

# SPRAWOZDANIE PRZETWARZANIE ROZPROSZONE

**Temat:** Implementacja algorytmu Le Lanna Changa-Roberts

**Imię i nazwisko:** Kacper Kapela

**Nr indeksu:** 128984

## Opis algorytmu:

- Algorytm Le Lanna Changa-Roberts jest algorytmem elekcji, który służy do wyboru jednego wyróżnionego procesu (dokonywane najczęściej w przypadkach awarii). Jest typem algorytmu pierścionowego - w krążącej po pierścieniu informacji, każdy proces umieszcza swój identyfikator, jeśli jest większy niż aktualny – na zakończenie proces zostaje ustalony.
- Algorytm rozwiązuje w prosty sposób problem dostępu do sekcji krytycznej
- **Pierścień z żetonem:**
  - Każdy z procesów zna identyfikator procesu poprzedzającego i następującego po nim – w ten sposób procesy ułożone są w topologii pierścienia
  - Żeton (początkowo przydzielony losowo) krąży dookoła pierścienia
  - Wejście może uzyskać tylko proces posiadający aktualnie żeton
- Proces może wysyłać wiadomości tylko w jednym kierunku (unidirectional)
- Algorytm gwarantuje wybór (w przypadku niezawodnego przesyłania komunikatów)
- Algorytm jest asynchroniczny (procesy nie odbierają informacji zwrotnej o dotarciu wysłanego komunikatu). Wykonanie funkcji nieblokującej kończy się niemal natychmiast, więc proces może kontynuować obliczenia bez oczekiwania na jej zakończenie.
- Algorytm wymaga maksymalnie przesłania 3N1 komunikatów
- Wadą algorytmu jest nietolerowanie uszkodzeń -> awaria procesu wymaga rekonfiguracji pierścienia
- Czas oczekiwania może być długi (czas przesłania n1 komunikatów)

## Implementacja algorytmu:

→ Procesy ułożone są w pierścień i wysyłają komunikaty tylko w jednym kierunku – komunikaty zawierają wartość zmiennych: **kandydat i wybrany**

→ Każdy proces może być w stanie: bierny – nie uczestniczący lub czynny – uczestniczący w elekcji

→ Początkowo każdy proces ustala swój stan na bierny

→ Proces bierny zmienia swój stan na czynny kiedy rozpoczyna elekcję (i wysyła swój identyfikator pi w komunikacie) lub kiedy otrzymuje komunikat z wartością *kandydat*

(jeśli **kandydat** > **pi** wtedy przesyła dalej wartość kandydat,

jeśli **kandydat** < **pi** wtedy podstawia kandydat:=pi i przesyła dalej nową wartość kandydat)

→ Proces czynny pi kiedy otrzymuje komunikat z wartością *kandydat* wtedy:

- ✓ jeśli **kandydat** > **pi** przesyła dalej wartość *kandydat*

- ✓ jeśli **kandydat** < **pi** nie robi nic (ta własność jest istotna jeśli kilka procesów na raz rozpoczyna elekcję)
- ✓ wreszcie jeśli **kandydat** = **pi** proces ustawia swój stan na bierny, podstawia wybrany := pi i wysyła komunikat z wartością wybrany

→ Proces czynny pi kiedy otrzymuje komunikat z wartością *wybrany* ustawia swój stan na bierny, przesyła komunikat dalej (chyba że sam jest wybranym) i kończy ustawiając swoją wartość elekt na otrzymaną wartość wybrany

