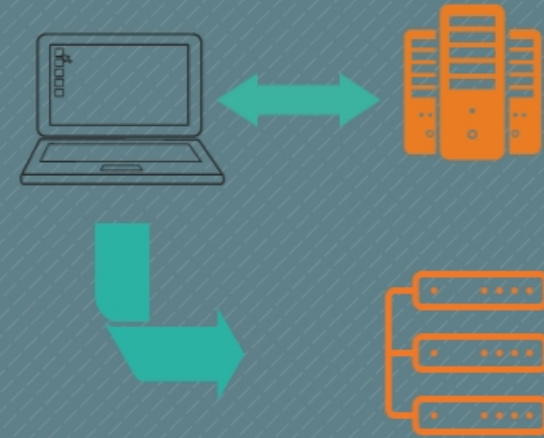


Esercitazione 1

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

Specifiche del Progetto

Il progetto prevede un cliente e due server.
Il cliente manda una richiesta al Discovery Server, specificando il file a cui vuole scambiare le righe; il Discovery Server risponde con la porta corrispondente al server che gestisce il file richiesto, se presente. Il cliente poi invia al RowSwap Server i numeri delle righe che vuole scambiare.



Discovery Server

Il Discovery Server ha il compito di reindirizzare il Cliente sulla porta corrispondente al file richiesto e d'inizializzare i RowSwap Server come thread. In seguito al controllo degli argomenti e all'inizializzazione dei RowSwap Server, apre una socket sulla porta specificata e si mette in ascolto. Alla ricezione del pacchetto dal Cliente, lo legge e controlla se il file richiesto è presente tra i RowSwap Server. In caso di esito positivo, risponde con la porta del RowSwap Server corrispondente, altrimenti con -1 (errore).

```
while (true) {
    socket.receive(pacchetto);
    System.out.println("Pacchetto ricevuto");
    dati = pacchetto.getData();
    byteReader = new ByteArrayInputStream(dati);
    reader = new DataInputStream(byteReader);
    clientAddress = pacchetto.getAddress();
    clientPort = pacchetto.getPort();
    fileName = reader.readUTF();

    found = false;

    for (i = 1; i < files.length && !found; i++) {
        if (fileName.compareTo(files[i]) == 0) {
            found = true;
        }
    }

    pacchetto.setAddress(clientAddress);
    pacchetto.setPort(clientPort);

    if (found) {
        writer.writeInt(porte[i - 1]);
    } else {
        writer.writeInt(-1);
    }

    dati = byteWriter.toByteArray();
    pacchetto.setData(dati);
    socket.send(pacchetto);

    reader.close();
}
```

RowSwap Server

Il RowSwap Server è un Thread del DiscoveryServer. Esso ha il compito di ricevere dal Cliente due numeri e di scambiare le righe corrispondenti nel file da esso posseduto. Per fare ciò si basa su 3 file ausiliari per ottimizzare il lavoro. Al termine dell'operazione ne stampa a video l'esito.

```
while (true) {
    socket.receive(packet);
    dati = packet.getData();
    byteReader = new ByteArrayInputStream(dati);
    reader = new DataInputStream(byteReader);
    linea1 = reader.readInt();
    linea2 = reader.readInt();
    if (linea1 != linea2) {
        success = false;
        if (linea1 > linea2) {
            tmp = linea2;
            linea2 = linea1;
            linea1 = tmp;
        }

        ● ● ●

        for (i = 0; i < linea1; i++) {
            tmpWriter.write(fileReader.readLine() + "\n");
        }
        riga1 = fileReader.readLine();
        for (; i < linea2; i++) {
            ausWriter1.write(fileReader.readLine() + "\n");
        }
        riga2 = fileReader.readLine();
        if (riga2 != null) {
            while ((riga = fileReader.readLine()) != null) {
                ausWriter2.write(riga + "\n");
                i++;
            }
            ausWriter1.close();
            ausWriter2.close();
            fileReader.close();
            tmpWriter.write(riga2 + "\n");
            for (k = 0; k < linea2 - linea1; k++) {
                tmpWriter.write(ausReader1.readLine() + "\n");
            }
            tmpWriter.write(riga1 + "\n");
            for (k = 0; k < i - linea2; k++) {
                tmpWriter.write(ausReader2.readLine() + "\n");
            }
            tmpWriter.close();
            ausWriter1.close();
            ausWriter2.close();
            srcFile.delete();
            success = tmpFile.renameTo(srcFile);
        }
    } else {
        success = true;
    }
}
```


Cliente

Il Cliente deve inizialmente inviare una richiesta al DiscoveryServer contenente il nome del file a cui vuole scambiare le righe. Il DiscoveryServer gli risponde, se il file esiste, con la porta del RowSwap Server corrispondente. Successivamente manda al RowSwap Server due interi che rappresentano le righe del file a cui bisogna scambiare le righe.

```
int porta = Integer.parseInt(args[1]);
InetAddress address = InetAddress.getByName(args[0]);
DatagramSocket socket = new DatagramSocket();
socket.setSoTimeout(30000);
DatagramPacket pacchetto = new DatagramPacket(new byte[255], 255, address, porta);
ByteArrayOutputStream byteWriter = new ByteArrayOutputStream();
DataOutputStream writer = new DataOutputStream(byteWriter);
writer.writeUTF(args[2]);
byte dati[] = new byte[255];
dati = byteWriter.toByteArray();
pacchetto.setData(dati);
socket.send(pacchetto);

socket.receive(pacchetto);
dati = pacchetto.getData();
ByteArrayInputStream byteReader = new ByteArrayInputStream(dati);
DataInputStream reader = new DataInputStream(byteReader);
int res = reader.readInt();
if (res < 0) {
    System.out.println("Errore, nessun TextSwapServer è disponibile per il file indicato");
    System.exit(1);
} else {
    int riga1 = (int) (Math.random() * 100);
    int riga2 = (int) (Math.random() * 100);
    byteWriter.reset();
    writer.flush();
    writer.writeInt(riga1);
    writer.writeInt(riga2);
    dati = byteWriter.toByteArray();
    pacchetto.setPort(res);
    pacchetto.setLength(8);
    pacchetto.setData(dati);
    socket.send(pacchetto);
    System.out.println("Richiesto lo scambio delle righe " + riga1 + " e " + riga2);
    socket.close();
}
```

Grazie per l'ascolto

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