Лекція 3

Безпека Linux

Ключові поняття безпеки ОС (слайд 1 з 2)

- У безпеці ОС говорять про:
  - **суб’єкти** безпеки (наприклад, користувач `armed-penguin`)
  - **об’єкти** безпеки (або *ресурси* — наприклад, файл `cactus`)
  - **дії**, які суб’єкту дозволено (або не дозволено) виконувати з об’єктом (наприклад, користувачу `armed-penguin` дозволено читати файл `cactus`)
- В ОС зберігаються **правила**, які пов’язують суб’єкт, об’єкт та дію
- Ці правила часто прив’язуються до об’єктів (файл `cactus` має атрибути, де вказано, яким користувачам дозволено його читання, запис та виконання)
- Іноді ці правила прив’язуються до суб’єктів (користувачу `armed-penguin` можна перезавантажувати ОС)

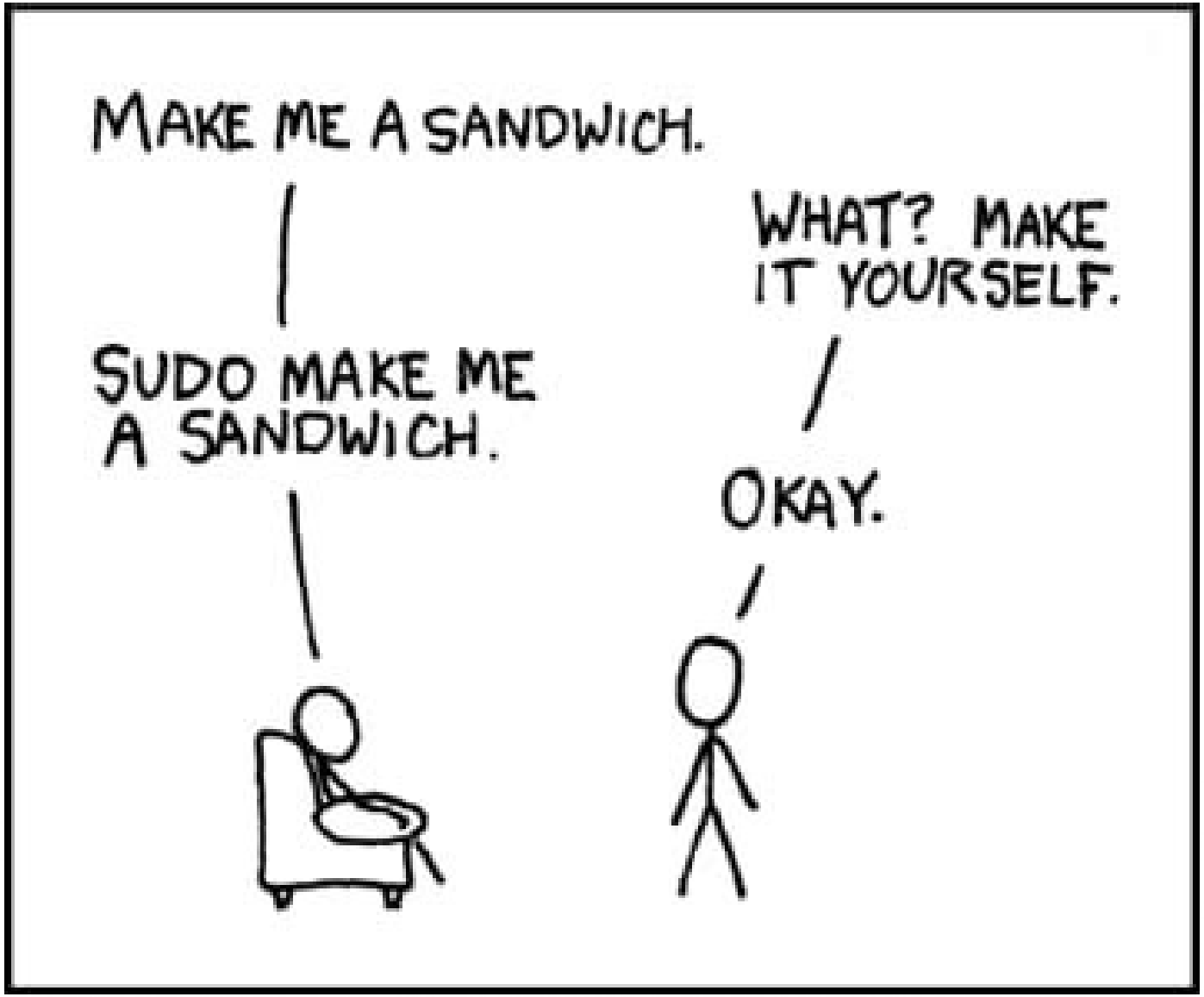
Ключові поняття безпеки ОС (слайд 2 з 2)

