

```
/* Svolgimento Compito 1, 3, 5, 7 - 21/12/05 */
```

```
#include <stdio.h>
#include <rpc/rpc.h>
#include "opfile.h"

main(int argc, char *argv[])
{
    CLIENT *clnt; /*gestore di protocollo*/

    int *ris1;
    ListaStanze *ris2;
    AggStanza as;
    AggUtente au;
    char *server;
    char *stanza, *utente, tipo;
    char ok[5];

    stanza=(char*)malloc(20);
    utente=(char*)malloc(20);

    if(argc !=2)
    {
        fprintf(stderr,"uso: %s nomehost\n",argv[0]);
        exit(1);
    }

    clnt=clnt_create(argv[1],OPFILEPROG,OPFILEVERS,"udp");

    if(clnt == NULL)
    {
        clnt_pcreateerror(argv[1]);
        exit(1);
    }

    printf("richieste servizio fino a fine file\n");
    printf("operazioni: AS=aggiungi stanza,EU=elimina utente, AU=aggiungi utente,\n");
    printf("VU=visualizza utente, ES=elimina stanza: ");

    while(gets(ok))
    {
        printf("Richiesto servizio: %s\n", ok);

        if( strcmp(ok, "AS")==0 )
        {
            printf("inserisci il nome della stanza che vuoi aggiungere: \n");
            gets(stanza);
            printf("inserisci il tipo della stanza che vuoi aggiungere (M)ulticast o (P)unto-a-punto: \n");
            tipo=getchar();
            // consumo il fine linea
            gets(utente); // usato come buffer da scartare
            while(tipo!='P' && tipo!='M'){
                if( tipo == EOF ){
                    printf("Richiesta terminazione, esco...");
                    clnt_destroy(clnt);
                    free(stanza);
                    free(utente);
                    exit(0);
                }
                printf("Tipo errato!! Inserisci il tipo della stanza che vuoi aggiungere (M)ulticast o (P)unto-a-punto: \n");
            }
        }
    }
}
```

```

        tipo=getchar();
        // consumo il fine linea
        gets(utente);
    }

    strcpy(as.nomeStanza, stanza);
    as.tipo=tipo;

    ris1=aggiungi_stanza_1(&as,clnt);

    if(ris1==(int *)NULL)
    {
        clnt_perror(clnt,"call failed");
    }
    else if(*ris1==-1)
    {
        printf("non riesco ad aggiungere la stanza\n");
    }
    else
    {
        printf("aggiungi stanza riuscita!\n",*ris1);
    }
} // AS

//richiesta la verifica della presenza di un titolo
else if( strcmp(ok, "EU")==0 )
{
    printf("inserisci il nome dell'utente che vuoi eliminare: \n");
    gets(utente);

    ris2=elimina_utente_1(&utente,clnt);

    if(ris2==(ListaStanze *)NULL)
    {
        clnt_perror(clnt,"call failed");
    }
    else if(ris2->ris==-1)
    {
        printf("problemi durante elimina utente\n");
    }
    else
    {
        int j, i;
        printf("le stanze liberate sono %i:\n", ris2->ris);
        for(j=0;j<=ris2->ris;j++)
        {
            printf(" %s di tipo %c\n",ris2->elenco[j].nomeStanza, ris2->elenco[j].tipo);
            // se libere stampa "L"
            for(i=0;i<K;i++)
                printf("%s \t\t", ris2->elenco[j].utente[i].nome);
            printf("\n\n");
        }
    }
} // EU

//richiesta l'aggiunta di un utente
else if( strcmp(ok, "AU")==0 )
{
    printf("inserisci il nome della stanza che dove vuoi aggiungere l'utente: \n");
    gets(stanza);
    printf("inserisci il nome dell'utente: \n");
    gets(utente);

    strcpy(au.nomeStanza, stanza);
    strcpy(au.nomeUtente, utente);

```

```

    ris1=aggiungi_utente_1(&au,clnt);

    if(ris1==(int *)NULL)
    {
        clnt_perror(clnt,"call failed");
    }
    else if(*ris1== -1)
    {
        printf("non riesco ad aggiungere l'utente\n");
    }
    else
    {
        printf("aggiungi utente riuscita!\n",*ris1);
    }
} // AU

//richiesta l'eliminazione di una stanza
else if( strcmp(ok, "ES")==0 )
{
    printf("inserisci il nome della stanza che vuoi eliminare: \n");
    gets(stanza);

    ris1=elimina_stanza_1(&stanza,clnt);

    if(ris1==(int *)NULL)
    {
        clnt_perror(clnt,"call failed");
    }
    else if(*ris1== -1)
    {
        printf("problemi durante l'eliminazione\n");
    }
    else
    {
        printf("eliminazione stanza riuscita!\n",*ris1);
    }
} // ES

//richiesta la visualizzazione di un utente
else if( strcmp(ok, "VU")==0 )
{
    printf("inserisci il nome dell'utente che vuoi visualizzare: \n");
    gets(utente);

    ris2=visualizza_utente_1(&utente,clnt);

    if(ris2==(ListaStanze *)NULL)
    {
        clnt_perror(clnt,"call failed");
    }
    else if(ris2->ris== -1)
    {
        printf("problemi durante l'esecuzione della visualizza utente\n");
    }
    else
    {
        int j, i;
        printf("ecco le stanze da visualizzare sono %i:\n", (ris2->ris)+1);
        for(j=0;j<=ris2->ris;j++)
        {
            printf(" %s di tipo %c\n",ris2->elenco[j].nomeStanza, ris2->elenco[j].tipo);
            // se libere stampa "L"
            for(i=0;i<K;i++)
                printf("%s \t\t", ris2->elenco[j].utente[i].nome);
        }
    }
}

```

```
        printf("\n\n");
    }
}
} // VU
else printf("Servizio scelto non valido\n");

printf("richieste servizio fino a fine file\n");
printf("operazioni: AS=aggiungi stanza,EU=elimina utente, AU=aggiungi utente,\n");
printf("VU=visualizza utente, ES=elimina stanza: ");

} //while

clnt_destroy(clnt);
free(stanza);
free(utente);
printf("Esco dal client");
}
```