

## 2° Esercitazione (da svolgere):

### Socket multicast e Socket Java senza connessione

#### Server

Si vuole realizzare un **DiscoveryServer**, cioè un server che venga attivato da *riga di comando* con la seguente invocazione:

```
DiscoveryServer file1 MCAddr1 ... fileN MCAddrN
```

e che svolge il ruolo di permettere la interazione tra **Clients** e **MulticastServer**

Per ognuno dei file passati come argomenti il DiscoveryServer realizza e attiva un **MulticastServer** simile a quello visto nella seconda parte dell'esercitazione svolta, con l'unica differenza che questa volta ciascun MulticastServer è realizzato come processo lanciato dal DiscoveryServer.

Inoltre, ogni MulticastServer viene attivato con un particolare *indirizzo IP* di classe D (passato anch'esso da riga di comando) e su *una porta scelta casualmente* al momento della creazione della socket (costruttore vuoto) *leggendo il file* corrispondente.

Ad esempio, il primo MulticastServer leggerà da *file1* e sarà avviato sull'indirizzo multicast *MCAddr1*.

Il **DiscoveryServer**, dopo aver attivato tutti i processi MulticastServer, si pone in ascolto, su una socket datagram, delle richieste da parte dei client. Le richieste sono messaggi contenenti un *nome di file*. Se il nome di file è uno di quelli passati ad un MulticastServer attivo, il DiscoveryServer restituisce al client un messaggio contenente *l'indirizzo multicast* e la *porta* del MulticastServer per il file richiesto, altrimenti il DiscoveryServer restituisce un messaggio di *file non disponibile*.

#### Client

Il client intende leggere le linee fornite da un MulticastServer. Esso viene attivato specificando da riga di comando il *nome del file* di cui leggere le stringhe e l'*indirizzo IP* e la *porta* dell'intermediario **DiscoveryServer**:

```
LineClient ipDiscoveryServer portDiscoveryServer file
```

Ogni client invia al DiscoveryServer una richiesta con il *nome del file* che vuole ricevere. Alla risposta di insuccesso, se il file non è fra quelli noti al DiscoveryServer, il client termina; altrimenti utilizza *l'indirizzo multicast* e la *porta* ottenuti per aggiungersi al gruppo multicast, quindi riceve alcune righe del file a cui è interessato. Alla terminazione delle ricezioni il client termina.

## Proposta di estensione

### Attivazione

Si estenda il programma sviluppato in modo che gestisca dinamicamente la registrazione dei MulticastServer e che consenta ai Client di interrogare il DiscoveryServer per ottenere l'elenco dei file distribuiti. Si assuma che DiscoveryServer e MulticastServer siano tutti in esecuzione sulla stessa macchina.

**DiscoveryServer:** viene esteso in modo da abilitare la registrazione dinamica dei MulticastServer. Il DiscoveryServer mantiene una tabella con le corrispondenze fra il *nome del file* richiesto e l'*endpoint del MulticastServer (IP multicast, porta)*.

L'operazione di **registrazione** verifica l'unicità del nome di file e dell'endpoint usato dal server e inserisce la entry. L'operazione di verifica controlla che non esista un altro server registrato per lo stesso file, e che non esista un altro server collegato sullo stesso indirizzo IP multicast e sulla stessa porta. L'operazione di **visualizzazione** restituisce i file registrati in tabella (file ed endpoint).

**MulticastServer:** viene realizzato come processo separato e viene esteso in modo da consentire la registrazione dinamica presso il DiscoveryServer. Si propone la seguente interfaccia di attivazione da riga di comando:

```
MulticastServer portDiscoveryServer  
                    ipMCast portMCast file
```

Una volta attivato, il MulticastServer si registra presso il discovery server e, se l'esito della registrazione è positivo, inizia il ciclo di invio del file.

**Client:** il Client deve consentire di interrogare il DiscoveryServer e di scegliere il file attraverso l'interazione con l'utente. Prima di attivare qualunque ricezione dal MulticastServer, il client richiede al DiscoveryServer la visualizzazione della *lista dei file correntemente attivi*, e permette all'utente di scegliere il file a cui è interessato, abilitandone la ricezione e visualizzazione.

### Terminazione

Si estenda ulteriormente la soluzione proposta in modo da consentire anche la terminazione e la de-registrazione dinamica dei MulticastServer presso il DiscoveryServer.

### Suggerimento

Estendere MulticastServer in modo da consentire l'interazione col programma da riga di comando e l'invio di un comando di terminazione da console.

Sono possibili altre alternative? Quali?

### Consegna

*Chi vuole può inviare via email lo svolgimento dell'estensione ai docenti, in modo da discutere la soluzione ed eventualmente pubblicarla sul sito del corso*