

Web を支える情報通信技術 (5)

Web サーバの構築

情報基盤センター
佐藤周行・中山雅哉・西村健

1. はじめに

さて前回までで、Web の基本的な仕組みをページの表現方法からネットワークを流れるデータに至るまで、元になる仕様をたどりつつ説明してきた。今回は前回までに習得した知識を総動員して実際に Web サーバを立ち上げてみよう。自分で手を動かして動作を目で見ることでさらに理解が深まるはずである。

注意

次回以降本講義では今回構築した Web サーバを元に演習を行なっていく。サーバのドメイン名が変わると演習に支障が出る場合があるため、極力今回座っている席に次回以降も座っていただきたい。

2. Web サーバって何？

これまでの講義では、コンピュータおよびネットワーク上の動作をブラウザの挙動を中心に見てきた。それではブラウザの対向にある「Web サーバ」とは何だろうか？

Web サーバに限らず一般的にサーバとは、あるプロトコルに従ってクライアントからの要求を受け付け、それに対する応答を返す「もの」¹のことである。Web サーバの場合は HTTP というプロトコルに従って Web クライアントから”GET”(等)によって要求された HTML ファイル(等)を返す。(プロトコルは通信路上のデータの書式を規定するだけで、サーバ上でのデータの扱い(どのような形で保管する？どこに保管する？)は規定していないことに注意)

ブラウザのように目に見える形で動作しないので実感しにくいですが、サーバは Word や Excel と同じでプログラムの形で存在しており、起動/終了も(何らかの方法によって)行なうことができる。

演習

今ログインしているコンピュータ上でいくつかのサーバが稼動しているか確認してみよう。

¹ 一般的にサーバと言った場合、コンピュータ上で動作するソフトウェアを指す場合とサーバソフトウェアが動作している物理的なコンピュータを指す場合があるが、この講義では特に区別なく扱う。基本的にソフトウェアを指し、「サーバ上」のような文脈ではコンピュータを指す場合がある。

iMac 端末上でのサーバの稼働状況を調べるには、Terminal アプリケーション内で netstat コマンドを使う。(state)の欄が LISTEN となっているものが見つかるだろう。その行の Local Address を見れば、どんなサービスを LISTEN (要求を受け付けるために待機) しているかが分かる。

```
$ netstat -an
...
tcp4      0      0      127.0.0.1.631    *.*      LISTEN
...
```

Q1. Web サーバは世界中で何台くらいあるのだろうか？

3. Apache って何？

今回の演習には Web サーバとして Apache を使用する。

Web サーバにはその実装によって多くの種類が存在し、Apache(正確には Apache HTTP Server)もその一つである。他に代表的なものとして Microsoft の IIS(Internet Information Services)がある。

Apache は世界中でもっとも使われている () Web サーバで、無料で、動作が非常に安定しており、セキュリティ面での高い実績を誇る。詳細は以下の Web サイトを参照してほしい。

<http://httpd.apache.org/>

ソフトウェア、特にサーバソフトウェアが広く使われていること、長く使われていることは重要な指標の一つである。なぜなら、攻撃対象としてそれだけ多くの人目に晒されてきているということであり、逆説的に多くの不具合・脆弱性が修正されていると考えられるからである。(もちろん開発体制や開発ポリシーなど他にも重要な指標はたくさんあるが)

Netcraft 社による 2007 年 5 月の調査によれば、全世界の Web サーバの 56% が Apache を使用している。詳細は以下の Web ページにある。

http://news.netcraft.com/archives/web_server_survey.html

この調査を見れば、118,023,363 台の Web サーバを調査したことが分かる。ただし、

- 調査できたのがこれだけということで、調査から漏れた Web サーバも存在する。

- ホスト名が異なるだけで実体は 1 台のサーバでも重複してカウントされている。
- 実際に意味のあるコンテンツを提供しているかどうかは別問題。
- コンピュータの台数を表しているわけではない。(Web サーバの機能として複数サイトのコンテンツを一台の物理的なコンピュータでサービスすることができるため (virtual host 機能))

ことに注意が必要である。

Q2. Netcraft はどのような調査方法をとったのだろうか？ある Web サーバがどのようなサーバソフトウェアを使用しているかの調査のためには何を調べればよいだろうか？

