

PSIW projekt zaliczeniowy

1 stycznia 2022

1. Wstęp

Celem projektu jest stworzenie programu pozwalającego komunikację pomiędzy użytkownikami (czatu) opartego na architekturze rozproszonych serwerów. W ramach realizacji projektu konieczne będzie opracowanie protokołu komunikacyjnego, stworzenie aplikacji klienta i serwera, oraz przygotowanie prostej dokumentacji.

2. Opis środowiska

System czatu opierać się ma na wielu klientach łączących się z siecią serwerów. Każdy klient ma wspierać wysyłanie dwóch typów wiadomości:

- publicznych — widocznych dla wszystkich użytkowników czatu
- prywatnych — skierowanych do konkretnego użytkownika i widocznych tylko przez niego (użytkownicy rozróżniani powinni być po unikalnych nazwach).

Każdy serwer podzielony powinien być na niezależne kanały (pokoje). Wiadomości publiczne wysyłane w ramach kanału powinny być widoczne dla innych uczestników tego kanału, jednak nie powinny dotrzeć do pozostałych użytkowników podłączonych do serwera. Kanały nie mają ograniczać się do jednego serwera, tzn. jeżeli istnieje kanał o tej samej nazwie na dwóch serwerach to stanowi on jeden kanał i wiadomości wysyłane do pierwszego serwera muszą zostać przekazane również do drugiego. Kanał ma być tworzony dynamicznie, w momencie, kiedy podłącza się do niego pierwszy klient. Kanały mają być jednoznacznie identyfikowane przez nazwę. Klient w danym momencie musi należeć co najmniej do jednego pokoju. Każdy klient połączony jest z jednym serwerem, jednak do serwera podłączonych jest wielu klientów (nie więcej niż założona maksymalna liczba klientów). Każdy serwer komunikuje się z wszystkimi innymi dostępnymi serwerami. W ten sposób klienci komunikują się z klientami podłączonymi do dowolnego serwera.

3. Wymagania ogólne

- (a) Wszystkie komponenty systemu napisane są w języku C.
- (b) Komunikacja między aplikacjami odbywa się wyłącznie za pomocą mechanizmu kolejek komunikatów.
- (c) Praca jest w 100% samodzielna.
- (d) Termin oddania projektu: nie później niż przedostatnie zajęcia (**17.01.2022**)

4. Komponenty systemu

Klient

Aplikacja kliencka służy jako interface dla użytkownika. Interface musi być czytelny i wygodny w obsłudze. Funkcje realizowane przez klienta:

- przyjęcie nazwy użytkownika (nazwy muszą być unikalne),
- zarejestrowanie użytkownika w zadanym serwerze,
- wyświetlanie listy użytkowników zalogowanych na serwerze,
- wyświetlenie listy zarejestrowanych kanałów (pokoje)
- wyświetlenie listy użytkowników zapisanych do pokoju
- zapisanie się użytkownika do pokoju
- wypisanie się użytkownika z pokoju
- wyświetlanie czatu prowadzonego na wskazanym kanale (każda wiadomość to czas wysłania, nazwa użytkownika i treść) — wyświetlane jest ostatnich 10 wiadomości,
- możliwość wysłania wiadomości na wskazany kanał
- możliwość odebrania wiadomości z kanału
- możliwość odebrania wiadomości prywatnej (wyraźnie rozróżnionej),
- możliwość wysłania wiadomości prywatnej do dowolnego użytkownika (w ramach pokoju do których należy wysyłający użytkownik)
- możliwość wylogowania z systemu
- odpowiadanie na komunikaty *heartbeat* serwera.

Klient porozumiewa się wyłącznie z serwerem, z którym się połączył. Całość komunikacji odbywa się za pomocą kolejek komunikatów. Odświeżanie interface-u odbywa się automatycznie. Adres serwera (numer kolejki komunikatów), z którym klient ma się połączyć, podawany jest przez użytkownika w trakcie uruchamiania lub działania aplikacji (nie może być wkompiłowany w wersję binarną aplikacji).

