

UXP1A

Projekt wstępny

1) Treść zadania:

Napisać wieloprotocowy system realizujący komunikację w języku komunikacyjnym Linda przy wykorzystaniu potoków nazwanych i centralnego procesu koordynującego.

2) Interpretacja treści zadania i założenia:

Zadanie polega na dostarczeniu biblioteki statycznej umożliwiającej procesom komunikację zgodną z językiem Linda:

- a) Procesy powinny komunikować się poprzez wspólną przestrzeń krotek, którą należy zapewnić.
- b) Krotka jest to tabela o dowolnej długości, która może składać się z danych typu string, int oraz float
- c) Należy dostarczyć funkcję, która umożliwi umieszczenie danej krotki w przestrzeni krotek.
- d) Należy dostarczyć funkcję, która pobierze i w sposób atomowy usunie z przestrzeni, krotkę zgodną z zadanym wzorcem.
- e) Wzorzec krotki jest to ciąg znaków, który specyfikuje typu danych poszczególnych pól krotki i może jednocześnie zawierać również następujące warunki logiczne dla danego pola ==, <, <=, >, >=. Wzorzec przyjmuje następujący format <typ danej>:<operator><wartość> lub w przypadku gdy wartość może być dowolna <typ danej>:*
- f) Należy dostarczyć funkcję, która umożliwi analogiczne pobranie krotki, jednak bez usuwania jej z przestrzeni.
- g) Próba pobrania krotki nie pasującej do wzorca powinna zakończyć się zawieszeniem procesu do czasu pojawienia się pasującej krotki, lub przekroczenia maksymalnego czasu oczekiwania.

Jako realizację projektu rozumie się dostarczenie:

- biblioteki udostępniającej API do umieszczania i pobierania krotek.
- procesu sterującego – demona, który będzie przechowywał krotki
- zestawu testów
- dokumentacji końcowej.

Założenia:

- maksymalny rozmiar krotki jest określony statycznie i jest on tak dobrany aby zapis do potoku był atomowy.
- dla danej typu float nie ma warunku ==
- dla stringów operacje logiczne rozumiane są jako porównanie leksykograficzne
- w systemie, w którym uruchamiany będzie program istnieje katalog \tmp\linda, a wszystkie procesy korzystające z biblioteki mają prawo zapisu i odczytu do i z tego katalogu
- biblioteka nie implementuje wielowątkowości i wszystkie funkcje należy zewnętrźnie synchronizować
- timeout jest to czas liczony od otrzymania przez daemona żądania wydania krotki, jeśli przed jego upływem nie pojawi się krotka spełniająca zadany warunek uznaje się wtedy, że klient powinien zostać o tym poinformowany

3) Opis funkcjonalności oraz API

Biblioteka zostanie wykonana przy użyciu metodologii obiektowej i dostarczać będzie następujące API:

```
class LindaClient
{
public:
    //Umieszcza krotkę w przestrzeni
    static void push( QVariantList &record );

    //Pobiera krotkę i usuwa ją z przestrzeni
    static QVariantList pull( QString &format, long timeout );

    //Pobiera krotkę z przestrzeni ale jej nie usuwa
    static QVariantList preview(QString &format, long timeout );
};
```

Funkcja push umieszcza krotkę w przestrzeni krotek, funkcja blokująca do czasu zapisania krotki do potoku.

Funkcje pull oraz preview pobierają krotkę z przestrzeni i są blokujące do czasu dostarczenia do procesu zadanej krotki z serwera, lub upłynięcia timeoutu. Timeout podawany jest w milisekundach i jest to czas po jakim klient zostanie odblokowany jeśli krotka nie pojawi się w programie koordynującym. Podanie wartości timeout równej -1 oznacza czekanie w nieskończoność. Ponad to operacja pull atomowo usuwa pobieraną krotkę z przestrzeni krotek.

4) Podział na moduły

System składał się będzie z następujących modułów:

- moduł klienta (API udostępniane poprzez bibliotekę), który umożliwia operację na krotkach, dokonuje rozbioru formatu wejściowego itd.
- moduł serwera – przechowuje dane w pamięci i zajmuje się przeszukiwaniem itd.
- moduł testujący – zestaw aplikacji komunikujących się przy użyciu zaimplementowanego systemu.

5) Opis protokołu komunikacyjnego.

W systemie występują dwie strony komunikacji – klient czyli dowolny proces używający implementowanej biblioteki, oraz serwer czyli daemon dostarczony razem z biblioteką, który przechowuje krotki. Komunikacja odbywa się poprzez potoki nazwane i możemy wyróżnić w niej 3 przypadki tj. trzy rodzaje żądań, które mogą być kierowane przez klienta do serwera:

a) Umieszczenie krotki w przestrzeni

Klient zapisuje do potoku serwera dane krotki, która ma zostać zapisana tj. jej format oraz wartości danych. Operacja jest blokująca do czasu, gdy w potoku serwera nie będzie wystarczającej ilości miejsca do zapisania tej komendy. Serwer czyta po kolei komendy z swojego potoku, po czym je wykonuje. Wykonanie tej komendy przez serwer ma dwa warianty:

- jeśli nikt nie czeka na krotkę o takiej zawartości to jest ona dodawana do wielozbioru krotek przechowywanych przez serwer.
- jeśli istnieje proces, który zgłosił żądanie pobrania krotki o takich danych to krotka ta jest przekazywana do tego procesu i zależnie od rodzaju żądania jest ona zapisywana do pamięci lub nie.

b) Pobranie i usunięcie krotki z przestrzeni

Klient zapisuje do potoku serwera wzorzec krotki, którą chce pobrać oraz ścieżkę do pliku jego potoku (tworzenie potoku w chwili pierwszej potrzeby jego użycia) oraz maksymalny czas oczekiwania na krotkę. Serwer po odczytaniu takiego żądania sprawdza dostępność krotki:

- jeśli krotka jest dostępna to jest ona usuwana z przestrzeni, potok klienta jest otwierany, krotka zapisywana, a potok zamykany, jeśli wystąpi błąd, należy zaprzestać zapisu, zamknąć potok, a krotka powinna dalej być przechowywana w pamięci
- jeśli krotka nie jest dostępna, uruchamiany jest timer. Jeśli krotka pojawi się przed upływem czasu, jest ona wysyłana do klienta jak wyżej, jeśli czas upłynie to do klienta wysyłana jest wiadomość o przekroczeniu limitu czasu.

c) Pobranie krotki bez usunięcia jej z przestrzeni – czynność analogiczna do poprzedniej ale krotka nie jest usuwana z przestrzeni krotek.

Zatem w całym systemie możemy wyróżnić pięć rodzajów wiadomości o następującej budowie:

- a) umieść krotkę
 - nagłówek informujący o typie komunikatu
 - ciąg znaków mówiący o typach pól krotki
 - pola krotki
- b) pobierz krotkę i usuń z przestrzeni
 - nagłówek informujący o typie komunikatu
 - wyrażenie opisujące warunki, jakie musi spełniać krotka
 - maksymalny czas oczekiwania
- c) pobierz krotkę
 - nagłówek informujący o typie komunikatu
 - wyrażenie opisujące warunki, jakie musi spełniać krotka
 - maksymalny czas oczekiwania
- d) krotka znaleziona
 - nagłówek informujący o typie komunikatu
 - ciąg znaków mówiący o typach pól krotki
 - pola krotki
- e) czas upłynął
 - nagłówek informujący o typie komunikatu



