

ネットワークプログラミングを利用したアプリケーション開発

日本大学 文理学部情報科学科

5419045 高林 秀

2022 年 2 月 3 日

概要

本稿は、今年度発展プログラミングの課題研究として Processing を用いたネットワークプログラミングを使用したアプリケーション開発を行うものである。本稿前半部では、開発の際に利用した技術やコードに関して説明を行う。本稿後半部では、実際に説明した技術を用いて、Processing 上で実行可能なアプリケーションを作成する。

目次

1	目的	1
2	Processing の概要	2
3	ネットワークプログラミング概要	2
3.1	プロトコル	2
3.2	TCP/IP	3
4	クライアントサイドプログラミング	3
5	サーバーサイドプログラミング	3
6	アプリケーション実装	3
6.1	開発環境	3
6.2	制作内容	4
6.3	挙動説明	4
6.4	工夫点	4
7	巻末資料	4

1 目的

本稿の目的は、今年度発展プログラミングの課題研究として Processing を用いたネットワークプログラミングを使用したアプリケーション開発を行うものである。前半部にて基本的な技術用語の説明を通して、プログラミングの際に必要な知識の復習を行う。後半部では、実際に Processing 上で実行可能なアプリケーション

ン開発を通してネットワークプログラミングを自身のプログラムに実装する。

2 Processing の概要

まず、本稿の開発言語である Processing に関して軽く触れる。

Processing は、オープンソースで開発されているプログラミング言語である。主に電子アートと、グラフィック・ビジュアルデザインに特化した機能を持ち合わせており、他のプログラミング言語よりも視覚的な表現とフィードバックを簡単に得れるという特徴を持つ。

この言語は、米国出身の Casey Reas 氏と Benjamin Fry 氏によって開発・設計された。Reas 氏は、マサチューセッツ工科大学の MIT メディアラボの美学および計算グループの一部として、メディアアートと科学の理学修士号を取得している人物であり、Fry 氏は、データの視覚化に関する専門知識を持つアメリカ人デザイナーで、カーネギーメロン大学でコンピューターサイエンスの副専攻であるコミュニケーションデザインの BFA を取得し、修士号と博士号を取得している人物である。

Processing の構文は、Java に似ており、Java をより単純化した様な構文で構成される。開発の際には、skechbook と呼ばれる統合開発環境 (IDE) を利用して行う。Processing のコードとして記述された各クラスは、最終的に内部で Java にコンパイルされ実行される。その関係上、クラス内の静的変数や静的メソッドは通常禁止で、それらを利用するには開発者が明示的に「純粋 Java モード」を指定する必要がある。



図1 Processing ロゴ

出典：<https://ja.wikipedia.org/wiki/Processing>

3 ネットワークプログラミング概要

この章では、開発に関連する基本的な技術用語の説明を簡単に行う。

3.1 プロトコル

コンピュータ同士がネットワークを介して通信を行うには、データの形式や送受信の手順を定めた「プロトコル」と呼ばれるルールのようなものが必要である。プロトコルにはいくつか種類があり同一のプロトコルに従っているコンピュータ同士のみネットワーク通信を行うことが可能だ。

単純なアプリケーション開発でも、ネットワーク通信を行うには様々な決め事（通信先の機器の指定、データ型の指定等）を定める必要があり、異常な手間がかかる。したがって実際にネットワークプログラミングを行う際には「プロトコルスタック」と呼ばれる、プロトコルを階層化した概念が取り入れられる。この理論的モデルが「OSI 参照モデル」と呼ばれるもので、それを簡略化したモデルが「TCP/IP」と呼ばれるプロトコルスタックである。

階層	OSI参照モデル	TCP/IPの階層	主なプロトコル	接続機器
第7層	アプリケーション層	アプリケーション層	HTTP・POP3・SMTP	ゲートウェイ
第6層	プレゼンテーション層			
第5層	セッション層			
第4層	トランスポート層	トランスポート層	TCP・UDP	ルータ・L3スイッチ
第3層	ネットワーク層	インターネット層	IP・ICMP	
第2層	データリンク層	ネットワーク インタフェース層	Ethernet・PPP	ブリッジ・L2スイッチ
第1層	物理層			リピータ

図2 OSI参照モデルとTCP/IP、その他プロトコルの関係図

出典：<https://ans1-blog.hatenablog.com/entry/2019/05/22/OSI%E5%8F%82%E7%85%A7%E3%83%A2%E3%83%87%E3%83%AB%E3%81%A8TCP%E3%83%8FIP%E3%81%AB%E3%81%A4%E3%81%84%E3%81%A6>

3.2 TCP/IP

TCP/IP は一般的にネットワークプロトコル群の総称を指し、接続先のコンピュータと通信を行うアプリケーションを指定するため、IP アドレスとポート番号を利用する。IP アドレスは、コンピュータの住所のようなもので、個別にコンピュータごとに割り当てられた文字列である。ポート番号は、通信を行うアプリケーションを指定する際に使用する。

前項の図の通り、OSI 参照モデルでは細かく 7 層に通信機能が分類されているが、プログラミングを行う際には細かすぎるため実用的ではなかった。したがって、よりシンプルかつ実用的な階層構造にしたものが TCP/IP である。現代のネットワーク通信ではこの TCP/IP が用いられている。

4 クライアントサイドプログラミング

5 サーバーサイドプログラミング

6 アプリケーション実装

6.1 開発環境

今回の開発は仮想マシン上で行った。下記に当時の環境を示す。

- ホスト OS : Window10 Home 20H2
- 仮想 OS : Ubuntu 20.04.2 LTS
- CPU : Intel(R)Core(TM)i7-9700K @ 3.6GHz
- GPU : Nvidia Geforce RTX2070 OC @ 8GB
- ホスト RAM : 16GB
- 仮想 RAM : 4GB

- Processing version : 3.5.3

6.2 制作内容

6.3 挙動説明

6.4 工夫点

7 巻末資料

本稿で使用した画像、プログラムコード等はすべて以下のリンク先に掲載している。必要に応じてご覧頂きたい。

- GoogleDrive:
- GitHub:<https://github.com/tsyu12345/advancedPrg/tree/master/No15-report3>