- **Автентифікація (Authentication):** кожна дія виконується певним суб'єктом, ідентичність якого встановлено системою
- **Авторизація (Authorization):** система регулює, які дії дозволені тим чи іншим суб'єктам
- **Аудит (Accounting):** система документує події, зокрема пов'язані з безпекою
- **A**uthentication + **A**uthentication + **A**ccounting = AAA

Користувачі та групи у Linux

- Відомості про користувачів Linux зберігаються у файлі `/etc/passwd`
- Відомості про групи Linux зберігаються у файлі `/etc/group`
- Приклади системних облікових записів: `root`, `daemon`, `syslog`
- Приклади системних груп: `sudo`, `daemon`, `users`
- У Debian-сумісних дистрибутивах для нового користувача автоматично створюється однойменна група (користувач `armed-penguin` — група `armed-penguin`)

Користувач root і механізм sudo



Користувач `root` і механізм `sudo`

- Привілейований користувач (адміністратор) в Linux називається **root**
- Але бути "рутом" весь час **небезпечно**: тоді *все* запускається під `root`
- У сучасних Linux для адміністративних задач використовується **sudo**
- Механізм **sudo** виконує від імені адміністратора тільки конкретну дію
- Використовувати команду `sudo` можна членам групи **sudo**
- Обліковка `root` може бути відключена за замовчуванням
- Але користувач `root` все одно є (і володіє важливими системними файлами і каталогами)

Робота з користувачами та групами

Створити користувача `armed-penguin`:

```
$ sudo adduser armed-penguin
```

Створити групу `penguins`:

```
$ sudo groupadd penguins
```

Додати користувача `armed-penguin` до `penguins`:

```
$ sudo usermod -aG penguins armed-penguin
```

Видалити користувача `armed-penguin` разом з домашнім каталогом:

```
$ sudo userdel -r armed-penguin
```

Видалити групу `penguins`:

```
$ sudo groupdel penguins
```

Ролі та дозволи у Linux

Ролі:

- **u** – користувач / користувач-власник (owner / **u**ser owner)
- **g** – група / група-власниця (**g**roup owner)
- **o** – інші / решта користувачів (**o**thers)
- **a** - всі користувачі (**a**ll)

Дозволи:

- **r** - читання (**r**ead)
- **w** - запис (**w**rite)
- **x** - виконання (e**x**ecute)
- **-** - дозвіл відсутній

```
olena@ubuntu:~$ ls -l file1
-rw-rw-r-- 1 olena olena 0 лис 27 04:50 file1
```

рядок повноважень *користувач* *група*

-rw-rw-r--

u g o

Дозволи у Linux: файли *vs* каталоги

Дозвіл	Інтерпретація для файлів	Інтерпретація для каталогів
Читання (***r***)	Переглядати та копіювати вміст	Одержувати перелік файлів (включаючи приховані елементи, але не включаючи детальних відомостей про елементи каталогу*)
Запис (***w***)	Змінювати й зберігати зміни	Додавати та вилучати файли. **! Не працюватиме** без ***x***
Виконання (***x***)	Виконувати (запускати як процес)	“Потрапляти всередину” (виконувати *cd,* використовувати ім'я каталогу у шляхах до підкаталогів). **! Працюватиме** без ***r***

