

Systemy Operacyjne - laboratorium

Ćwiczenie 3 - Synchronizacja monitorów

Paweł Bęza

1. Treść zadania

W systemie są dwie kolejki FIFO o maksymalnym rozmiarze S które przechowują wiadomości o następującej strukturze – id kolejki, id producenta, id wiadomości. Na kolejkach operują trzy procesy:

- a) czytelnicy - procesy te sparametryzowane są id kolejki na której pracują; procesy te oczekują na pojawienie się wiadomości w swojej kolejce, gdy taka się pojawi wypisują jej treść (bez usuwania wiadomości z kolejki) a następnie zasypiają na losowy okres czasu. Dopuszczalne jest aby wiele procesów tego typu operowało jednocześnie na kolejce a także by procesy te wielokrotnie wypisały ta sama wiadomość. Nie jest natomiast dopuszczalne aby procesy te czytały zawartość kolejki w sytuacji gdy operuje na niej proces z którejś z pozostałych kategorii,
- b) pisarze-producenty - co losowy okres czasu tworzą i umieszczają w jednej z kolejek nową wiadomość. Jeżeli proces wyprodukuje wiadomość a kolejka w której zamierzał ją umieścić jest pełna, to proces taki oczekuje aż w kolejce powstanie wolne miejsce,
- c) pisarze-konsumenci - co losowy okres czasu zabierają po jednej wiadomości z każdej kolejki - jeżeli w którejś z kolejek nie ma wiadomości to proces ten oczekuje na pojawienie się nowej wiadomości.

Należy zapewnić sprawne działanie systemu (bez zakleszczeń oraz zagłodzenia któregoś z procesów, zaginięcia czy też zduplikowania wiadomości) w przypadku gdy występuje N procesów klasy pierwszej, M procesów klasy drugiej i P procesów klasy trzeciej!

2. W celu zaimplementowania rozwiązania:

- Utworzę klasę **SyncQueue** dziedziczącą po klasie **Monitor** reprezentującą kolejkę FIFO, której dostęp będzie chroniony przez następujące warunki (klasa **Condition**):
 - **full** oraz **empty** - mają te same funkcje co w klasycznym problemie *producentów-konsumentów*. **Full** powstrzymuje *pisarzy-konsumentów* przed konsumowaniem jeśli kolejka jest pusta. Natomiast **empty** powstrzymuje *pisarzy-producentów* przed pisaniem do pełnej kolejki.**Full** inicjalizujemy wartością 0, a **empty** wartością długości kolejki
- Utworzę klasę **SharedQueues**, która będzie trzymała kolejki, które będą chronione przed dostępem za pomocą wspomnianych monitorów tak jak to było opisane wyżej

3. Testy

Stworzę po 2 wątki *czytelników* czytających z 2 różnych kolejek oraz analogicznie 2 wątki *pisarzy*. Ponadto utworzę wątek konsumenta biorącego z każdej z kolejek po 1 wiadomości.

Każdy z podanych procesów po wykonaniu swojego zadania będzie zasypiał na losowy okres czasu.