

KONCEPCJA LAB_04 - SYNCHRONIZACJA Z UŻYCIEM MONITORÓW

ŁUKASZ POKORZYŃSKI

ZAŁOŻENIA WSTĘPNE

Zadanie zostanie zrealizowane na wątkach. System kolejek zostanie umieszczony w osobnej bibliotece, która zdefiniuje używane struktury i dostępne operacje na kolejkach. Biblioteka będzie zrealizowana przy pomocy kodu użytego w poprzednim zadaniu z rozszerzeniem o klasę monitora.

KOLEJKI

Kolejki będą stworzone przy pomocy monitorów i będą klasami napisanymi w języku C++. Jako, że implementacja będzie opierała się na wątkach, to dostęp do kolejek będzie definiowany przez mutexy. Priorytet wiadomości zawsze będzie rozpatrywany na samym początku, by odpowiednio wstawić wiadomość do kolejki i mieć ustaloną kolejność priorytetów (najwyższy > wysoki > normalny). Pierwsza wstawiona wiadomość np. o najwyższym priorytecie zachowuje swoją pozycję i przy pojawieniu się kolejnej o priorytecie najwyższym, ta nowa wskazuje za wcześniej wrzuconą (zachowujemy strukturę FIFO). Komunikaty będą przechowywane w odpowiedniej tablicy. Dostępne będą standardowe operacje na kolejkach, takie jak wgląd do wiadomości z przodu kolejki, dodawanie, pobieranie.

SYSTEM

Trzy kolejki A, B i C. Poza standardowymi producentami i konsumentami będziemy mieli producenta ochronnego, który będzie wrzucał pusty komunikat o najwyższym priorytecie do kolejki najbardziej zapełnionej. Proces konsumenta jest zmodyfikowany o to, że przy otrzymaniu wiadomości o najwyższym priorytecie obsługuje 5 kolejnych komunikatów bez pół sekundowego opóźnienia.