**Synchronizacja – monitory**

**1.Klasa bufor**

a) Struktury danych

- int \*buf (wskaźnik na tablice, służącą za bufor),

- int head, int tail (indeks na pierwszy element w tablicy i pierwsze wolne miejsce w tablicy),

- int lenght, int size (liczba elementów w tablicy oraz rozmiar tablicy),

- condition full, empty (zmienne warunkowe umożliwiające synchronizację),

b) Metody

- void insert\_item (int) ( wstawianie elementu do bufora z wkorzystaniem metod monitora do zapewnienia synchronizacji),

- int remove\_item (void) ( wyjmowanie elementów z bufora z wykorzystaniem metod monitora do zapewnienia synchronizacji),

**2.Program symulujący problem producentów i konsumentów**

Funkcje:

1. Int setId ( void ) (funkcja przypisująca id wątkom producentów),
2. Void \*producer(void) ( funkcja symulująca produceta wstawiającego określoną liczbę produktów do bufora),
3. Void \*consumer(void) (funkcja symulująca konsument wyjmującego produkty z bufora),

Funkcja main:

Wczytuje liczbę producentów, rozmiar bufora oraz liczbę produktów jaką musi wyprodukować każdy producent. Tworzy bufor o podanym wcześniej rozmiarze. Tworzy odpowiednią liczbę wątków producenta oraz wątek konsumenta. Czeka na zakończenie wszystkich wątków i kończy program.