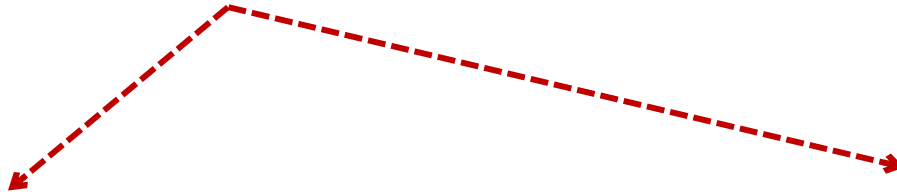


# 3. Блок керування процесом

? Що ОС має знати про процес?



**Де процес розміщений?**

*Щоб знайти його.*

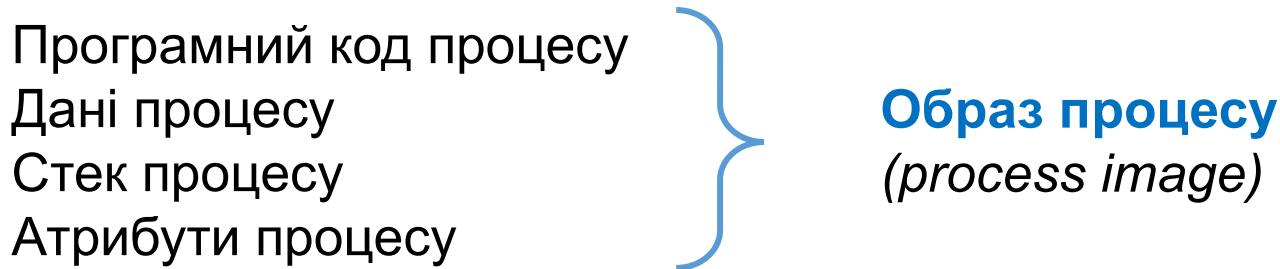
- Здебільшого в оперативній пам'яті...
- ...але частина може бути на диску.
- Можливо, неперервним діапазоном...
- ...або ні.
- Треба знати *адреси*

**Які атрибути має процес?**

*Щоб керувати ним, тобто:*

- ...відрізнити його від інших процесів (ідентифікатор);
- ...знати, що йому можна, а що ні (безпекові атрибути);
- ...пам'ятати все, що стосується виконання процесу на ЦП;
- ...знати, чи цей процес важливіший за інші (пріоритети);
- ...стежити, скільки ресурсів процес споживає;
- ...знати про зв'язки процесу з відкритими файлами, пристроями введення-виведення і т. д.

### 3. Блок керування процесом



*Атрибути - у блоці керування процесом.*

**Блок керування процесом** (*process control block, PCB*) - сукупність атрибутів процесу, потрібних ОС для керування цим процесом.

З *точки зору програмування* блок керування процесом може бути організовано по-різному (наприклад, як структури даних, як об'єкти).

*Деталі реалізації* блоку керування процесом залежать від конкретної ОС.

Але низка атрибутів блоку керування процесом досить *типова*.

# 3. Блок керування процесом

## Типові атрибути блоку керування процесом

### Основні ідентифікатори

- ідентифікатор процесу (Process Identifier - **PID**), унікальний у межах системи
- ідентифікатор батьківського процесу (parent PID, **PPID**)

### Інформація стану процесу (використовується для перемикання контекстів)

- вміст реєстрів ЦП, зокрема операнди, результат останньої команди, лічильник команд (яку команду виконувати наступною) тощо
- вказівники вершин стеків

### Безпекові атрибути

- ідентифікатор користувача, від імені якого запущено процес
- ідентифікатор групи, від імені якої запущено процес
- інші безпекові атрибути

*(продовження - на наступному слайді)*

# 3. Блок керування процесом

## Типові атрибути блоку керування процесом (продовження)

### Відомості для планувальника

- стан процесу
- пріоритет (пріоритети)
- вказівники на черги, у яких перебуває процес
- відомості про подію, на яку очікує процес

### Відомості для керування пам'яттю

- залежать від системи керування пам'яттю в ОС (базовий і межовий реєстри, таблиця сегментів, таблиця сторінок тощо).

