

Kradzieje nastrojów

Adrian Kokot (148165), Wiktor Szymański (148084)

K - ilość kradziejów

S - ilość sprzętu do kradziejowania

N - ilość stanowisk

REQ_S - żądanie dostępu do sprzętu do kradziejowania

REQ_L - żądanie dostępu do miejsca w laboratorium

ACK_S - potwierdzenie, że wiemy o cudzym żądaniu dostępu do sprzętu

ACK_L - potwierdzenie, że wiemy o cudzym żądaniu do laboratorium

RELEASE_S - zwolnienie sprzętu do kradziejowania

RELEASE_L - zwolnienie miejsca w laboratorium

i-ty proces postępuje następująco:

Wysyła REQ_S do wszystkich kradziejów (łącznie z samym sobą).

Gdy zachodzą równocześnie warunki:

- Własne żądanie jest na jednym z pierwszych S miejsc kolejki
- Od wszystkich pozostałych kradziejów otrzymaliśmy wiadomości o starszej etykietce czasowej

Kradziej otrzymuje sprzęt

Po losowym czasie wysyła REQ_L do wszystkich kradziejów (łącznie z samym sobą).

Po losowym czasie wysyła RELEASE_S

Gdy zachodzą równocześnie warunki:

- Własne żądanie jest na jednym z pierwszych N miejsc kolejki
- Od wszystkich pozostałych kradziejów otrzymaliśmy wiadomości o starszej etykietce czasowej

Kradziej otrzymuje miejsce w laboratorium

Po losowym czasie wysyła RELEASE_L

Obsługa wiadomości przez proces i-ty:

Aktualizujemy zegar Lamporta oraz j-tą pozycję wektora znaczników czasowych ostatnio otrzymanych wiadomości

Po otrzymaniu REQ_S od procesu j-tego, proces i-ty wstawia żądanie do kolejki żądań (posortowanej po ich znacznikach czasowych). Następnie wysyła ACK_S. Znacznik ACK_S musi być większy od REQ_S

Po otrzymaniu RELEASE_S od procesu j-tego, usuwa jego żądanie z kolejki.

Po otrzymaniu REQ_L od procesu j-tego, proces i-ty wstawia żądanie do kolejki żądań (posortowanej po ich znacznikach czasowych). Następnie wysyła ACK_L. Znacznik ACK_L musi być większy od REQ_L

Po otrzymaniu RELEASE_L od procesu j-tego, usuwa jego żądanie z kolejki.

Struktury i zmienne:

- *rank* - identyfikator procesu
- *size* - liczba kradzieży
- *toolsNumber* - liczba narzędzi
- *positionsNumber* - liczba miejsc w laboratorium
- *ackCount* - licznik wiadomości ACK od pozostałych procesów
- *globalLamport* - zegar Lamporta
- *processesClocks* - lista zegarów pozostałych procesów
- *toolsQueue* - kolejka priorytetowa narzędzi
- *positionsQueue* - kolejka priorytetowa miejsc w laboratorium
- struct *queue* - struktura przechowująca kolejkę priorytetową
- struct *node* - struktura elementu kolejki zawierająca numer procesu oraz zegar

Wiadomości:

Wszystkie wiadomości składają się z znacznika czasowego (zegar Lamporta), identyfikatora procesu oraz tagu mówiącego o tym, czy dana wiadomość dotyczy kolejki narzędzi czy miejsc w laboratorium.

- **REQUEST** - żądanie o dostęp do sekcji krytycznej
- **ACK** - potwierdzenie dostępu do sekcji krytycznej
- **RELEASE** - opuszczenie sekcji krytycznej

Stany:

Początkowym stanem procesu jest Start.

- RequestTool / RequestLab - wysła żądanie o dostęp do sekcji krytycznej (odpowiednio o narzędzie i miejsce w laboratorium)
- WaitingForTool / WaitingForLab - czeka na dostęp do sekcji krytycznej
- UsingTool / UsingLab - jest w sekcji krytycznej

Szkic algorytmu:

Proces *n* ubiegający się o narzędzie wysła do wszystkich pozostałych procesów żądania o dostęp **REQUEST** z tagiem **TOOL** oraz wstawia swoje żądanie do kolejki *toolsQueue*.

Pozostałe procesy odsyłają **ACK** z tagiem **TOOL** do procesu *n* oraz wstawiają jego żądanie do kolejki *toolsQueue*.

