

/**

- * W zadaniu mamy pięć wątków: producenta i konsumenta.
- * Każdy producent może wyprodukować zadaną z góry liczbę wartości liczby typu int, tj. 4
- * Każdy wątek producenta produkuje wartości i przechowuje je we współdzielonym buforze o długości 3,
- * podczas gdy wątek konsumenta pobiera wartości z bufora.
- * Pierwszy zapis do bufora należy zrealizować dla indeksu 2.
- * Alokacja pamięci bufora powinna być zrealizowana w oparciu o calloc/malloc.
- *
- * Używamy muteksu i dwóch semaforów (full, empty) do synchronizacji dostępu do buforów.
- * Muteks służy do zarządzania dostępem do sekcji krytycznej.
- * Semafony umożliwiają wątkom sygnalizować sobie nawzajem, czy można wykonać operacje zapisu/odczytu.
- *
- * Każdy wątek producenta generuje liczbą losową, a następnie wykonuje operację na semaforze.
- * Jeśli zapis jest możliwy to zajmuje muteks, wstawia wcześniej wygenerowaną liczbę do bufora o indeksie (na pozycji) in
- * wyświetla komunikat na ekranie
- * (który producent wykonał operację na buforze, zawartość bufora o indeksie (na pozycji) in, wartość indeksu in)
- * inkrementuje wartość indeksu in (operacja modulo), zwalnia mutex
- * a następnie wykonuje operacje na semaforze.
- *
- * Każdy wątek konsumenta wykonuje operacje na semaforze.

- * Jeśli odczyt jest możliwy to zajmuje muteks
- * wczytuje zawartość bufora o indeksie (z pozycji) out
- * wyświetla komunikat na ekranie
- * (który spośród trzech konsumentów wykonał operację na buforze, pobrana zawartość bufora, wartość indeksu out)
- * modyfikuje wartość indeksu out
- * Zwalnia muteks, a następnie wykonuje operacje na semaforze.
- *
- * Funkcja main tworzy wątki producenta i konsumenta
- */