МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
імені Семена Кузнеця

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
імені Семена Кузнеця

Л е к ц і я № *4*

Змістовний модуль № 1: Системне програмування в Windows

по курсу "**Системне програмування**"

Тема лекції:

Процеси та потоки

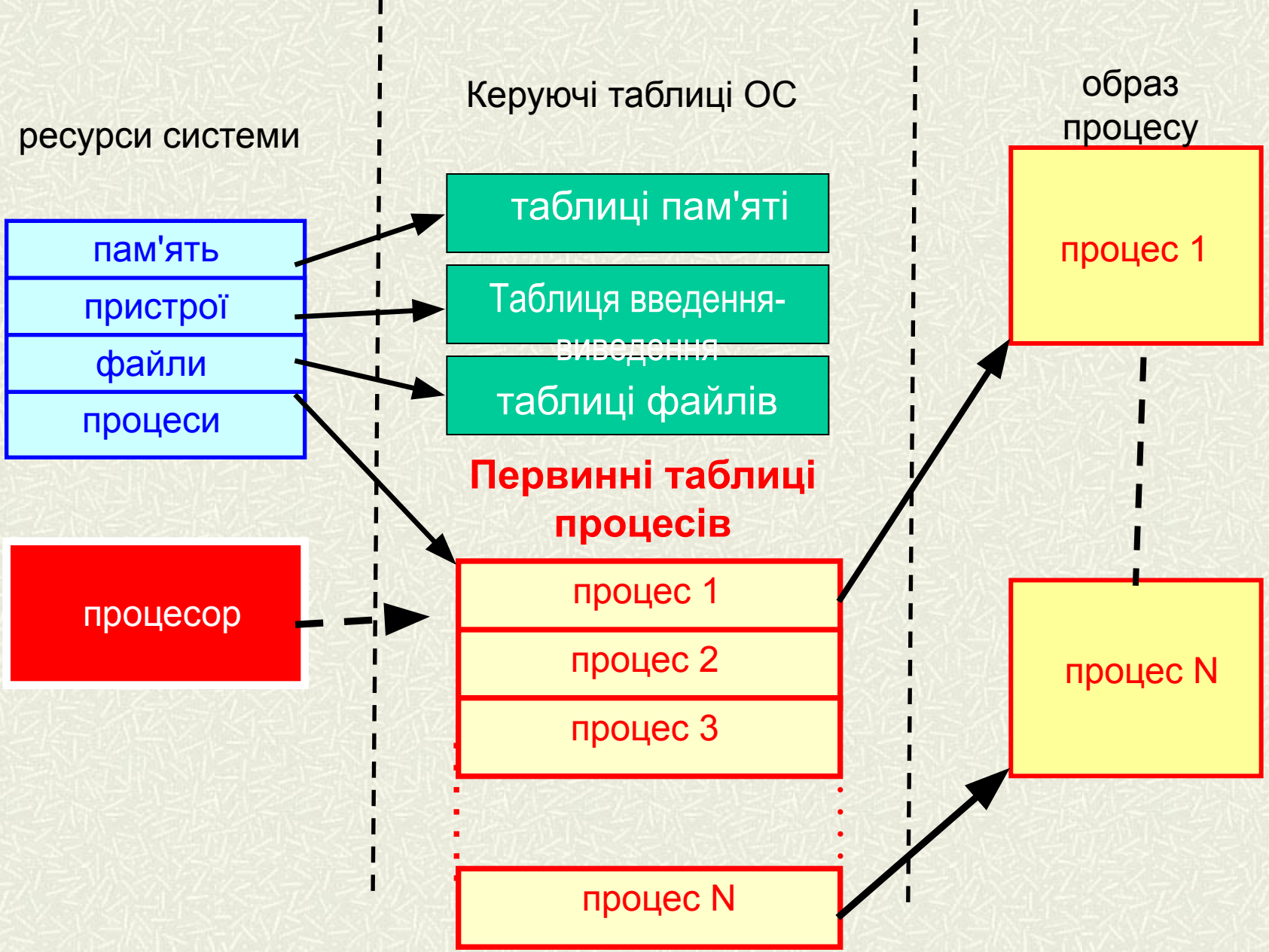
Лектор:
Доцент кафедри Інформаційних систем
кандидат технічних наук, доцент
Голубничий Дмитро Юрійович

05.09.2021

НАВЧАЛЬНІ ПИТАННЯ:

1. Процеси.
2. Потоки.
3. Планування процесів і потоків.
4. Система пріоритетів.

