

ネットワークアプリケーション

第5回 Javaによるネットワークプログラミング

石井 健太郎

(423研究室・オフィスアワー火3限)

スケジュール

- 9月27日 第1回「TCP/IPプロトコルスイート」
- 10月4日 第2回「Javaによるウィンドウプログラミング」
- 10月11日 第3回「ネットワークアプリケーションのプログラミングモデル」
- 10月18日 第4回「Javaによるネットワークプログラミング」
- 10月25日 第5回「Javaによるネットワークプログラミング」
- 11月8日 第6回「Javaによるネットワークプログラミング」
- 11月15日 第7回「Javaによるネットワークプログラミング」 **最終課題(1)**
- 11月17日** 第8回「ウェブプログラミングについて」

スケジュール

- 11月22日 第9回「JavaScriptによるクライアントサイドウェブプログラミング」
- 11月29日 第10回「JavaScriptによるクライアントサイドウェブプログラミング」
- 12月6日 第11回「JavaScriptによるクライアントサイドウェブプログラミング」
- 12月13日 第12回「JavaScriptによるクライアントサイドウェブプログラミング」
- 12月20日 第13回「JavaScriptによるクライアントサイドウェブプログラミング」
- 1月10日 第14回「JavaScriptによるクライアントサイド...」 **最終課題(2)**
- 1月19日 第15回「まとめと演習」

- **前回サンプルプログラムの修正点**

- **前回の演習課題ヒント**

- **受信したデータをウィンドウへ反映させるには？**

- コールバックインスタンスを用いる(実演)

- **文字列を2つの数値に分割・変換したいときは？**

- StringTokenizer ・ Integer.parseInt () を用いる

- <https://docs.oracle.com/javase/jp/8/docs/api/java/util/StringTokenizer.html>

- <https://docs.oracle.com/javase/jp/8/docs/api/java/lang/Integer.html>

- **バイナリデータとして2つの数値を送るには？**

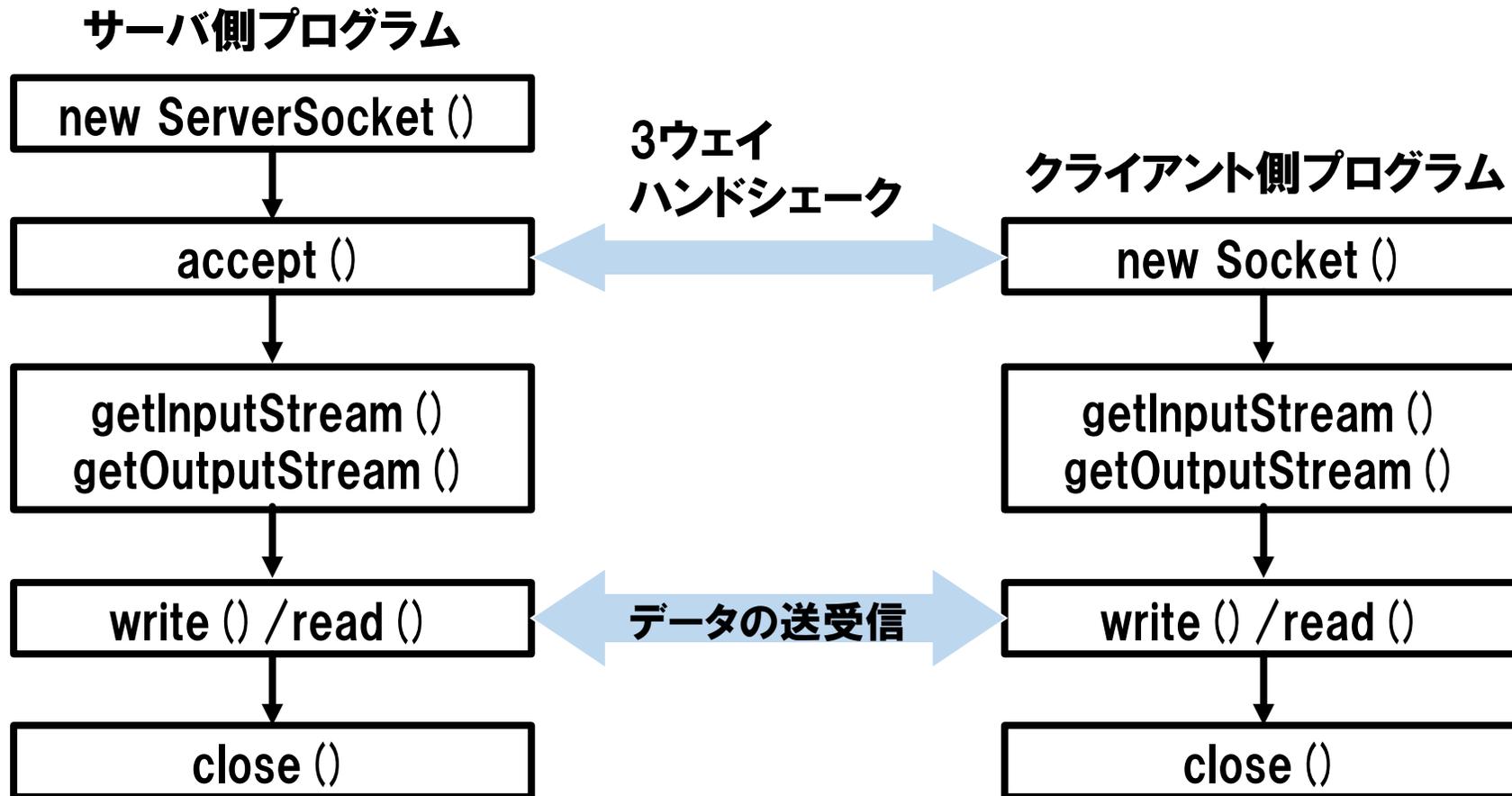
- 1つのパケットに2つの数値を埋め込む
 - 1つの数値を埋め込んだパケットを2つ送る
(UDPなので、どちらか一方が届かないということがありえる)

