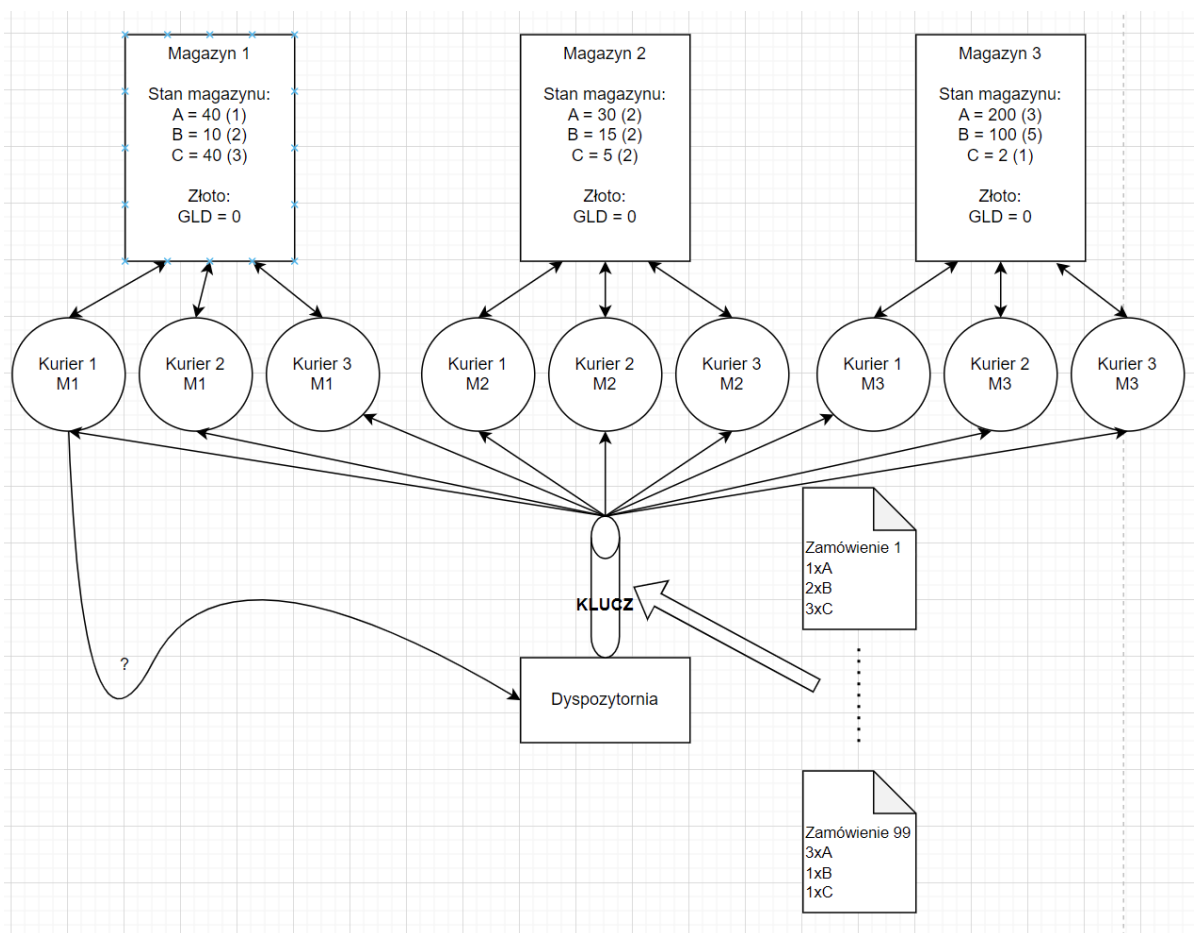


Projekt zaliczeniowy PSiW

W ramach projektu będą Państwo tworzyć projekt systemu, który “symuluje” działanie dyspozytorni zamówień oraz trzech firm magazynowych. Ogólna koncepcja aplikacji polega na tym, że:

1. Odpalają Państwo 4 terminale
2. W każdym z terminali odpalają Państwo odpowiednią aplikację wieloprocesową (3x - magazyn z 3 kurierami oraz 1x dyspozytornię)
3. Aplikacje te komunikują się ze sobą zgodnie z regułami opisanymi poniżej i po określonym czasie kończą się wypisując finalny stan aplikacji. W terminalu powinno się wypisywać co się dzieje np. w terminalu dyspozytorni mamy informację: “zlecono zlecenie 14 na 3xA, 2xB, 1xC” albo “zlecenie 14 odebrał kurier 3 magazynu 1”.