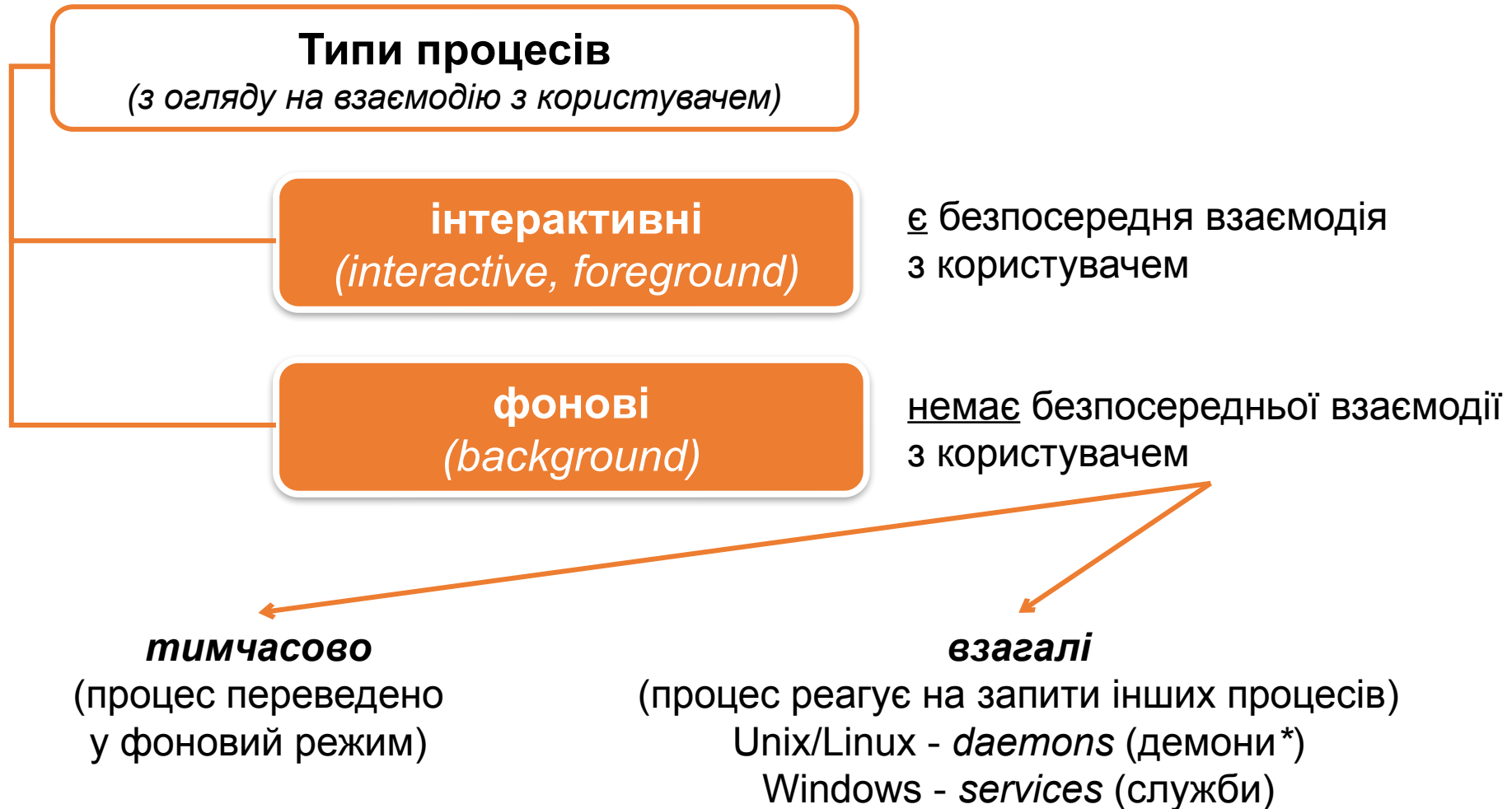
### Облік споживання процесом ресурсів

- використаний час ЦП і загальний час роботи процесу
- часові обмеження тощо

### Відомості про введення-виведення

- перелік пристроїв введення-виведення, виділених процесу
- перелік відкритих файлів тощо

## 4. Інтерактивні і фонові процеси. Створення і завершення процесу



\* **Примітка.** Зазвичай перекладають як *демони*, хоча *демони* - це *demons*, а *daemons* - це радше *духи*. Перше має більш негативний відтінок, друге - менш негативний.

## 4. Інтерактивні і фонові процеси. Створення і завершення процесу

### Процес може створюватися:

- під час ініціалізації ОС (наприклад, *systemd* у Linux)
- під час виконання інших процесів

### Процес може завершуватися:

- коли процес виконав свою роботу до кінця  
(і сам робить відповідний системний виклик для завершення)
- коли процес завершено іззовні...
  - ...бо процес перевикористав ресурси
  - ...бо процес більше не потрібний
  - ...бо завершено батьківський процес (у деяких системах, це т. з. *cascading termination* - каскадне завершення)

## 4. Інтерактивні і фонові процеси. Створення і завершення процесу

Батьківський процес може отримувати статус завершеного дочірнього процесу.

У Unix/Linux є особливий стан процесу - *зомбі (zombie)*.

Процес у стані **зомбі** фактично завершений, але його блок керування ще не вилучено (ці дані треба спершу передати батьківському процесу).

Остаточне завершення робить батьківський процес - системним викликом `waitpid()`.

# Для самоїтнього читання

1. [**Silberschatz, Galvin, Gagne, 2018**] Chapter 1 (paragr. 1.2.1, Chapter 3 (paragr. 3.1).
2. [**Stollings, 2017**] Chapters 3-4.
3. [**Tanenbaum, Bos, 2014**] Chapter 2 (paragr. 2.1) + chapter 1 (paragr. 1.4)
4. [**Шеховцов, 2009**] Розділи 2-3.