

SOI.A Zadanie na laboratorium 4.

Grupa pon. 12:15-14:00

1. Zaimplementować mechanizm kolejek wiadomości.
2. Wykorzystując zaimplementowany mechanizm kolejkowy przygotować symulację produkcji pierogów (produkcja pierogów mięsnych, na specjalne zamówienie przyspieszona produkcja pierogów grzybowych i z serem).

Uwagi szczegółowe do punktu 1:

- Kolejki należy zaimplementować w systemie Linux przy użyciu monitorów.
 - Dopuszczalna – ale absolutnie nie wymagana – jest zmiana implementacji z wieloprosesowej na wielowątkową.
 - W wariancie wielowątkowym pamięć wspólna jest zapewniona przez sam proces macierzysty, więc wystarczy zapewnić implementację monitorów w oparciu np. o mutexy.
 - W wariancie wieloprosesowym należy wykorzystać semaforey i pamięć wspólną, jak w zadaniu 3, ukrywając je pod postacią monitorów.
- Kolejki powinny zapewniać:
 - ograniczenie pojemności kolejki
 - mechanizm producent-konsument
 - obsługę priorytetową wskazanych obiektów (obiekty o większym priorytecie wyprzedzają w kolejce te o niższym priorytecie)
 - dostarczanie według zasady FIFO (first in – first out) w ramach każdego priorytetu.
- Zalecana jest implementacja biblioteki realizującej funkcje mechanizmu kolejkowego, wykorzystywanej przez pozostałe programy implementowane w ramach zadania.
- Wejściówka będzie weryfikować znajomość definicji monitora, różnic pomiędzy monitorem a innymi mechanizmami synchronizacji, oraz różnic pomiędzy procesami a wątkami.

Uwagi szczegółowe do punktu 2:

- W symulacji pojawiają się następujące elementy:
 - Wytwórnia farszu mięsnego – proces produkujący 5 jednostek farszu na sekundę. Farsz przekazywany jest do magazynu farszu.
 - Wytwórnia farszu grzybowego – proces produkujący przeciętnie jedną jednostkę farszu grzybowego na 3 sekundy (konieczna generacja losowa o takiej wartości oczekiwanej). Farsz przekazywany jest do magazynu farszu.
 - Wytwórnia farszu serowego – proces lub wątek produkujący przeciętnie dwie jednostki farszu serowego na 3 sekundy (konieczna generacja losowa o takiej wartości oczekiwanej). Farsz przekazywany jest do magazynu farszu.
 - Wytwórnia ciasta – proces produkujący 8 jednostek ciasta na sekundę. Ciasto przekazywane jest do magazynu ciasta.
 - Wytwórnia pierogów – proces pobierający po jednej jednostce farszu i ciasta z magazynów i przekazująca jedną jednostkę pierogów do magazynu pierogów. Wykonuje pięć takich operacji na sekundę.
 - Dyspozytornia – odbiera trzy jednostki pierogów na sekundę i wysyła je do odbiorców (czyli usuwa z modelu).
 - Magazyny – kolejki o pojemności 100 jednostek.
- Farsz grzybowy produkowany jest na specjalne zamówienie, więc produkcja takich

pierogów jest priorytetowa – taki farsz jest w pierwszej kolejności pobierany z kolejki, a takie pierogi w pierwszej kolejności ładowane.

- Farsz serowy również produkowany jest na specjalne zamówienie, ale marża na tych pierogach jest niższa, niż grzybowych, więc produkcja takich pierogów ma priorytet pośredni – taki farsz jest pobierany z kolejki w drugiej kolejności, kiedy nie ma w niej grzybowego, a takie pierogi podobnie, w drugiej kolejności ładowane.
- Oczywiście jeśli nie ma miejsca w magazynie odbiorczym, produkt nie jest wytwarzany.
- Szczegóły nie ujęte w niniejszym opisie są otwarte – należy je dookreślić w koncepcji.
- Wyniki mogą być wizualizowane w dowolny czytelny sposób, umożliwiający interpretację. Zwracam uwagę, że w przypadku wysyłania komunikatów przez kilka procesów na jedną konsolę tekstową istnieje ryzyko mieszania się poszczególnych strumieni znaków. Można tego uniknąć przez użycie innego podejścia (osobne konsole tekstowe, wizualizacja graficzna 2D, 3D, telepatia, cokolwiek) lub poprzez synchronizację – w końcu i tak jej używamy...
- Należy zademonstrować działanie systemu w sytuacji, gdy jest po jednej wytwórni każdego typu i jedna dyspozytornia, a także pokazać dobry (a najlepiej optymalny) sposób doboru liczby poszczególnego rodzaju procesów. Uwaga – zawsze jest tylko jeden magazyn każdego typu. Uwaga druga – wydajności niektórych elementów systemu **różnią się** od założonych w zadaniu 3!

Uwaga dodatkowa: Zastosowane podejście do synchronizacji warto sprawdzić pod kątem możliwości wystąpienia blokad. Nie jest to wymagane, ale pomaga uniknąć błędów, choć wymaga sporo pracy.

Jeśli ktoś zdecyduje się wykonać takie badanie i zrobi to, modelując przyjęte rozwiązanie w dostępnym na stronie przedmiotu narzędziu Dedan, po czym zademonstruje wyniki – automatycznie **zaliczy także zadanie 5** (z punktacją zależną od poprawności wykonania modelu).