



Università degli Studi di Bologna
Facoltà di Ingegneria

Corso di Reti di Calcolatori T

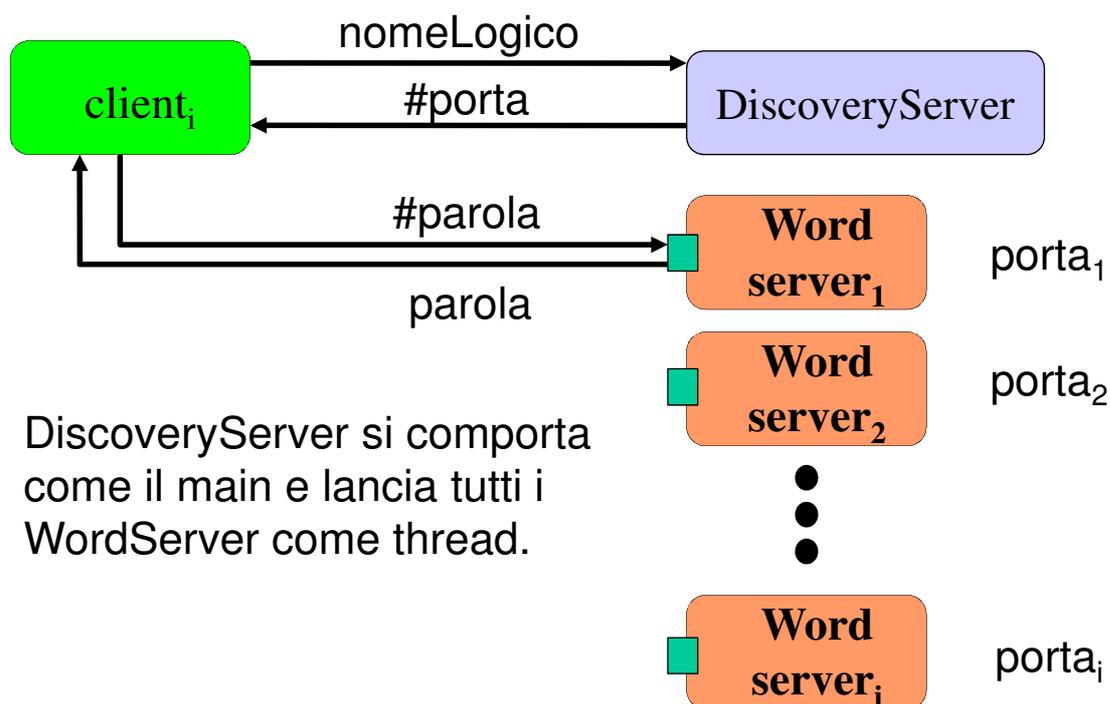
Esercitazione 1 (proposta) Socket Java senza connessione

Luca Foschini

Anno accademico 2013/2014

Esercitazione 1 1

Architettura distribuita



Esercitazione 1 2

Specifica leggiParola()

Il **WordServer** deve mettere a disposizione il metodo statico

```
static String leggiParola(int parolaNumero,  
String nomeFile, int lunghezzaMaxParola)
```

che dati in input il numero della parola all'interno del file (*parolaNumero*), il nome del file (*nomeFile*) e la lunghezza massima della parola che si potrà individuare (*lunghezzaMaxParola*, cioè lunghezza massima del buffer di appoggio):

- Apre il file;
- Legge il file a carattere individuando le parole e contando il numero della parola all'interno del file;
- Restituisce la parola trovata.

Come separatori ammessi si considerino i caratteri ' ' e '\n'.

Esercitazione 1 3

Specifica leggiParola()

Ad esempio per il *file.txt* con contenuto:

*Questo è
il contenuto file.
Questa la fine.*

`leggiParola(2, "file.txt", 256)` → restituisce la stringa "è"

`leggiParola(5, "file.txt", 256)` → restituisce la stringa "file."

`leggiParola(100, "file.txt", 256)` → restituisce una stringa di errore, ad esempio "Raggiunta la fine del file."

`leggiParola(1, "file.txt", 2)` → restituisce una stringa di errore, ad esempio "Buffer troppo corto per la parola cercata."

Esercitazione 1 4

Specifica Server

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

*DiscoveryServer portaDiscoveryServer nomeLogico1 file1 port1...
nomeLogicoN fileN portN*

e che svolge il ruolo da abilitatore della interazione tra **Clients** e **WordServer**, ossia permetta di fornire ai clienti le informazioni di nome per ottenere i dati di interesse.

Per ognuno dei **file passati come argomenti** il DiscoveryServer realizza e attiva un **WordServer**; *ciascun WordServer è realizzato come thread (coresidente e) lanciato dal DiscoveryServer e in ascolto su una porta UDP diversa.*

Esercitazione 1 5

Dettagli Server

Ogni **WordServer**, al momento della creazione della socket, viene attivato sulla **porta IP indicata dall'utente** al lancio del DiscoveryServer (costruttore *DatagramSocket (port)*, con porta) e **legge il file** indicato da riga di comando facendolo corrispondere al **nome logico** indicato.

Ad es., il primo WordServer, legato a **nomeLogico1**, leggerà da **file1** e sarà avviato sulla porta UDP **port1**.

Il **DiscoveryServer**, dopo l'attivazione dei processi WordServer, si pone in ascolto, su una socket datagram, delle richieste da parte dei client. Le richieste sono messaggi contenenti un **nome logico**. Se il nome logico è legato a un WordServer attivo, il DiscoveryServer restituisce al client un messaggio contenente **la porta** del WordServer per il nome logico richiesto, altrimenti il DiscoveryServer restituisce un messaggio di **file non disponibile**.

