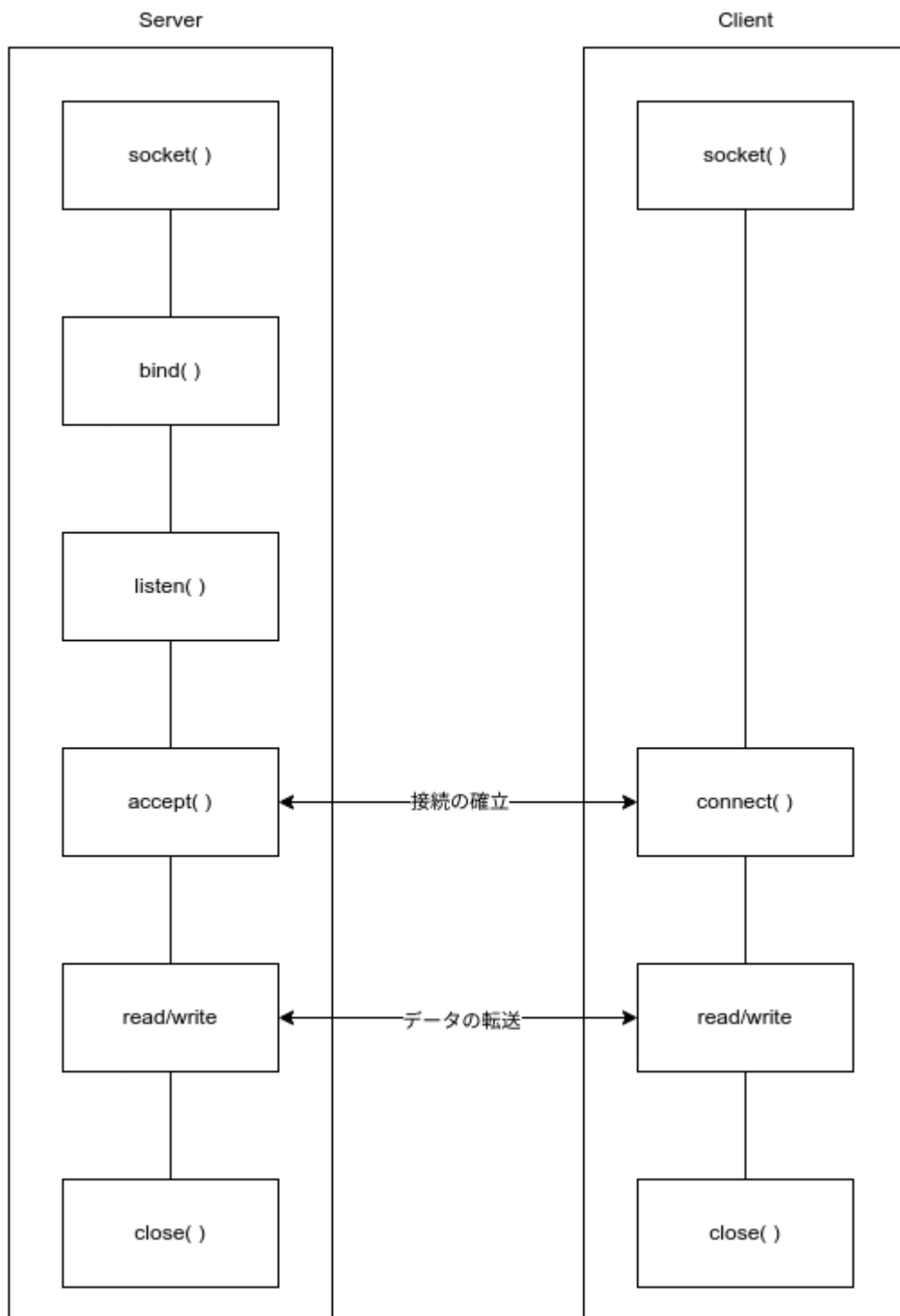


13 ネットワークプログラミング 3

ソケット通信3

- socket通信の流れを確認・理解しておこう



UDPを使用して通信する

- 前回まではTCPを使用した
 - 今回はUDPを使用してみる

UDP受信プログラム

【socket_udp_receive.c】

```
#include <netinet/in.h>
#include <stdio.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <unistd.h>

#define SERVER_PORT 60002
#define BUF_SIZE 4096

int main(int argc, char const *argv[]) {
    // ソケットオブジェクトの作成
    int sockfd = socket(AF_INET, SOCK_DGRAM, 0);
    if (sockfd < 0) {
        perror("socket");
        exit(1);
    }

    // IPアドレスとポート番号を作成したソケットオブジェクトに紐づける
    struct sockaddr_in server_addr;
    server_addr.sin_family = AF_INET;
    server_addr.sin_port = htons(SERVER_PORT);
    server_addr.sin_addr.s_addr = INADDR_ANY;

    if (bind(sockfd, (struct sockaddr *)&server_addr, sizeof(server_addr)) < 0) {
        perror("bind");
        exit(1);
    }

    // データを受信する
    struct sockaddr_in senderinfo;
    socklen_t addrlen;
    char buffer[BUF_SIZE];
    ssize_t received_len;

    addrlen = sizeof(senderinfo);
    received_len = recvfrom(sockfd, buffer, sizeof(buffer) - 1, 0, (struct sockaddr
    *)&senderinfo, &addrlen);
    if (received_len < 0) {
        perror("recv");
    }
}
```

```

        exit(1);
    }
    buffer[received_len] = '\0'; // NuLL文字を追加

    // 標準出力に受信したデータを表示する
    printf("%s\n", buffer);

    // ソケットを閉じる
    close(sockfd);

    return 0;
}

