

## 今回のテーマ

インターネットの（超）基礎  
ネットワークプログラミング

## インターネットの（最低限の）基礎

### 用語から解説

インターネット：

インターネットは、世界中のたくさんのネットが相互接続されて構成されている。インターネット自体は誰かが管理しているものではなく、各ネットは各々管理/運営され、それらが相互に接続していることで機能している。

網の目状につながっている為、常に同じ経路で通信が行なわれるわけではないし、すべてのネットワークが正常に稼働しているとも限らない。

そのため通信におけるデータの信頼性は低くなるが、あるネットにトラブルが発生して通信できなくなっても別の経路で通信を確保できるため、障害に強く柔軟性が高いともいえる。

IPアドレス：

IP (Internet Protocol)とはインターネットを使う上での約束事であり、これに従ってインターネット上にIPネットワークが構成されている。

IPアドレスはネットワークに繋がっているものを識別するための番号になる。ネットワークに繋がっている各マシンやルータなど、インターネットにつながぐために必須である。

IPアドレスには、各組織に割り当てられたユニークな番号（グローバルアドレス）と、自由に使ってよい番号（プライベートアドレス）がある。

グローバルアドレスはインターネット上でユニークなので通信先を一意に決定できるが、プライベートアドレスは誰でも使えるので番号が重なってしまう。そのためローカルなLAN内だけで利用できる。

現在のIP v4では大量のコンピュータや機器のすべてにグローバルアドレスは与えられないので、対外的にはグローバルアドレスを持ち、内部ではプライベートアドレスを使うことが多くなっている。この場合、インターネットに出る際にプライベートアドレスは、グローバルアドレスに変換される。

IP v6ではアドレス空間が広がりすべての機器にグローバルアドレスを

割り当てることも可能になっている。

ちなみに、DNSによってグローバルアドレスをドメイン名が割り当てられている為、アドレスではなくドメイン名でインターネットにアクセスできるようになっている。

例：202.43.240.2（グローバル）、192.168.\*.\* や 61.\*.\*.\* など、8ビット（0-255）までの4つの数値の組み合わせ（計32ビット）で示す。

イーサネット：

ethernet はネットワークの物理的な規格で、コネクタの形状や物理的なデータ通信の規格が決められている。イーサネットによる物理的な規格とTCP/IPによる通信規約によってインターネットができているともいえる。

MACアドレス：

Media Access Controlアドレスはネットワークカードなどネットワーク機器をユニークに識別するための番号である。ネットワークカードなどの製造メーカーの番号と製品一つ一つに違う番号が割り当てられるため、MACアドレスから機器を特定できるようになっている。

例：01-1E-8C-72-\*.\* など計48ビットで表す。上位24ビットが機器メーカー。

TCP/IP

TCP(Transmission Control Protocol)とIPを総称した通信プロトコルで、インターネット上でもっとも一般的に使われているプロトコル。

通信するためには、データのある程度の大きさの packets に分けて送信する。packets には宛先のアドレスやデータ自身が入っている。

インターネット自体ではデータの信頼性を確保できないので、packets を送り出しても正しい順番で到着しないこともあるし、途中でpackets が失われることもある。

そのため、TCP/IPではデータやり取りの手順を決め、データが失われたら再送信の要求をしたり、packets の順番を正しい順番にしたりすることで通信の信頼性を確保している。

このようにお互いにやり取りしながら通信するので、仮想的に通信路を確保した安定した通信となる。

UDP：

User Datagram Protocol はTCPのようにセッションを確保しないで、垂れ流し的にデータを送る通信方法。

そのため、packets が失われたり、順番が違う可能性がある。

ただし、セッションを確保しないのでLAN内の多数の機器に一齐送信するbroadcastができる。

ポート番号：

IPアドレスで通信先の情報はわかるが、どのプログラムがデータを受け取るかを指定するのがポート番号になる。お互いに番号を決めておいて、そのポート番号でデータのやり取りを行なう。ポート番号は0から65535まで指定できるが、若い番号はmailやhttpなど有名なプロトコル用となっているため、1024以降の大きい番号を使うようにする。

プロトコル：

httpやftp,mailなどはアプリケーションプロトコル、一般的にはプロトコルと呼ばれ、アプリケーションやサービスに対応したデータの形式や手順を定めたものである。