* Щоб переглядати ще й детальні відомості про елементи каталогу, має бути не лише дозвіл *r*, а й дозвіл *x* ← **Важливо!!!**

Перегляд безпекових атрибутів

Розширені відомості про файл **cactus** (зокрема і його безпекові атрибути):

```
$  ls -l cactus
```

Ще більш розширені відомості про файл **cactus**:

```
$  stat cactus
  Файл: cactus
  Розмір: 2598       Блоків: 8        Блок в/в: 4096   звичайний файл
Пристрій: 252,1    Inode: 7744467     Посилання: 1
Доступ: (0664/-rw-rw-r--)  Uid: ( 1000/   olena)   Gid: ( 1000/   olena)
Доступ: 2026-06-11 16:26:39.864559988 +0300
Модиф.: 2026-06-11 16:26:50.592058317 +0300
 Зміна: 2026-06-11 16:26:50.592058317 +0300
Створ.: 2026-06-11 16:26:39.863559942 +0300
```

Методи запису дозволів

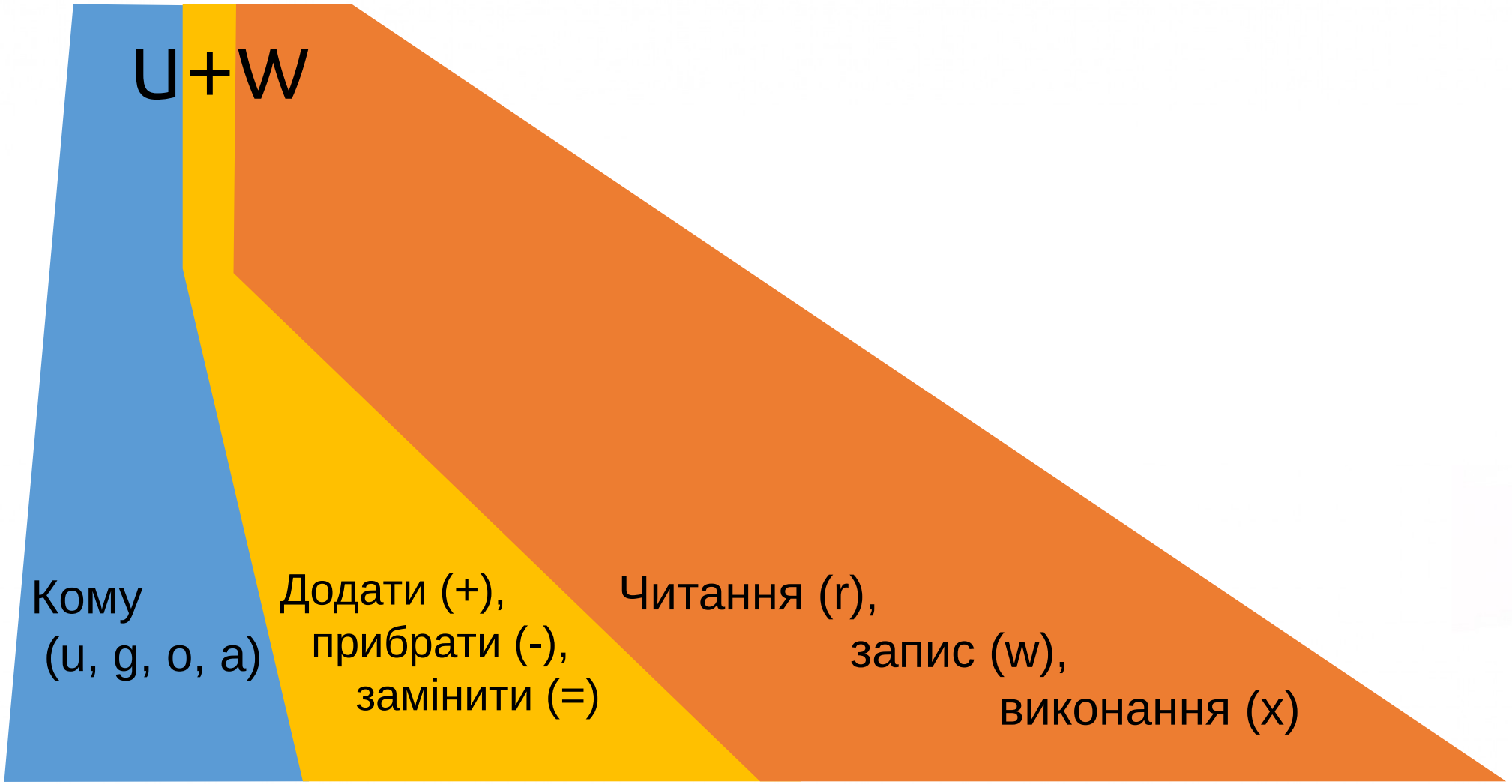
Методи запису дозволів

символьний	числовий
r	4
w	2
x	1

Методи запису дозволів: як це працює

Символьний метод	Двійкове значення	Десяткове значення
rwx	111	4+2+1=7
rw-	110	4+2+0=6
r-x	101	4+0+1=5
r--	100	4+0+0=4
-wx	011	0+2+1=3
-w-	010	0+2+0=2