- **(発展)異なる種類のデータを混在させるには？**

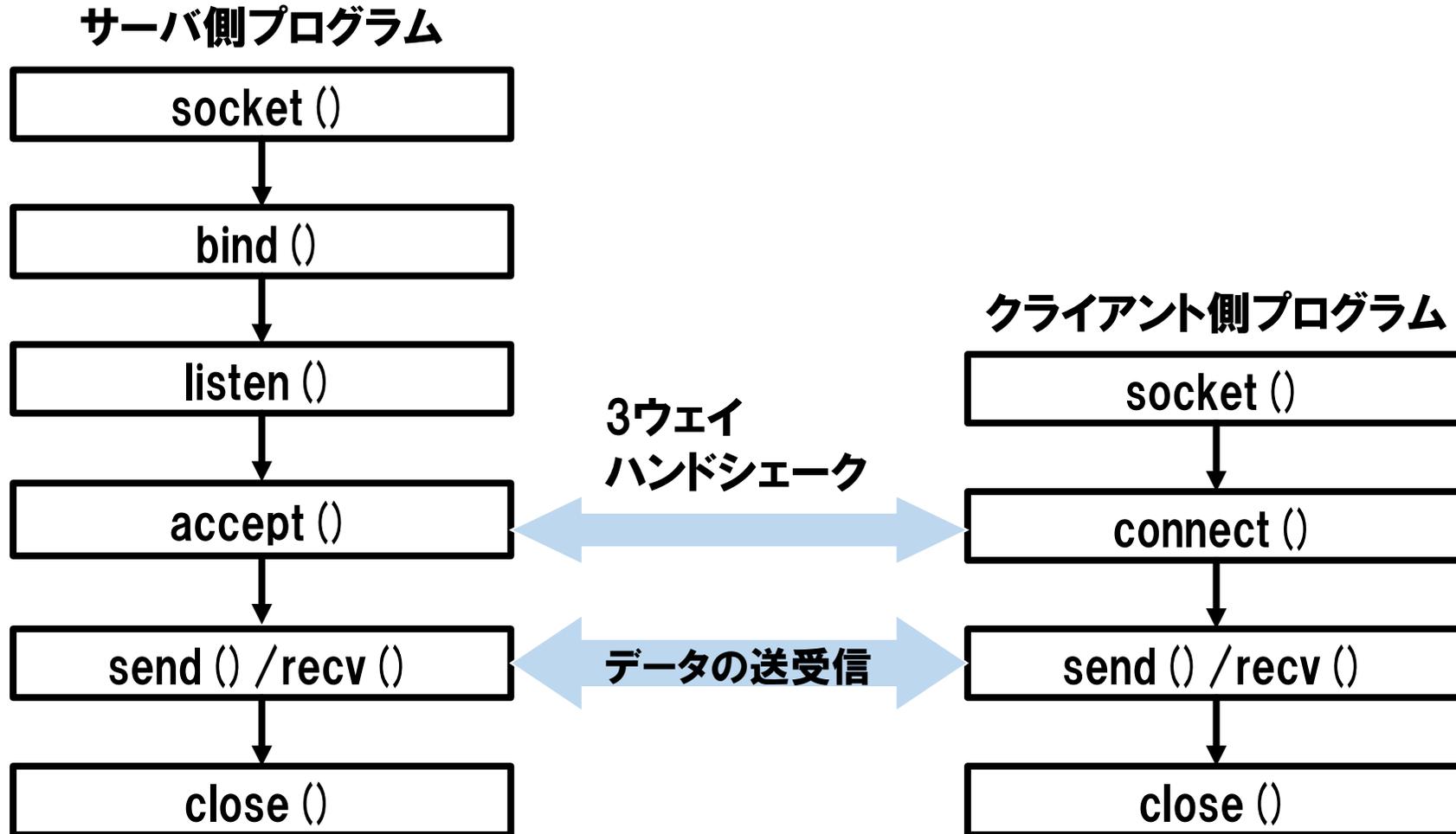
- 例えば、マウスの**クリック**と**ドラッグ**を区別したいときは、**クリック**または**ドラッグ**を意味するヘッダデータを付与する

