1 Treść zadania

Implementacja "bufora komunikacyjnego" - struktury danych mieszczącej maksymalnie `M` elementów, pozwalającej na wyjmowanie elementów w kolejności umieszczania i odwrotnej. Należy zadbać o zabezpieczenie przed czytaniem z pustego bufora, zapisem do pełnego bufora i jednoczesnym użyciem bufora przez różne procesy.

2 Założenia

1. Stała `M` jest określona przy uruchamianiu programu `#define`
2. Typ elementów jest określony przed rozpoczęciem implementacji

3 Realizacja

3.1 Semafory

full

Semafor inicjalizowany wartością `M`, dekrementowany przy umieszczeniu elementu w buforze, a inkrementowany przy wyjęciu elementu z bufora. Po wypełnieniu bufora, semafor zablokuje możliwość umieszczenia kolejnego elementu w kolejce.

empty

Semafor inicjalizowany wartością zero, inkrementowany przy dodawniu elemntów, a dekrementowany przy usuwaniu elementów z kolejki. Zabezpiecza przed usunięciem elementu z pustej kolejki.

mutex

Semafor binarny, zabezpieczający przed jednoczesnym korzystaniem z bufora. Przyjmuje następujące wartości:

- 0 dla zablokowanego dostępu
- 1 dla otwartego dostępu

3.2 Kolejka

Kolejka zostanie zrealizowana za pomocą tablicy wskaźników o długości `M+2`. Zerowy i pierwszy element tablicy będą przechowywać informacje odpowiednio o położeniu pustego elementu przed początkiem oraz końca kolejki. Przy dodawaniu i usuwaniu elementów kolejki, początek i koniec są odpowiednio przesuwane, tak aby zapewnić prawidłowy dostęp.

4 Planowana implementacja

4.1 Obsługa semaforów

Do prawidłowej realizacji semaforów potrzebne są następujące funkcje:

`semalloc` i `semdealloc`

Dwie funkcje odpowiadające za alokację i dealokajcę semaforów.

`seminit`

Funkcja odpowiedzialna za inicjalizację semafora zadaną wartością.

`porberen`

Funkcja sprawdzająca wartość zadanego semafora i w przypadku, gdy ta wartość jest większa lub równa zero, wstrzymująca program. Jeśli wartość jest większa od zera, funkcja dekrementuje semafor.

`verhogen`

Funkcja inkrementująca semafor.

4.2 Obsługa kolejki

Do obsługi kolejki potrzebne są trzy funkcje użytkowe i dwie funkcje pomocnicze.

`push`

Funkcja umieszczająca element w miejscu wskazywanym przez zerowy element tablicy.

`pop_front`

Funkcja pobierająca element z pierwszego miejsca tablicy (następnego za elementem wskazywanym przez zerowy element tablicy).

`pop_back`

Funkcja pobierająca element z ostatniego miejsca tablicy (wskazywanego przez jej pierwszy element)

`increment` i `decrement`

Funkcje modyfikujące wartości zerowego i pierwszego elementu tablicy, biorące pod uwagę jej wielkość. Funkcja `increment`, wywoływana jest podczas użycia `pop_front` w celu przesunięcia wskazania na pusty element. Funkcja `decrement`, wywoływana jest przy `push` i `pop_back` aby przesunąć odpowiednio wskazanie na pusty element oraz na koniec kolejki.

4.3 Funkcje dostępu

Funkcje `pop_front`, `pop_back` i `push` realizujące powyższe operacje, ale zabezpieczone za pomocą semaforów. Te funkcje dają możliwość implementowania kilku buforów.

5 Testowanie

Konieczne jest przetestowanie następujących aspektów implementacji:

- jednoczesne pisanie do bufora kilku producentów
- jednoczesne pisanie i czytanie przez producenta i konsumenta
- jednoczesne czytanie przez kilk konsumentów
- zapisywanie do pełnego bufora
- czytanie pustego bufora
- czytanie i pisanie w prawidłowych warunkach