

6.7 ПОТОКИ У WINDOWS 2000

Одним з основних понять, які вводяться при розгляді механізму організації мультизадачного режиму виконання програм в 32-х розрядних операційних системах MS Windows, є потік (thread).

Для кожного додатка Win32 ОС створює окремий процес. Контекст процесу включає віртуальний адресний простір додатка і ряд інших системних ресурсів, які використовуються потоками. Один або декілька потоків можуть бути організовані всередині одного процесу. Такі потоки спільно використовують пам'ять і інші ресурси, виділені процесу. Перемикання потоків, які належать різним процесам, є складнішою процедурою, ніж перемикання потоків, які належать одному процесу.

6.7.1 Багатопоточність

Кожен процес повинен містити, принаймні, один потік який називається першим потоком (primary thread). Цей потік формується при створенні процесу. Інші потоки можуть бути створені згодом з будь-якого існуючого потоку процесу за допомогою функцій API [28]. Процеси в ОС Windows 2000 (W2K) організовані так, щоб забезпечити підтримку різних операційних середовищ. Процеси в різних середовищах відрізняються рядом параметрів, включаючи такі:

- іменування процесів;
- підтримка потоків у процесах;
- спосіб представлення процесів;
- спосіб захисту ресурсів процесів;
- взаємозв'язок процесів один з одним.

Відповідно структури і сервіси процесів, що надаються ядром W2K, порівняно прості і мають загальне призначення. Процеси реалізовані як об'єкти. При вході користувача в систему створюється ознака доступу, куди входить ідентифікатор безпеки користувача. Кожен процес, який створюється цим користувачем або запускається ним, має копію цієї ознаки. Вказана ознака використовується ОС, щоб підтвердити можливість доступу користувача до захищених об'єктів, або можливість виконання спеціальних функцій в системі і в захищених об'єктах.

З процесами пов'язаний і ряд блоків, в яких визначається віртуальний адресний простір, закріплений в даний момент за процесом. Процес не може безпосередньо змінювати ці структури, в цьому він залежить від менеджера віртуальної пам'яті, який надає сервіс для виділення пам'яті процесу.

До складу процесу також входить таблиця об'єктів, яка управляє відомими процесу об'єктами. Для кожного потоку, що входить в цей процес, є один дескриптор. Крім того, процес має доступ до файлових об'єктів і до розділів спільно використовуваної пам'яті.

Кожен процес W2K включає такі компоненти (рис. 6.7):

- один або декілька потоків;
- віртуальний адресний простір, відмінний від адресних просторів інших процесів, за винятком випадків явного розподілу пам'яті;
- один або більше сегментів коду, включаючи код DLL;
- один або більше сегментів даних, що містять глобальні змінні;
- рядки з інформацією про змінні оточення, такі як поточний шлях пошуку файлу тощо;
- пам'ять купи (динамічні структури даних) процесу;
- ресурси процесу (відкриті дескриптори, файли, інші купи).



Рисунок 6.7 – Структура процесу і його потоків Атрибути процесу і потоку в Windows наведені в таблиці 6.2.

