Soluzione Esercitazione 8

Leonardo Bambini, Patrick Di Fazio, Matteo Longhi,
Giorgio Mastrotucci, Luca Torzi

Requisiti

Si vuole realizzare una applicazione C/S che consente di effettuare operazioni remote su file in un nodo remoto

In particolare le specifiche del server prevedono che:

- si possano contare i caratteri, la parole e le linee di un file di testo
- contare il numero di file presenti nel direttorio indicato dal client, la cui dimensione è maggiore di N specificato.

Il client ha un comportamento da filtro.

Il Client

Il Cilent da filtro in base alla procedura attiva la prima o la seconda procedura in modo ciclico

```
if (procedure == 1)
{
    printf("Inserisci il nome del file");
    scanf("%s", contafile);
    resconta = contafile_1(&contafile, cl);
    if (resconta == NULL)
    {
        clnt_perror(cl, server);
        exit(1);
    }
    /*if (resconta->first == NULL || resconta->first == NULL || resconta->first == NULL)
    {
        clnt_perror(cl, server);
        exit(1);
    }*/
    printf("Inseriti i valori: %d-%d-%d", resconta->first, resconta->second, resconta->third);
}
else if (procedure == 2)
{

    printf("Inserisci il nome del file e dimensione");
    scanf("%s",filename);
    dirscan.filename=filename;
    scanf("%d", &((&(dirscan))->dimfile));
    ris = contadir_1(&dirscan, cl);
    if (ris == NULL)
    {
        clnt_perror(cl, server);
        exit(1);
    }
    printf("risultato:%d\n",*ris);

}

printf("Inserisci procedura");
```

Implementazione

file_scan accetta nome di file e restituisce una struct di tre interi

```
while(read(f,&c,1)>0){
    res.first+=1;
    if(c==' ')res.second+=1;
    if(c=='\n'){
        res.second+=1;
        res.third+=1;

    }
```

```
while((dent=readdir(d))!=NULL){

    //preparazione path
    memset(path,'\0',sizeof(path));
    strcpy(path,dir->filename);
    strcat(path,"/");
    strcat(path,dent->d_name);

    //controllo apertura file e cartelle speciali
    if((f=open(path,O_RDONLY))>=0 && strcmp(dent->d_name,".")!=0 && strcmp(dent->d_name,"..")!=0){
        if(lseek(f,SEEK_END,0)>=dir->dimfile)ris+=1;
        close(f);
    }
}
```

dir_scan accetta nome del direttorio e soglia e in caso di successo resituisce un intero positivo.

XDR: Definizioni (procedure.x)

Per generare il **Client**:
gcc -o procedure procedure_client.c procedure_clnt.c procedure_xdr.c

Per generare il **Server**:
gcc -o procedure_server procedure_proc.c procedure_svc.c procedure_xdr.c

Per generare **header stub e file conversioni**:
rpcgen procedure.x

```
struct Dir_scan { string filename <256>; int dimfile;};
struct Res { int first; int second; int third; };
program FILEPROG {
version FILEVERS {
Res CONTAFILE (string) = 1;
int CONTADIR (Dir_scan) = 2;
} = 1;
} = 0x20000013;
```