

6.2 МОДЕЛЬ ПОТОКУ

У багатопотоковій системі при створенні процесу ОС створює для кожного процесу адресний простір і як мінімум один потік виконання. Фактично це майже визначення процесу. При створенні потоку так само, як і при створенні процесу, операційна система генерує спеціальну інформаційну структуру – описувач потоку, який містить ідентифікатор потоку, дані про права доступу і пріоритет, стан потоку і іншу інформацію. У початковому стані потік (чи процес, якщо йдеться про систему, в якій поняття «потік» не визначається) знаходиться в призупиненому стані. Момент вибірки потоку на виконання здійснюється відповідно до прийнятого в цій системі правила надання процесорного часу і з урахуванням усіх існуючих на даний момент потоків і процесів. У випадку, якщо коди і дані процесу знаходяться в області підкачування, необхідною умовою активізації потоку процесу є також наявність місця в оперативній пам'яті для завантаження його виконуваного модуля.

Коли говорять про процеси, то тим самим хочуть відмітити, що у кожного процесу є свій адресний простір, що містить текст програми і дані, кожному процесу призначаються свої ресурси. Іншими словами, у разі процесів ОС вважає їх абсолютно незв'язаними і незалежними. При цьому ОС бере на себе роль арбітра в конкуренції між процесами з приводу ресурсів.

Концепція потоку додає до моделі процесу можливість одночасного виконання в одному і тому ж середовищі процесу декількох програм, достатньою мірою незалежних, Декілька потоків, працюючих паралельно в одному процесі, аналогічні декільком процесам, що йдуть паралельно на одному комп'ютері. Потоки мають деякі властивості процесів, тому їх іноді називають спрощеними процесами. Термін багатопоточність також використовують для опису використання декількох потоків в одному процесі.

На рис. 6.1, а представлені три звичайні процеси, у кожного з яких є власний адресний простір і один потік управління. На рис. 6.1, б представлено один процес з трьома потоками управління.

При запуску багатопотокового процесу в системі з одним процесором потоки працюють по черзі. Ілюзія паралельної роботи декількох різних послідовних процесів

створюється шляхом постійного перемикання системи між процесами. Багатопоточність реалізується приблизно так же.

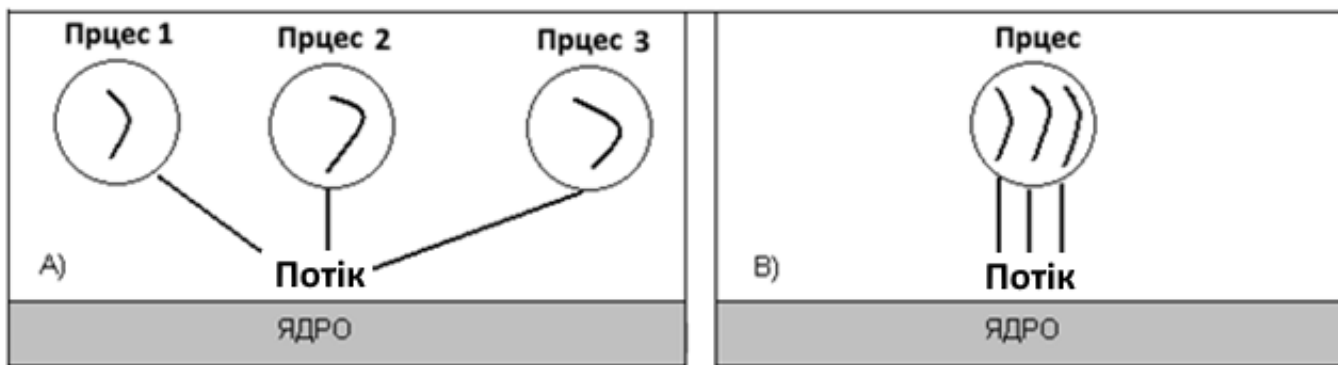


Рисунок 6.1 – Три процеси з поодинокими потоками (А);
один процес з трьома потоками (В)

Процесор швидко перемикається між потоками, створюючи враження паралельної роботи потоків. Різні потоки в одному процесі не так незалежні, як різні процеси. В усіх потоків один і той же адресний простір, що означає спільне використання глобальних змінних. Оскільки будь-який потік має доступ до будь-якої адреси елементу пам'яті процесу, один потік може прочитувати, записувати інформацію в стек іншого потоку. Захисту не існує, оскільки це неможливо і не потрібно. Як показано в табл. 6.1. потоки розділяють не лише адресний простір, але й відкриті файли, дочірні процеси тощо.

