# SPRAWOZDANIE

SOI - Laboratorium

Monitory

Dominik Szaciłowski

#### 1.Treść zadania

Napisz w C++ dla srodowiska Linux, system kolekcjonowania krotkich wiadomosci tekstowych (maks. 64 znakow, ale nie mniej niz 8 znakow). System ma bazowac na synchronizacji dostepu do zasobow wykorzystujący mechanizm monitorow.

Zadaniem budowanego systemu ma byc niezawodne zbieranie od klientow wiadomosci, liczba klientow moze byc duza, ale system musi byc gotowy do obsluzenia minimum 5 klientow.

Klienci - pojedynczy pod-proces lub watek - 'wrzucaja' wiadomosci do systemu, oprocz samej tresci wiadomosci wybieraja priorytet wrzucanej wiadomosc (np.: 0 to zwykly, 1 to priorytetowy).

System moze zbierac wiadomosci tylko w *\_jednym\_* pojemnym buforze. Mechanizm wkladania nowych wiadomosci do tego bufora musi uwzgledniac priorytety. Wszelkie operacje na buforze powinny byc optymalizowane w taki sposob by nie kopiowac niepotrzebnie wiadomosci, oraz zapewnic by czas wkladania wiadomosci oraz czas wyjmowania byly mozliwie jak najkrotsze.

Dodatkowo dla systemu utworzony ma byc pod-proces lub watek 'czytelnik' zebranych wiadomosci. Jego zadaniem jest pobieranie z bufora i przedstawianie wiadomosci tekstowych na konsoli tekstowej. Zaklada sie, ze 'czytelnik' bedzie pobieral wiadomosci z bufora, a w buforze wiadomosci beda juz poukladane zarowno wzgledem priorytetow jaki i czasu ich włozenia.

Przemysl bardzo dokladnie metode automatycznego testowania powstalego systemu. W szczegolnosci zwroc uwage na pokazanie rownoczesnego dzialania wielu procesow (czy watkow) umieszczajacych wiadomosci, w tym takze z roznymi priorytetami oraz wspoldzialanie w tym czasie 'czytelnika'.

## 2. Zarys rozwiązania

Została stworzona klasa Bufor, ktora bazuje na klasie std::list<Message>, gdzie Message to struktura stworzona do obsługi wiadomości. W klasie Bufor został umieszczony wskaźnik na pierwszy element bez priorytetu, aby w każdej chwili bezproblemowo można było umieścić wiadomość piorytetową w odpowiednim miejscu. Klasa BuforWithMonit dziedziczy po klasie Monitor z biblioteki <Monitor.h>. Służy na do połącznenia Monitora I bufora oraz udostępnia operacje addMessage() oraz readMessage(), których mogą użyć odpowiednio producenci jak I konsumenci bez wystąpienia wyścigów krytycznych.

#### 3. Utworzone definicje w programie

MMAX, MMIN – bufor.h, odpowiednio maksymalna I minimalna długość wiadomości EMPTY\_MESS – bufor.h, domyślna warość Message.text EMPTY\_ID – bufor.h, domyślna wartość Message.sender\_id MAX\_SIZE – main.cpp, rozmiar bufora do testów

4. Utworzone struktury I klasy

```
struct Message {
    std::string text = EMPTY MESS;
    int sender_id = EMPTY_ID; //id osoby wysyłającej wiadomość
    bool isPriority = false;
                                   //true oznacza, że wiadomość jest
                                          priorytetowa
    std::chrono::high_resolution_clock::time_point sended_at;
                                                                 //dokładny czas
                                                           wysłania wiadomości
};
class Bufor {
public:
    Bufor(size_t maxSize);
    bool push_back(const Message& message);
                                               //dodanie wiadomości na koniec
                                                      kolejki
    Message pop_front();
                                               //usuniecie pierwszej w kolecje
                                               wiadomosci
    size_t getSize();
```

```
size_t getMaxSize();
private:
    std::list<Message> messageList;
    std::list<Message>::iterator firstNormalMess;
    size_t size = 0;
    size_t maxSize;
};
class BuforWithMonit : private Monitor {
public:
    BuforWithMonit(size_t maxSize);
    bool addMessage(Message message);
                                            //dodanie wiadomosci bez wyścigów,
                                           return false w przypadku niepowodzenia
    Message readMessage();
                                            //odczytanie wiadomośći bez wyścigów
private:
                                           //warunki na pusty I pełny bufor
    Condition empty, full;
    Bufor bufor;
};
5. Testy
      W pliku main.cpp zostały zaiplementowane cztery testy.
```

