

## サーバサイドプログラミングの演習環境

山 守 一 徳\*

サーバサイドプログラミングは、世の中の多くの Web システムで用いられており、それらを大学で教えることは重要なことであると思われる。そこで、今年度の電子計算機実習の授業において、パソコン教室の環境を使い、サーバサイドプログラミングのシステムを用いた演習を行った。その結果、現在のパソコン教室の環境では演習がしにくいことがわかり、改善策について提案する。本文では、サーバサイドプログラミングについて紹介し、その中でも Servlet を用いたシステムを開発したため、そのシステムについても紹介する。

キーワード : Java、Servlet、DNS、Web、サーバサイドプログラミング

### 1. はじめに

情報教育課程の情報教育コース3年生向けに開講されている電子計算機実習では、システム開発に関する能力を高めるための演習が行なわれている。近年、世の中では Web を用いたイントラネット又はインターネット用のシステムが広まっており、システム開発の演習を行なう場合にも、それらに類似したシステムに取り組むことが重要である。そこで、サーバサイドプログラミングを用いたシステムを演習に用いた。その経験を通じて、現在のパソコン教室の環境では演習がしにくいことがわかり、改善策について提案する。

### 2. サーバサイドプログラミングとは

クライアントからの要求に応じて処理を行ない、返答を行なうサーバのうち、有名なものに Web サーバがある。以前は、クライアントに返す HTML ファイルは固定のもので、動的に変更させることはできないページが主流だったが、最近では、クライアントからの要求に応じて返すページの内容を変化させることが頻繁に行なわれている。例えば、データベースへの検索

要求に対して、検索結果を表示させるようなページなどである。その動的に変化するページを生成する場合、クライアント側でプログラムを実行させるのではなく、サーバ側でプログラムを実行させるようにコーディングを行なうことをサーバサイドプログラミングと呼ぶ。

サーバサイドプログラミングで用いるプログラムとしては、CGI (Common Gateway Interface) がこれまで主流であった。これは、sh、csh などのシェルスクリプトやコマンドラインで利用できるコマンドによるバッチプログラムや Perl、C、C++、Visual Basic などの言語を用いてコーディングを行なうものである。その中でも CGI と Perl の組み合わせが主流であったが、CGI は、クライアントからの要求がある度に、プロセスを起動しては終了するという動作を行なうため、サーバに負荷がかかり、クライアントの体感速度が悪くなる現象が起きるという問題がある。Web サーバに Apache を用いる場合には、mod-perl というモジュールを組み込むとプロセス起動のオーバーヘッドを軽減させることができるが、CGI のプログラムはセキュリティホールとなる危険性が高いことと実行形式がサーバの OS に依存することから使うことは避けられつつある。Web サーバを提供しているサービス業者でも CGI の利用は不可としているところもある。

\* 三重大学教育学部情報教育講座

一方、サーバサイドプログラミングに PHP (PHP:Hypertext Preprocessor) というスクリプト言語を用いることも多い。これは、Oracle、Infomix、Sybase、MicrosoftSQLServer、ODBC、MySQL、PostgreSQL などの多彩な RDBMS (リレーショナルデータベース管理システム) をサポートしているのが特徴であり、これらのデータベースシステムと合体して用いられることが多い。また、Web サーバに Apache を用いる場合には、mod\_php というモジュールを組み込むと要求の度にプロセスの起動・終了が発生するというオーバーヘッドの問題を解決させることができる。

PHP は、スクリプトエンジンとも呼ばれるインタプリタがスクリプトの解析と実行を行なうのであるが、そのスクリプト言語をコンパイルしてソースコードを見えなくしておくことができるため、業者がプログラムを開発し商品として提供する場合には、この PHP を用いる場合が多い。

PHP と対抗して、本命になりつつあるサーバサイドプログラミングに Servlet がある。今回は Servlet を演習に採用した。Servlet は、サーバ上で動く JAVA プログラムであり、詳細は後述する。JAVA は別授業で教えていることもあり、今回の演習の受講生には馴染みがある言語である。

その他のサーバサイドプログラミングに、Microsoft 社が開発した ASP (Active Server Pages) があり、これは当初、IIS (Internet Information Server) 上でのみ動作したが、Apache など徐々に動作する Web サーバが増えている。しかし、Microsoft 製品との親和性から、ASP を使うならば IIS を利用する場合が多いと思われる、その場合にはセキュリティアップデートを頻繁に行なわなければならないという問題がある。