Serwer Serwer realizuje większość logiki systemu i odpowiada za komunikację z użytkownikami oraz innymi serwerami. Użytkownik, który uruchamia serwer przekazuje mu jako parametr wejściowy nazwę pliku konfiguracyjnego, w którym zapisana jest informacja o identyfikatorach kolejek komunikatów na których nasłuchują pozostałe serwery. Każdy serwer może obsłużyć do 5 klientów. Wszystkie serwery nawiązują łączność z wszystkimi innymi serwerami. Czynności wykonywane przez serwer:

- rejestrowanie i wyrejestrowanie użytkowników
- przesłanie na żądanie klienta listy użytkowników / kanałów / użytkowników korzystających z kanału
- rejestrowanie nowego kanału
- zapisanie/wypisanie użytkownika do/z kanału
- odbiór i przesłanie odebranych wiadomości do wszystkich swoich użytkowników korzystających z danego kanału
- odbiór i przesłanie wiadomości prywatnych do wskazanego użytkownika na tym serwerze
- weryfikacja obecności klientów (mechanizm heartbeat)
- odbiór wiadomości i wysyłanie ich do wszystkich innych serwerów o ile są na nich odpowiednie kanały

- przesłanie informacji o zakończeniu pracy do pozostałych serwerów (skutkuje to usunięciem informacji o serwerze na pozostałych serwerach)

Klienci nie wykryci przez mechanizm *heartbeat* nie mogą komunikować się z serwerem bez ponownego zalogowania. Serwer powinien odrzucać ich komunikaty, informując o błędzie (nawet, jeśli de facto nie przerwali działania).

5. Ułatwienia projektu

Kosztom obniżenia maksymalnej oceny z projektu możliwa jest rezygnacja z części funkcjonalności:
Projekt na ocenę 4.5

- Pominięcie mechanizmu *heartbeat*

Projekt na ocenę 4.0

- Pominięcie komunikacji pomiędzy wieloma serwerami — zakładamy, że istnieje tylko 1 serwer, więc nie ma potrzeby przy jego uruchomieniu podawania nazwy pliku konfiguracyjnego, komunikacja tylko pomiędzy użytkownikami dołączonymi do tego serwera

Projekt na ocenę 3.5

- Pominięcie mechanizmu *heartbeat*
- Pominięcie komunikacji pomiędzy wieloma serwerami — zakładamy, że istnieje tylko 1 serwer, więc nie ma potrzeby przy jego uruchomieniu podawania nazwy pliku konfiguracyjnego, komunikacja tylko pomiędzy użytkownikami dołączonymi do tego serwera

Projekt na ocenę 3.0

- Pominięcie mechanizmu *heartbeat*
- Pominięcie komunikacji pomiędzy wieloma serwerami — zakładamy, że istnieje tylko 1 serwer, więc nie ma potrzeby przy jego uruchomieniu podawania nazwy pliku konfiguracyjnego, komunikacja tylko pomiędzy użytkownikami dołączonymi do tego serwera
- pominięcie wyświetlania ostatnich 10 wiadomości czatu

6. Forma oddania projektu

- Wykonany samodzielnie projekt, należy przesłać mailem.
- Przesłane pliki powinny zawierać:
 - pełne źródła aplikacji, kompilujące się bez ostrzeżeń
 - poszczególne pliki powinny mieć następujące nazwy: `inf?????.s.c` — serwer `inf?????.k.c` — klient, `inf?????.*.ch` — pozostałe pliki nagłówkowe
 - plik tekstowy README zawierający: instrukcję kompilacji, instrukcję uruchomienia, krótki opis zawartości poszczególnych plików *.c
 - plik tekstowy PROTOCOL opisujący protokół komunikacji między komponentami projektu, w szczególności dokładny opis używanych struktur
-
- Podstawą oceny jest terminowe oddanie projektu zgodnego z powyższą specyfikacją.
- Oddanie projektu po terminie oznacza obniżenie oceny o 0.5 za każdy rozpoczęty tydzień zwłoki
- Wykrycie plagiatu skutkuje automatyczną oceną niedostateczną dla wszystkich zaangażowanych