Assignment #3: P+2 server part

This is the first of two assignments dealing with a P2P application. Protocol P+2 specification has been provided. This assignment will deal with creating the server components of a P+2 peer software. There are three basic server components on a P+2 peer:

1. A peer may get a query (to port P) requesting the contents of the list of known peers. A single thread will be created for this task.
2. A peer may get a query (to port P+1) asking for files in the shared folder matching a certain string. A single thread will be created for this task.
3. A peer may get a query (to port P+2) for downloading an specific file contained in the shared folder. File transfer may take a while, so it is recommended to create one thread to process each download request. However and to keep the assignment complexity low, a single thread will be created for this task.

Your Code

Your code has to conduct the following tasks:

1. The program needs three command-line parameters:
    - **a port number** (P) this peer will use for the client management service
    - a string specifying the name of the **shared folder** for the peer to use
    - a string in the format IPaddress:port specifying the **initial peer** address. This third parameter will be used to start populating your list of known peers.
2. Each one of the server threads on this assignment will accept connections (to ports P, P+1 and P+2) and after a connection is accepted a text line will be read from it. Depending on the port, the string read will have a different meaning.
    a) For port P, the string identifies the connecting peer address in the format IPaddress:port. This peer address has to be added to the list of known peers.
    b) For port P+1 the string received contains the substring we have to look for in the shared folder filenames.
    c) For port P+2 the string represents the filename of the folder the peer wants to download.
3. Each request will receive the proper answer. Server thread will close the connection once the answer is sent.
    a) For port P requests, the answer is a list of lines containing each one peer address in the format IPaddress:port.
    b) For port P+1 requests, the answer is the list of filenames (if any) that matches the substring requested. If that substring was "" then the answer will contain all the filenames in the shared folder (at least if you're using String's class contains() method for the matching code).
    c) For port P+2 the answer is a binary transfer of the file, similar to the one we use for the HTTP server in the lab.
4. In this assignment only half of the peer functionality is built. That means you cannot (yet) use your code for searching on other peers files.

Some tips

- Be sure your Firewall software is not blocking ports you're trying to use.
- You may use telnet windows as a client of your server (for testing purposes) by just typing:

```
telnet localhost port
```

- For each port number (P, P+1, P+2) please remember the line you send using telnet as a client will have a different meaning.

- This program will have three different ServerSockets listening on ports P, P+1 and P+2.
- Main class name MUST be `P2P.`
- Command line will be:

  ```
  java P2P port folder initial_peer
  ```
- A simple way of creating the peer address string (given a Socket s) is:

  ```
  String me = s.getLocalAddress().getHostAddress()+":"+port;
  ```

Due date

Your Java source code has to be submitted by email by **March 13th, 2009**. Email subject line MUST be Assignment#3 to ensure proper processing of your submission.

misan@disca.upv.es

If you have any doubt about the assignment I suggest you to stop by my office during office hours (posted on the web).