

# Laboratorium Podstawy Przetwarzania Rozproszonego

## SPRAWOZDANIE z zadania .....

Nazwisko Imię	album	termin zajęć
Mikołaj Pudlicki	132311	środa 16:50
Tomasz Dziecioł	132216	

### Część I – Algorytm rozwiązania

#### 1. Definicja problemu

W małym miasteczku Warszawce (nie mylić z Warszawą) wybudowano termy miejskie. Aby wejść do basenu, klient musi skorzystać z jednej z 3 szatni, gdzie w każdej dostępnych jest  $M$  szafek (to uznajemy za pojemność szatni). Klienci mogą być mężczyznami lub kobietami (dla uproszczenia innych przypadków nie rozważamy). Mężczyźni i kobiety nie mogą równocześnie przebywać w tej samej szatni (nie w Warszawce!). Szatnie nie są z góry przypisane do mężczyzn lub kobiet -- do pustej szatni może wejść równie dobrze kobieta, jak i mężczyzna. Klienci nie zmieniają płci i nie próbują dostać się na siłę do szatni zajętej przez płć przeciwną. Po wyjściu z basenu klienci wracają do tej szatni, z której wyszli. Mamy w sumie  $K$  klientów term (kobiety + mężczyźni), gdzie  $K > 3M$ . Napisać program dla procesu klienta, umożliwiający wielokrotne korzystanie z term. Szatnie i szafki należy traktować jako zasoby. Basen ma nieograniczoną pojemność.

#### 2. Założenia przyjętego modelu komunikacji

- asynchroniczny system z wymianą komunikatów
- topologia połączeń: [każdy z każdym](#)
- wymagana pojemność kanału: [4 wiadomości w jednym kierunku](#)
- inne wymagane własności sieci komunikacyjnej: [kanały typu FIFO](#)

#### 3. Algorytm wzajemnego wykluczania

Dane Globalne:

- `int M` – liczba szafek per szatnia
- `int K` – liczba klientów term

Dane procesu:

- `int id` – identyfikator(rank)
- `int T` – aktualny zegar skalarny Lamporta
- `int TLast` – etykieta czasowa, kiedy proces chciał ostatnio wejść do szatni
- `_LOCAL_QUEUE Queue[K]` – lokalna kolejka priorytetowa żądań

W skład struktury `_LOCAL_QUEUE` wchodzi pola:

- `int type` – 0 = kobieta, 1=mężczyzna
- `int cloakroom` – numer szatni, liczba od 0-2, w przypadku braku przydziału -1
- `int TLast` - etykieta czasowa, kiedy proces chciał ostatnio wejść do szatni

Na samym początku klient losuje szatnię, do której chce wejść.

Gdy klient  $P_i$  chce wejść do szatni:

`Queue_i.clear()` – czyszczenie kolejki

`TLast_i = T_i` – zapamiętanie czasu, kiedy proces chciał wejść do szatni

`Queue_i[id_i].TLast = T_i` – dodanie swojej zgody do kolejki

`SendRequestToAll(idi_, TLast_i)` – wysłanie żądania do wszystkich klientów(procesów)

Kiedy klient  $P_i$  otrzyma żądanie od  $P_j$ :

`T_i = max(TLast_j, T_i) + 1` – wybieramy maksimum z etykiety czasowej  $P_j$  otrzymanej w żądaniu, a etykietą czasową procesu  $P_j$  i inkrementujemy o 1

`if(isQueueEmpty())` – sprawdzamy czy proces  $P_i$  stara się o dostęp do szatni

`sendAcceptanceToP_j(id_i, T_i)` – wysyłamy aktualną etykietę czasową

else

`sendAcceptanceToP_j (id_i, TLast_i)` – wysyłamy etykietę czasową, kiedy proces  $P_i$  chciał wejść do szatni

W momencie otrzymania zgody od  $P_j$  klient  $P_i$ :

`Queue_i[id_j].TLast = T_j`

`if(queueSize() == K and isOtherSexInCloakroom() and sum(Queue_i[id_i].cloakroom, Queue_id_i)<M)` – sprawdzamy czy otrzymaliśmy zgody od wszystkich procesów, czy nie ma innej płci w szatni, oraz czy w szatni jest mniej osób, niż M

`goChangeAndSwimming()`

Po wyjściu z szatni klient  $P_i$ :

`sendInformationToEveryone(id_i)` – wysłanie informacji o końcu przebywania w szatni

Gdy klient  $P_i$  odbierze informacje o wyjściu klienta  $P_j$  z szatni:

`Q_i[id_j].Tlast = MINIMUM_PRIORITY`

## 4. Analiza złożoności komunikacyjnej algorytmu

*złożoność pojedynczego przebiegu jednej instancji algorytmu (czyli z punktu widzenia pojedynczego procesu)*

- złożoność komunikacyjna pakietowa, wyrażona w liczbie komunikatów:  $3(n-1)$

- złożoność czasowa przy założeniu jednostkowego czasu przesłania pojedynczego komunikatu w kanale:  
3 jednostki czasu

*należy wyznaczyć **dokładną** złożoność (nie rząd złożoności), a gdy możliwe są różne przypadki – należy podać złożoność pesymistyczną oraz średnią,*

## **Część II – Implementacja rozwiązania**

*kod źródłowy implementacji w środowisku PVM/MPI*