

11.1 СТРАТЕГІЇ УПРАВЛІННЯ СТОРІНКОВОЮ ПАМ'ЯТТЮ

Одна з основних задач ОС – управління віртуальною пам'яттю. При виборі стратегії розв'язання цієї задачі ключовим питанням стає продуктивність: вимагається скоротити кількість переривань через відсутність сторінки в основній пам'яті, оскільки їх обробка призводить до істотних накладних витрат. Крім того, ОС повинна активізувати готовий до роботи процес на час виконання повільних операцій введення-виведення.

Що ж відбувається, коли потрібної сторінки в пам'яті немає або операція звернення до пам'яті недопустима? Природно, що операційна система має бути якось сповіщена про те, що сталося. Для цього використовується механізм виняткових ситуацій. При спробі виконати подібне звернення до віртуальної сторінки виникає виняткова ситуація «**сторінкове порушення**» (**page fault**), що призводить до виклику спеціальної послідовності команд для обробки конкретного виду сторінкового порушення.

Сторінкове порушення може відбуватися в найрізноманітніших випадках: за відсутності сторінки в оперативній пам'яті, при спробі запису в сторінку з атрибутом «тільки читання» або при спробі читання або запису сторінки з атрибутом «тільки виконання». У будь-якому з цих випадків викликається обробник сторінкового порушення, що є частиною операційної системи. Йому передається причина виникнення виняткової ситуації і віртуальна адреса, звернення до якої викликало порушення.

Нас цікавитиме конкретний варіант сторінкового порушення – звернення до відсутньої сторінки, оскільки саме його обробка багато в чому визначає продуктивність сторінкової системи. Коли програма звертається до віртуальної сторінки, відсутньої в основній пам'яті, операційна система повинна виділити сторінку основної пам'яті, перемістити в неї копію віртуальної сторінки із зовнішньої пам'яті і модифікувати відповідний елемент таблиці сторінок.

Підвищення продуктивності обчислювальної системи може бути досягнуте за рахунок зменшення частоти сторінкових порушень, а також за рахунок збільшення

швидкості їх обробки. Час ефективного доступу до відсутньої в оперативній пам'яті сторінки складається з:

1. Обслуговування виняткової ситуації (page fault).

2. Читання (підкачування) сторінки з вторинної пам'яті. Іноді, при нестачі місця в основній пам'яті, необхідно виштовхнути одну із сторінок з основної пам'яті у вторинну, тобто здійснити заміщення сторінки.

3. Відновлення виконання процесу, що викликав даний page fault.

Для розв'язання першої і третьої задач ОС виконує до декількох сотень машинних інструкцій впродовж декількох десятків мікросекунд. Час підкачування сторінки близький до декількох десятків мілісекунд. Проведені дослідження показують, що ймовірності page fault 5×10^{-7} виявляється достатньо, щоб понизити продуктивність сторінкової схеми управління пам'яттю на 10%.

Таким чином, зменшення частоти page faults є одним з ключових задач системи управління пам'яттю. Його розв'язання пов'язане з правильним вибором алгоритму заміщення сторінок. Тому програмне забезпечення управління пам'яттю ОС має бути пов'язане з реалізацією нижченаведених стратегій.

Стратегія вибірки визначає, в який момент слід переписати сторінку з вторинної пам'яті в первинну. Існує два основні варіанти вибірки – за запитом і з упередженням. Алгоритм вибірки за запитом вступає в дію в той момент, коли процес звертається до відсутньої сторінки, вміст якої знаходиться на диску. Його реалізація полягає в завантаженні сторінки з диска у вільну фізичну сторінку і корекції відповідного запису таблиці сторінок.

Алгоритм вибірки з упередженням здійснює випереджаюче читання, тобто окрім сторінки, що викликала виняткову ситуацію, в пам'ять завантажуються декілька сторінок, що її оточують (зазвичай сусідні сторінки розташовуються в зовнішній пам'яті послідовно і можуть бути зчитані за одне звернення до диска). Такий алгоритм покликаний зменшити накладні витрати, пов'язані з великою кількістю виняткових ситуацій, що виникають при роботі зі значними об'ємами даних або коду. Крім того, оптимізується робота з диском.

Стратегія розміщення визначає, в яку ділянку первинної пам'яті помістити сторінку, що поступає. У системах із сторінковою організацією все просто – у будь-

який вільний сторінковий кадр. У разі систем з сегментною організацією потрібна стратегія, аналогічна стратегії з динамічним розподілом.

Стратегія заміщення визначає, яку сторінку треба виштовхнути в зовнішню пам'ять, щоб звільнити місце в оперативній пам'яті. Розумна стратегія заміщення, реалізована у відповідному алгоритмі заміщення сторінок, дозволяє зберігати в пам'яті найнеобхіднішу інформацію і тим самим понизити частоту сторінкових порушень. Заміщення повинне відбуватися з урахуванням виділеної кожному процесу кількості кадрів. Крім того, треба вирішити, чи повинна сторінка, що заміщається, належати процесу, який ініціював заміщення, чи вона має бути вибрана серед усіх кадрів основної пам'яті.