

Лекція 2

Процеси у Linux



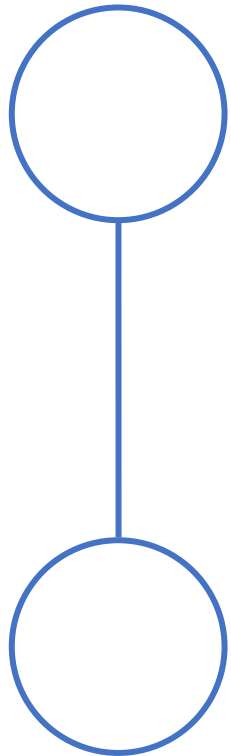
Поняття процесу в операційній системі

- **Процес** — абстракція, що представляє програму під час виконання
- Сучасні процеси мають один або кілька **потоків**
- **Потік** — програмний код, виконуваний на ЦП (зазвичай — код потокової функції)
- Набір адрес, до яких процес має доступ, називається **адресним простором**
- **Адресний простір** процесу містить код та дані процесу
- Якщо процес має декілька потоків, то вони працюють у **спільному адресному просторі** (наприклад, глобальні змінні в них будуть спільні)



Ієрархія процесів у Linux

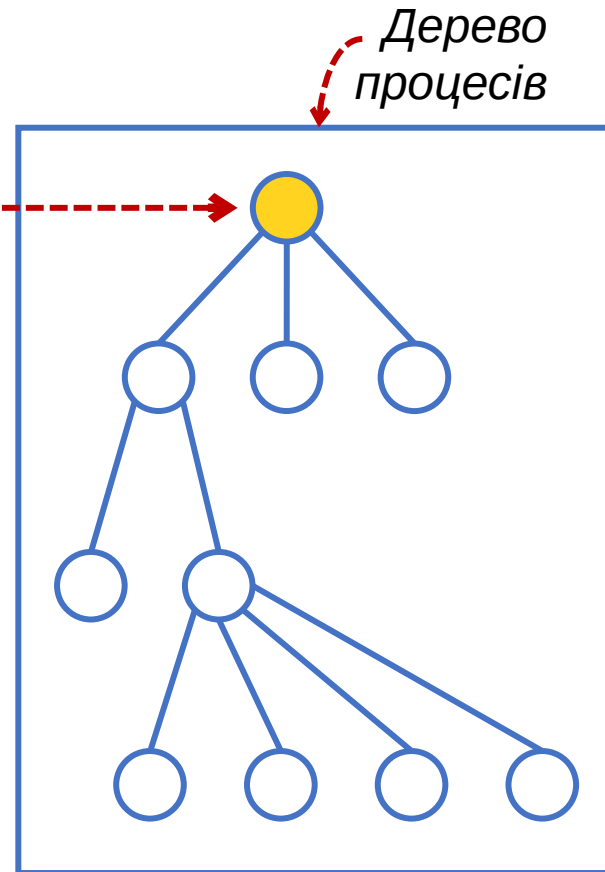
У Linux процеси пов'язані між собою ієрархічно:



Батьківський процес
(*parent process*)

Дочірній процес
(*child process*)

Процес
з PID 1
(зазвичай
init або systemd)



Дерево
процесів

Я теж хочу
бути деревом
процесів
😞





Стани процесів у Linux

- **Стан «Запущено» (Running)** 🏃
 - АБО процес виконується центральним процесором **безпосередньо зараз**,
 - АБО процес **претендує на виконання** і почне виконуватися, щойно його вибере планувальник
- **Стан «Очікування» (Sleeping)** 😴
 - для продовження виконання **має статися певна подія** (наприклад, звільнитися ресурс)
- **Стан «Зупинено» (sTopped)** 🛑
 - процес **зупинено** (але його можна поновити)
- **Стан «Неперервний сон» (uninterruptable sleep, D)** 😓
 - схоже на S, але **не реагує на сигнали** (часто — виконує I/O)
- **Стан «Зомбі» (Zombie)** 🧟
 - завершений, але **структуру даних ще не вилучено**
- **Стан «Мертвий» (Dead, X)**
 - **завершений** (такі процеси не повинні відображатися)



