

# Esercitazione 9

Belano Andrea, Ceccolini Gabriele, Loddo Filippo, Merenda Simone.  
Gruppo 12

# Specifiche del Progetto

Lo scopo del progetto è quello d'implementare un sistema di votazione per una competizione tra giudici, ognuno dei quali è a capo di un gruppo di candidati. Il server mantiene per ogni partecipante alcune informazioni, ovvero candidato, giudice, categoria, nome del file che descrive il candidato, fase e un numero che corrisponde al totale di voti ricevuti.

Bisogna implementare, usando RPC di SUN, due procedure invocabili:

- classifica\_voto
- aggiungi\_voto



# Inizializzazione

La struttura dati è inizializzata nel file votafattorex\_svc.c utilizzando una variabile extern per far sì che sia visibile a tutta l'applicazione

```
struct l
{
    char candidato[255];
    char giudice[255];
    char categoria;
    char nomeFile[255];
    char fase;
    int voto;
};

typedef struct l l;

#define N 10000
```

```
int
main (int argc, char **argv)
{
    ...

    for (int i = 0; i < N; i++) {
        strcpy(voti[i].candidato, "L");
        strcpy(voti[i].giudice, "L");
        strcpy(voti[i].nomeFile, "L");
        voti[i].categoria = 'L';
        voti[i].fase = 'L';
        voti[i].voto = -1;
    }

    svc_run ();
    fprintf (stderr, "%s", "svc_run returned");
    exit (1);
    /* NOTREACHED */
}
```

# Aggiungi\_voto

Il metodo aggiungi\_voto deve aggiungere o sottrarre un voto a un candidato in base alla richiesta dell'utente. Al termine aggiorna la struttura dati.

```
int* aggiungi_voto_1_svc(voto* voto, struct svc_req *rp) {
    static int result = 0;
    for (int i = 0; i < N; i++) {
        if (!strcmp(voto->candidato, voti[i].candidato)) {
            if (voto->operazione == 'A') {
                voti[i].voto += 1;
            } else {
                if (voti[i].voto > 0) {
                    voti[i].voto -= 1;
                }
            }
            result = 1;
            break;
        }
    }
    return &result;
}
```

# Classifica\_voto

Il metodo classifica\_voto deve restituire una struttura che contiene i nomi di tutti i giudici ordinati rispettivamente secondo il punteggio dato ai candidati.

```
classifica* classifica_voto_1_svc(void* v ,struct svc_req *rp) {
    static classifica * result;

    if (init) {
        xdr_free((xdrproc_t) xdr_classifica, (char *) result);
    }
    result = (classifica *) malloc(sizeof(classifica));

    for (int i = 0; i < NUMGIUDICI; i++) {
        result->giudice[i] = (char *) malloc(sizeof(char) * 255);
        strcpy(result->giudice[i], "");
        result->punti[i] = 0;
    }
    int last = 0;
    int found = 0;
    for (int i = 0; i < N; i++) {
        if (voti[i].voto != -1) {
            found = 0;
```

```
            for (int k = 0; k < NUMGIUDICI; k++) {
                if (!strcmp(result->giudice[k], voti[i].giudice)) {
                    result->punti[k] += voti[i].voto;
                    found = 1;
                    break;
                }
            }
            if (!found) {
                strcpy(result->giudice[last], voti[i].giudice);
                result->punti[last] += voti[i].voto;
                last++;
            }
        }
    }
    quickSortR(result, 0, NUMGIUDICI - 1);
    init = 1;
    return result;
}
```