Таблиця 6.2 – Атрибути процесу і потоку в Windows

Процес	Потік
Ідентифікатор процесу – унікальне значення, що ідентифікує процес в ОС	Ідентифікатор потоку – унікальне значення, що ідентифікує потік, коли він викликає сервіс.
Дескриптор захисту – описує, хто створив процес, права доступу тощо.	Контекст потоку – набір значень реєстрів, якими визначається стан виконуваного потоку.
Базовий пріоритет – базовий пріоритет процесу.	Динамічний пріоритет – пріоритет виконуваного потоку в даний момент. Базовий пріоритет – нижній пріоритет динамічного пріоритету потоку.
Спорідненість процесів за замовчанням – заданий за замовчанням набір процесів, де можливе виконання потоків.	Спорідненість процесів за потоком – множина процесів, де можливе виконання потоків.
Час виконання – сумарний час, витрачений на виконання усіх потоків у процесі.	Час виконання потоку – сукупний час, витрачений на виконання потоку в режимі користувача і режимі ядра.
Лічильник уведення/виведення – змінні, в які заносяться відомості про кількість і тип операцій введення/виведення, виконаних потоками процесу.	Статус сповіщення – це прапор, який вказує, чи слід потоку виконувати асинхронний виклик процедури.
Лічильник операцій з віртуальною пам'яттю – це змінні, в які заносяться відомості про кількість і тип операцій з віртуальною пам'яттю, виконаних потоками процесу	Лічильник призупинень – в ньому вказується, скільки разів виконання потоку було призупинене без подальшого відновлення.
Квоти – максимальна кількість сторінкової пам'яті і процесорного часу доступного процесу	Маркери режиму анонічного втілення – це часова ознака доступу.
Порти виключення/відладки – це канали обміну інформацією між процесами, в які диспетчер процесів повинен відправляти повідомлення при виникненні виняткових ситуацій.	Порт завершення – це канал обміну інформацією між процесами, куди диспетчер процесів відправляє повідомлення при завершенні потоку.
Статус виходу – причини завершення процесу	Статус виходу – причини завершення потоку

Усі потоки процесу спільно використовують код, глобальні змінні і ресурси процесу. Кожен потік планується незалежно.

Деякі атрибути потоку подібні до атрибутів процесів. Значення таких атрибутів потоку витягаються зі значень атрибутів процесу. Наприклад, у багатопроцесорній системі споріднені процесори за потоком – це декілька процесорів, на яких може

виконуватися цей потік. Вони співпадають з множиною процесорів споріднених по процесу або є його підмножиною. Інформація, що міститься в контексті потоку, дозволяє ОС призупиняти і поновлювати потоки. Основні функції управління процесами і потоками показані в таблиці. 6.3.

Таблиця 6.3 – Сервіси процесу і потоку в Windows

Процес	Потік
Створення процесу CreateProcess().	Створення потоку CreateThread().
Відкриття процесу OpenProcess().	Відкриття потоку OpenThread()
Інформація за запитом процесу	Інформація за запитом потоку
Інформація з наладки процесу	Інформація з наладки потоку
Поточний процес GetCurrentProcessID()	Поточний потік GetCurrentThreadID()
Припинення процесу ExitProcess()	Завершення потоку ExitThread()
	Отримання контексту
	Установка контексту
	Призупинення потоку Delay()
	Відновлення потоку
	Сповіщення потоків
	Перевірка сповіщення потоку

ОС Windows 2000 підтримує паралельне виконання процесів, оскільки потоки різних процесів можуть виконуватися одночасно. Більш того, декільком потокам одного і того ж процесу можуть бути виділені різні процесори, і ці потоки також можуть виконуватися одночасно. Потоки одного і того ж процесу можуть обмінюватися між собою інформацією за допомогою загального адресного простору і мають доступ до загальних ресурсів процесу.

6.7.2 Стани потоків в ОС Windows

При запуску додатка Win32 операційна система автоматично створює новий процес і перший потік процесу. Усі інші потоки можуть бути створені з існуючих потоків. Другий потік може бути створений з першого, третій з першого або другого тощо. Немає ніяких обмежень того, з якої точки програми створюється новий потік.

В ОС W2K потік у ході свого існування може мати один з **шести станів** (рис. 6.8) [10]. Життєвий цикл потоку починається в той момент, коли програма створює новий потік. Менеджер процесів виділяє пам'ять для об'єкту-потіку і звертається до ядра, щоб ініціалізувати об'єкт-потік ядра.

Готовність. При пошуку потоку на виконання диспетчер переглядає тільки потоки, що знаходяться в стані готовності, в яких є все для виконання, але бракує тільки процесора.