たとえば、httpならクライアントであるブラウザからWebサーバに対して、  
GET /

などと送ると、トップページのhtmlが返ってくる。

クライアントとサーバの間のリクエストや応答などを取り決め、その手順を守ることで、ブラウザやhttpサーバが実現されている。

詳しくは、ネットワーク概論でやると思います。

自分で調べるならOSI参照モデルとかを調べると良いかも。

## ネットワークプログラミング

標準ライブラリにあるNetを使ってプログラミングを行う。

Netには、TCP/IP通信を簡単に行なうためのライブラリが用意されている。

本来TCP/IP通信を行なうには、ソケットを作成しポート番号を割り当てたり、

通信をしたり、エラー処理をしたりといった細かい手順が必要だが、

このライブラリでは、サーバ側とクライアント側で各々オブジェクトを作成して、

read/writeするだけで通信ができる。

### 仕組みと手順

サーバ側

- 1 ポート番号を指定してサーバ側ソケットを作成する。  
(この場合はServerオブジェクト)
- 2 通信の要求がくるのを待っている。
- 3 クライアント側から要求が来たらソケットを繋いで通信路を確保する。  
(勝手にやってくれる)
- 4 あとは各クライアントとデータの送受信を行なう。
- 5 必要なくなったら切断する。

クライアント側

- 1 ポート番号とサーバのアドレスを指定してクライアント側ソケットを作成する。  
(Clientオブジェクト)
- 2 サーバと接続する。(勝手にやってくれる)
- 3 繋がったらサーバとデータの送受信を行なう。

### 練習問題 1 :

referenceにあるサーバとクライアントのサンプルを使って改良し、サーバ側でマウスをクリックするごとにクライアント側の背景色がだんだんと変わっていくようにする。

なお、Client側で使っているアドレス 127.0.0.1 はそのマシン自体のローカルなアドレスを示す番号で、同じマシン内で通信する場合には、このアドレスを使うと良い。

referenceにあるサンプル

サーバ側

---

```
import processing.net.*;
```

//ライブラリを使うためには、ライブラリをインポートする必要がある。

```
Server myServer;
```

```
int val = 0;
```

```
void setup(){
```

```
  size(200,200);
```

```
  myServer = new Server(this, 21526); // port番号を指定してサーバを作成
```

```
}
```

```
void draw(){
```

```
  background(val);
```

```
  myServer.write(val); // 繋がっているクライアントにデータを送信
```

```
}
```

---

クライアント側

---

```

import processing.net.*;
Client myClient;
int dataIn;

void setup(){
  size(200,200);
  myClient = new Client(this, "127.0.0.1", 21526);
  // サーバのアドレスとポートを指定してクライアントを作成
}

void draw(){
  if(myClient.available() > 0){ // データが来ていたら
    dataIn = myClient.read(); // データを読み込む
  }
  background(dataIn);
}
----

```

### 練習問題 2 :

サーバ側でマウスをクリックした場所に、クライアント側で四角を描くプログラムを作成する。

x座標とy座標を文字列で送り、クライアント側では文字列から2つの数値に直す。split() または、splitTokens()を使い文字列を分割し、float()などを使って数値に戻すと2つの数値を送ることができる。

### 練習問題 3 :

練習問題 2 では、いくつか問題があった。

マウスクリックを連打してもエラーが出ないように、改行文字を区切り文字として使ってエラーが出ないようにする。

また、clientEvent()を使ってサーバからデータが来たときだけ処理するように変更する。

### 授業内課題 1 :

練習問題 3 を改良して、クライアント側でもマウスをクリックするとサーバ側に四角を描くようにする。つまり、双方向でマウスをクリックすると相手側に描画するようにする。

## サーバクライアント型のプログラム

よくあるサーバクライアント型のシステムでは、サーバは常駐型のプログラムとして

常にクライアントからの要求を待ち続け、要求があると接続しサービスを提供する。  
このようなタイプのプログラムを試してみる。  
サーバでは、クライアントの状況を管理し、複数のクライアントが  
連携して動いているように見せる。

#### **練習問題 4 :**

あたかも2つのウィンドウが繋がっているように、円が跳ね返りながら  
行き来するようなプログラムを作る。  
サーバとクライアントのプログラムで各々ウィンドウを作り、  
2つのウィンドウが座標的に繋がっているかのように見せる。  
serverEvent() を使って、クライアントの数を2つまでに制限する。  
このNetライブラリでは、クライアントの区別ができないため、クライアント側で  
自分の位置を決めるようにする。  
(右側か左側か、ほんのちょっとだけ違う2つのプログラムを作る)  
円の位置は、サーバで管理する。

#### **授業内課題 2 :**

練習問題 4 を拡張して、4つのウィンドウが繋がっているかのように描画を行なう。

#### **課題 :**

前回の課題をネットワーク化して、ネットワーク型のゲームとして実現してみる。  
Areaクラスを使って四角を5x5の25個描画する。  
inside()メソッドを使って、動き回る円がエリア内にあるときに、  
その場所をクリックするとそのエリアが取れる。  
クライアント数は2つに制限し、お互いの取得したエリアの数が表示されるようにする。