

# 年 報

第 1 号

2004年2月

香川大学総合情報基盤センター



# 目 次

## 解説

MRTGによるネットワーク監視 .....	中村邦彦 ...	1
香川大学における接続コンピュータの現状 .....	古川善吾,三谷祐平 ...	16
香川大学におけるウイルス対策メールシステム .....	最所圭三 ...	23
X3Dについて - Web3Dは復権するか? - .....	香川考司 ...	33
JavaによるWebアプリ開発の勧め( ) - TomcatとCocoon2利用環境の構築 - .....	青木昌三 ...	45
JavaによるWebアプリ開発の勧め( ) - Cocoon2利用によるWebページ作成 - .....	青木昌三 ...	66
PC(パーソナルコンピュータ)を自分で組み立ててみよう .....	黒田 勉 ...	91
ウィルスチェックサーバによるアラートメールの集計 .....	青木昌三,瀬野芳孝,曾根計俊,高橋多恵 ...	121
事業報告(旧香川医科大学情報メディアセンター)		
旧香川医科大学情報メディアセンター事業報告 .....		131
学内ネットワーク基盤整備事業 .....	河内一芳 ...	132
情報リテラシ教育設備の更新と導入 .....	真鍋芳樹,立川敏明,寺中正人 ...	133
マルチメディア教育支援事業 .....	斉藤純代 ...	134
学生生活支援事業 .....	金谷尚明 ...	135
チュートリアル教育支援事業 .....	永井雅巳 ...	136
学内動画配信システムの整備 .....	岡田宏基 ...	137
国家試験合格支援設備の整備 .....	木村正司 ...	138
実験実習機器センター機器予約システム .....	永井由紀子 ...	139
学術データ発表、保全支援システム .....	西 望,宮中 宏,多田達史 ...	140
生体分子の立体構造に関する教育・研究支援事業 ...	徳光 浩,松下 治,西 望 ...	141
機器センターにおける「体験教室」の実施及び、研究効率の向上に向けてのサーバー化 .....	夕部幸一,漆原正英 ...	144
放射線教育支援システム .....	須田博文 ...	145
バーチャルPCR実習設備の整備 .....	佐藤忠文 ...	146
院内感染防止対策支援事業 .....	樋口和子 ...	147

地域医療情報ネットワークの形成を促進するための研究開発 .....	原 量宏 ...	148
旧香川医科大学情報メディアセンター支援事業機器一覧 .....		149
報告（旧香川大学情報処理センター）		
情報処理センター日誌 .....		153
利用申請一覧 .....		156
センター利用状況 .....		161
計算機利用統計 .....		163
資料		
香川大学総合情報基盤センター規則 .....		169
香川大学総合情報基盤センター運営委員会規則 .....		171
編集後記 .....		173

# MRTGによるネットワーク監視

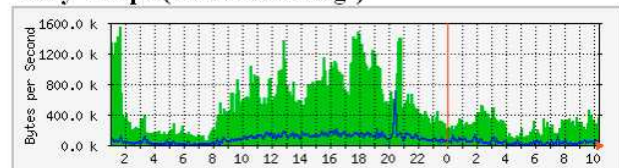
経済学部 中村邦彦  
nakamura@ec.kagawa-u.ac.jp

## 1 はじめに

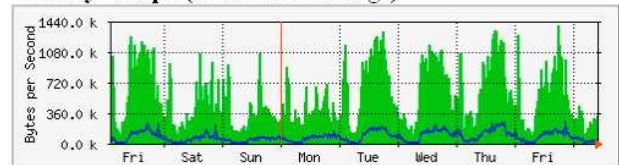
MRTGというのは、Multi Router Traffic Grapherのアクロニムである。その名のとおり、ネットワーク上のルーターに出入りするデータ量を収集し、それをグラフにして表示するプログラムである。筆者はこれを1998年に導入し、以来、大学対外ルーターのトラフィックの監視に使用している。どういうグラフかというのは見ていただく方がよいので、図1にそれを示す。ただし、本当はこれらのグラフはウェブブラウザの画面に、いくつかの数値と共に表示されるのであるが、紙面の都合で簡略化している。本物は<http://www.kagawa-u.ac.jp/mrtg/>で見ることができる。ひとつのグラフには、ルーターに入ってくるデータと出て行くデータが5分ごとに計測されて表示されている。上から順に5分間の平均、30分間の平均、2時間間の平均、1日の平均のグラフになっている。1日平均のグラフには14ヵ月分が表示されており、これを見れば、ときどき問題は発生したものの、ほぼ順調に通信量が増加していることが分かる。その後筆者はMRTGをネットワークトラフィックだけではなく、他の事象の監視にも利用してきた。

SINET側 ATM回線のトラフィック

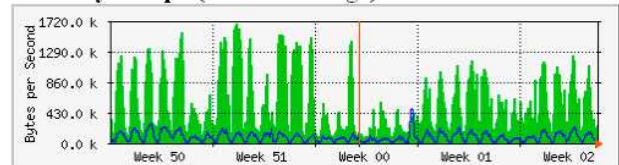
'Daily' Graph (5 Minute Average)



'Weekly' Graph (30 Minute Average)



'Monthly' Graph (2 Hour Average)



'Yearly' Graph (1 Day Average)

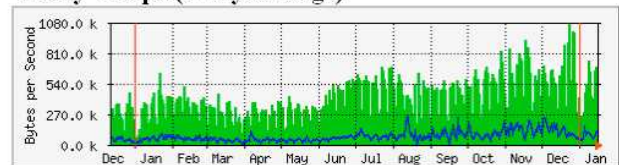


図1 MRTGのグラフ

昨年末(2003年)に、学内から学外に向けて、何度かIPスプーフィングによるDoS攻撃があった。送信元のIPアドレスが偽りであるから、そのパケットを捕まえて見ただけでは、それが学内のどこから来ているのかわからない。それで学内のあらゆるルーターやハブのトラフィックを監視した方がよさうということになった。監視はMRTGによるのが簡単でわかりやすいということから、ここに再度MRTGについての紹介記事を書くことにした。最近のLinuxの各ディストリビューションでは、MRTGのインストールは容易になっており、トラフィック監視の設定はごく短時間でできる。各部署のネットワーク管理者には是非とも監視の設定を行っていただきたい。

なお、本稿では、第2節でLinuxサーバーへのMRTGのインストール、第3節でルーター・ハブのトラフィック監視、第4節でLinuxサーバーへのSNMPエージェントの導入とサーバー監視、第5節で独自の監視用スクリプトの書き方をそれぞれ解説した。

## 2 MRTG のインストール

MRTG は多くの UNIX 系 OS だけでなく、Window NT 系の OS にも対応しているので、インストール作業は、よほど特殊なシステムでもなければ難しいことはない。最近の Linux ディストリビューションは、標準で MRTG のバイナリーパッケージが用意されており、インストールコマンドを実行すればすぐにも MRTG が使えるようになる。パッケージを利用できない場合は、ソースプログラムからコンパイルしなければならないが、基本的に必要なものは、MRTG それ自身の他、ウェブサーバー、Perl とグラフィックイメージを作成するための GD というライブラリーである。Perl と C コンパイラーが使える環境があればそれほど難しいことではないであろう。

### 2.1 Debian の場合

まず、筆者が愛用している Debian GNU/Linux の場合について説明する。もし、Apache がまだインストールされていないければ次のコマンドでインストールする。

```
# apt-get install apache
```

ここで行頭の#は、利用者がスーパーユーザーであることを示すプロンプト記号である。何項目か問合せがあるが、すべてデフォルト設定でかまわない。これで、ウェブブラウザから <http://サーバー/> に接続すれば標準のトップページが表示されるはずである。Apache の設定については他の文献がたくさんあるので、細かいことは省略する。

次に MRTG をインストールする。これも次のコマンドでインストールできる。

```
# apt-get install mrtg
```

これで MRTG とともに、そのシステムに必要な libgd1、libsnmp-session-perl などが、もしその時点でインストールされていないければ、自動的にインストールされる。いくつか問合せがあるが、すべてデフォルト設定でかまわない。ひとつだけ注意しておく、Debian の MRTG は、ウェブサーバーのドキュメントルートが /var/www であると仮定して、MRTG の監視データを/var/www/mrtg 以下に作成しようとするということである。Apache の設定を変更している場合は、手直しが必要になる。

Debian の安定版 Woody で用意されているソフトウェアのバージョンは、Apache が 1.3.26-0woody3、MRTG が 2.9.17-4 となっており、やや古いだが、実用上は問題がない。

### 2.2 Red Hat 系の場合

Red Hat 系の最新ディストリビューションは Fedora Core 1 になっているので、この場合について簡単に紹介しておく。Red Hat 9 までは、Red Hat Network を利用してパッケージの更新をするには、いちいちユーザー登録をしてから up2date を使わなければなかったが、現在はユーザー登録なしで、Debian と同様の apt を利用することが可能になっている。

まず apt の rpm パッケージを Fedora Project のウェブサイト<sup>1</sup> からダウンロードする。ファイル名は apt-0.5.15cnc1-0.fdr.3.1.i386.rpm である。スーパーユーザーになってそれをインストールする。

```
# rpm -ivh apt-0.5.15cnc1-0.fdr.3.1.i386.rpm
```

---

<sup>1</sup> [http://linux2ch.bbzone.net/index.php?FaqDistributionFedora#content\\_1\\_8](http://linux2ch.bbzone.net/index.php?FaqDistributionFedora#content_1_8)

ついで、パッケージ情報を最新のものに更新し、更にインストールされているパッケージをアップグレードする。

```
# apt-get update
# apt-get upgrade
```

これで新しいパッケージをインストールする準備ができた。

もし、Apache がインストールされていない場合は、次のようにしてインストールする。Red Hat では Apache のパッケージ名は httpd である。

```
# apt-get install httpd
```

MRTG のインストールも同様で、次のようにする。

```
# apt-get install mrtg
```

Fedora Core 1では、Apache は httpd-2.0.47-10、MRTG は mrtg-2.10.5-1になっている。Apache のドキュメントルートは/var/www/html に設定されている。その中を見てもファイルが何も無いが、ブラウザでアクセスすれば Fedora Core Test Page が表示されるようになっている。MRTG のファイルはドキュメントルートが/var/www/html であるにもかかわらず、/var/www/mrtg に作成されることになっているので気になったが、/etc/httpd/conf.d/mrtg.conf で次のように設定されているので問題はなかった。ただし、そのままでは localhost 以外の接続を拒否するようになっていたので、すべてのアクセスを許可するように修正した。

```
Alias /mrtg /var/www/mrtg

<Location /mrtg>
    Order deny,allow
    # Deny from all
    # Allow from localhost
    # Allow from .example.com
    Allow from all
</Location>
```

Red Hat の古いバージョンでも Freshrpms.net<sup>2</sup>に各バージョンに対応した apt パッケージが用意されているので、apt を使うことはできる。apt が使えないときは、必要なパッケージをダウンロードしてから rpm コマンドでインストールする。

### 3 ルーター・ハブのトラフィック監視

MRTG は SNMP(Simple Network Management Protocol)により、トラフィック情報を収集する。したがって、監視対象のルーター・ハブは、SNMP エージェント機能を持っているものでなければならない。ルーターがその機能を持っているのは当然であるが、安価なスイッチングハブは、通常は対応していない。幸い、旧情報処理センターで一括導入したハブはすべて SNMP エージェント機能を持っている。この機能を持っているルーター・ハブは IP アドレスとコミュニティ名が定義されているはずなので、

---

<sup>2</sup> <http://freshrpms.net/>

この情報が必要になる。わからない場合はそれぞれの管理者に問い合わせで欲しい。

MRTG の設定ファイルは、初期状態では/etc/mrtg.conf になっているが、これから複数のファイルを作成することになるので、新しいディレクトリ/etc/mrtg を作って、この中に置くことにする。

```
# mkdir /etc/mrtg
# cd /etc/mrtg
```

MRTGの設定ファイルは、cfgmakerというコマンドで自動的に作成できる。IPアドレスがip\_address、コミュニティ名が community\_name のルーター・ハブに対する設定ファイルを作成するには、次のようにする。

```
# cfgmaker community_name@ip_address >mrtg.cfg
```

これで特にエラーメッセージがなければ設定ファイルが作成されているので、次のようにして MRTG を起動してみよう。

```
# mrtg /etc/mrtg/mrtg.cfg
```

最初は更新するためのファイルが存在しないので、ウォーニングが表示されるが、それは問題がない。3回目からは出なくなる。これで /var/www/mrtg に多数のファイルが作成されているはずである。その中の HTML ファイルをブラウザで見れば表示できるが、直接ファイル名を指定して表示するのは面倒なので、それらを一覧表示するウェブページを作成する。そのためのツールが用意されているので、/var/www/mrtg に移動してからそのコマンドを入力する。

```
# cd /var/www/mrtg
# indexmaker /etc/mrtg/mrtg.cfg >index.html
```

これで、URL を http://サーバー名/mrtg/ として接続すれば、図2のように表示される。ひとつのグラフがルーターのひとつのポートに対応している。ただし、最初は表示するデータがないので、グラフの中身は何もないはずである。



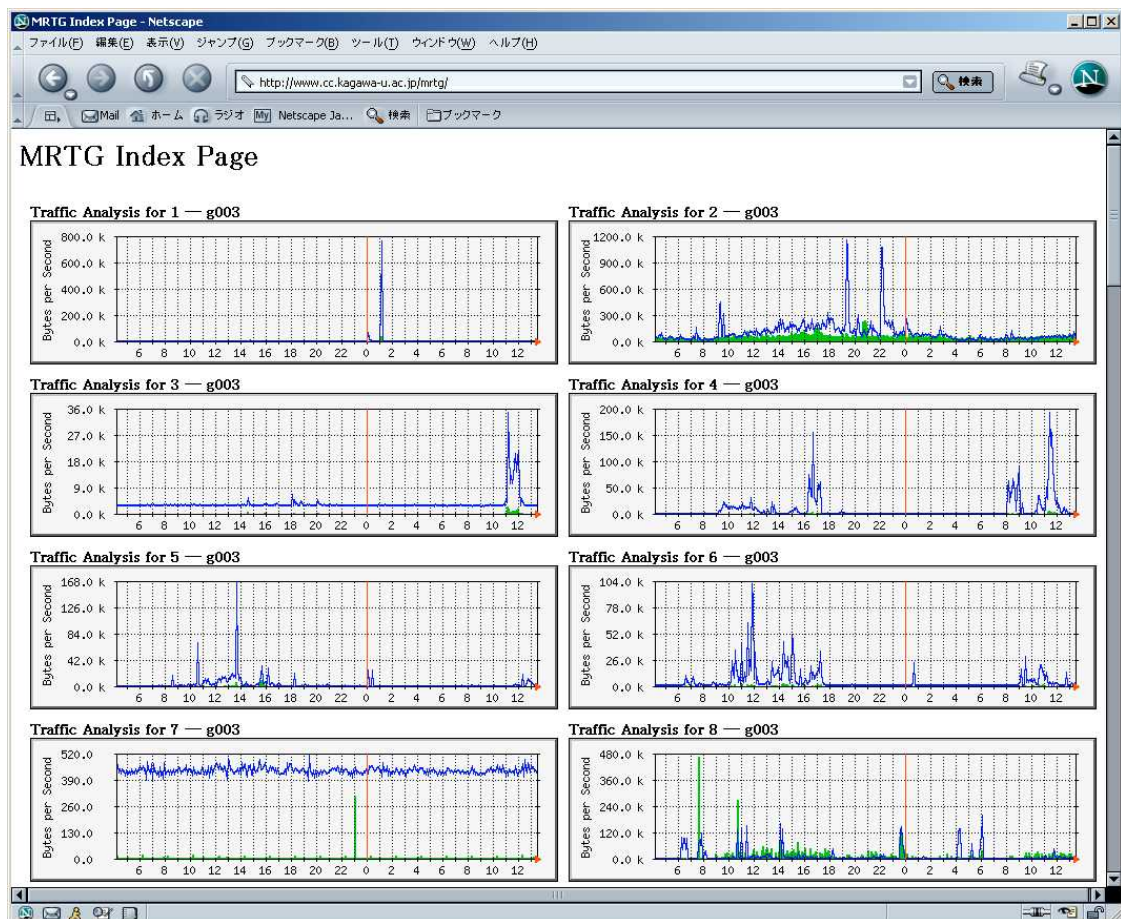


図2 MRTG のインデックスページ

図2ではグラフが縦2列に表示されているが、画面幅が狭いときはこれを1列にしたいとなるであろう。そのときは次のようにすればよい。

```
# indexmaker --columns=1 /etc/mrtg/mrtg.cfg >index.html
```

### cron による自動起動の設定

MRTG は cron により、5分ごとに起動される。その設定は /etc/cron.d/mrtg にある。

```
0-55/5 * * * * root if [ -x /usr/bin/mrtg ] && [ -r /etc/mrtg.cfg ]; then
/usr/bin/mrtg /etc/mrtg.cfg >> /var/log/mrtg/mrtg.log 2>&1; fi
```

先に、mrtg.cfg の場所を変更したので、これを次のように修正する。

```
0-55/5 * * * * root if [ -x /usr/bin/mrtg ] && [ -r /etc/mrtg/mrtg.cfg ];
then /usr/bin/mrtg /etc/mrtg/mrtg.cfg >> /var/log/mrtg/mrtg.log 2>&1;
fi
```

if 文があつて複雑に見えるが、if 文を消去して次のようにしてもよい。

```
0-55/5 * * * * root /usr/bin/mrtg /etc/mrtg/mrtg.cfg >>/var/log/mrtg/mrtg.log 2>&1
```

## 表示設定の変更

MRTG で自動作成された設定ファイルでは、トラフィック量のグラフはバイト/秒で表示され、グラフの時間軸が左向きになっている。これを変更したいと思う人が多いと思うので、簡単な方法を紹介する。

mrtg.cfg の先頭付近に次のような行がある。この行頭の#を消去することにより、すべてのグラフの単位がビット/秒に、時間軸が右向きに変更される。

```
#Options[_]: growright, bits
```

## 多数のルーター・ハブを監視する場合

監視したいルーターやハブがたくさんある場合は、全部をひとつの mrtg.cfg にまとめてもよいが、include 文を使う方がよいであろう。つまり、個々のルーター・ハブについての設定ファイルを cfmaker で作る際に、router1.cfg、router2.cfg、・・・のように設定ファイルを作成しておき、それを、mrtg.cfg に include するようにする。その例を示す。

```
### Global Config Options
WorkDir: /var/www/mrtg
### Global Defaults
# to get bits instead of bytes and graphs growing to the right
Options[_]: growright, bits

include: /etc/mrtg/router1.cfg
include: /etc/mrtg/router2.cfg
include: /etc/mrtg/router3.cfg
include: /etc/mrtg/router4.cfg
.....
```

以下省略

この場合、includeされる設定ファイルの Global 設定部分は念のためコメントアウトしておく。

さて、こうして多数のポートを監視した場合に、マシンの負荷がどの程度になるか気になるであろう。筆者のマシンの例であるが、CPU が PentiumIII 450MHz x 2で 120ポートを監視させたとき、処理に20秒程度かかっている。この程度であれば問題はないであろう。

## 4 サーバーの監視

### 4.1 NET-SNMP のインストール

UNIX 系のサーバーであれば SNMP エージェント機能を持つソフトウェアをインストールすることにより、ネットワークのトラフィックだけでなく、様々な物理量を MRTG で監視することができる。主要な Linux のディストリビューションには NET-SNMP パッケージが含まれている。これは UNIX 系の OS で SNMP(Simple Network Management Protocol) サービスを利用するためのソフトウェアで、以前は、UCD-SNMP と呼ばれていたものである。

Debian の場合は、複数のパッケージに分割されているが、snmp と snmpd をインストールすれば、それに必要なパッケージもインストールされる。Debian の NET-SNMP のバージョンは4.2.3-2であるが、本家 The NET-SNMP Project の最新バージョンは5.1になっている。

```
# apt-get install snmpd snmp
```

Debian ではこれで snmpd と snmptrapd が自動的に起動するように設定されるが、ここでは snmptrapd を使う予定がないので、これを止めておく。これは/etc/default/snmpd の中で、

```
TRAPDRUN=yes
```

となっているところを、no に変更する。

```
TRAPDRUN=no
```

SNMP エージェントの設定ファイルは、/etc/snmp/snmpd.conf である。Debian の場合は、少し修正するだけですぐに使えるような snmpd.conf が用意されているが、ここでは Fedora Core1 でも使えるような最小限の設定ファイルを作成した。

/etc/snmp/snmpd.conf の内容

```
com2sec readonly localhost public
com2sec readonly 監視サーバー public
group MyROGroup v1 readonly
view all included .1 80
access MyROGroup "" any noauth exact all none none
disk /
disk /var
load
```

ここで、監視サーバーのところは、MRTG をインストールしている監視サーバー名か、その IP アドレスを書く。このエージェントは localhost とその監視サーバーからの問合せにだけ応答するようになる。最後に disk と load という行があるが、disk はディスクパーティションの使用量を監視するため、load は CPU のロードアベレージを監視するためである。snmpd.conf の説明をすると長くなるので、とりあえずこれだけにしておく。詳しくは、参考文献を見て欲しい。

これで snmpd を再起動してから、次のように snmpwalk で system 情報を表示してみる。

```
# snmpwalk localhost public system
```

ずらずらと system 情報が表示されたら snmp エージェントが動作していると考えてよいであろう。

Red Hat 後継の Fedora Core1 では、net-snmp と net-snmp-utils になっている。バージョンは 5.1-2.1 である。

インストールは次のコマンドで行う。

```
# apt-get install net-snmp net-snmp-utils
```

自動起動スクリプトは snmpd と snmptrapd に分かれているので、snmpd の方だけを設定しておく。設定ファイルはあらかじめ用意されているものがあるが、今回は debian と同じものを使った。設定ができたなら snmpd を起動する。snmpwalk はコマンドの書き方が変わったようで、system オブジェクトを表示

させるには次のようにする。

```
# snmpwalk -v 1 -c public localhost system
```

#### 4.2 ネットワークカードの入出力の監視

そこで、このサーバーのネットワークトラフィックを MRTG に監視させるように、MRTG の設定ファイルを作成する。と言っても、このサーバーはすでに SNMP ですべてのネットワークカードを流れるトラフィックを読み出すことができるようになっているので、ルーターを監視するのと同じことである。

```
# cfgmaker public@サーバー名 >server1.cfg
```

とすれば、ルーターのときと同様にトラフィック管理のための設定ファイルが作成できる。

#### 4.3 ロードアベレージの監視

システムのロードアベレージを知るには、通常は uptime などのコマンドを使う。

```
$ uptime
22:28:17 up 45 days, 10:41, 1 user, load average: 0.02, 0.06, 0.07
```

load average として3つの値が表示されるが、順に1分平均、5分平均、15分平均である。SNMP で得られる情報を確認するには次のようにする。

```
$ snmpwalk localhost public .1.3.6.1.4.1.2021.10.1
enterprises.ucdavis.laTable.laEntry.laIndex.1 = 1
enterprises.ucdavis.laTable.laEntry.laIndex.2 = 2
enterprises.ucdavis.laTable.laEntry.laIndex.3 = 3
enterprises.ucdavis.laTable.laEntry.laNames.1 = Load-1
enterprises.ucdavis.laTable.laEntry.laNames.2 = Load-5
enterprises.ucdavis.laTable.laEntry.laNames.3 = Load-15
enterprises.ucdavis.laTable.laEntry.laLoad.1 = 0.26
enterprises.ucdavis.laTable.laEntry.laLoad.2 = 0.14
enterprises.ucdavis.laTable.laEntry.laLoad.3 = 0.10
enterprises.ucdavis.laTable.laEntry.laConfig.1 = 12.00
enterprises.ucdavis.laTable.laEntry.laConfig.2 = 12.00
enterprises.ucdavis.laTable.laEntry.laConfig.3 = 12.00
enterprises.ucdavis.laTable.laEntry.laLoadInt.1 = 26
enterprises.ucdavis.laTable.laEntry.laLoadInt.2 = 14
enterprises.ucdavis.laTable.laEntry.laLoadInt.3 = 10
enterprises.ucdavis.laTable.laEntry.laLoadFloat.1 = Opaque: Float:
0.260000
enterprises.ucdavis.laTable.laEntry.laLoadFloat.2 = Opaque: Float:
0.140000
enterprises.ucdavis.laTable.laEntry.laLoadFloat.3 = Opaque: Float:
0.100000
```

3つのロードアベレージがいくつかの形式で得られることがわかる。MRTG は整数値しか取り扱えないので、laLoadInt.1と laLoadInt.2を使うことにする。今度はMRTGの設定ファイルを自動的に作成することはできないので、次のような記述をエディターで追加する。

```
Target[Is_Load]:1.3.6.1.4.1.2021.10.1.5.2&1.3.6.1.4.1.2021.10.1.5.1:
```

```

public@localhost
MaxBytes[Is_Load]: 1000
Title[Is_Load]: Is Load Average
PageTop[Is_Load]: <H1>Is Load Average</H1>
Options[Is_Load]: gauge, nopercent, growright
YLegend[Is_Load]: Load Average(%)
ShortLegend[Is_Load]: &nbsp;
LegendI[Is_Load]: 5min av:&nbsp;
LegendO[Is_Load]: 1min av:&nbsp;
WithPeak[Is_Load]: ymw

```

MRTG の通常のグラフは、表示期間が長くなるに伴って平均する間隔も長くなる。監視する立場から考えると、平均値よりはピーク値の方が重要なことがある。それを表示させる設定が、上記設定ファイルの最後の行、“WithPeak[Is\_Load]: ymw” である。ymw という設定は、yearly(1日平均)、monthly(2時間平均)、weekly(30分平均)のグラフにおいて、5分間平均のピーク値を表示することを指示している。このケースでは1分間平均を表示させているので、測定間隔を1分にして、WithPeakの値を“ymwd”にすれば、5分平均のグラフにも1分平均のピークが表示されるかもしれないと考えたが、それはできない。MRTG は測定の最小間隔を5分と想定しているの、たとえ1分ごとに測定しても、その値は5分間で平均された値が表示される。WithPeakにdを追加しても無視されるだけである。一方、MRTG はピーク値を記録しており、1分間隔で測定した場合にはそのピーク値が記録されていた。それなら、5分平均グラフにもピーク値を表示させられないかと考えてソースプログラムを見たところ、次のような部分が見つかった。

```

if ($$rcfg{'suppress'}{$$router} !~ /d/) {
    # VMS: should work for both now
    push (@exec, "i", "$$cfg{'imagedir'}$$rcfg{'directory'}{$$router}$$router}-day.
    ${main::GRAPHFMT}", $sign*$maxvi, $sign*$maxvo, $maxx, $maxy, , $xscale, $yscale,
    $growright, $step, $bits, $ytics, $yticsf);
    @mxvls = ("d");
    push (@metas, "$$cfg{'imagedir'}$$rcfg{'directory'}{$$router}$$router}-day.
    ${main::GRAPHFMT}",

```

この3行目を次のように修正すれば、5分間グラフでもピーク値を表示させることができる。

```

$rop = ($$rcfg{'withpeak'}{$$router} == /d/) ? "p" : "i";
push (@exec, $rop, "$$cfg{'imagedir'}$$rcfg{'directory'}{$$router}$$router}-day.