4. Web サーバ構築演習

閑話休題。

先にあげた Apache のサイトにはダウンロードのためのリンクがあり、ソースをダウンロードしてインストールすることも可能であるが、時間の制約もあり、ここでは iMac 端末にすでにインストールされている Apache を使用する。ただしさきほどの netstat の演習で見たように起動はされてないようである。

なお、Apache は現在複数のバージョン(1.3.x, 2.0.x, 2.2.x)がメンテナンスされており、今回使用するのはバージョン 1.3 である。各バージョンで設定方法は若干異なり、以下で述べる方法は他のバージョン、つまり現在メジャーになっているバージョン 2.x ではそのまま適用できない部分もある。この演習は方法を学ぶのではなく動作原理を理解することに焦点を当てており、このような方式をとっていることをご理解いただきたい。

上でも書いたように、Web サーバの設定方法に共通仕様があるわけではなく、それぞれの Web サーバ開発者 / 組織が好き勝手に設定方法を定義している。つまり Apache と IIS では設定方法は全く異なる。これは各 Web サーバの独自性を考えれば当然のことではあるが、プロトコルや HTML が RFC 等で共通仕様が厳密に定義されていたのとは対照的である。あくまでここでの目的は設定ファイルの書式などの詳細を知ることではなく、演習を通して Web の仕組み、Web サーバの役割を実感してもらうことであるという部分をもう一度強調しておきたい。

話を進める前に誤解を避けるために書いておくが、ソフトウェアとしてのクライアント / サーバとコンピュータとしてのクライアント / サーバの概念は独立であり、netstat の演習で見たように HTTP のクライアントコンピュータであるみんなが使用している iMac 端末でも各種のサーバが既に稼働しているし、今回のように自分で Web サーバを稼働させることも可能である。

注意

外部ネットワークから丸見えな環境で Web サーバを稼働させるのは攻撃してくれと言っているようなもので、推奨しない。この講義の演習以外で Web サーバを立ち上げる場合は十分注意してほしい。

4.1. とりあえず起動してみる

Apache の起動は通常のアイコンをクリックする方法ではできない。また、稼動中も Dock に表示されることはない。起動は Terminal アプリケーション内で `apachectl` コマンドを実行する。

```
$ apachectl start
...
```

どのような結果になっただろうか？

おそらく記録（ログ）を保存するためのファイルがオープンできないというエラーメッセージが表示されたはずだ。これは Apache が管理者権限（正確には root 権限）を持つユーザが起動することを大前提としてインストールされているためである。自分がサーバ（この場合 iMac 端末）を所有している場合は管理者権限を持つユーザになってそのまま Apache を起動することができる²が、ECCS 環境では不可能である。以下で設定を変更して自分達でも起動できるようにしよう。

4.2. 設定変更の方法

Apache では設定変更は設定ファイルを編集することにより行なう。設定ファイルには、どのポートでどのような要求にどのように答えるか、また、Web コンテンツはどこにあって、記録（ログ）をどこに保存するか、等の全ての情報が書き込まれている。テキスト編集ソフト一つあれば設定変更できるというメリットはあるものの、このような方式に慣れていない初心者にはとっつき難いものとなっている。

Apache の設定ファイルは以下の場所に存在する。()

```
/private/etc/httpd/httpd.conf
```

ただしこちらにも管理者権限が必要で、権限がないとファイルを編集することができない。そこで自由に読み書き可能なホームディレクトリ以下を使用し、設定ファイルをここにコピーして使用することにする。ホームディレクトリのパスは各自で異なるが

```
/home02/utnish
```

のようなパスになる。Terminal アプリケーションを開いたときのカレントディレクトリで

² 実は管理者権限を持つユーザならシステム環境設定から Apache の起動・終了が可能であるが、ここではその方法は割愛する。

ある。自分のホームディレクトリのパスは `pwd` コマンドにより確認できる。また Finder ウィンドウの左欄で家アイコン(と自分のユーザ名)が表示されている項目をクリックすると表示される場所でもある。

```
$ pwd
/home02/utnish
```