それ以外に主流でないサーバサイドプログラミングには、Python や国産の Ruby などがある。どちらもインタプリタ型のスクリプト言語

であるが、Python は日本語処理機能が弱く日本では普及していない。

### 3. 演習に用いたシステム

#### 3. 1 システムの概要

演習に用いたシステムは、DNS (ドメインネームシステム) 登録を Web ブラウザを通して行なうことができるシステム<sup>1)</sup>である。学内には、DNS サーバが多数存在し、各 DNS サーバの管理者はそれぞれのセグメントまたはドメインの範囲内において、IP アドレスとマシン名の対のデータを追加・変更・削除したりして管理している。この状況をさまざまな面で改善すべく独自開発したシステムである。

このシステムは、(1) DNS 登録を 1 台のマシン上でありながらも複数の管理者が各自の管理領域のみ扱うことができ、従来の分散管理体制と同様の体制で運営することができる。(2) DNS 登録せずに IP アドレスを割り当てたマシンの管理も同時に行うことができ、従来の 2 重管理をなくすることができる。(3) 学内の全マシンの情報がデータベースの中に登録されるために、情報処理センターのような対外接続管理部門において、セキュリティ上必要となる学内全マシンの情報一元管理をすることが可能となる。以上の 3 つが可能となるシステムである。

この結果、(1) 学内のトータル発生費用を抑制することができる。(2) バッチ当て作業など全管理者の作業時間を抑制することができる。(3) セキュリティ対策を推進させることができる。以上の 3 つの利点を生むことができる。

実運用されれば、本学の場合で 53 台の DNS サーバが 1 台に集約でき、セカンダリネームサーバを用意するなどの安定稼働のための配慮を加えれば、十分に効果を達成し運用可能であると思われる実用的システムである。

本システムのログイン画面を図 1 に示す。

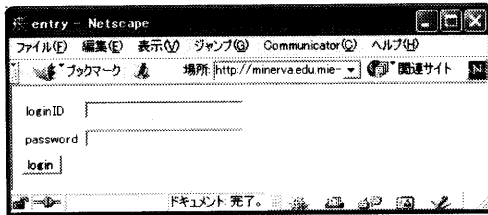


図1 ログイン画面

ログイン直後の画面を図2に示す。

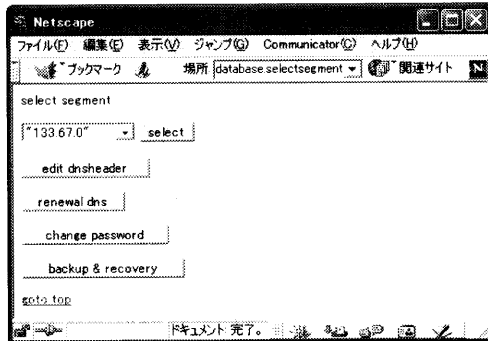


図2 ログイン直後の画面

データベースのレコードの項目としては、IPアドレスとマシン名だけでなく、利用者名、管理者名、設置場所、用途、機種種別、ドメイン名、MACアドレス、備考の情報を持つこととした。入力する画面の例を図3に示す。