```

その変更により表示されたのが図3である。グラフの一番上の線（実際の色はマゼンタ）が1分間平均のピーク値で、修正前ではピーク値が150%程度になっていたが、実は210%が観測されていたことがわかる。

### ~Daily~ Graph (5 Minute Average)

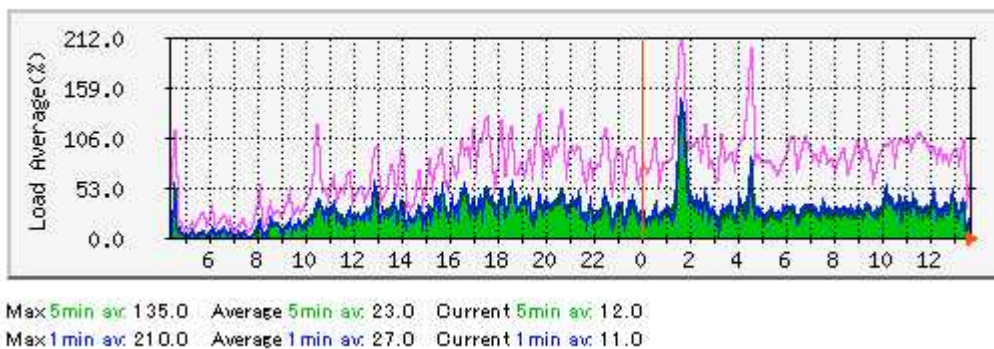


図3 Load Average 5平均のグラフにピーク値を表示させた

## 4.4 メモリー使用量の監視

メモリーに関する情報を snmpwalk で表示させてみる。

```
$ snmpwalk localhost public .1.3.6.1.4.1.2021.4
enterprises.ucdavis.memory.memIndex.0 = 0
enterprises.ucdavis.memory.memErrorName.0 = swap
enterprises.ucdavis.memory.memTotalSwap.0 = 979956
enterprises.ucdavis.memory.memAvailSwap.0 = 971068
enterprises.ucdavis.memory.memTotalReal.0 = 515964
enterprises.ucdavis.memory.memAvailReal.0 = 25436
enterprises.ucdavis.memory.memTotalFree.0 = 996504
enterprises.ucdavis.memory.memMinimumSwap.0 = 16000
enterprises.ucdavis.memory.memShared.0 = 0
enterprises.ucdavis.memory.memBuffer.0 = 49196
enterprises.ucdavis.memory.memCached.0 = 215716
enterprises.ucdavis.memory.memSwapError.0 = 0
enterprises.ucdavis.memory.memSwapErrorMsg.0 =
```

これらの値をそのままグラフにしてもあまり面白くない。例えばスワップの使用量を表示したければ、memTotalSwap から memAvailSwap を差し引かなければならない。MRTG に加減算をさせるときは、必ずペア同士で演算しなければならないので面倒である。また実メモリーの使用量として、memTotalReal から memAvailReal を差し引いたものを表示してもあまり面白くない。なぜなら、Linux は実メモリーの空きがあると、それを入出力バッファとディスクキャッシュに割り当ててしまうので、実メモリーの使用量を監視していても常にほとんどが使用されているように見えてしまうのである。バッファとキャッシュに使用されている分を差し引いたメモリー使用量を知るには、free というコマンドを使うとよい。

```
$ free
              total          used          free       shared    buffers     cached
Mem:          515964      490024      25940           0        32264     213264
-/+ buffers/cache: 244496      271468
Swap:         979956           9596      970360
```

この例では、実メモリ516MB のうち、使用されているのは490MB であるが、バッファとキャッシュ

分を差し引いた使用量は244MB であることがわかる。このような量を MRTG で計算させるのはますます面倒であるので、これは外部プログラムを使って行うことにした。そのやり方は次節で紹介する。

#### 4.5 ディスク使用量の監視

ディスクの使用量に関する情報を確認してみる。

```
$ snmpwalk localhost public .1.3.6.1.4.1.2021.9.1.8
enterprises.ucdavis.dskTable.dskEntry.dskUsed.1 = 237300
enterprises.ucdavis.dskTable.dskEntry.dskUsed.2 = 1766596
enterprises.ucdavis.dskTable.dskEntry.dskUsed.3 = 37525484
nakamura@is:/var/tmp$ snmpwalk localhost public .1.3.6.1.4.1.2021.9.1
enterprises.ucdavis.dskTable.dskEntry.dskIndex.1 = 1
enterprises.ucdavis.dskTable.dskEntry.dskIndex.2 = 2
enterprises.ucdavis.dskTable.dskEntry.dskIndex.3 = 3
enterprises.ucdavis.dskTable.dskEntry.dskPath.1 = /
enterprises.ucdavis.dskTable.dskEntry.dskPath.2 = /var
enterprises.ucdavis.dskTable.dskEntry.dskPath.3 = /home
enterprises.ucdavis.dskTable.dskEntry.dskDevice.1 = /dev/md0
enterprises.ucdavis.dskTable.dskEntry.dskDevice.2 = /dev/md2
enterprises.ucdavis.dskTable.dskEntry.dskDevice.3 = /dev/md3
enterprises.ucdavis.dskTable.dskEntry.dskMinimum.1 = 100000
enterprises.ucdavis.dskTable.dskEntry.dskMinimum.2 = 100000
enterprises.ucdavis.dskTable.dskEntry.dskMinimum.3 = 100000
enterprises.ucdavis.dskTable.dskEntry.dskMinPercent.1 = -1
enterprises.ucdavis.dskTable.dskEntry.dskMinPercent.2 = -1
enterprises.ucdavis.dskTable.dskEntry.dskMinPercent.3 = -1
enterprises.ucdavis.dskTable.dskEntry.dskTotal.1 = 1922188
enterprises.ucdavis.dskTable.dskEntry.dskTotal.2 = 19222656
enterprises.ucdavis.dskTable.dskEntry.dskTotal.3 = 87407304
enterprises.ucdavis.dskTable.dskEntry.dskAvail.1 = 1587244
enterprises.ucdavis.dskTable.dskEntry.dskAvail.2 = 16479516
enterprises.ucdavis.dskTable.dskEntry.dskAvail.3 = 45441740
enterprises.ucdavis.dskTable.dskEntry.dskUsed.1 = 237300
enterprises.ucdavis.dskTable.dskEntry.dskUsed.2 = 1766596
enterprises.ucdavis.dskTable.dskEntry.dskUsed.3 = 37525484
enterprises.ucdavis.dskTable.dskEntry.dskPercent.1 = 13
enterprises.ucdavis.dskTable.dskEntry.dskPercent.2 = 10
enterprises.ucdavis.dskTable.dskEntry.dskPercent.3 = 45
enterprises.ucdavis.dskTable.dskEntry.dskPercentNode.1 = 3
enterprises.ucdavis.dskTable.dskEntry.dskPercentNode.2 = 3
enterprises.ucdavis.dskTable.dskEntry.dskPercentNode.3 = 1
enterprises.ucdavis.dskTable.dskEntry.dskErrorFlag.1 = 0
enterprises.ucdavis.dskTable.dskEntry.dskErrorFlag.2 = 0
enterprises.ucdavis.dskTable.dskEntry.dskErrorFlag.3 = 0
enterprises.ucdavis.dskTable.dskEntry.dskErrorMsg.1 =
enterprises.ucdavis.dskTable.dskEntry.dskErrorMsg.2 =
enterprises.ucdavis.dskTable.dskEntry.dskErrorMsg.3 =
```

ディスクの使用量はパーティションごとに取り出すことができる。バイト数で表示してもよいがここでは簡単に利用率を使うことにする。MRTG はグラフを作成するだけでなく、各値について許容範囲を設定し、その範囲から外れた場合に指定したプログラムを起動することができる。そこで、ディスクの使用量が設定値を超えた場合、Eメールで通知するように設定してみることにする。

Global セクションに ThreshDir を追加する。

```
ThreshDir: /var/lib/mrtg/thresh
```

これがないと、観測値が設定範囲を超えているときは、指定プログラムが毎回起動される。これを追加することにより状態が記録されるようになり、プログラムの起動は、観測値が範囲内から範囲外に変わったときだけ起動されるようになる。

Target 部分の記述を次のように設定する。もちろん、これ以外のものは必要に応じて設定する。

```
Target[DiskVar]:1.3.6.1.4.1.2021.9.1.9.2&1.3.6.1.4.1.2021.9.1.9.2:public@localhost
```

```
ThreshMaxI[DiskVar]: 80
```

```
ThreshProgI[DiskVar]: /etc/mrtg/warnmail.pl
```

1.3.6.1.4.1.2021.9.1.9.2 は enterprises.ucdavis.dskTable.dskEntry.dskPercent.2 を表す。ここではトラフィックの IN と OUT にあたる2つの量に同じ量を割り付けている。

ThreshMaxIを80に設定したので、利用量が80%を超えたときにThreshProgIで指定したプログラムが起動される。以下は warnmail.pl のリストである。

```
#!/usr/bin/perl -w

$sendmail = '/usr/sbin/sendmail';
$from = "From: postmaster¥@is.ec.kagawa-u.ac.jp¥n";
$to = "To: nakamura¥@is.ec.kagawa-u.ac.jp¥n";
$subject = "Subject: MRTG Warning¥n¥n";

$timestamp = localtime();
$item = $ARGV[0];
$thresh = $ARGV[1];
$value = $ARGV[2];

$body = <<"END";
MRTG Warning Message:
$timestamp: $item $value exceeds $thresh.
END

open(LOG, ">>/var/log/mrtg/warning.log");
print LOG $body;
close(LOG);

$header = $from . $to . $subject;
open(MAIL, "| $sendmail -i -t");
print MAIL $header;
print MAIL $body;
close(MAIL);
exit 0;
```

MRTG はスクリプトに、項目名、閾値、項目の値を引数として渡すので、メールにそれらを埋め込んでいる。メールの送信は標準的な sendmail インターフェイスを使った。



## 5 外部監視スクリプトの作成

SNMP では望む監視データが得られないときは、利用者自身がそのプログラムを用意する方法がある。これを外部監視スクリプトと呼ぶことにする。それは MRTG の設定ファイルで、Target の指定を次のように書く。

```
Target[識別名]: `監視スクリプト`
```

このスクリプトは標準出力に次の4行を出力するように作成する。通常は3,4行目には0を出力しておけばよい。

1. 行目 最初の値の現在の状況。通常は「入力バイトのカウント」
2. 行目 2番目の値の現在の状況。通常は「出力バイトのカウント」
3. 行目 監視対象の稼働時間を示すstring (人間が読めるフォーマットで)。
4. 行目 監視対象の名前を示すstring

例として、SNMP エージェントから得られるメモリー情報から、MRTG で表示させたい値を計算し、MRTG に渡す Perl スクリプトを紹介する。Perl から SNMP エージェントにアクセスするために、Net::SNMP という Perl モジュールを使用している。

```
#!/usr/bin/perl -w
use strict;
use vars qw($session $error $response);
use Net::SNMP;
my $oid_memTotalSwap = '1.3.6.1.4.1.2021.4.3.0';
my $oid_memAvailSwap = '1.3.6.1.4.1.2021.4.4.0';
my $oid_memTotalReal = '1.3.6.1.4.1.2021.4.5.0';
my $oid_memAvailReal = '1.3.6.1.4.1.2021.4.6.0';
my $oid_memBuffer = '1.3.6.1.4.1.2021.4.14.0';
my $oid_memCached = '1.3.6.1.4.1.2021.4.15.0';

($session, $error) = Net::SNMP->session(
    -hostname => shift || 'localhost',
    -community => shift || 'public',
    -port      => shift || 161
);

if (!defined($session)) {
    printf("ERROR: %s.\n", $error);
    exit 1;
}

my @oids = ($oid_memTotalSwap, $oid_memAvailSwap, $oid_memTotalReal,
            $oid_memAvailReal, $oid_memBuffer, $oid_memCached);

if (!defined($response = $session->get_request(@oids))) {
    printf("ERROR: %s.\n", $session->error());
    $session->close();
    exit 1;
}
```

```

my $memTotalReal = $response->{$oid_memTotalReal};
my $memAvailReal = $response->{$oid_memAvailReal};
my $memBuffer    = $response->{$oid_memBuffer};
my $memCached    = $response->{$oid_memCached};
my $memTotalSwap = $response->{$oid_memTotalSwap};
my $memAvailSwap = $response->{$oid_memAvailSwap};

$session->close();

#print "memTotalReal $memTotalReal¥n";
#print "memAvailReal $memAvailReal¥n";
#print "memBuffer $memBuffer¥n";
#print "memCached $memCached¥n";
#print "memTotalSwap $memTotalSwap¥n";
#print "memAvailSwap $memAvailSwap¥n";

my $memReal = $memTotalReal - ($memAvailReal + $memBuffer + $memCached);
my $memTotal = ($memTotalReal - $memAvailReal) + ($memTotalSwap -
$memAvailSwap);

print "$memReal¥n";
print "$memTotal¥n";
print "0¥n";
print "0¥n";
exit 0;

```

## 6 むすびに代えて

以上見てきたように、MRTG を使えば、ネットワークトラフィックの監視が簡単に行える。また、ネットワークトラフィック以外の時系列量もグラフにして表示することができるので、便利である。そうは言っても、MRTG はもともと、ルーターのトラフィックを監視する目的で作成されたフリーソフトウェアである。思わぬところで限界を感じることもある。以下思いついたものをあげておく。

- ・小数点付きのデータを扱えない

これはマシン室や機器内部の温度を表示させようとしたときに気が付いた。温度は摂氏で小数点以下第1位まで計測されているのだが、そのままでは MRTG には入力できなかった。現在は MRTG には温度を10倍した整数を与えて、表示するときに10分の1にすることができるので、表示できないことはないのだが、面倒である。

- ・入力した数値と異なる値が表示される

これはセンターPC の利用者数を表示させたときに気が付いた<sup>3</sup>。MRTG は与えた数値とは違う数値を表示することが多かった。計測時刻がちょうど5分の倍数になっていないときに発生する。多数の PC が稼動しているかどうかを調査するのは独立したプログラムで行っている。その結果を MRTG が読み出して表示するようにしている。稼動している PC の調査はちょうど5分の倍数の時刻に走らせているが、その調査には数十秒かかる。すべての PC が稼動していれば調査はあっという間に終わるのだが、応答しない PC についてはしばらく待たなければならないし、何度かリクエストを再送信もしてい

---

<sup>3</sup> <http://orphe.cc.kagawa-u.ac.jp/pcmon/pcstatus.html>

るので、どうしても時間がかかるのである。それで MRTG は調査するプログラムを起動してから1分後に起動している。つまり、MRTG は毎時1分、6分、11分、・・・に起動されている。ところが、MRTG のログファイルをみると、最終的に記録される時刻は0分、5分、10分、・・・になっている。その時刻の値は測定値から補間によって計算されているらしく、その際に値が変えられていた。。50人が49人になってもあまり問題はないが、1人が0人になったのでは大いに困る。この場合は測定時刻を強制的に書き換えて対応した。

以上、最後に問題点を指摘したが、それでも MRTG は大いに役に立つ優れたソフトウェアには違いない。開発者の方々、日本語訳をしてくださっている方々に深く感謝したい。

## 参考文献

- [1] MRTG のオリジナルサイト、<http://people.ee.ethz.ch/~oetiker/webtools/mrtg/>
- [2] MRTG の日本語のサイト、<http://www.mrtg.jp/doc/>
- [3] 中村邦彦、MRTG によるネットワークトラフィック調査、香川大学情報処理センター年報第7号、1999年2月
- [4] NET-SNMP の公式サイト、The NET-SNMP Project Home Page、<http://www.net-snmp.org/>
- [5] 富成章彦著、「オープンソースを使ったネットワーク監視術 By MRTG&SNMP」、セレンディップ、2001/8
- [6] マーク・A. ミラー著、トップスタジオ訳、「SNMPインターネットワーク管理 ネットワーク管理の標準プロトコルとツールを徹底解説」、翔泳社、1998/11
- [7] ダグラス・R. マウロ、ケビン・J. シュミット著、土本 康生、福田 剛士訳、「入門SNMP」、オライリー・ジャパン、2002/7

# 香川大学における接続コンピュータの現状

工学部信頼性情報システム工学科 古川 善吾  
工学研究科信頼性情報システム専攻 三谷 祐平

## 1 はじめに

香川大学における情報通信技術利用の現状の一端を把握するために接続されているコンピュータの接続台数を調査している。その概要および今後の方針について報告する。

## 2 コンピュータ台数

コンピュータの接続台数は、キャンパス情報ネットワーク（以下では、誤解のない場合には、単に「ネットワーク」）の利用者の全体像を示す指標の1つと考えられる。

パーソナルコンピュータの(以下「パソコン」)の進歩によってコンピュータを共同で利用する時代から、1人1台を経て、利用者が複数台のパソコンを必要に応じて使用する時代になってきている。この変化を促進した技術としては、動的にIPアドレスを割り当てるDHCP (dynamic host configuration protocol) の開発と携帯型パソコンの普及が挙げられる。複数台のパソコンを1人の利用者が状況や目的に応じて使い分けているために、コンピュータの接続台数が直接、利用者の人数を表しているわけではない。ただし、ある時点で、安全性を確認したり、異常な通信を行うコンピュータを見つけ出す対象の台数を表している。

接続されているコンピュータを調べる方法としては、使用される可能性のあるすべてのIPアドレスにパケットを送信して応答があるかどうかを調べるIPスキャンが考えられる。

コンピュータへのアドレスの割当方法には、利用者が明示的にアドレスを指定する「固定アドレス方式」と、コンピュータがDHCPプロトコルを用いて利用可能なアドレスを割り当てる「動的アドレス方式」とがある。固定アドレス方式では、同じIPアドレスを利用者が使用しないために何らかの管理をする必要があるため、接続されるコンピュータの台数を大まかに把握することが可能であった。しかしながら、動的割当方式を採用すると接続されるコンピュータ台数を把握することがますます困難になってしまう。

2000年4月に開設された工学部キャンパスでは、学生に携帯型パソコンを入学時に購入させることにしたために、開設当初から学部でDHCPサーバを立ち上げて動的アドレス方式を採用してきた。そのために、ネットワークに接続されているコンピュータの台数は、ほとんど把握できなかった。そこで、2002年にIPスキャンによって調査したところ、約400台から約800台の間で接続されているコンピュータの台数が大きく変化していることが判った[2]。

工学部での経験に基づいて2003年6月には、接続されているコンピュータ台数の調査を香川大学全学で行うようにした[1]。これは、香川大学全体でのコンピュータの利用状況を把握することが目的であるが、全学的にDHCPサーバを導入して、運用することが計画されており、その利用状況を把握できればと考えて調査している。その結果、約1,200台から約3,000

台の間で台数が変化していることが判った。

### 3 調査方法

#### 3.1 IP アドレス

香川大学では、クラス B のネットワークアドレス(133.92.0.0)を 24 ビットのサブネットアドレス(133.92.xxx.0)として部局等に割り付けている。使用しているサブネットについては総合情報基盤センター(以降「基盤センター」と記す)で管理している。なお、医学部は、クラス C のネットワークアドレスを使用しており、ここでは調査の対象から除いた。

サブネット内の利用方法は、各部局等に任せられている。例えば、工学部では、講義室や学科のサブネットでは、半分を固定アドレス方式のアドレス(以降「固定アドレス」と記す)として利用し、半分を動的アドレス方式のアドレス(以降「動的アドレス」と記す)として利用している。

#### 3.2 ICMP パケットの利用

接続されているコンピュータの調査には、基盤センターで使用が登録されているサブネットアドレスのすべてのアドレスに IP プロトコルとして規定されている ICMP(Internet Control Message Protocol) 応答要求パケットを送信し、応答があったアドレスを持つコンピュータが接続されているものと判断する。これは、コンピュータやコンピュータネットワークの不正利用のための対象となるコンピュータを検出する手段(IP スキャン)として紹介されている [3] 方法である。工学部での試行においては、全教官に IP スキャンについて了解を求めたところ、4 台のコンピュータについては、除外の申請があった。基盤センター(旧香川大学情報処理センター(以下「処理センター」と記す))に了解をえた。

この ICMP 応答要求パケットを送信して応答が返るか否かを判断するために ping(packet internet groper) コマンドを用いた。このコマンドは、IP アドレスを指定してそのアドレスのコンピュータへの到達可能性を調べるために一般的に用いられるコマンドである。以下のような特徴がある。

1. コンピュータネットワークの管理のために規定された IP レベルのプロトコルであり、IP で接続されているコンピュータは、ほとんどが応答を返す。
2. 個人用防火壁システム(Firewall)の中には、応答を返さないように設定できるソフトウェアがあり、完全に計数することはできない。
3. ルータなどのように複数の IP インターフェイスを持つコンピュータは、重複して計数される。

しかしながら、昨年 8 月(2003/8)にワーム「MS プラスター」およびその変種(以降「MS プラスター類」と記す)が旧香川大学のネットワークに侵入してしまった。MS プラスター類

は、まず IP スキャンによって感染したコンピュータから到達可能なコンピュータを検出した後に、自分自身を感染させることを試みるワームである。そのために、ICMP 応答要求パケットに応答を返さないよう個人用防火壁システムを設定しているところのワームの侵入を阻止することができ、急速に個人用防火壁システムが導入され、ICMP 応答要求パケットに対して応答を返さないコンピュータが増加した。

### 3.3 80 番ポートの確認

MS プラスター類の感染によって、ICMP パケットを利用した調査では、誤差が大きいのではないかと考えられた。そこで、他の調査方法として TCP(transmission control protocol) プロトコルで 80 番ポートでの通信の試みを行った。一般に、80 番ポートは、HTTP (hypertext transfer protocol) で用いられるポートである。最近、ルータなどのネットワーク機器もその設定に HTTP を用いることが多いので、より多くの IP アドレスについてその到達可能性を確認できるのではないかと考えられる。

### 3.4 図表示

接続されているコンピュータの台数を表示するために通信量を図表示するために開発されたツール MRTG (multi router traffic grapher) [6]を用いた。MRTG は、通信量の日変化(5分平均の変化)、週変化(30分)、月変化(2時間)、年変化(1日)を図表示するツールである。もともとは、SNMP (simple network management protocol) に基づいて通信量の変化を自動的に取り出して表示するツールである。

このツールで図表示するために、上で述べた調査方法によって得られたコンピュータの接続台数を整形して出力する。接続台数の調査間隔は、通信量の場合と同様に最小を5分とした。これは、工学部では、約1,100名の学生が携帯型パソコンを所有しており、コンピュータ台数の変動がかなりあること、図表示では変化が滑らかになる、などのためである。

最小の調査間隔を5分とするために、以下のような工夫を行った。ただし、実際に、使用したコマンドは、ICMP パケットを送信するアドレスをファイルで一括指定のできる fping コマンド[4]を用いた。

- 応答確認は、1つの ICMP パケットに対する応答の有無で判断する。
- 応答がないことの時間切れの判断を 100msec で行う。
- 異なったコンピュータへのパケットの送信間隔を 1sec とする。
- 調査するアドレス(全学で 130 サブネット、約 33,000 アドレス) を 4 つに分割し、並行して 4 つのプロセスで接続を確認する。

また、80 番ポートの確認には、Linux 版の nmap プログラム[5]を用いた。ただし、現在のところ、5分の調査間隔では接続確認が間に合わない場合がある。

#### 4 コンピュータの接続台数

接続台数の計測結果グラフを図1~4に示す。グラフは、2004年1月14日17時40分のデータである。昨年9月に計測用のコンピュータのディスクを交換したので、年間グラフは9月以降のデータだけである。