Po odpaleniu testu 1 otrzymamy nastepujący wynik:

```
Terminal
File Edit View Search Terminal Help
Wybierz test:
0. Wyjscie
1. Zasymuluj wysylanie wiadomosci przez 5 uzytkownikow w jednym momencie.
1. Zasynutom
2. Testuj przepelnienie bufora.
3. Testuj pobranie wiadomosci z pustego bufora.
4. Dodanie wiadomosci o zlej dlugosci.
W tym tescie po 1s 5 uzytkownikow wysle po 1 wiadomosci w tym samym momencie.
Nastepnie zostanie odpalony reader, ktory wyswietli wiadomosci oraz czas w mikrosekundach od wyslania
wiadomosci do dodania jej do bufora
Proba dodania wiadomosci o tresci: Wiadomosc2. ID: 2 Priorytet: 0
Dodano do bufora wiadomosc o tresci: Wiadomosc2. ID: 2 Priorytet: 0
Proba dodania wiadomosci o tresci: Wiadomosc5. ID: 5 Priorytet: 1
Dodano do bufora wiadomosc o tresci: Wiadomosc5. ID: 5 Priorytet: 1
Proba dodania wiadomosci o tresci: Wiadomosc4. ID: 4 Priorytet: 1
Dodano do bufora wiadomosc o tresci: Wiadomosc4. ID: 4 Priorytet: 1
Proba dodania wiadomosci o tresci: Wiadomosc3. ID: 3 Priorytet: 1
Dodano do bufora wiadomosc o tresci: Wiadomosc3. ID: 3 Priorytet: 1
Dodano do bufora wiadomosc o tresci: Wiadomosc3. ID: 3 Priorytet: 1
Proba dodanta wtadomosci o tresci: wtadomosc3. ID: 3 Priorytet: 1
Dodano do bufora wtadomosc o tresci: Wtadomosc3. ID: 3 Priorytet: 1
Proba dodanta wtadomosci o tresci: Wtadomosc1. ID: 1 Priorytet: 0
Dodano do bufora wtadomosc o tresci: Wtadomosc1. ID: 1 Priorytet: 0
Sciagnieto z bufora wtadomosc o tresci: Wtadomosc5. ID: 5 Priorytet: 1
Sciagnieto z bufora wtadomosc o tresci: Wtadomosc4. ID: 4 Priorytet: 1
Sciagnieto z bufora wtadomosc o tresci: Wtadomosc3. ID: 3 Priorytet: 1
Sciagnieto z bufora wtadomosc o tresci: Wtadomosc2. ID: 2 Priorytet: 0
Sciagnieto z bufora wtadomosc o tresci: Wtadomosc2. ID: 2 Priorytet: 0
Sciagnieto z bufora wiadomosc o tresci: Wiadomosc1. ID: 1 Priorytet: 0
Tekst: Wiadomosc5. ID:5 Priorytet:1 Czas(mikrosekundy):84
Tekst: Wiadomosc3. ID.3 Priorytet:1 Czas(Mikrosekundy):140
Tekst: Wiadomosc3. ID:3 Priorytet:1 Czas(Mikrosekundy):190
Tekst: Wiadomosc3. ID:3 Priorytet:0 Czas(Mikrosekundy):15
Tekst: Wiadomosc1. ID:1 Priorytet:0 Czas(Mikrosekundy):253
Wybierz test:
0. Wyjscie
       Zasymuluj wysylanie wiadomosci przez 5 uzytkownikow w jednym momencie.
      Testuj przepelnienie bufora.
Testuj pobranie wiadomosci z pustego bufora.
Dodanie wiadomosci o zlej dlugosci.
```

W pierwszych 10 linijkach otrzymujemy komunikaty o odpowiednio probie dodania jak I dodaniu do bufora wiadomości. Następne 5 linijek informuje nas o ściągnięciu z bufora wiadomości przez readera I zapisanie ich do tablicy. Ostatnie 5 linijek to już tylko wyświetlenie tablicy zgodnie z ich kojenością w buforze. Czas(mikrosekundy) oznacza czas po którym udało się wątkowi wysłać

wiadomość od wspólnie dla wszystkich ustalonego punktu. Co możemy zauważyć to to, że wiadomości zostały odebrane przedewszystkim w kolejności piorytetowej, a póżniej czasowej.

