

Realizzare il seguente servizio di trasferimento file. La rete è formata da un server e da vari client, tutti uguali.

- Il client si registra sul server, fornendo una *username*, una lista di file disponibili per la copia ed una porta di ascolto (TCP, vedi oltre)
- Una volta registrato, un client può chiedere al server la lista degli altri utenti registrati e la lista dei file condivisi da ciascun client (identificato con *username*).
- Quando vuole scaricare un file, il client chiede al server le informazioni associate allo *username* con cui vuole connettersi, ovvero l'indirizzo IP e la porta TCP di ascolto (il trasferimento avviene su un canale TCP)
- Il download avviene connettendosi direttamente col client, fornendo il nome del file da scaricare.
- Periodicamente il server interroga i client, per:
 - a. verificare che sono ancora attivi
 - b. aggiornare la lista dei file condivisi
- Tutti i messaggi tra client e server sono UDP.

Usare C o C++. Niente grafica.

Struttura per Autenticazione

```
typedef struct {
```

```
    char ip[17];
```

```
    char username[20];
```

```
} CLIENT;
```

```
void registra(char* username, char* ip, int porta) {
```

```
    File *fp;
```

```
    fp = fopen("elencocontenuti.txt", "a+");
```

```
    if (fp == NULL) //Controlla
```

```
        printf("errore apertura file\n");
```

```
        exit(-1);
```

```
}
```

```
    fprintf(fp, "%s %s %d\n", ip, username, porta) // Scrivere nel file ip, user, porta
```

```
    fclose(fp) //chiusura
```

```
}
```

```
bool login(char* username, char* ip, int porta) {
```

```
    File *fp;
```

```
    Client c;
```

```
    f = fopen("elencocontenuti.txt", "r");
```

```
    if (f == NULL)
```

```
        printf("errore apertura file\n");
```

```
        exit(-1);
```

```
}
```

```
    rewind(f);
```

```
    char buffer[1000];
```

```
    char *iptemp, *usertemp;
```

```
while (fscanf(fp, "%s[\n]\n", buffer) == 1) {
```

```
    iptemp = strtok(buffer, " ");
```

```
    osertemp = strtok(NULL, " ");
```

```
    printf("%s = %s %s = %s\n", iptemp, ip, osertemp, username);
```

```
    if (strcmp(username, osertemp, strlen(username) == 0) && strcmp(iptemp, ip, strlen(ip)) {  
        return true;
```

```
    }
```

```
    fclose(fp);  
    return false;
```

→ Vedlams se i Registrat

```
void registra_file_client(char *ip, char *files) {
```

```
    File *fp;
```

```
    File *fp_tmp;
```

```
    Client c;
```

```
    fp = fopen("denoedenti.txt", "r");
```

```
    fp_tmp = fopen("denoedtmp.txt", "w+");
```

```
    if (f == NULL || fp == NULL)
```

```
    {  
        print("Errore file\n");  
        exit(-1);
```

```
    }
```

```
    char buffer[1000];
```

```
    char msg[1000];
```

```
    char *iptmp;
```

```
    while (fscanf(fp, "%s[\n]\n", buffer) == 1) {
```

```
        strcpy(msg, buffer);  
        iptemp = strtok(msg, " ");
```

```

    if (strcmp(ip_tmp, ip) == 0) {
        fprintf(fp, "%s %s\n", buffer, files);
    } else {
        fprintf(fp_tmp, "%s\n", buffer);
    }
}

```

```

fclose(fp);
fclose(fp_tmp);
remove("elenco utenti.txt");
rename("elenco tmp.txt", "elenco utenti.txt");
}

```

```

void * manda_lista_utenti() {

```

```

    File *fp;
    Client c;
    fp = fopen("elenco utenti.txt", "r");
    if (fp == NULL) {
        printf("Errore Apertura file\n");
        exit(-1);
    }

```

```

    char buffer[1000];
    char msg[1000];
    char *ip, *username;
    char manda[1000];
    strcpy(manda, "");

```

```

    while (fscanf(fp, "%s\n", buffer) == 1) {
        strcpy(msg, buffer);
        ip = strtok(buffer, " ");
        username = strtok(NULL, " ");
        strcat(manda, username);
        strcat(manda, " ");
    }

```

```
printf ("%s\n", manda);  
fclose(fp);  
return strdup (manda);  
}
```

```
char* dato_utente_torna_lista (char* utente) {  
    File *fp;  
    Client *c;  
    fp = fopen ("elenco utenti.txt", "r");  
    if (fp == NULL) {  
        printf ("Errore apertura file);  
        exit (-1);  
    }  
    char buffer [1000];  
    char manda [1000];  
    char *ip, *username, *ip, *resto;  
    strcpy (manda, "");  
    while (fscanf (fp, "%s [^\n]\n", buffer) == 1) {  
        ip = strtok (buffer, " ");  
        username = strtok (NULL, " ");  
        p = strtok (NULL, " ");  
        resto = strtok (NULL, "\n");  
        if (strcmp (username, utente) == 0) {  
            strcat (manda, resto);  
        }  
    }  
    fclose (fp);  
    return strdup (manda);  
}
```