

student:        Patryk Jan Sozański  
grupa:           215

## SYSTEMY OPERACYJNE: LABORATORIUM NR 3 RAPORT Z WYKONANEGO ĆWICZENIA

### Treść zadania

Napisz usługę "chat" dająca użytkownikom możliwość komunikacji między sobą.

Załącz, że istnieją trzy grupy użytkowników tej usługi:

- administratorzy,
- użytkownicy VIP,
- zwykli użytkownicy.

Użytkownicy VIP mają podwyższone prawa i ich wiadomości "wskakują" na początek kolejki rozsyłanych wiadomości do pozostałych użytkowników. Administratorzy w celach prewencyjnych (np.: po przeczytaniu wulgarного komunikatu) mogą w dowolnie wybranym przez siebie momencie wyczyścić całą kolejkę wiadomości.

Załącz, że wszyscy użytkownicy komunikują się w wyłącznie jednym temacie.

Przemyśl metodę testowania powstałego systemu, w szczególności zwróć uwagę na pokazanie w działaniu równoczesnego działania wszystkich grup użytkowników.

Zwróć uwagę, że test powinien wykonywać się automatycznie - bez udziału człowieka podczas samego testu. Generowanie fikcyjnych tekstowych wiadomości jest tutaj jak najbardziej wskazane.

### Sposób rozwiązania z fragmentami kodu źródłowego:

#### 1) Użyte semaforzy:

```
sem_t sem_mutex;           // Semafor pilnujący dostępu do bufora wiadomości.  
sem_t sem_pusty;           // Semafor informujący o tym, że lista wiadomości jest pusta.  
sem_t sem_pelny;           // Semafor informujący o tym, że lista wiadomości jest pełna.  
sem_t *sem_przeczytane;    // Tablica semaforów. Semafor informujący o tym, czy nowa wiadomość  
                           // została już przeczytana.  
sem_t *sem_dostepne;       // Tablica semaforów. Semafor informujący o dostępności nowej wiadomości.
```

#### 2) Pamięć współdzielona:

```
node_t* wiadomosci;        // Lista wiadomości.
```

#### 3) Problem konsument-producent:

Każdy użytkownik ma dwie opcje (wątki):

- słuchania - odczytywanie nowej wiadomości od listonosza.
- wysyłania - wysyłanie nowej wiadomości do listonosza.

Gdy wiadomość została wysłana, semafor `sem_dostepne` [`indeks_uzytkownika`] zmienia wartość i informuje o tym użytkownika.

Gdy użytkownik przeczyta wiadomość, powiadomi o tym listonosza semaforem `sem_przeczytane` [`indeks_uzytkownika`].

Listonosz wysyła wiadomość do kolejnego użytkownika. Listonosz jest konsumentem. Pobiera po kolei wiadomości z kolejki i rozsyła je do wszystkich użytkowników.

Uwaga: Więcej szczegółów w pliku `main.c`.

#### 4) Funkcje:

Opis funkcji znajduje się w pliku `main.c`.

