

# 情報科学演習C 課題2レポート

氏名 山久保孝亮  
所属 大阪大学基礎工学部情報科学科ソフトウェア科学コース  
メールアドレス u327468b@ecs.osaka-u.ac.jp  
学籍番号 09B22084  
提出日 2024年5月21日  
担当教員 平井健士, 中島悠太

# 1 課題 2-1

## 1.1 アルゴリズム

今回私が作成した echoclient プログラムは以下のような流れで処理を実行する.

1. 入力形式の確認
2. ソケットの生成
3. ホストが存在するかどうかを確認
4. ソケットアドレス再利用の指定
5. ソケット接続先の情報設定
6. ソケットをサーバーに接続
7. メッセージをサーバーに送信
8. メッセージをサーバーから受信
9. 6 7 を繰り返し

以下でその詳細について述べる.

### 1.1.1 入力形式の確認

今回の echoclient プログラムは第一引数としてホスト名を指定するという使用法を想定している. そのため, それ以外の使用を制限するために main 関数のコマンドライン引数である argc の値を確認することで実現した. argc は指定した引数の個数+1 の値を表すのでこれが 2 出ない場合は正しい書式でないとしてエラーメッセージを表示し, プログラムを終了する.

### 1.1.2 ソケットの生成

ソケットを生成するには, socket システムコールを使用する. socket システムコールの引数は 3 つあり, 1 目がドメインの種類, 2 目がソケットの型, 3 目が使用するプロトコルの種類を指定する. 以下のそれぞれの詳細について述べる. [2][3]

- ドメインの種類は AF\_INET, AF\_INET6, AF\_UNIX, AF\_RAW のいずれかである. これにより使用するドメイン内のアドレスの形式である, アドレスファミリーが決定される. これらのアドレスファミリーは sys/socket.h インクルードファイルによって定義されている.
- ソケットの型には SOCK\_STREAM, SOCK\_DGRAM, SOCK\_RAW のいずれかである. これによりソケットの型が指定される.  
SOCK\_DGRAM は UDP を使用したプロセスの通信を可能にする. データグラムソケットは SOCK\_RAW は, 内部プロトコル (IP など) のインターフェースを提供し, AF\_INET, AF\_INET6 ドメインでサポートされている.  
SOCK\_STREAM は TCP を使用したプロセスの通信を可能にする. ストリームソケットは信頼性の高い, 順序付けされた重複の無い双方向データフローを提供する.

- プロトコルは 0, IPPROTO\_UDP, IPPROTO\_TCP のいずれかである。ドメインの種類とソケットの型によってそれぞれデフォルトのプロトコルがあるので、たいていの場合デフォルトを表す 0 を用いればよい。

また、正常に実行された場合は負でないソケット記述子を返し、以上があれば-1 を返す。今回は UDP を使用するので第一引数には AF\_INET, 第二引数には SOCK\_DGRAM, 第三引数には 0 を指定した。正常に socket システムコールが終了したかどうかを確認するために返り値を格納する int 型変数 sock の値が 0 より小さいかどうかを条件分岐で判定する。正常に実行されていない場合はエラーメッセージを出力しプログラムを終了する。

### 1.1.3 ホストが存在するかどうかを確認

ホストが存在するかどうかを確認するために gethostbyname システムコールを使用した。[4] 引数は一つでホスト名を指す文字列である。正常終了すると hostent 構造体へのポインタを返し、エラーが発生した場合は NULL を返す。エラーの種類は以下のとおりである。

- HOST\_NOT\_FOUND：引数で指定されたホストが見つからなかったとき
- TRY\_AGAIN：ローカルサーバが権限サーバから応答を受信しなかったのもう一度試すよう言われた時
- NO\_RECOVERY：リカバリー不能エラー
- NO\_ADDRESS：要求されたホスト名は有効だが DNS サーバで IP アドレスが見つからなかったとき
- SERVICE\_UNAVAILABLE：指定されたネームサービスが実行されていないか使用可能ではないとき