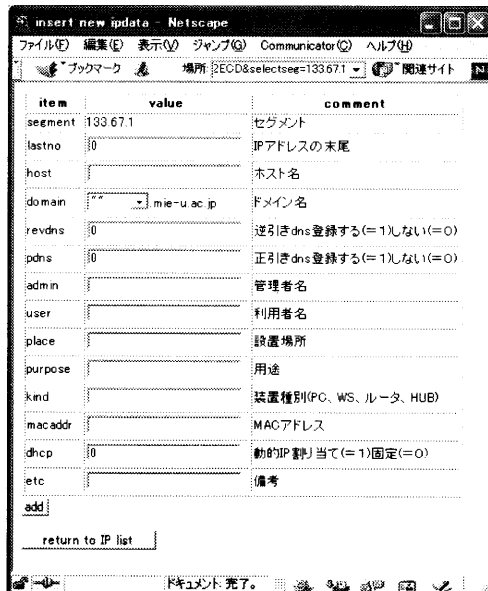


図3 入力画面例

また、編集する画面の例を図4に示す。

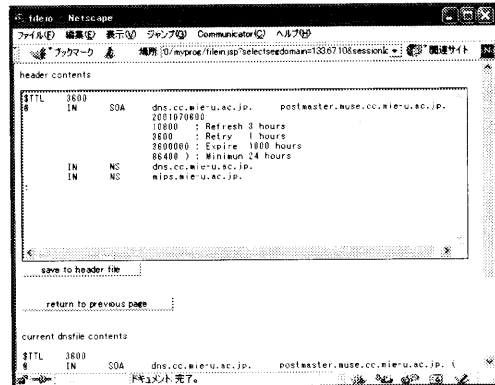


図4 編集画面例

DNSの設定ファイルの前半部分(SOAレコード、CNAMEレコード、MXレコードなどの部分)は、セグメントおよびドメインごとにテキストファイルの形で保存されている。CNAMEレコードの変更やMXレコードの変更などはこのファイルをWebブラウザを通して編集する。

### 3.2 システムの実現方法

Servletは、Servletコンテナと呼ばれるプログラム上で動くJavaのプログラムであり、演習に用いたシステムは、基本的にそのServletプログラムである。

Servletを用いる利点は、Javaプログラムであるので、ServletをサポートするWebサーバとJVM(仮想マシン)さえあれば、さまざまなプラットフォーム上で動作する点にある。また、ServletがServletコンテナ上に常駐することで、プロセスの生成・消滅を繰り返すというCGIのオーバーヘッドの問題も解決している。他にもServletは、一般的なJavaプログラムと同様に、JavaBeansと呼ばれる部品化したプログラムを呼び出せたり、JDBCと呼ばれるデータベースにアクセスするためのインタフェースを利用できたりする。

演習に用いたシステムでは、データベースにアクセスする部分にJavaBeansを用い、その中からJDBCを利用している。JavaBeansを用い

ることで、データベースをアクセスする SQL 文付近のプログラム部分を部品として閉じ込め、採用するデータベースシステムに依存した箇所を集約することができる。また、SQL 文付近のプログラムをクライアントから隠蔽し、データベースにいたずらされることを防ぐ効果もある。

一方、表示画面部分は JSP (Java Server Pages) を用いている。JSP は、HTML の中に独自のタグを埋め込み、その中に Java コードを記述することによって、画面のデザインを容易に記述できるという特徴がある。もちろん、Java コード部分から JavaBeans を呼び出せたりする。JSP は JSP コンテナの上で動作するが、Servlet コンテナが、JSP コンテナの機能も備えているため、環境構築に苦労することはない。

演習に用いた Servlet コンテナとしては、Tomcat-4.0.3 を採用し、Web サーバとしては、Apache-1.3.27 を用いた。JVM には、Sun Microsystems 社の Java 2 Platform Standard Edition (J2SE)-1.4.0 を用いた。さらに、データベースとしては、PostgreSQL-7.2.1 を採用し、JDBC2 を用いた。DNS サーバのソフトとしては、bind-8.3.3 を用いた。用いた OS は、Solaris2.8 であり、ハードウェアは、SUN Blade 100 である。

開発したプログラムは約 4400 行から成っている。JSP ファイルが 13 個、Java ファイルが 13 個、うち 7 個は JavaBeans から成る。

## 4. 演習内容

### 4.1 課題の進め方

システム開発の演習の内容は、(1) ユーザからの要求（今回は教師からの要求）を聞き取り、要求仕様書を作成すること、(2) システム設計を行ない、システム仕様書を作成すること、(3) システム仕様書の内容に基づきシステム開発を行なうこと、(4) ユーザ（システムの発注者）

にプレゼンテーションを行なうように、システム特徴を PowerPoint を用いて発表し、作ったシステムのデモを見せることである。以上の内容を数コマの授業で行なう。そのためには、システムをある程度作成したものを教師が用意し、そのシステムを元にして学生が独自に改善を加えるという手法を用いた。演習の受講生は 42 名存在し、42 種類の改善提案に対応することは教師側に無理があるので、1 グループ 6 名から成る 7 つのグループに分け、グループ毎に改善提案を行ない、各仕様書にまとめることとした。