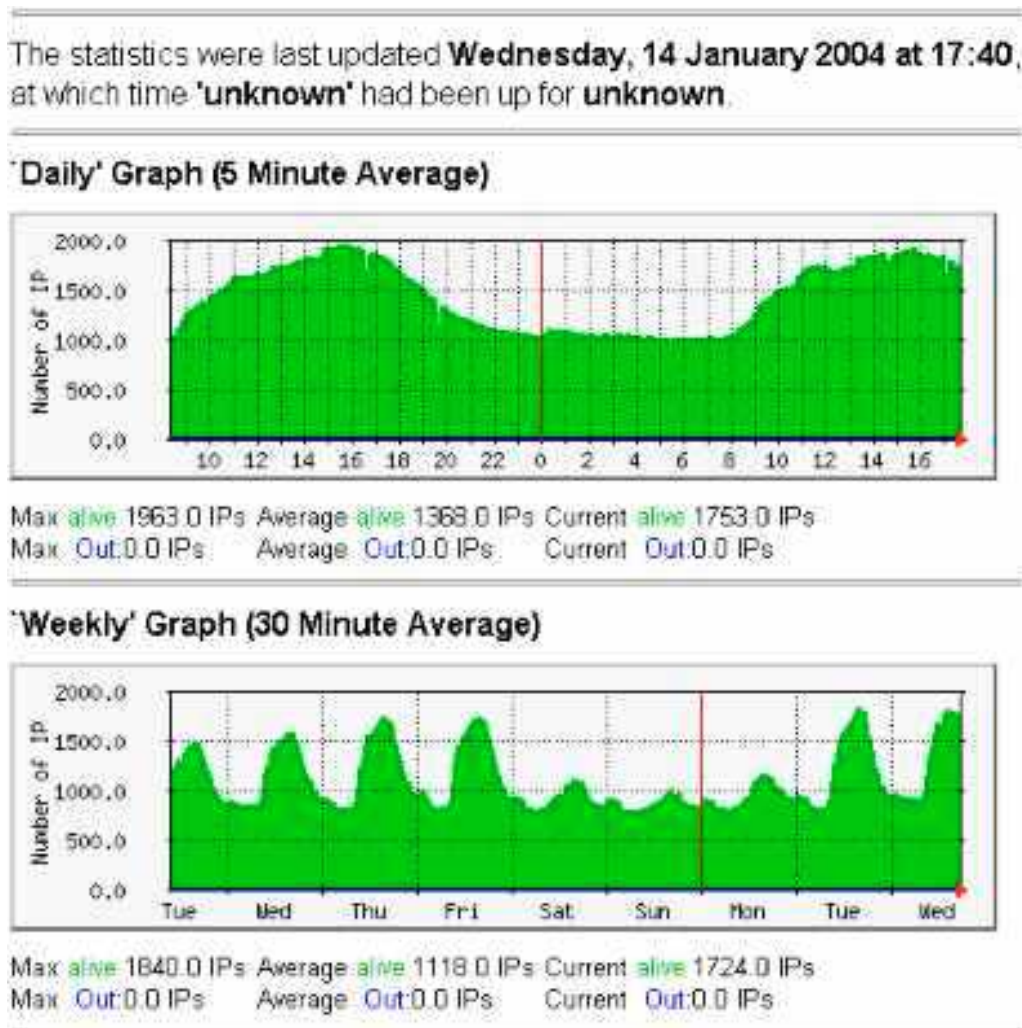
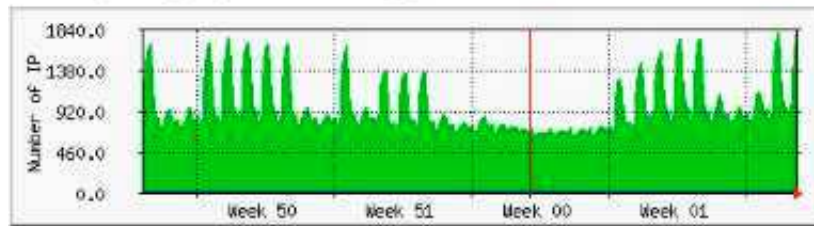


図1 ICMP パケットを用いた計測(1)

図1から5分ごとの計測グラフから、2004年1月14日17時40分に1,753台が接続されている。それまでの1日の5分ごとの計測では、最大値が1,963台であり、平均値が1,368台である。

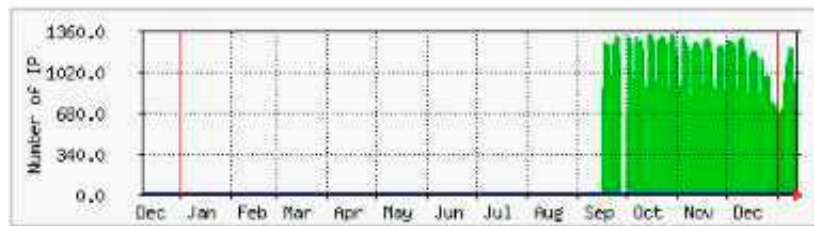
ログデータを解析したところ、接続台数の最大値は、表1に示すとおり、ICMPパケットで計測したとき2,111台であり、80番ポートを利用したとき2,951台である。計測法によって840台の差が生じている。

### Monthly' Graph (2 Hour Average)



Max **alive** 1806.0 IPs Average **alive** 964.0 IPs Current **alive** 1789.0 IPs  
Max **Out**: 0.0 IPs Average **Out**: 0.0 IPs Current **Out**: 0.0 IPs

### Yearly' Graph (1 Day Average)



Max **alive** 1345.0 IPs Average **alive** 1053.0 IPs Current **alive** 974.0 IPs  
Max **Out**: 0.0 IPs Average **Out**: 0.0 IPs Current **Out**: 0.0 IPs

図2 ICMP パケットを用いた計測(2)

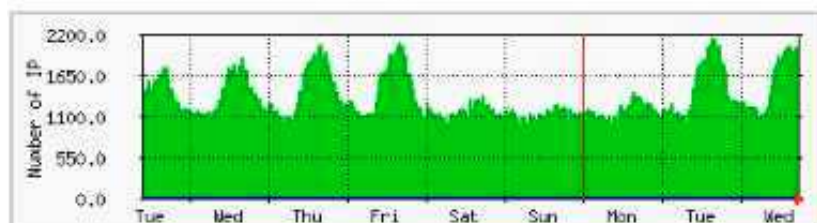
The statistics were last updated **Wednesday, 14 January 2004 at 17:40**,  
at which time '**unknown**' had been up for **unknown**.

### Daily' Graph (5 Minute Average)



Max **alive** 2217.0 IPs Average **alive** 1601.0 IPs Current **alive** 1975.0 IPs  
Max **Out**: 0.0 IPs Average **Out**: 0.0 IPs Current **Out**: 0.0 IPs

### Weekly' Graph (30 Minute Average)



Max **alive** 2168.0 IPs Average **alive** 1348.0 IPs Current **alive** 1945.0 IPs  
Max **Out**: 0.0 IPs Average **Out**: 0.0 IPs Current **Out**: 0.0 IPs

図3 80 番ポートを用いた計測(1)



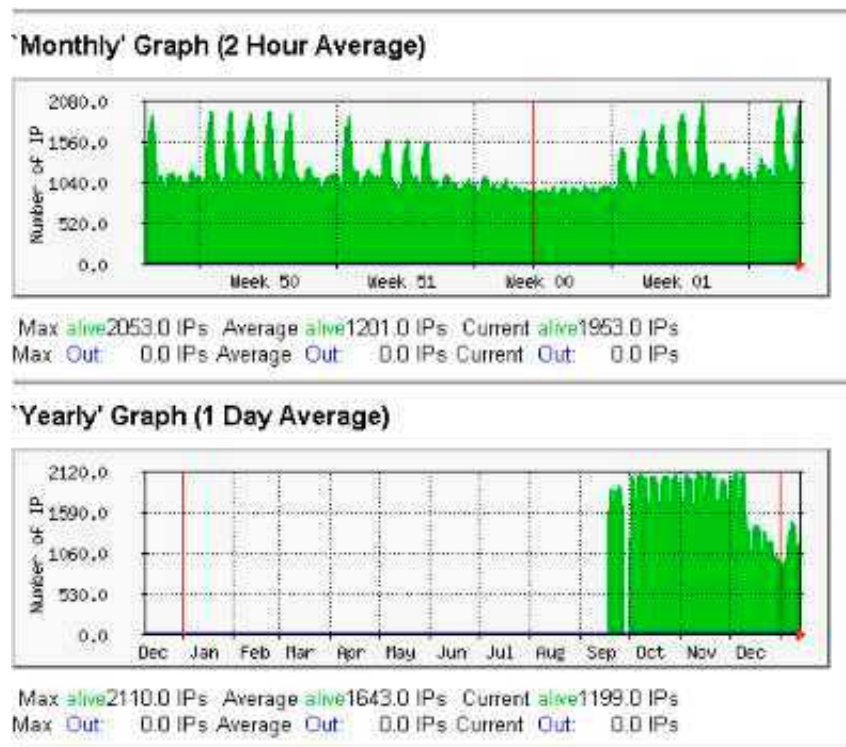


図4 80番ポートを用いた計測(2)

表1 接続台数の最大値

	ICMP パケット	80 番ポート
2003 年 12 月 9 日 17 時	2,115	2,951

## 5 今後の対応

接続されているコンピュータは、約 1,400 台から約 3,000 台の間で変化している。常時接続されているコンピュータとしては、ルータやスイッチ等のネットワーク機器、サーバ類などが考えられる。一方、変動するコンピュータとしては、利用者用のコンピュータシステムやネットワークプリンタなどが考えられる。図 1, 図 3 では、少々判りにくい毎日夜中に 0:00-3:00 に約 100 台程度のコンピュータの増加が見られる。これは、工学部の専門情報処理教育用システムで利用者用コンピュータシステムの設定の初期化を行う作業を行うために起動しているためであった。さらに、全学での MAC アドレス認証システムの導入によって全学での DHCP サービスを開始した[7]ので、より多くのコンピュータが接続されるものと考えられる。このように 3,000 台以上のコンピュータが接続される可能性のあるネットワークとして安全性の確保がますます重要になる。

さらに、2003 年 10 月の旧香川大学と旧香川医科大学(現医学部)との統合によって、接続されるされているコンピュータの台数は、さらに増加している。医学部では、クラス C のア

ドレスが使用されているので、コンピュータの接続状況を確認するには、調べる IP アドレスの範囲を拡大する必要がある。

## 6 おわりに

このように、約 3,000 台以上のコンピュータが接続されるようになると、それらのコンピュータの安全性を基盤センターだけで確保することは、ますます困難になってきている。基盤センターでは、ファイアウォールサーバ[8]やネットワーク監視補助システム[9]、ウィルスチェックサーバ[10]などを導入してネットワークの安全性確保に努めている。しかしながら、MS プラスター類の感染に見られるようにウィルスやワームの侵入を完全に遮断することは困難である。そのために、各利用者が自分が利用するコンピュータの安全性の確保に気を配って頂くことが重要になる。

## 参考文献

- [1] 三谷祐平, 古川善吾: 香川大学キャンパス情報ネットワークの現状, 平成 14 年度電機関係学会四国支部連合大会講演論文集, p. 295, 2003
- [2] 堀口徹, 古川善吾: 工学部キャンパス情報ネットワークの現状, 平成 14 年度電機関係学会四国支部連合大会講演論文集, p. 305, 2002
- [3] IPUSIRON: ハッカーの教科書, データハウス, 2001
- [4] fping プログラム: <http://www.fping.com/>
- [5] nmap プログラム : <http://www.insecure.org/nmap/>
- [6] MRTG プログラム : <http://people.ee.ethz.ch/~oetiker/webtools/mrtg/>
- [7] 総合情報基盤センター幸町分室: 認証 DHCP サーバの運用について, 2003 年 12 月  
<http://www.cc.kagawa-u.ac.jp/user/in/qipman.html>
- [8] 青木昌三, 瀬野芳孝, 曽根計俊: ファイアウォールでのパケット阻止状況の概要, 香川大学情報処理センター年報, pp.24-63, 2001 年 2 月
- [9] 中村邦彦: ネットワーク監視補助システム, 香川大学情報処理センター年報, pp. 18-23, 2001 年 2 月
- [10] 最所圭三: 香川大学におけるウィルス対策メールシステム, 香川大学総合情報基盤センター年報, 2004 年 2 月

# 香川大学におけるウィルス対策メールシステム

最所圭三 (香川大学工学部信頼性情報システム工学科)

## 1 はじめに

インターネットの普及により、多くのコンピュータはネットワークに接続されている。これにより、世界中のコンピュータと瞬時に情報交換ができる反面、OS やネットワークアプリケーションのセキュリティホールをついた侵入やコンピュータウィルス (以降ウィルス) の感染の危機にさらされている。

香川大学では、2003年3月に情報処理センター (現総合情報基盤センター) の機器更新に伴いセキュリティの強化を行った。2003年3月までのシステムではファイアウォールのみで外部からの直接の攻撃を防いでいたが、新システムでは電子メール (以降単にメールと表記する) に対してウィルスチェックを行うサーバを導入し、情報処理センターがサービスするメールサーバ (以降部局メールサーバ) 宛てのメールにウィルスが含まれているかどうかを検査することができるようになった。検査の結果、ウィルスが含まれている場合は、可能なら除去し、可能でなければそのメールを隔離し、ウィルスが含まれていたことを宛先のメールアドレスに通知する。

メールを介してウィルスが感染する場合、ウィルスはメールの添付ファイルに潜み、そのメールを送られたコンピュータに侵入する。ここではウィルスを含むメールをウィルスメールと呼ぶことにする。ウィルスメールがコンピュータに届けられただけではウィルスには感染しないが、その添付ファイルを展開し、実行したりアプリケーションで開いたりすることで感染する。感染したコンピュータは使用しているメーラ (メールの送受信、閲覧、作成を行うソフト) が持つアドレス帳などから次の感染先を選びメールに自分自身を含めて送る。従って、何の対策も施さなければ、ねずみ算的にそのウィルスが広がることになる。

ウィルスを防ぐ方法として、ウィルスメールを見ないか削除することであるが、その見極めは難しい。このため、多くのユーザは、ウィルスバスターやノートン・アンチウィルスなどのウィルス対策ソフトウェアを導入している。ウィルス対策ソフトウェアは、(1) メールをサーバから取り込むときに検査する、(2) メールや添付ファイルをファイルとして保存するときに検査する、(3) 定期的にディスクの中身を検査するなどの処理を行う。これら全てを行っているものもある。ウィルスチェックサーバは(1)の方法をメールがメールサーバに届く前に行うものであり、これをを通してメールサーバがメールを受け取ることによりユーザはウィルスを含まないメールのみを受け取ることになる。

本報告では、このウィルスチェックサーバを組み込んだウィルス対策メールシステムの解説を行う。

## 2 ウィルス対策メールシステムの設計

ウィルスチェックサーバを導入する前は、図 1 に示すように、内や学外のメールサーバの区別なく直接メールの交換を行っていた。さらに、同じサーバ上のユーザに対するメールは直接宛先のユーザのメールボックスに書き込まれていた。この形態でウィルスチェックを行うためには各メールサーバにウィルスチェック機能を持たせなければならない。しかし、このためには台数分のウィルスチェックのソフトウェアが必要になりメールサーバに比例したコストが掛かる上に、研究室で立ち上げているメールサーバ宛てのメールを検査できない。

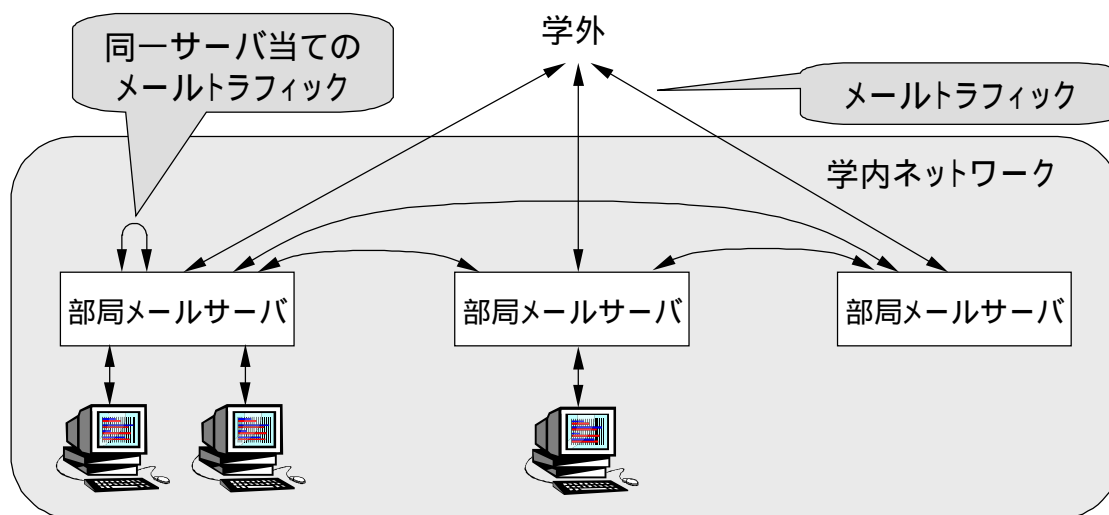


図 1 ウィルスチェックサーバ導入前のメールの流れ

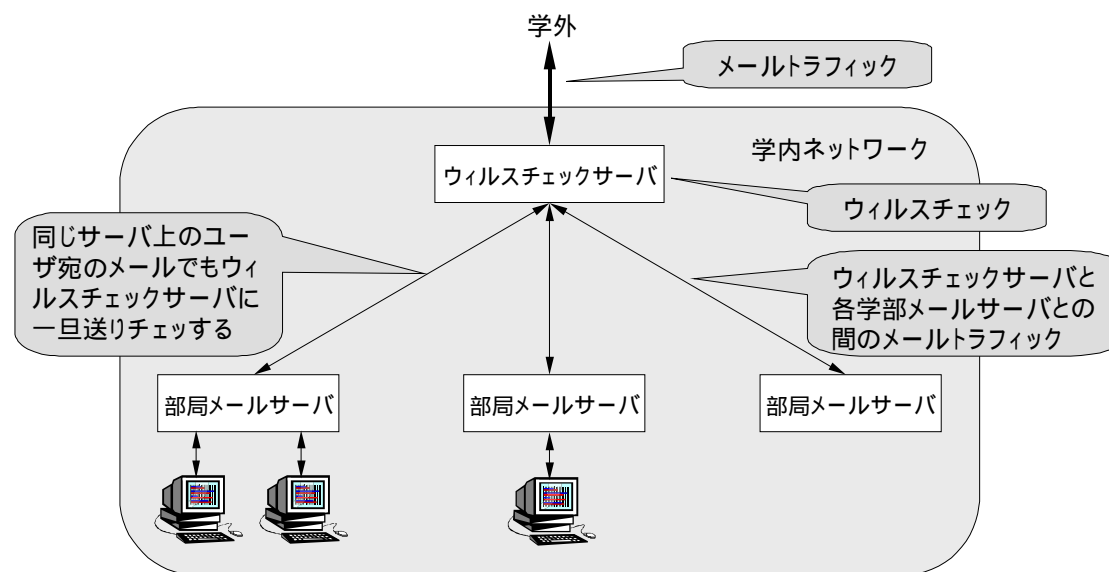


図 2 ウィルスチェックサーバ導入後のメールの流れ

そこで、図 2 に示すように、1 つのウイルスチェックサーバを置き、学外 / 学内 / 同一メールサーバ内に関係なく全てのメールがウイルスチェックサーバを経由して配送できれば 1 台のウイルスチェックサーバだけで学内の全てのメールサーバ宛てのメールおよび全てのメールサーバから発想されるメールの検査を行うことができることになる。これにより、メール経由でウイルスに感染したり、ウイルスの感染源となることを防ぐことができる。つまり、たとえ学外でウイルスに感染されたコンピュータが学内に接続してメール経由でウイルスを他のコンピュータに感染させようとしても防ぐことができるということである。しかし、図 2 の方式を実現するためには下記の 3 つの問題を解決しなければならない。

1. 学外からの全てのメールのウイルスチェックサーバへの配送
2. ウィルスチェックサーバから宛先のメールサーバへの配送
3. 学内メールサーバからの全てのメールのウイルスチェックサーバへの配送

これらの問題を解決するために、新システムで導入した学内向け DNS サーバを用いた。学内向け DNS サーバについては次節で、解決法については 4 節で説明する。

### 3 学内向け DNS サーバ

インターネットにおける全ての通信は、IP アドレスと呼ぶインターネットに接続されているコンピュータに割り当てられている番号を用いて行われる。この番号は 32 ビットからなり、通常オクテット (8 ビット) に分けそれを上位オクテットから 10 進数で表現したものをピリオドで連結した形式で表す。たとえば香川大学のホームページのトップページのサーバの IP アドレスは 133.92.11.23 である。しかし、皆さんはこのサーバを指定するときこの番号ではなく、www.kagawa-u.ac.jp という名前 (以降ホスト名) で指定するだろう。ホスト名から IP アドレスに変換する機構が DNS (Domain Name System) である。新システムでは、特定のコンピュータを学内からだけホスト名で参照できるようにするために学内向け DNS サーバを導入した。

以下、DNS および学内向け DNS サーバについて解説する。

#### 3.1 DNS の仕組み

DNS では、名前を階層的に管理している。www.kagawa-u.ac.jp を例にとると、図 3 に示すように、日本 (jp) の学術関係 (ac) の香川大学 (kagawa-u) の中のコンピュータで名前は www であることを示す。図 3 で、それぞれの をドメインと呼び、 の中の でそれらがコンピュータでない場合、それを外側の から見てサブドメインと呼ぶ。www.kagawa-u.ac.jp の名前から、ac は jp、kagawa-u は ac.jp のサブドメインとなる。正式なホスト名を表す場合は、www.kagawa-u.ac.jp. のように最後にピリオドを付けなければならない。最後のピリオドは世界のトップドメインを表すが、通常は自動的に補完されることが多いので省略することが多い。ただし、メールアドレスを表す場合は逆にピリオドを付けてはならない (メールサ

サーバとは異なるので注意すること)。

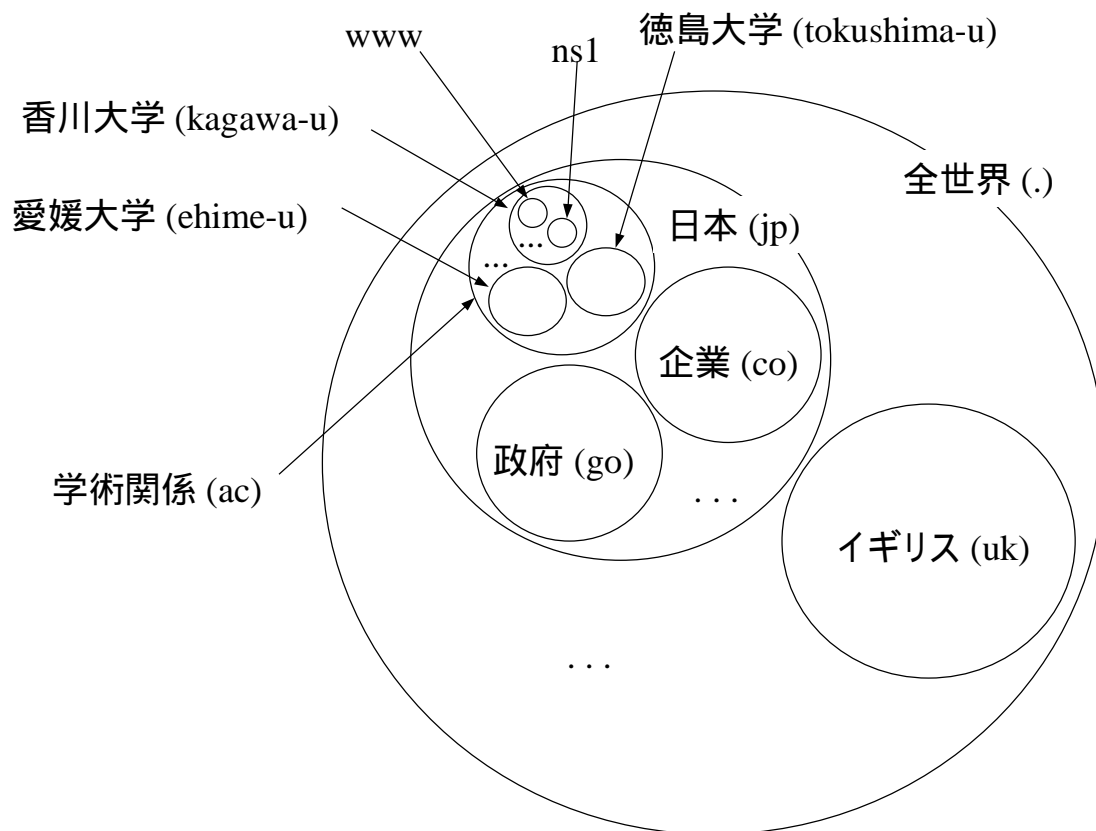


図3 DNS の概念

各ドメインには、そのドメインに含まれるサブドメインやコンピュータのホスト名とIPアドレスを関係づけるネームサーバ (以降 DNS サーバ) と呼ばれるコンピュータが存在する。例えば、香川大学では ns1.kagawa-u.ac.jp がこれに当たる。あるドメインのネームサーバは、(1) そのドメインに直下のコンピュータのホスト名とIPアドレスの対応、(2) サブドメインの名前とそのサブドメインのDNSサーバとの対応を管理するとともに、ホスト名からIPアドレスの変換のサービスも行う。DNSサーバに世界のトップドメインを管理しているDNSサーバを与えることにより、自分の管理下のドメイン以外のホスト名の問い合わせが来たときは、トップドメインを管理しているDNSサーバから順にたどり、最終的にそのホスト名を管理しているサーバからIPアドレスを問い合わせることができる。例えば、www.yahoo.co.jp. のIPアドレスは、トップドメインを管理しているサーバ、jp. を管理しているDNSサーバ、co.jp. を管理しているDNSサーバ、yahoo.co.jp. を管理しているDNSサーバの順に問い合わせることにより得ることができる。DNSは、ホスト名からIPアドレスへの変換だけでなく、メールサービスを行っているホスト名、あるホストの別名の設定やIPアドレスからホスト名の変換も行うことができる。

DNSサーバへの登録は、以下に示すような設定ファイル (ここではゾーンファイルと呼ぶ)

に登録することで行われる。

```
$ORIGIN kagawa-u.ac.jp.  
@          IN      MX      10      vcheck2.kagawa-u.ac.jp.  
          IN      MX      100     vcheck1.kagawa-u.ac.jp.  
ns1        IN      A        133.92.11.17  
www        IN      A        133.92.11.23  
ftp        IN      CNAME    www.kagawa-u.ac.jp.  
vcheck1    IN      A        133.92.6.38  
vcheck2    IN      A        133.92.6.39
```

