

[SOI] - Koncepcja lab 4 (t4)

Treść zadania

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 monitorów. 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 semaforey mają tylko operacje P i V.

Założenia

1. Podział procesów:

- **A1** - generowanie liczb parzystych modulo 50, jeżeli < 10 liczb parzystych
- **A2** - generowanie liczb nieparzystych modulo 50, jeżeli parzystych $>$ nieparzystych
- **B1** - zjadają liczby parzyste, jeżeli ≥ 3 liczby
- **B2** - zjadają liczby nieparzyste, jeżeli ≥ 7 liczb.

2. bufor FIFO ma cztery metody:

- wkładamy liczbę
- bierzemy liczbę
- sprawdzamy liczby na wyjściu
- zliczamy elementy parzyste i nieparzyste

Implementacja

1. Implementacja w języku C/C++

2. Tworzymy jeden główny monitor, który będzie zaimplementowany na podstawie pliku `monitor.h`. Będzie on posiadał odpowiednie metody.

3. Zostanie utworzony bufor z metodami opisanymi w [założeniach](#)

4. Zostaną zaimplementowane metody `canProdEven`, `canProdOdd`, `canConsEven`, `canConsOdd` posiadające implementacje opisane w [założeniach](#)

5. Zostaną zaimplementowane metody `prodEven`, `prodOdd`, `consEven`, `consOdd`. Każda z nich będzie odwoływała się do głównego monitora (wejście do sekcji krytycznej, sprawdzanie warunku i oczekiwanie, sprawdzeniu innych metod i wyjście z sekcji krytycznej).

Testowanie

Testem będzie odpowiednie spowolnienie działania programu i wyświetlanie użytkownikowi aktualnie dodanej/usuniętej liczby oraz stanu bufora po każdorazowej zmianie stanu, po to aby sprawdzić, czy

wykonywane są poprawne operacje.

Scenariusze testowe

1. sam producent parzysty
2. sam producent nieparzysty
3. sami konsumenci parzyści
4. sami konsumenci nieparzyści
5. sami producenci
6. sami konsumenci
7. normalny test, ze wszystkimi
8. zakleszczenie

Autor

Imię i Nazwisko: Mateusz Brzozowski

Nr. Indeksu: XXXXXXXX