

5.7 ОПЕРАЦІЇ НАД ПРОЦЕСАМИ

5.7.1 Набір операцій

Процес сам не може перейти з одного стану в інший. Зміною стану процесів займається операційна система, здійснюючи операції над ними. Кількість таких операцій в нашій моделі співпадає з кількістю стрілок на діаграмі станів. Їх зручно об'єднати в три пари:

1. Створення процесу – завершення процесу.
2. Призупинення процесу (перехід із стану виконання в стан готовність) – запуск процесу (перехід із стану готовність в стан виконання).
3. Блокування процесу (перехід із стану виконання в стан очікування) – розблокування процесу (перехід із стану очікування в стан готовність).

Надалі, коли ми вивчатимемо алгоритми планування, в нашій моделі з'явиться ще одна операція, що не має пари: зміна пріоритету процесу.

Операції створення і завершення процесу є одноразовими, оскільки застосовуються до процесу не більше одного разу (деякі системні процеси ніколи не завершуються при роботі обчислювальної системи). Усі інші операції, пов'язані зі зміною стану процесів, будь то запуск або блокування, як правило, є багаторазовими. Розглянемо детальніше, як операційна система виконує операції над процесами.

5.7.2 Одноразові операції

Складний життєвий шлях процесу в комп'ютері розпочинається з його народження. Будь-яка операційна система, що підтримує концепцію процесів, повинна мати засоби для їх створення. У дуже простих системах (наприклад, в системах, спроектованих для роботи тільки для одного конкретного застосування) усі процеси можуть бути породжені на етапі старту системи. Складніші ОС створюють процеси динамічно, в міру необхідності. Ініціатором народження нового процесу після старту операційної системи може виступити або процес користувача, що вчинив спеціальний системний виклик, або сама операційна система, тобто, зрештою, теж деякий процес. Процес, що ініціював створення нового процесу, прийнято називати процесом-батьком

(parent process), а знову створений процес – процесом-дитиною (child process). Процеси-діти можуть, у свою чергу, породжувати нові процеси тощо.

При народженні процесу система заводить новий PCB із станом процесу народження і починає його заповнення. Новий процес отримує свій власний унікальний ідентифікаційний номер.

Для виконання своїх функцій процес-дитина вимагає певних ресурсів: пам'яті, файлів, пристроїв введення-виведення тощо. Існує два підходи до їх виділення. Новий процес може отримати у своє користування деяку частину батьківських ресурсів, можливо, розділяючи з процесом-батьком і іншими процесами-дітьми права на них, або може отримати свої ресурси безпосередньо від операційної системи. Інформація про виділені ресурси заноситься в PCB.

Після наділу процесу-дитини ресурсами необхідно занести в його адресний простір програмний код, значення даних, встановити програмний лічильник. Тут також можливі два рішення. У першому випадку процес-дитина стає дублікатом процесу-батька за реєстровим і призначеним для користувача контекстам. При цьому повинен існувати спосіб визначення хто для кого з процесів-двійників є батьком. У другому випадку процес-дитина завантажується новою програмою з якого-небудь файлу.

Операційна система UNIX дозволяє породження процесу тільки першим способом; для запуску нової програми необхідно спочатку створити копію процесу-батька, а потім процес-дитина повинна замінити свій призначений для користувача контекст за допомогою спеціального системного виклику. Операційні системи VAX/VMS і WINDOWS NT допускають тільки друге рішення.

Породження нового процесу як дублікату процесу-батька призводить до можливості існування програм (тобто виконуваних файлів), для роботи яких організовується більше чим один процес. Можливість заміни призначеного для користувача контексту процесу по ходу його роботи (тобто, завантаження для виконання нової програми) призводить до того, що в рамках одного і того ж процесу можуть бути послідовно виконані декілька різних програм.

Після того як процес наділений змістом, в PCB дописується інформація, що залишилася, і стан нового процесу змінюється на готовність.

Після завершення процесу, операційна система переводить його в стан закінчив виконання і звільняє усі асоційовані з ним ресурси, роблячи відповідні записи в блоці управління процесом. При цьому сам PCB не знищується, а залишається в системі ще деякий час. Це пов'язано з тим, що процес-батько після завершення процесу-дитини може запросити операційну систему про причину завершення процесу і/або статистичну інформацію про його роботу. Подібна інформація зберігається в PCB «мертвого» процесу до запиту процесу-батька або до кінця його діяльності, після чого всі сліди процесу остаточно зникають з системи. В операційній системі UNIX процеси, що знаходяться в стані закінчив виконання, прийнято називати процесами зомбі.

5.7.3 Багаторазові операції

Одноразові операції призводять до зміни кількості процесів, що знаходяться під управлінням операційної системи, і завжди пов'язані з виділенням або звільненням певних ресурсів. Багаторазові операції, навпаки, не призводять до зміни кількості процесів в операційній системі і не пов'язані з виділенням або звільненням ресурсів.

Запуск процесу. З числа процесів, що знаходяться в стані готовність, операційна система вибирає один процес для подальшого виконання. Критерії і алгоритми такого вибору будуть детально розглянуті в розділі «Планування процесів». Для обраного процесу операційна система забезпечує наявність в оперативній пам'яті інформації, необхідної для його подальшого виконання. Те, як вона це робить, буде в деталях описано в розділі «Управління пам'яттю». Далі стан процесу змінюється на виконання, відновлюються значення регістрів для цього процесу, і управління передається команді, на яку вказує лічильник команд процесу. Всі дані, необхідні для цього відновлення контексту, витягаються з PCB процесу, над яким здійснюється операція.