У 1-у стовбці перераховані елементи, спільно використовувані усіма потоками процесу, а в 2-у – елементи індивідуальні для кожного потоку.

Перший стовбець містить елементи, що є властивостями процесу, а не потоку. Наприклад, якщо процес відкриває файл, цей файл тут же стає видимим для потоків, і вони можуть прочитувати і записувати інформацію у файл. Це логічно, оскільки процес, а не потік є одиницею управління ресурсами. Якби у кожного потоку був власний адресний простір і інші ресурси, то це були б окремі процеси.

Легковаговими ці задачі називають ще тому, що ОС не повинна для них організовувати повноцінну віртуальну машину. Ці задачі не мають своїх власних ресурсів, вони розвиваються в тому ж адресному просторі, можуть користуватися тими ж файлами, віртуальними пристроями і іншими ресурсами, що і цей процес. Єдине, що їм необхідно мати, – це процесорний ресурс. В однопроцесорній системі

потоки розділяють між собою процесорний час так само, як це роблять звичайні процеси, а в мультипроцесорній системі можуть виконуватися одночасно.

Таблиця 6.1 – Елементи процесів и потоків

Елементи процесу	Елементи потоків
Адресний простір	Лічильник команд
Глобальні змінні	Регістри
Відкриті файли	Стек
Дочірні процеси	Стан
Необроблені аварійні сигнали	
Сигнали і їх обробники	
Інформація про використання ресурсів	

Слід врахувати, що кожен потік має свій власний стек, що і показано на рис. 6.2. Стек кожного потоку містить по одному фрейму для кожної вже викликаної, але такої, що ще не повернула управління, процедури. Такий фрейм містить локальні змінні процедури і адресу повернення управління після закінчення її виклику. Наприклад, якщо процедура X викликає процедуру Y, а Y викликає процедуру Z, то при виконанні Z в стеку будуть фрейми для X, Y і Z. Кожен потік буде, як правило, викликати різні процедури і, отже, мати середовище виконання, що відрізняється від інших. Тому кожному потоку потрібний свій власний стек.

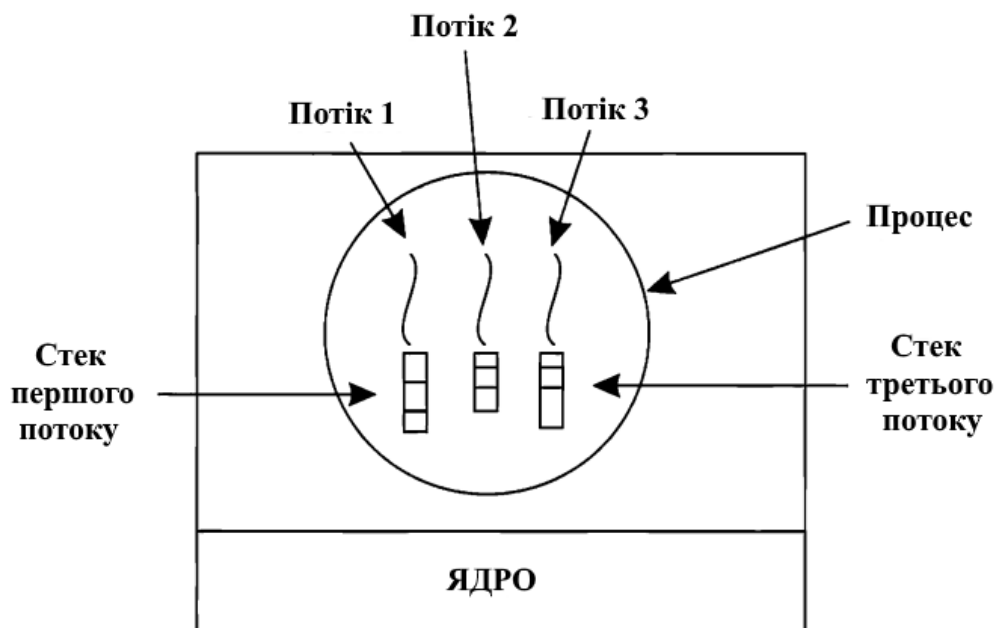


Рисунок 6.2 – Для кожного потоку є свій власний стек