Si presti particolare attenzione alle funzioni di controllo sulle porte utilizzate, **evitando possibili sovrapposizioni**.

Esercitazione 1 6

Specifica Client

Il **client** intende leggere le parole fornite da un **WordServer**. Esso viene attivato specificando da riga di comando il **nome logico** legato al WordServer da cui leggere le parole e l'**indirizzo IP** e la **porta** dell'intermediario **DiscoveryServer**:

WordClient IPDiscoveryServer portDiscoveryServer nomeLogico

Ogni client invia al DiscoveryServer una richiesta con il **nome logico** del server a cui si vuole collegare. Alla risposta di insuccesso, se il nome logico non è fra quelli noti al DiscoveryServer, il client termina; altrimenti utilizza **la porta** ottenuta per contattare la propria richiesta al server e richiede alcune parole del file a cui è interessato.

Il client termina dopo avere completato le ricezioni di interesse.

Esercitazione 1 7



Proposta di estensione

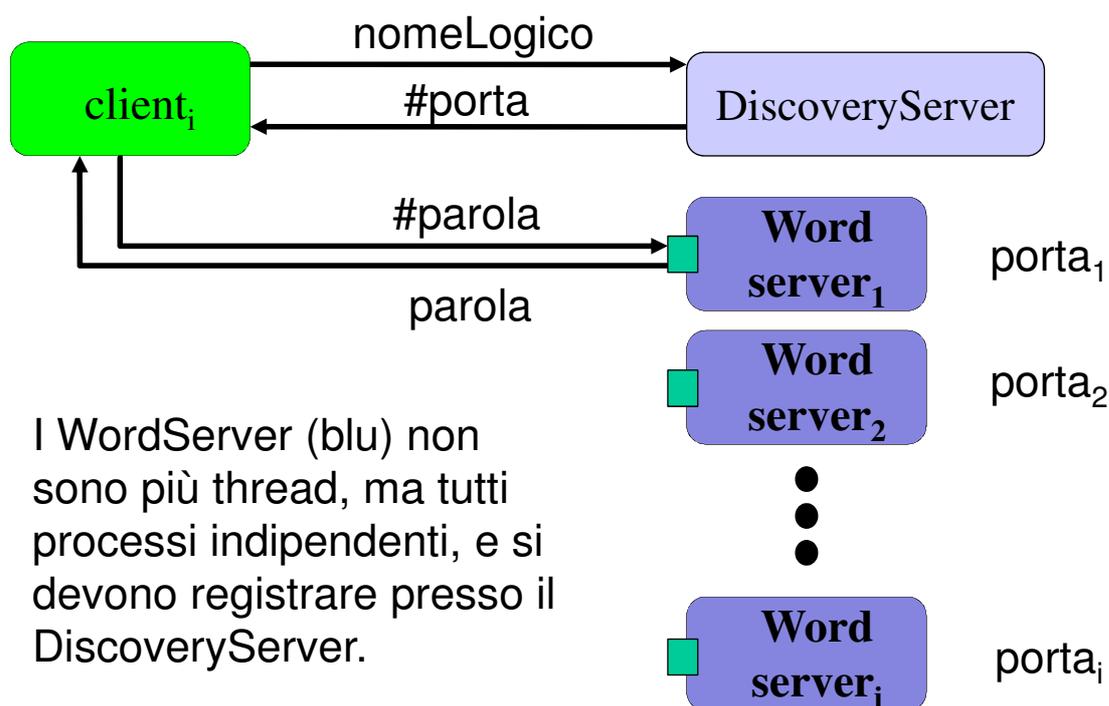


Si estenda il programma sviluppato in modo che gestisca dinamicamente la registrazione dei WordServer e che consenta ai Client di interrogare il DiscoveryServer per ottenere l'elenco dei file distribuiti. Si assuma che DiscoveryServer e WordServer siano tutti in esecuzione sulla stessa macchina.

Quello che vogliamo è che il **DiscoveryServer** possa prevedere **sia di fornire una lista dei suoi server correntemente attivi** sia di **consentire attivazioni e disattivazioni** durante l'esecuzione dei servizi.

Esercitazione 1 8

Architettura distribuita



Esercitazione 1 9



Proposta di estensione: dettagli



DiscoveryServer: viene esteso in modo da abilitare la registrazione dinamica dei WordServer e invocato con:

DiscoveryServer porta

Il DiscoveryServer mantiene una **tabella con le corrispondenze fra il nome logico richiesto e la porta su cui il WordServer è in ascolto.**

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

Esercitazione 1 10



Dettagli attivazione: WordServer



Il **WordServer** viene realizzato come processo separato (lanciato in una Java Virtual Machine – JVM – diversa) e viene esteso in modo da consentire la registrazione dinamica presso il DiscoveryServer.

Si propone la seguente interfaccia di attivazione da riga di comando:

```
WordServer portDiscoveryServer portWordServer  
nomeLogico nomeFile
```

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

Esercitazione 1 11



Dettagli attivazione: 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 WordServer, il client richiede al DiscoveryServer la visualizzazione della **lista dei nomi logici correntemente attivi**, e permette all'utente di scegliere il nome logico a cui è interessato, abilitando la comunicazione col WordServer.



Gestione terminazione



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

Esercitazione 1 12



Suggerimenti



Estendere WordServer 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