

# IPC Short Message Service — projekt zaliczeniowy

20 stycznia 2020

Autorem zadania jest dr hab. inż. Anna Kobusińska

## 1 Zasady ogólne

Celem projektu jest stworzenie programu pozwalającego na komunikację (chat) użytkowników, odbywającą się poprzez wymianę wiadomości pomiędzy nimi. Tworzona aplikacja powinna wykorzystywać mechanizm kolejek komunikatów. W ramach realizacji projektu konieczne jest stworzenie 2 programów: klienta i serwera. Każdy klient może wysyłać i otrzymywać wiadomości do/od jednego z użytkowników systemu, lub do/od grupy składającej się z wielu użytkowników. W wymianie wiadomości pomiędzy klientami zawsze pośredniczy serwer (bezpośrednia komunikacja pomiędzy klientami jest zabroniona). Otrzymywane wiadomości wyświetlane są na ekranie.

## 2 Funkcjonalność szczegółowa

Funkcje realizowane przez klienta:

1. Obsługa konta użytkownika:
  - zalogowanie
  - wylogowanie
  - podgląd listy:
    - zalogowanych użytkowników
    - zapisanych do danej grupy tematycznej
2. Obsługa grup tematycznych:
  - zapisanie się do grupy
  - wypisanie się z grupy
  - podgląd listy dostępnych grup

### 3. Wiadomości:

- wysłanie wiadomości do grupy
- wysłanie wiadomości do użytkownika
- odebranie wiadomości

Funkcje realizowane przez serwer:

- tworzenie użytkowników i grup na podstawie pliku konfiguracyjnego (w gotowym rozwiązaniu należy dostarczyć plik konfiguracyjny tworzący konta dla 3 grup (group[1-3]) i dziewięciu użytkowników (test[1-9] )
- przesyłanie wiadomości otrzymanych od klienta/klientów do adresata/adresatów
- potwierdzenia wykonania/odrzućenia żądania przez serwer

Funkcjonalność dodatkowa:

1. zablokowanie możliwości odbioru wiadomości (z grupy, od użytkownika) i informacja o nieudanych próbach logowania / blokowanie konta po  $k$  próbach
2. wprowadzenie priorytetów wiadomości
3. realizacja serwera w sposób współbieżny (serwer składa się z wielu procesów komunikujących się za pomocą pamięci współdzielonej i semaforów)

## 3 Wymagania ogólne

1. **Termin oddania projektu: ostatnie zajęcia (07.02.2020),**
2. Wszystkie komponenty systemu napisane są w języku C.
3. Komunikacja między aplikacjami odbywa się wyłącznie za pomocą kolejek komunikatów
4. Praca jest w 100% samodzielna.

## 4 Forma zaliczenia

1. **Projekt bezbłędny, bez dodatkowej funkcjonalności to max. ocena 4,5. Ocena 5,0 będzie otrzymana po zaimplementowaniu jednego z punktów 1-2 opisanych w funkcjonalności dodatkowej.**
2. W przypadku zaimplementowania dodatkowej funkcjonalności w punkcie 3, otrzymują państwo +0,5 oceny z kolokwium (o ile zostało ono zaliczone i nie otrzymało się oceny 5.0)

3. Wykonany **samodzielnie** projekt, skompresowany w jednym archiwum, należy przesłać na adres prowadzącego zajęcia w terminie przez niego wskazanym.

Archiwum, nazwane `imie.nazwisko.indeks.tar.gz`, zawierać musi:

- pełne źródła aplikacji, kompilujące się bez ostrzeżeń (flaga `-Wall` kompilatora)
  - poszczególne pliki powinny mieć następujące nazwy:
    - \* `inf????_s.c` — serwer
    - \* `inf????_k.c` — klient
    - \* `inf????_*.ch` — pozostałe pliki nagłówkowe
- skrypt do kompilacji lub plik `Makefile`
- 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
- archiwum nie powinno zawierać zbędnych plików binarnych (produktów kompilacji).

4. Podstawą oceny jest **terminowe oddanie projektu zgodnego z powyższą specyfikacją**.

- Oddanie projektu z niepełną funkcjonalnością lub błędami skutkować będzie obniżeniem oceny końcowej.
- Oddanie projektu po terminie oznacza obniżenie oceny o 0.5 za każdy rozpoczęty tydzień zwłoki.

5. Wykrycie plagiatu skutkuje automatyczną oceną niedostateczną dla wszystkich zaangażowanych.