Warszawa, 05.11.2019

student: Patryk Jan Sozański

grupa: 2I5

SYSTEMY OPERACYJNE: LABORATORIUM NR 3

RAPORT Z WYKONANEGO ĆWICZENIA

**Treść zadania**

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

Załóż, 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 wulgarnego komunikatu) mogą w dowolnie wybranym przez siebie momencie wyczyścić cala kolejkę wiadomości.

Załóż, 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 semafory:**

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.

1. **Pamięć współdzielona:**

node\_t\* wiadomosci; // Lista wiadomości.

1. **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\_użytkownika] zmienia wartość i informuje o tym użytkownika.

Gdy użytkownik przeczyta wiadomość, powiadomi o tym listonosza semaforem sem\_przeczytane [indeks\_użytkownika].

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.

1. **Funkcje:**

Opis funkcji znajduje się w pliku main.c.

**Testy:**

1. **Ogólne działanie:**

**Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie**

**Obraz zawierający tekst

Opis wygenerowany automatycznie**

Jak widać powyżej początkowo funkcje użytkowników w trybie pracy nadawcy rozpoczynają swoją pracę i wysyłają wiadomości, co powoduje zapełnianie się kolejki wiadomości. Gdy kolejka osiąga maksymalną długość, uruchamia się funkcja listonosza, która przenosi wiadomości do funkcji użytkowników w trypie 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.

1. **Przepełnienia bufora:**

Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

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.

1. **Użytkownik VIP:**

**Obraz zawierający tekst

Opis wygenerowany automatycznie**

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.

1. **Admin:**

**Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie**

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 kolejkę wiadomości, a do funkcji użytkowników w miejsce wiadomości niedozwolonej wysyłany zostaje komunikat „ADMIN USUNAL WIADOMOSC”.