

Programowanie równoległe
Laboratorium 6.

Cel:

- doskonalenie umiejętności programowania z wykorzystaniem biblioteki wątków „pthread”
- zaimplementowanie funkcji realizującej konstrukcję „bariery”, przydatnej w synchronizacji procesów równoległych

Kroki:

1. Utworzenie katalogu roboczego (np. lab_5).
2. Na podstawie materiałów z wykładu utworzenie i zaimplementowanie algorytmu realizującego funkcję bariery: dowolny wątek może zakończyć realizację funkcji „bariera”, dopiero po wywołaniu tej funkcji przez pozostałe wątki. Możliwy sposób rozwiązania:
 - a) utworzenie zmiennej globalnej zliczającej liczbę wątków, które wywołały funkcję „bariera”
 - b) związanie z tą zmienną zmiennej warunku i odpowiedniego muteksa
 - c) opracowanie mechanizmu funkcjonowania bariery z wykorzystaniem powyższych zmiennych
3. Wpisanie funkcji „bariera” do osobnego pliku źródłowego
4. Uzupełnienie programu z laboratorium („całka”) o wykorzystanie funkcji bariery: każdy wątek wypisuje poprawną wartość obliczonej całki.
5. Opracowanie pliku Makefile do kompilacji
6. Kompilacja, testowanie
7. Uzupełnienie programu z laboratorium („pub”) o wykorzystanie funkcji bariery: wątki nie wychodzą z baru pojedynczo, ale czekają na siebie:
 - a) wątki wychodzą w jednej grupie
 - b) wątki wychodzą w grupach o określonej liczbie wątków

Problemy dodatkowe: W jaki sposób zagwarantować możliwość wielokrotnego używania bariery i jak próbować umożliwić istnienie wielu barier w jednym programie.

Warunki zaliczenia:

1. Obecność na zajęciach i wykonanie co najmniej kroków 1-6
2. Oddanie jednostronicowego sprawozdania z kodem źródłowym procedury w C i zawartością pliku Makefile (w przypadku wykonania kroków dodatkowych obie wersje pliku Makefile).