

# impreza-na-polanie

Algorytm programowania rozproszonego do umawiania się zwierząt na imprezę.

W pewnym lesie żyje sobie spokojnie  $Z$  zajączków i  $N$  niedźwiedzi. Co pewien czas zajączki urządzą imprezy. W tym celu uzgadniają między sobą, na której z  $P$  nierozróżnialnych polan odbędzie się impreza. Na polanie zmieści się  $S$  zajączków, przy czym jeden niedźwiedź zajmuje tyle miejsca, co 4 zajączki.

Każdy zajączek przynosi na imprezę alkohol (to nie zasób, po prostu przed imprezą zajączek deklaruje, co przynosi). Jeżeli na imprezie pojawi się niedźwiedź, zajączki muszą zezwolić mu na picie na sępa, co oznacza, że muszą przynieść dodatkowy alkohol także dla misiów. Impreza z samymi niedźwiedziami nie może się odbyć, bo nie ma kto przynieść alkoholu.

Założenia:

$$P * (S - 1) < Z$$

Z rozszerzonej zasady szufladkowej Dirichleta, pewne jest, że jakaś impreza dojdzie do skutku, wtedy i tylko wtedy, gdy liczba zajączków jest większa od iloczynu liczby polan i pojemności polan pomniejszonej o 1.

## Struktury i zmienne:

1.  $Z$  - liczba zajączków
2.  $N$  - liczba niedźwiedzi
3.  $P$  - liczba polan
4.  $S$  - pojemność każdej polany
5. *party\_number* - lokalny numer imprezy, na którą wybiera się zwierzę
6. *req\_count* - na ilu imprezach już byłem
7. *que* - kolejka zwierząt ubiegająca się o polanę, do której ja chcę wejść, a mam większy priorytet;
8. *ack\_count* - liczba zwierząt, które odpowiedziały pozytywnie na rządanie (niedźwiedzie liczone razy 4)
9. *alco\_count* - zmienna do przechowywania liczby potwierdzeń na zapytanie **ALCO**
10. *enter\_count* - licznik zwierząt, które weszły na polanę (niedźwiedzie liczone razy 4)

## Wiadomości

Wszystkie wiadomości są oznaczone znacznikiem czasowym (timestampem), modyfikowanym zgodnie z zasadami skalarnego zegara logicznego Lamporta.

1. **REQ** [*nr*] [*id*] [*req*] żądanie dostępu do  $n$ -tej polany
2. **ACK** [*id*] [*req*] - potwierdzenie dostępu do sekcji krytycznej z polaną
3. **ENTER** [*nr*] [*id*] wejście na polanę
4. **ALCO** [*nr*] [*id*] żądanie możliwości przyniesienia więcej alkoholu (dla niedźwiedzi)
5. **OK** [*id*] - potwierdzenie możliwości przyniesienia więcej alkoholu (dla niedźwiedzi)
6. **END** [*nr*] - komunikat o zakończeniu imprezy.

Użyte zmienne:

**id** - indeks polany

**nr** - numer procesu

**req** - lokalny indeks żądania i odpowiedzi

## Stany

**REST** - Stan początkowy, nie ubieganie się o sekcję krytyczną. Zwierzę przechodzi do tego stanu po zakończonej imprezie na polanie.

**WAIT** - Stan oczekiwania na odpowiednią liczbę potwierdzeń możliwości wejścia na daną polanę.

**GLADE** - Stan oczekiwania na polanie w celu zebranie się kworum do zorganizowania imprezy.

**MOREALCO** - Stan oczekiwania na odpowiedź czy dane zwierze może przynieść więcej alkoholu.

**SELFALCO** - Stan, w który przechodzi zwierzę po odpowiedzeniu na żądanie **ALCO** (przynosi alkohol tylko dla siebie).

## Opis szczegółowy algorytmu dla procesu i:

Algorytm dzielimy na trzy główne części: dostanie dostępu do żądanej polany, wyznaczenie, kto przyniesie alkohol na imprezę oraz wykrycie końca imprezy. Przejście między częściami algorytmu odbywa się automatycznie po skończeniu poprzedniej części, a podział zrobiony jest jedynie w celu lepszego zilustrowania działania algorytmu.

