Multithreaded Java-SocketServer

Sebastian Schmid S0543196

Übung 4 Verteilte Systeme

Aufgabenstellung

In dieser Aufgabe sollte der SocketServer über Multithreading derart erweitert werden, dass er mehrere Anfragen gleichzeitig bearbeiten kann.

Zusätzlich sollten die ID des Threads derart geloggt werden, dass Anfragen sicher zugeordnet werden können.

Lösung

Neue Klasse Receiver startet multithreaded Serverkomponente

Um den Server sauber multithreaded zu bekommen wurde der Teil, der auf die Anfragen des Clients wartet ausgelagert.

Dieser startet dann die Serverkomponente welche ein Runnable erweitert.

```
1  /**
2   * Erstellt von  sebsch on 28.11.16.
3   */
4  public class Receiver implements ReceiverInterface {
5
6      private ServerSocket welcomeSocket;
7
8      public Receiver() throws IOException {
9          this.welcomeSocket = new ServerSocket(6789);
10     }
11
12     @Override
13     public void rcv() throws IOException {
14
15         System.out.println("[ Master ] Server ["
16                 + this.welcomeSocket.getInetAddress()
17                 + "] started. Awaiting querys on Port "
18                 + this.welcomeSocket.getLocalPort() + ".");
19
20         while (true){
21
22             Socket connectionSocket = welcomeSocket.accept();
23             System.out.println("[ Master ] Receved query from " +
                   connectionSocket.getInetAddress() + ".");
24             Thread srv = new TCPServer(connectionSocket);
```

```
            System.out.println("[ Master ] Made a new thread. ID: " + srv.getId() + ", Name.
                " + srv.getName());
            srv.start();

        }
    }
}
```

```
class TCPServer extends Thread implements ServerInterface{

    (...)

    @Override
    public void run() {

        try {
            System.out.println(this.PID + "Serving thread started.");
            this.handleQuery();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    (...)

}
```

Ermitteln der Prozesskennungen

Auslesen der PID

Um die PID also die Systemweite Prozesskennung zu ermitteln, wurde über getRuntime ein shellBefehl auf der bash ausgeführt.

```
 public String getPID() throws IOException {
        byte[] bo = new byte[100];
        String[] cmd = {"bash", "-c", "echo $PPID"};
        Process p = Runtime.getRuntime().exec(cmd);
        p.getInputStream().read(bo);
        return new String(bo).trim();
    }
```

Ermitteln der Java ID

Um die ID des Threads zu ermitteln wurde folgender Befehl aufgerufen.

```
System.out.println("[ Master ] Made a new thread. ID: " + srv.getId() + ", Name. " +
    srv.getName());
```