

# 情報工学実験C ネットワークプログラミング

氏名:大西 隼也  
学籍番号:09427510

出題日: 2017 年 12 月 12 日  
提出日:2018 年 1 月 30 日  
締切日:2018 年 1 月 30 日

## 1 概要

本実験では、基本的な通信方式である TCP/IP, UDP/IP によるネットワークプログラミングについて学習する。また、分散システムの基本的な形式であるクライアントサーバモデルの仕組みを学習する。最終的に、クライアントサーバモデルに基づくプログラムを作成する。

## 2 クライアント・サーバモデルの通信の仕組みについて

### 2.1 プロセス間通信の場合

ソケットとはファイルの入出力と同様の処理でプロセス間通信を実現するためのもの  
通信手順  
通信相手プロセスとの間にソケットを生成する  
ソケット番号（ファイルディスクリプタの番号）が返る  
ソケットに対して send や recv を実行することでデータの送受信を実現する（ソケット番号で指定）  
通信が終了したらソケットを削除する

### 2.2 インターネットでの通信の仕組み

計算機の識別方法は、インターネット上のすべての計算機には、一意の IP アドレスが割り振られている (IP:Internet Protocol)  
同じ IP アドレスを複数の計算機が持つことはない, (IP アドレスで計算機を特定可能)

### 2.3 DNS (Domain Name Service)

インターネット上のホスト名と IP アドレスを対応させるシステム  
IP アドレスでは覚えにくく直感的にわかりにくい  
ホスト名の例:www.okayama-u.ac.jp  
ホスト名から IP アドレスに（正引き）、IP アドレスからホスト名に（逆引き）変換可能  
DNS サーバに処理を依頼すると、変換結果が得られる（nslookup コマンド）

## 2.4 ポート番号

同時に複数の計算機と通信することが考えられる

通信相手計算機に複数のプログラムが存在する場合どうやって相手を識別するのか？

→補助アドレスとしてポート番号を利用する

ポート番号:0-65535 の間で指定可能

サービス種別を判別するために用いられる

IP アドレスとポート番号で通信相手プログラムを指定

150.46.30.130 と 80 番ポート：岡山大学の web サーバ

well-known ポート

1023 番までのポートは主要なプロトコルで用いられる番号が決まっている

FTP(20,21), SMTP(25), DNS(53), HTTP(80), POP3(110)

## 2.5 クライアントサーバモデル

クライアント:サービスを受けるプロセス

サーバに処理を要求, 受け取った結果を利用して処理を行う

サーバ:サービスを提供するプロセス

クライアントから要求された処理を行い, 結果を返す

## 2.6 メッセージの内容

実際にクライアントとサーバはどのようなメッセージをやりとりしているのか

プロトコル:通信規約

ネットワークを介してコンピュータ同士が通信を行う上で相互に決められた約束事の集合

RFC: インターネットの標準技術を定めた文書にて記されている

web クライアントはサーバに要求メッセージを送信し, 応答メッセージを受信することを繰り返して, 処理を行っている.

## 2.7 クライアントの処理の流れ

1. 通信相手の IP アドレスを取得
2. ソケットの作成
3. 接続の確立
4. 要求メッセージを送信
5. 応答メッセージを受信
6. 応答メッセージを処理
7. ソケットの削除

## 2.8 TCP/IP での送受信関数

主にクライアントで使う物を書く

- gethostbyname: IP アドレスを得る
- socket: ソケットを作成する
- connect: コネクションを確立させる
- send: メッセージを送信する
- recv: メッセージを受信する
- close: ソケットを削除する

主にサーバで利用される物は,

- socket: ソケットを作成する
- bind: ソケットに名前をつける
- listen: 接続要求を待つ
- accept: 接続要求を受け付ける
- send: メッセージを送信する
- recv: メッセージを受信する

## 2.9 サーバプログラムの処理

サーバは要求メッセージの到着を常に待ち, 要求メッセージが到着したら処理を行い, 結果を送信する.

- ソケットの作成
- ソケットに名前をつける
- 接続要求の受付を開始する
- 接続要求を受け付ける
- 要求メッセージを受信
- 要求メッセージを処理
- 応答メッセージを送信する
- 次の接続要求の受付を開始する

### **3 名簿管理プログラムのクライアント・サーバプログラムの作成方針**

#### **3.1 名簿管理プログラムの仕様について**

基本的にはプログラミング演習で作成した名簿管理プログラムの入出力部分を send 関数や recv 関数を用いて書き換えを行い，サーバ，クライアント間で通信を行えるように実装し直す。

### **4 プログラム及び，その説明**

TCP/IP のプロトコルの説明

### **5 プログラムの使用法**

### **6 プログラムの作成過程に関する考察**

#### **6.1 工夫した点**

##### **6.1.1 ソケットの再送待機状態対策**

#### **6.2 作成に苦勞した点**

##### **6.2.1 %P コマンドの実装について**

### **7 得られた結果に関する考察**

### **8 作成したプログラム**