- **本日は以下のネットワークプログラミングを行う**
 - **TCPによるバイナリデータの送受信プログラム**
 - **TCPによるテキストデータの送受信プログラム**

TCPによるデータの送受信



(参考) TCPによるデータの送受信(C言語編)

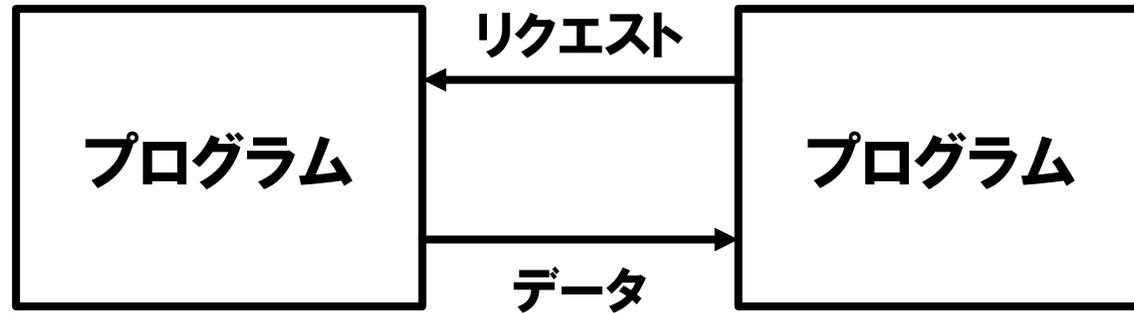


ソケット通信でプログラム同士が連携するには

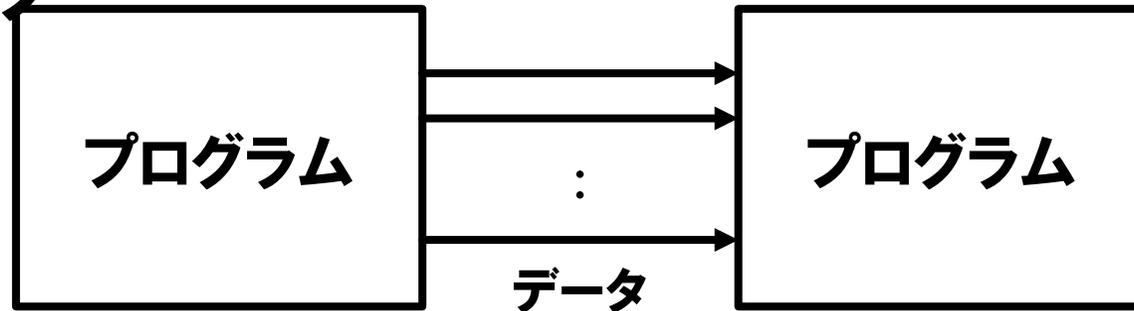
- 実現したいアプリケーションにより, 下記の選択を行う必要がある
 - メッセージパッシングかストリーミングか
 - テキストデータかバイナリデータか
 - TCPかUDPか