先頭の \$ORIGIN は管理しているドメイン名を表し、それ以降に登録されている名前とピリオドで連結したものがホスト名になる。‘ @ ’ はそのドメイン自身と同じ名前であることを表し、ここでは kagawa-u.ac.jp. を表す。

“ IN A ” は IP アドレスとの対応を表し、例えば ns1.kagawa-u.ac.jp. の IP アドレスは 133.92.11.17 になる。

“ IN MX 数字 ” はメールサーバとの対応を表し、この例ではメールのドメイン（メールアドレスで@以下の部分）が kagawa-u.ac.jp のメールサーバは vcheck2.kagawa-u.ac.jp. であることを示す。MX は 1 つのメールアドレスに対して複数のメールサーバを割り付けることができる。この場合、MX の後ろの数字の小さいものからメールを転送できるか試み、失敗すれば次に数字の大きなものに転送を試みていく。

“ IN CNAME ” はあるホスト名のコンピュータを別の名前参照するために用いるもので、例では ftp.kagawa-u.ac.jp. の実体は www.kagawa-u.ac.jp. であることを示す。これにより、1 つのコンピュータを目的に応じて別の名前参照することが可能となる。

MX も CNAME にはどのようなホスト名を指定してもよいが、当然ながら指定されたホスト名のコンピュータの管理者の了解を取らなければならない。

ネームサーバに名前を登録する利点はこれだけでなく、ホスト名でホームページのサーバを公開している場合に、サーバの IP アドレスが変わってもネームサーバの登録内容を変更するだけで、他にアナウンスしなくても良いことである。IP アドレスを公開していた場合、IP アドレスが変われば全ての公開先に連絡しなければならない。

### 3.2 学内向け DNS

皆さんの中には、名前参照したいのだが学内のみで使うコンピュータなので、外部から名前参照されたくない等の理由により DNS に登録されていない方もおられるだろう。

このため、新システムでは、世界的な DNS から外れた（つまり ac.jp. を管理している DNS サーバが知らない）DNS サーバ（以後、学内向け DNS サーバ）を設置した。ただし、学内向け DNS サーバには世界のトップドメインを管理している DNS サーバが与えられているので、香川大学のドメイン以外に対しては今までの DNS サーバ（以後、学外向け DNS サーバ）と同じ結果が得られる。これらの DNS サーバのゾーンファイルに、図 4 で示すような内容を登

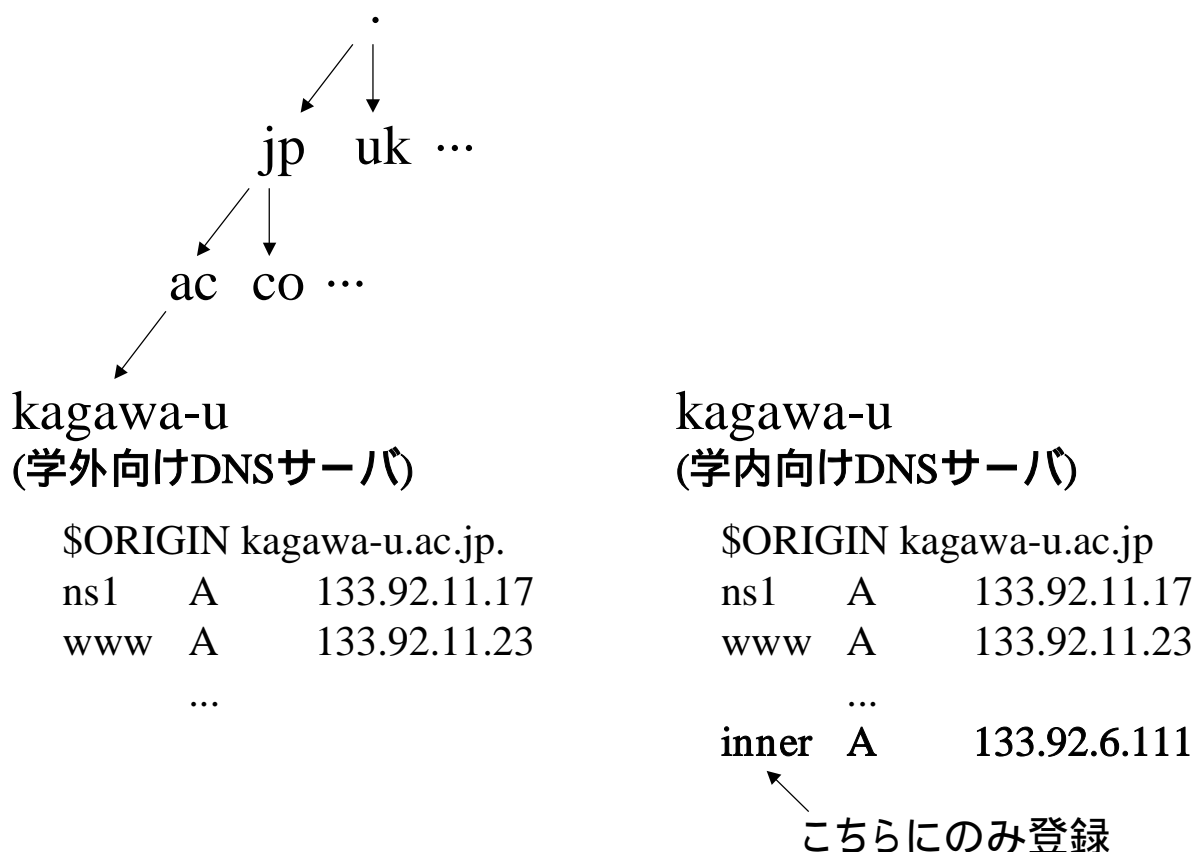


図4 ゾーンファイルの設定例

録したらどうなるであろうか。両方のサーバとも ns1.kagawa-u.ac.jp と www.kagawa-u.ac.jp の IP アドレスを与えるが、inner.kagawa-u.ac.jp の IP アドレスは学内向け DNS サーバにも与える。学内向け DNS サーバは世界的な DNS に入っていないので、このサーバを直接指定しない限りは inner.kagawa-u.ac.jp の IP アドレスを得ることができない。つまり、学内からだけ学内向け DNS サーバを指定できるようにすることで、このサーバにのみ登録されたホスト名は、学内だけで有効になる。

学内向け DNS サーバを使用するためには、2つの作業が必要になる。1つは学内でのみホスト名で参照するコンピュータを学内向け DNS サーバに登録することで、もう1つは学内のみで参照できるホスト名を参照したいコンピュータの DNS サーバを学内向け DNS サーバに変更することである。前者は学内向け DNS サーバの管理者が行い、後者は自分自身で行うことになる。新システムで同様に利用できるようになった DHCP サーバによって IP アドレスが割り当てられるコンピュータに対しては、自分で固定して設定していない場合は自動的に学内向け DNS サーバが割り当てられるように設定しているので、特に設定する必要はない。ただし、学内向け DNS サーバは学外からアクセスできないので、DNS サーバを固定で指定している場合に、そのコンピュータを学外で使用すると学内向け DNS サーバにアクセスできな



いので、注意が必要である。学外向け DNS サーバおよび学外向け DNS サーバのホスト名と IP アドレスは現時点では以下のようにになっている。

	学内向け DNS サーバ	学外向け DNS サーバ
工学部以外	kns1.kagawa-u.ac.jp(133.92.6.17)	ns1.kagawa-u.ac.jp(133.92.11.17)
工学部	ens1.eng.Kagawa-u.ac.jp(133.92.167.2)	dns.eng.kagawa-u.ac.jp(133.92.167.17)

#### 4 ウィルスチェックサーバの運用

本節では、2節で挙げた問題点を解決し、ウィルス対策メールシステムを運用するための方法を示す。

##### 1. 学外からの全てのメールのウィルスチェックサーバへの配送

これは、ウィルスチェックを行うメールのドメインのメールサーバがウィルスチェックサーバになるように、DNS サーバで設定すればよい。例えば、ゾーンファイルの内容を以下のようにすればよい。

```
$ORIGIN kagawa-u.ac.jp.
@           IN      MX 10      vcheck2.kagawa-u.ac.jp.
           IN      MX 100     vcheck1.kagawa-u.ac.jp.
$ORIGIN cc.kagawa-u.ac.jp.
@           IN      MX 10      vcheck1.kagawa-u.ac.jp.
           IN      MX 100     vcheck2.kagawa-u.ac.jp.
```

中略

```
vcheck1    IN      A           133.92.6.38
vcheck2    IN      A           133.92.6.39
```

実際のシステムでは、負荷分散とウィルスチェックサーバの障害時の対策のために、ホスト名が vcheck1.kagawa-u.ac.jp. と vcheck2.kagawa-u.ac.jp. であるウィルスチェックサーバを2台導入しているが、部局によって優先するサーバが異なっている。

##### 2. ウィルスチェックサーバから宛先のメールサーバへの配送

2つの方法がある。1つはウィルスチェックサーバにメールのドメイン毎にメールサーバを指定するものである。もう1つは学内向け DNS サーバと組み合わせることである。後者は、上記の学外向け DNS サーバのゾーンファイルに対応する部分を以下のように変更し、ウィルスチェックサーバが用いる DNS サーバを学内向け DNS サーバに設定することで実現できる。この DNS サーバを用いると kagawa-u.ac.jp のメールサーバは mail.cc.kagawa-u.ac.jp. というメールサーバであることが得られる。

```
$ORIGIN kagawa-u.ac.jp.  
$ORIGIN kagawa-u.ac.jp.  
@          IN      MX 10          mail.cc.kagawa-u.ac.jp.  
$ORIGIN cc.kagawa-u.ac.jp.  
@          IN      MX 10          mail.cc.kagawa-u.ac.jp.  
mail      IN      A              133.92.6.29
```

略

本システムでは、ウイルスチェックサーバが参照する DNS サーバ以外を変更する必要がないので学内向け DNS サーバと組み合わせる方法を採用した。

研究室でメールサーバを立ち上げた場合には、学外向け DNS サーバにはウイルスチェックサーバを、学内向け DNS サーバには立ち上げたメールサーバを MX で指定することで、学外向け DNS サーバを参照しているサーバからのメールに対してウイルスチェックサーバでの検査が行われる。

### 3. 学内メールサーバからの全てのメールのウイルスチェックサーバへの配送

同一サーバ内のユーザ宛以外のメールに関しては、smarthost と呼ぶ方法で実現できる。メールサーバは通常、同一サーバ内のユーザ宛以外のメールをそのメールの宛先のメールサーバの MX を検索して送るが、smarthost が設定されている場合は、smarthost で指定されているメールサーバに送る。

この機能は古いメールサーバには装備されていない場合がある。このようなサーバを研究室のメールサーバとした場合は、そのサーバが用いる DNS サーバを学外向け DNS サーバにしなければならない。以下で説明するが、セキュリティ強化のために、部局メールサーバはウイルスチェックサーバ以外から受け取らないように設定しているからである。

同一サーバ内のユーザ宛でのメールをウイルスチェックサーバに送ることを実現するためには、新たなメールサーバの導入が必要であった。この新たなメールサーバがダウンした場合、部局メールサーバ宛の全てのメールの配送が止まるなどの問題点があるので、この方法を採用しなかった。代わりに各ユーザのメールの送信先をウイルスチェックサーバに指定するように変更してもらうことで対処した。つまり、ユーザからのメールを部局メールサーバを通じて送らないようにすることにより、部局メールサーバから同一サーバ宛のメールがなくなる。しかし、以前の設定のままのユーザがいた場合、そのユーザがウィルスメールを同一サーバ内のユーザに送信した場合に、ウィルス検査が行われずに宛先のユーザにウィルスメールが送られることになる。このセキュリティホールをなくすために、部局メールサーバはウイルスチェックサーバ以外からメールを受け取らないように設定した。つまり、設定を忘れたユーザからのメールは拒否されるので、部局メールサーバにウィルスメールが入り込まなくなる。

これを機会に各部局で使用する POP サーバ (メールを受信するサーバ) と SMTP サーバ

(メールを送信するサーバ) の名前を統一した。職員の場合は、それぞれ pop.部局名.kagawa-u.ac.jp と smtp.部局名.kagawa-u.ac.jp とした。部局名には、ed (教育学部)、jl (法学部)、ec (経済学部)、eng (工学部)、ag (農学部)、cc (総合情報基盤センター)、ao (事務局) がある。また学生に対しては、それぞれ stopop.部局名.kagawa-u.ac.jp と stsmtp.部局名.kagawa-u.ac.jp とした。pop と stopop の各部局メールサーバの別名、smtp と stsmtp はウイルスチェックサーバの別名として定義されている。

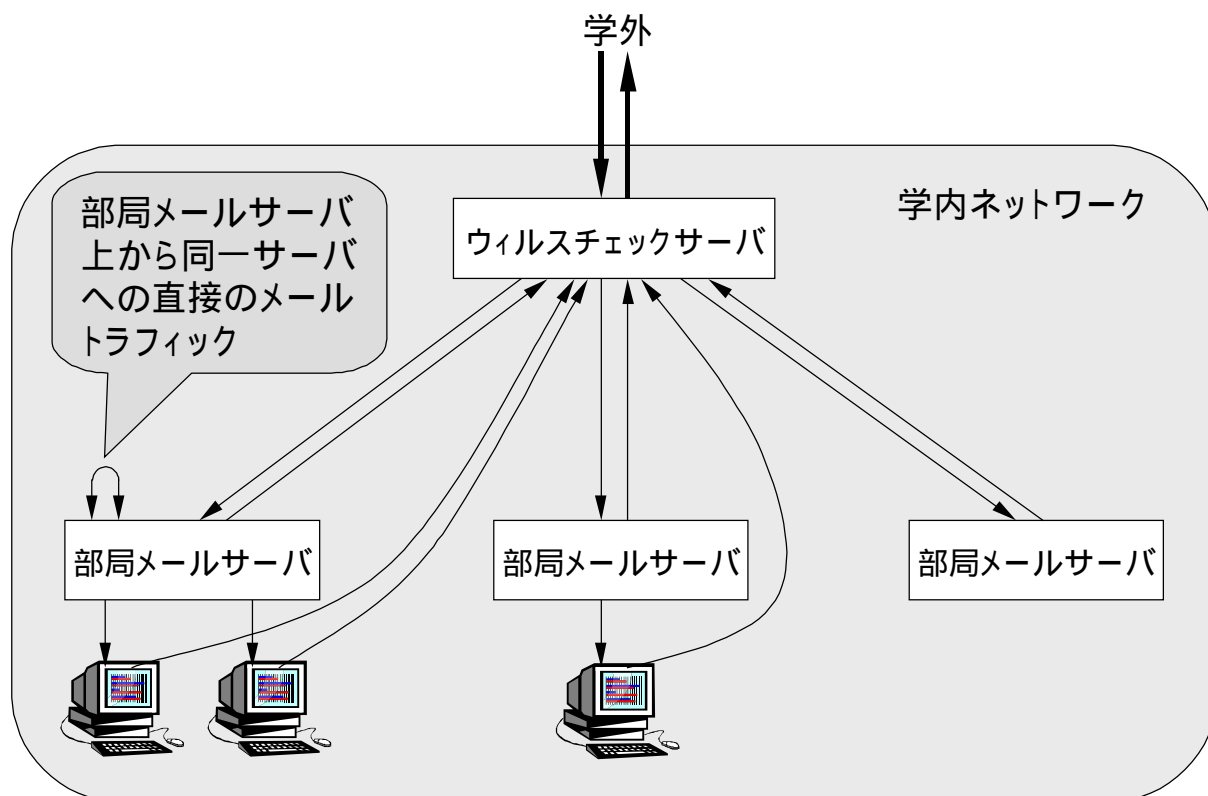


図5 ウィルスチェック対策メールシステムにおけるメールの流れ

以上の方策を行うことにより、部局メールサーバに関しては図5で示すようなメールの配送を行うことができるようになった。ユーザが部局メールサーバにリモートログインしUNIXの機能を用いて自サーバのユーザ宛にメールを出さない限り、部局メールサーバ宛での全てのメールはウイルスチェックサーバを通ることになる。部局メールサーバにログインして直接メールを出す問題は残っているが、一般ユーザがこのような語りで利用することはまずなく、必要なら学部サーバへのリモートログインを禁止することで完全になくすことも可能である。

## 5 まとめ

以上、香川大学におけるウィルス対策メールシステムの解説を行った。本システムで防ぐことができるウィルスはメールを経由したものであり、ホームページにアクセスすることによって感染するウィルスやMS-BLASTのようなインターネットに繋いだけで感染するようなウィルスを防ぐことができないので注意してもらいたい。

あと、ウィルスチェックサーバを導入する副産物として、今までは研究室でメールサーバを立ち上げる場合は、ファイアウォールでそのサーバのSMTPのポートを通すように設定しなければならなかったが、メールを実際に受けるサーバをファイアウォールの中にあるウィルスチェックサーバだけにできるので、メールの送受信にのみ用いるならファイアウォールへの登録を必要としなくなったことがある。さらに、学外のメールサーバと直接通信をしないので、学外からメールサーバのソフトのセキュリティホールをついた攻撃を受けなくなることも利点の1つである。

# X3D について

## — Web3D は復権するか？ —

香川 考司  
工学部信頼性情報システム工学科

### 概要

2002年7月にX3Dの最終ワーキングドラフトがWeb3D Consortiumにより発表された。X3DはVRML97の後継として設計された3Dグラフィックスのファイル形式の規格である。2003年2月にISOに提出されており、審査を経て、ISOで国際標準規格として認められる予定である。本稿ではこのX3Dを簡単に紹介することにする。

## 1 X3Dとは

X3DとはeXtensible 3Dの略であり、VRML97の後継として設計された3Dグラフィックスのファイル形式の規格である。

VRML97<sup>1</sup>はその名の通り1997年に制定され、WWWで(HTMLのような2次元だけではなく)3次元のコンテンツを表現する手段として、大きな期待を寄せられていた。しかし結局当初騒がれたほどには普及しなかった。これは当時のコンピュータとネットワークの性能が貧弱だったのが大きな原因であると思われる。当時のネットワークの性能ではVRMLを閲覧するためのソフトウェア(プラグイン)をダウンロードするには、とても時間がかかった<sup>2</sup>。それでいて見ることができる絵はデータを小さくしているためにみずばらしくて、しかもハードウェアの性能が足りないため滑らかに動かなかった。

しかし、それから5年以上たった今、状況は大きく変わったようである。ADSLや光ケーブルなどのブロードバンドが普及し、X3D/VRMLのプラグインはそれほど大きなソフトウェアとは感じられなくなった。またデータの大きさをケチる必要もなくなって、物体を高精細にモデル化できるようになった。最近のNVIDIA社やATI社などの3Dグラフィックスアクセラレータは、ちょっとした3D図形の描画には充分すぎるほどの性能を持っている。

一方、X3D/VRMLに代わるような標準的な3Dグラフィックスフォーマット

<sup>1</sup>VRML2.0はVRML97がISOで認められる前のドラフト段階での仕様の名前である。VRML97とVRML2.0の間に違いはほとんどない。

<sup>2</sup>28.8kbpsのモデムを使っていたと仮定すると、理想的な状況でも当時の代表的なVRMLプレイヤーであるCosmoPlayer(約3.2MB)のダウンロードに15分程度かかっていた計算になる。

トは（少々値段のはるソフトウェアで作成するフォーマットを別とすれば<sup>3</sup>）他に適当なものは見当たらないようである。WWW のコンテンツを、効果的に伝えるために 2 次元ではなく 3 次元を利用したいと思うことは、誰でも時々あるのではないかと思う。これまでは適当なフォーマットがないために我慢せざるを得なかった。しかし、X3D の制定をきっかけに WWW で 3D コンテンツを利用しようと言う気運が再び盛り上がる可能性がある。VRML97 の時のようにブームとはならなくても、国際標準規格として認められれば、少なくとも異種ソフトウェア間のデータ交換用フォーマットとして定着することが期待できる。

X3D は eXtensible というその名の通り、容易に拡張ができるように設計されている。逆にこれによりコアの部分は小さくでき、必要に応じて閲覧ソフトを軽量に実装することもできるようになった。そのため、携帯電話などモバイル機器での利用が広まる可能性もある。

さらに X3D の最大の特徴は XML[1] 化された文法である。HTML (XHTML) や SVG[2] など、XML に基づく他のファイル形式と同じように、<何か> ~ </何か> という形のタグを用いる。例えば円筒形の定義は

```
<Shape>
  <Appearance>
    <Material diffuseColor="0.6 0.4 0" />
  </Appearance>
  <Cylinder radius="2" />
</Shape>
```

というようになる。

一方、X3D には従来の VRML97 と同系統の文法（VRML エンコーディング）も用意されている。上と同等の内容がこのエンコーディングでは、以下のようなになる。

```
Shape {
  appearance Appearance {
    material Material {
      diffuseColor 0.6 0.4 0
    }
  }
  geometry Cylinder {
    radius 2
  }
}
```

これに対して先程の文法を XML エンコーディングと言う。HTML に慣れていると恐らく XMLの方が読みやすいのではないだろうか？ また XML は Java などプログラム言語で操作するときにも、操作のライブラリがあって便利である。本稿では、これから主流になるとと思われる XML エンコーディングを中心に紹介することにする。（“X3D”という言葉は狭い意味に用いられた場合は、XML エンコーディングの方のみを指すこともある。）

<sup>3</sup>大学は基本的にピンボーなところなのでタグというのは重要な特徴の一つである。

なお、X3D は、VRML エンコーディングでは VRML97 の上位互換となるように設計されている。これにより従来の VRML97 の資産もそのまま継承できる。2 つのエンコーディング間の変換プログラムなども入手可能である。

## 2 X3D 関連ソフトウェアを入手する

本稿の執筆時現在<sup>4</sup>、X3D の XML エンコーディングを読み込める Web ブラウザ用の VRML プラグインが、いくつか出初めている。Web ブラウザに VRML/X3D 用のプラグインがインストールされているかどうかは、“VRML Plugin and Browser Detector” のページ<sup>5</sup> で調べることができる。また、新しい VRML/X3D のプラグインの情報も、このページで手に入れるのが便利なのである。

今のところ、Windows 用には、Media Machines 社の Flux<sup>6</sup>、Octaga 社の Octagon Player<sup>7</sup>、Vcom3D 社の Vcom3D Venues<sup>8</sup>、Bitmanagement Software GmbH の BS Contact VRML<sup>9</sup>などのいくつかのプラグインが利用できる。現時点で、筆者がお勧めするのは BS Contact VRML である。

一方、Linux や Mac 用は利用できるプラグインはまだ少なく、今のところ FreeWRL<sup>10</sup> という Netscape/Mozilla 用のオープンソースのプラグインだけのようである。このようなプラットフォームで X3D 用のプラグインが普及するまでの一時しのぎとして、XML エンコーディングの X3D ファイルを、XSLT で従来の VRML97 に変換することもできる。この変換はファイルの作成者が前もって行なっておく必要がある。そうすると閲覧者は VRML97 用プラグインを利用して再生することが可能である。(もちろんこの場合は X3D で新しく導入された特徴は利用することはできない。) また、VRML エンコーディングの X3D の場合は、新しく拡張された部分を使っていなければ、従来の VRML97 用のプラグインで、たいてい再生できるようである。

従来の VRML97 用プラグインとしては、Compute Associates 社が現在配布している CosmoPlayer (<http://ca.com/cosmo/>)、Parallel Graphics 社の Cortona (<http://www.parallelgraphics.com/products/cortona/>) などが Windows と Mac で利用できる。Cortona は VRML97 対応とはいえ、X3D のいくつかの機能を先取りしているようである。

X3D の一般的な情報 (仕様やその他のソフトウェアの情報) については、Web3D コンソーシアム内のホームページ (<http://www.web3d.org/x3d.html>)

<sup>4</sup>2004 年 1 月

<sup>5</sup><http://cic.nist.gov/vrml/vbdetect.html>

<sup>6</sup><http://mediamachines.com/download.html> Internet Explorer 専用 (ActiveX bridge を利用すれば、Netscape/Mozilla でも利用可能とのことである。)

<sup>7</sup>[http://www.octaga.com/prod\\_octagonfpl.html](http://www.octaga.com/prod_octagonfpl.html) Internet Explorer 専用

<sup>8</sup><http://www.vcom3d.com/Viewer.htm> Internet Explorer 専用

<sup>9</sup>[http://www.bitmanagement.de/products/bs\\_contact\\_vrml.en.html](http://www.bitmanagement.de/products/bs_contact_vrml.en.html) Internet Explorer および Netscape/Mozilla 用

<sup>10</sup><http://freewrl.sourceforge.net/>

から入手できる。

## 3 X3D を作成する

### 3.1 X3D-Edit について

X3D の作成には、通常のテキストエディタも使用できるが、X3D の作成に特化した構造化エディタとして、X3D-Edit がある。X3D-Edit は <http://www.web3d.org/TaskGroups/x3d/translation/README.X3D-Edit.html> から入手可能である。X3D-Edit は VRML97 への変換・VRML97 プラグインを使用して表示という作業をボタン一つで実行してくれる。

少し大きな X3D ファイルを作成するには、エラーを防ぐために是非とも X3D-Edit のような構造化エディタを使うべきであろう。しかし、ここでは、小さい例しか紹介しないので、X3D ファイルは通常のテキストエディタ（メモ帳・Emacs など）で作成することにする。

### 3.2 スケルトン

X3D のファイルのスケルトンは次のようになる。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE X3D PUBLIC
    "ISO//Web3D//DTD X3D 3.0//EN"
    "http://www.web3d.org/specifications/x3d-3.0.dtd">
<X3D profile="Interchange">
<head>
  <meta name="description" content="Simple Test" />
</head>
<Scene>

