

ネットワーク報告書

学生証番号：144704

氏名：高橋 右

1. 前提事項

12回目のPDFに、当初「pdf,src,bin」を出すように指示されていたが、次に更新されたときには「src,bin」を出すように指示されていた。しかし、不安であったため、144704.PDFも一応出しておく。（図で説明した方が理解してもらいやすいためでもある。）

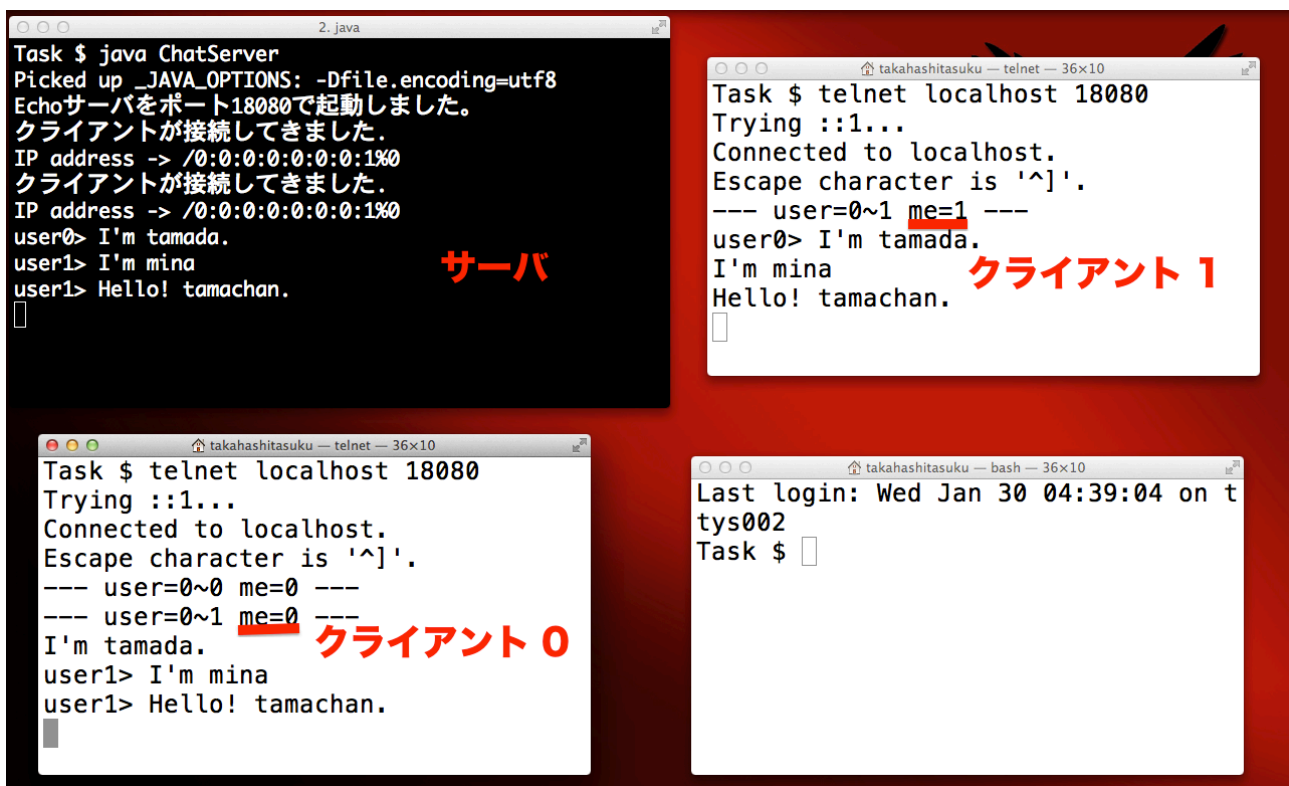
2. プログラムの説明（ごく簡単に）

複数クライアントでチャットを行うチャットクライアントサーバプログラム。

3. このプログラムの特徴

このプログラムはクライアント同士がチャットを行う物だが、エラーが起きないためと、ユーザーインターフェースのため、ほんの少しだけ手を加えた。

ここでは、動作ごとに順を追って説明したいと思う。以下の図1を見てほしい。



```
Task $ java ChatServer
Picked up _JAVA_OPTIONS: -Dfile.encoding=utf8
Echoサーバをポート18080で起動しました。
クライアントが接続してきました。
IP address -> /0:0:0:0:0:0:1%0
クライアントが接続してきました。
IP address -> /0:0:0:0:0:0:1%0
user0> I'm tamada.
user1> I'm mina
user1> Hello! tamachan.

Task $ telnet localhost 18080
Trying ::1...
Connected to localhost.
Escape character is '^]'.
--- user=0~1 me=1 ---
user0> I'm tamada.
I'm mina
Hello! tamachan.

Task $ telnet localhost 18080
Trying ::1...
Connected to localhost.
Escape character is '^]'.
--- user=0~0 me=0 ---
--- user=0~1 me=0 ---
I'm tamada.
user1> I'm mina
user1> Hello! tamachan.

Last login: Wed Jan 30 04:39:04 on tty002
Task $
```