```

- `server_addr.sin_addr.s_addr = INADDR_ANY;` の `INADDR_ANY` とは何ですか？

※注

- 前回の流れで、`server_addr` といった変数名にしているが、何がserverなのか？
 - データを送信する方？
 - 受信する方？
 - 送受信を行う場合のserverとは？

UDP送信プログラム

【socket_udp_send.c】

```

#include <arpa/inet.h>
#include <netinet/in.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <unistd.h>

#define SERVER_PORT 60002

void usage(const char *);

int main(int argc, char const *argv[]) {
    const char *server_ip;

```

```

if (argc != 2) {
    usage(argv[0]);
} else {
    server_ip = argv[1];
}

int sockfd = socket(AF_INET, SOCK_DGRAM, 0);

struct sockaddr_in addr;
addr.sin_family = AF_INET;
addr.sin_port = htons(SERVER_PORT);
inet_pton(AF_INET, server_ip, &addr.sin_addr.s_addr);

char message[] = "Hello, World";
ssize_t send_len;

send_len = sendto(sockfd, message, strlen(message), 0, (struct sockaddr *)&addr,
sizeof(addr));
if (send_len < 1) {
    perror("sendto");
    exit(1);
}

close(sockfd);

return 0;
}

void usage(const char *msg) {
    printf("使用方法：%s IPアドレス\n", msg);
    exit(1);
}

```

- 新しい関数を前回使用した関数と比較して確認してください。

| 関数名 | 引数や戻り値 | 働き・仕様など詳細 |
|--------------|--------|-----------|
| socket () | | |
| recvfrom () | | |
| recv () | | |
| sendto () | | |
| send () | | |

調査と修正

- 以下のUDPを使用した通信の内容と違いについて調べてまとめてください。
 - ブロードキャスト通信とは何か？
 - マルチキャスト通信とは何か？
- 先のプログラム【socket_udp_send.c】をブロードキャスト送信ができるように変更してください。
 - 正しく、ブロードキャストできているかの動作確認を行ってください。

ブロッキング・ノンブロッキング処理

- これまでの例は、すべてブロッキングモードを使用した例
 - 受信関数を呼び出した時に、データが届くまで処理が止まる。
- この方法だと、データが来るか来ないかわからない時、他の処理をすることができない。
 - ノンブロッキングモードの処理方法を学ぼう
 - 具体的にどうすれば良いか？
- ノンブロッキング処理をした受信プログラムを作成してください。
 - ファイル名「`nonblocking_receive.c`」
 - UDPを使用する
 - ポートは、60003を使用する
 - ANYからの接続を許可する
 - 受信がない場合は「`.`」等を表示して、受信中であることが分かるようにする。
 - もしくは `sleep()` を使用して、少しwaitをかけ、動作しているのが分かるようにする（1秒～2秒）
 - 動作しているのを確認したら、コメントアウトしても良い
 - 無限ループで、受信し続ける
 - 送信側は、先に使用した `socket_udp_send.c` を修正して、通信するようにする。
- 必要であれば、以下の関数を使用してもよい。
 - コードの内容をよく読んでから使用すること。

```
/**
 * @brief ソケットをノンブロッキングモードに設定する関数
 *
 * 指定されたソケットディスクリプタに対して、O_NONBLOCKフラグを設定し、
 * ノンブロッキングモードに変更する。
 *
 * @param sockfd ノンブロッキングモードに設定するソケットのファイルディスクリプタ
 * @return int 成功時は0、失敗時は-1を返す。
 */
int set_socket_to_non_blocking_mode(int sockfd) {
    // 現在の設定を取得
    int flags = fcntl(sockfd, F_GETFL, 0);
    if (flags == -1) {
        perror("fcntl");
        return -1;
    }
    // 現在の設定にO_NONBLOCKを追加
    if (fcntl(sockfd, F_SETFL, flags | O_NONBLOCK) == -1) {
        perror("fcntl");
        return -1;
    }
    return 0;
}
```

端末制御ライブラリ

- 画面表示を思い通りにするために、便利なライブラリが存在する
 - ncurses

<https://ja.wikipedia.org/wiki/Ncurses>

- ライブラリを利用できるようにする

```
$ sudo apt install libncurses5-dev
```

- 仕組みはどうなっているのか？
 - どのようなことができるのか？

<https://www.kushiro-ct.ac.jp/yanagawa/ex-2017/2-game/01.html>

<https://www.kushiro-ct.ac.jp/yanagawa/ex-2017/2-game/doc/curses.html>

【List0.c】 上記URLのList0.c

```
#include <ncurses.h>

int main() {
    initscr(); // 端末制御の開始

    start_color(); // カラーの設定
    init_pair(1, COLOR_RED, COLOR_BLUE); // 色番号1を赤文字／青地とする
    bkgd(COLOR_PAIR(1)); // 色1をデフォルト色とする

    erase(); // 画面表示
    move(10, 20);
    addstr("Hello World");
    refresh();

    timeout(-1);
    getch(); // キー入力

    endwin(); // 端末制御の終了
    return (0);
}
```

【コンパイル&リンク】

```
$ gcc curses_hello.c -lncurses
$ ./a.out
```

- 動かしてください。
- 最低限の仕組みと使い方を理解してください。
- ncursesライブラリを使用した例を探してみてください。