

## 12.2 АЛГОРИТМИ ПЛАНУВАННЯ, ЗАСНОВАНІ НА КВАНТУВАННІ

В основі багатьох витісняючих алгоритмів планування лежить концепція квантування. Відповідно до цієї концепції кожному потоку по черзі для виконання надається обмежений безперервний період процесорного часу – квант.

**Квант** – це часовий інтервал, протягом якого процесу дозволено займати процесор, тобто дозволено перебувати в стані виконання. Поняття кванта ґрунтується на періоді таймера, який називається **тіком**. **Тік (tick)** – мінімально можливий часовий інтервал, який дорівнює дискретному проміжку часу між двома сигналами таймера, що генеруються через кожні 55 мс.

Квант дорівнює цілому числу тіків у проміжку від 1 до 255. Збільшення кванта уповільнює реакцію системи при збільшенні черги, а зменшення – збільшує частку накладних часових витрат на переключення процесів. Для ефективної роботи системи необхідно знаходити розумний компроміс.

Відповідно до алгоритмів, заснованих на квантуванні, зміна активного потоку відбувається, якщо:

- потік завершився і залишив систему;
- виникла помилка;
- потік перейшов в стан очікування;
- вичерпаний квант процесорного часу, відведений даному потоку.

Потік, який вичерпав свій квант, переводиться в стан готовність і очікує, коли йому буде надано новий квант процесорного часу, а на виконання відповідно до визначеного правила вибирається новий потік з черги готових. Таким чином, жоден процес не займає процесор надовго, і з цієї причини квантування широко використовується в системах розподілу часу. Граф станів потоку, зображений на рис. 12.3, відповідає алгоритму планування, заснованому на квантуванні.

Кванти, що виділяються потокам, можуть бути однаковими для всіх потоків або змінюватися в різні періоди життя процесу. Розглянемо випадок, коли всім потокам надаються кванти однакової довжини  $q$  (рис. 12.4).



Рисунок 12.3 – Граф станів потоку в системі з квантуванням

Якщо в системі є  $n$  потоків, то час, який потік проводить в очікуванні наступного кванта, можна грубо оцінити як  $q(n)$ . Чим більше потоків в системі, тим більше час очікування, тим менше можливості у кількох користувачів вести одночасну інтерактивну роботу.



Рисунок 12.4 – Ілюстрація розрахунку часу очікування в черзі

Але якщо величина кванта обрана дуже невеликою, то значення  $q(n)$  все одно буде досить мале для того, щоб користувач не відчував дискомфорту від присутності в системі інших користувачів. Типове значення кванта в системах поділу часу становить десятки мілісекунд.

Якщо квант короткий, то сумарний час, який проводить потік в очікуванні процесора, прямо пропорційній часу, необхідному для його виконання (тобто часу, який треба було б надати для виконання цього потоку при монопольному використанні обчислювальної системи).

Чим більший квант, тим вище ймовірність того, що потоки завершаться в результаті першого ж циклу виконання, і тим менше явною стає залежність часу очікування потоків від їх часу виконання. При досить великому кванті часу алгоритм квантування вироджується в алгоритм послідовної обробки, властивий однопрограми системам, при якому час очікування завдання в черзі взагалі ніяк не залежить від його тривалості.

Потоки отримують для виконання квант часу, але деякі з них використовують його не в повному обсязі, наприклад через необхідність виконати введення або виведення даних. У результаті виникає ситуація, коли потоки з інтенсивними зверненнями до введення-виведення використовують тільки невелику частину виділеного їм процесорного часу. Алгоритм планування може виправити цю «несправедливість».

В якості компенсації за повністю невикористані кванти потоки отримують привілеї при подальшому обслуговуванні. Для цього планувальник створює дві черги готових потоків (рис. 12.5).

Черга 1 утворена потоками, які прийшли в стан готовності в результаті вичерпання кванта часу, а черга 2 – потоками, в яких завершилася операція введення-виведення. При виборі потоку для виконання насамперед проглядається друга черга, і тільки якщо вона порожня, квант виділяється потоку з першої черги.

