there is a legend for nano editor shortcuts:

<code>^G</code> Довідка	<code>^O</code> Виписати	<code>^W</code> Пошук	<code>^K</code> Вирізати	<code>^T</code> Виконати	<code>^C</code> Місце	<code>M-U</code> Скасувати
<code>^X</code> Вихід	<code>^R</code> Чит. файл	<code>^_\</code> Заміна	<code>^U</code> Вставити	<code>^J</code> Вирівняти	<code>^/</code> До Рядка	<code>M-E</code> Повторити



Кілька професійних прийомів з Bash

Посторінкове відображення файлу:

```
$ more big-file
```

або

```
$ less big-file
```

(вихід з режиму посторінкового перегляду — Q)

Пошук рядків зі словом **cyber** у виводі команди **ls**:

```
$ ls | grep cyber
```

Розбиття виводу команди на сторінки:

```
$ ls -R | more
```

або

```
$ ls -R | less
```

(вихід з режиму посторінкового перегляду — Q)

Навіщо?

Бо в чистому CLI немає смуги прокрутки!
(А іноді так просто швидше)

+ **Ще корисні команди**
будуть на практичному
занятті



Віртуалізація. Віртуальні машини

- Технологій віртуалізації багато, і вони дуже різноманітні
- Віртуалізувати можна всю операційну систему, можна — лише середовище для виконання окремої програми, можна — середовище робочого столу тощо
- Віртуальна машина (virtual machine, VM) — це зімітований всередині фізичного комп'ютера віртуальний комп'ютер, на яке встановлено окрему операційну систему
- ОС, встановлена на фізичний комп'ютер — це основна ОС (host OS)
- ОС, встановлена на віртуальну машину — це гостьова ОС (guest OS)
- Для створення VM та керування ними є спеціальне ПЗ — гіпервізори (hypervisors)
- Гіпервізори I типу працюють безпосередньо на фізичному обладнанні (KVM, MS Hyper-V)
- Гіпервізори II типу працюють як програма у межах основної ОС (Oracle VirtualBox, VMware Workstation)