図1： チャットサーバプログラムの動作

図1で、各クライアントに「--- user=0~1 me=0 ---」とプロットした箇所があるが、これは「user=0~1」がユーザの番号で、「me=0」が自分の番号となっている。そのため、左下のターミナルはクライアント0、右上のターミナルはクライアント1としている。また、「user1>」というプロットは、相手から送られて来たメッセージを示しており、ユーザの番号に対応している。このようにしている理由は、ユーザがquitをして減ったとき、ユーザがtelnetをして増えたときの為に、このようにしている。図2を見てほしい。

```

Task $ java ChatServer
Picked up _JAVA_OPTIONS: -Dfile.encoding=utf8
Echoサーバをポート18080で起動しました。
クライアントが接続してきました。
IP address -> /0:0:0:0:0:0:1%0
クライアントが接続してきました。
IP address -> /0:0:0:0:0:0:1%0
user0> I'm tamada.
user1> I'm mina
user1> Hello! tamachan.

```

サーバ

```

Task $ telnet localhost 18080
Trying ::1...
Connected to localhost.
Escape character is '^]'.
--- user=0~1 me=1 ---
user0> I'm tamada.
I'm mina
Hello! tamachan.
--- user=0~0 me=0 ---

```

クライアント 0

```

Connected to localhost.
Escape character is '^]'.
--- user=0~0 me=0 ---
--- user=0~1 me=0 ---
I'm tamada.
user1> I'm mina
user1> Hello! tamachan.
quit
Connection closed by foreign host.
Task $

```

```

Last login: Wed Jan 30 04:39:04 on t
tys002
Task $

```

図2：チャットサーバプログラムの動作

図2の左下のターミナルでquitをしたときに、右上のターミナルのようにuser数が減るように設計した。また、右上のターミナルのクライアントが、me=1からme=0へと変わっているが、これはArrayListを使ってプログラムをしているためであり、quitによって抜けたクライアントの分を詰めたためである。これにより、クライアント1がクライアント0へと変わった。

わかりやすく例えると、「0.ゴリラ、1.ラッパ、2.パリラ」の配列があった場合、「1.ラッパ」を削除するとArrayListでは「0.ゴリラ、1.パリラ」となる。

そのため、1.ラッパだったのが、1.パリラに変更される。このままだと、ゴリラからすれば、ラッパとパリラの区別が出来なくなるので、クライアントが入って来たとき、またquitしたときに、誰がサーバに繋いでいるのかを表示している（これが --- user=0~1 me=0 ---の存在理由である）。

わざわざArrayListを使った理由は、ユーザが増えて来ると、通常に配列を用意している場合では、ユーザが減ったときその配列は使われなくなってしまうし、抜けた配列を詰めるのは困難であるため、ArrayListを導入した。では次に、下の図3を見てほしい。

```
2. java
Echoサーバをポート18080で起動しました。
クライアントが接続してきました。
IP address -> /0:0:0:0:0:0:1%0
クライアントが接続してきました。
IP address -> /0:0:0:0:0:0:1%0
user0> I'm tamada.
user1> I'm mina
user1> Hello! tamachan.
クライアントが接続してきました。
IP address -> /0:0:0:0:0:0:1%0
user0> Hello! Omoto.
user1> Hey! mina.
user1> Palira

```

サーバ

```
takahashitasuku — telnet — 36x10
Trying ::1...
Connected to localhost.
Escape character is '^]'.
--- user=0~1 me=1 ---
user0> I'm tamada.
I'm mina
Hello! tamachan.
--- user=0~0 me=0 ---
--- user=0~1 me=0 ---
Hello! Omoto.

```

クライアント 0

```
takahashitasuku — bash — 36x10
Task $

```

```
takahashitasuku — telnet — 36x10
Task $ telnet localhost 18080
Trying ::1...
Connected to localhost.
Escape character is '^]'.
--- user=0~1 me=1 ---
user0> Hello! Omoto.
Hey! mina.
Palira

```

クライアント 1

図3：チャットサーバプログラムの動作

図3では右下のターミナルを見てわかるように、telnetでクライアントを増やしている。この事によって、ターミナル右上のクライアント0と、ターミナル右下のクライアント1がChatできるようになる。これらのことから、**ユーザがquitをして減ったとき、ユーザがtelnetをして増えたときの対応をすることが出来ていることが確認できる。**