

# Sprawozdanie z zadania na SK2

## Zdalne wyłączanie komputerów

### 1 Protokół komunikacji

1. Klient porozumiewa się z serwerem z użyciem protokołu TCP - zależy mi na niezawodności przesłania uprawnień, a także przesłania na sprzęt klienta odpowiedniej komendy wyłączającej sprzęt
2. Klient po zainicjowaniu połączenia przesyła nasłuchującemu serwerowi - a ściślej wątkowi, który z tego serwera wychodzi bufor tekstu w następującym formacie:  
komenda  
uid  
gid  
wynik ls -lnH \$(which shutdown)  
wynik ls -lnH \$(which init)  
czas  
Gdzie:  
komenda - jedna cyfra - 0, jeśli shutdown, 1, jeśli reset  
uid, gid - user id, group id, którymi posługuje się aplikacja klienta  
2 kolejne linie są użyteczne do sprawdzenia uprawnień komend, którymi mogą wyłączyć system  
czas - za ile minut ten komputer ma zostać wyłączony? (w minutach)
3. Serwer po parsingu tych danych wysyła klientowi bufor tekstu, w którym znajduje się komenda, którą klient ma wykonać na systemie - w szczególności komentarz, jeśli klient jest pozbawiony odpowiednich praw.

### 2 Struktura serwera

1. Serwer tworzy socketa, manipuluje jego opcjami(setsockopt), wiąże tego socketa z portem 1234 i adresem, następnie zaczyna nasłuchiwać na tym sockecie do 5 połączeń i je akceptować - nawiązane połączenia obsługuje funkcja handleConnection.
2. Funkcja handleConnection przekazuje nowemu wątkowi wykonującemu funkcję ThreadBehavior deskryptor, z którym może się komunikować i wraca do maina.
3. Funkcja ThreadBehavior odbiera od klienta bufor tekstu, wysyła go do funkcji parsującej parse\_wisdom, jej wynik (zamknięty w strukturze wisdom) jest przekazywany do funkcji grant\_wisdom tworzącej stringa przesyłanego do klienta; następnie wychodzę z wątku.
4. Funkcja parse\_wisdom operując na danych przesyłanych przez klienta zwraca informację, do jakich komend ma on uprawnienia a także jaki jest czas, za jaki komputer ma zostać wyłączony/zresetowany.
5. Funkcja grant\_wisdom operując na wyniku funkcji parsującej zwraca komendę, która zostanie wysłana do klienta.

## 3 Struktura klienta

- 1.