  <!-- Scene graph nodes are added here. -->

</Scene>
</X3D>
```

1～4 行目は XML ではおなじみの書き方である。1 行目の encoding の値はここでは UTF-8 になっているが、使用する文字コードに応じて ISO-2022-JP, SHIFT\_JIS, EUC-JP などに変更する。

5 行目の X3D タグの profile= の部分が、X3D に特徴的な部分で、このファイルで使用するタグの集合を指定する。Core, Interchange, Interactive, Immersive, MPEG4, Full の順に使用できるタグの種類が増える。（その代わり再生に必要なソフトウェアが重くなる。）このうち Immersive が従来の VRML97 に相当するようである。

<head> ～ </head> の部分（6～8 行目）はオプションであるが、上記のように meta タグで description（内容の短い説明）を入れておくと良いだろう。



そして、<Scene> ~ </Scene>の部分（9～11行目）に実際の3D物体を記述する。

XMLエンコーディングされたX3Dファイルの拡張子は.x3dである。（VRMLエンコーディングの場合は従来通り.wrlであるが、x3dvという新しい拡張子を採用するという情報もあるようだ。）

X3Dで作成したグラフィックスをHTMLファイル中に画像として埋め込むためには、HTMLのobjectタグを用いる。

```
<object data="test.x3d" type="model/x3d+xml"
        width="256" height="256">
X3Dを表示できないブラウザではこの部分が出力されます。
</object>
```

type属性に用いる文字列（MIME type）は、model/x3d+xmlを指定する<sup>11</sup>。

### 3.3 3D物体

それでは、Sceneに物体を追加してみよう。上記のスケルトンのコメント（<!-- Scene graph nodes are added here. -->）の部分を次のように書き換えてみる。（以下の例ではすべてこの部分のみを示す。）

```
<Shape>
  <Cylinder />
</Shape>
```

Shapeというタグの中にCylinderというタグを入れることによって、円筒形（Cylinder）が現れる（図1左）。しかし、これでは色が白（実際は明るい灰



図1: 白とピンクのCylinder

色）で立体感がほとんどない。そこでこの円筒に色をつけてみることにする。色を付けるにはAppearanceとMaterialというタグを用いる。MaterialタグのdiffuseColorという属性にRGB値を指定することによって、物体に色をつけることができる（図1右）。

```
<Shape>
  <Cylinder />
</Shape>
```

<sup>11</sup>ただし、正式にはMIME typeはまだ決まっていないようである。

```

    <Appearance>
      <Material diffuseColor="1 0.5 0.5" />
    </Appearance>
  </Shape>

```

なお、diffuseColor という属性は物体が反射する色を指定する。これに対して emissiveColor という、物体が自ら発する光の色を指定する属性もある。

Cylinder は既定値では半径 1, 高さ 2 になっている。radius と height という属性を変更することによって、この値を変更することができる。

```

<Shape>
  <Cylinder height="1.5" radius="1.8" />
  <Appearance>
    <Material diffuseColor="1 0.5 0.5" />
  </Appearance>
</Shape>

```

この結果を図 2 左に示す。

### 3.4 物体の移動

3D 物体はデフォルトでは中心に座標の原点が来るように配置される。物体を平行移動させたり、回転させたりするには Transform というタグで移動させたい部分を囲む (図 2 右)。

```

<Transform rotation="1 0 0 0.785">
  <Shape>
    <Cylinder />
    <Appearance>
      <Material diffuseColor="1 0.5 0.5" />
    </Appearance>
  </Shape>
</Transform>

```

Transform タグには次のような属性がある。

属性名	説明	例
translation	平行移動の各座標軸方向	1 0 0
rotation	回転軸 (1~3 番目) と回転量 (4 番目)	1 0 0 0.785
center	回転と拡大・縮小の中心点	0 1 0
scale	拡大・縮小の各座標軸方向	2 1 0.5
scaleOrientation	拡大・縮小の向き (回転軸と回転量)	0 1 0 0.785

角度の単位はラジアンである。例えば translation="1 0 0" は x 軸方向へ 1 移動することを表し、rotation="1 0 0 0.785" は x 軸中心に 0.785 ラジアン (約 45 度) 回転することを表す。なお、Transform タグで平行移動・回転・拡大が同時に 2 つ以上指定された時は、拡大・縮小 → 回転 → 平行移動の順に適用される。



図 2: Cylinder の属性と回転

### 3.5 いろいろな 3D 形状

基本的な物体形状としては、Cylinder の他に Box (直方体)、Sphere (球)、Cone (円錐) などがある。また、文字列を表示するには Text というタグを用い、表示する文字列を string という属性で与える。

```
<Shape>
  <Text string="Hello!" />
</Shape>
```

ちなみに Text タグで string に日本語を与えると、現時点では BS Contact VRML をはじめ、多くの X3D プラグインでうまく表示されないようである。

図 3 にこれらのさまざまな形状を適当な角度から見たものを示す。これら

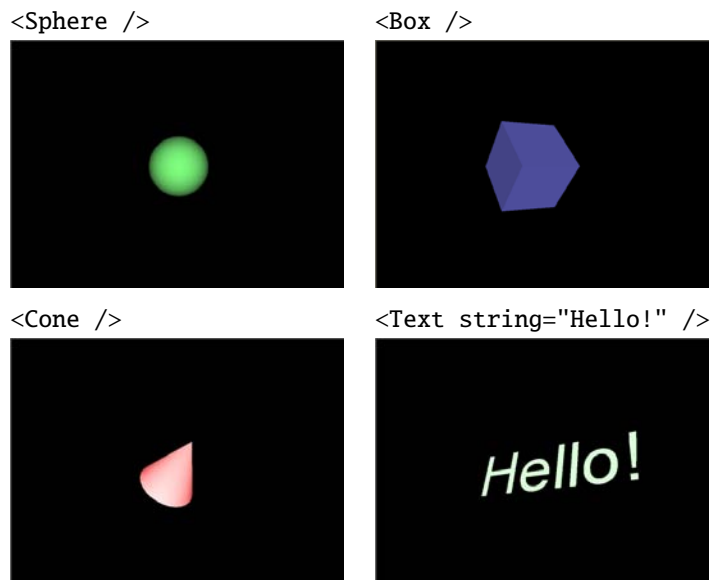


図 3: いろいろな形状

のタグは次に示すような属性を指定することもできる。

タグ	属性	意味	既定値
Box	size	$x, y, z$ 軸方向のサイズ	2 2 2
Sphere	radius	半径	2
Cone	height	高さ	2
	bottomRadius	底面の半径	1
Text	string	文字列	(必須)

さらに、ここでは詳しく説明できないが、IndexedFaceSet (複数のポリゴンを組み合わせた形状)、Extrusion (2次元の多角形に厚みを持たせた形状)、ElevationGrid (2次元座標の上で高さを指定する形状) など、もう少し複雑な形状を表現できるタグもある。

### 3.6 ハイパーリンク

HTML でテキストや画像にリンクを設定できるのと同じように、X3D でも物体にリンクを貼りつけることができる。Anchor というタグを用いる。

```
<Anchor url="http://stwww.eng.kagawa-u.ac.jp/">
  <Shape>
    <Sphere />
  </Shape>
</Anchor>
```

これで、url という属性でリンク先を指定する。リンクを貼り付けた物体をクリックすると、指定されたリンク先にジャンプする。

### 3.7 視点の変更

視点はデフォルトでは (0, 0, 10) という座標にあり、原点方向 (つまり (0, 0, -1) 方向) を向いている。(VRML の座標系は  $x$  軸が画面の右向き、 $y$  軸が画面の上向き、 $z$  が画面から手前方向に伸びる。つまり右手座標系である。)

Viewpoint というタグで視点を他の座標に指定することができる。視点は複数指定することができて、その場合はブラウザの操作で視点を切替えることができるようになる。例えば、

```
<Scene>
  <Shape>
    <Box />
  </Shape>
  <Viewpoint description="first" orientation="1 0 0 -0.785"
    position="0 10 10" />
  <Viewpoint description="second" orientation="0 1 0 0.785"
    position="10 0 10" />
  <Viewpoint description="third" orientation="-1 1 0 1.57"
    position="10 10 0" />
</Scene>
```

とすると、3つの地点から原点方向にある立方体を見ることができる。WebコンテンツとしてX3Dを見る時、閲覧者にとってはマウスで視点を移動することは、慣れないと難しいので、コンテンツの作成者がいくつかの適切な視点をあらかじめ用意しておくのが親切であると思われる。

X3Dにはこの他にも多くの種類のタグがある。ここではとてもすべては紹介できないので、興味を持った方は、X3Dの仕様について、[http://www.web3d.org/TaskGroups/x3d/x3d\\_spints.html](http://www.web3d.org/TaskGroups/x3d/x3d_spints.html) を参考にしていきたい。また、X3Dのサンプルは<http://www.web3d.org/TaskGroups/x3d/translation/examples/contents.html> で見つけることができる。

## 4 X3Dをプログラムで生成する

エディタ上で手作業でX3Dファイルを作成すると、どうしても作成可能な作品には限界がある。ここではプログラムでX3Dファイルを作成してみよう。ここで作成するのは図4のような色立体である。プログラミング言語は何でも構わないが、ここではC言語を用いることにする。プログラムのソースは図5のようになる。

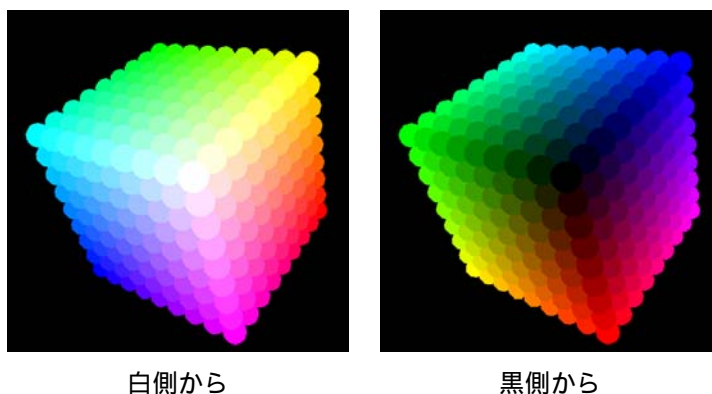


図4: X3Dによる色立体

二重引用符はCの文字列を表すのに使用するため、X3D中の文字列を囲むためには、二重引用符(“)ではなく、一重引用符(’)を使用している。XMLではどちらの引用符も同じように使えることになっている。

ループの中で $9 \times 9 \times 9 = 729$ 個の球を色と位置を変えながら描画している。なお、3Dアクセラレータのないコンピュータで再生するためにはstepの値を2くらい( $5 \times 5 \times 5 = 125$ 個の球)にした方が良いでしょう。

Viewpointは(6, 6, 6) (白側)と(-6, -6, -6) (黒側)の2地点から原点方向に向くように設定している。

```

#include<stdio.h>
#include<math.h>

int main(int argc, char** argv) {
    int i, j, k, step=1;

    /* X3D のヘッダ部などを出力する */
    printf("<?xml version='1.0' encoding='UTF-8'?>\n");
    printf("<!DOCTYPE X3D PUBLIC\n");
    printf("    'ISO//Web3D//DTD X3D 3.0//EN'\n");
    printf("    'http://www.web3d.org/specifications/x3d-3.0.dtd'\n");
    printf("<X3D profile='Interchange'\n");
    printf("<head\n");
    printf("    <meta name='description' content='Color Cube' />\n");
    printf("</head\n");
    printf("<Scene\n");

    /* 3重ループで Sphere を出力する */
    for(i=0; i<=8; i+=step) {
        for(j=0; j<=8; j+=step) {
            for(k=0; k<=8; k+=step) {
                printf("<Transform translation='%f %f %f'\n", 0.5*i-2, 0.5*j-2, 0.5*k-2);
                printf("    <Shape\n");
                printf("        <Appearance\n");
                printf("            <Material emissiveColor='%f %f %f'\n",
                    (double)(i*32)/256.0, (double)(j*32)/256.0, (double)(k*32)/256.0);
                printf("            diffuseColor='0 0 0' />\n");
                printf("        </Appearance\n");
                printf("        <Sphere radius='0.25' />\n");
                printf("    </Shape\n");
                printf("</Transform\n");
            }
        }
    }

    /* 視点の設定 */
    printf("    <Viewpoint description='white' orientation='-1 1 0 %f'\n",
        acos(sqrt(3)/3.0));
    printf("    position='6 6 6' />\n");
    printf("    <Viewpoint description='black' orientation='-1 1 0 %f'\n",
        -acos(-sqrt(3)/3.0));
    printf("    position='-6 -6 -6' />\n");

    /* 閉じタグ */
    printf("</Scene\n");
    printf("</X3D>\n");
}

```

図 5: cube.c

## 5 X3Dの新しい機能

ここまでで紹介した X3D の機能は、実は VRML97 にも既にあったものばかりである。しかし、XML の文法を採用したおかげで、従来の VRML に比べると、学習する上で負担が少なくなっていることは間違いないだろう。プラグインも相対的に軽くなったことでもあるし、たとえ VRML97 相当の機能しか使わないとしても、X3D を試してみる価値はあるのではないかと思う。

しかし X3D で新しく拡張された部分ももちろん注目に値する。X3D で拡張された機能はたくさんあるのだが、ここでは NURBS を簡単に紹介する。NURBS は Non-Uniform Rational B-Spline の略であり、少ない代表点を指定することで滑らかな曲面を表現する手法の一つである。2D グラフィックスのベジエ (Bezier) 曲線や B スプライン曲線の 3D 版と考えれば良いだろう。

従来の VRML97 では、作成できる 3D 形状は基本的にポリゴン (多角形) の集まりで、しかもポリゴンが粗く見映えが悪かった。ゲームや映画の CG のような美しい 3D と比べるとガッカリするようなものだった。しかし、NURBS を利用すると滑らかな美しい曲面が表現できる。図 6 は NURBS で表現され

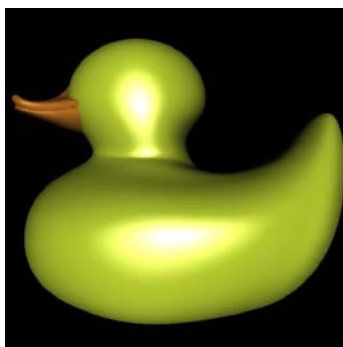


図 6: NURBS によるアヒル

たアヒル<sup>12</sup>である。この曲面を微小なポリゴン (IndexedFaceSet) だけで表現しようとする膨大なデータが必要になって、ブロードバンドと言えどもデータのダウンロードがたいへんである。NURBS を使用すると、このアヒルの場合 10 キロバイト程度のファイルサイズですむ。

もちろんこのような曲面のデータは、手作業でつくるのは難しい。3D モデリングソフトで作成してエクスポートするか、プログラムで生成することになるであろう。

ところで、X3D には (VRML97 も) 意外なことに 3D テキストを表現する命令がないようである<sup>13</sup>。ここで言う 3D テキストとは Word や PowerPoint

<sup>12</sup>URL は [http://www.web3d.org/TaskGroups/x3d/translation/examples/Nurbs/originals/nrbduck\\_0.wrl](http://www.web3d.org/TaskGroups/x3d/translation/examples/Nurbs/originals/nrbduck_0.wrl) である。

<sup>13</sup>Cortona や BS Contact VRML は独自の拡張として、3D テキストを表現する命令を持っている。

の“ワードアート”で作れるような、厚みを持ったテキストのことである。X3DのTextタグには厚さを指定する属性はない。3DテキストはX3DではIndexedFaceSetで表現しなければならず、現状では3Dモデリングソフトで作成する必要があるようだ。

## 6 おわりに

本稿を読んで、X3Dを試してみようという気になっていただければ幸いである。ところでX3DをWebページで使う場合にはいくつか注意しなければいけない点があると思われる。最後に、本稿を準備中にX3D/VRMLで遊んでいて気付いたことについて述べる。

まず、まだ現状ではハードウェアの3Dグラフィックスアクセラレータの性能にはバラツキがある。特にノートパソコンや省スペースパソコンなどは、3Dグラフィックスアクセラレータと呼べるほどのモノをつんでいないのが普通である。そのようなパソコンで表示する必要がある場合は、3D物体は“軽め”のものにとどめておくことが必要である。

また、3D空間を移動したりするための操作法はプラグインごとにも微妙な違いがあり、よっぽど慣れていないと操作にまごついてしまう。そのためユーザが不慣れな操作をしなくても良いように、視点(Viewpoint)をいくつかあらかじめ指定しておくことは重要であろう。プラグインを持っていない人やプラグインが存在しないプラットフォームを使用している人のために、サーバサイドで(前もって、もしくは動的に)2次元のフォーマットに変換しておくのも場合によっては有効な方法であろう。X3DはXMLなので、この目的のために、Apache Cocoonのようなさまざまな道具を利用することができると思われる。

XML関連の道具の充実ぶりを考えると、VRMLのXML化は当然の方向であると思われる。Windows以外にもLinux、Mac、さらには携帯電話などのプラットフォームでX3Dプラグインが一般に利用できるようになり、3Dグラフィックスの形式の標準として普及することを期待したいと思う。

## 参考文献

- [1] 青木 昌三「XMLプログラミングによるHTMLファイルの作成— シラバスファイルを例として—」年報 第9号 (香川大学情報処理センター, 2001年2月)
- [2] 香川 考司「SVGの勧め」年報 第10号 (香川大学情報処理センター, 2002年2月)



# Java による Web アプリ開発の勧め ( )

- Tomcat と Cocoon2 利用環境の構築 -

教育学部 青木 昌三

## はじめに

サーバサイドでのダイナミックな Web ページ作成を中心とする Web アプリケーション (Web サーバ上でのアプリケーション, Web アプリ) の開発手段の 1 つとして, Java によるアプローチがある。その中でも, 身近で手軽なソリューションの 1 つは, Tomcat と Cocoon2 を利用することであろう。

Tomcat は, Java によるダイナミックな Web ページ作成で用いられる, サーブレット (Servlet) やビーンズ (Beans), HTML ファイル中に Java コードを埋め込んだかたちの JSP (Java Server Pages) ファイルを利用するにあたって必要な Java Servlet/JSP 解釈・実行エンジン (Servlet コンテナ) で, アパッチ・ソフトウェア・ファウンデーション, Jakarta プロジェクト<sup>(\*)</sup>の開発によるものである。Tomcat の紹介記事を目にした<sup>(1)</sup>, また実際に JSP ファイルの記述による Web ページ作成を試みた方も少なくないであろう。

Cocoon2 (Cocoon バージョン 2) は, 同じ Jakarta プロジェクトによる, 「XML, XSLT 技術による XML 文書出版フレームワーク」<sup>(2)</sup> と呼ばれるものである。Cocoon2 では XML (eXtensible Markup Language) と XSLT (eXtensible Stylesheet Language Transformation) 併用による同一 XML 文書からの多様な Web ページ作成が図られることに加え, XML ファイル中に Java コードを埋め込んだかたちの XSP (eXtensible Server Pages) ファイルが利用でき, そこでは, Web アプリ作成に際して必要とされるロジック (Logic) とビュー (View) の分離が, ビューを主体とする JSP に比してより明確となっている。

本稿では, JSP や XSP の利用環境である Tomcat と Cocoon2 のインストールとその動作検証を中心に紹介する<sup>(3)</sup>。Java によるダイナミック Web ページ開発に関心ある方の Tomcat, Cocoon2 利用の出発点として頂ければ幸いである<sup>(4)</sup>。

読者の Java 利用環境が様々であろうと考えられるので, Tomcat と Cocoon2 のインストール等については, 次の OS 上において行っている。

Windows 系 : Windows2000(SP2), WindowsXP

Linux 系 : Red Hat Linux 7.3, Vine Linux 2.5, Linux MLD 6

FreeBSD 系 : FreeBSD-Release 4.6.2<sup>(5)</sup>

(\*1) <http://www.jakarta.org/>

(\*2) <http://xml.apache.org/cocoon/> 上の Documentation を参照。

(\*3) 本稿の内容は, 情報処理センター発行の Newsletter No.49 (June 16, 2003) における記事「Java による Web 開発の勧め - JSP, XSP 利用環境の構築 -」に訂正・加筆をおこなったものである。特に FreeBSD-Release4.9 における FreeBSD ネイティブ Java のテストと Cocoon-2.1.3 に焦点を当てた内容が加わっている。

(\*4) 本稿では JSP, XSP の構文や Cocoon2 の利用についての詳細には言及しない。JSP については文献(2), (3)を, また XSP や Cocoon2 については文献(4), (5)を参照されたい。関連サイトやイン

ストールした Tomcat , Cocoon 上のドキュメントやチュートリアルが参考となる。XML,XSLT の利用例については、文献(6)をも参照のこと。

(\*5)FreeBSD ネイティブな Java を用いた検証では、FreeBSD-Release 4.9 を用いた。

## Java2SDK のインストール インストール

Tomcat-4.x を利用するには Java2 の開発環境 SDK ( Software Development Kit , Java2SDK )が必要で、インストールが未だの場合は新しいバージョンの SDK をダウンロードし、Tomcat のインストールに先立ってインストールしておく。Java2SDK のダウンロードは次の URL から行う。

<http://java.sun.com/j2se/>

筆者の環境では既に j2sdk-1\_4\_0 がインストール済みであるが、用いたバイナリ版 SDK は次の通りである<sup>(\*)</sup>。

Windows の場合： j2sdk-1\_4\_0-win.exe

Linux の場合： j2sdk-1\_4\_0-linux-i386.bin または

j2sdk-1\_4\_0-linux-i386-rpm.bin ( RedHat 系 Linux )

FreeBSD の場合： j2sdk-1\_4\_0-linux-i386.bin<sup>(\*)</sup>

(\*1)本稿執筆時点の 2003 年 12 月現在の最新バージョンは、j2sdk-1\_4\_2\_03 である。新しいバージョンのものをインストールする場合には、本文中の該当箇所を適当に読みかえてほしい。

(\*2)FreeBSD-Release4.9 での FreeBSD ネイティブな Java の場合、j2sdk-1\_4\_2\_01 である。

### Windows でのインストール

j2sdk-1\_4\_0-win.exe の実行で、指示に従うことによりインストールが行える。筆者の場合、インストール先ディレクトリを c:\¥j2sdk1.4.0 としている。

### Linux でのインストール

j2sdk-1\_4\_0-linux-i386.bin を用いる場合は、インストールするディレクトリ ( 例えば /usr/local/ ) にファイルをコピーした後インストール先で次のように実行する。

```
# sh j2sdk-1_4_0-linux-i386.bin
```

Red Hat 系 Linux の場合には j2sdk-1\_4\_0-linux-i386-rpm.bin を用いると簡便である。適当なディレクトリでバイナリファイルを展開した後、rpm コマンドによりインストールが可能である ( 以下参照 )。インストール先ディレクトリは /usr/ java/ j2sdk1.4.0/ となっている。

```
# sh j2sdk-1_4_0-linux-i386-rpm.bin
# rpm -ivh j2sdk-1_4_0-linux-i386-rpm
```

### FreeBSD でのインストール

インストールの対象としたのは j2sdk-1\_4\_0-linux-i386.bin であり、Linux と同様にインストールしてよいのであるが、動作の安定性を確保する意味で ports コレクションからインストールを行った ( 下記参照 )<sup>(\*)</sup><sup>(\*)</sup><sup>(\*)</sup>。

```
vividy# cd /usr/ports/java/linux-jdk14
vividy# make
vividy# make install
```

j2sdk がインストールされたディレクトリは、/usr/local/linux-jdk1.4.0 である。

(\*1) 上記の ports コレクションからのインストールは FreeBSD-Release4.6.2 の場合であり、FreeBSD-Release4.9 で Linux Java を ports コレクションからインストールする場合には、/usr/ports/java/linux-sun-jdk14 からとなる。ここで用いられる j2sdk は j2sdk-1\_4\_2\_01-linux-i586.bin で、ソフトウェアライセンスに関わる制限から、個人の責任でダウンロードし、予め/usr/ports/distfiles/内にコピーしておくことが必要である。j2sdk のインストール先は /usr/local/linux-sun-jdk1.4.2\_01 である。

(\*2) FreeBSD では Linux バイナリ互換機能の設定が必要である。FreeBSD のインストールの際に「Linux エミュレータ機能を有効とするか」の質問に対して「Yes」としておけば、必要なファイルがインストールされると共に/etc/rc.conf ファイル内に

```
linux_enable="Yes"
```

が設定される。もし、インストール後にエミュレータ機能を設定するのであれば、ports コレクションから usr/ports/emulators/linux\_base をインストールし、/etc/rc.conf ファイルに linux\_enable="Yes" を書いておく。また、/etc/fstab ファイルに次の行を追加して、Linux プロセスファイルシステムを有効としておくことも必要である。

```
linproc          /compat/linux/proc          linprocfs        rw          0          0
```

この準備を整えた上で、j2sdk のインストールを行う。もし、準備を済ませずに ports コレクションから j2sdk のインストールを行おうとすると、途中で次のような警告が示されることとなる。

```
EROOR:
You have to have LINPROCFS mounted before stating to build of native JDK 1.4.1.
You may do it by
following set of commands:
# kldload linprocfs
and
# mount -t linprocfs linprocfs /compat/linux/proc
```

参考までに、/etc/fstab ファイルの内容を示しておく。

