

# Przetwarzanie Równoległe – zadanie 3A

09.04.2013

Zaprojektuj i zaimplementuj synchronizację dla poniższego problemu. Należy użyć mechanizmów MPI.

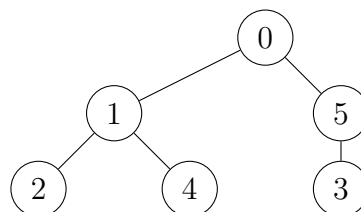
## Problem.

1. Siatka szpiegowska ma strukturę hierarchiczną – każdy szpieg (poza szefem siatki) kontaktuje się z dokładnie jedną osobą z wyższego poziomu, pozostając kontaktem dla pewnej liczby szpiegów z niższego poziomu. Szpiegzy mają kryptonimy będące kolejnymi liczbami. Szef ma zawsze numer 0.

Struktura siatki jest określona w (supertajnym) pliku `siatka`. Pierwsza linia oznacza liczbę szpiegów, w kolejnych liniach zapisane są krawędzie: `[id szpiega z wyższego poziomu] [spacja] [id szpiega z niższego poziomu]`.

Poniżej znajduje się przykładowy plik i odpowiadająca mu struktura sieci.

6
0 1
0 5
1 4
1 2
5 3



Zakładamy, że sieć jest drzewem (czyli jest spójna i nie ma cykli), nie trzeba tego sprawdzać samemu.

2. Każdy szpieg wysyła swoim podwładnym komunikat z żądaniem zdania raportu na temat liczności sił wroga. Po wysłaniu raportu przez pewien czas (losowy, długość należy dobrać samemu, żeby było widać co się dzieje) szpieg zajmuje się zacieraniem śladów. Po zakończeniu zacierania śladów, szpieg czeka na raporty swoich podwładnych. Po otrzymaniu wszystkich raportów, przekazuje raport swojemu przełożonemu.
3. Szpiedzy na najniższym poziomie (czyli tacy, którzy nie mają podwładnych), obserwują siły wroga (trwa to pewien losowy czas) i przekazują przełożonemu raport na ich temat. Raport składa się z dwóch liczb – liczby żołnierzy wroga i godziny dokonania obserwacji. Można przyjąć, że obie te wartości są losowymi liczbami całkowitymi (z jakiegoś rozsądnego przedziału).
4. Po zebraniu odpowiedzi od podwładnych, szpieg przekazuje swojemu przełożonemu **najbardziej aktualny** raport (tj. ten, który ma najwyższą wartość reprezentującą czas). W przypadku dwóch równie aktualnych raportów, szpieg wybiera raport z największą liczbą żołnierzy.
5. Ostatecznie szef powinien wybrać i wydrukować najbardziej aktualny raport z całej sieci.

### **Wymagania techniczne.**

1. Rozwiązanie należy zaimplementować na klastrze Omega.
2. Dla każdego szpiega tworzymy jeden proces.
3. Do utworzenia połączeń pomiędzy szpiegami należy stworzyć topologię wirtualną (grafową).
4. Komunikacja pomiędzy procesami powinna być nieblokująca.
5. Raport powinien być strukturą, której składowymi są dwie liczby.
6. W kodzie powinny być komentarze objaśniające rozwiązanie.
7. W konsoli powinny pojawiać się komunikaty informujące o aktualnym stanie procesów.

8. Aplikacja powinna być oddana w postaci kodu źródłowego i dołączonego pliku makefile. Polecenie `make compile` powinno kompilować aplikację, a polecenie `make run n=[n] plik=[plik]` powinno wywołać program dla  $n$  procesów i wskazanej w pliku struktury sieci. Domyślna wartość dla *plik* to `./siatka` (tj. polecenie `make run n=6` jest równoważne z `make run n=6 plik=./siatka`).
9. **Termin wgrania rozwiązania: 21.04.2013 (23:59)**