

Synchronizacja - semaforey

Jakub Robaczewski

Semaforey:

Zaimplementuje 3 semaforey:

- `Queue_full` <- kolejka pełna
- `Queue_empty` <- kolejka pusta
- `Queue_lock` <- semafor mutex, zapewniający wyłączny dostęp

Algorytm programu:

Message:

- Czekaj na podniesiony `queue_lock` i podniesiony `queue_full`, zablokuj `queue_lock`
 - Można uśpić na losowy przedział czasu, jeśli zablokowany
- Wstaw element do kolejki (w przypadku VIP z przodu)
- `queue_empty` += 1
- Sprawdź czy jest miejsce i ewentualnie `queue_full` += 1
- Odblokuj `queue_lock`
- Sprawdź czy osiągnął maksymalną liczbę wiadomości:
 - Zakończ program
- Kontynuuj

Main:

- Jeżeli opuszczony semafor `queue_empty`:
 - Czekaj na podniesienie semafora `queue_lock`, zablokuj
 - Wyciągnij element z kolejki
 - `queue_full` -= 1
 - `queue_empty` -= 1
 - Odblokuj semafor `queue_lock`
- Po obsłużeniu określonej liczby sygnałów – zakończ działanie

Główna pętla:

Uruchom main

Uruchom 2 instancje message (normalne i VIP), realizowane za pomocą `fork()` lub `pthread`

Testowanie:

- Wypisywanie czasu od początku działania programu (`time(0)`) dla każdej wypisywanej wiadomości. Ułatwi sprawdzanie kolejności
- Każda zmiana stanu semafora printuje jego stan na ekranie.
- Log jest również zapisywany do pliku, żeby ułatwić jego późniejszą analizę
- Wiadomości generowane przez pisarzy są w postaci `{id_wiadomości}_{id_pisarza}`