

## **ZAŁOŻENIA WSTĘPNE**

Jako rozwiązanie problemu zostanie zaimplementowany program napisany w języku C++ w środowisku Linux Ubuntu. Do poprawnej synchronizacji zostaną użyte semaforey oraz pamięć współdzielona POSIX. System kolejek priorytetowych FIFO zostanie napisany w osobnym pliku .h, zmodyfikowany model przedstawiony w treści zadania zostanie stworzony w pliku .c wykorzystującym wcześniej wspomniany plik nagłówkowy.

## **KOLEJKI**

Kolejka zrealizowana będzie jako klasa, która jako dane będzie przetrzymywała adres do pamięci współdzielonej oraz jej wielkość (ilość miejsc, które mogą zostać wypełnione przez dane). Przechowywane dane w pamięci współdzielonej będą osobno ustawiane w kolejkę jednokierunkową przy pomocy struktury "Node", która umożliwi na łatwiejsze zarządzanie elementami w przypadku wystąpienia komunikatów priorytetowych. Zwykle komunikaty będą ustawiane na "tail" bez zbędnego szukania miejsca, te priorytetowe w przypadku istnienia tylko zwykłych będą dopisywane na "head", w przeciwnym wypadku będą ustawiane jako ostatni priorytetowy, ale przed pierwszym zwykłym. W przypadku usuwania komunikatu z kolejki, będzie to realizowane przez metodę kolejki.

Semaforey, zaimplementowane w kolejce, będą nam mówiły o:

- możliwości wejścia do kolejki i wykonania operacji (semafor binarny)
- liczbie zajętych miejsc w kolejce
- liczbie wolnych miejsc w kolejce.

Wielkość kolejki będzie ustalana przy tworzeniu obiektu w programie implementującym model.

Metody do sprawdzania wypełnienia, stworzone na potrzeby obiektów kolejek, będą standardowymi metodami, do których dostęp będzie możliwy w każdym momencie bez blokowania, zwracającymi aktualny wielkość kolejki.

## **SYSTEM**

Mamy trzy kolejki - A, B i C. Do każdej z nich mamy przypisanego producenta (odpowiednio A, B, C) oraz konsumenta/prosumenta (znów, odpowiednio A, B, C). Istnieje również jeden producent specjalny, który losowo przydziela komunikat to wybranej kolejki. Procesy w postaci producentów, konsumentów, itd. będą synchronizowane przy pomocy semaforów w kolejkach i odpowiednio usypiani, jeśli kolejka jest pusta/pełna, bądź wstrzymywani jeżeli jakiś proces właśnie korzysta z danej kolejki. Każdy z producentów ma zadaną szybkość wstawiania komunikatów i w zależności od stanu wypełnienia kolejki produkuje, bądź wstrzymuje się od dalszego produkowania komunikatów. Prosumenci odczytują komunikaty i z pewnym dobranym prawdopodobieństwem dodają literkę do komunikatu i w zależności od odczytanej uprzednio literki wstawiają komunikat do odpowiedniej kolejki (A, B, C). Jeżeli komunikat był pusty, to usuwają go z kolejki.

“Losowość” zostanie osiągnięta przez wbudowane funkcje generatora liczb pseudolosowych. Do odpowiedzenia na zadane w treści zadania pytania będą używane stosowne “stemple” czasowe i odpowiednie przekierowania do plików z wynikami.