メッセージパッシングかストリーミングか

- メッセージパッシング



- ストリーミング



メッセージパッシングかストリーミングか

- **メッセージパッシング**

- 交換されるデータが状況によって変わる場合
- 同期をとって動作する場合

- **ストリーミング**

- 交換されるデータ形式が決まっている場合
- 最新のデータのみを必要とする場合 (cf. UDPの説明)

テキストデータかバイナリデータか

- **テキストデータ (ASCII文字列)**
 - **トラブルが少ない**
 - **バイトオーダーは関係ない**
 - **データがあるのか確認しやすい**
 - **区切り文字(列)を利用してリスト構造を表現しやすい**
- **バイナリデータ**
 - **データのサイズを抑えられる**
 - **バイトオーダーに注意**
 - **ネットワークに流れるデータ(IPヘッダ・TCPヘッダなども含めて)はビッグエンディアンが使用されている**
 - **(参考) htonl (), htons (), ntohl (), ntohs ()**

(参考) テキストデータかバイナリデータか

- Base64エンコーディング
 - 64種類の文字でデータを表現する

- URLエンコーディング
 - 特殊文字やマルチバイト文字を「%(16進数)」で表現する

TCPかUDPか

- TCP

- エンドツーエンドの信頼性の保証・制御行う
 - 到着順序保証・フロー制御・輻輳制御
- コネクション型

- UDP

- エンドツーエンドの制御を行わない
 - (多くの場合)TCPより高速
- コネクションレス型

TCPかUDPか

- TCP

- もれなくデータを転送したい場合
- データの順序が重要である場合
- TLS/SSLにより通信路を暗号化したい場合
- 通信オーバーヘッドが気にならないとき
 - 交換されるデータに対して、伝送路の帯域・端末の処理能力に余裕がある場合

- UDP

- データにもれがあっても問題ない場合
 - 動画・音声のストリーミング
 - 重要性の低いメッセージ
- データの順序が重要でない場合
 - 最新のデータのみを必要とする場合
- あて先があらかじめ決められない場合
 - プロードキャストによる対話相手の発見
- 通信オーバーヘッドが気になるとき
 - 交換されるデータに対して、伝送路の帯域・端末の処理能力がそこそこたいへんな場合

TCPかUDPか

- TCP

- **接続するまでの手順は多い。接続してからのプログラミングは比較的簡単。**
 - `new ServerSocket ()`, `ServerSocket.accept ()`, `new Socket ()`, `Socket.connect ()`,
 - `PrintWriter.println ()`, `BufferedReader.readLine ()`,
`DataOutputStream.writeInt ()`, `DataInputStream.readInt ()`
 - `ServerSocket.close ()`, `Socket.close ()`

- UDP

- **1回の送受信の記述は簡単。送信の都度、あて先を指定する。**
 - `new DatagramSocket ()`, `new DatagramPacket ()`
 - `DatagramSocket.send ()`, `DatagramSocket.receive ()`
 - `DatagramSocket.close ()`

演習

- TCPによるデータの送受信を用いて、ウィンドウアプリケーションを作成する
 - まず、テキストデータにするか、バイナリデータにするか、選択する
 - 基本課題
 - マウスのクリック位置を相手がたプログラムに送信して、そのデータを文字で表示させる
 - さらに、受信したデータに応じた位置(相手のクリック位置)に図形を描画させるなど、見た目におもしろいプログラムに仕上げよう
 - 発展課題
 - クリック位置だけでなく、いろんなデータを送受信して、おもしろいアプリケーションを考えよう
 - 例えば、ドラッグすると相手がたに線が引ける協調お絵かきソフトとするなど
- プログラム(TCP*.java)の内容を、1つのテキストファイルにコピーして、次々回の講義開始時まで、提出フォルダ(Xドライブ)にファイルで提出
 - X:\N科専門¥石井講師¥ネットワークアプリケーション¥第5回
 - ファイル名は「<学生番号>.txt」とする(ハイフンなし) 例: N14999.txt

- **来週は北山祭準備でお休みです**
- **次回は11月8日(火)です**