Proces *n* używa narzędzia gdy posiada **ACK** od wszystkich pozostałych procesów, jest na jednym z pierwszych *toolsNumber* miejsc kolejki *toolsQueue* oraz znaczniki czasowe wszystkich pozostałych procesów są starsze niż czas wysłania żądania **REQUEST**. Po skończeniu używania narzędzia proces oddaje narzędzie do ładowania i po losowym czasie wysła **RELEASE** z tagiem **TOOL**.

Proces *n* ubiegający się o miejsce w laboratorium wysła do wszystkich pozostałych procesów żądania o dostęp **REQUEST** z tagiem **LAB** oraz wstawia swoje żądanie do kolejki *positionsQueue*.

Pozostałe procesy odsyłają **ACK** z tagiem **LAB** do procesu *n* oraz wstawiają jego żądanie do kolejki *positionsQueue*.

Proces zajmuje miejsce gdy posiada **ACK** od wszystkich pozostałych procesów, jest na jednym z pierwszych *positionsNumber* miejsc kolejki oraz znaczniki czasowe wszystkich pozostałych procesów są starsze niż czas wysłania żądania **REQUEST**. Po skończeniu używania laboratorium proces zwalnia miejsce i wysła **RELEASE** z tagiem **LAB**.

Priorytet w kolejkach jest ustalany za pomocą zegara lamporta oraz identyfikatora procesu. Im większy zegar tym mniejszy priorytet, a w przypadku tego samego zegara decyduje identyfikator procesu - im mniejszy identyfikator, tym wyższy priorytet.

Opis szczegółowy algorytmu dla procesu i:

Działanie w odpowiednich stanach:

- Start: Stan początkowy, proces przechodzi z niego do RequestTool.

- 38 - RequestTool: Ze stanu RequestTool następuje przejście do stanu WaitingForTool po uprzednim wysłaniu wiadomości **REQUEST** z tagiem **TOOL** do wszystkich innych procesów, ustawieniu *ackCount* na zero oraz wstawieniu własnego żądania do kolejki *toolsQueue*. Wszystkie wiadomości **REQUEST** mają tę samą wartość zegara.
- 39 - WaitingForTool: Oczekiwanie na dostęp do sekcji krytycznej. Ze stanu WaitingForTool następuje przejście do stanu UsingTool pod warunkiem, że proces dostał wiadomość ACK od wszystkich pozostałych procesów, jest na jednym z pierwszych *toolsNumber* miejsc kolejki *toolsQueue* oraz znaczniki czasowe wszystkich pozostałych procesów są starsze niż czas wysłania żądania **REQUEST**.
- 40 - UsingTool: Przebywanie w sekcji krytycznej. Po losowym czasie następuje przejście do stanu RequestLab oraz utworzenie osobnego wątku, który po losowym czasie wysyła wiadomość **RELEASE** z tagiem **TOOL** do pozostałych procesów oraz usuwa swoje żądanie z *toolsQueue*.
- 41 - RequestLab: Ze stanu RequestLab następuje przejście do stanu WaitingForLab po uprzednim wysłaniu wiadomości **REQUEST** z tagiem **LAB** do wszystkich innych procesów, ustawieniu *ackCount* na zero oraz wstawieniu własnego żądania do kolejki *positionsQueue*. Wszystkie wiadomości **REQUEST** mają tę samą wartość zegara.
- 42 - WaitingForLab: Oczekiwanie na dostęp do sekcji krytycznej. Ze stanu WaitingForLab następuje przejście do stanu UsingLab pod warunkiem, że proces dostał wiadomość **ACK** od wszystkich pozostałych procesów, jest na jednym z pierwszych *positionsNumber* miejsc kolejki *positionsQueue* oraz znaczniki czasowe wszystkich pozostałych procesów są starsze niż czas wysłania żądania **REQUEST**.
- 43 - UsingLab: Przebywanie w sekcji krytycznej. Po losowym czasie następuje przejście do stanu RequestTool oraz wysłanie wiadomości **RELEASE** z tagiem **LAB** do pozostałych procesów oraz usunięcie swojego żądania z *positionsQueue*.
- 44 **Reakcja na wiadomości:**
- 45 W przypadku jakiegokolwiek wiadomości proces wstawia informację o aktualnym zegarze procesu wysyłającego wiadomość do *processesClocks*.
- 46 - **ACK**: Proces zwiększa wartość licznika *ackCount*.
- 47 - **RELEASE**: Proces usuwa najbardziej priorytetowe żądanie danego procesu z odpowiedniej kolejki (*toolsQueue* w przypadku tagu **TOOL** i *positionsQueue* w przypadku tagu **LAB**)
- 48 - **REQUEST**: Proces wstawia żądanie danego procesu do odpowiedniej kolejki oraz odsyła wiadomość **ACK** z aktualnym stanem zegara