Призупинення процесу. Робота процесу, що знаходиться в стані виконання, призупиняється в результаті якого-небудь переривання. Процесор автоматично зберігає лічильник команд і, можливо, один або декілька регістрів у стеку виконуваного процесу і передає управління за спеціальною адресою обробки цього переривання. На цьому діяльність апаратної частини з обробки переривання завершується. За вказаною адресою розташовується одна з частин ОС. Вона зберігає

динамічну частину системного і реєстрового контекстів процесу в його РСВ, переводить процес в стан готовність і приступає до обробки переривання.

Блокування процесу. Процес блокується, коли він не може продовжувати свою роботу, не дочекавшись виникнення якої-небудь події в обчислювальній системі. Для цього він звертається до операційної системи за допомогою певного системного виклику. Операційна система обробляє системний виклик (ініціалізує операцію введення-виведення, додає процес в чергу процесів, що чекають звільнення пристрою або виникнення події тощо). ОС зберігає необхідну частину контексту процесу в його РСВ і переводить процес із стану виконання в стан очікування. Детальніше ця операція розглядатиметься в розділі «Управління введенням-виведенням».

Розблокування процесу. Після виникнення в системі якої-небудь події, ОС треба точно визначити яка саме подія сталося. Потім операційна система перевіряє: чи знаходився деякий процес у стані очікування для цієї події і, якщо знаходився, переводить його в стан готовність, виконуючи необхідні дії, пов'язані з настанням події (ініціалізація операції введення-виведення для чергового очікуючого процесу тощо). Ця операція, як і операція блокування, буде детальніше описана в розділі «Управління введенням-виведенням».

5.7.4 Перемикання контексту

Досі ми розглядали операції над процесами ізольовано, незалежно одна від одної. Насправді ж діяльність мультипрограмної операційної системи складається з ланцюжків операцій, що виконуються над різними процесами і супроводжуються перемиканням процесора з одного процесу на інший. Для прикладу спрощено розглянемо, як в реальності може виникати операція розблокування процесу, очікуючого введення-виведення (рис. 5.8).

При виконанні процесором деякого процесу (на рисунку – процес 1) виникає переривання від пристрою введення-виведення, який сигналізує про закінчення операцій на пристрої. Над процесом, що виконується, робиться операція призупинення. Далі, операційна система розблоковує процес, що ініціював запит на введення-виведення (на рис. 5.8 – процес 2), і здійснює запуск призупиненого або

нового процесу, вибраного при виконанні планування (на рис. 5.8 був вибраний розблокований процес).

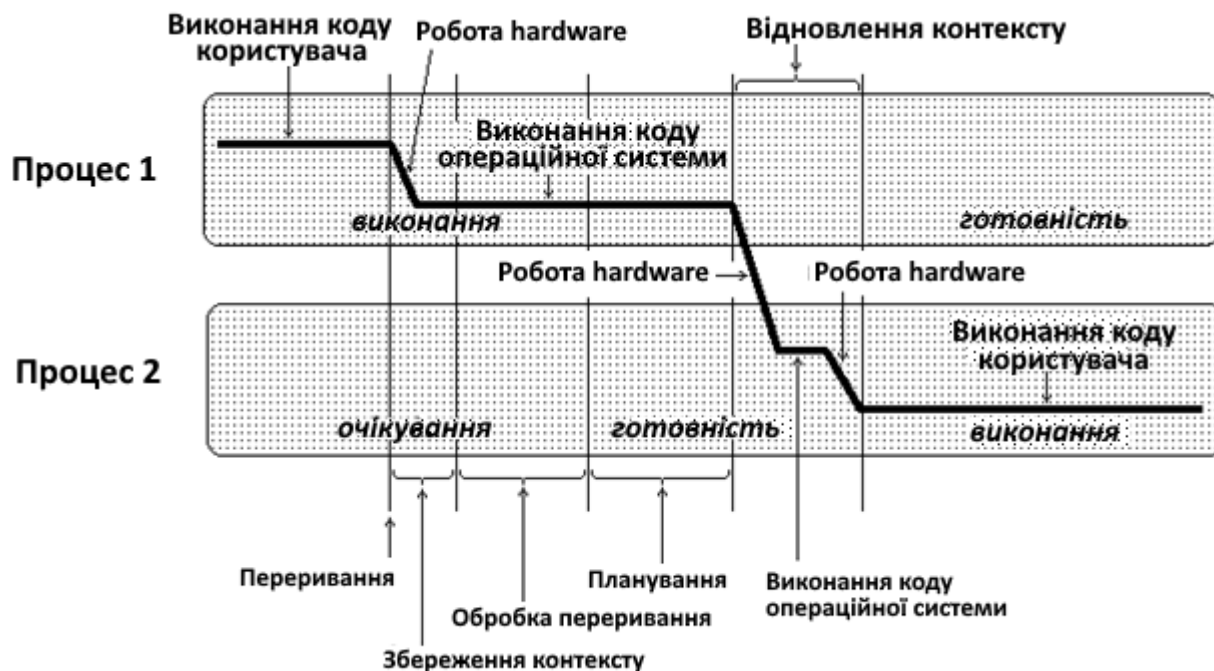


Рисунок 5.8 – Виконання операції розблокування процесу

Для коректного перемикавання процесора з одного процесу на інший необхідно зберегти контекст процесу, що виконувався, і відновити контекст процесу, на який буде перемкнутий процесор. Така процедура збереження/відновлення працездатності процесів називається перемиканням контексту. Час, витрачений на перемикання контексту, не використовується обчислювальною системою для здійснення корисної роботи і є накладними витратами, що знижують продуктивність системи. Він змінюється від машини до машини і зазвичай знаходиться в діапазоні від 1 до 1000 мікросекунд.