SOI – Semafory

Koncepcja

1. Zadanie

Mamy bufor FIFO na liczby całkowite. * Procesy A1 generują kolejne liczby parzyste modulo 50, jeżeli w buforze jest mniej niż 10 liczb parzystych. * Procesy A2 generują kolejne liczby nieparzyste modulo 50, jeżeli liczb parzystych w buforze jest więcej niż nieparzystych. * Procesy B1 zjadają liczby parzyste pod warunkiem, że bufor zawiera co najmniej 3 liczby. * Procesy B2 zjadają liczby nieparzyste, pod warunkiem, że bufor zawiera co najmniej 7 liczb. W systemie może być dowolna liczba procesów każdego z typów. Zrealizuj wyżej wymienioną funkcjonalność przy pomocy semaforów. Zakładamy, że bufor FIFO poza standardowym put() i get() ma tylko metodę umożliwiającą sprawdzenie liczby na wyjściu (bez wyjmowania) oraz posiada metody zliczające elementy parzyste i nieparzyste. Zakładamy, że semafory mają tylko operacje P i V.

2. Schemat Ogólny

W zadaniu mamy cztery rodzaje procesów (A1, A2, B1, B2) działających na wspólnym buforze FIFO. Kluczowym jest użycie semaforów do synchronizacji tych procesów, z zachowaniem określonych warunków.

3. Struktury i Semafory

- Bufor FIFO:

    a. std::vector<int> buffer: Wektor do przechowywania danych.

    b. size_t capacity: Maksymalny rozmiar bufora.

    c. Semaphore mutex: Binarny semafor służący jako blokada do ochrony sekcji krytycznej (dostęp do bufora).

    d. Semaphore empty: Semafor do sygnalizowania, że bufor jest pusty. Jego początkowa wartość to capacity.

    e. Semaphore full: Semafor do sygnalizowania, że bufor jest pełny. Jego początkowa wartość to 0.
    f. Semaphore semEvenLessThan10: używany przy procesie A1
    g. Semaphore semMoreEvans: używany przy procesie A2
    h. Semaphore semAtLeastThree: używany przy procesie B1
    i. Semaphore semAtLeastSeven: używany przy procesie B2
    j. Semaphore semFrontEven: używany przy B1
    k. Semaphore semFrontOdd: używany przy B2

Bufor ma również liczniki evenCount i oddCount do śledzenia liczby parzystych i nieparzystych elementów. Metody put, getB1 i getB2 używają semaforów full, empty, i mutex do koordynacji dostępu do bufora, zapewniając, że operacje są bezpieczne pod względem współbieżności. Dodatkowo put w zależności od tego czy dostanie liczbę parzystą czy nie korzysta z semEvenLessThan10 lub semEvenLessThan10. GetB1 korzysta z semafora semAtLeastThree oraz semGrontEven, a getB2 z sumAtLeastSeven, semFrontOdd.

4. Testowanie

Poprzez printowanie zawartości bufora i obserwowanie czy bufor zachowuje się jak należy.

5. Wstępna implementacja

Wstępna implementacja bufora znajduje się w fifo.cpp. Implementacja wymaga popraw.