**Przykładowe wywołanie algorytmu dla 3 komputerów**(*mpirun -host lab-net-8,lab-net-11,lab-net-12 -np 3 lelann*)

```

Terminal - student@lab-net-12:~/Desktop
File Edit View Terminal Tabs Help

Process name: [[41951,1],0]
Exit code: 1
-----
student@lab-net-12:~/Desktop> mpirun -host lab-net-8,lab-net-11,lab-net-12 -np 3 lelann
-----
[lab-net-12.cs.put.poznan.pl] - [Proces 2] : Generuje UID= 948504. Zaczynam swoja prace..
[lab-net-12.cs.put.poznan.pl] - [Proces 2] : Wysylam uid= 948504 do ziomeczka o id --> 0
[lab-net-8.cs.put.poznan.pl] - [Proces 0] : Generuje UID= 998236. Zaczynam swoja prace..
[lab-net-11.cs.put.poznan.pl] - [Proces 1] : Generuje UID= 284257. Zaczynam swoja prace..
[lab-net-12.cs.put.poznan.pl] - [Proces 2] : Odebralem liczbe 284257 od procesu 1
[lab-net-12.cs.put.poznan.pl] - [Proces 2] : [Recv<UID] - Recv=284257, UID= 948504 --> Nie robie nic...
[lab-net-12.cs.put.poznan.pl] - [Proces 2] : Odebralem liczbe 998236 od procesu 1
[lab-net-12.cs.put.poznan.pl] - [Proces 2] : [Recv<UID] ==> Recv=998236, UID= 948504, czyli wysylam dalej do 0 ...
[lab-net-8.cs.put.poznan.pl] - [Proces 0] : Wysylam uid= 998236 do ziomeczka o id --> 1
[lab-net-8.cs.put.poznan.pl] - [Proces 0] : Odebralem liczbe 948504 od procesu 2
[lab-net-8.cs.put.poznan.pl] - [Proces 0] : [Recv<UID] - Recv=948504, UID= 998236 --> Nie robie nic...
[lab-net-11.cs.put.poznan.pl] - [Proces 1] : Wysylam uid= 284257 do ziomeczka o id --> 2
[lab-net-11.cs.put.poznan.pl] - [Proces 1] : Odebralem liczbe 998236 od procesu 0
[lab-net-11.cs.put.poznan.pl] - [Proces 1] : [Recv<UID] ==> Recv=998236, UID= 284257, czyli wysylam dalej do 2 ...
[lab-net-12.cs.put.poznan.pl] - [Proces 2] : Odebralem liczbe 998236 od procesu 1
[lab-net-12.cs.put.poznan.pl] - [Proces 2] : Otrzymałem własnie informacje zwrotna o tym ze 998236 wygral xd
[lab-net-12.cs.put.poznan.pl] - [Proces 2] : No coz w takim razie rozsyłam tego newsa dalej do 0
[lab-net-8.cs.put.poznan.pl] - [Proces 0] : Odebralem liczbe 998236 od procesu 2
[lab-net-8.cs.put.poznan.pl] - [Proces 0] : [Recv<UID] Oho!! Własnie dotarla do mnie wiadomosc ze jestem procesem o największym UID!
[lab-net-8.cs.put.poznan.pl] - [Proces 0] : LIDER WYSYLA DO 1 info o tym ze on wygral i jest liderem
[lab-net-11.cs.put.poznan.pl] - [Proces 1] : Odebralem liczbe 998236 od procesu 0
[lab-net-11.cs.put.poznan.pl] - [Proces 1] : Otrzymałem własnie informacje zwrotna o tym ze 998236 wygral xd
[lab-net-11.cs.put.poznan.pl] - [Proces 1] : No coz w takim razie rozsyłam tego newsa dalej do 2
[lab-net-8.cs.put.poznan.pl] - [Proces 0] : OHO WIADOMOSC DO MNIE POWROCILA CZYLI WSZYSCY JUZ WIEDZA ZE JESTEM LIDEREM!
student@lab-net-12:~/Desktop> mpirun -host lab-net-8,lab-net-11,lab-net-12 -np 3 lelann^C
student@lab-net-12:~/Desktop> ^C
student@lab-net-12:~/Desktop> mpirun -host lab-net-8,lab-net-11 -np 6 test
-----

```