Po odpaleniu testu2 otrzymamy:

```
Terminal

File Sidir View Search Terminal Help

Z. Testuj przepolnośne buforea

Jestuj poznach e kudomosci z pustego bufora.

4. Dodane władomosci z ciej długosci.

W tym tesce jeden uzytkownik probuje wysia z rwadomosci z rzedujnaksymalny rozmiar bufora to 6).

Możeny zalobszemosci ze planika na sciapniacie przynajmiej jedenja władomosci przed dodanika, 7 władomosci zeka jednika, na sciapniacie przynajmiej jednija władomosci przed dodanika, 7 władomosci zeka jednika, na sciapniacie przynajmiej jednija władomosci przed dodanika, 99].

Proba dodania władomosci o tresci: Hadomosci 1D: 1 Prforytet: 0

Proba dodania władomosci o tresci: Hadomosci 1D: 1 Prforytet: 0

Proba dodania władomosci o tresci: Hadomosci 1D: 1 Prforytet: 0

Proba dodania władomosci o tresci: Hadomosci 1D: 1 Prforytet: 0

Proba dodania władomosci o tresci: Hadomosci 1D: 1 Prforytet: 0

Proba dodania władomosci o tresci: Hadomosci 1D: 1 Prforytet: 0

Proba dodania władomosci o tresci: Hadomosci 1D: 1 Prforytet: 0

Proba dodania władomosci o tresci: Hadomosci 1D: 1 Prforytet: 0

Proba dodania władomosci o tresci: Hadomosci 1D: 1 Prforytet: 0

Proba dodania władomosci o tresci: Hadomosci 1D: 1 Prforytet: 0

Proba dodania władomosci o tresci: Hadomosci 1D: 1 Prforytet: 0

Proba dodania władomosci o tresci: Hadomosci 1D: 1 Prforytet: 0

Proba dodania władomosci o tresci: Hadomosci 1D: 1 Prforytet: 0

Dodania do bufora władomosci o tresci: Hadomosci 1D: 1 Prforytet: 0

Bodania do bufora władomosci o tresci: Hadomosci 1D: 1 Prforytet: 0

Bodania do bufora władomosci o tresci: Hadomosci 1D: 1 Prforytet: 0

Bodania do bufora władomosci 1D: 1 Prforytet: 0

Bładomosci d
```

W tym teście sprawdzamy co się dzieje w przypadku przepełnienia bufora. Wysłanych zostaje 7 wiadomości przy maksymalnym rozmiarze 6. Jak widzimy pierwsze 6 wiadomości poprawnie zostaje dodane do bufora zaraz po próbie dodania. 7 wiadomośc jednak najpierw próbuje dostać sie do bufora, akcja zostaje wtrzymana w związku z brakiem miejsca w buforze. Zostaje ona dodana dopiero po tym jak ściągnięto wiadomość 1.

## Po odpaleniu testu3 otrzymamy:

```
Wybierz test:

(a) Wyjscie

(b) Wyjscie

(c) Wyjscie

(c) Zasymuluj wysylanie wiadomosci przez 5 uzytkownikow w jednym momencie.

(c) Testuj przepelnienie bufora.

(d) Testuj pobranie wiadomosci z pustego bufora.

(e) Dodanie wiadomosci o zlej dlugosci.

(e) Wywolujemy odbranie wiadomosci jednak nie wyslalismy wczesniej zadnej.

(e) Wywolujemy odbranie zawiesza sie czakajac na dostarczenie wiadomosci.

(e) Niestety program trzeba sakonczyc wciskajac Ctrl+C

(e) Odpalenie readera.

(e) Bufor jest pusty. Nastepuje zawieszenie readera...
```

Tym razem sprawdzamy reakcję readera na brak wiadomości w buforze. Jak widać został on wtrzymany prawidłowo.

Po odpaleniu testu4 otrzymamy:

```
File Edit View Search Terminal Help

Wybterz test:
0. Wylscte
1. Zasymulug wysylante wtadonosct przez 5 uzytkowntkow w jednyn monencie.
2. Testug przepelniente bufora.
3. Testug przepelniente bufora.
4. Dodanie wtadonosct o ztej długosct.

4 Proba dodania wtadonosct o ztej długosct.

4 Proba dodania wtadonosct o tresci: Min ID: 1 Priorytet: 0

Nie udalo sie wyslac wtadonosci. Wtadonosc za krotka lub za długa.

Proba dodania wtadonosct o tresci: Normalna Wtadonosc ID: 1 Priorytet: 0

Dodano do bufora wtadonosc o tresci: Normalna Wtadonosc ID: 1 Priorytet: 0

Scłagnieto z bufora wtadonosc o tresci: Normalna Wtadonosc ID: 1 Priorytet: 0

Wybterz test:
0. Wylscie
1. Zasymulug wysylante wtadonosci przez 5 uzytkowntkow w jednyn monencie.
2. Testug przepelniente bufora.
3. Testug pobranie wtadonosci z pustego bufora.
4. Dodanie wtadonosci o złej długosci.
```

Test ostani wysyła odpowiednio wiadomość za krótką oraz o normalnej długości. Jak widzimy wiadomośc za krótka nie została dodana oraz został wyświetlony komunikat. Wiadomość o normalnej długości została dodana I odebrana prawidłowo.