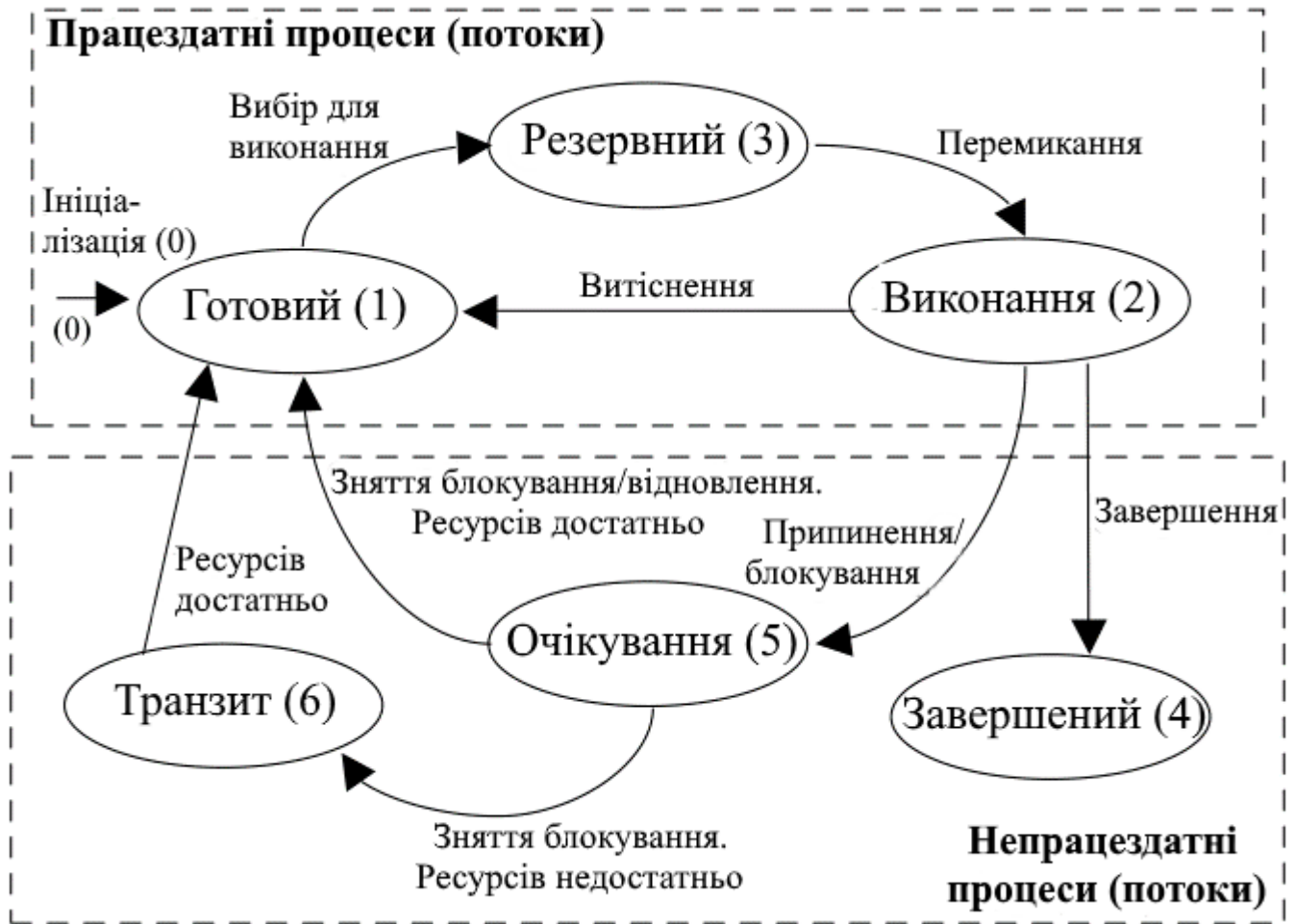


Рисунок 6.8 – Стан потоків в ОС Windows

Резервний (Першочергова готовність). Для кожного процесора системи вибирається один потік, який виконуватиметься наступним (найперший потік в черзі). Коли умови дозволяють, відбувається перемикання на контекст цього потоку.