--x	001	0+0+1=1
---	000	0+0+0=0

Зміна дозволів (символьний метод)



Зміна дозволів (символьний метод)

Дозволити власнику запуск файлу `cactus.sh`:

```
$ chmod u+x cactus.sh
```

(якщо це чужий файл – попереду додаємо `sudo`)

Прибрати дозвіл на перегляд файлу `cactus` для решти користувачів:

```
$ chmod o-r cactus
```

Зміна дозволів (числовий метод)

740

Власнику	Групі	Решті
rwx	**r--**	**---**
4+2+1	**4+0+0**	**0+0+0**

Зміна дозволів (числовий метод)

Дозволити власнику всі дії з файлом `cactus`, а групі та решті – лише читання:

```
$ chmod 744 cactus
```

Дозволити читання файлу `cactus` лише власнику:

```
$ chmod 400 cactus
```

Зміна власників файлу

Зробити користувача `armed-penguin` власником файлу `cactus`:

```
$ sudo chown armed-penguin cactus
```

Зробити користувача `armed-penguin` і групу `penguines` власниками файлу `cactus`:

```
$ sudo chown armed-penguin:penguines cactus
```

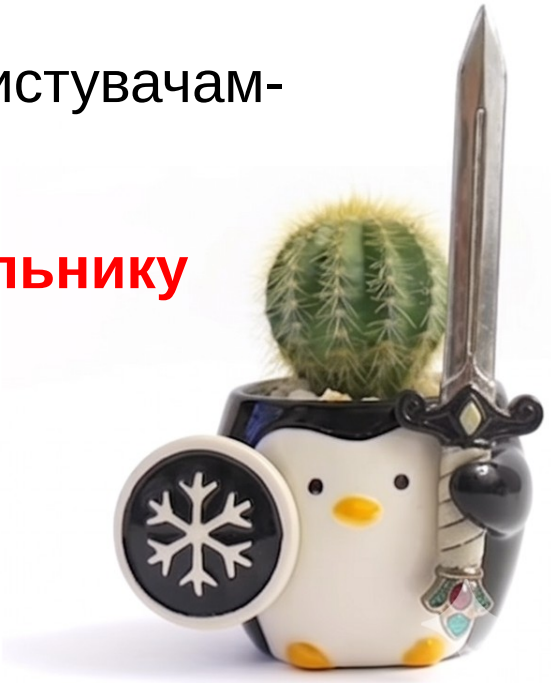
Зробити групу `penguins` власницею файлу `cactus`:

```
$ chgrp armed-penguin cactus
```