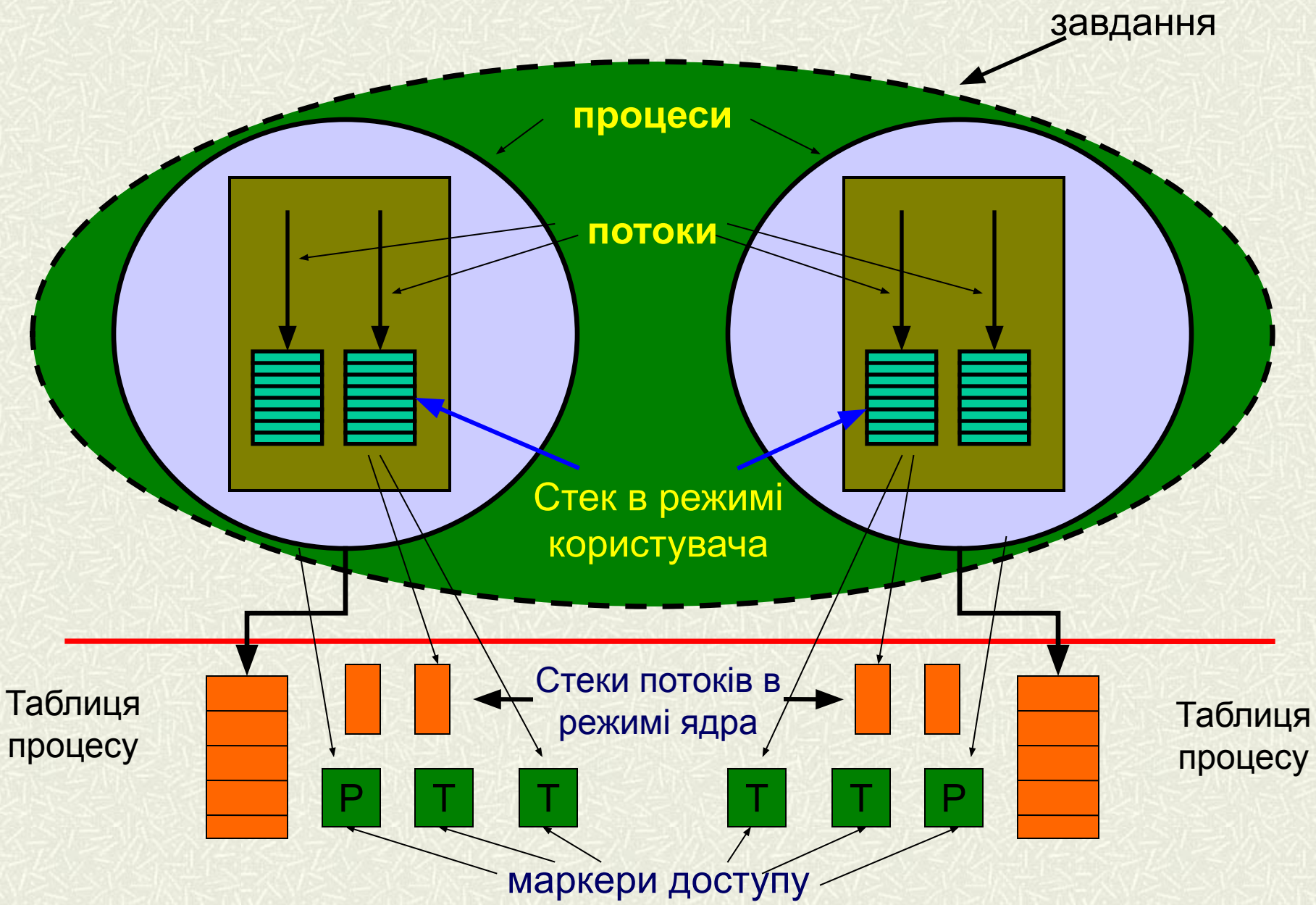
1. Процеси



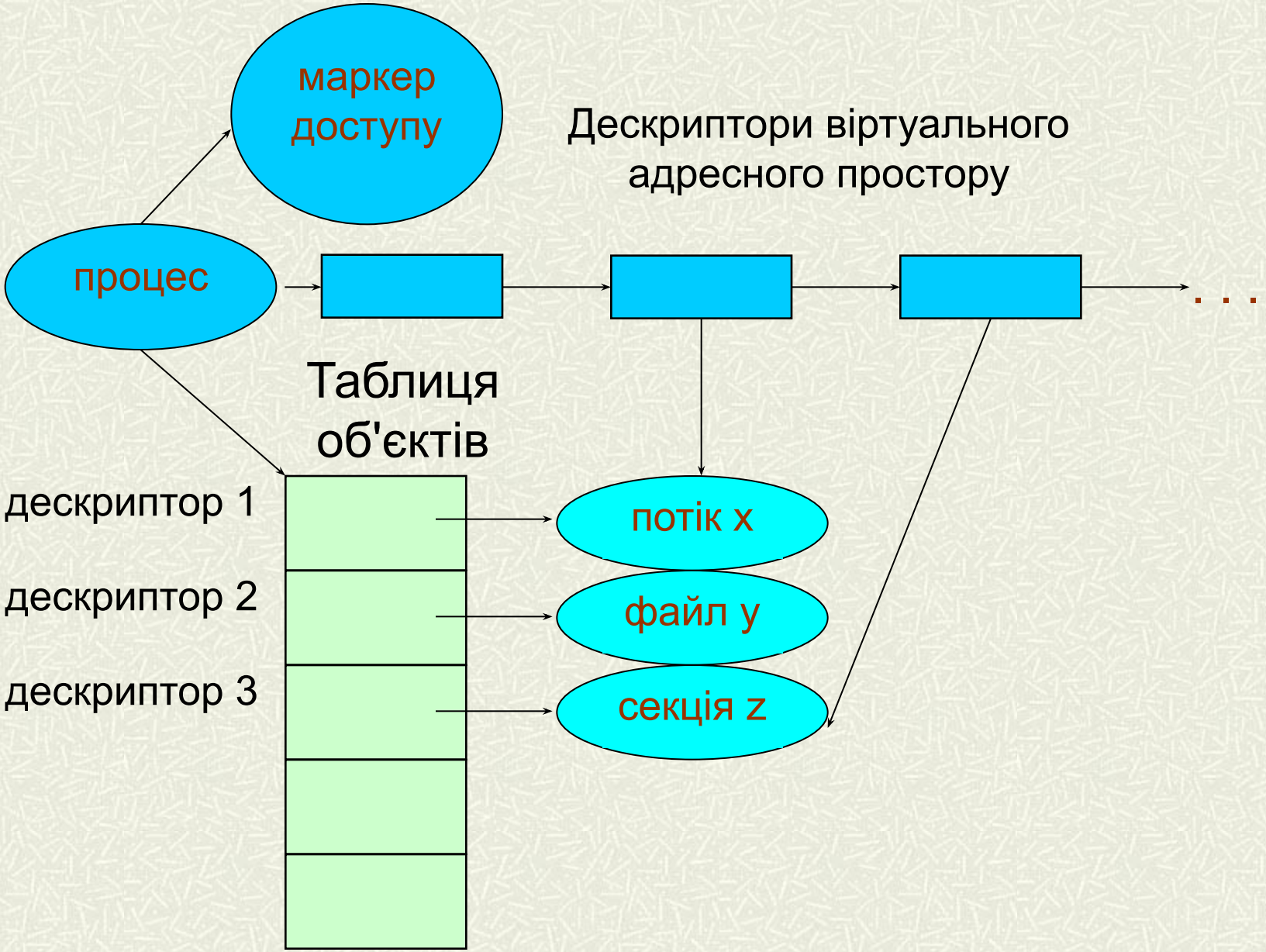
Концепція процесів і потоків

Поля таблиці процесів

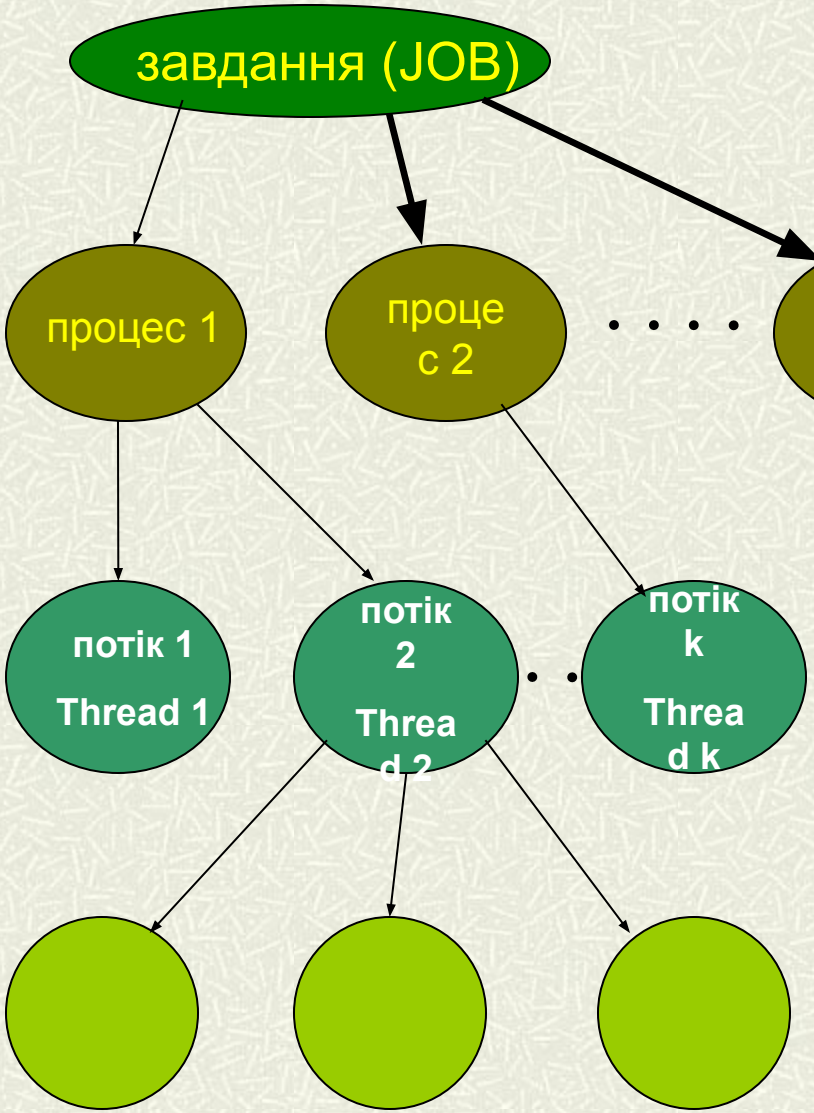
управління процесом	управління пам'яттю	управління файлами
регістри	Покажчик на текстовий сегмент	Кореневий каталог
Лічильник команд	Покажчик на сегмент даних	Робочий каталог
покажчик стека	Покажчик на сегмент стека	Дескриптори файлу
стан процесу		ідентифікатор користувача
пріоритет		ідентифікатор групи
параметри планування		
ідентифікатор процесу		
Батьківський процес		
Група процесу		
Час початку процесу		
Використане процесорний час		



Взаємозв'язок між завданнями, процесами і потоками



Процес і його ресурси (приклад)



волокна (Fibers)

об'єкти

Назва	опис
завдання	Набір процесів із загальними квотами і лімітами
процес	Контейнер для ресурсів і потоків
потік	Виконання коду в процесі
волокно	Полегшений потік, повністю керований в просторі користувача

процес *(process)* - абстракція, що описує поточну програму

процес (process) - це програма та системні ресурси, необхідні для її роботи.

процес - програма, яка перебуває в режимі виконання

процес *в Win32, Win64* **-** це об'єкт, який не виконується, а просто "володіє" виділеним йому адресним простором, іншими словами процес є структурою в пам'яті.

У виконавчій системі ***процеси*** - це об'єкти, що створюються і знищуються диспетчером об'єктів

Тип об'єкта — **процес**

Атрибути тіла об'єкта

- ідентифікатор
- маркер доступу
- базовий
- Процесорний спорідненість за
- розміри ням
- Час
- Лічильник введення /
- Лічильник операцій віртуальної
- Порти винятків-
- коди

сервіси

- створити
- відкрити
- Запросити інформацію
- Встановити інформацію
- Поточний
- завершити
- Виділити-звільнити віртуальну
- Читання / запис в віртуальну пам'ять
- Захистити віртуальну
- Блокувати / розблокувати віртуальну
- Опитати віртуальну
- Скинути віртуальну пам'ять на диск

Об'єкт -

Атрибути об'єкта - процесу

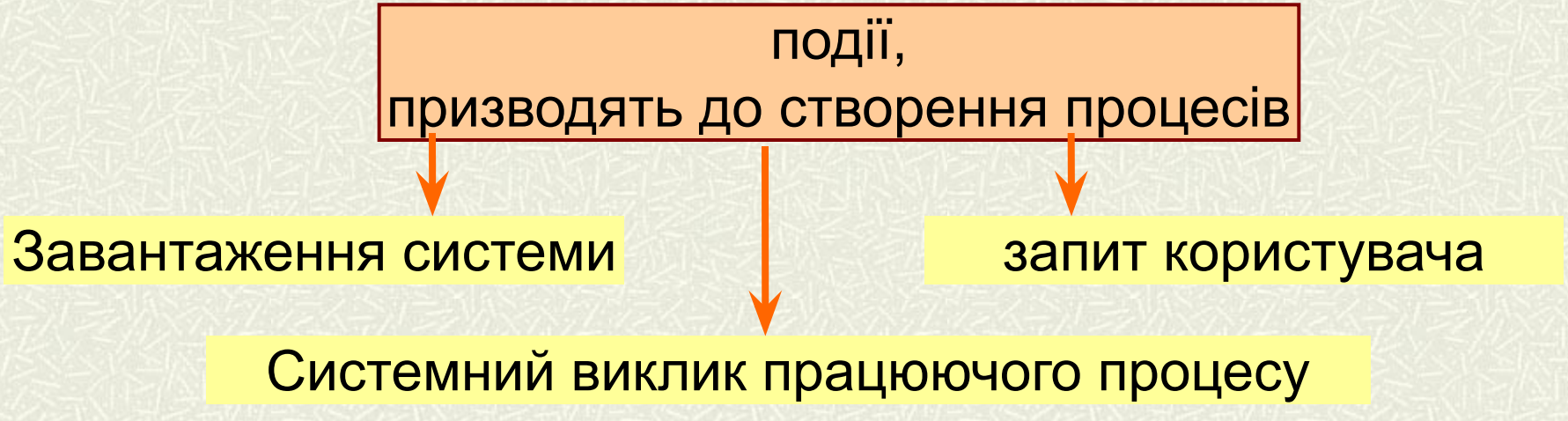
Атрибут	призначення
ідентифікатор процесу	Унікальне значення, яке ідентифікує процес в ОС
маркер доступу	Об'єкт виконавчої системи, що містить інформацію про права зареєстрованого в системі користувача, якого представляє цей процес
базовий пріоритет	Базовий пріоритет потоків процесу
Процесорний спорідненість за замовчуванням	Набір процесорів, на яких потоки процесу можуть виконуватися за замовчуванням
розміри квот	Максимальний обсяг резидентної і нерезидентної системної пам'яті, простору в файлі підкачки і процесорного часу, що виділяється для користувача процесу
Час виконання	Загальний час виконання всіх потоків процесу
Лічильники введення-виведення	Змінні, в яких записується число і тип операцій введення-виведення, виконаних потоками процесу
Лічильники операцій віртуальної пам'яті	Змінні, в яких записується число і тип операцій віртуальної пам'яті, виконаних потоками процесу
Порти винятків-налагодження	Канали комунікацій між процесами, за якими диспетчер процесів посилає повідомлення, якщо один з потоків процесу викликає виняток
код завершення	Причина завершення процесу

сервіси процесів

сервіс	функція
створити процес	CreateProcess ()
відкрити процес	OpenProcess ()
запросити інформацію процесу	GetPriorityClass (), GetProcessTimes (), GetProcessWorkingSetSize (), GetExitProcessCode () і т.д.
встановити інформацію процесу	SetPriorityClass (), SetProcessWorkingSetSize ()
Поточний процес	GetCurrentProcess (), GetCurrentProcessId ()
завершити процес	ExitProcess () або TerminateProcess ().