Poniższy schemat przedstawia w formie graficznej ogólną koncepcję projektu:



- 1x dyspozytornia
- 3x firma magazynowa

Dyspozytornia

- aplikacja uruchamiana w jednym z 4 otwartych terminali.

- Aplikacja co jakiś czas (np. co 0,5 sek lub 0,025 sek) losuje zamówienie i umieszcza je w “kanale komunikacyjnym”. Losowanie można wykonać za pomocą procedury rand lub za pomocą jakiejś biblioteki (proszę napisać komentarz w kodzie skąd pochodzi ta funkcja).
- Aplikację uruchamiamy w terminalu za pomocą polecenia:
\$./dystrybutornia <klucz> <liczba_zamówień> <max_A_per_zam.>
<max_B_per_zam.> <max_C_per_zam.>

Czyli np.

\$./dystrybutornia “kolejka” 1000 5 3 4

- klucz - klucz globalnego systemu przesyłania zamówień - implementacja to może być kolejka fifo, kolejka komunikatów, czy pamięć współdzielona. (każde zamówienie to trójka <1, 5, 29> oznaczająca zamówienie na 1x surowiec typu A, 5x B i 29x C).
- liczba zamówień - po prostu int mówiący ile razy należy wylosować zamówienia
- max_X_per_zam - ile maksymalnie można danego surowca zamówić per zamówienie
- Dyspozytornia się wyłącza jak wszystkie procesy magazyny się już wyłączyły. Wówczas wypisywana jest wartość GLD jakie dyspozytornia zapłaciła za wszystkie wykonane zlecenia.

Firma magazynowa:

- odpalamy 3 takie procesy
- każdy z nich uruchamiany jest z podaniem ścieżki do pliku konfiguracyjnego oraz z kluczem dostępowym do globalnego systemu otrzymywania zamówień:

./m1 ./m1_konf.txt “kolejka”

- plik konfiguracyjny zawiera informacje:
 - ile jakiego surowca zawiera magazyn np. 1000xA, 2000xB, 3000xC
 - ile kosztuje dany surowiec
- po załadowaniu pliku konfiguracyjnego tworzeni są kurierzy (podprocesy procesu magazynu i jest ich dla każdego magazynu 3) nasłuchujący na kanale globalnym dyspozytorni, oraz tworzą kanał komunikacji z magazynem
- sposób komunikacji pomiędzy magazynem w obie strony (pobieranie z magazynu i informacja zwrotna czy udało się pobrać z magazynu i ile to kosztowało) może być wykonane za pomocą potoków, kolejek fifo, kolejek komunikatów lub pamięci współdzielonej
- Każdy kurier co jakiś czas pobiera zamówienie z dyspozytorni, po jego pobraniu wysyła informacje do magazynu z prośbą o pobranie odpowiedniej liczby zasobów
- magazyn informuje kuriera czy da się zrealizować zamówienie i przesyła mu informacje ile ono kosztuje
- jak nie da się zrealizować zamówienia to magazyn informuje o tym kuriera i kurier się wyłącza (nie realizując i nie oddając tego zamówienia)
- kurier jak otrzyma potwierdzenie realizacji zamówienia wraz z ceną to przesyła tą informację (w dowolny sposób poznany na zajęciach) do dyspozytorni aby ta mogła wyliczyć sumę GLD jaką płaci za wszystkie zrealizowane zamówienia.
- Jak wszyscy kurierzy danego magazynu się wyłączą to magazyn się wyłącza

- kurier jak nie może zrealizować zamówienia (brak surowców w magazynie) albo jak nie udało mu się pozyskać żadnego nowego zlecenia przez ostatnie 150 sek. to się wyłącza
- Magazyn się wyłącza jak jak wszyscy jego kurierzy są wyłączeni. Po wyłączeniu się wypisuje zawartość magazynu oraz liczbę zarobionego GLD.

Gra kończy się jak wszystkie magazyny się wyłączą i wypiszą stan magazynu oraz zawartość skarbca. Dyspozytornia też powinna wypisać ile GLD zapłaciła.

Ocenie podlega:

1. Wykonanie poprawne zadania wraz z wykorzystaniem TYLKO dozwolonych metod poznanych na zajęciach dot. mechanizmów systemowych i procesowych.
2. Odporność na błędy (gra nie działa jak podaliśmy błędne dane)
3. przygotowany schemat działania programu w formie opisanego rysunku, tłumaczącego "kto" (proces) tworzy jaką kolejkę / pamięć współdzieloną i jak się komunikuje z resztą systemu

PROJEKTY oddajemy na ostatnich zajęciach w semestrze. Projekt należy przed zdawaniem wysłać na adres: andrzej.stroinski@cs.put.poznan.pl