(ця команда працюватиме і без `sudo`, якщо змінює власник)

SUID, SGID, Sticky bit

- Крім **`rwx`**, є додаткові біти:
  - **SUID (SetUID, Set User ID)** – дозволяє решті користувачів запускати виконуваний файл від імені користувача-власника
  - **SGID (SetGID, Set Group ID)** – дозволяє решті користувачів запускати виконуваний файл від імені групи-власниці (ще використовується для спільних каталогів)
  - **Sticky bit** – дозволяє видаляти файли з даного каталогу лише користувачам-власникам цих файлів
- Неправильно налаштовані SUID / SGID можуть **дозволити атакувальнику отримати підвищені привілеї (privilege escalation)!**

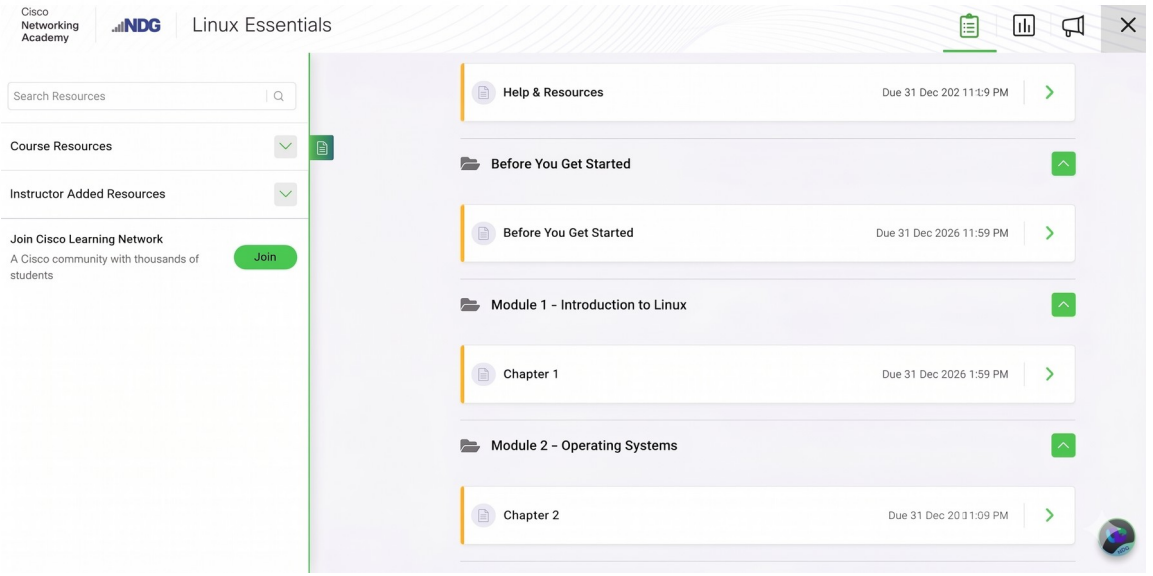