```
# See the fstab(5) manual page for important information on automatic mounts
# of network filesystems before modifying this file.
#
# Device          Mountpoint      FStype  Options      Dump    Pass#
/dev/ad0s1b      none            swap    sw           0       0
/dev/ad0s1a      /               ufs     rw           1       1
/dev/ad0s1e      /home           ufs     rw           2       2
/dev/ad0s1g      /usr            ufs     rw           2       2
/dev/ad0s1f      /var            ufs     rw           2       2
/dev/acd0c       /cdrom          cd9660  ro,noauto    0       0
proc             /proc           procfs  rw           0       0
linproc         /compat/linux/proc          linprocfs        rw          0          0
```

FreeBSD での Java 環境の設定については以下の文献が参考となる。

後藤 大地：「FreeBSD で構築する Java 環境」UNIX USER , 2001/2  
 , ソフトバンク パブリッシング

後藤 大地：「FreeBSD4.5-RELEASE の新機能と Java 環境」 UNIX USER , 2002/4  
 , ソフトバンク パブリッシング

(\*3) FreeBSD にネイティブな Java のインストールは、/usr/ports/java/jdk14/ において行う。FreeBSD-Release4.9 においてインストールを行ったが、以下のパッチセットファイルと JDK の

ソースコードファイル，更には，同時にインストールが行われる関係で，Linux Java 用の j2sdk-1\_4\_2\_01-linux-i586.bin ファイルを自己の責任においてダウンロードし，  
/usr/ports/distfiles/内に置いておくことが必要であった。

必要な2つのファイル：

パッチセットファイル   bsd-jdk14-patches-3.tar.gz

ソースコードファイル   j2sdk-1\_4\_1-src-scsl.zip

これらファイル入手先については，Makefile 内に記述されており，もしファイルが存在しないときは，ファイルの入手先が示されダウンロードするよう表示される。

FreeBSD ネイティブなjdkのインストール先は，/usr/local/jdk1.4.1であり，Linux Javaも同時に/usr/local/linux-sun-jdk1.4.2\_01/にインストールされる。尚，j2sdkのメイクにあたっては約2.5GBのディスク空き容量が必要であること，また，半日以上といった長時間（筆者が用いたようなCPUパワーが弱いPCの場合では約7時間程度）が必要であることに注意する。

## 環境変数の設定

環境変数 JAVA\_HOME に Java2SDK のインストール先ディレクトリを設定する。

### Windows2000/XP の場合

「コントロールパネル」 「システム」 「詳細」 「環境変数」により設定する。

### B シェル系の場合（例）

```
JAVA_HOME=/usr/java/j2sdk1.4.0 ; export JAVA_HOME
```

### C シェル系の場合（例）

```
setenv JAVA_HOME /usr/local/linux-jdk1.4.0
```

## Tomcat-4.1 のインストール

Tomcat-4.1 の2003年5月，テスト時点での Release Builds バージョンは4.1.24であった<sup>(\*)</sup>。  
サポートする Java Servlet/JSP の仕様は，それぞれ Java Servlet 2.3 と JSP1.2 である。

### Tomcat-4.1 の入手

Tomcat-4.1.24 のダウンロード元の URL は次の通りである<sup>(\*)</sup>。

<http://jakarta.apache.org/tomcat/>

各 OS によって，それぞれ次の Release Builds のバイナリ版を入手した。

Windows の場合： tomcat-4.1.24.exe ファイル<sup>(\*)</sup>。

Linux，FreeBSD の場合： tomcat-4.1.24.tar.gz または tomcat-4.1.24.zip ファイル。

(\*1)本稿執筆時点(2003.12)での最新バージョンは4.xで4.1.29，5.xで5.0.16である。5.xでサポートするJava Servlet/JSPの仕様はそれぞれ1ランク上がって2.4/2.0となっている。本文中のファイル名等は，実際にインストールするバージョンのものに読みかえてほしい。

(\*2)<http://www.meisei-u.ac.jp/mirror/apache/dist/Jakarta/>への接続が便利である。

(\*3)tomcat-4.1.24.zipファイルでもよいが，tomcat-4.1.24.exeはWindowsのインストーラ仕様なので，インストールが極めて簡便である。

## Tomcat-4.1 インストールの実際

### Windowsでのインストール

Windowsのインストーラ仕様にそったtomcat-4.1.24.exeを用いてインストールを行う。tomcat-4.1.24.exeのアイコンをダブルクリックすれば後は殆ど自動的である。Tomcatのインストールについては多くの文献で紹介されているので詳細はそれら文献に譲るが，Tomcat-4.1で加わったことを含め幾つかの場面を紹介しておこう。

\* インストールの際に J2SDK のインストール先が自動的に検出される ( 図 1 )。

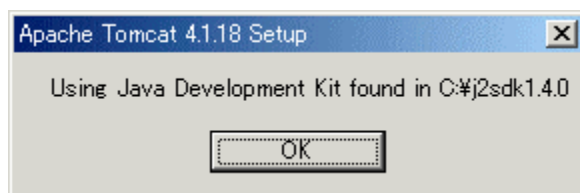


図 1 : 検出された J2SDK のインストールディレクトリ

\* オプションとして Windows2000/XP のサービスに登録が可能である。デフォルトではサービスに登録しない ( 図 2 )。

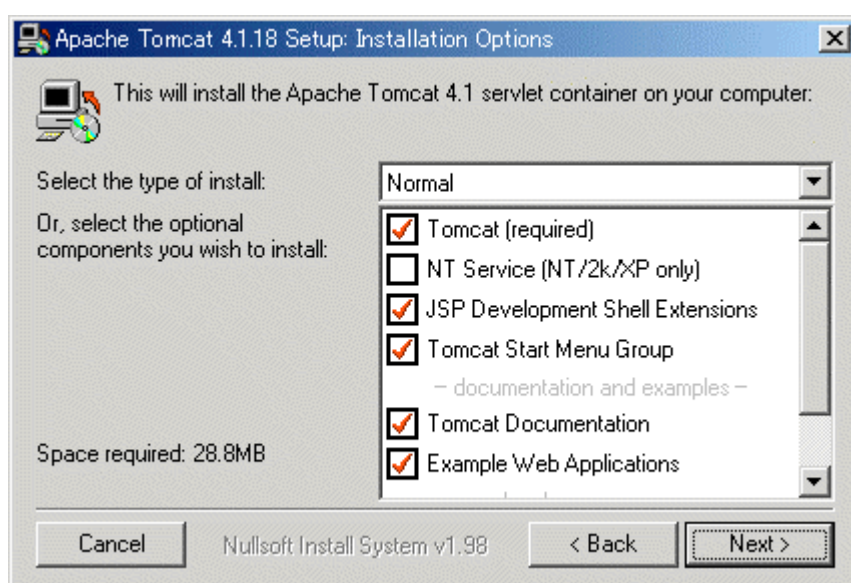


図 2 : オプション設定画面

\* Administrator としてログインする際のパスワードの設定画面が表示される ( 図 3 )。Administration Tool による Tomcat の各種設定 ( Tomcat-4.1 からサポート ) を行うにはログインパスワードが必要である。インストールの段階ではスキップしても差し支えない。

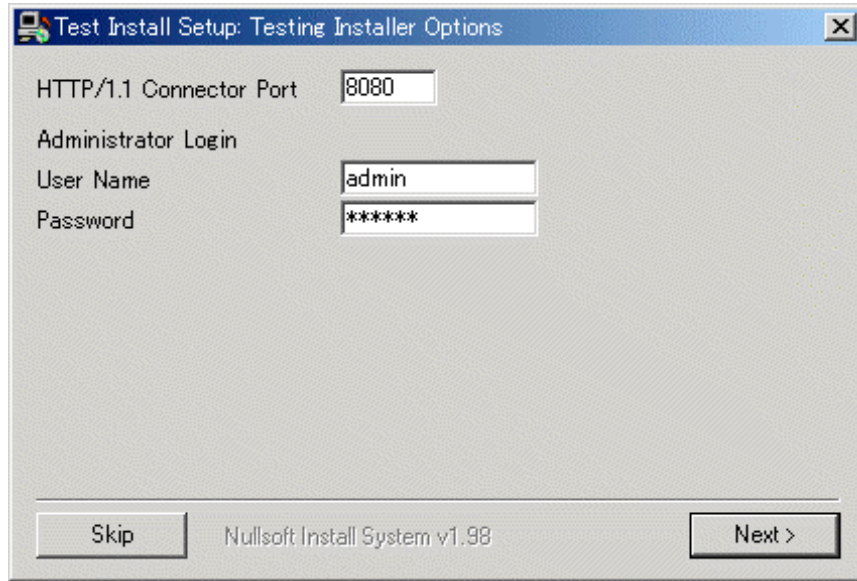


図 3 : ログインパスワード設定画面

\* インストール完了後，起動プログラム等は「スタート」「プログラム」に登録される（図 4）。

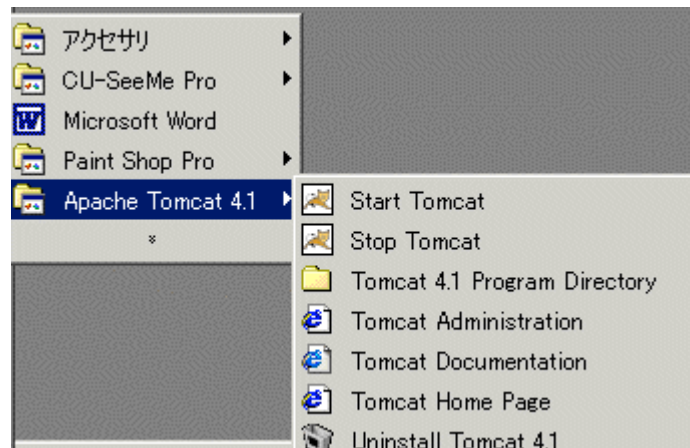


図 4 : 登録された起動プログラム等

#### Linux, FreeBSD でのインストール

tomcat-4.1.24.tar.gz をインストール先のディレクトリ（例えば，/usr/local/）にコピーし，以下のように tar コマンドで展開する。インストール先ディレクトリは，例えば，/usr/local/jakarta-tomcat-4.1.24 となる<sup>(\*)</sup>。

```

# tar zvxf tomcat-4.1.24.tar.gz          解凍・展開
vividy# cd jakarta-tomcat-4.1.24
vividy# ls
LICENSE                bin                    shared
README.txt             common                temp
RELEASE-NOTES-4.1.txt conf                    webapps
RELEASE-PLAN-4.1.txt  logs                  work
RUNNING.txt            server
vividy#

```

(\*1)FreeBSD では ports コレクションからのインストールが可能である。FreeBSD-Release4.9 でインストールされる Tomcat のバージョンは、  
 /usr/ports/www/jakarta-Tomcat4 : 4.1.27  
 /usr/ports/www/jakarta-tomcat5 : 5.0.10  
 である。関連するファイルは Makefile に書かれている fetch 先に存在しないが、  
<http://archive.apache.org/dist/Jakarta>  
 から入手可能である。make の際、/usr/local/jdk1.4.1 にネイティブな jdk がインストールされているかどうかチェックされるので、予め jdk をインストールしておくことが必要である。  
 インストール先はそれぞれ /usr/local/jakarta-Tomcat4.1 と /usr/local/jakarta-tomcat5.0 となっており、/usr/local/bin/ 内に起動ファイル tomcat41ctl, tomcatctl50ctl が用意される (以下参照)

```

sl-epson# ls
autopoint      libtool13      msgfilter      registervm     zipcloak
freetype-config libtoolize13   msgfmt         tomcat41ctl   zipgrep
funzip         msgattrib     msggrep       tomcat50ctl   zipinfo
gettext       msgcat        msginit       unregistervm zipnote
gettextize    msgcmp        msgmerge      unzip         zipsplit
gm4           msgcomm       msgunfmt      unzipsfx
gmake         msgconv       msguniq       xgettext
iconv         msgen         ngettext      xmlwf
javavm        msgexec       pkg-config    zip

```

この起動スクリプトによる Tomcat の起動では、たとえば HTTP 接続先ポートが 8180 などとなっていることに注意する。以下の URL をも参照してほしい。  
<http://www.emergence.jp/ci/documents/setup/setup21.ndoc.html>

### 環境変数 CATALINA\_HOME の設定

Tomcat のインストールが済んだならば、環境変数 CATALINA\_HOME にインストール先ディレクトリを設定する。設定方法は、前出の JAVA\_HOME の場合と同様である。参考までに FreeBSD での登録状況を示す。

```

vividy# printenv
===== 中略 =====
JAVA_HOME=/usr/local/linux-jdk1.4.0
CATALINA_HOME=/usr/local/jakarta-tomcat-4.1.24

```

### Tomcat の起動と終了

#### Windows での起動と終了

#### Tomcat の起動

「スタート」「プログラム」「Apache Tomcat4.1」「Start Tomcat」による。コマンドプロンプト画面で次のように表示される。

```

2003/03/22 14:08:42 org.apache.commons.modeler.Registry loadRegistry
情報: Loading registry information
2003/03/22 14:08:43 org.apache.commons.modeler.Registry getRegistry
情報: Creating new Registry instance
2003/03/22 14:08:43 org.apache.commons.modeler.Registry getServer
情報: Creating MBeanServer
2003/03/22 14:08:44 org.apache.coyote.http11.Http11Protocol init
情報: Initializing Coyote HTTP/1.1 on port 8080
サービス Tomcat-Standalone を起動します
Apache Tomcat/4.1.18
2003/03/22 14:08:48 org.apache.coyote.http11.Http11Protocol start
情報: Starting Coyote HTTP/1.1 on port 8080
2003/03/22 14:08:48 org.apache.jk.common.ChannelSocket init
情報: JK2: ajp13 listening on /0.0.0.0:8009
2003/03/22 14:08:48 org.apache.jk.server.JkMain start
情報: Jk running ID=0 time=0/16 config=C:¥Tomcat 4.1¥conf¥jk2.properties

```

### Tomcat の終了

「スタート」「プログラム」「Apache Tomcat4.1」「Stop Tomcat」による。Tomcat 終了の表示と共にコマンドプロンプト画面が消える。

Linux, FreeBSD での起動と終了

### Tomcat の起動

\$ CATALINA\_HOME/bin/startup.sh スクリプト実行により Tomcat が起動する(以下参照)

```

vividy# cd /usr/local/jakarta-tomcat-4.1.24/bin
vividy# ls
bootstrap.jar          jasper.bat            shutdown.sh
catalina.bat           jasper.sh             startup.bat
catalina.sh            jspc.bat              startup.sh
commons-daemon.jar     jspc.sh               tomcat-jni.jar
cpappend.bat           setclasspath.bat      tomcat.exe
digest.bat             setclasspath.sh       tool-wrapper.bat
digest.sh              shutdown.bat           tool-wrapper.sh
vividy# ./startup.sh
Using CATALINA_BASE:   /usr/local/jakarta-tomcat-4.1.24
Using CATALINA_HOME:   /usr/local/jakarta-tomcat-4.1.24
Using CATALINA_TMPDIR: /usr/local/jakarta-tomcat-4.1.24/temp
Using JAVA_HOME:       /usr/local/linux-jdk1.4.0
vividy#

```

### Tomcat の終了

\$ CATALINA\_HOME/bin/shutdown.sh スクリプトの実行による<sup>(\*)</sup>。

(\*) \$ CATALINA\_HOME/bin/ディレクトリ内の startup.bat , shutdown.bat は Windows で用いられる bat ファイルである。



## Tomcat の動作確認

Tomcat が正常にインストールされ動作するかどうかの確認が必要である。Tomcat は Web サーバ機能を備えており、デフォルトでは port 番号 8080 でアクセスが可能である。

### トップページへのアクセス

Web ブラウザで `http://localhost:8080/` にアクセスして以下の画面が表示されれば正常にインストールされている。

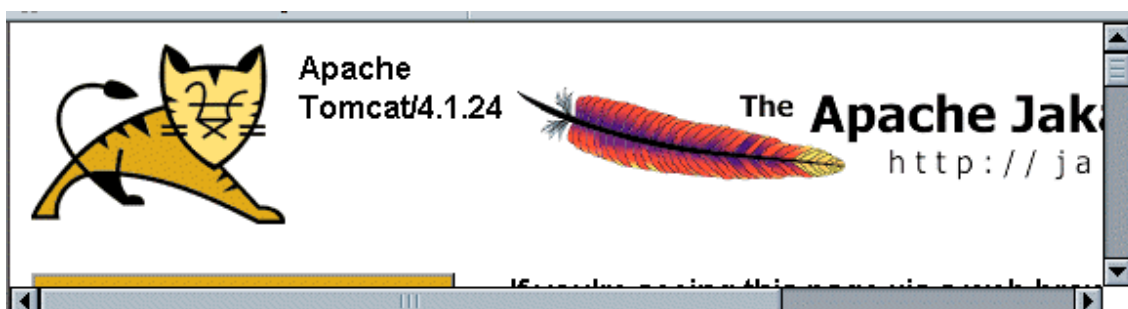


図 5 : Tomcat-4.1 のトップページ ( Windows2000 )

### Examples ページへのアクセス

幾つかのサンプルページにアクセスして、各 OS 上での Tomcat4.1 の動作を更に確認する。

### FreeBSD 上の Tomcat-4.1 にアクセス<sup>(\*)</sup><sup>(\*)</sup>

JSP Example である Date の実行 ( 図 6 参照 )

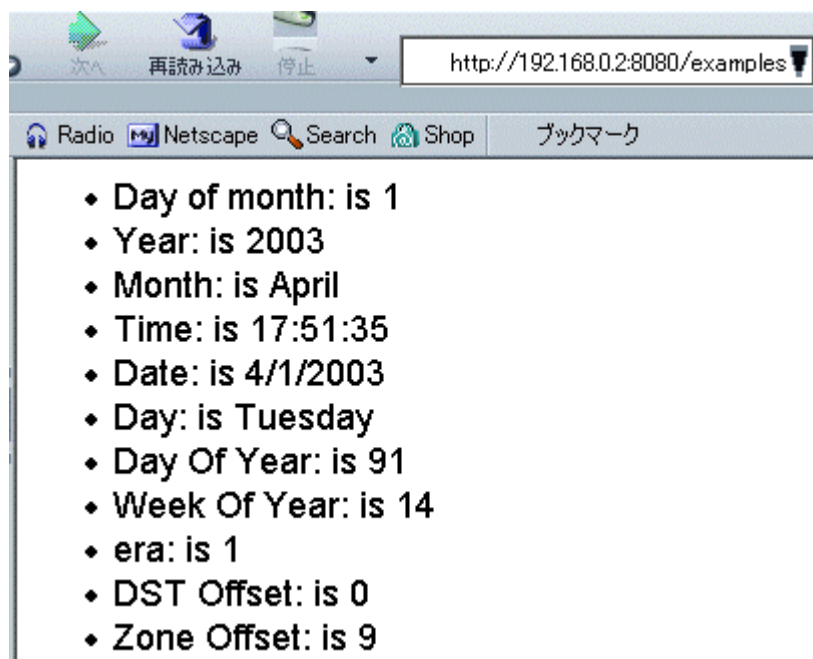


図 6 : JSP Example の DATE の実行画面 ( FreeBSD )

## Red Hat Linux 上の Tomcat-4.1 にアクセス

JSP Example である Checkbox の実行 ( 図 7 参照 )



図 7 : JSP Example の Checkbox の実行画面 ( Red Hat Linux )

## Vine Linux 上の Tomcat-4.1 にアクセス

Servlet Example である HelloWorld の実行 ( 図 8 参照 )



図 8 : Servlet Example の Hello World の実行画面 ( Vine Linux )

(\*1)Tomcat-4.0.3 の場合 ,Windows2000 ,Red Hat Linux ,Vine Linux ,Linux MLD 6 では Java2SDK として j2sdk-1.4.0 を用いて正常に動作したが ,FreeBS-Release4.6.2 では j2sdk-1.4.0 との組み合わせではエラーが生じ java2SDK パッケージとして j2sdk-1\_3\_1-linux-i386.bin を用いる必要があった。

Tomcat-4.1.24 では j2sdk-1.4.0 との組み合わせで各 OS とも (FreeBSD-Release 4.6.2 においても ,そしてまた ,FreeBSD-Release 4.9 では j2sdk-1\_4\_2\_03-linux-i586.bin との組み合わせにおいても ) エラーが生じることなく動作した。

(\*2)FreeBSD-Release4.9 上で ,

Tomcat として最も新しいバージョンである Tomcat-4.1.29 と Tomcat-5.0.16

jdk として FreeBSD ネイティブな jdk1.4.1 と linux-sun-jdk1.4.2\_01

(linux-sun では j2sdk-1\_4\_2\_01-linux-i586.bin を利用)

の組み合わせで Tomcat の動作確認を行った。Tomcat のインストールは ,ports コレクションからではなく ,tar.gz ファイルを直接展開する方法によっている。以下に結果を示しておく。

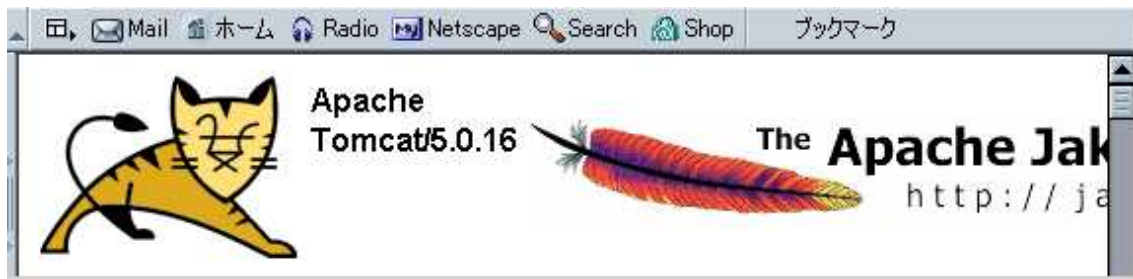
	FreeBSD ネイティブな jdk1.4.1	linux-sun-jdk1.4.2_01
Tomcat-4.1.29	正常に動作	正常に動作
Tomcat-5.0.16	正常に動作	core dumped エラー (注)

jdk として FreeBSD ネイティブな jdk1.4.1 を使い ,Tomcat5.0.16 を起動させた場合の様子を次に示す (以下の図をも参照)

```

sl-epson# pwd
/usr/local/jakarta-tomcat-5.0.16
sl-epson# ./bin/startup.sh
Using CATALINA_BASE:   /usr/local/jakarta-tomcat-5.0.16
Using CATALINA_HOME:   /usr/local/jakarta-tomcat-5.0.16
Using CATALINA_TMPDIR: /usr/local/jakarta-tomcat-5.0.16/temp
Using JAVA_HOME:       /usr/local/jdk1.4.1
sl-epson#
===== この間にブラウザからアクセスして動作を確認 =====
sl-epson# ./bin/shutdown.sh
Using CATALINA_BASE:   /usr/local/jakarta-tomcat-5.0.16
Using CATALINA_HOME:   /usr/local/jakarta-tomcat-5.0.16
Using CATALINA_TMPDIR: /usr/local/jakarta-tomcat-5.0.16/temp
Using JAVA_HOME:       /usr/local/jdk1.4.1
sl-epson#           //エラーを生じることなく正常に動作し、終了

```



図：Tomcat-5.0.16 のトップページ



図：JSP2.0 Examples 画面（Basic Arithmetic 等も正常に実行・表示）

jdk として linux-sun-jdk1.4.2\_01 を用いた場合，Tomcat-5.0.16 では jdk に core dumped エラーが生じ，正常に動作しなかった。このときの記録を示す。

```

sl-epson# pwd
/usr/local/jakarta-tomcat-5.0.16
sl-epson# ./bin/startup.sh
Using CATALINA_BASE:   /usr/local/jakarta-tomcat-5.0.16
Using CATALINA_HOME:   /usr/local/jakarta-tomcat-5.0.16
Using CATALINA_TMPDIR: /usr/local/jakarta-tomcat-5.0.16/temp
Using JAVA_HOME:       /usr/local/linux-sun-jdk1.4.2_01
sl-epson#
===== この間にブラウザから接続し動作を確認           core dumped error =====
sl-epson# ./bin/shutdown.sh

```

```

Using CATALINA_BASE:   /usr/local/jakarta-tomcat-5.0.16
Using CATALINA_HOME:   /usr/local/jakarta-tomcat-5.0.16
Using CATALINA_TMPDIR: /usr/local/jakarta-tomcat-5.0.16/temp
Using JAVA_HOME:       /usr/local/linux-sun-jdk1.4.2_01
Catalina.stop: java.net.ConnectException: Connection refused
java.net.ConnectException: Connection refused
    at java.net.PlainSocketImpl.socketConnect(Native Method)
    at java.net.PlainSocketImpl.doConnect(PlainSocketImpl.java:305)
    at
java.net.PlainSocketImpl.connectToAddress(PlainSocketImpl.java:171)
    at java.net.PlainSocketImpl.connect(PlainSocketImpl.java:158)
    at java.net.Socket.connect(Socket.java:452)
    at java.net.Socket.connect(Socket.java:402)
    at java.net.Socket.<init>(Socket.java:309)
    at java.net.Socket.<init>(Socket.java:124)
    at
org.apache.catalina.startup.Catalina.stopServer(Catalina.java:436)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.
java:39)
    at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAcces
sorImpl.java:25)
    at java.lang.reflect.Method.invoke(Method.java:324)
    at
org.apache.catalina.startup.Bootstrap.stopServer(Bootstrap.java:322)
    at
org.apache.catalina.startup.Bootstrap.main(Bootstrap.java:400)
sl-epson#

```

(注) Tomcat-5.0.16 の Linux 上での動作 :

FreeBSD-Release4.9 上で linux-sun-jdk1.4.2\_01 を用いた場合に正常に動作しなかったので (実は linux-sun-jdk1.4.2\_03 でも同様), Linux 上での動作検証を下記の状況下で行った (残念ながら 2003 年 12 月の時点では, PC の廃棄により, Red Hat Linux, Vine Linux が動作する環境が筆者の自宅から既に失われてしまっている)。

プラットフォーム: Linux MLD 6

jdk: j2sdk-1\_4\_0-linux-i386-rpm.bin と j2sdk-1\_4\_2\_03-linux-i586.bin

結果は, 2 つの jdk とともに, 正常に Tomcat が起動されて, トップページ, Examples ページが問題なく表示された。

## Cocoon2 のインストールと動作確認

### Cocoon2 ファイルの入手

ダウンロードしたファイルはテストを行った 2003 年 5 月の時点での最新バージョンであるバイナリ版の

cocoon-2.0.4-vm14-bin.zip, または cocoon-2.0.4-vm14-bin.tar.gz

である (vm14 は j2sdk-1.4.x 対応を意味している) <sup>(\*)</sup>。これらのファイルは, 次の URL からダウンロードできる。

<http://xml.apache.org/cocoon/>

(\*1)2003 年 12 月時点での最新バージョンは cocoon-2.1.3 である。

## インストール

### Windows2000 でのインストール

\* cocoon-2.0.4-vm14-bin.zip を適当なディレクトリ内で展開する。

\* 作成された cocoon-2.0.4 ディレクトリ内の install ファイルに書かれている指示に従って、同ディレクトリ内の cocoon.war ファイルを \$CATALINA\_HOME¥webapps 内にコピー・配置する。

以上でインストールは終了である。

### Linux, FreeBSD でのインストール

インストール手順は Windows の場合と全く同じである（以下参照）。