Багатозадачні ОС втрачають деяку кількість процесорного часу для виконання допоміжних робіт під час перемикання контекстів задач. При цьому запам'ятовуються і відновлюються регістри, прапори і покажчики стека. Витрати на ці допоміжні дії залежать від величини кванта часу, тому чим більший квант, тим менше сумарні накладні витрати, пов'язані з перемиканням потоків.

В алгоритмах, заснованих на квантуванні, яку б мету вони не переслідували (перевага коротких або довгих завдань, компенсація недовикористаного кванта), не використовується ніякої попередньої інформації про завдання. При надходженні

завдання на обробку ОС не має ніяких відомостей про те, чи є воно коротким, чи довгим, наскільки інтенсивними будуть його запити до пристроїв введення-виведення тощо.

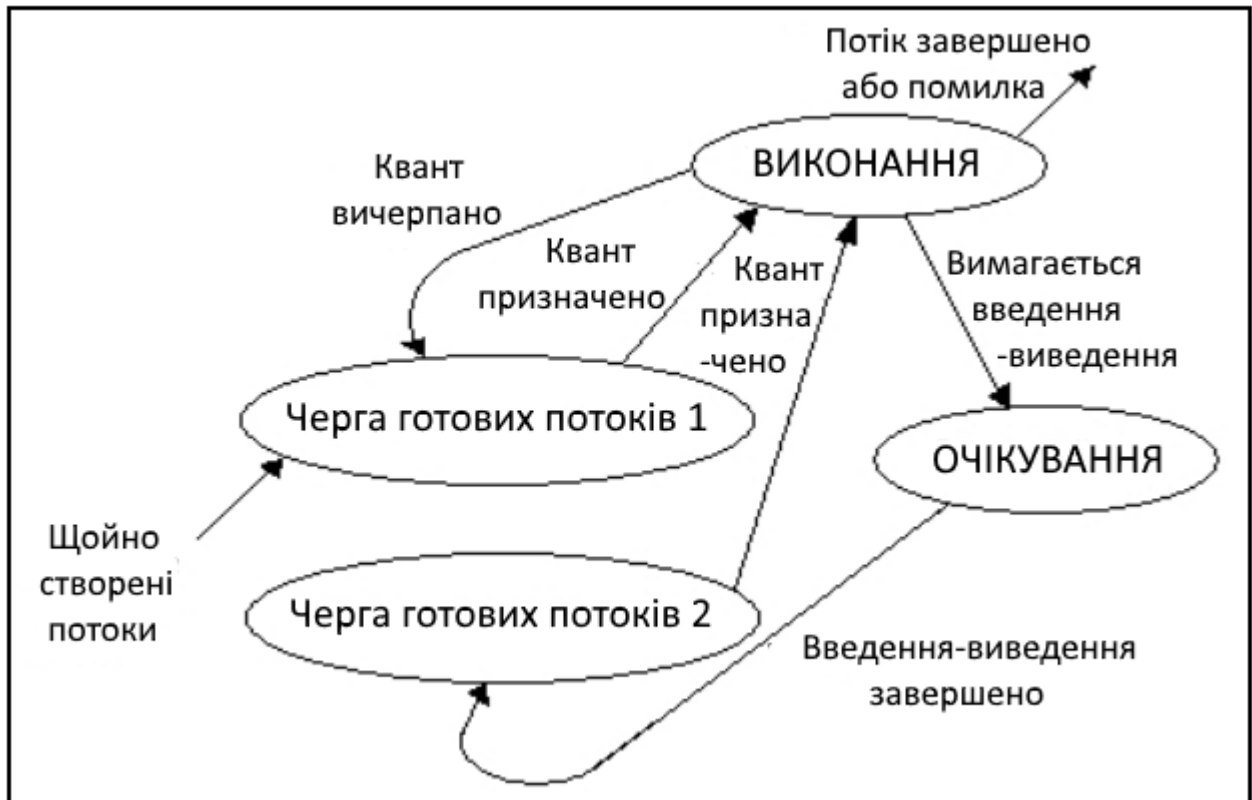


Рисунок 12.5 – Квантування з перевагою потоків, які інтенсивно звертаються до пристроїв введення-виведення