створення процесів

Створити процес - це перш за все означає створити описувач процесу, в якості якого виступає одна чи кілька інформаційних структур, що містять всі відомості про процес, необхідні ОС для управління ним



BOOL **CreateProcess** (LPCSTR *lpApplicationName*, LPSTR *lpCommandLine*, LPSECURITY_ATTRIBUTES *lpProcessAttributes*, LPSECURITY_ATTRIBUTES *lpThreadAttributes*, BOOL *bInheritHandles*, DWORD *dwCreationFlags*, LPVOID *lpEnvironment*, LPCSTR *lpCurrentDirectory*, LPSTARTUPINFO *lpStartupInfo*, LPPROCESS_INFORMATION *lpProcessInformation*);

Порядок пошуку файлу виконуваної програми

1. В каталозі, в який завантажено додаток

↓ нем / ає

2. У поточному каталозі

↓ нем / ає

3. У системному каталозі Windows (% SystemRoot% \)

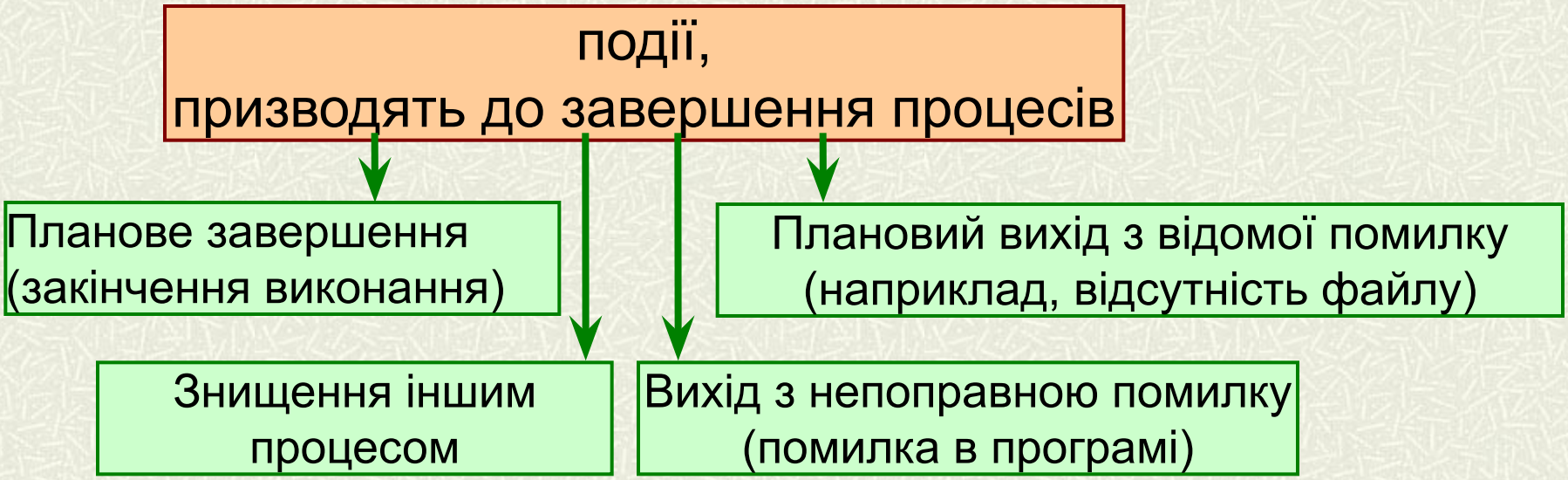
↓ нем / ає

4. У каталозі Windows, Який повертає функція GetWindowDirectory()

↓ нем / ає

5. У каталогах, перерахованих у змінній середовища PATH, Яка визначає порядок пошуку в них

завершення процесу



функція ExitProcess() завершує процес і все його потоки і повертає звичайний код завершення.

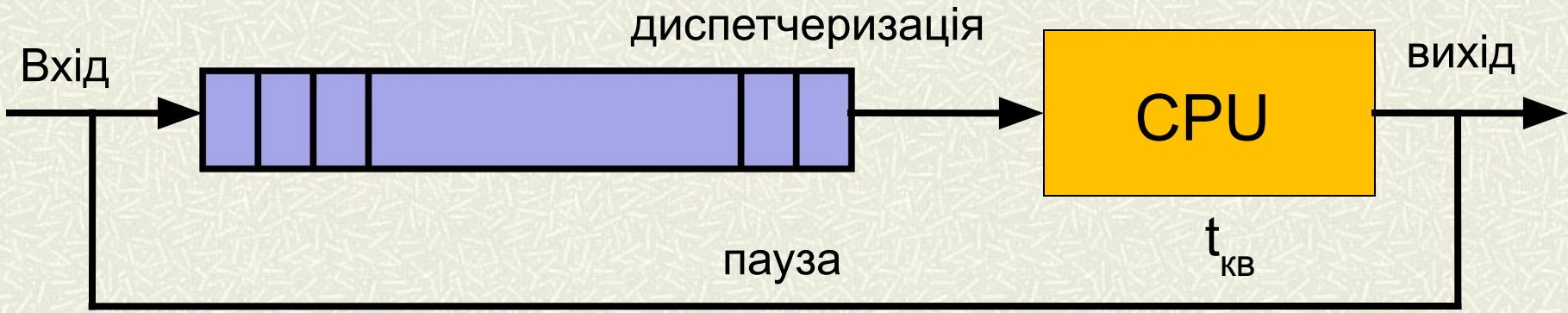
VOID ExitProcess (UINT *uExitCode*);

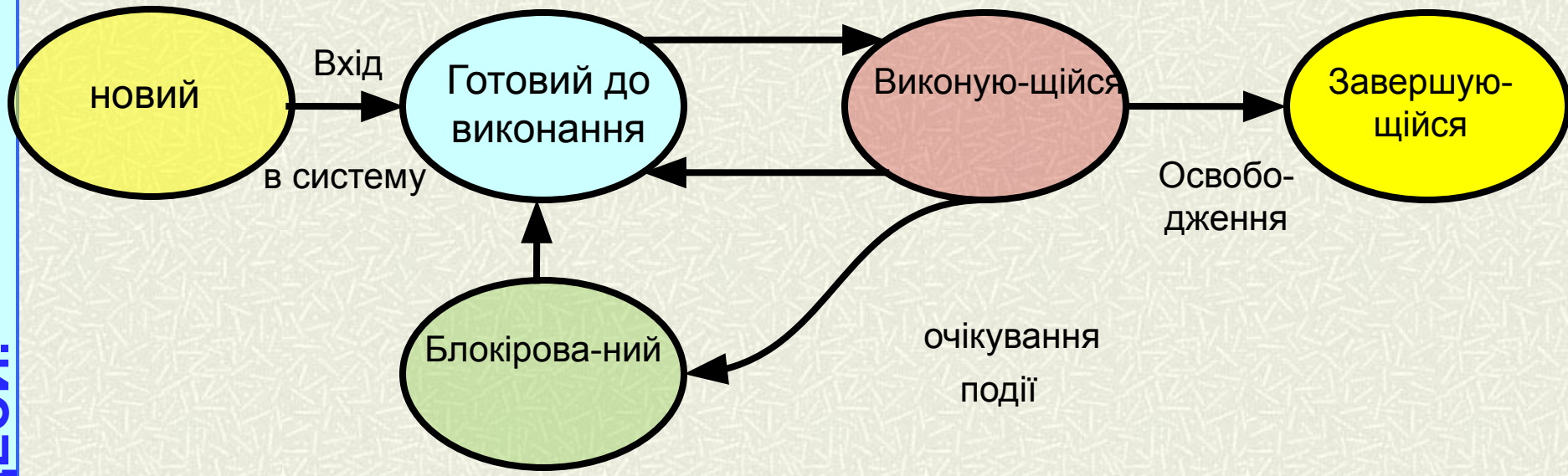
функція TerminateProcess()є аварійним засобом завершення процесу. викликається **ззовні** (!).

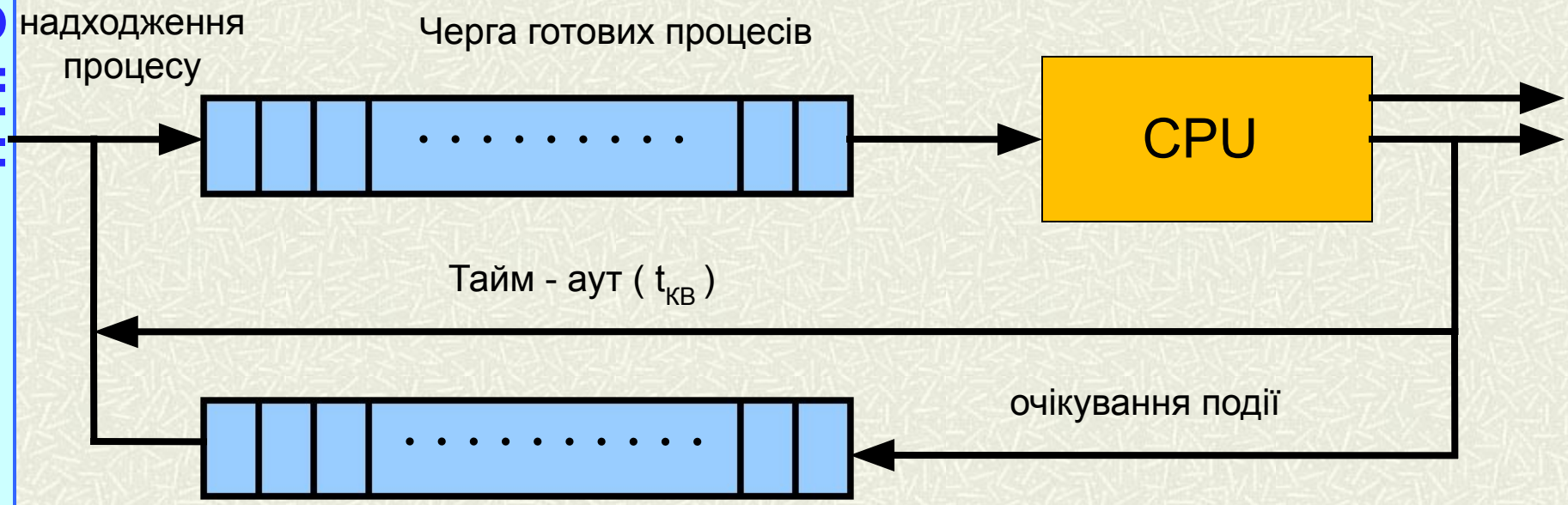
BOOL TerminateProcess (HANDLE *hProcess*, UINT *uExitCode*);

Найпростіша модель процесу

Граф станів і переходів

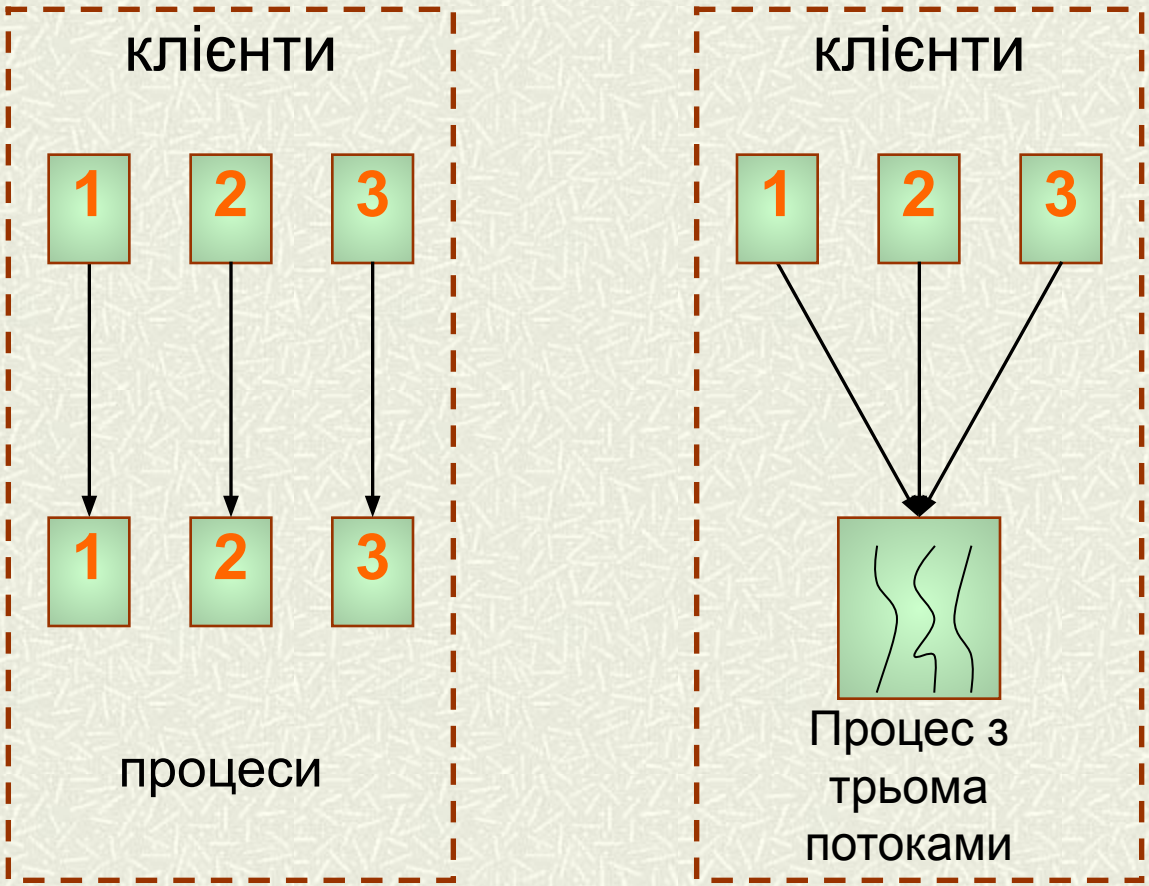



новий
Вхід
в систему
Готовий до виконання
Виконую-щійся
Завершую-щійся
Освобо-дження
Блокірова-ний
очікування події


