

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 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 semafony mają tylko operacje P i V.

Koncepcja

Zakładam implementację w języku C++ z wykorzystaniem pięciu semaforów. Cztery z nich będą odpowiedzialne za obsługę warunków pobierania i generowania liczb (semafor będzie się odblokowywał, jeśli dany warunek będzie spełniony), a piąty będzie pilnował aby do bufora miał dostęp tylko jeden wątek.

Takie rozwiązanie powinno zapewnić odpowiednią synchronizację. Należy jednak jeszcze przemyśleć kwestię wyścigów, które mogą wystąpić, jeśli przy tym samym stanie bufora zostaną spełnione dwa warunki. Wtedy trzeba będzie wybrać jeden z semaforów, który odblokujemy.

Klasa bufora będzie zawierać listę liczb całkowitych, licznik parzystych i nieparzystych oraz standardowe metody opisane w treści zadania. Dodatkowo zaimplementuję metody sprawdzające postawione warunki i odblokowujące odpowiednie semafony.

Napiszę 4 funkcje, które będą wykonywały akcje zgodne z założeniami z treści zadania (generowanie, zjadanie określonych liczb)

1. Zablokuj swój semafor
2. Zablokuj semafor ogólny
3. Wczytaj / zapisz
4. Odblokuj nowy semafor (zgodnie ze stanem bufora)
5. Odblokuj semafor ogólny

Każda funkcja będzie wykonywana w osobnym wątku / wątkach (zgodnie z założeniem zadania).

Testowanie

Zakładam testowanie przez wywołanie po kilka testowych wątków każdej grupy (A1, A2, B1, B2), które będą próbować wykonywać akcje na buforze.

Dodam również każdorazowe sprawdzanie warunków podczas próby aktualizacji bufora przez proces. Jeśli proces dostanie dostęp, podczas gdy nie powinien go mieć wtedy program nas o tym poinformuje. Tak samo podczas próby pobierania z pustego bufora.

Dodatkowo, po każdej operacji na buforze będę go wypisywał na standardowe wyjście i opóźniał program, aby sprawdzić, czy jego zawartość zgadza się z tym, co zostało założone. Na wyjście wypisywana będzie również informacja, o grupie procesu, kiedy ten wykona akcję.