

.NET Frameworkで作る Windowsサーバー

作ればわかるアプリケーション
の動作とメカニズム

新連載 Webサーバーを自作する

秋月巖ソリューション事務所
秋月 巖 AKIZUKI, Iwao
<http://www.akizuki.co.jp/>

level				
1	2	3	4	5

Technology Tools
<input checked="" type="checkbox"/> Visual Basic
<input type="checkbox"/> Visual C#
<input type="checkbox"/> Visual C++
<input type="checkbox"/> SQL Server
<input type="checkbox"/> Oracle
<input type="checkbox"/> Access 2002
<input type="checkbox"/> ASP.NET
<input type="checkbox"/> Other:

Samples
<small>・この記事で取り上げたソースコードおよびサンプルプログラムは、 http://www.shoeisha.com/mag/windev/からダウンロード可能です。</small>



サーバーアプリ を作る

Microsoftが提供する製品にWindows Server Systemと呼ばれる一群がある。このカテゴリーに属する製品は、以前、BackOfficeと呼ばれていた。これらの製品はサーバーマシンにインストールされ、接続されたクライアントに機能を提供するサーバーアプリケーションである。

この新連載の主旨は、このようなサーバープログラムを主に、.NET Frameworkを用いて作成しようというものである。

もちろん、Microsoftの製品を買えば、それでもいいわけだが、作ってみることでどのように動作しているのかを理解することができるし、あるいは特殊な機能を追加したものを作成することもできる。ただしMicrosoftのような大きな会社が何十万円から何百万円で販売するものを簡単に作れるかという、それはやはり無理である。とはいえ、そのコアになる機能だけを作成するならば、それほ

ど難しいことではない。サーバーアプリケーションというのはたいそうなことをしているように見えるが、基本的な機能は簡単なものが多い。また、MicrosoftのサーバーOSには、高度な部分がコンポーネントなどの形で無料でOSによって提供されていることも少なくないので、そういうものを利用することも考えている。



技術的な範囲は 制約しない

サーバーアプリケーションには負荷が集中するので、その対応も必要である。そのような場合、たとえば、サービスの配信にIIS (Internet Information Services)を用いるというのも方法である。IISはWebサーバーだから、Webアプリケーションでしか利用できないような印象を持つ方もいるかもしれないが、XML Webサービスを利用したり、専用クライアントを使用することで汎用のサービスアプリケーションとして使用することもできる。とはいえ、そのように利用する場合でもプロトコルはHTTPに



なってしまう。これは、元がWebサーバーなので致し方ない。だから、たとえば、市販のメールクライアントプログラムとIISが対話をするというのにはありえない。

しかし、メールクライアントとWebアプリケーションを自作して、IIS経由でWindowsアプリケーションでメールをするような処理も可能である。このような場合、クライアントアプリケーションも作ることになる。たとえばメールサーバーであるExchange Serverが専用のクライアントアプリケーションを持つようなものである。だから、サーバーアプリケーションを作るといっても、クライアントのプログラム開発をしないわけではない。ただ、個人的には専用クライアントよりもWebブラウザで機能にアクセスできるほうが望ましいと思っている。そのような場合には、当然、ASP.NETのようなサーバースクリプトで実装することになるだろう。だから、プログラミングもWebクライアント、Webアプリケーション、TCP/IPクライアント、TCP/IPサーバーとさまざまな形で記述することになる。技術的な範囲は制約しないつもりである。また、市販されている製品の機能を単にまねるだけの場合もあれば、独自の機能を持つものを作ることもありえる。あるいは、同じ機能を別の方法で実装するというようなパターンもありえる。



さまざまな可能性

サーバーアプリケーションができると、インターネットでいろんな独自の機能を提供できるようになる。Webサーバー、メールサーバー、ストリーミングサーバーなど、インターネットでインパクトを与えたサービスは、どれもサーバーアプリケーションがシステムのコアになっているのはご存知のはずである。

というわけで、第1回の今回はWebサーバーを自作することにする。コードを見ればWebサーバーが、本来、どれほどシンプルな機能のものなのかがわかるだろう。Webの場合、サーバーよりもWebブラウザのほうがはるかに高度な処理が要求される。もちろん、Webブラウザコントロールのようなものを使えば一瞬で独自Webブラウザが作成できるが、それはまた別のことだ。

もちろん、ここで作ったWebサーバーはWebアプリケーションを実行できないどころか、バイナリの処理もできない。だから、扱えるのはあくまでもハイパーテキストファイルだけである。とはいえ、バイナリの扱いはもちろん、Web

アプリケーションを実行できるようにするのも、実はそれほど難しいことではない。Webアプリケーションの機能をVisual Studio .NETで作成してDLLとしてコンパイルし、必要なところで呼び出せばいいだけである。もう覚えている人も少ないだろうが、Visual Basic 6.0で提供されたWebクラスと同じような動作形態となる。ただし、ASPのようにHTMLファイルにプログラムコードを埋め込むような記述法のプログラムを実行することはできない。コードの中でリアルタイムでコンパイラを実行するという手もなくはないが、あまりまともな方法とはいえない。ASP.NETのようなWebフォームをサポートするものはさらに困難である。WebフォームというのはWebサーバー側の問題ばかりでなく、開発ツールにも依存したものだからである。



Webサーバーはファイルサーバー

WebサーバーとはHTTPサーバーのことであり、ハイパーテキストのファイルサーバーである。といっても実際には、どのようなファイルでもサービスできる。今、考えたのだが、FTPを扱えるWebブラウザならば、FTPサーバーを相手にWebブラウジングをすることができそうである。Internet ExplorerはFTPクライアントとして使えるが、それはあくまでファイルの取得のためであり、ファイルの閲覧のためではないはずである。しかしInternet Explorerに取得したファイルを自動的にウィンドウに表示する機能さえあれば、FTPサーバーとInternet Explorerを使ってWebブラウジングもできることになる。まあ、FTPサーバーをWebサーバーとして使うことに特にメリットがあるわけではないのだが、HTTPのファイルアップロード機能が弱いことを考えれば、少しはメリットがあるかもしれない。



FTPサーバーはWebサーバーにもなる

と、ここまで書いたところで、もしやと思い、FTPのルートフォルダにindex.htmファイルを配置し、Internet Explorerの「アドレス」に次のように指定したところ、ファイルの内容をWebページとして表示することができた。

```
ftp://localhost/index.htm
```