Приклади атрибутів процесів

- **PID** = Process Identifier — унікальний ідентифікатор процесу
- **STAT / S** = State — стан процесу
- **USER** — користувач, від імені якого запущено процес
- **PR** = Priority — пріоритет процесу (у Linux менше число — вищий пріоритет)
- **NI** = Niceness — «рівень люб'язності» (корегує пріоритет  )
- **%CPU** — відсоток використання процесом ЦП
- **%MEM** — відсоток використання процесом оперативної пам'яті
- **COMMAND** — команда, якою було запущено процес



Виведення відомостей про процеси

Відомості про процеси, запущені з даного терміналу:

```
$ ps
```

Відомості про всі процеси:

```
$ ps aux
```

Відомості про всі процеси в динаміці + керування ними:

```
$ top
```

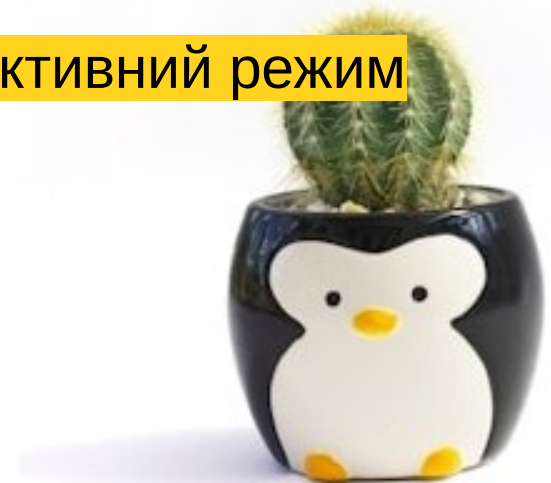
Як **top**, але сучасніше (може не бути в мін. установках Linux):

```
$ htop
```



Фонові процеси

- Фоновий процес (background process) працює без явної взаємодії з користувачем
- У Linux фонові процеси називають демонами (daemons) 🐉
- Імена демонів часто закінчуються на літеру d
- Демони реалізують системні сервіси 🛠️ або сервіси, встановлені користувачем 👤
- Інтерактивні процеси (foreground processes) розраховані на взаємодію з користувачем
- Інтерактивний процес можна запустити у фоні / повернути в інтерактивний режим



Команда `systemctl`

Команда `systemctl` дозволяє керувати фоновими процесами.

Запуск демона `ssh`:

```
$ systemctl start ssh
```

Перевірка стану демона `ssh`:

```
$ systemctl stop ssh
```

Більше — на практичному занятті.



~~Знищення зомбі~~ Завершення процесів

Завершення процесу за його PID (тут PID = 1234)

```
$ kill 1234
```

Завершення процесу за його ім'ям (тут ім'я процесу — **nano**)

```
$ killall nano
```

(якщо є кілька **nano** — завершаться всі)

Обов'язково перевіряємо, чи процес завершився, наприклад:

```
$ ps 1234
```

Якщо процес досі є, завершуємо з сигналом 9 (примусово):

```
$ kill -9 1234
```



Основні технології встановлення ПЗ у Linux

- Дистрибутивозалежні менеджери пакунків (`dpkg`, `rpm`, `pacman`)
- Віртуалізація прикладних програм (Snap, Flatpak, AppImage)
- Віртуальні контейнери (Docker, LXC)
- Інші шляхи (ПЗ в архівах; Github; компіляція з джерельних кодів)



Встановлення ПЗ у Debian-сумісних дистрибутивах

- Debian-сумісні дистрибутиви — дистрибутиви, які походять від Debian Linux
- Приклади: Debian, Ubuntu, Mint, Kali
- Debian-сумісні дистрибутиви використовують:
 - менеджер пакунків (package manager) `dpkg`
 - файли пакунків у форматі `.deb`
 - команди `dpkg`, `apt`, `aptitude`
 - [додатково] інші механізми встановлення (архіви, Git, джерельні коди, Snap, Flatpak, AppImage)



Команди для встановлення ПЗ (Dpkg/Deb)

Встановлення пакунка **steam** з **deb**-файлу **steam_latest.deb**

```
$ sudo dpkg -i steam_latest.deb
```

Нюанси: 1) deb-файл треба попередньо завантажити;
2) dpkg НЕ вміє автоматично враховувати залежності (бракує бібліотеки - помилка)

Виправлення помилки відсутніх бібліотек:

```
$ sudo apt --fix-broken install
```

Встановлення пакунку **tree** з репозиторіїв:

```
$ sudo apt update  
$ sudo apt install tree
```

Деінсталяція пакунка **tree**:

```
$ sudo apt remove tree
```

