

Programowanie współbieżne i rozproszone – laboratorium

Instrukcja do ćwiczenia laboratoryjnego VI

Zmienne warunkowe i bariery

Autor: S. Samolej

Rzeszów, 2013

Podziękowania:

Składam podziękowania panu dr inż. Jędrzejowi Ułasiewiczowi za udostępnienie materiałów wykładowych i laboratoryjnych, które były inspiracją do opracowania tych materiałów.

1. Proszę skompilować, uruchomić i przeanalizować przykładowy program „cond_var1.c”. W programie jeden z wątków oczekuje na przekroczenie przez zmienną count wartości 12, podczas gdy dwa inne wątki zwiększają jej wartość. W chwili, gdy zmienna count przekracza wartość 12 wątek, który to wykrywa wysyła sygnał do zablokowanego wątku. Blokada wątku odbywa się z zastosowaniem zmiennej warunkowej.
2. Proszę skompilować, uruchomić i przeanalizować przykładowy program „bariera.c”. W programie zastosowano barierę blokującą wykonywanie wątków, dopóki dwa z nich nie zgłoszą chęci oczekiwania na niej.
3. Proszę skompilować, uruchomić i przeanalizować przykładowy program „spinlock.c”. W programie wykonują się dwa wątki. Sekcje krytyczne wątków objęte są wirującymi blokadami. Jeden z nich stara się przejąć sekcję i wykonać obliczenia, podczas gdy drugi sprawdza, czy sekcja krytyczna może być wykonywana i jeśli nie to kontynuuje swoje obliczenia.
4. Proszę zaproponować rozwiązanie następującego problemu producenta-konsumenta:
Pojedynczy producent modyfikuje pojedynczą zmienną dzieloną i po jej modyfikacji powiadamia konsumenta/konsumentów, że nastąpiła modyfikacja (powiadomienie odbywa się przez modyfikację pojedynczej flagi). Konsument pobiera dane, tylko wtedy, kiedy otrzyma powiadomienie. Do rozwiązania problemu należy zastosować zmienną warunkową.
Uwaga: jeśli planowana jest większa liczba konsumentów warto rozważyć zastosowanie funkcji `pthread_cond_broadcast()` do odblokowania odczytu wszystkich konsumentów.