Виконання. Як тільки відбувається перемикання контекстів, потік переходить у стан виконання і знаходиться в ньому до тих пір, поки або ядро не витіснить його через те, що з'явився пріоритетніший потік, або закінчився квант часу, виділений цьому потоку, або потік завершиться взагалі, або він за власною ініціативою перейде в стан очікування.

Очікування. Потік може входити в стан очікування декількома способами. Потік за своєю ініціативою чекає деякий об'єкт для того, щоб синхронізувати своє виконання. Операційна система (наприклад, підсистема введення- виведення) може

чекати в інтересах потоку. Підсистема оточення може безпосередньо змусити потік призупинити себе. Коли очікування потоку добіжить кінця, він повертається в стан готовності.

Транзит (Перехідний стан). Потік входить в перехідний стан, якщо він готовий до виконання, але ресурси, які йому потрібні, зайняті. Наприклад, сторінка, що містить стек потоку, може бути вивантажена з оперативної пам'яті на диск. При звільненні ресурсів потік переходить в стан готовності.

Завершення. Коли виконання потоку закінчилося, він входить в стан завершення. Знаходячись в цьому стані, потік може бути або видалений, або не видалений. Це залежить від алгоритму роботи менеджера об'єктів, відповідно до якого він і вирішує, коли видаляти об'єкт.

Виконання потоку **завершується**, коли функція потоку повертає управління. Це нормальний і рекомендований спосіб завершення роботи потоку. Проте в ряді випадків застосовуються інші методи. Потік може бути завершений функціями `ExitThread`, `TerminateThread` або функціями `ExitProcess`, `TerminateProcess`. Остання пара функцій закриває потік разом з процесом, якому він належить. Функції `ExitXXX` «правильніші». `TerminateThread` і `TerminateProcess` використовуються тільки в критичних обставинах, наприклад при завершенні потоків і процесів у разі помилки, після якої нормальне функціонування програми неможливе.

Windows 2000 може працювати в обчислювальній системі з симетричною багатопроцесорною архітектурою. У такій системі потоки можуть паралельно виконуватися на декількох процесорах.

ОС Windows 2000 підтримує симетричну багатопроцесорну конфігурацію апаратного забезпечення. Потоки будь-якого процесу, включаючи потоки виконавчої системи, можуть виконуватися на будь-якому процесорі. За умови відсутності обмежень на засіб процесорів мікроядро виділяє готовому до виконання потоку процесор, який звільняється першим. При цьому гарантується, що жоден процесор не простоюватиме або не виконуватиме потік з нижчим пріоритетом, якщо готовий процес з вищим пріоритетом.

У Windows 2000 потік має додатковий атрибут маску процесорів (affinity). Маска визначає підмножину мікропроцесорів, на яких може виконуватися потік. Цей атрибут задається функцією SetThreadAffinity.

```
DWORD SetThreadAffinityMask (  
HANDLE hThread, // посилання на потік  
DWORD dwThreadAffinityMask // маска affinity);
```

Маска може бути задана відразу для всіх потоків, які належать одному процесу. Для цього використовується функція SetProcessAffinityMask. Окрім маски для потоку можна програмно задати так званий ідеальний процесор (ideal processor). Ідеальний процесор – це процесор, який має перевагу для виконання потоку. Якщо в момент перемикання потоку вільні декілька процесорів, включаючи ідеальний, потік виконуватиметься на ідеальному процесорі. Ідеальний процесор призначається функцією SetThreadIdealProcessor.

```
DWORD SetThreadIdealProcessor(  
HANDLE hThread, // посилання на потік  
DWORD dwIdealProcessor); // номер ідеального процесора
```

Наведені вище операції допустимі тільки в Windows, починаючи з технології NT. Наприклад, ОС Windows 95 розрахована на роботу в системі з одним процесором.