

## Informacje ogólne

- Protokół opiera się na jednej kolejce publicznej i n kolejek prywatnych, gdzie n to liczba aktywnych klientów
- Maksymalna liczba zalogowanych użytkowników to 16
- Maksymalna liczba utworzonych grup tematycznych to 16

## Poprawność danych

- Nick oraz nazwa grupy musi mieć długość od 1 do 16 znaków;
- Wiadomość musi zawierać treść nie większą niż 256 znaków.
- Użytkownik może należeć maksymalnie do 4 grup tematycznych jednocześnie.

## Struktura komunikatu

```
struct comm {  
    int id; // pid  
    int commType; // numer komunikatu  
    char nick[16];  
    char group[16];  
    char mess[256]; // treść wiadomości  
}
```

```
struct msgbuf {  
    long msgRecipient;  
    struct comm msgContent;  
};
```

## Struktury danych serwera

- 16 elementowa tablica zawierająca nazwy istniejących nicków użytkowników
- 16 elementowa tablica zawierająca nazwy istniejących grup tematycznych
- 16 elementowa tablica 2 wymiarowa, w której przechowywane są tablice 2-elementowe, kojarzące PID z nickiem użytkownika (tablica zalogowanych użytkowników)
- 64 elementowa tablica 2 wymiarowa, w której przechowywane są tablice 2-elementowe, kojarzące nick z przynależnością do grupy (tablica użytkowników w grupie)

## Typy komunikatów

Do każdego komunikatu należy wypełnić pola *pid* i *comm\_type*

Numer komunikatu	Opis	Dodatkowe wymagane pola	Nadawca
-1	błędna odpowiedź od serwera	mess	serwer
0	prawidłowa odpowiedź od serwera	mess	serwer
1	rozpoczęcie komunikacji		klient
2	zalogowanie	nick	klient
2	zalogowanie	nick	serwer

3	wylogowanie		klient
4	lista użytkowników		klient
4	lista użytkowników	mess	serwer
5	lista użytkowników w grupie	group	klient
5	lista użytkowników w grupie	group, mess	serwer
6	zapis do grupy	group	klient
7	wypisanie z grupy	group	klient
8	lista grup		klient
8	lista grup	mess	serwer
9	wiadomość do użytkownika	nick, mess	klient
9	wiadomość do użytkownika	nick, mess	serwer
10	wiadomość do grupy	group, mess	klient
10	wiadomość do grupy	nick, group, mess	serwer

## Komunikaty

- **odpowiedź z błędem**

```
pid = getpid();
comm_type = -1;
mess = "[komunikat]"
```

- **odpowiedź sukces**

```
pid = getpid();
comm_type = 0;
mess = "[komunikat]"
```

- **rozpoczęcie komunikacji**

```
pid = getpid();
comm_type = 1;
```

- **zalogowanie (klient)**

```
pid = getpid();
comm_type = 2;
nick = "nick podany przez użytkownika";
```

- **zalogowanie (serwer)**

```
pid = 0;
comm_type = 2;
nick = "nick podany przez użytkownika";
```

- **wylogowanie**

```
pid = getpid();
comm_type = 3;
```

- **lista użytkowników**

```
pid = getpid();
```

comm\_type = 4;

- **lista użytkowników (serwer)**

pid = 0;

comm\_type = 4;

mess = "lista użytkowników";

- **lista użytkowników w grupie (klient)**

pid = getpid();

comm\_type = 5;

group = "nazwa grupy";

- **lista użytkowników w grupie (serwer)**

pid = 0;

comm\_type = 5;

group = "nazwa grupy";

mess = "lista użytkowników";

- **zapis do grupy**

pid = getpid();

comm\_type = 6;

group = "nazwa grupy";

- **wypisanie z grupy**

pid = getpid();

comm\_type = 7;

group = "nazwa grupy";

- **lista grup (klient)**

pid = getpid();

comm\_type = 8;

group = "nazwa grupy";

- **lista grup (serwer)**

pid = 0;

comm\_type = 5;

mess = "lista grup";

- **wysłanie wiadomości do użytkownika (klient)**

pid = getpid();

comm\_type = 9;

nick = "nick odbiorcy";

mess = "komunikat";

- **wysłanie wiadomości do użytkownika (serwer)**

pid = 0;

comm\_type = 9;

nick = "nick nadawcy";

mess = "komunikat";

- **wysłanie wiadomości do grupy (klient)**

pid = getpid();

comm\_type = 10;

nick = "nick nadawcy";

```
group = "nazwa grupy";  
mess = "komunikat";
```

- **wysłanie wiadomości do grupy (serwer)**

```
pid = 0;  
comm_type = 10;  
nick = "nick nadawcy";  
group = "nazwa_grupy";  
mess = "komunikat";
```

- **prośba o pomoc**

```
pid = getpid();  
comm_type = 11;
```

## Komendy klienta

- zalogowanie: login
- wylogowanie: logout
- lista użytkowników: userlist
- lista użytkowników w grupie: groupuserlist
- zapis do grupy: join
- wypisanie z grupy: leave
- lista grup: grouplist
- wiadomość do użytkownika: send
- wiadomość do grupy: sendgroup