## Testy:

### 1) Ogólne działanie:

```
UZYTKOWNIK 1 NADAWCA: ROZPOCZAL WYSYLANIE WIADOMOSCI
UZYTKOWNIK 1 NADAWCA: WIADOMOSC "SOI" WYSLANA
### KOLEJKA WIADOMOSCI:
### SOI
UZYTKOWNIK 0 NADAWCA: ROZPOCZAL WYSYLANIE WIADOMOSCI
UZYTKOWNIK 0 NADAWCA: WIADOMOSC "TE" WYSLANA
### KOLEJKA WIADOMOSCI:
### SOI
### TE
UZYTKOWNIK 1 NADAWCA: ROZPOCZAL WYSYLANIE WIADOMOSCI
UZYTKOWNIK 1 NADAWCA: WIADOMOSC "FIG" WYSLANA
### KOLEJKA WIADOMOSCI:
### SOI
### TE
### FIG
UZYTKOWNIK 2 NADAWCA: ROZPOCZAL WYSYLANIE WIADOMOSCI
UZYTKOWNIK 2 NADAWCA: WIADOMOSC "OSM" WYSLANA
### KOLEJKA WIADOMOSCI:
### SOI
### TE
### FIG
### OSM
UZYTKOWNIK 0 NADAWCA: ROZPOCZAL WYSYLANIE WIADOMOSCI
LISTONOSZ: ROZPOCZAL PRACE
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 0
UZYTKOWNIK 0 NADAWCA: WIADOMOSC "SOI" WYSLANA
### KOLEJKA WIADOMOSCI:
### TE
### FIG
### OSM
### SOI
UZYTKOWNIK 2 SLUCHACZ: ROZPOCZAL CZYTANIE WIADOMOSCI
UZYTKOWNIK 1 SLUCHACZ: ROZPOCZAL CZYTANIE WIADOMOSCI
UZYTKOWNIK 0 SLUCHACZ: ROZPOCZAL CZYTANIE WIADOMOSCI
UZYTKOWNIK 0 SLUCHACZ: OTRZYMAL WIADOMOSC "SOI"
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 1
UZYTKOWNIK 1 SLUCHACZ: OTRZYMAL WIADOMOSC "SOI"
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 2
UZYTKOWNIK 2 SLUCHACZ: OTRZYMAL WIADOMOSC "SOI"
### KOLEJKA WIADOMOSCI:
### TE
### FIG
### OSM
### SOI
```

```

LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 0
UZYTKOWNIK 0 SLUCHACZ: OTRZYMAL WIADOMOSC "TE"
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 1
UZYTKOWNIK 1 SLUCHACZ: OTRZYMAL WIADOMOSC "TE"
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 2
UZYTKOWNIK 2 SLUCHACZ: OTRZYMAL WIADOMOSC "TE"
  ### KOLEJKA WIADOMOSCI:
  ### FIG
  ### OSM
  ### SOI
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 0
UZYTKOWNIK 0 SLUCHACZ: OTRZYMAL WIADOMOSC "FIG"
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 1
UZYTKOWNIK 1 SLUCHACZ: OTRZYMAL WIADOMOSC "FIG"
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 2
UZYTKOWNIK 2 SLUCHACZ: OTRZYMAL WIADOMOSC "FIG"
  ### KOLEJKA WIADOMOSCI:
  ### OSM
  ### SOI
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 0
UZYTKOWNIK 0 SLUCHACZ: OTRZYMAL WIADOMOSC "OSM"
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 1
UZYTKOWNIK 1 SLUCHACZ: OTRZYMAL WIADOMOSC "OSM"
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 2
UZYTKOWNIK 2 SLUCHACZ: OTRZYMAL WIADOMOSC "OSM"
  ### KOLEJKA WIADOMOSCI:
  ### SOI
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 0
UZYTKOWNIK 0 SLUCHACZ: OTRZYMAL WIADOMOSC "SOI"
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 1
UZYTKOWNIK 1 SLUCHACZ: OTRZYMAL WIADOMOSC "SOI"
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 2
UZYTKOWNIK 2 SLUCHACZ: OTRZYMAL WIADOMOSC "SOI"
  ### KOLEJKA WIADOMOSCI:
  ### WCISNIJ KLAWISZ ENTER

```

Jak widać powyżej początkowo funkcje użytkowników w trybie pracy nadawcy rozpoczynają swoją pracę i wysyłają wiadomości, co powoduje wypełnianie się kolejki wiadomości. Gdy kolejka osiąga maksymalną długość, uruchamia się funkcja listonosza, która przenosi wiadomości do funkcji użytkowników w trybie pracy słuchacza i tym samym znikają one z kolejki, a funkcja wypisuje na ekranie treść wiadomości. Gdy kolejka jest pusta, program kończy pracę.

Aby cała sekwencja działań opisana powyżej mogła poprawnie funkcjonować, niezbędne jest poprawne działanie semaforów pilnujących dostępu do zasobów pamięci współdzielonej, co udało się osiągnąć.

## 2) Przepełnienia bufora:

```
UZYTKOWNIK 1 NADAWCA: ROZPOCZAL WYSYLANIE WIADOMOSCI
UZYTKOWNIK 1 NADAWCA: WIADOMOSC "# 2" WYSLANA
### KOLEJKA WIADOMOSCI:
### # 4
### # 3
### # 2
UZYTKOWNIK 1 NADAWCA: ROZPOCZAL WYSYLANIE WIADOMOSCI
UZYTKOWNIK 1 NADAWCA: WIADOMOSC "# 1" WYSLANA
### KOLEJKA WIADOMOSCI:
### # 4
### # 3
### # 2
### # 1
UZYTKOWNIK 1 NADAWCA: ROZPOCZAL WYSYLANIE WIADOMOSCI
UZYTKOWNIK 1 NADAWCA: ROZPOCZAL WYSYLANIE WIADOMOSCI
LISTONOSZ: ROZPOCZAL PRACE
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 0
UZYTKOWNIK 1 NADAWCA: WIADOMOSC "# 5" WYSLANA
### KOLEJKA WIADOMOSCI:
### # 3
### # 2
### # 1
### # 5
UZYTKOWNIK 2 SLUCHACZ: ROZPOCZAL CZYTANIE WIADOMOSCI
UZYTKOWNIK 1 SLUCHACZ: ROZPOCZAL CZYTANIE WIADOMOSCI
UZYTKOWNIK 0 SLUCHACZ: ROZPOCZAL CZYTANIE WIADOMOSCI
UZYTKOWNIK 0 SLUCHACZ: OTRZYMAL WIADOMOSC "# 4"
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 1
UZYTKOWNIK 1 SLUCHACZ: OTRZYMAL WIADOMOSC "# 4"
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 2
UZYTKOWNIK 2 SLUCHACZ: OTRZYMAL WIADOMOSC "# 4"
### KOLEJKA WIADOMOSCI:
### # 3
### # 2
### # 1
### # 5
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 0
UZYTKOWNIK 1 NADAWCA: WIADOMOSC "# 6" WYSLANA
### KOLEJKA WIADOMOSCI:
### # 2
### # 1
### # 5
### # 6
```