надходження процесу
Черга готових процесів
CPU
Тайм - аут ($t_{КВ}$)
очікування події

2. Потоки

потік (Thread) **- це незалежний шлях виконання всередині процесу, що розділяє разом з процесом загальний адресний простір, код і глобальні дані.**



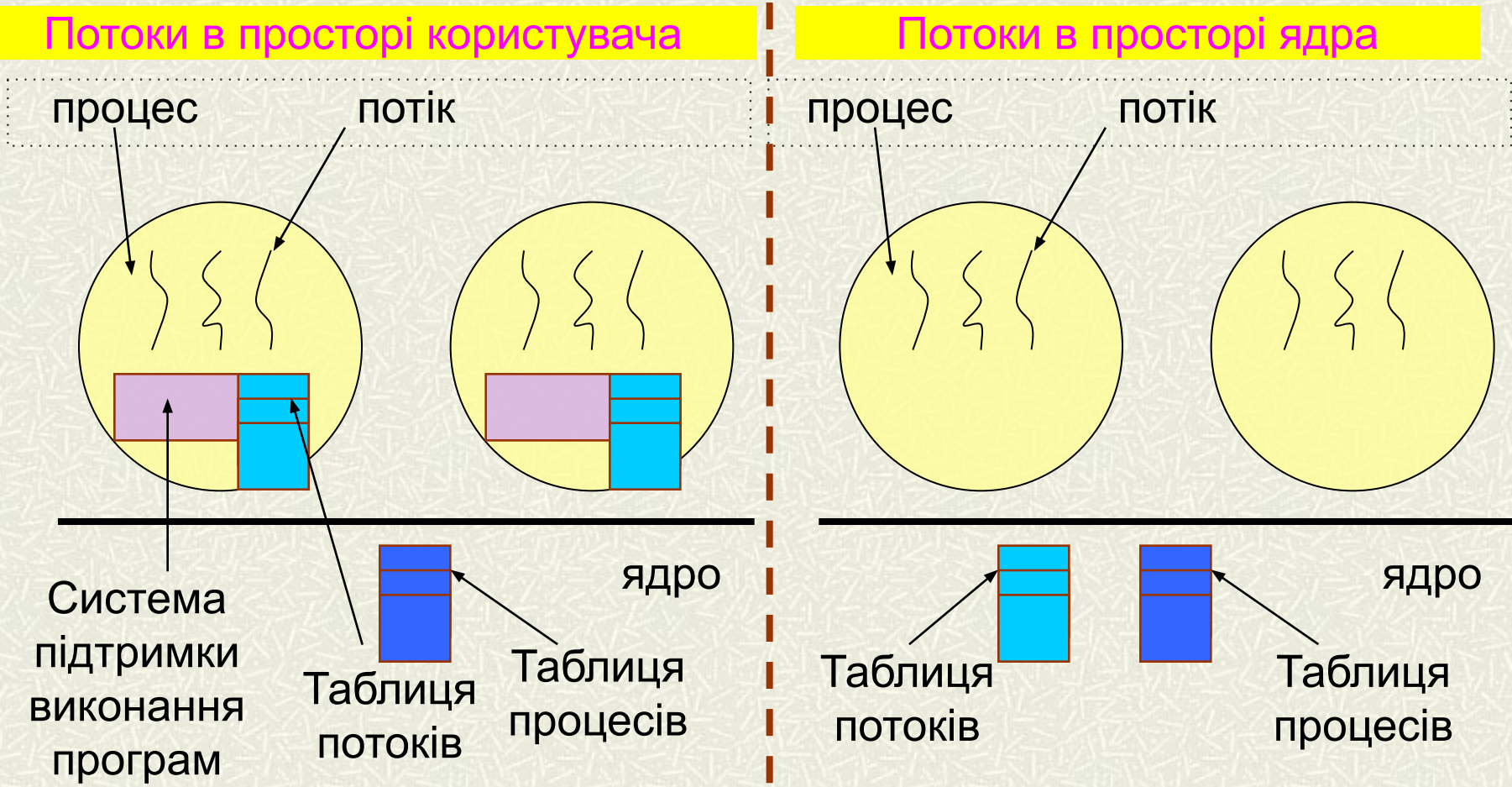
Переваги використання потоків:

1. Спрощення програми

2. Швидкість створення потоку

3. Підвищення продуктивності

Порівняння багатопотокової системи з однопоточному



Реалізація потоків у просторі користувача, ядра

ПОТІК НА РІВНІ КОРИСТУВАЧА

ПЕРЕВАГИ:

- можна реалізувати в ОС, що не підтримує потоки без будь-яких змін в ОС;
- висока продуктивність, оскільки процесу не потрібно перемикатися в режим ядра і назад;
- ядро про потоках нічого не знає і управляє однопоточні процесами;
- є можливість використання будь-яких алгоритмів планування потоків з урахуванням їх специфіки;
- управління потоками покладається на програму користувача.

НЕДОЛІКИ:

- системний виклик блокує не тільки працює потік, але і всі потоки того процесу, до якого він відноситься;
- додаток не може працювати в многопроцессорном режимі, так як ядро закріплює за кожним процесом тільки один процесор;
- при запуску одного потоку жоден інший потік а рамках одного процесу не буде запущений поки перший добровільно не віддасть процесор;
- всередині одного потоку немає переривань за таймером, в результаті чого неможливо створити планувальник за таймером для почергового виконання потоків.

ПОТІК НА РІВНІ ЯДРА

ПЕРЕВАГИ:

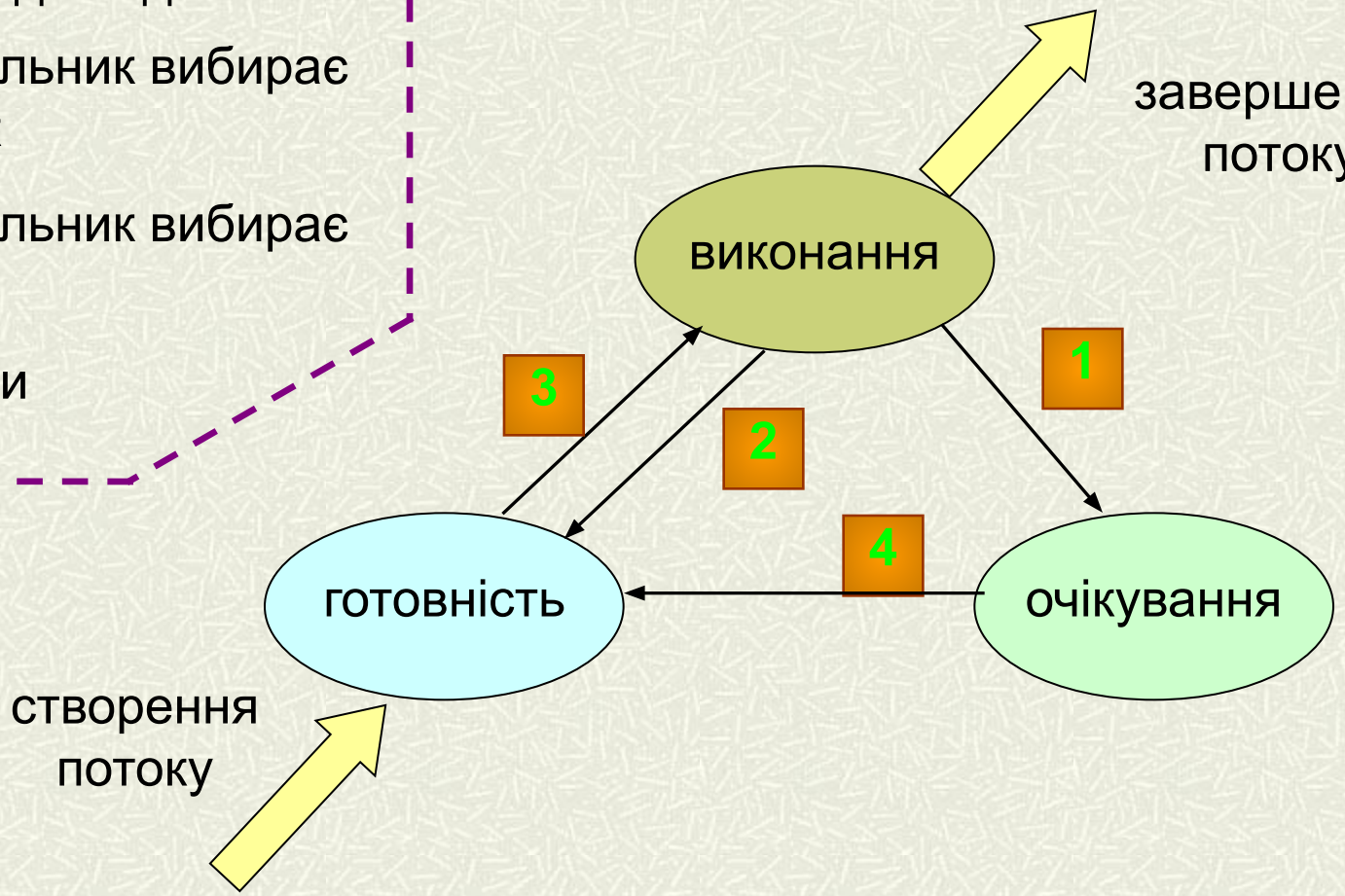
- можливо планування роботи декількох потоків одного і того ж процесу на декількох процесорах;
- реалізується мультипрограмування в рамках всіх процесів (у тому числі одного);
- при блокуванні одного з потоків процесу ядро може вибрати інший потік цього ж (або іншого процесу);
- процедури ядра можуть бути багато-.