# Cliente

Il cliente chiede all'utente il servizio da utilizzare, V per aggiungere un voto e C per la classifica. Riceverà come risultato i dati elaborati per la classifica e un booleano per il metodo aggiungi\_voto, oppure un messaggio di errore

```
printf("Inserisci il servizio (V per aggiungi_voto C per classifica_voto)\n");
while (scanf("%c", &c) != EOF) {
    gets();
    switch (c) {
        case 'V' :
            printf("Inserisci il nome del candidato:\n");
            scanf("%s", nomeCandidato);
            gets();
            printf("Inserisci l'operazione (A per aggiungere S per sottrarre):\n");
            scanf("%c", &operazione);
            gets();
            while (operazione != 'A' && operazione != 'S') {
                printf("Inserire solo A o S\n");
                scanf("%c", &operazione);
                gets();
            }
            strcpy(v->candidato, nomeCandidato);
            v->operazione = operazione;
            res = aggiungi_voto_1(v, c1);
            if (res == NULL) {
                printf("Chiamata fallita\n");
                continue;
            }
            if (!*res) {
                printf("Il candidato non esiste\n");
                continue;
            }
        }
    }
```

```
    }
    printf("Voto inserito con successo\n");
    free(v->candidato);
    free(v);
    break;

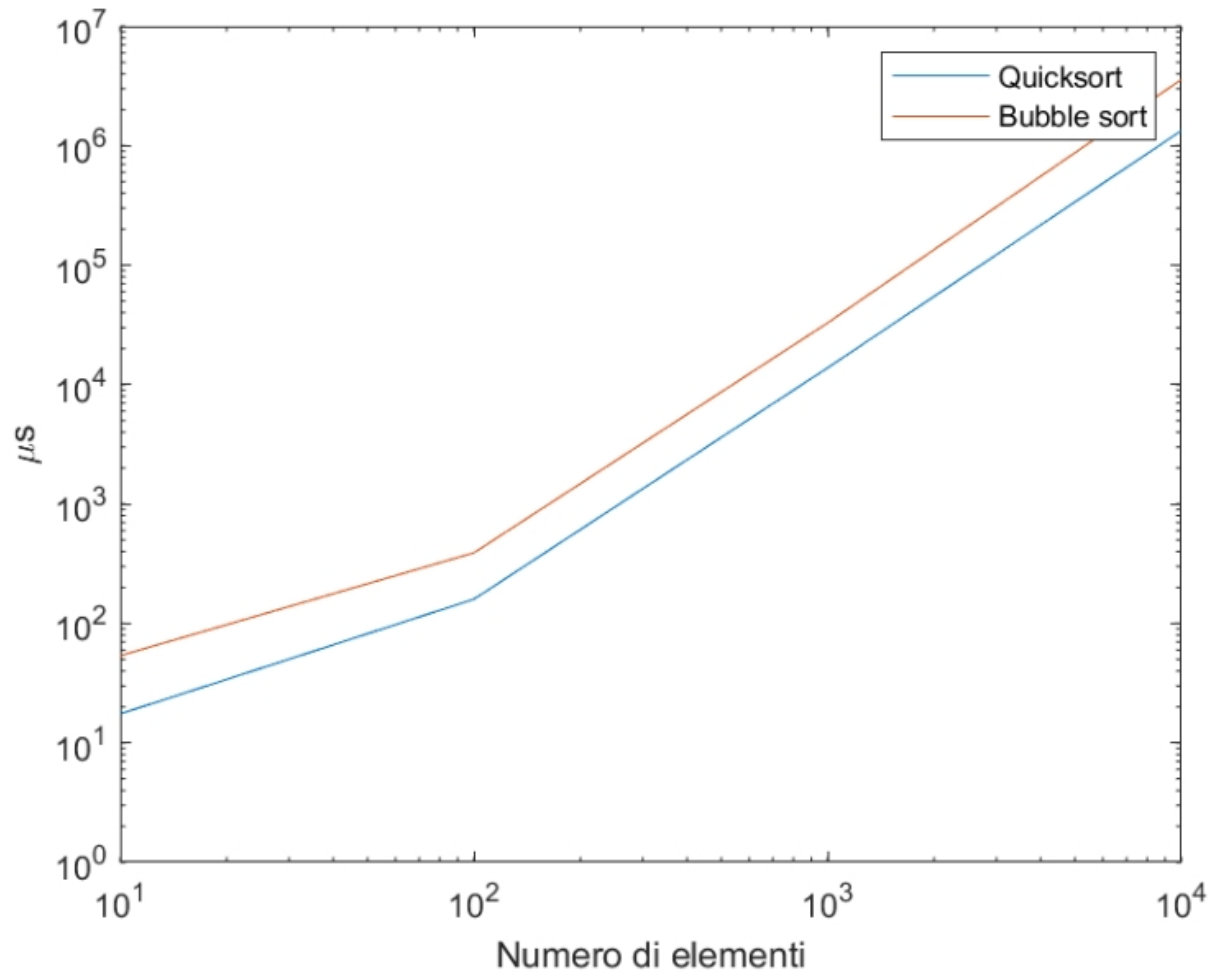
    case 'C' :
        r = classifica_voto_1(0, c1);

        if (r == NULL) {
            printf("Chiamata fallita\n");
            continue;
        }

        printf("La classifica è la seguente:\n");
        for (int i = 0; i < NUMGIUDICI; i++) {
            printf("%s: %d punti\n", r->giudice[i], r->punti[i]);
        }
        break;

    default:
        printf("Inserisci un carattere valido\n");
        break;
}
```

# Test



# Grazie per l'ascolto

Belano Andrea, Ceccolini Gabriele, Loddo Filippo, Merenda Simone.  
Gruppo 12