Powyższy fragment testu obrazuje działanie programu w sytuacji, gdy kolejka wiadomości osiągnęła maksymalną długość, a funkcja użytkownika próbuje dodać nową wiadomość. Wówczas funkcja listonosza rozpoczyna pracę, pobiera z kolejki wiadomość, tym samym zwalniając miejsce kolejnej wiadomości. Wzbudzenie funkcji listonosza powoduje zmianę wartości semaforów, za czym podąża rozpoczęcie pracy funkcji użytkowników w trybie słuchacza. Wiadomość, dla której nie było miejsca w kolejce, nie jest więc usuwana, lecz czeka do zwolnienia się miejsca w kolejce (zmiana wartości semaforów), po czym jest dodawana na koniec kolejki.

### 3) Użytkownik VIP:

```
UZYTKOWNIK 1 NADAWCA: ROZPOCZAL WYSYLANIE WIADOMOSCI
UZYTKOWNIK 1 NADAWCA: WIADOMOSC "# UZYTKOWNIK 2" WYSLANA
### KOLEJKA WIADOMOSCI:
### # UZYTKOWNIK 2
UZYTKOWNIK 0 NADAWCA: ROZPOCZAL WYSYLANIE WIADOMOSCI
UZYTKOWNIK 0 NADAWCA: WIADOMOSC "# UZYTKOWNIK 1" WYSLANA
### KOLEJKA WIADOMOSCI:
### # UZYTKOWNIK 2
### # UZYTKOWNIK 1
LISTONOSZ: ROZPOCZAL PRACE
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 0
UZYTKOWNIK VIP 2 NADAWCA: ROZPOCZAL WYSYLANIE WIADOMOSCI
UZYTKOWNIK VIP 2 NADAWCA: WIADOMOSC "# UZYTKOWNIK VIP" WYSLANA
### KOLEJKA WIADOMOSCI:
### # UZYTKOWNIK VIP
### # UZYTKOWNIK 1
UZYTKOWNIK VIP 2 NADAWCA: ROZPOCZAL WYSYLANIE WIADOMOSCI
UZYTKOWNIK VIP 2 NADAWCA: WIADOMOSC "# UZYTKOWNIK VIP" WYSLANA
### KOLEJKA WIADOMOSCI:
### # UZYTKOWNIK VIP
### # UZYTKOWNIK VIP
### # UZYTKOWNIK 1
UZYTKOWNIK VIP 2 SLUCHACZ: ROZPOCZAL CZYTANIE WIADOMOSCI
UZYTKOWNIK 1 SLUCHACZ: ROZPOCZAL CZYTANIE WIADOMOSCI
UZYTKOWNIK 0 SLUCHACZ: ROZPOCZAL CZYTANIE WIADOMOSCI
UZYTKOWNIK 0 SLUCHACZ: OTRZYMAL WIADOMOSC "# UZYTKOWNIK 2"
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 1
UZYTKOWNIK 1 SLUCHACZ: OTRZYMAL WIADOMOSC "# UZYTKOWNIK 2"
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK VIP 2
UZYTKOWNIK VIP 2 SLUCHACZ: OTRZYMAL WIADOMOSC "# UZYTKOWNIK 2"
### KOLEJKA WIADOMOSCI:
### # UZYTKOWNIK VIP
### # UZYTKOWNIK VIP
### # UZYTKOWNIK 1
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 0
UZYTKOWNIK 0 SLUCHACZ: OTRZYMAL WIADOMOSC "# UZYTKOWNIK VIP"
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 1
UZYTKOWNIK 1 SLUCHACZ: OTRZYMAL WIADOMOSC "# UZYTKOWNIK VIP"
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK VIP 2
UZYTKOWNIK VIP 2 SLUCHACZ: OTRZYMAL WIADOMOSC "# UZYTKOWNIK VIP"
### KOLEJKA WIADOMOSCI:
### # UZYTKOWNIK VIP
### # UZYTKOWNIK 1
```

Powyższy fragment testu obrazuje działanie programu w sytuacji, gdy do kolejki wiadomości dodawane są wiadomości od użytkowników VIP. Trafiają one wtedy na początek kolejki i są w pierwszej kolejności obsługiwane przez funkcje. W przypadku pojawienia się dwóch wiadomości od użytkowników VIP, obie są ustawiane na początku kolejki przed wiadomościami użytkowników standardowych.