### 4.2 用いた演習環境

パソコン教室には 40 台の Windows95 の学生用パソコンが存在し、そこに急遽別の箇所から持ってきたパソコンを加え、42 台の学生用パソコンを用意した。さらに、Solaris 2.8 のサーバを 1 台使い、そのサーバ上に、Tomcat、Apache、PostgreSQLなどをインストールした。7 つのグループ用に、それぞれのソースファイルをディレクトリを区別してサーバ上に置き、そのファイルをパソコンからログインして直接編集するかまたは編集済みファイルをファイル転送するなどして、ファイルに修正を加えることを行なった。7 つのグループは、Web ブラウザから URL を指定してアクセスする時に、別々の URL を指定することで各自のソースファイルをアクセスすることができる。

## 5. わかった問題点と今後の解決策

### 5.1 問題点

学生は、別授業で Java を学習しているが、今回の学生は JSP や JavaBeans は初めてであったため、プログラムを修正できるようになるには、学生の自習時間が必要であった。UNIX のサーバ上で、Java をコンパイルしたりするのに、UNIX の環境変数の設定とか UNIX のコマンドに慣れておらず、四苦八苦しながら作業している状態であった。また、データベースの授業も別に開講されているが、全学生が受講している訳ではなく、データベースの

テーブルを構築するために SQL コマンドを実行するとかテーブルを変更するにはかなりの知識を必要とした。そのため、データベースそのものを改善するグループは現れなかった。ほとんどのグループは改善項目として、画面表示の日本語化やヘルプ表示の追加を挙げた。JavaScript を用いたグループもいた。

問題となったのは、Servlet のプログラムの変更を反映させる場合、Tomcat を再起動し、Apache を再起動させる必要があり、その再起動が問題となった。各グループごとに、Tomcat や Apache が動いている訳ではないので、各グループごとに再起動したいとするタイミングがずれて、1つのグループによる再起動が他グループに影響することがあった。

また、PostgreSQL 内にデータベースを一つ構築し、その中の同じテーブルに対し、7つのグループのプログラムがアクセスする形のシステムを最初に提示したために、テーブル内のデータがあるグループが変更した場合に、他のグループに影響することが起きた。特に、ユーザ管理テーブルの中の特権ユーザのパスワードの値を変更した場合、他グループで特権ユーザによるログインができない等の問題が生じた。今回は、教師がパスワードを初期に戻す作業を実施して演習を継続させたが、テーブルはグループ毎に用意した方が良く思われた。

学生にとっての大きな問題としては、Servlet のプログラム修正を自宅で実施し確認することができないという点があった。動作確認するためには Servlet コンテナが動く Web サーバが必要であり、教室で用いたサーバは、学外からアクセスができないようにファイアウォールで守られている環境にあった。そのため、学生の自宅からはアクセスができなかった。サーバを学外からアクセス可にすると、この授業以外でもそのサーバを用いているため、ネットワーク攻撃された場合に他の授業への影響が大きい。自宅から演習を行いたい学生に対しては、学校に

出てくるように指導するしかなかった。

## 5. 2 解決策

### (1) Tomcat+Apache の再起動問題

この問題に対しては、サーバをグループ毎に全部で7台用意すると解決する。それには、学生用パソコンを Windows95 でなく、サーバとして立ち上げられるようにすれば良い。具体的には、パソコンのハードディスク容量は 80GB ぐらいが標準となっていて、大きなハードディスクを持たせれば、2つの OS をインストールでき、電源を入れた時に、OS の選択ができるようにすれば良い。このパソコンの管理の仕方として、ディスクのドライブを丸々コピーして CD-R に退避し、壊れた時には退避した CD-R から復元するというユーティリティソフト<sup>2)</sup>を用いて管理すると良い。サーバとして立ち上げられるパソコンとそうでないパソコンとを区別する場合には、別々の CD-R に保存すれば良い。最近では、Linux と Windows 系 OS とのデュアルブートにし、Linux を立ち上げて UNIX の授業を行なっている他大学も存在する。教室内の全端末をデュアルブートにできれば、CD-R も数枚にはなるが1種類で済む。