Mac 内での「場所」は、パス（絶対パス）という形で表現される。（ URL ）
/（スラッシュ）から始まり、各階層のディレクトリ名を/を挟んで列挙し、最後にファイル名を記したものである。Finder において任意のファイルがどこに存在するかを調べるには、右クリックして「情報を見る」を選択して表示されるウィンドウの「一般情報」の「場所」を見ればよい。そのファイルの絶対パスは、「場所 +/+ ファイル名」となる。また、本講義では一貫して「ディレクトリ」という呼び方をしているが、「フォルダ」と同一のものを指す。

ちなみに Apache の主要ファイル構成は下記のようにになっている。

<code>/usr/sbin/apachectl</code>	起動用ラッパースクリプト
<code>/usr/sbin/httpd</code>	Apache HTTP Server プログラム
<code>/Library/WebServer/Documents/...</code>	コンテンツ格納場所（とりあえず移動不要）
<code>/usr/libexec/httpd/...</code>	モジュール（移動不要）
<code>/private/var/log/httpd/...</code>	ログファイル格納場所（移動要）
<code>/private/etc/httpd/httpd.conf</code>	設定ファイル（移動要）

4.2.1. 作業場所の作成

まず設定ファイルのコピーを行なう前に、今回の演習で使用するファイル全てを格納する専用のディレクトリを作成しよう。Finder でホームディレクトリを表示し「ファイル」メニューの「新規フォルダ」で作成できる。名称は「httpd」とする。

上記で作成した専用ディレクトリに設定ファイルをコピーする。このファイルは Finder では表示できないので、Terminal アプリケーションから行なう必要がある。以下のコマンドを実行すれば、ホームディレクトリの httpd ディレクトリ内に httpd.conf ファイルがコピーされる。

```
$ cp /private/etc/httpd/httpd.conf httpd/
```

編集はテキストエディットにより行なう。httpd.conf には関連付けされていないのでダブルクリックでは起動できない。代わりに、「アプリケーション」ディレクトリにある「テキストエディット」アイコンをダブルクリックして起動し、「ファイル」メニューの「開く」から httpd.conf を指定して開く。その際、オプションの「リッチテキストのコマンドを無視する」にチェックを入れるようにすると無用の混乱を避けられる。

起動時に上記でコピーされた設定ファイルを参照させるには、apachectl の書き換えを行なわなければならない。やはり直接書き換えることは不可能なので設定ファイルと同様に Terminal アプリケーションから下記コマンドでコピーし、

```
$ cp /usr/sbin/apachectl httpd/
```

テキストエディットで以下の行を修正する。各自のホームディレクトリで置き換えること。

```
< HTTPD=/usr/sbin/httpd
---
> HTTPD="/usr/sbin/httpd -f /home02/utnish/httpd/httpd.conf"
```

もう一つ今後のために以下の行も書き換えておくと良い。

```
< PIDFILE=/private/var/run/httpd.pid
---
> PIDFILE=/home02/utnish/httpd/httpd.pid
```

ファイルを変更したら実行する前に忘れずに保存すること。

今書き換えた apachectl を実行するには、Terminal アプリケーションで apachectl の代わりに httpd/apachectl を使用する。(カレントディレクトリの中の httpd ディレクトリの中の apachectl ファイルを実行する、の意)

```
$ httpd/apachectl start
...
```

もちろんまだ設定ファイルを書き換えていないので、先ほどと同じエラーメッセージが出るはずである。

4.2.2. 設定ファイルの書き換え

テキストエディットで httpd.conf を開き、以下の手順で内容を書き換えよう。それぞれ簡易な説明が付いているが、自分が何を变更しているのかを意識しながら行なうことが望ましい。

設定ファイルに関するドキュメントは Web 上に存在する。さきほど紹介した Apache のサイトから左メニューの”Documentation”の下にある”Version 1.3”をクリックすると表示される。設定ファイル内の記述に関して詳細を知りたい場合は「リファレンスマニュアル」の「実行時の設定方法」にある索引から知りたいキーワードを探すと良いだろう。

```
< ServerRoot "/usr"
---
> ServerRoot "/home02/utnish/httpd"
```

全体的なディレクトリ起点の指定。以降に出てくるログファイル等はここを起点にして指定することができる。

```
< PidFile "/private/var/run/httpd.pid"
---
> PidFile "httpd.pid"
< ScoreBoardFile "/private/var/run/httpd.scoreboard"
---
> ScoreBoardFile "httpd.scoreboard"
```

