

Dawid Królak 145 383
Tomasz Strzeszewski 145 361

Dokumentacja projektu Winiarze

Opis problemu:

Założenia na temat środowiska komunikacyjnego:

- pełna asynchroniczność
- brak wyróżnionych procesów (podział jedynie na Winiarzy i Studentów, bez specjalnych procesów w żadnej z tych kategorii)
- kanały między procesami są FIFO i niezawodne
- procesy nie ulegają awarii
- pojemność kanałów jest nieograniczona

Parametry programu:

W - ilość procesów Winiarzy
S - ilość procesów Studentów
B - ilość Bezpiecznych Miejsc
M - maksymalna ilość wina

Złożoność komunikacyjna:

- optymistyczna: $2*(W-1) + S + 1$
- pesymistyczna $3*(W-1) + M*(S+1)$

Złożoność czasowa:

- optymistyczna: 4
- pesymistyczna $3 + 2*M$

Każdy proces podzielono na dwa wątki - główny i komunikacyjny. Pierwszy zajmuje się sterowaniem najważniejszymi działaniami każdego procesu, przykładowo wątek główny Winiarza zajmuje się produkcją wina i transakcjami ze Studentami, a wątek komunikacyjny dogaduje z pozostałymi Winiarzami, kto następny wejdzie do Bezpiecznego Miejsca.

Winiarz:

Wątek główny:

1. Winiarz produkuje losową ilość X wina (zajmuje mu to proporcjonalną ilość czasu)
2. Winiarz wysyła request do pozostałych Winiarzy, sygnalizując chęć zajęcia jednego z Bezpiecznych Miejsc.
3. Wątek blokuje się aż do momentu otrzymania od innych Winiarzy sygnałów ACK w ilości W-B.
4. Po odblokowaniu Winiarz zajmuje bezpieczne miejsce i pozostaje w nim, aż nie pozbędzie się całego wyprodukowanego wina poprzez transakcje ze Studentami.
 - 4.1 w bezpiecznym miejscu Winiarz wysyła informację o ilości posiadanego wina do wszystkich studentów
 - 4.2 następnie oczekuje na wiadomość od studenta, który wino zabiera
 - 4.3 jeśli Winiarz nie pozbył się jeszcze całego wina, wraca do punktu 4.1
 - 4.4 jeśli pozbył się całego wina to opuszcza Bezpieczne Miejsce
5. Winiarz wysyła ACK do wszystkich Winiarzy ubiegających się o miejsce.

6. Winiarz wraca do punktu 1.

Wątek komunikacyjny:

1. Winiarz oczekuje na wiadomości od pozostałych Winiarzy
2. Jeśli otrzymał request o Bezpieczne Miejsce to:
 - jeśli zegar Lamporta nadawcy był mniejszy (lub zegar Lamporta był równy, ale id procesu było mniejsze), odsyła ACK
 - w przeciwnym wypadku dodaje nadawcę do lokalnej kolejki, którą opróżni dopiero w punkcie 5. wątku głównego
3. Jeśli otrzymał ACK, to jeśli zegar Lamporta nadawcy był większy od tego, który Winiarz miał w momencie ostatniego nadania requestów (zabezpieczenie przed przyjmowaniem przestarzałych ACK) to zlicza ile już takich sygnałów otrzymał.
 - Jeśli jest ich równo (lub więcej niż) W-B to wątek komunikacyjny wysyła sygnał do głównego, aby ten mógł się odblokować i wejść do bezpiecznego miejsca, a sam wątek komunikacyjny blokuje się i czeka, aż winiarz tego miejsca nie opuści.
 - W przeciwnym wypadku wątek powraca do punktu 1.

Student

Każdy Student utrzymuje kolejkę Winiarzy, w kolejności przychodzenia ofert, a także liczebności wina w tych ofertach. Studenci wysyłają do pozostałych informację o przychodzących do nich ofertach.

Wątek główny:

1. Student planuje imprezę, na którą będzie potrzebował losową ilość X wina.
2. Student wysyła request do pozostałych studentów, sygnalizując chęć dokonania transakcji z Winiarzem.
3. Wątek blokuje się, aż do momentu otrzymania od pozostałych Studentów sygnałów ACK w ilości $S-1$.
4. Po odblokowaniu Student dokonuje próby wyboru Winiarza do transakcji
 - jeśli nie otrzymał od żadnego Winiarza oferty, to czeka aż taka do niego dotrze
 - jeśli istnieją oferty, Student przewiduje, które oferty wezmą Studenci znajdujący się przed nim w kolejce (każdy student z każdą wiadomością od innego porządkuje ich sobie w lokalnej kolejce). Jeśli nie jest w stanie wybrać Winiarza to wraca do czekania na kolejne oferty
 - jeśli Studentowi udało się wybrać Winiarza (ponieważ miał dostatecznie wysokie miejsce w kolejce), to wysyła do niego informację o ilości zabieranego wina i wraca do kolejki
 - jeśli Studentowi nie udało się zebrać wymaganej ilości wina to wraca do punktu 2.
 - w przeciwnym wypadku Student opuszcza kolejkę i wysyła aktualizację oferty wybranego Winiarza do pozostałych Studentów
5. Student wysyła ACK do wszystkich Studentów ubiegających się o transakcję.
6. Student odczekuje losową ilość czasu (imprezuje) po czym wraca do punktu 1.

Wątek komunikacyjny:

1. Student oczekuje na wiadomości od Studentów lub Winiarzy
2. Jeśli otrzymał request o transakcję to:

- jeśli zegar Lamporta nadawcy był mniejszy (lub zegar Lamporta był równy, ale id procesu było mniejsze), odsyła ACK
- w przeciwnym wypadku dodaje nadawcę do lokalnej kolejki, którą opróżni dopiero w punkcie 5. wątku głównego

3. Jeśli otrzymał ACK, to jeśli zegar Lamporta nadawcy był większy od tego, który Student miał w momencie ostatniego nadania requestów (zabezpieczenie przed przyjmowaniem przestarzałych ACK) to zlicza ile już takich sygnałów otrzymał.

- Jeśli jest ich równo (lub więcej niż) S-1 to wątek komunikacyjny wysyła sygnał do wątku głównego, aby ten mógł się odblokować i wejść w gotowość do wybrania winiarza i transakcji, a sam wątek komunikacyjny blokuje się i czeka, aż Student tego miejsca nie opuści.

4. Jeśli otrzymał ofertę od Winiarza, dodaje go na koniec swojej kolejki Winiarzy i aktualizuje ilość oferowanego przez niego wina

5. Jeśli otrzymał informację od innego Studenta o otrzymanej przez Winiarza ofercie, dodaje ją na koniec odpowiadającej temu Studentowi kolejki w strukturze decyzyjnej.

6. Jeśli otrzymał aktualizację oferty Winiarza od Studenta, dokonuje odpowiednich zmian w liście ofert