

TD 2/TP 1 – Serveur FTP

François LEPAN

31 janvier 2013

1 API socket Java

1.1 Quelle méthode Java permet d’attendre l’arrivée de demandes de connexions sur un port TCP ?

```
mySocket.accept();
```

1.2 Comment lire des données sur une socket en Java ?

```
serveur = new ServerSocket(21);

while( true) {
    new MyThread(
        serveur.accept(); //
    ).start();
}
```

1.3 Comment écrire des données sur une socket en Java ?

Avec un `getInputStream()` sur la socket retourner par `accept` pour lire les données.
Avec un `getOutputStream()` sur la socket retourner par `accept` pour écrire les données

2 Le protocole FTP

2.1 Écrire le pseudo-code d’un serveur FTP traitant les commandes citées ci-dessus

```
USER Jules
153 jules connu
PASS password
154 jules connecté
PORT x,y,z 50001
155 OK
RETR blonde.mp4
// ou STORE pour envoyé au serveur
envoyé
```

2.2 Écrire le code Java du serveur FTP en respectant les noms de classes suivantes

```
public class FTPServer {
    ServerSocket server;
    int port;

    public FTPServer(int port) {
        this.port = port;
    }

    public void processRequest () {
        try {
            new FTPServerThread(this.server.accept()).start();
        } catch (IOException e) {
            System.err.println("FTPServer: Unable to accept the connection "
                +" with the ServerSocket: "+this.server);
            e.printStackTrace();
        }
    }

    public void initSocket() {
        try {
            this.server = new ServerSocket(this.port);
        } catch (IOException ioe) {
            System.err.println("FTPServer: Unable to connect to the port: "
                +this.port+" with ServerSocket");
        }
    }

    public void run() {
        this.initSocket();
        while (true) {
            this.processRequest();
        }
    }

    public static void main(String[] args) {
        new FTPServer(2121).run();
    }
}

public FTPServeurThread extends Thread {

    private Socket socket;
    FtpRequest request;

    public FTPServeurThread(Socket socket) {
```

```

        this.socket = socket;
    }

    @override
    public void run() {
        InputStream is = socket.get();
        this.processRequest(is.readLine());
    }

    public void processRequest(String s) {
        String[] cmd = parseLine(s);
        if (cmd[0] == "USER") {
            processUSER(cmd);
        } else if ()
            .
            .
            .
    }

    public void processUSER (String s) {
    }

    public void processPASS (String s) {
    }

    public void processRETR (String s) {
    }

    public void processSTOR (String s) {
    }

    public void processLIST (String s) {
    }

    public void processQUIT (String s) {
    }
}

```

2.3 Enfin, rajouter les requêtes PWD, CWD, CDUP.