НЕДОЛІКИ:

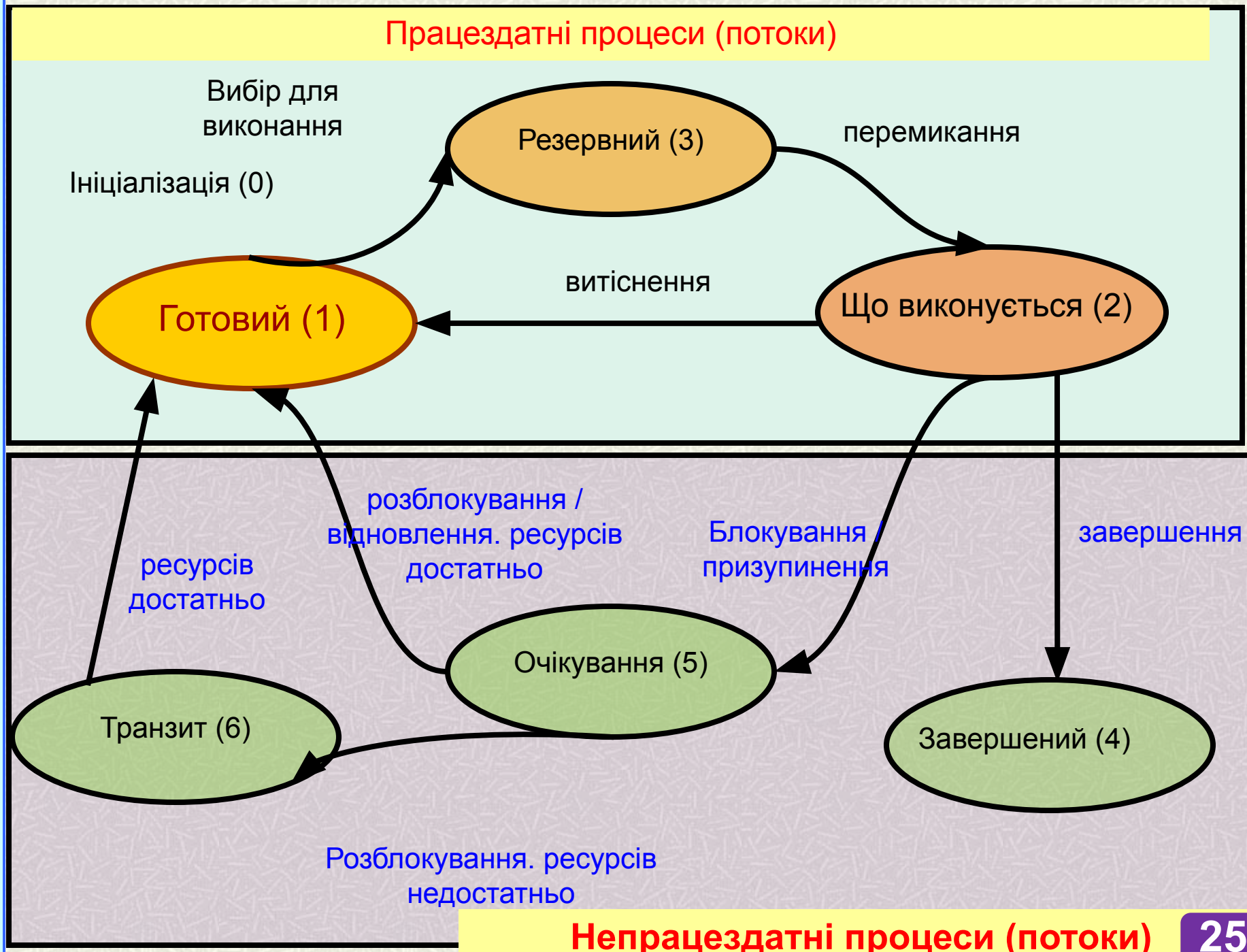
- Необхідність дворазового перемикання режиму користувач - ядро, ядро - користувач для передачі управління від одного потоку до іншого в рамках одного і того ж процесу.

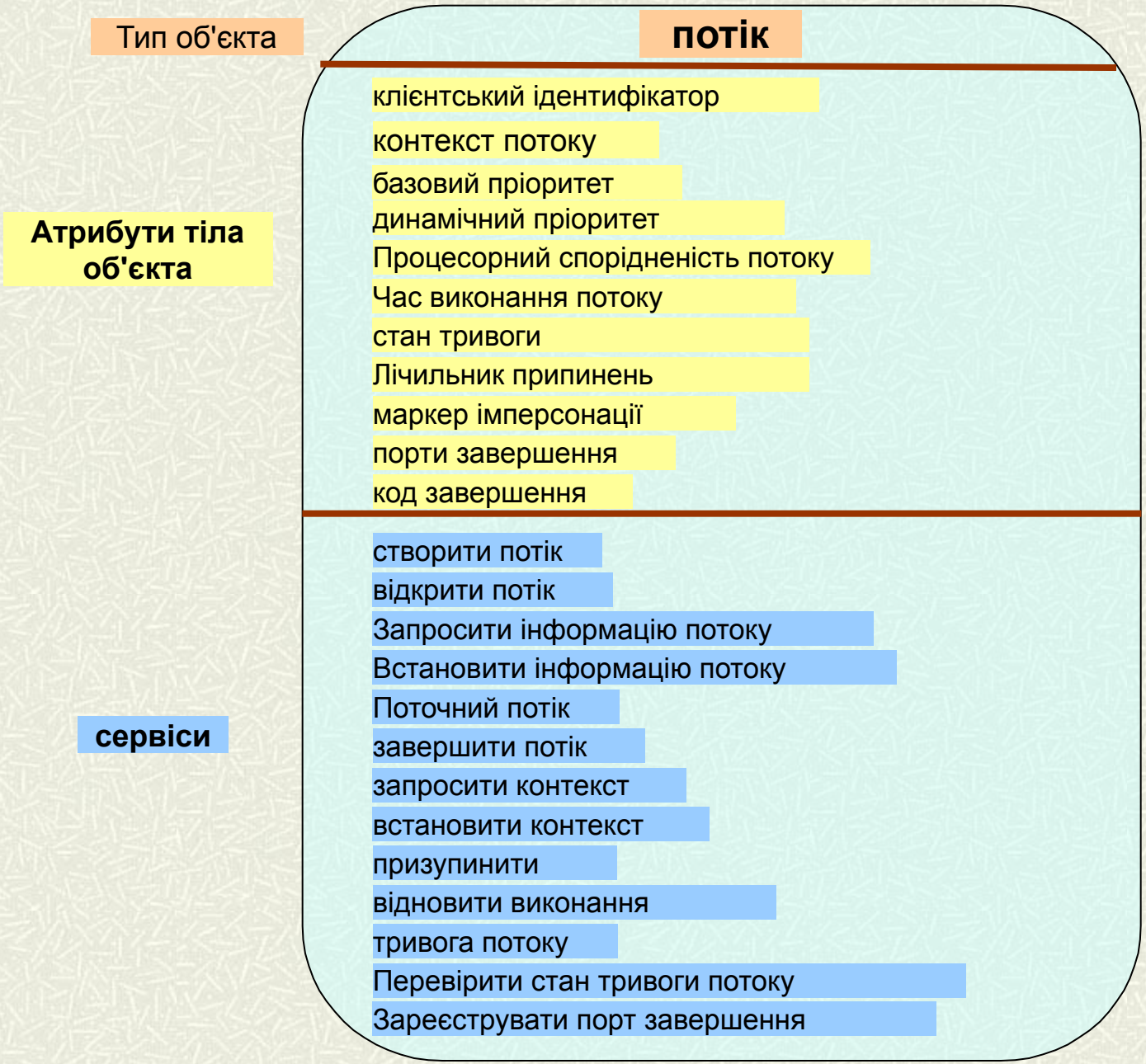
стан потоків

1. Потік блокується, чекаючи вхідних даних
2. Планувальник вибирає інший потік
3. Планувальник вибирає цей потік
4. Надійшли вхідні дані



стан потоку

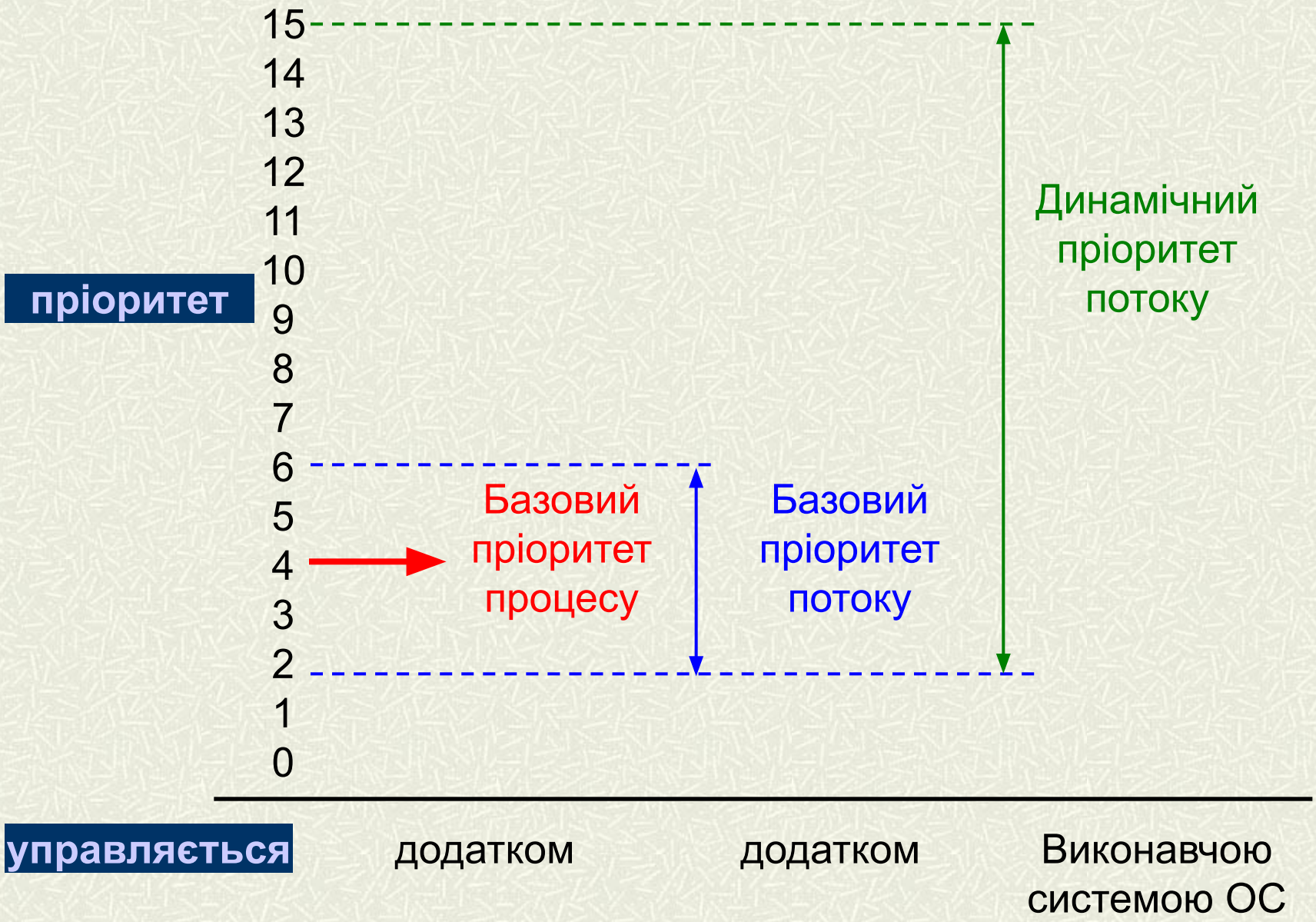

Працездатні процеси (потоки)
Вибір для виконання
Ініціалізація (0)
Резервний (3)
перемикання
Готовий (1)
витіснення
Що виконується (2)
розблокування / відновлення. ресурсів достатньо
Блокування / призупинення
завершення
ресурсів достатньо
Очікування (5)
Транзит (6)
Завершений (4)
Розблокування. ресурсів недостатньо
Непрацездатні процеси (потоки)


