

# Systemy operacyjne

## Labortorium 7

Przemysław Ziaja  
303187

April 30, 2020

### Zadanie 0

Trzeba było otworzyć kolejke przy pomocy `O_NONBLOCK`, aby program nie czekał na pojawienie się klienta oraz utworzyć deskryptor wpisujący do tej kolejki (nie jest on nigdzie wykorzystywany w programie) aby program nie rzucał błędów. Analogiczne komendy znajdują się w kodzie klienta, w kolejce która czyta wiadomości od serwera.

W ostatnim podpunkcie nie udało mi się namierzyć błędu. Jedynie zwiększając wartość `MESSAGE_BUF_SIZE` do 10000 udało mi się odpalić program, ale zaraz po wpisaniu wiadomości pojawia się błąd. Potem zmieniłem `MESSAGE_BUF_SIZE` na domyślny systemu oraz wpisałem to do wartości funkcji `send` i `receive`, ale dalej to nie rozwiązało problemu.

### Zadanie 1

W tym zadaniu wystarczyło zsyntetyzować program z podpunktu 1. zad 0 z programem `Eliza`. Ponieważ nie udało mi się dokładnie zgłębić działania `Elizy`, tzn. jaki jest algorytm udzielania odpowiedzi, dlatego postanowiłem całe wyjście `STDOUT` z serwera przekierować do pliku i utworzyć deskryptor czytający z pliku do serwera jako. Klientem jest lekko zmodyfikowany program klienta z tego samego podpunktu.

### Zadanie 2

Sytuacja wygląda podobnie jak w poprzednim zadaniu. Wystarczyło przerobić podpunkt 2 z zadania 0. Kod w kliencie znacząco się uprościł, ponieważ nie trzeba na zmianę wysyłać informacji, tylko wysyłać i nasłuchuje wyników. Natomiast po stronie serwera trzeba było obsłużyć kilku klientów, co przełożyło się na dodanie tablicy z klientami oraz kilku petli `for` do zwrotu informacji do klientów.