```
vividy# tar zxvf cocoon-2.0.4-vm14-bin.tar.gz
vividy# ls cocoon-2.0.4
CREDITS          KEYS             cocoon.war       legal
INSTALL         README           docs
vividy# cd cocoon-2.0.4
vividy# cp cocoon.war /usr/local/jakarta-tomcat-4.1.24/webapps/
vividy# cd /usr/local/jakarta-tomcat-4.1.24/webapps/
vividy# ls
ROOT             cocoon.war       manager.xml      webdav
admin.xml        examples         tomcat-docs
vividy#
```

### Cocoon2 の動作確認

Cocoon2 は Tomcat によりコンパイルされてサーブレットとして動作する。Tomcat を起動すると cocoon.war ファイルが展開され Cocoon2 の機能が利用可能となる。このとき、\$CATALINA\_HOME¥webapps/内には Cocoon2 のトップディレクトリとなる cocoon ディレクトリが作成される。この cocoon ディレクトリ内の構成を次に示す（図9参照）。

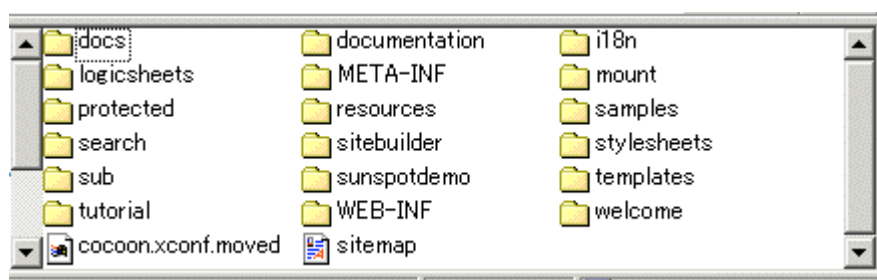


図9：\$CATALINA\_HOME/webapps/cocoon/内の構成

### Cocoon2 のトップページへのアクセス

Web ブラウザで <http://localhost:8080/cocoon/> にアクセスすることにより、次の Cocoon2 のトップページが表示されれば、Cocoon2 のセットアップが一応完了である（Cocoon-2.0.4 の場合）。



図 10 : Cocoon2 のトップページ

#### Windows2000/XP と各 Linux の場合

各 OS 上で、図 10 に示されるトップページが表示され、Cocoon2 が正常に機能していることが確かめられた。

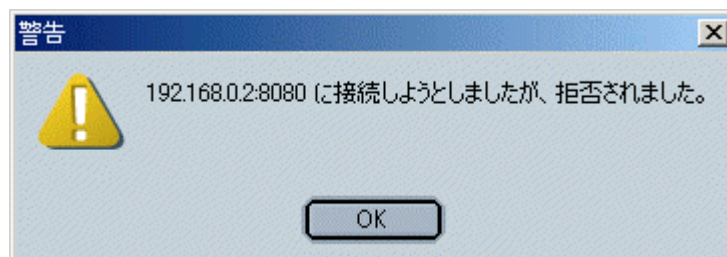
#### FreeBSD の場合

FreeBSD-Release4.6.2 上で jdk として j2sdk-1\_4\_0-linux-i386.bin を用いた場合、Cocoon-2.0.4 ではエラーが生じた<sup>(\*)</sup>。

ここでは、FreeBSD-Release4.9 上でのテスト結果を示しておこう（2003 年 12 月時点）。上述したように、同 FreeBSD 上で Tomcat が正常に動作したのは、FreeBSD ネイティブな jdk1.4.1 を用いた Tomcat-4.1.29 と Tomcat-5.0.16、linux-sun-jdk1.4.2\_01 を用いた場合の Tomcat-4.1.29 である。Cocoon には Cocoon-2.0.4 を用いる。

	FreeBSD ネイティブな jdk1.4.1	linux-sun-jdk1.4.2_01
Tomcat-4.1.29	Cocoon-2.0.4 は正常に動作	エラー <sup>(*)</sup>
Tomcat-5.0.16	HTTP Status 404 のエラー <sup>(*)</sup>	テスト無し

(\*1)FreeBSD-Release4.6.2 におけるエラーは、Cocoon2 トップページへのアクセスのみならず http://localhost:8080/への Tomcat のオープニング画面アクセスに際しても起こるもので（Tomcat そのものが正常に機能しない。下図参照）、cocoon.war が解凍/コンパイルされた時点にまで遡るようである。Cocoon-2.0.4 に替えて Cocoon-2.0.3 を試みたが、同様のエラーであった。



因みに、Cocoon2 の設定を取り除いて Tomcat 機能のみとすると、エラーは除外される。Cocoon2 利用には、別の J2SDK が必要である。参考までに、このとき生じたエラーログを次に示す。



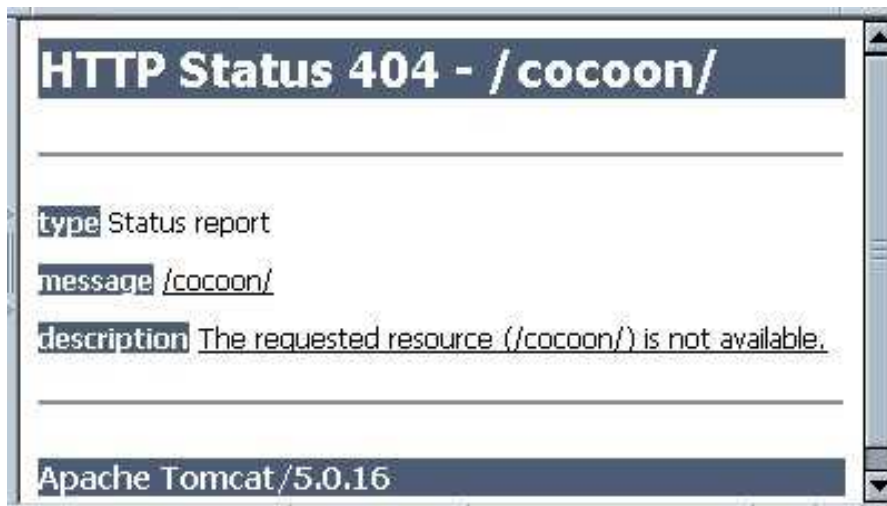
```

vividy# cat hs_err_pid281.log
Unexpected Signal : 11 occurred at PC=0x2A12FD8F
Function=[Unknown.]
Library=(N/A)
NOTE: We are unable to locate the function name symbol for the error
      just occurred. Please refer to release documentation for possible
      reason and solutions.
Current Java thread:
Dynamic libraries:
Can not get information for pid = 281
Local Time = Sun Apr  6 07:45:33 2003
Elapsed Time = 116
#
# HotSpot Virtual Machine Error : 11
# Error ID : 4F530E43505002D3
# Please report this error at
# http://java.sun.com/cgi-bin/bugreport.cgi
#
# Java VM: Java HotSpot(TM) Client VM (1.4.0-b92 mixed mode)
#

```

(\*2) 上述の(\*1)と同様のエラー

(\*3) HTTP Status 404 のエラーは URL が該当しない、要求されるメタ情報がない (Not Found, RFC 2068) ときのエラーである (下図を参照)。 Cocoon-2.0.4 の cocoon.war ファイルも正常にコンパイル・実行され cocoon ディレクトリも作成されて、Tomcat-5.0.16 そのものは正常に動作しているにも拘らずのエラーである。 Cocoon-2.0.4 が Tomcat-5.0.16 に対応していないためと思われるが、 Cocoon-2.0.4 の設定ファイルを変更することにより対応が可能となるのかもしれない。 Linux での状況を確認するため Linux MLD 6 でテストを行ったが、 Tomcat-5.0.16、 Cocoon-2.0.4 の組み合わせではやはり同様のエラーとなった。 尚、 Tomcat-5.0.16 と Cocoon-2.1.3 の組み合わせでは正常に Cocoon ページにアクセスが可能である。 これについては後述の「 Cocoon-2.1.3 について」において簡単に紹介する。



#### Examples ページへのアクセス

Cocoon トップページへのアクセスによって各 OS 上の Cocoon2 の動作を一応確認したが、更に Examples ページ/cocoon/hello.\* にアクセスして各 OS における Cocoon2 の動作状況を確認める<sup>(\*)</sup>。用いた Web ブラウザは、Vine Linux と Red Hat Linux 上の Cocoon2 へのア

クセスについては Windows2000 にインストールした Netscape 7.0.1 , Linux MLD 6 上の Cocoon2 については同 Linux に予めインストールされていた Galeon 1.0.3 である。

#### hello.html へのアクセス

Vine Linux 上での Cocoon2 の動作確認のため

<http://192.168.0.2:8080/cocoon/hello.html>

へのアクセスで得られた画面 ( 図 11 参照 )



図 11 : /cocoon/hello.html へのアクセス画面 ( Vine Linux )

#### hello.pdf へのアクセス

Red Hat Linux 上での Cocoon2 の動作確認のため ,

<http://192.168.0.2:8080/cocoon/hello.pdf>

へのアクセスで得られた画面 ( 図 12 参照 )

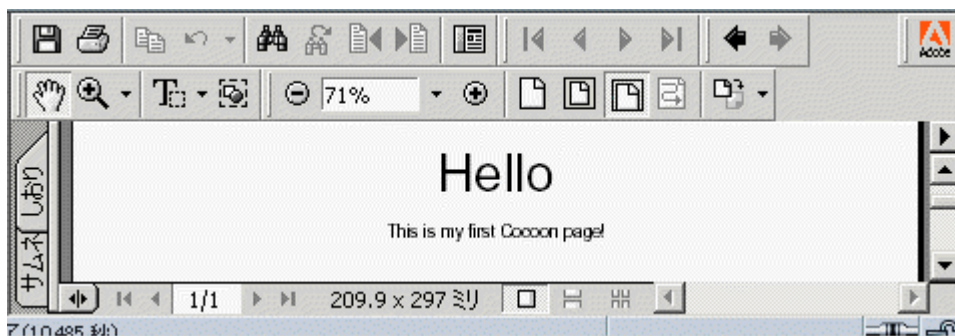


図 12 : /cocoon/hello.pdf へのアクセス画面 ( Red Hat Linux )

#### hello.svg へのアクセス

Linux MLD 上での Cocoon2 の動作確認のため ,

<http://localhost:8080/cocoon/hello.svg>

へのアクセスで得られた画面 ( 図 13 参照 )<sup>(\*)</sup>。



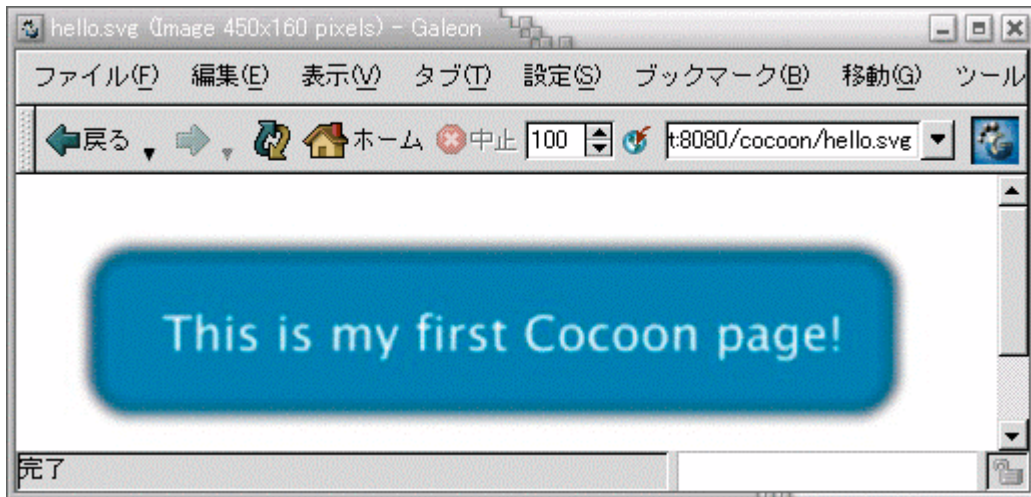


図 13 : /cocoon/hello.svg へのアクセス画面 (Linux MLD)

### /xsp/hello へのアクセス

FreeBSD-Release4.9 上での Cocoon2 の動作確認のため、

<http://192.168.0.2:8080/cocoon/xsp/hello.pdf>

へのアクセスで得られた画面 (図 14 参照)。ここでは、FreeBSD ネイティブな jdk1.4.1、Tomcat-4.1.29 が用いられている。



図 14 : /cocoon/xsp/hello へのアクセス画面 (FreeBSD)

(\*1)/cocoon/hello.html, hello.pdf, hello.svg 作成の出発点となるソースファイルは hello-page.xml ファイルである。Cocoon2 では同じ xml ファイルから出発して XSLT の記述により様々な異なるフォーマットのファイルを作成できる。hello-page.xml の内容は次のように簡単なものである。

```
<?xml version="1.0"?>
<!-- CVS: $Id: hello-page.xml,v 1.1 2002/03/27 10:09:31 nicolaken Exp $ -->
<!DOCTYPE page [
  <!ELEMENT page (title?, content)>
  <!ELEMENT title (#PCDATA)>
  <!ELEMENT content (para+)>
  <!ELEMENT para (#PCDATA)>
]>
<page>
  <title>Hello</title>
  <content>
    <para>This is my first Cocoon page!</para>
  </content>
</page>
```

(\*2)Galeon 1.0.3 では hello.svg を表示できたが ,hello.pdf は対応していないとして表示できなかった。  
Linux MLD 6 にインストール済みの Mozilla-0.9.8 ブラウザは , hello.svg を表示するとともに ,  
hello.pdf をも xpdf を用いて表示した。

もう一つの実例 - SVG データからの JPEG , PNG 画像表示 -

これまでは Cocoon2 に予め準備されている Examples ページへのアクセス状況を紹介したが ,最後に SVG(Scalable Vector Graphics)データをもとにした JPEG , PNG 画像の表示例を紹介しておこう ( 図 15 )。用いた SVG データは本学工学部の香川 考司先生から提供していただいたものであり<sup>(7)</sup> , 図 15 は , Windows2000 にインストールされた Cocoon2 に同 OS 上の Netscape 7.0.1 ブラウザからアクセスして得られたものである。

示した例は , 同じ1つの XML データから異なるフォーマットのデータファイルを出力することができる Cocoon2 の特徴的な機能の1つの良い例となっている。Cocoon2 が「XML , XSLT 技術による XML 文書出版フレームワーク」と呼ばれる所以である。



図 15 : SVG データの JPEG , PNG 画像表示

Cocoon-2.1.3 について

2003 年 12 月時点での最新バージョンである Cocoon-2.1.3 について紹介する。

Cocoon-2.1.3 はソースで提供されている ( <http://xml.apache.org/cocoon/> )

cocoon-2.1.3-src.tar.gz 或いは cocoon-2.1.3-src.zip

FreeBSD , Linux でのインストール或いは war ファイル作成は容易で , 展開して得られる INSTALL.txt ファイルに従って , 例えば , 次のように実行すればよい。

```
# tar zxvf cocoon-2.13-src.tar.gz
# cd cocoon-2.3.1/
# setenv JAVA_HOME /usr/local/jdk1.4.1
# ./build.sh
(war ファイルの作成では , ./build.sh war)
===== これで , cocoon-2.1.3 を実行することができる =====
===== 以下は , cocoon-2.1.3 を Tomcat-5.0.16 で用いる場合 =====
# cp ./build/cocoon-2.1.3/cocoon.war /usr/local/jakarta-tomcat-5.0.16/webapps/
```

## Cocoon-2.1.3 の実行

Cocoon-2.1.3 は Tomcat を利用せずとも実行可能である(この場合は cocoon.war ファイルの作成は必要ない)。cocoon-2.1.3 ディレクトリにおいて次のように実行し、<http://localhost:8888/>にブラウザでアクセスすればよい。

```
# ./cocoon.sh servlet
```

以下に、FreeBSD-Release4.9 (アドレス 192.168.0.8) 上で実行した Cocoon-2.1.3 にアクセスして得られた画面を示す(図 16, 17, 18 参照)(\*)。



図 16 : Cocoon-2.1.3 のオープニング画面 ( <http://192.168.0.8:8888/> )



図 17 : <http://192.168.0.8:8888/samples/hello-world/>へのアクセス画面



図 18 : <http://192.168.0.8:8888/samples/i18n/simple.xml> へのアクセス画面

## Cocoon-2.1.3 の Tomcat での実行

Cocoon-2.0.4 は Tomcat-4.1.29 上では正常に動作したが、Tomcat-5.0.16 上では上述したように、オープニングページ等へのアクセスに問題を残している。それでは Cocoon-2.1.3 はどうであろうか。FreeBSD-Release4.9 を用いてのテストでは (jdk は FreeBSD にネイティブな jdk1.4.1)、Cocoon-2.1.3 は Tomcat-4.1.29、Tomcat-5.0.16 上で共に問題なく動作し、cocoon ページへのアクセスも正常であった。これまでの FreeBSD-Release4.9 でのテスト結果をまとめると次のようになる。

	Cocoon-2.0.4	Cocoon-2.1.3
Tomcat-4.1.29	正常にアクセス可能	正常にアクセス可能
Tomcat-5.0.16	HTTP Status 404 のエラー	正常にアクセス可能 <sup>(*)</sup>

参考までに、Tomcat-5.0.16 上での Cocoon-2.1.3/Samples にアクセスした様子を示しておく (図 19 参照)。ポート番号が 8888 ではなく 8080 であることに注意する。

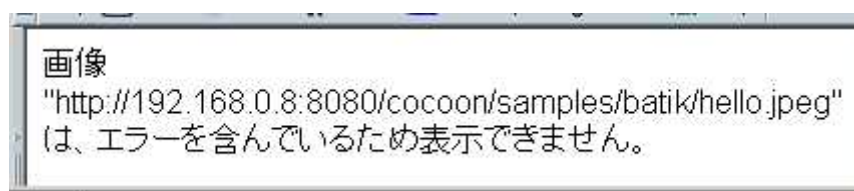


図 19 : <http://192.168.0.8:8080/cocoon/samples/>へのアクセス画面

(\*1)2003 年 9 月時点でのバージョン Cocoon2.1 において、Cocoon は「XML Web Development Framework」であるとなっている。以下は、<http://cocoon.apache.org/index2.html> からの引用である。

*The release of the long-awaited 2.1 version of Cocoon on August 13th marks the transition from a publishing-oriented XML/XSLT server engine towards a componentized XML-based web application development framework.*

このことを示す様に、Samples ページには様々な例がリストアップされアクセス可能となっている。未完成でエラーが表示される例も無くはないが (下図の例参照)。



例えば Excel のスプレッドシートフォーマット Hello ページなど (図参照)、各種フォーマットに対応するページを与える Web アプリ開発環境を目指していることが強く伺われる。



(\*2)Cocoon-2.1.3 が Tomcat-5.0.16 上で正常に動作することについては ,Linux MLD 6 においても確かめられた (j2sdk-1\_4\_0-linux-i386-rpm.bin と j2sdk-1\_4\_2\_03-linux-i586.bin の jdk を利用 )

## おわりに

Tomcat と Cocoon2 の利用環境の構築方法について紹介した。一読して頂ければ , Windows2000/XP , Linux , FreeBSD の各 OS 上で , Tomcat と Cocoon2 を利用できる環境を迷うことなく整えることができるものと思っている。

XML と XSLT を用いた Web ページ作成について紹介してから早や 3 年近くが経過した<sup>(6)</sup>。その利用が強く期待されながらも今一步と思われた XML , XSLT であったが , Cocoon2 の登場によりダイナミックな Web アプリ作成の有力なテクノロジーの 1 つとして再浮上してきた感じがする。Cocoon2 を用いて幾つかの Web ページ作成を行って得た実感である。

本稿では , 実際の Web アプリ作成を通しての Cocoon2 の素晴らしさやその威力については触れていない。それらについてはあらためて別の機会に紹介したい。

最後に , SVG データを快く提供くださった本学工学部の香川 考司先生に心からお礼申し上げます。

## 参考文献

- (1)青柳 龍也 :「Web アプリケーション開発の手引き」 JavaWorld 2002 April ,  
鯨崎 義之 :「超入門 ! Tomcat」 JavaWorld 2002 November ,  
鯨崎 義之 :「Tomcat 最新活用術」 JavaWorld 2003 January ,  
アイ・ディ・ジー・ジャパン
- (2)荒井 美千子 :「JavaServer Pages (1)~」 UNIX MAGAZINE 2001.3~ , アスキー
- (3)菊田 英明 :「SE・プログラマ スタートアップテキスト JSP [基礎]」  
2002.2 , 技術評論社 ( ISBN 4-7741-1340-9 )
- (4)坂上 徹 他 :「Apache Cocoon2 XML による Web サイト構築とコンテンツ管理」  
2002.9 ソフト・リサーチ・センター ( ISBN 4-88373-173-1 )
- (5)ラジヨス・モチャル :「Cocoon2 の威力を知る」 JavaWorld 2003 April ,  
アイ・ディ・ジー・ジャパン
- (6)青木 昌三 :「XML プログラミングによる HTML ファイルの作成 - シラバスファイル为例として - 」 香川大学情報処理センター年報 第 9 号 ( 2001 年 2 月 )
- (7)香川 考司 :「SVG の勧め」香川大学情報処理センター 年報 第 10 号 ( 2002 年 2 月 )



# Java による Web アプリ開発の勧め ( )

## - Cocoon2 利用による Web ページ作成 -

教育学部 青木 昌三

### はじめに

Newsletter No.49 において、JSP(Java Server Pages) や XSP(eXtensible Server Pages) の利用環境である Tomcat と Cocoon2 について、身近なプラットフォーム OS 上でのインストール方法とその動作検証を紹介した<sup>(1)</sup>。

Java Servlet/JSP 解釈・実行エンジンである Tomcat 上での JSP ファイル記述による Web ページ作成については既に馴染み深いものとなっており、本稿では、「XML, XSLT 技術による XML 文書出版フレームワーク」<sup>(\*)1</sup>であると言われる Cocoon2 を利用した Web ページ作成について、具体例を示しつつ紹介する<sup>(\*)2</sup>。

採りあげて紹介している Web ページの題材は、シラバスの表示<sup>(2)</sup>や休講通知の表示といったごく身近なテーマを念頭に筆者が作成した断片的なパーツを中心としており、Cocoon2 による Web ページ作成のほんの一部分でしかない<sup>(\*)3</sup>。然しながら、Cocoon2 の威力や面白さを感じとるにはじゅうぶんであらうと思われる。筆者が直面した問題の解決法についても触れている。

(\*1)<http://xml.apache.org/cocoon/>上の Documentation を参照。文献(1)で紹介した Cocoon2 は cocoon2.04 であったが、2003 年 9 月での最新バージョン cocoon2.1 では、Cocoon は「XML Web Development Framework」であるとなっている。<http://cocoon.apache.org/index2.html> から以下に引用する。

*The release of the long-awaited 2.1 version of Cocoon on August 13th marks the transition from a publishing-oriented XML/XSLT server engine towards a componentized XML-based web application development framework.*

(\*2)本稿は、香川大学情報処理センターの Newsletter No.50 (September 19, 2003)における記事「Java による Web 開発の勧め ( ) - Cocoon2 利用による Web ページ作成 -」に加筆等を行ったものである。各 OS での Tomcat や Cocoon2 のインストール等については、文献(1)及び同掲記事「Java による Web 開発の勧め ( ) - Tomcat と Cocoon2 利用環境の構築 -」を参照されたい。

(\*3)Cocoon2 を用いての Web 作成については文献(3)、(4)の一読をお勧めする。インストールした Cocoon2 上のドキュメントやチュートリアルも参考となる。サンプルページについての設定ファイルを吟味することも、Cocoon2 によるプログラミング理解にとって有益である。

### Cocoon2 における処理の流れ

Cocoon2 は Tomcat 上で Servlet として動作するプログラムであるが、XML(eXtensible Markup Language)ファイルや XSP ファイルがどのように処理されて最終的な Web ページが得られるのか、Cocoon2 での一連の処理の流れを大雑把に眺めておこう<sup>(\*)1</sup>。

(\*1)詳しくは文献(3)、(4)や下記 URL 記事を参照してほしい。

<http://xml.apache.org/cocoon/index.html>, <http://xml.apache.org/cocoon/overview.html>

#### 処理の流れ - パイプライン(Pipeline) -

Cocoon2 では、パイプラインと呼ばれる一連の処理の中で XML や XSP ファイルが順次コ

ンポーネントと呼ばれるプログラムで処理されつつ最終的にはブラウザ等で表示されるページの出版へと至るプロセスをとる。パイプラインは大きく3段階の処理プロセスからなり、それぞれのプロセスに対応する様々なコンポーネントプログラムの実装により構成されている。

#### Generator

パイプラインの出発点となるプロセスを担当するコンポーネントで、入力されたファイルを読み取り、SAX(Simple API for XML)イベントをパイプラインの下流に送り出す。Generator コンポーネントには入力ファイルそれぞれに対応するものが用意されており、XML ファイルを入力として受け取る FileGenerator、XSP ファイルを受け取る SeverPagesGenerator 等がある。

#### Transformer

パイプライン処理プロセスの中間点を担うコンポーネントで、Generator から送られるXML ドキュメントやSAX イベントを別のXML ドキュメントやSAX イベントに変換する役目を果たす。代表的なものに、Generator から送られた SAX イベントを XSLT(eXtensible Stylesheet Language Transformation)ファイルに基いて処理し、新たな SAX イベントに変換して次のプロセスコンポーネントに送り出す、デフォルト指定の TraxTransformer(XSLT Transformer)がある。

#### Serializer

パイプラインの最終点となるプロセスを担当するコンポーネントで、Transformer からの SAX イベントを受け取って(ブラウザ等で)表示できるフォーマットのファイルを生成する。最も代表的なものは HTML ファイルを生成する HTMLSerializer で、他に XML ファイルを生成する XMLSerializer、PDF ファイルを生成する FOPSerializer(PDF)等が用意されている。

パイプラインでの一般的な処理の流れを図示すると次のようになる(図1参照)。尚、Generator と Serializer コンポーネントによる処理は共に必須であるが、Transformer コンポーネントによる処理については、無くてもよいし、また2つ以上の Transformer による段階的な処理を経ても別段構わないこととなっている。

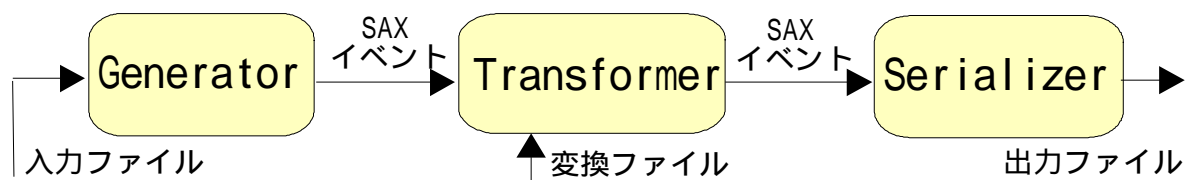


図1：パイプラインでの処理の流れ

#### パイプラインの記述 - サイトマップファイル -

パイプラインでの処理の流れを具体的に記述するのがサイトマップファイル(sitemap.xmap)である。サイトマップファイルでは、用いる入力ファイル、変換ファイル、出力ファイルとそれらの処理に係わるコンポーネントの関係が、処理の流れに沿って XML

形式で記述される。

サイトマップ記述において用いられるのが、Matcher コンポーネントや Selector コンポーネント等である。上記の図 1 では直線的なパイプライン処理の流れを示したが、パイプラインは複線的であって一向に構わない。例えば、Selector コンポーネントを用いて条件により分岐するなど、複線的なパイプライン処理をサイトマップ中に記述することも可能である。

また、サイトマップファイルでは、最終的に出力されるファイルとそのファイルへのアクセスに用いられる URL との関連付けがなされる。この関連付けは自由度に富んでおり、ファイルへのアクセス用 URL がファイルのリソース名そのものである必要が全く無い点が大きな特徴となっている。

以下に、サイトマップの管理、Matcher コンポーネントを用いたサイトマップでの記述について、具体例を用いて示すこととしよう<sup>(\*1)</sup>。

(\*1) Cocoon2 を利用するにおいては、入力ファイルとしての XML ファイルや XSP ファイル、変換ファイルとしての XSLT ファイルの記述は勿論大事であるが、サイトマップファイルについて理解し習熟することもまた大切なのである。

#### サイトマップ管理 - サイトマップのマウント -

サイトマップにおける出力ファイルリソース名と URL の関連付けは、Web ページ管理、Web サイト管理そのものである。ところで、各 Web ページに関連するファイルはメンテナンスの容易さなどから、異なるフォルダ内に分割されて置かれるのが大方であろう。とすれば、1 つのサイトマップで全ての Web ページを管理するよりは、サイトマップもまた Web ページ毎に分割して、関係する各フォルダ、サブフォルダに置き管理すると便利である。

分割されたサイトマップを階層構造的に関連付け、結果として 1 つの体系だったサイトマップとする機構がサイトマップのマウントと呼ばれるものである。マウントされた各サイトマップは Cocoon2 実行時には 1 つのサイトマップとしてコンパイルされ、Servlet としてコンテナにリロードされる。

#### フォルダ構造

今回紹介する Web ページ作成にあたってのフォルダ構造は図 2 の通りである。図 2 のフォルダ構成では、ipc01 ~ ipc10 フォルダが各 Web ページ関連のファイル群を置くためのものとなっている。

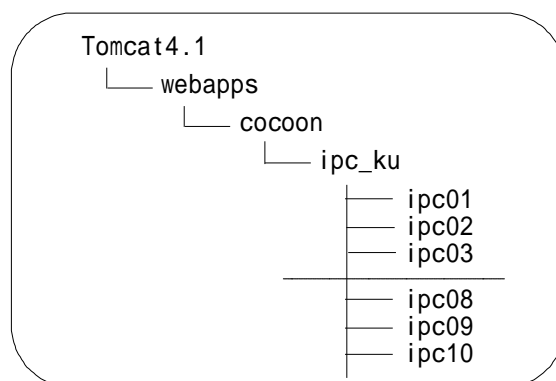


図 2 : Web ページ作成でのフォルダ構造



## サイトマップファイルのマウント

サイトマップファイルの中心となるのは、インストール時に配置されている cocoon フォルダ内の sitemap.xmap ファイルである。ipc\_ku フォルダ内、ipc01 ~ ipc02 内にそれぞれサブサイトマップファイル sitemap.xmap を置くこととする。サイトマップファイルをマウントする記述は次のようである。尚、マウントに必要な記述については文献(3)をも参照されたい。

cocoon¥sitemap.xmap ファイルへの追加記述内容：