Тип об'єкта
потік
Атрибути тіла об'єкта
клієнтський ідентифікатор
контекст потоку
базовий пріоритет
динамічний пріоритет
Процесорний спорідненість потоку
Час виконання потоку
стан тривоги
Лічильник припинень
маркер імперсонації
порти завершення
код завершення
сервіси
створити потік
відкрити потік
Запросити інформацію потоку
Встановити інформацію потоку
Поточний потік
завершити потік
запросити контекст
встановити контекст
призупинити
відновити виконання
тривога потоку
Перевірити стан тривоги потоку
Зареєструвати порт завершення

Об'єкт - потік

Атрибути об'єкта-потоку

Атрибут	призначення
ідентифікатор клієнта	Унікальне значення, яке ідентифікує потік при виклику їм сервера
контекст потоку	Набір значень регістрів і інших непостійних даних, що визначають стан виконання потоку
базовий пріоритет	Нижня межа динамічного пріоритету потоку
динамічний пріоритет	Пріоритет потоку на даний момент
Процесорний спорідненість потоку	Набір процесорів, на яких може виконуватися потік, (неповне) підмножина процесорного спорідненості процесу потоку
Час виконання потоку	Загальний час виконання потоку в призначеному для користувача режимі і режимі ядра
статус оповіщення	Прапор, який вказує на необхідність відпрацювання потоком асинхронного виклику процедури (APC)
Лічильник припинень	Кількість припинень виконання потоку без подальшого відновлення
маркер імперсонації	Тимчасовий маркер доступу, що дозволяє потоку виконувати дії від імені іншого процесу (використовується підсистемами)
порт завершення	Канал комунікації між процесами, в який диспетчер процесів посилає повідомлення при завершенні потоку (використовується підсистемами)
Код завершення потоку	Причина завершення потоку



пріоритети потоків

створення потоку

Потрібно інформація:

1. Розмір стека потоку
2. Атрибути захисту потоку
3. Адреса процедури, з якої починається виконання
4. Додаткове 32-розрядне значення
5. Прапори
6. Адреса для зберігання ідентифікатора потоку

HANDLE **CreateThread** (LPSECURITY_ATTRIBUTES *lpThreadAttributes*, DWORD *dwStackSize*, LPTHREAD_START_ROUTINE *lpStartAddress*, LPVOID *lpParameter*, DWORD *dwCreationFlags*, LPDWORD *lpThreadId*);

завершення потоку

функція **ExitThread()** використовується для нормального завершення потоку.

VOID ExitThread (DWORD *dwExitCode*);

функцію **TerminateThread()** слід викликати тільки в крайніх випадках, коли потік завис, і ні на які дії користувача не реагує

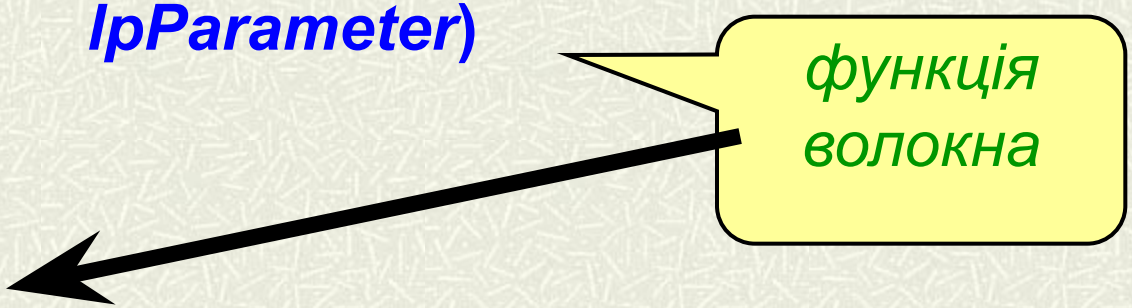
BOOL TerminateThread (HANDLE *hThread*, DWORD *dwExitCode*);

ВОЛОКНО

- спрощений потік, виконання якого планується в додатку вручну

Для створення волокон в додатку застосовується функція

LPVOID CreateFiber (DWORD *dwStackSize*, LPFIBER_START_ROUTINE *lpStartAdress*, LPVOID *lpParameter*)



VOID WINAPI FiberFunc (PVOID *lpParameter)*

перемикання між волокнами здійснюється:

VOID SwitchToFiber (LPVOID *lpFiber)*

Для очищення даних волокна і його видаленні використовують:

VOID DeleteFiber (LPVOID *lpFiber)*

3. Планування процесів і потоків

Процес не може сам перейти з одного стану в інший

пари операцій з потоками:

створення процесу	завершення процесу
призупинення процесу	запуск процесу
блокування процесу	розблокування процесу

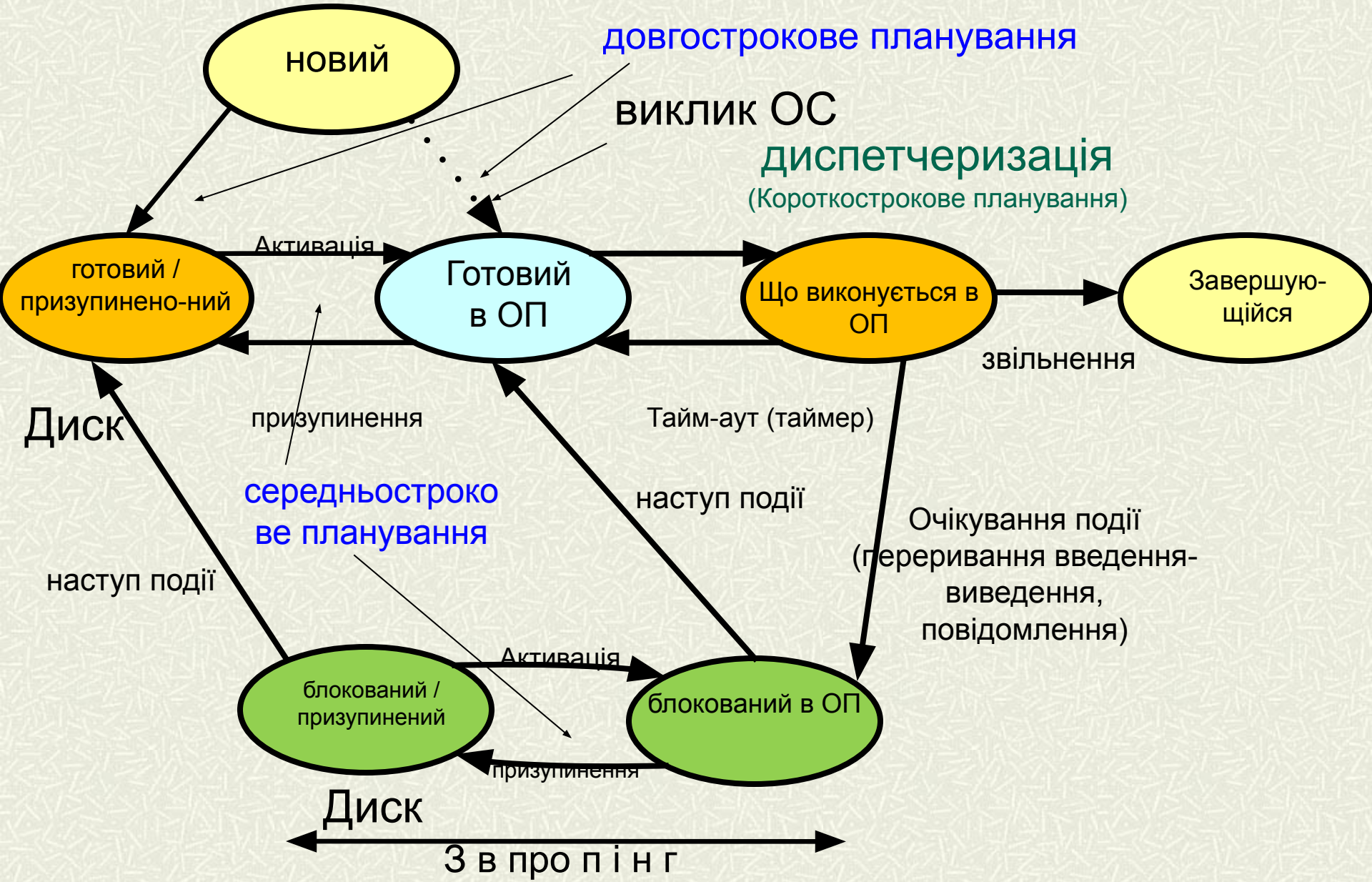
Мультипрограмування, або багатозадачність (multitasking) - це спосіб організації обчислювального процесу, при якому на одному процесорі поперемінно виконуються відразу кілька програм.

