Windowsmsys2ネットワークプログラミングWinsock

linux-network-programing」記事で、Linux標準ソケットによる簡単なTCP通信のサンプルプログラムを動かした。このプログラムをWindows上でも動かせるように試したので備忘録を残す。

msys2でのソケットライブラリー

---

msys2でも `#include <sys/socket.h>` などのLinux標準ソケット関数がそのまま利用できれば良いのだが、どうも使えない模様。

そこで、`#include <winsock2.h>` ヘッダーを指定しwinsock2ライブラリーを使う。winsock2ライブラリーは、Linux標準ソケット関数とかなり似ているが少し違うので修正が必要。
尚、winsock2ライブラリーは32bit版だけのようでmsys2のMINGW32環境で32bitアプリとしてコンパイルを行う。Windowsの64bitパソコンでも32bitアプリの実行は可能。

TCPソケット通信サンプルプログラム

以下の、サンプルプログラムでは、`WINSOCK_USE` の条件コンパイルで、Linux版と兼用できるソースにしている。

尚、システムコール時のエラー判定やタイムアウト監視など不十分なので、実際に使う場合はエラー処理などもよく吟味して利用する事。

単純なTCPサーバ

このサンプルは、接続してきたクライアントに対してHELLOという文字列を送信して終了。

tcp-test-server.c

```
#if defined(__MINGW64__)
#error Cannot compile in MINGW64 environment, MINGW32 required.
#elif defined(__MINGW32__)
#define   WINSOCK_USE        2.0        /* Only 32bit library */
#endif

#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#ifdef WINSOCK_USE
#include <winsock2.h>
#else
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
```

```
#endif

int
main(int argc, char *argv[])
{
#ifdef WINSOCK_USE
    WSADATA wsaData;
    SOCKET sock;
#else
    int sock;
#endif
    struct sockaddr_in server;
    char buf[32];
    int n;

#ifdef WINSOCK_USE
    // winsock2の初期化
    WSAStartup(MAKEWORD(2,0), &wsaData);
#endif

    /* ソケットの作成 */
    if ((sock = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
        perror("socket");
#ifdef WINSOCK_USE
        printf("error:%d\n",WSAGetLastError());
#endif
    }

    /* 接続先指定用構造体の準備 */
    server.sin_family = AF_INET;
    server.sin_port = htons(12345);
    server.sin_addr.s_addr = inet_addr("127.0.0.1");

    /* サーバに接続 */
    connect(sock, (struct sockaddr *)&server, sizeof(server));

    /* サーバからデータを受信 */
    memset(buf, 0, sizeof(buf));
    n = recv(sock, buf, sizeof(buf), 0);

    printf("%d, %s\n", n, buf);

    /* socketの終了 */
#ifdef WINSOCK_USE
    closesocket(sock);
    // winsock2の終了
    WSACleanup();
#else
    close(sock);
#endif
```

```
    return 0;
}
```

コンパイル方法

MINGW32シェルから下記コマンドを実行。

```
$ gcc -o tcp-test-server tcp-test-server.c
```

単純なTCPクライアント

このサンプルは、サーバに接続すると文字列を受信して表示。

tcp-test-client.c

```
#if defined(__MINGW64__)
#error Cannot compile in MINGW64 environment, MINGW32 required.
#elif defined(__MINGW32__)
#define   WINSOCK_USE        2.0        /* Only 32bit library */
#endif

#include <stdio.h>
#include <unistd.h>
//#include <string.h>
#include <sys/types.h>
#ifdef WINSOCK_USE
#include <winsock2.h>
#else
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#endif

int
main(int argc, char *argv[])
{
    struct sockaddr_in addr;
    struct sockaddr_in client;
#ifdef WINSOCK_USE
    int len;
    SOCKET sock0, sock;
    WSADATA wsaData;

    // winsock2の初期化
    WSAStartup(MAKEWORD(2,0), &wsaData);
#else
```

```
    socklen_t len;
    int sock0, sock;
#endif

    printf("*** TCP test(server) program start ***\n");
    usleep(500000);

    /* ソケットの作成 */
    sock0 = socket(AF_INET, SOCK_STREAM, 0);

    /* ソケットの設定 */
    addr.sin_family = AF_INET;
    addr.sin_port = htons(12345);
    addr.sin_addr.s_addr = INADDR_ANY;
    bind(sock0, (struct sockaddr *)&addr, sizeof(addr));

    /* TCPクライアントからの接続要求を待てる状態にする */
    listen(sock0, 5);

    /* TCPクライアントからの接続要求を受け付ける */
    len = sizeof(client);
    sock = accept(sock0, (struct sockaddr *)&client, &len);

    /*5文字送信*/
    send(sock, "HELLO", 5, 0);

#ifdef WINSOCK_USE
    /* TCPセッションの終了*/
    closesocket(sock);
    /* listen するsocketの終了*/
    closesocket(sock0);
    // winsock2の終了
    WSACleanup();
#else
    /* TCPセッションの終了*/
    close(sock);
    /* listen するsocketの終了*/
    close(sock0);
#endif

    return 0;
}
```

コンパイル方法

MINGW32シェルから下記コマンドを実行。

```
$ gcc -o tcp-test-client tcp-test-client.c
```

補足

- コンパイル時に、winsock2のライブラリー（libws2_32.lib）を明示的に-lws2_32指定しなくてもコンパイルエラーなく実行ファイルを生成した。
- MINGW64シェルの64bitコンパイル環境でも、-lws2_32を指定すればコンパイルはできたが、実行時の挙動がどうも怪しかった。

関連記事

- linux-network-programing
- □msys2□実行に必要なDLLを確認する

参考

---

1. winsockプログラミング 2. Windows Console でUDPの送受信 3. □□□□□にてSocket通信を行うには

From:
https://yokoweb.net/dokuwiki/ - 新石器Wiki

Permanent link:
https://yokoweb.net/dokuwiki/programing/linux-programing/network-programing/msys2-winsock-network-programing?rev=1626350825

Last update: 2021/07/15 21:07