また, hostent 構造体は以下のようなメンバを持っている。[5]

- h\_name：ホストの名称
- h\_aliases：ホストの代替名のリスト
- h\_addrtype：返されるアドレスのタイプ
- h\_length：アドレスの長さ
- h\_addr\_list：ホストのネットワークアドレスのリスト

したがって, hostent 構造体の構造体変数のポインタとして server を定義し、これに gethostbyname システムコールの返り値を格納した。また、引数にはコマンドライン引数の argv[1] を使用した。その後 gethostbyname システムコールが正常に終了したかどうかを判定するために server に NULL ポインタが返されていないかを条件分岐によって確認する。もし返されていればエラーメッセージを出力しプログラムを終了する。

### 1.1.4 ソケットアドレス再利用の指定

setsockopt 関数を使ってソケット関連のオプションを設定する。引数は 5 つあり、いかにその詳細を記述する。[6][7]

- 第一引数：オプションの適用先であるソケット
- 第二引数：オプションが設定しているレベル。様々なレベルがあるが, SOL\_SOCKET は第一引数で指定したソケットのオプションのレベルが得られる。

- 第三引数:指定するソケットオプションの名前. 様々なオプションがあるが, 今回使用する SO\_REUSEADDR について記述する. これは通常一定時間ほかのソケットがそのポートを使えなくなってしまうことを防ぐことができるようになる.[8]
- 第四引数: オプションデータへのポインタ.SO\_REUSEADDR を有効にするには整数型で 1 の値が渡される.
- 第五引数: オプションデータの長さ.

また, 返り値は正常に実行された場合は 0 を返し, されなかった場合は-1 を返す. したがって, 今回作成したプログラムでは int 型変数 reuse を 1 に設定し, setsockopt 関数の引数に使用して正常に実行されたかを確認するために返り値が 0 未満かどうかで条件分岐をさせた. 正しく実行されていなかった場合はエラーメッセージを出力しプログラムを終了した.

#### 1.1.5 ソケット接続先の情報設定

#### 1.1.6 ソケットをサーバーに接続

#### 1.1.7 メッセージをサーバーに送信

#### 1.1.8 メッセージをサーバーから受信

### 1.2 実行結果

## 2 課題 2-2

### 2.1 アルゴリズム

### 2.2 実行結果

## 3 発展課題

## 4 感想

今回の課題を通して UDP 及び TCP への理解が深まり, またプログラムを使ってどのようにサーバーとクライアントが接続されるのかを理解することができた. 個人的にネットワークには興味があるので, 今後の課題も興味を持ちながら取り組んでいきたいと思う.

## 5 謝辞

今回の課題を通して質問対応, レポート採点等をしてくださった教授, TA の皆様方ありがとうございました. 今後の課題もよろしくお願いいたします.

## 参考文献

- [1] 情報科学演習 C 指導書
- [2] <https://www.ibm.com/docs/ja/zos/2.5.0?topic=functions-socket-create-socket> 5/21 アクセス
- [3] <https://ryuichi1208.hateblo.jp/entry/2020/03/20/144644> 5/21 アクセス
- [4] <https://www.ibm.com/docs/ja/aix/7.3?topic=g-gethostbyname-subroutine> 5/21 アクセス
- [5] [https://www.qnx.com/developers/docs/8.0/com.qnx.doc.neutrino.lib\\_ref/topic/h/hostent.html](https://www.qnx.com/developers/docs/8.0/com.qnx.doc.neutrino.lib_ref/topic/h/hostent.html) 5/21 アクセス
- [6] <https://www.ibm.com/docs/ja/zos/2.4.0?topic=functions-setsockopt-set-options-associated-socket> 5/21 アクセス
- [7] <https://docs.oracle.com/cd/E19455-01/806-2730/sockets-49/index.html> 5/21 アクセス
- [8] <https://qiita.com/bamchoh/items/1dd44ba1fbef43b5284b> 5/21 アクセス
- [9] 5/21 アクセス