

# Proposta soluzione esercitazione 8 RPC

Mattia Innocenti - 825046  
Federico Macchiavelli - 825621  
Andrea Proia - 825784  
Luca Bartolomei - 825005

# Specifiche servizio

L'applicativo ha come obiettivo un'applicazione C/S che consenta di effettuare 2 operazioni remote:

1. un servizio per **contare caratteri, parole e linee** di un file di testo
2. un servizio per **contare numero file** presenti su direttorio remoto scelto la cui dimensione sia maggiore di una **soglia indicata**

Client\_host

Client

file\_scan (argp, clnt)  
o  
dir\_scan (argp, clnt)

scan\_clnt.c

Server\_host

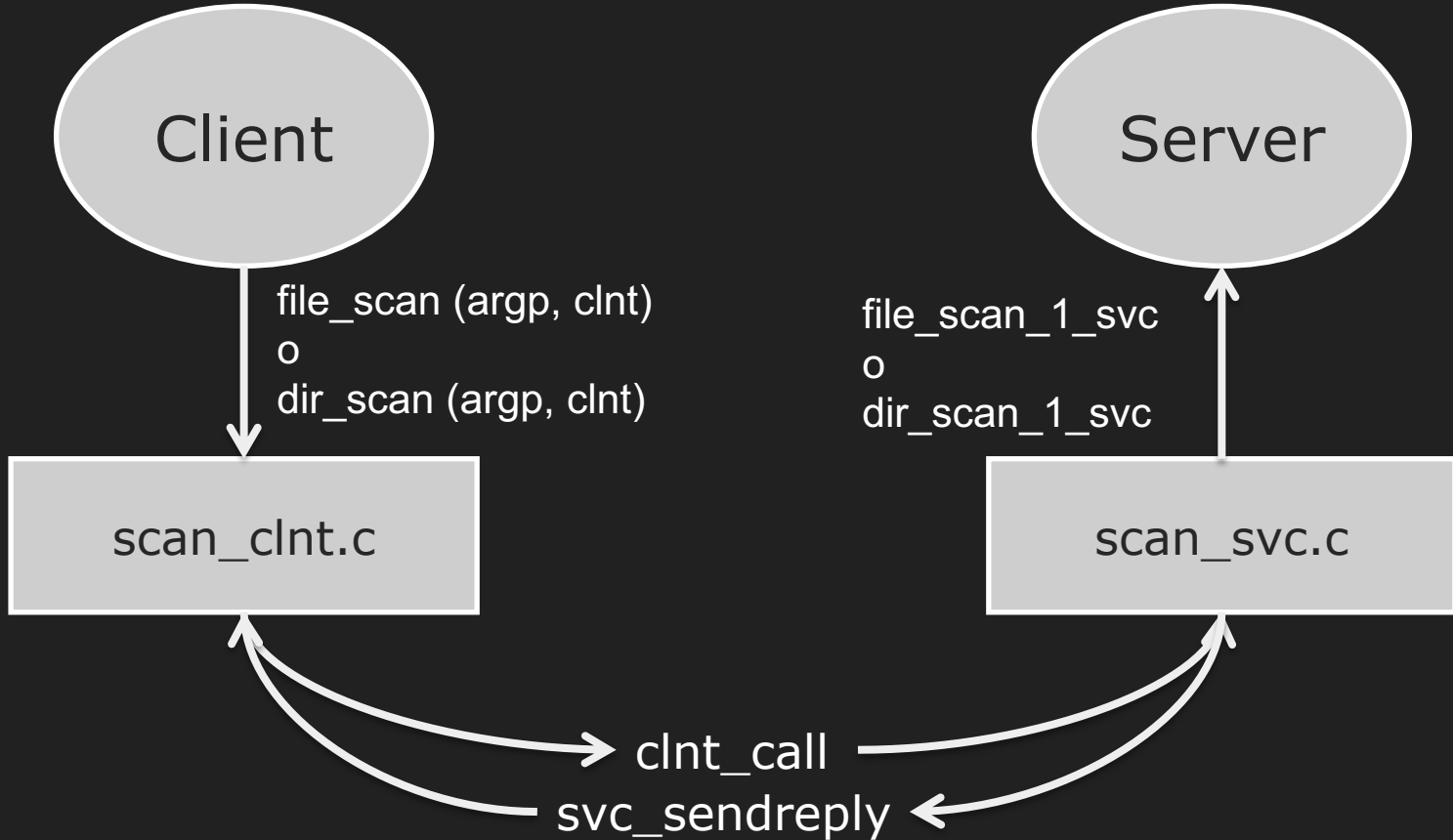
Server

file\_scan\_1\_svc  
o  
dir\_scan\_1\_svc

scan\_svc.c

clnt\_call

svc\_sendreply



# Client - struttura

## Operazioni preliminari:

- Creazione struttura CLIENT e “stringa” server
- Controllo parametri

## Struttura ciclica (filtro):

1. Richiesta tipologia di operazione da effettuare all'utente
2. Richiesto nome file remoto o nome direttorio remoto e soglia
3. Invio richiesta corrispondente all'operazione
4. Controllo esito operazione ( != NULL) e solo a questo punto dereferenzio e stampo a video

# Server - architettura

Per l'architettura del Server ci limitiamo a definire le implementazioni delle procedure.

Nel file XDR vengono definite le procedure remote insieme alle strutture dati necessarie.

Comando RPCGEN genera automaticamente lo stub e le funzioni di conversione XDR.

Al termine dell'esecuzione di una procedura remota il Server effettua la deallocazione dei parametri in ingresso forniti dal Client (allocazione eseguita automaticamente dal supporto)

# Server - struttura FILE\_SCAN()

- Creazione statica struttura **OutputFileScan** e sua inizializzazione
- Apertura file e controllo errori
- Scan dell'intero file aggiornando ad ogni carattere letto le variabili presenti nella struttura **OutputFileScan**
- Chiusura file e restituzione del risultato passato per riferimento

# Server - struttura DIR\_SCAN()

- Creazione statica variabile risultato
- Apertura directory desiderata e controllo errori
- Lettura ciclica degli elementi directory e creazione path in caso l'elemento sia "regular file"
- Calcolo dimensione del file tramite **stat** e aumento variabile risultato in caso superi la soglia desiderata
- Chiusura directory e restituzione risultato passato per riferimento

# Conclusioni

Tramite RPCGEN vengono processati i costrutti specificati sul file xdr generando automaticamente gli stub di Client e Server, file header e file xdr.

Il Server registra i suoi servizi tramite **portmapper** e si pone in attesa infinita (svc\_run)

Per ogni chiamata ricevuta dallo stub del Client viene avviata la procedura di Dispatching mediante la quale viene scelta e gestita la procedura richiesta, trattando correttamente i parametri di input ed output.