### (2) テーブルが共有している問題

データベースのテーブル変更を行なうぐらいにプログラムの改善を行なわせるには、テーブルはグループごとに用意するのが良い。先のパスワード変更のトラブル事例も起きなくなる。この実現方法は、1台のサーバにおいても可能ではあるが、7台のサーバを用意した場合には、プログラムそのものにグループごとの差異部分を加える必要がなくなる。この7台体制にした場合、演習中にデータベースのテーブルを壊されて復活させなければならない教師の作業が7倍になる可能性もあるが、全グループに影響が出るよりは良い。TA (ティーチングアシスタント) という先生をサポートしてくれる支援者がいると大変助かるので、TA も導入すべきである。

### (3) 自宅から演習ができない問題

この授業のための学外からアクセス可能な専

用サーバを1台用意するという方法が考えられるが、管理者の負担が増えるとか学生が大学に来なくなるとか、一長一短がある。管理者でもある一個人としては、大学に出てきて演習するという方を選びたい。

## 6. おわりに

サーバサイドプログラミングの演習を行なったことは、学生には大きな刺激になり、多くの技術を垣間見ることができたと思われる。演習に用いたシステムでは、JAVA、Servlet、JSP、JavaBeans、SQL、PostgreSQL、Tomcat、Apache、UNIX コマンドだけでなく、DNSサーバの技術も必要であり、ネットワークの勉強にもなったと思う。ただし、グループごとに演習を行なうため、個人の技術習得には差が出ている。しかし、グループ内で質問し合い、各自の能力に合わせた習得ができていると思う。今回は、同一システムをグループ毎に改善するという方法を取ったために、グループ間でも質問し合い、学生間で勉強し合うこともできていた。

このようにサーバサイドプログラミングを学習させることは、社会に出た時に役立つ人間に育てるために必要なことであると思う。新しい技術を垣間見させ、本人の勉強不足を認識させると同時に、新しい技術への取り組み方を経験させることは、大いに役に立つはずである。単に、簡単なプログラム開発を教えるだけでは、現在社会で広く利用されている技術との乖離が大きい。単体のプログラムさえも満足に作れない学生もいるが、是非とも、サーバサイドプログラミングは教えるべきである。それには、現在のパソコン教室の設備が貧弱である。幸いにもパソコンの値段は下がってきており、高性能を期待しなければ4万円でパソコン本体が買える程度にまでなってきている。パソコンを買い換え、デュアルブートにして、サーバ側の教育を行なえるならば、今回見つかった問題の2つは解決する。

次に教えるべきこととしては、今後世の中はインターネット全体に分散配置されたアプリケーションやデータを柔軟に利用可能にするようになるとわれ、そのための分散 Web アプリケーションの実現方法や XML (eXtensible Markup Language) という言語を教育していかなければならないと考える。

また、現在はサーバでプログラムを実行させ、クライアントは GUI による表示を担当するという構成が広く用いられているが、Web ブラウザのソフトの違いによる表示画面の差異が無視できないようになってくるであろうし、サーバ側の負担が大きいため、クライアントでも少しはプログラムを実行させることが再び増えてくるとと思われる。少し以前に、クライアント上で Java プログラムを実行させる Applet は好まれなかったが、再び Applet に代わるものが現れ、携帯電話や家電製品などさまざまなクライアントの能力に応じた表示方法を実現できる言語が現れると思う。そのため、HTML の言語の限界が見えてきて、また変わった言語が出てくるであろう。

近い将来では、処理能力の高いクライアントには、プウラグインソフトがインストールされることが増えて、サーバとクライアントの区別も薄れるかもしれない。事実、24 時間接続のブロードバンドネットワークが普及して広まっている P2P (ピアツーピア) アプリケーションは、各端末がすべてサーバのような働きをしている。

このような世の中の流れが掴め、かつ、新しい技術に対しても独力で取り組んでいける人材を教育学部の情報教育課程で養成できれば良いであろう。

## 参考文献

- 1) 山守一徳：分散管理方式による DNS 登録機能付 IP アドレス管理システム、2003 年情

- 報学シンポジウム、01-02、pp. 9-12 (Jan 2003) 製、<http://www.netjapan.co.jp/P-powerquest/di/V60/info/di60.html> (2003 年 1 月現在)
- 2) Drive Image 2002 日本語版 : Power Quest