4) Admin:

```
LISTONOSZ: ROZPOCZAL PRACE
LISTONOSZ: WIADOMOSCI DLA ADMIN 0
UZYTKOWNIK 2 NADAWCA: ROZPOCZAL WYSYLANIE WIADOMOSCI
UZYTKOWNIK 2 NADAWCA: WIADOMOSC "# UZYTKOWNIK 1" WYSLANA
### KOLEJKA WIADOMOSCI:
### # UZYTKOWNIK 2 WIADOMOSC A
### # UZYTKOWNIK 2 WIADOMOSC C
### SOI
### # UZYTKOWNIK 1
UZYTKOWNIK 2 SLUCHACZ: ROZPOCZAL CZYTANIE WIADOMOSCI
UZYTKOWNIK 1 SLUCHACZ: ROZPOCZAL CZYTANIE WIADOMOSCI
ADMIN 0 SLUCHACZ: ROZPOCZAL CZYTANIE WIADOMOSCI
ADMIN 0 SLUCHACZ: OTRZYMAL WIADOMOSC "# UZYTKOWNIK 2 WIADOMOSC B"
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 1
UZYTKOWNIK 1 SLUCHACZ: OTRZYMAL WIADOMOSC "# UZYTKOWNIK 2 WIADOMOSC B"
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 2
UZYTKOWNIK 2 SLUCHACZ: OTRZYMAL WIADOMOSC "# UZYTKOWNIK 2 WIADOMOSC B"
### KOLEJKA WIADOMOSCI:
### # UZYTKOWNIK 2 WIADOMOSC A
### # UZYTKOWNIK 2 WIADOMOSC C
### SOI
### # UZYTKOWNIK 1
LISTONOSZ: WIADOMOSCI DLA ADMIN 0
ADMIN 0 SLUCHACZ: OTRZYMAL WIADOMOSC "# UZYTKOWNIK 2 WIADOMOSC A"
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 1
UZYTKOWNIK 1 SLUCHACZ: OTRZYMAL WIADOMOSC "# UZYTKOWNIK 2 WIADOMOSC A"
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 2
UZYTKOWNIK 2 SLUCHACZ: OTRZYMAL WIADOMOSC "# UZYTKOWNIK 2 WIADOMOSC A"
### KOLEJKA WIADOMOSCI:
### # UZYTKOWNIK 2 WIADOMOSC C
### SOI
### # UZYTKOWNIK 1
LISTONOSZ: WIADOMOSCI DLA ADMIN 0
ADMIN 0 SLUCHACZ: OTRZYMAL WIADOMOSC "# UZYTKOWNIK 2 WIADOMOSC C"
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 1
UZYTKOWNIK 1 SLUCHACZ: OTRZYMAL WIADOMOSC "# UZYTKOWNIK 2 WIADOMOSC C"
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 2
UZYTKOWNIK 2 SLUCHACZ: OTRZYMAL WIADOMOSC "# UZYTKOWNIK 2 WIADOMOSC C"
### KOLEJKA WIADOMOSCI:
### SOI
### # UZYTKOWNIK 1
LISTONOSZ: WIADOMOSCI DLA ADMIN 0
ADMIN 0 SLUCHACZ: OTRZYMAL WIADOMOSC "SOI"
KOMUNIKAT ADMINA: WYKRYTO ZAKAZANE SLOWO
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 1
UZYTKOWNIK 1 SLUCHACZ: OTRZYMAL WIADOMOSC "# ADMIN USUNAL WIADOMOSC"
LISTONOSZ: WIADOMOSCI DLA UZYTKOWNIK 2
UZYTKOWNIK 2 SLUCHACZ: OTRZYMAL WIADOMOSC "# ADMIN USUNAL WIADOMOSC"
### KOLEJKA WIADOMOSCI:
### WCISNIJ KLAWISZ ENTER
```

Powyższy fragment testu obrazuje działanie programu w sytuacji, gdy treść wiadomości jest kontrolowana przez funkcję admina. W tym przypadku zakazanym słowem jest słowo „SOI”. Funkcja admina weryfikuje po kolei wszystkie wysyłane wiadomości. W przypadku, gdy napotka się na zakazane słowo, czyści kolejną wiadomość, a do funkcji użytkowników w miejsce wiadomości niedozwolonej wysyłany zostaje komunikat „ADMIN USUNAL WIADOMOSC”.