

Implementacje monitorów

Monitor jest strukturalnym narzędziem synchronizacji wątków. W językach obiektowych (c++,java) implementuje się go jako klasę. Wszystkie metody monitora muszą być wykonywane z zachowaniem wzajemnego wykluczania. Oznacza to, że w klasie reprezentującej monitor musi istnieć jeden klucz dostępu do wszystkich metod. Korzystając z biblioteki pthread można użyć w tym celu wspólnego mutexu dla ochrony wszystkich metod.

ćwiczenie 1

- a) Zapisać w pseudokodzie implementację semafora przy pomocy monitora. Monitor powinien implementować metody wait i signal
- b) Zaimplementować ten monitor w języku C wykorzystując funkcje z biblioteki pthread. Sprawdzić jego działanie na przykładzie programu p1.c.
- c) Zaimplementować ten monitor w języku Java. Sprawdzić jego działanie na przykładzie programu p1.java.

ćwiczenie 2

Bariera jest narzędziem, które służy do wstrzymania grupy wątków w określonym miejscu programu współbieżnego. Wątki zostają wstrzymane w momencie wywołania operacji *wait* na barierze do czasu, aż liczba wstrzymanych wątków nie osiągnie pewnej ustalonej wartości progowej. Wartość progowa jest parametrem charakterystycznym dla danej bariery. Dzięki temu możemy spowodować, że wszystkie wątki zaczną wykonywać swoje zadania w tym samym momencie.

- a) Zapisać w pseudokodzie implementację bariery przy pomocy monitora. Monitor powinien implementować metodę czeka. Wątek wywołujący metodę czeka ma zostać wstrzymany w przypadku, gdy liczba wstrzymanych wątków będzie mniejsza od pewnej wartości progowej zdefiniowanej jako próg bariery. Gdy liczba wątków wywołujących metodę czeka osiągnie próg bariery wtedy ostatni z tych wątków nie będzie wstrzymany tylko spowoduje wznowienie pozostałych wątków.
- b) Zaimplementować ten monitor w języku C wykorzystując funkcje z biblioteki pthread. Sprawdzić jego działanie na przykładzie programu p2.c.

Zadania dodatkowe

1. Napisać w języku C++ implementację monitora producent-konsument na podstawie pseudokodu w pliku „wykład3.pdf” wykorzystując narzędzia synchronizacji z biblioteki pthread.
2. Napisać w języku C++ implementację monitora czytelnicy-pisarze na podstawie pseudokodu w pliku „wykład3.pdf” wykorzystując narzędzia synchronizacji z biblioteki pthread.