ファイルの詳細には立ち入らないが、書き込み可能なパスを指定する。

```
< LoadModule config_log_module "libexec/httpd/mod_log_config.so"
---
> LoadModule config_log_module "/usr/libexec/httpd/mod_log_config.so"
以下同様
```

元々上で変更した ServerRoot ディレクトリ/usr 起点の指定であったが、元のファイルを参照するために/usr を補っている。全ての LoadModule の行に対してこの変更を行わなければならない。

```
< Port 80
---
> Port 8080
```

前回説明があったように、ポート 80 番でサーバを立ち上げるには管理者権限が必要である。今回は 8080 番ポート(http-alt)を使用する。

```
< ErrorLog "/private/var/log/httpd/error_log"
---
> ErrorLog "error_log"
< CustomLog "/private/var/log/httpd/access_log" common
---
> CustomLog "access_log" common
```

ログファイルの保存場所を変更する。上記によりエラーログは作業場所 httpd ディレクトリの中の error_log ファイルに記録されるようになる。

```
< Include /private/etc/httpd/users/*.conf
---
> #Include /private/etc/httpd/users/*.conf
< Include /System/Library/WebObjects/Adaptors/Apache/apache.conf
---
> #Include /System/Library/WebObjects/Adaptors/Apache/apache.conf
```

追加の設定ファイルを読み込むことによって設定が不用意に上書きされるのを防止するためコメントアウト（無効化）する。単にこの2行を削除しても構わない。

以上である。ファイルを変更したら実行する前に忘れずに保存すること。

注意

この設定ファイルでは、http://...:8080/~ユーザ名/という URL によってホームディレクトリの Sites（「サイト」）ディレクトリの内容が公開されてしまう。ディレクトリの中に秘密情報を保持している場合は注意すること。

4.3. Apache が起動できたことの確認方法

apachectl コマンドによって Apache が起動できたかどうかを確認するには、まず Terminal 上でエラーが表示されていないことを確認する。ここで” start: httpd started”のみ表示されるようになれば、第一段階は突破である。

次に、設定ファイルに設定したエラーログファイル(error_log)にエラーが記録されていないことを確認する。ログファイルにはエラー以外の情報も書き込まれるが、日付の次が”[crit]”(critical)となっている行がないことを確認すれば良いだろう。例えばエラーメッセージが”(13)Permission denied: make_sock: could not bind to port 80”と記録されていれば、80 番ポートを使用しようとして root 権限がないために失敗したことを示している。

次に、netstat コマンドを実行して指定したポートがサービスを提供できる状態になって

いることを確認する。”tcp4 ... (中略) ... *.8080 ... (中略) ... LISTEN”となっている行があるかどうかで確認できる。

最後に、実際に Safari でアクセスしてみる。URL のドメイン名の部分は全てのコンピュータで自分自身を指すように設定されている IP アドレス 127.0.0.1 を利用すると良いだろう。すなわち Safari のアドレス欄に

```
http://127.0.0.1:8080/
```

と入力し移動してみよう。「あなたの予想に反して、このページが見えているでしょうか？」と書かれたページが表示されれば、Apache の起動は成功である。

4.4. Apache の終了

起動した Apache を終了するには `apachectl stop` を使う。

```
$ httpd/apachectl stop
```

また、起動したままログアウトした場合も強制終了させられるようである。

Q3. 各端末のドメイン名は端末名(caXXXXX 等)に `.ecc.u-tokyo.ac.jp` を繋げたものである。他の人の端末名を教えてもらい、Web サーバにアクセスしてみよう。

Q4. 自分が立ち上げた Web サーバ上に自由にコンテンツを作成してみよう。設定ファイルのどこを変更する必要があるか？

Q5. 世の中の Web コンテンツを実現するためにサーバにどのような機能が必要と考えられるか？それが Apache で実現されているかどうか調べてみよう。

Q6. 今回構築した Web サーバは、ECCS の外からはアクセスできない。それはなぜか？ネットワーク層を勉強している者は ECCS のネットワーク構成から、そうでない者はブラウザの挙動をシミュレートすることによって明らかにせよ。