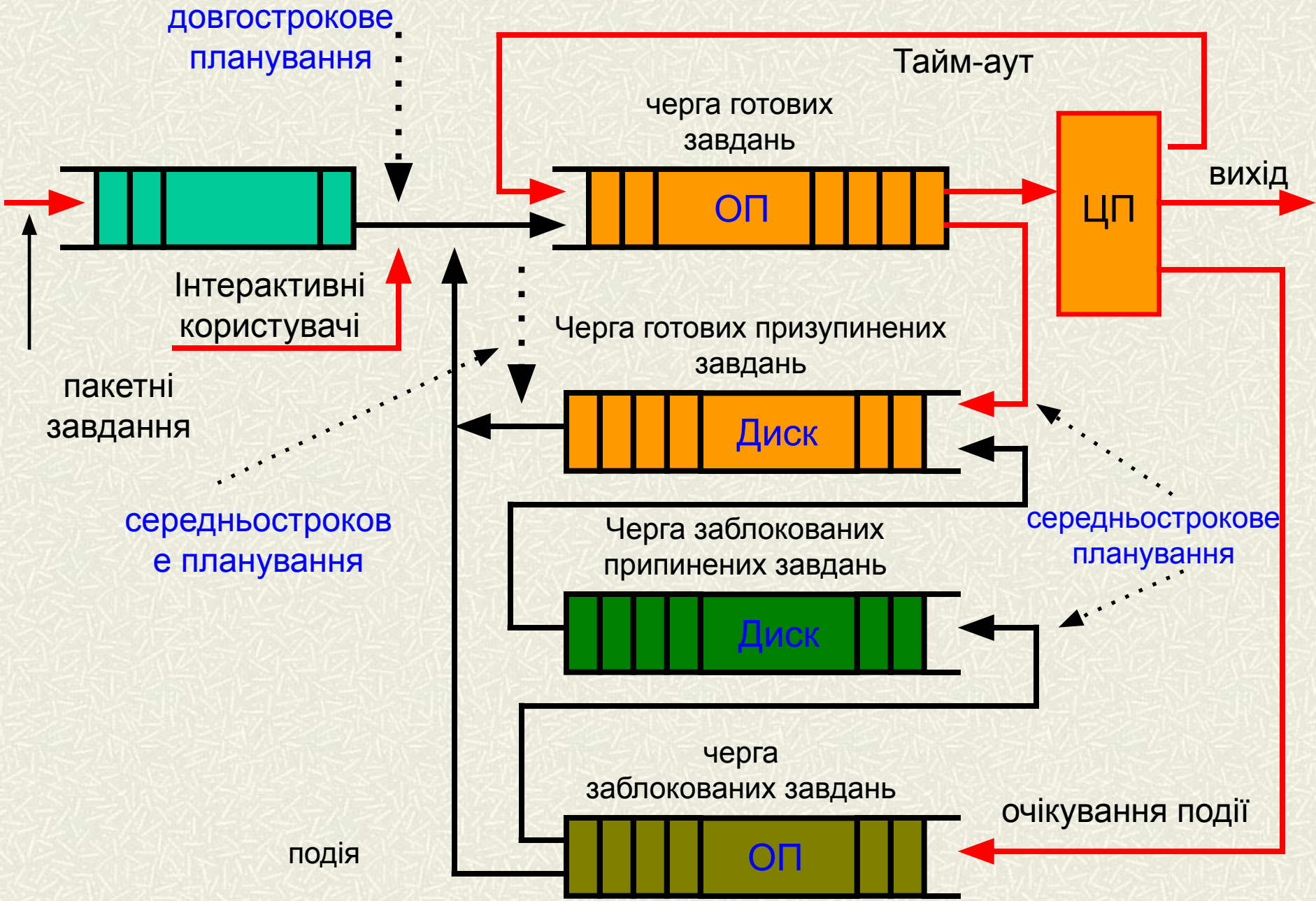
планування - забезпечення почергового доступу процесів до одного процесора

планувальник (scheduler) - відповідає за планування частина операційної системи

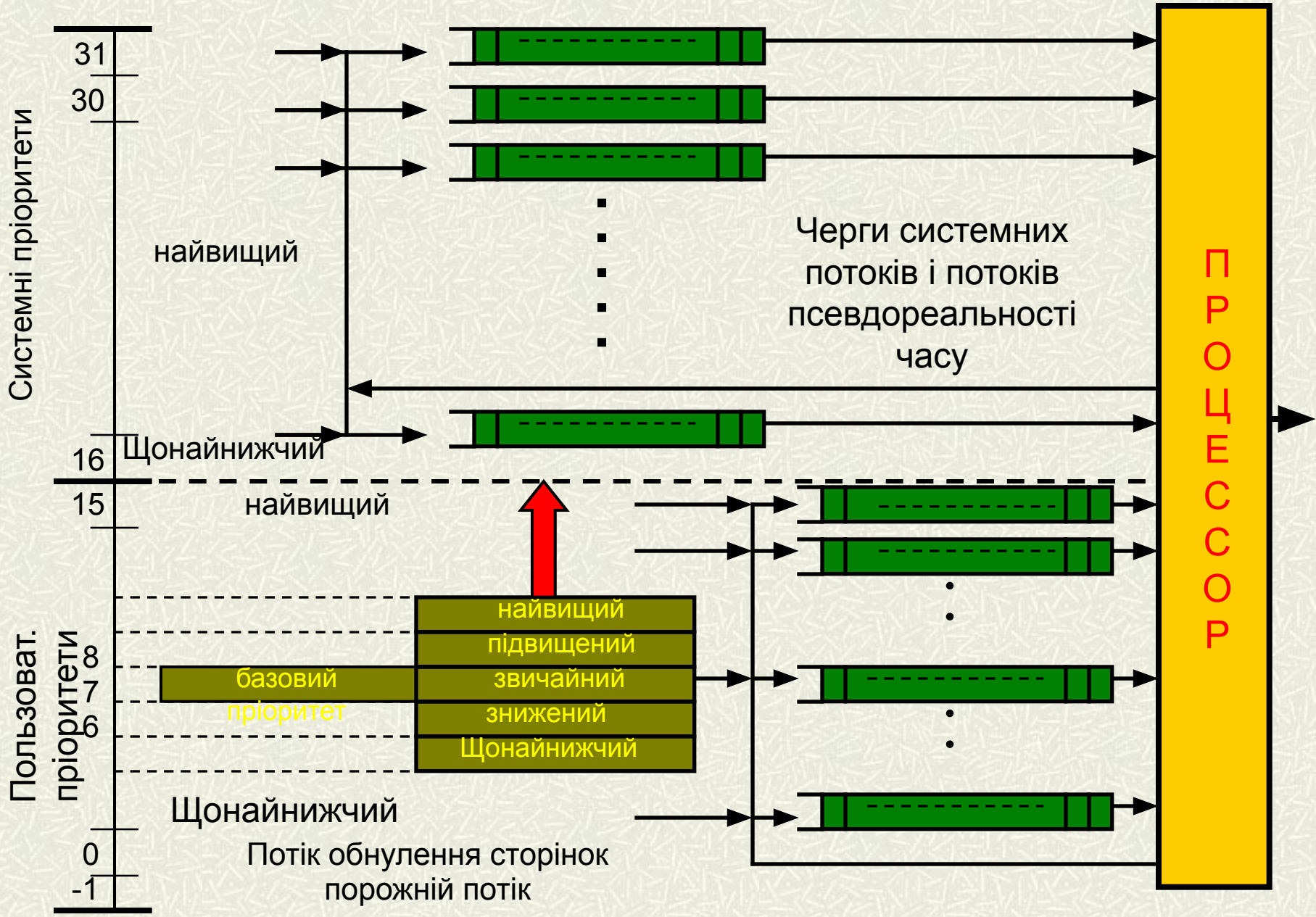
Класифікація алгоритмів планування

СХЕМА ПЛАНУВАННЯ З УРАХУВАННЯМ ЧЕРГ ЗАВДАНЬ (ПОТОКОВ)

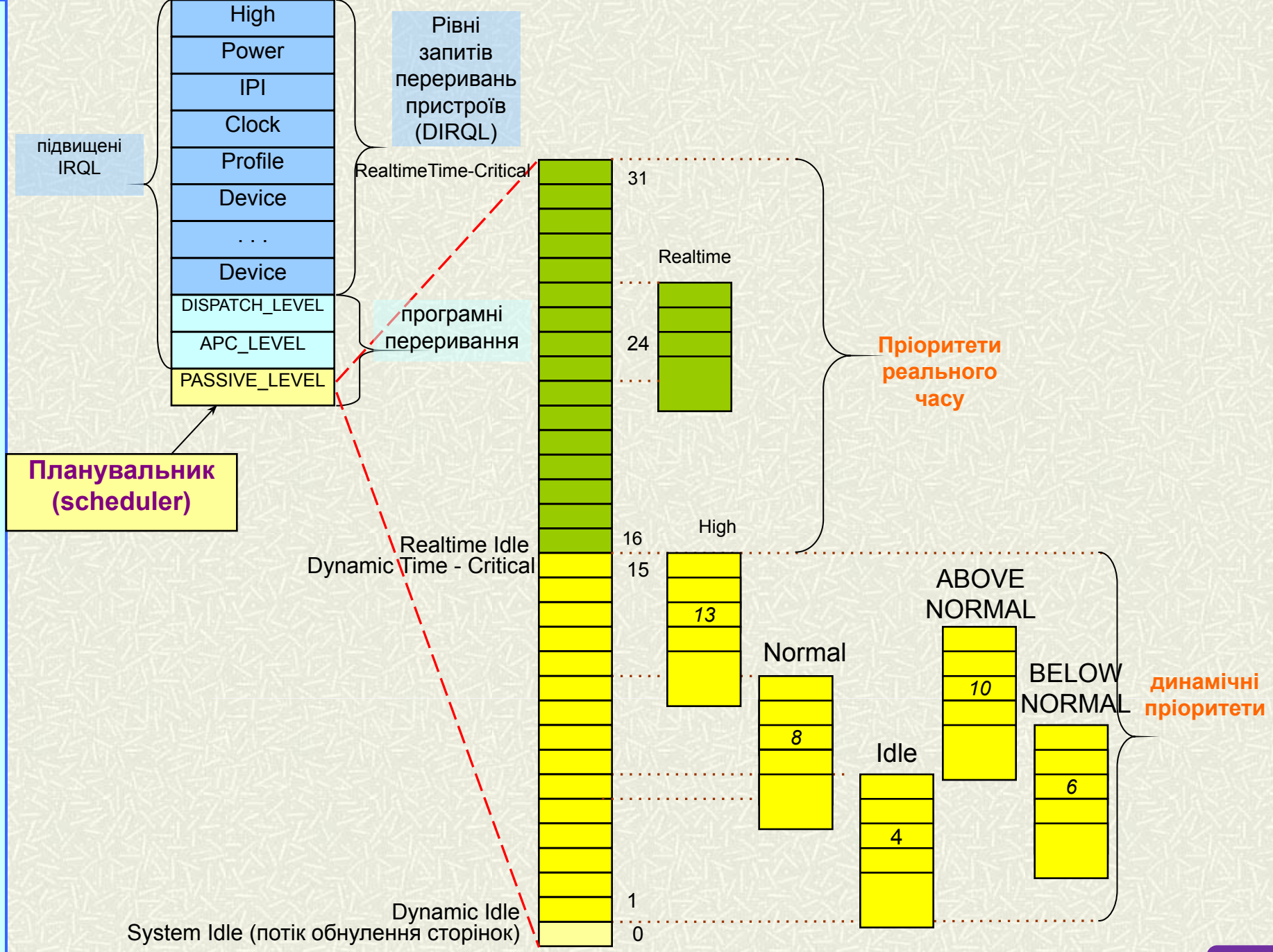



довгострокове планування
Тайм-аут
черга готових завдань
ОП
ЦП
вихід
Інтерактивні користувачі
пакетні завдання
Черга готових призупинених завдань
Диск
середньостроков е планування
Черга заблокованих припинених завдань
середньострокове планування
Диск
черга заблокованих завдань
очікування події
подія
ОП

4. Система пріоритетів



система пріоритетів



система пріоритетів

ЗМІНА БАЗОВОГО ПРІОРИТЕТУ ПОТОКУ

збільшення пріоритету

+ 1 - завершення введення-виведення по диску;

+2 - для послідовної лінії;

+ 6 - клавіатура;