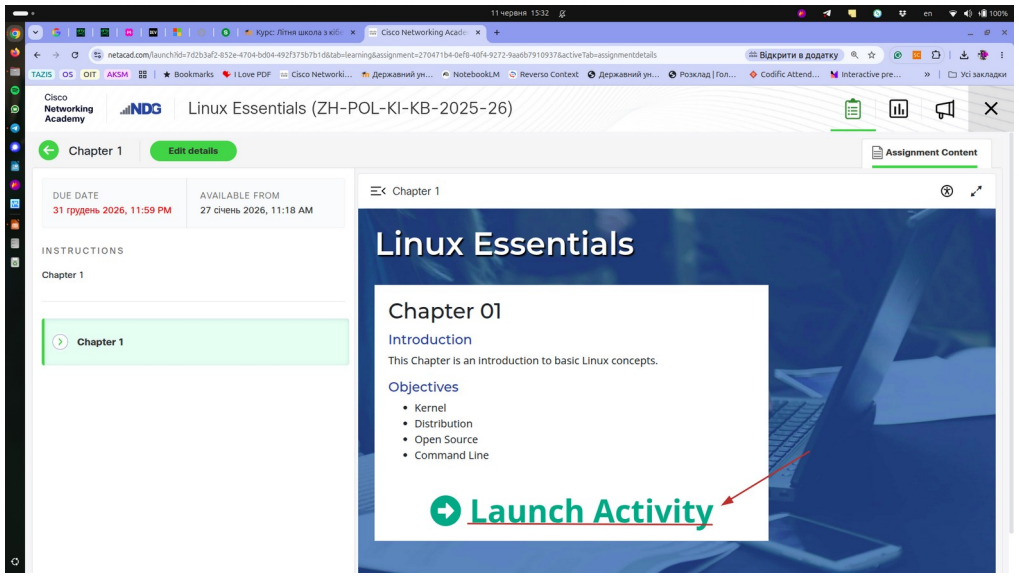
Приклад SUID

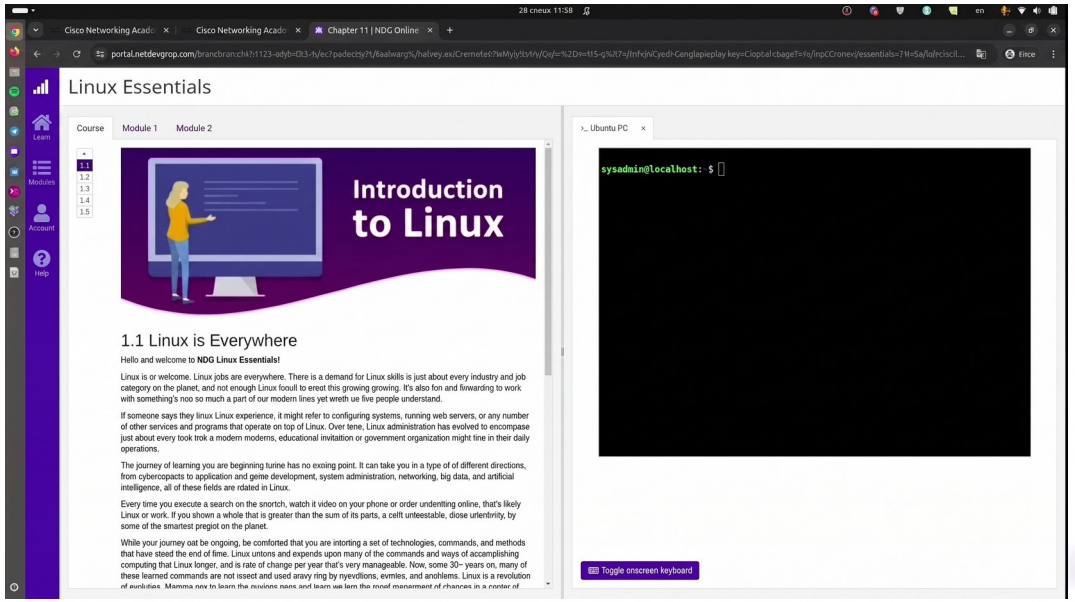
```
$ stat /usr/bin/passwd
  Файл: /usr/bin/passwd
  Розмір: 64152     Блоків: 128       Блок в/в: 4096   звичайний файл
Пристрій: 252,1    Inode: 15207360    Посилання: 1
Доступ: (4755/-rwsr-xr-x)  Uid: (    0/    root)   Gid: (    0/    root)
Доступ: 2026-06-09 17:36:04.985013966 +0300
Модиф.: 2024-05-30 17:52:35.000000000 +0300
 Зміна: 2024-10-11 09:38:53.648811832 +0300
Створ.: 2024-10-11 09:38:51.998290441 +0300
```

(тому звичайний користувач може змінити пароль – лише собі)

Курс NDG Linux Essentials на netacad.com







Проходимо
у липні 2026 року
дистанційно
(вам прийде лист
із запрошенням на курс)