### Dostęp do żądanej polany

---

- **REST** stan początkowy.
  - Zwierzę pozostaje w stanie **REST** do czasu, aż nie podejmie decyzji na jaką imprezę chce iść.
  - Reakcja na wiadomości:
    - **REQ**: odsyła **ACK**
    - każda inna wiadomość jest nie możliwa, więc ignorowana.
- 

- Wysyła rządanie **REQ** i przechodzi do stanu **WAIT**
  - Pozostaje w stanie **WAIT** tak długo aż nie uzyska wystarczającej liczby pozwoleń, a następnie przechodzi do stanu **GLADE** wysyłając informację **ENTER**
  - Reakcja na wiadomości:
    - **REQ**: jeśli priorytet żądania jest mniejszy (zegar Lamporta ma większą wartość albo równą, ale większy numer PESEL) zwierzę dokłada żądanie do listy *Que* i nie odpowiada. W przeciwnym wypadku odsyła **ACK**
    - **ACK**: zwiększa zmienną *Cnt* o jeden w przypadku odpowiedzi od zajęczka albo o 4 gdy jest ona od niedźwiedzia i nic nie odpowiada
    - każda inna wiadomość jest nie możliwa, więc ignorowana.
- 

### Przynoszenie alkoholu

---

- Zwierzę czeka w stanie **GLADE** do momentu zapełnienia się polany.
- Gdy wykryje że lista zwierząt biorących udział w imprezie jest pełna wysyła do wszystkich uczestników wiadomość **ALCO**, zerują licznik **alco\_count** i przechodzi do stanu **MOREALCO**

- Reakcja na wiadomości:
    - **REQ**: dodaje do kolejki *Que*
    - **ACK**: ignoruje
    - **ALCO**: jeśli pytanie odnosi się do danej polany odpowiada **OK** i przechodzi do stanu **SELFALCO**
    - **ENTER**: zwiększenie licznika *enter\_count*
    - każda inna wiadomość jest nie możliwa, więc ignorowana.
- 
- W stanie **MOREALCO** są zwierzęta, które chcą zostać organizatorami imprezy, a zatem przynieść więcej alkoholu.
  - Są w nim tak długo, aż każdy uczestnik zgodzi się na jego kandydaturę lub inny uczestnik przejmie odpowiedzialność za zorganizowanie imprezy.
  - Reakcja na wiadomości przez **NIEDŹWIEDZIE**:
    - **OK**: zwiększenie licznika *alco\_count* oraz brak odpowiedzi
    - **ALCO**(od zająca): żaden niedźwiedź nie chce przynieść alkoholu, więc jeśli jest taka okazja, żeby zrzucić odpowiedzialność na zająca, zawsze ją wykorzystuje. Odpowiada **OK** i przechodzi do stanu **SELFALCO** czyli dla niedźwiedzi stan w którym wie, że ktoś przyniesie mu alkohol.
    - **ALCO**(od niedźwiedzia): jeśli dany niedźwiedź ma większy priorytet od tamtego to nie odpowiada, a jeśli mniejszy to odpowiada **OK** i przechodzi do stanu **SELFALCO**
  - Reakcja na wiadomości przez **ZAJĄCE**:
    - **OK**: zwiększenie licznika *alco\_count* oraz brak odpowiedzi
    - **ALCO**(od niedźwiedzia): wysła wiadomość **ALCO** do wszystkich procesów i przechodzi w stan **MOREALCO**
    - **ALCO**(od zająca): jeśli ma większy priorytet od tamtego to nie odpowiada, a jeśli mniejszy to odpowiada **OK** i przechodzi do stanu **SELFALCO**
- 
- Zwierze w stanie **SELFALCO** przynosi alkohol tylko dla siebie jeśli w ogóle i wie, że nie chce przynieść go więcej, więc na zapytanie **ALCO** czy tamten może przynieść więcej zawsze odpowiada **OK**.

## Wykrycie końca imprezy

Gdy organizatorem staje się niedźwiedź (osoba odpowiedzialna za przyniesienie więcej alkoholu) wysła on wiadomość **END**, a więc odwołuje imporezę uważając, że nie jest to fair i on się tak nie bawi.

Kiedy organizatorem imprezy zostaje zając przynosi on alkohol dla siebie (tak jak każdy inny porządny zając) oraz alkohol dla niedźwiedzi. Impreza trwa tak długo aż organizator się nie zmęczy. Wtedy ogłasza wszystkim, że to koniec poprzez wiadomość **END**.

Niezależnie od powodu końca imprezy wszystkie zwierzęta aktualizują listę oczekujących gości na danej polanie poprzez jej wyzerowanie, a członkowie imprezy lecą kaca w stanie **REST**.