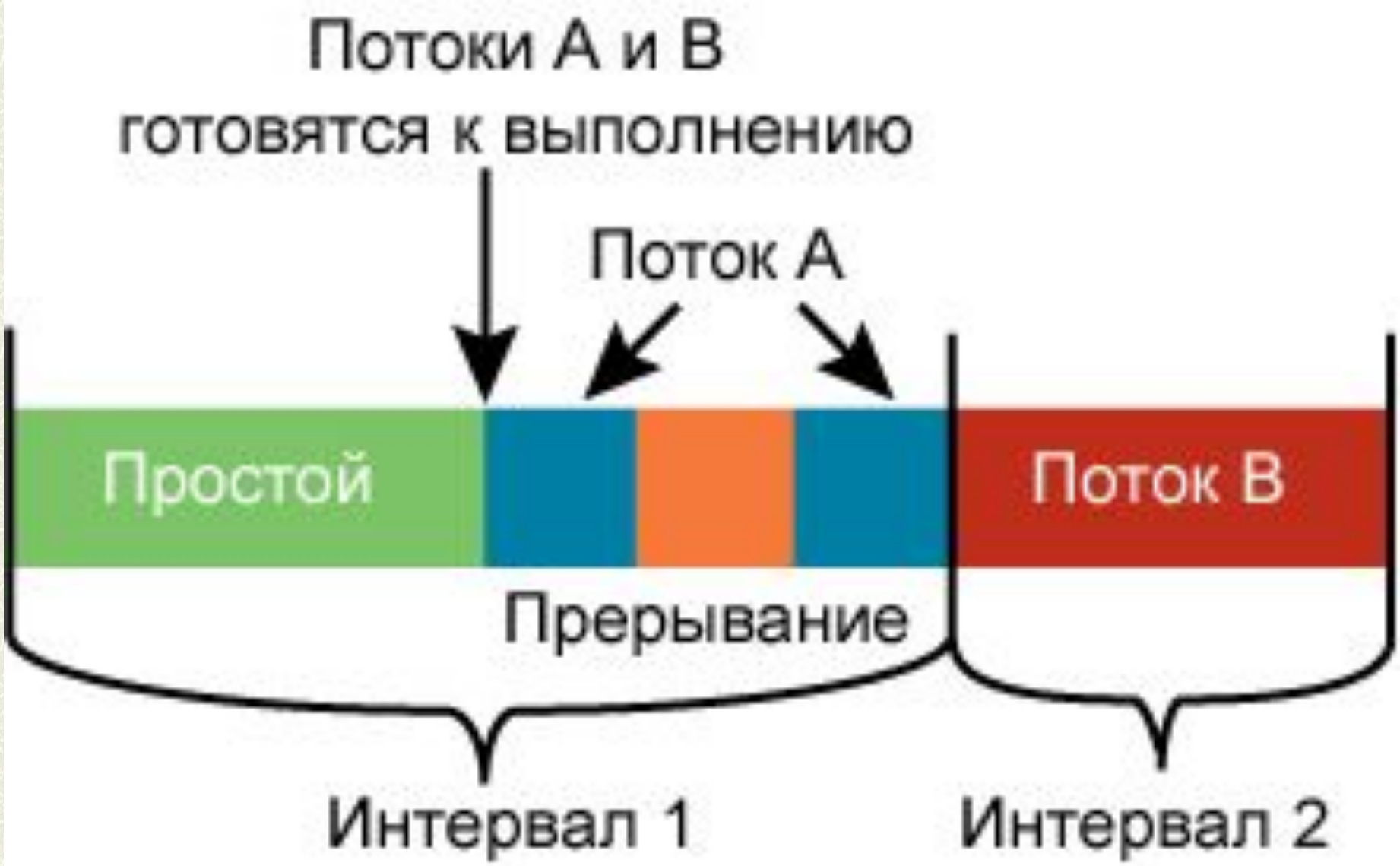
+ 8 - звукова карта;

+ 2 - знімається блокування по семафора (для потоку переднього плану);

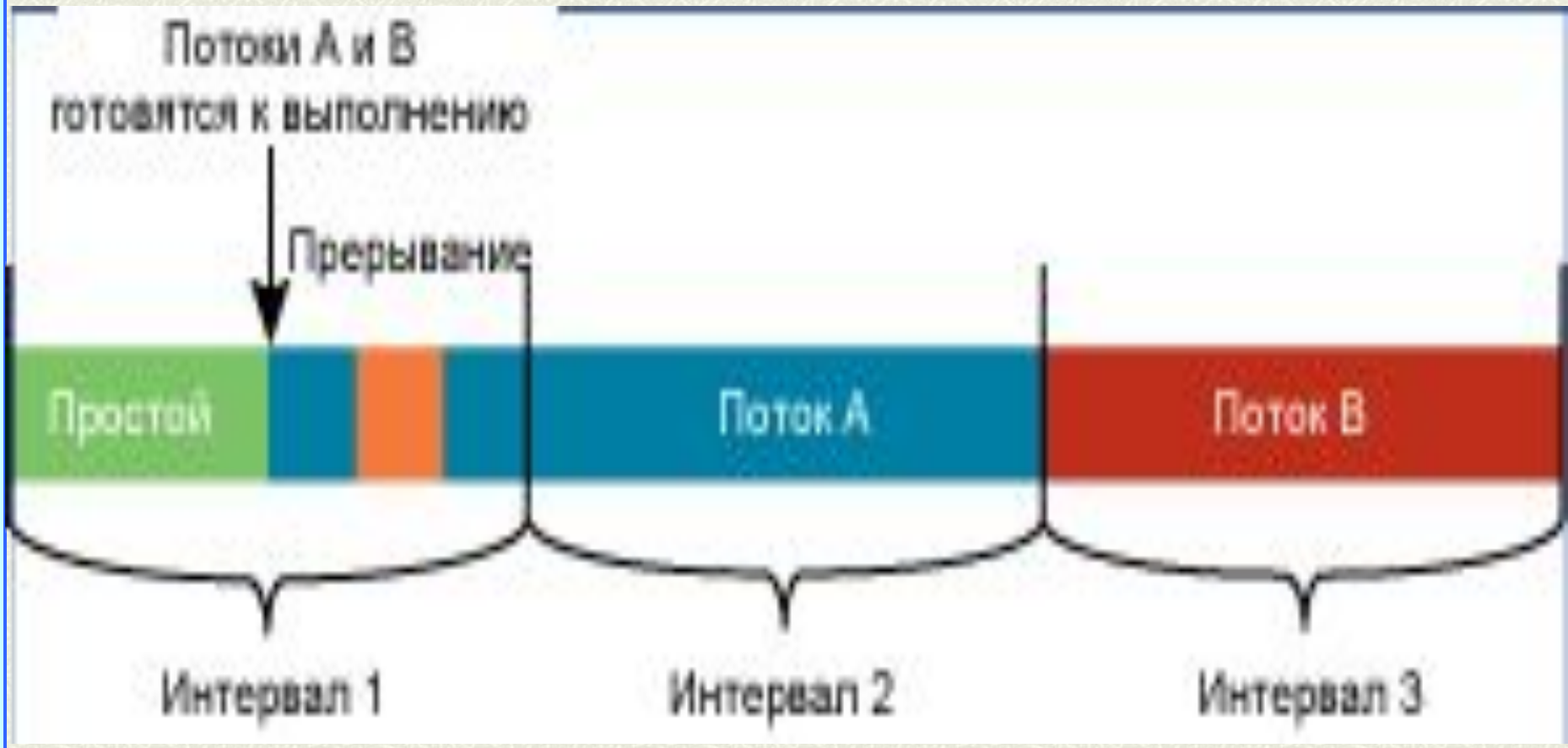
+ 1 - знімається блокування по семафора (для потоку непереднього плану); пріоритет 15 на 2 кванта процесора, якщо готовий до виконання потік простоює понад деякого директивного часу.

зменшення пріоритету

- 1 - якщо повністю використаний квант часу процесора (багаторазово, аж до базового пріоритету).



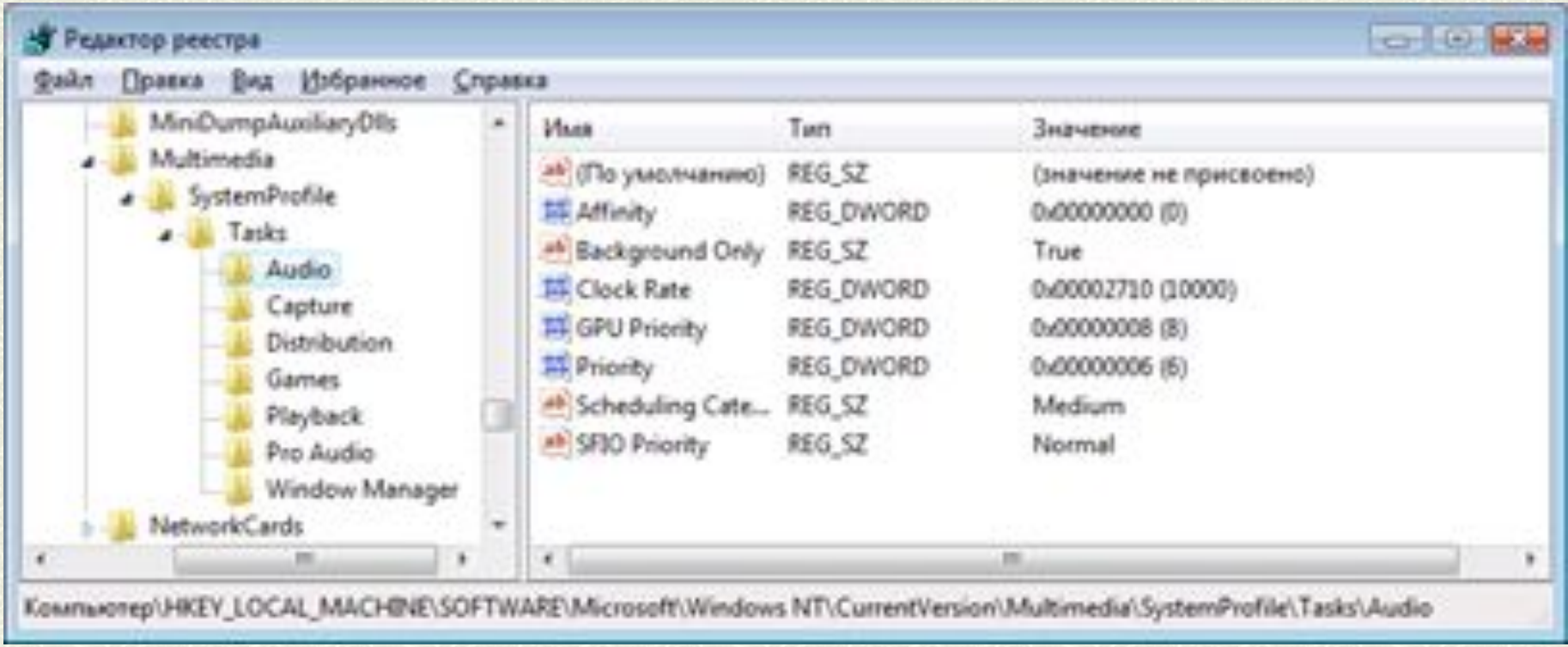
Нерівномірний розподіл ресурсів потокам



Планування розподілу ресурсів на основі циклів в ОС Windows

служба **Multimedia Class Scheduler Service (MMCSS)** реалізована в файлі **% SystemRoot% \ System32 \ Mmcss.dll**

HKEY_LOCAL_MACHINE \ Software \ Microsoft \ WindowsNT \ Currentversion \ Multimedia \ SystemProfile \ Tasks



Визначення аудіозадачі за розкладом класів мультимедіа

Дякую за увагу!