

14 Dicembre 2021



Bernardini Claudio
Corsetti Luca
Giardina Gianluca
Straccali Leonardo

Esercitazione 9

Gestione votazione FattoreX

Obiettivi esercitazione

Hard skills

1

Invocazione di procedure remote
mediante RPC di SUN

2

Implementazione funzioni gestione
votazioni

Soft skills

1

Collaborazione per raggiungere un fine comune

2

Organizzazione del proprio lavoro in team

2

fattorex.x

```
const MAXLENGTHSTRING = 256;
const NUMGIUDICI = 3;

struct Giudice{
    string nomeGiudice <MAXLENGTHSTRING>;
    int punteggioTot;
};

struct Classifica{
    Giudice classificaGiudici [NUMGIUDICI];
};

struct Voto{
    string nomeCandidato <MAXLENGTHSTRING>;
    char tipoOp;
};

program VOTAFATTOREX {
    version VOTAFATTOREXVERS {
        Classifica CLASSIFICA_GIUDICI() = 1;
        int ESPRIMI_VOTO(Voto) = 2;
    } = 1;
} = 0x20000020;
```

```
typedef struct{
    char candidato[MAXLENGTHSTRING];
    char giudice[MAXLENGTHSTRING];
    char categoria ;
    char nomeFile[MAXLENGTHSTRING];
    char fase;
    int voto;
} Tupla;

static Tupla tabella[NUMROWTABLE];
static int isSetupTabella = 0;
```

```
void setupTabella(){

    strcpy(tabella[0].candidato, "Giuseppe");
    strcpy(tabella[0].giudice, "DJFrancesco");
    tabella[0].categoria = '0';
    strcpy(tabella[0].nomeFile, "Giuseppe.txt");
    tabella[0].fase = 'A';
    tabella[0].voto = 20;

    strcpy(tabella[1].candidato, "Antonio");
    strcpy(tabella[1].giudice, "DJFrancesco");
    tabella[1].categoria = 'U';
    strcpy(tabella[1].nomeFile, "Antonio.txt");
    tabella[1].fase = 'B';
    tabella[1].voto = 70;
```

esprimi_voto

```
int * esprimi_voto_1_svc(Voto* votazione, struct svc_req *reqstp){
    static int result;

    if (!isSetupTabella) setupTabella();

    result = -1;
    for (int indexTable = 0; indexTable < NUMROWTABLE; indexTable++){

        if (strcmp(tabella[indexTable].candidato, votazione->nomeCandidato) == 0){

            switch (votazione->tipoOp)
            {
                case 'A':
                {
                    tabella[indexTable].voto++;
                    printf("[Incrementa Voto] : Aggiornato voto candidato %s a votazione %d\n", tabella[indexTable].candidato, tabella[indexTable].voto);
                    result = 0;
                }
                break;

                case 'S':
                {
                    if (tabella[indexTable].voto > 0) {
                        tabella[indexTable].voto--;
                        printf("[Decrementa Voto]: Aggiornato voto candidato %s a votazione %d\n", tabella[indexTable].candidato, tabella[indexTable].voto);
                        result = 0;
                    } else printf("[Decrementa Voto]: impossibile decrementare voto candidato %s essendo pari a 0\n", tabella[indexTable].candidato);
                }
                break;
            }
        }
    }
}
```


fattorex_client.c

```
char *host, choice[DIMINPUTSTRING];
int *result;

CLIENT *gestoreTrasporto; //Gestore di trasporto

Classifica *classificaGiudici; //Struttura di Output
Voto * votazione; //Struttura di Input

if (argc > 3){//verifica parametri invocazione
    printf ("Errore invocazione: %s [serverHost]\n", argv[0]);
    exit (1);
}
host = (argc == 2) ? argv[1] : "localhost"; //se non specificato viene assegnato "localhost"

votazione = malloc(sizeof(Voto));
votazione->nomeCandidato = (char*)malloc(MAXLENGTHSTRING);
printf("Creazione struttura dati Input avvenuta con successo!\n");

gestoreTrasporto = clnt_create (host, VOTAFATTOREX, VOTAFATTOREXVERS, "udp");
if (gestoreTrasporto == NULL){
    clnt_pcreateerror (host);
    exit (1);
}
printf("Creazione del gestore di trasporto avvenuta con successo!\n");
```

```
else if( strcmp(choice,"2")==0 ){
    printf("\nInserisci il nome del candidato: \n");
    scanf("%s", votazione->nomeCandidato);
    while (getchar() != '\n'); //pulisco buffer da eventuale garbage prima di lettura carattere

    printf("\nInserisci tipo di operazione (A (addiziona) oppure (S) sottrai ): \n");
    scanf("%c", &votazione->tipoOp);

    result = esprimi_voto_1(votazione, gestoreTrasporto);

    if(result == NULL){
        clnt_perror(gestoreTrasporto, host);
        exit(1);
    }
    printf("Invocazione esprimi_voto avvenuta con successo!\n");
    if(*result < 0)
        printf("Assegnazione del voto non riuscito: inserire nome candidato valido e operazione [A|S]\n");
    else if(*result == 0)
        printf("Votazione registrata con successo\n");
}
```

Gestione progetto

Coding



git



Team



14 Dicembre 2021



Bernardini Claudio
Corsetti Luca
Giardina Gianluca
Straccali Leonardo

Grazie!