パケットづくりではじめる ネットワーク入門

第25回 Windowsでのソケット・プログラミング

坂井 弘亮

FreeBSDやLinuxでTCP/IPによるネットワーク・プログラミングを行うための代表的なフレームワークは「ソケット」です.

ソケットはTCP/IPに限らず，プロセス間通信やTCP/IP以外のネットワーク通信などにも利用できる汎用的な通信フレームワークですが，TCP/IPの代名詞のように言われることも多いようです.

今回はWindows上でTCP/IPの通信を行うためのソケット・プログラミングについて説明し，簡単なソケット通信のプログラムをWindows向けに作成して動作確認してみます．またFreeBSD/Linuxでのソケット・プログラミングとの違いや，プログラムをFreeBSD/Linux/Windowsのマルチプラットフォームに対応させる方法，Windows上でネットワーク・プログラミングを行う際の注意点なども説明します.

FreeBSD/Linuxで TCP/IP通信する代名詞「ソケット」

Windows上でのソケットの利用法を説明する前に，ソケットについて簡単に説明しておきましょう.

● まずはプログラムを見てみる

FreeBSDやLinuxでIPv4でのTCP/IPによるコネクション型のストリーム通信を行うための，サーバ側とクライアント側のソケット・プログラミングについて説明します.

▶その1：サーバ側プログラム

サーバ側は，以下の手順でクライアントからの接続待ちをします.

- socket()でソケットを作成する
- bind()で接続を受け付けるアドレスやポート番号を設定する
- listen()で接続の準備をする
- accept()で接続待ちをする

実際のプログラム例は，**リスト1**のようになります．このプログラムはFreeBSDとLinuxの両方で利用できるように作成してあります.

リスト1では，30行目でsocket()によりソケットを作成しています．さらに52行目のbind()，56行目のlisten()により，ソケットに接続待ちをするための準備を行います.

実際の接続待ちは63行目のaccept()により行われます．ここでクライアントからの接続が来るまで処理はブロックし，接続を受け付けるとaccept()の戻り値として接続したクライアントとの通信用のソケットが返されます．accept()に使ったソケットは接続待ちのためのソケットなので通信には使いませんが，再度accept()を行えば，さらに接続を待ち受けることができます.

接続を受け付けたら，68行目でクライアントからのデータをread()によって受信し，その内容を73行目でwrite()によって標準出力に書き出しています．ファイル・ディスクリプタとソケットは同等であるため，read()/write()のようなシステム・コールは，ファイル・ディスクリプタとソケットの両者に対して利用できます.

▶その2：クライアント側プログラム

次にクライアント側について説明しましょう.

クライアント側は，以下の手順でサーバに接続します.

- socket()でソケットを作成する
- connect()でサーバに接続する

プログラム例を**リスト2**に示します．このプログラムも，FreeBSDとLinuxの両方で利用できます.

リスト2ではサーバ側と同様に，27行目でsocket()によりソケットを作成しています.

サーバ側と異なるのは，ソケットの作成後に39行目のconnect()ですぐに接続を行っていることです．このようにクライアント側は，あまり難しい手順を踏まずにサーバに接続することができます.

● ソケットは汎用的な通信フレームワーク

ここで，サーバ側やクライアント側で，接続のためのアドレスやポート番号の指定方法に注目してください.

サーバ側でのbind()や，クライアント側でのconnect()の際には，struct sockaddr_inという構造体が利用されています．これには**リスト1**

本連載のプログラムのソースコードは以下の筆者のホームページからダウンロードすることができます．ライセンスはKL-01というもので，組込み機器などでも自由に利用することができます.
http://kozos.jp/books/interface/ethernet2/