

Wydział informatyki Politechniki Białostockiej Przedmiot: Systemy Operacyjne	Data oddania: 4.05.2023
Temat: Projekt Komunikator użytkowników systemów  Zespół: 1. Tomasz Marchela 2. Maciej Maksymowicz	Prowadzący:  dr inż. Marcin Koźniewski

## 1. Sposób uruchamiania

Po skompilowaniu programu przy użyciu przygotowanego pliku makefile musimy uruchomić serwer: `./komunikator -s`. Od tego momentu serwer nasłuchuje logowanie i akcje użytkowników. Nie możemy na raz uruchomić dwóch serwerów, Po uruchomieniu serwera możemy w innej konsoli zalogować użytkownika `./komunikator -l nazwa`. Możemy także przed opcją `-l` dodać opcję `-d` która pozwoli na wysyłanie plików do nas przykładowo `./komunikator -d "ścieżka" -l nickname`.

## 2. Sposób testowania projektu

Testowanie projektu polega na obserwowaniu logów systemowych oraz komunikatów wypisywanych na standardowe wyjście oraz wyjście błędu. Pojawiają się tam informacje o tym czy dana akcja się udała lub nie. Oczywiście wszystkie komunikaty nie pokazują się zawsze w celu odbudowywania programu trzeba wpisać dyrektywę `#DEFINE DEBUG` wtedy wszystkie ważne komunikaty będą wypisywane na ekran.

## 3. Zaimplementowane funkcjonalności

- Wysyłanie wiadomości pomiędzy użytkownikami
- Kopiowanie plików pomiędzy użytkownikami
- Logowanie użytkowników
- Wylogowywanie użytkowników za pomocą `ctrl+c`
- Ważne akcje są przesyłane do logów systemowych, część akcji na standardowe wyjście błędów które nie przerywają programu na standardowe wyjście.
- Wyłączenie każdego klienta w przypadku wyłączenia serwera
- Wypisywanie wszystkich aktywnych użytkowników

## 4. Opis Funkcji

### **user\_read()**

Funkcja na początku działania deklaruje zmienne odpowiadające za treść komendy, nazwę wysyłającego, nazwę odbierającego, treść wiadomości oraz bufer do czytania z potoku fifo. Następnie w pętli while otwieramy plik potoku klienta i odczytujemy z niego potencjalne komendy (`/file` oraz `/msg`). Przy odczytywaniu komend posługujemy się funkcjami z pliku `frames.c` które odpowiednio dzielą przeczytaną ramkę. Jeśli wykryta jest opcja `/msg` na ekran wypisywany jest wysyłający oraz treść wiadomości. W przypadku `/file` rozpoczynamy pobieranie pliku przez klienta. Obsługiwane są wszelkie błędy związane z złą treścią komendy.

### **user sending()**

Funkcja deklaruje zmienne takie jak funkcja `user_read` plus ramkę która będzie przetrzymywać całe polecenie. Następnie odczytywane jest standardowe wejście. Przy odczytaniu wejścia dzielone jest za pomocą odpowiednich funkcji z `frames.c` i dodawane do ramki następnie wysyłana jest do potoku serwera za pomocą funkcji `user_frame_send`. Obsługiwane są polecenia `/msg` oraz `/file`. W razie powodzenia lub niepowodzenia wysyłane są odpowiednie wiadomości do logów systemowych

5. Funkcje zaczerpnięte z internetu

<https://msalamon.pl/odkryj-co-potrafi-wbudowany-w-zestawy-stm32-nucleo-programator-st-link/>

Materiały udostępnione na cezie.