```
<!-- mount ipc_ku sample pages -->      コメント
<map:match pattern="ipc_ku/**">
<map:mount check-reload="yes" src="ipc_ku/" uri-prefix="ipc_ku"/>
</map:match>
```

上記の追加記述は、<map:pipeline>タグと</map:pipeline>タグの間に記述されねばならない。記述場所としては、オリジナルの sitemap.xmap 中の次に示す個所のあたりでよいであろう。

```
<!-- pipeline mounting samples sitemaps -->
<map:pipeline>
  <!-- protected webapp example pipeline -->
===== 中略 =====
  <map:match pattern="samples/**">
    <map:mount check-reload="yes" src="samples/" uri-prefix="samples"/>
  </map:match>
===== この個所に追加記述する =====
</map:pipeline>
```

cocoon¥ipc\_ku¥sitemap.xmap ファイルへの記述内容：

例えば cocoon¥samples¥sitemap.xmap ファイルを cocoon¥ipc\_ku フォルダにコピーした上で、ファイルの終わりに近い部分に下記に示す内容を追加記述する<sup>(\*)</sup>。

```
<!-- samples automount -->      もとからあるコメント行
<!-- <map:match pattern="*/**">   もとからあったここからの3行をコメントとする
  <map:mount uri-prefix="{1}" src="{1}/" check-reload="yes"/>
  </map:match> -->
<!-- My Sample, 2003.4.10 -->   コメントなので書く必要なし
  <map:match pattern="ipc*/**">   ここからの3行</map:match>までを書き加える
  <map:mount uri-prefix="ipc{1}" src="ipc{1}/" check-reload="yes"/>
  </map:match>
<map:handle-errors>
===== 中略 =====          この部分は残しておいて差し支えない
</map:handle-errors>
```

cocoon¥ipc\_ku¥ipc01 ~ ipc10¥sitemap.xmap ファイル：

マウントについての記述は特に必要としないので、ipc01 ~ ipc10 各フォルダに、例えば cocoon¥samples¥sitemap.xmap ファイルをコピーした上で特に必要のない記述を削除しておけばよい<sup>(\*)</sup>。

(\*) コピーして得られた sitemap.xmap ファイルで

\* 先頭行から </map:components> の行までは残す。

\* <map:pipelines>行からファイルの最後の行</map:sitemap>までを残す。

\* 他の行は削除しても差し支えない。

<map:pipeline>行から</map:pipeline>行までの間には、作成する Web ページに関わる URL とファイルリソース名との関連付け等の事項を記述する。作成する Web ページに無関係な記述は削除して構わない。

### サイトマップ記述の具体例

サイトマップのマウント設定検証をも兼ね、簡単な Web ページ作成を通して Web ページへのアクセスに関わるサイトマップ記述の例を示そう。

Web ページは [http://localhost:8080/cocoon/ipc\\_ku/ipc03/hello](http://localhost:8080/cocoon/ipc_ku/ipc03/hello) にアクセスして得られる次のようなものであり (図 3 参照) <sup>(\*)1)</sup> , cocoon¥ipc\_ku¥ipc03 内に置かれた 2 つのファイル、hello.xml ファイル <sup>(\*)2)</sup> と hello.xsl ファイル <sup>(\*)3)</sup> から構成されている。



図 3 : “ Hello World 3 ” を表示するページ

ここで、cocoon¥ipc\_ku¥ipc03 はある Web ページ作成のためのフォルダであり、そこには主たるページのためのファイル群が置かれている。サイトマップ記述にあたっては次の方針を採ることとしよう。

[http://localhost:8080/cocoon/ipc\\_ku/ipc03/](http://localhost:8080/cocoon/ipc_ku/ipc03/) のアクセスに対しても “ Hello World 3 ” を表示する。

[http://localhost:8080/cocoon/ipc\\_ku/ipc03/index.html](http://localhost:8080/cocoon/ipc_ku/ipc03/index.html)(htm)等のアクセスに対しても、同様に “ Hello World 3 ” を表示する。

例えば、[http://localhost:8080/cocoon/ipc\\_ku/ipc03/abcdef.xyz](http://localhost:8080/cocoon/ipc_ku/ipc03/abcdef.xyz) のように、サイトマップ内に明示的に記述されている URL にマッチしない全てのランダムなアクセスに対して、“ Hello World 3 ” を表示する。

このときのサイトマップファイルへの記述例を以下に示す。

```
<!-- ===== Hello ===== -->      コメント
<map:pipelines>
  <map:pipeline>
    <map:match pattern="">              cocoon/ipc_ku/ipc03/へのアクセス
      <map:redirect-to uri="hello"/>    cocoon/ipc_ku/ipc03/hello にリダイレクト
    </map:match>
    <map:match pattern="index*">       cocoon/ipc_ku/ipc03/index.html 等へのアクセス
      <map:redirect-to uri="hello"/>    cocoon/ipc_ku/ipc03/hello にリダイレクト
    </map:match>
    <map:match pattern="hello">        cocoon/ipc_ku/ipc03/hello へのアクセス
      <map:generate src="hello.xml" type="file"/>  hello.xml を入力ファイルとする
      <map:transform src="hello.xsl"/>      XSLT ファイルとして hello.xsl を用いる
```

```

    <map:serialize type="html"/>          html ファイルとして出力
  </map:match>
===== この間に、このフォルダで作成される Web ページについての記述が入る =====
<map:match pattern="*">      これまでに記述されている URL にマッチしないアクセス
  <map:redirect-to uri="hello"/>    cocoon/ipc_ku/ipc03/hello にリダイレクト
</map:match>
</map:pipeline>
</map:pipelines>

```

(\*1)http://localhost:8080/cocoon/ipc\_ku/ipc03/hello の localhost は、Web サーバのホスト名または IP アドレスで置き換わるべきものである。以下でも localhost と表記するが、適当に読み替えてほしい。  
 (\*2)hello.xml の内容は次の通りである。

```

<?xml version="1.0"?>
<page>
<title>Hello</title>
<content>
<para>Hello World 3</para>
</content>
</page>

```

(\*3)hello.xsl の内容は次の通りである。

```

<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="page">
<html>
<head><title><xsl:value-of select="title"/></title></head>
<body><center>
<h2><xsl:value-of select="content/para"/></h2>
<h2>Thank You !</h2>
</center></body>
</html>
</xsl:template>
</xsl:stylesheet>

```

## Cocoon2 による Web ページの作成例

作成を試みた Web ページの幾つかを紹介する。Java2SDK-1.4.0 , Tomcat4.1.24 , Cocoon-2.0.4 をインストールした Web サーバの OS は Windows2000/XP と Red Hat Linux 7.3 ,Web ページに関連して作成した XML や XSP ファイル等は Windows 上のエディタで編集している。作成された Web ページの検証は Netscape 7.1 , Internet Explorer6.0.2 によった。

### 日本語文字コードを点検するデモページ

日本語を表示する Web ページ作成で最も厄介な問題は、日本語の文字コードに絡む問題である。シラバスの表示や休講通知といった日本語を表示する Web ページ作成では、先ずこの問題を解決しておかねばならない。

よく知られているように、Windows 系では一般に Shift\_JIS ,Unix 系では EUC-JP ,Java 言語でのプログラミングでは UTF-8 エンコーディングなどが用いられる。Windows 上のエディタでファイルを編集する場合には、通常は Shift\_JIS コードであることに注意する。

ここで採りあげるプログラムは、

`http://localhost:8080/cocoon/ipc_ku/ipc01/xsp-demo01?name=青木`のようなアクセスに対して、次の図4のようなページを表示する。“青木”の部分が“香川”であれば、“香川さん、こんにちは”となる。また、“?name=青木”の部分が省略されれば、“名前を忘れた”さん、こんにちは”となる。実に単純な Web ページであるが、日本語の取り扱いについて多くの示唆を与えてくれる。

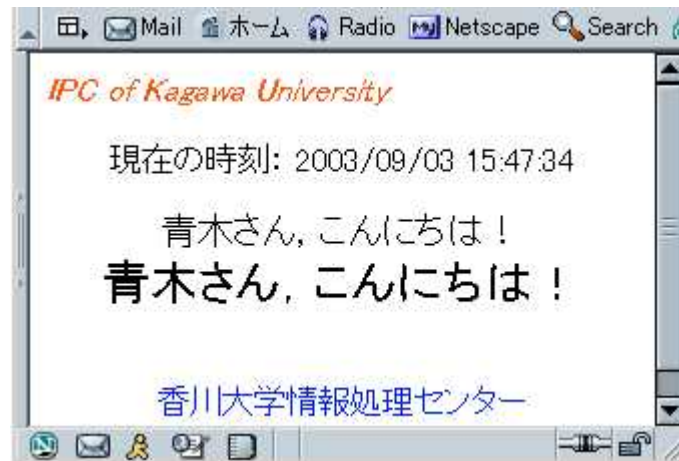


図4：日本語を表示するデモページ

### サイトマップファイル

先ず、`cocoon/ipc_ku/ipc01/sitemap.xmap` の関連する部分を眺めておこう。

```
<!-- Demo01 -->
<map:match pattern="xsp-demo01">
  <map:generate src="demo01.xsp" type="serverpages"/>
  <map:transform src="demo01.xsl"/>
  <map:serialize type="html"/>
</map:match>
```

これからも分かるように、出発点となるファイルは `demo01.xsp` であり、Generator には XSP ファイル処理に対応する `ServerPagesGenerator` コンポーネントが用いられている。Transformer はデフォルトの XSLT Transformer コンポーネント、Serializer は HTML ファイルを出力する `HTMLSerializer` となっている。

### demo01.xsp ファイル

少し長くなるが、文字エンコーディングの取り扱いとも関係し、また XSP ファイルの書式を紹介する意味もあるので、`demo01.xsp` ファイルの内容を次に示す。尚、XSLT ファイルである `demo01.xsl` については付録 A を参照のこと。

```

<?xml version="1.0" encoding="Windows-31J"?> (1)
<xsp:page language="java"
    xmlns:xsp="http://apache.org/xsp" (2)
    xmlns:xsp-request="http://apache.org/xsp/request/2.0"> (3)
<xsp:structure> (4)
  <xsp:include>java.util.*</xsp:include>
  <xsp:include>java.io.*</xsp:include>
</xsp:structure>
<xsp:logic>
  private String getNowDate(){
    return (new Date()).toLocaleString() ;
  }
  private String getName(){
    String st = <xsp-request:get-parameter name="name"/> ; (5)
    // String st = request.getParameter("name") ; //OK
    if (st != null) {
      try{
        st = new String(st.getBytes("ISO8859_1"),"JISAutoDetect") ; (6)
        // st = new String(st.getBytes("ISO8859_1"),"UTF-8") ;
      } catch (Exception e){}
    }
    else {
      st = "¥名前を入力を忘れた¥" ;
      try{
        st = new String(st.getBytes("ISO8859_1"),"JISAutoDetect") ;(7)
        // st = new String(st.getBytes("ISO8859_1"),"UTF-8") ;
      } catch (Exception e){}
    }
    return st ;
  }
</xsp:logic>
<data>
  <date><xsp:expr>getNowDate()</xsp:expr></date>
  <username><xsp:expr>getName()</xsp:expr></username>
</data>
</xsp:page>

```

上のリスト中で番号を付した部分を中心に，解説しておこう。

- (1)XML ファイルであることの宣言文である。この行の encoding="Windows-31J"であるが，Windows 上のエディタでの編集であることから通常は encoding="Shift\_JIS"と表されることが多い。然しながら，XML ファイル中に日本語文字列を記述した場合，encoding="Shift\_JIS"としたのでは，～， ， といった文字が?となって正しく表示されない。このことについては，文献(5)，(6)を参照のこと。
- (2)xsp 名前空間の宣言である。xsp ファイルではこの行を忘れてはならない。
- (3)xsp-request 名前空間の宣言であるが，request ロジックシート(タグライブラリ)を用いることを示している。行(5)がその使用例である。
- (4)Java のクラスライブラリを import する際の，XSP ファイルにおける記述を示している。上記リストでは，java.util.\*と java.io.\*を import している。
- (5)Web ページの FORM 等から name 変数へ入力された値を受け取っている。次の行の
 

```
String st = request.getParameter("name") ;
```

でも良い、というよりこの記述の方が(5)行の記述よりもうまく動作するようである。

(6)Java 言語プログラミングでは、日本語文字列のエンコーディングを Unicode に変換しなければならない。その際の文字化けを防ぐための定形文である<sup>(5)</sup>、<sup>(7)</sup>、<sup>(8)</sup>。Web ページの FORM から method="POST"で文字列を受け取る場合には、行(6)の次のコメント行である

```
// st = new String(st.getBytes("ISO8859_1"), "UTF-8"); を用いる(*1)。
```

(7)XSP, XML ファイル内に書かれた日本語文字列を Unicode に変換する際の定形文である。次の行はテストのためのもので不要である。

その他：<xsp:logic>タグ行と</xsp:logic>タグ行の間には、Java コードによるプログラミングを記述する。上記リストでのトップ要素<data>タグ行と</data>タグ行の間の記述が最終的に得られる XML データ表現で、要素<date>と要素<username>の値を<xsp:expr>タグを用いて Java コードの値から得ている。

(\*1) st = new String(st.getBytes("ISO8859\_1"), "UTF-8"); 行の"UTF-8"は、FORMでの method が"POST"である場合に、(多くの)ブラウザが Web ページの charset と同じエンコーディングで文字列をサーバに送信することによっている。Cocoon2 ではデフォルトで charset=UTF-8 である(以下の Web ページソースを参照)。このあたりの事情については文献(5)を参照してほしい。

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"> // この行に注目
<title>Sample XSP-demo01</title>
</head>
<body>
<font color="#ff0000"><i>IPC of Kagawa University</i></font>
<center>
<p>
    現在の時刻： 2003/09/04 6:05:08</p>
<font size="+1">青木さん，こんにちは！<br>
</font><font size="+2">青木さん，こんにちは！<br>
</font>
</center>
<center>
<font color="#0000ff">
<br>
<p>香川大学情報処理センター</p>
</font>
</center>
</body>
</html>
```

## シラバス表示・編集ページ

文献(2)で、シラバスを表示する HTML ファイル作成を例として、XML データを XSLT プロセッサで処理する方法を紹介した。そこでは、作成され一旦保存された HTML ファイルをあらためて Web ブラウザで閲覧するというものであった。Cocoon2 による処理では、用意する XML, XSLT ファイルは同じであるが、サイトマップに記述された URL への Web ブラウザからのアクセスに応じて、一連の処理が実行され、結果としての HTML ファイルがブラウザ側に返されることとなる。

とは言え、単なるシラバスの表示に話を限るならば、シラバスの内容が常時変更されるものでなくほぼ静的なページであることを考えると、Cocoon2 をわざわざ用いるメリットはそれほど大きくはないであろう。

シラバスを Cocoon2 で取り扱うとすれば、メリットが発揮できるのはその校正等の編集作業においてではないだろうか。プログラミングが可能な XSP の利点を生かすのである。およその手順は次の通りである。

- (1) サーバ側では、ユーザからのアクセスに応じてシラバスを表示するのと同じ方法で編集入力可能な Web ページを作成し、ユーザのブラウザに送信する（図 5 参照）。
- (2) ユーザは Web ページ上でシラバスデータを校正・編集しサーバに送信する。
- (3) ブラウザから受け取った文字列を XSP 内のプログラミングで処理し、シラバス XML データファイルを更新すると共に、結果を示す Web ページをブラウザ側に送信する（図 6）。

ここで、図 5、図 6 に関係するサイトマップを例に、サイトマップにおける正規表現と処理の流れについて触れておこう。サイトマップでの記述を次のようにしている。

```
<map:match pattern="e-hensyuu-*" > //図 5 の表示
  <map:generate src="files/{1}.xml" />
  <map:transform src="edit-syllabus.xsl" type="xt" />
  <map:serialize type="html" />
</map:match>
<map:match pattern="edit-data" > // 図 6 の表示
  <map:generate src="edit-data.xsp" type="serverpages" />
  <map:transform src="edit-data.xsl" />
  <map:serialize type="html" />
</map:match>
```

図 5 のページを表示する URL は、

[http://localhost:8080/cocoon/ipc\\_ku/ipc02/e-hensyuu-030301](http://localhost:8080/cocoon/ipc_ku/ipc02/e-hensyuu-030301)（\*のところが 030301）で、030301 はシラバスの科目コードである。これにより、ipc02/files/{1}.xml となっている部分が ipc02/files/030301.xml と置き換わって files/030301.xml ファイル（付録 B 参照）が校正・編集の対象となり、XML ファイルを入力として受け取る FileGenerator（デフォルトの Generator）に引き渡される。

次いで、TraxTransformer（デフォルトである XSLT Transformer）が Generator からの SAX イベントを XSLT ファイルである edit-syllabus.xsl（付録 C 参照）に基いて処理し、そこからの SAX イベントが更に HTMLSerializer に引き継がれて、最終的には HTML ファイルがブラウザに送られることとなる。尚、特に意味は無いのであるが、

```
<map:transform src="edit-syllabus.xsl" type="xt" />
```

の部分で、XSLT プロセッサにデフォルトのものでなく xt を用いている。

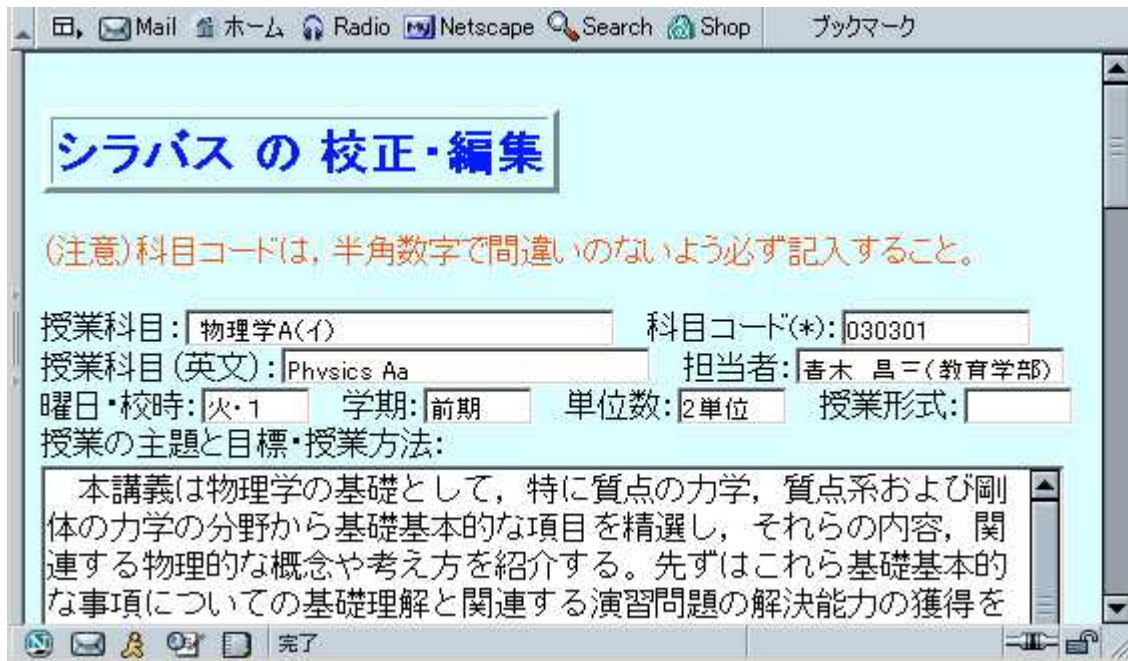


図5：シラバスの校正・編集画面

edit-syllabus.xml ファイルの記述からも分かるように、図5のページソースにおける FORM 要素の action 属性値を”edit-data”としているので、図5のページからデータが送られた後は edit-data が URL として呼び出される。そのときブラウザから送信されてきたデータを処理しシラバスのデータファイル 030301.xml の内容を更新すると共に、図6のページのもととなる XML データを生成するのが edit-data.xsp ファイルである<sup>(\*)</sup>。図6のページを与える XSLT スタイルシートが edit-data.xml となっている。



図6：校正・編集が終了した旨の表示

(\*1)edit-data.xsp では、Java プログラミングにより受け取ったデータを処理しファイル内容を更新している。XSP ファイルの例として記載したいところであるが、作成の途中で長らく作業を中断している関係で入力された特殊な文字の取り扱い等などに不備な点も多く、また既に 300 行を超える内



容となっていることもあり詳細については割愛する。

### 休講通知ページの編集と表示

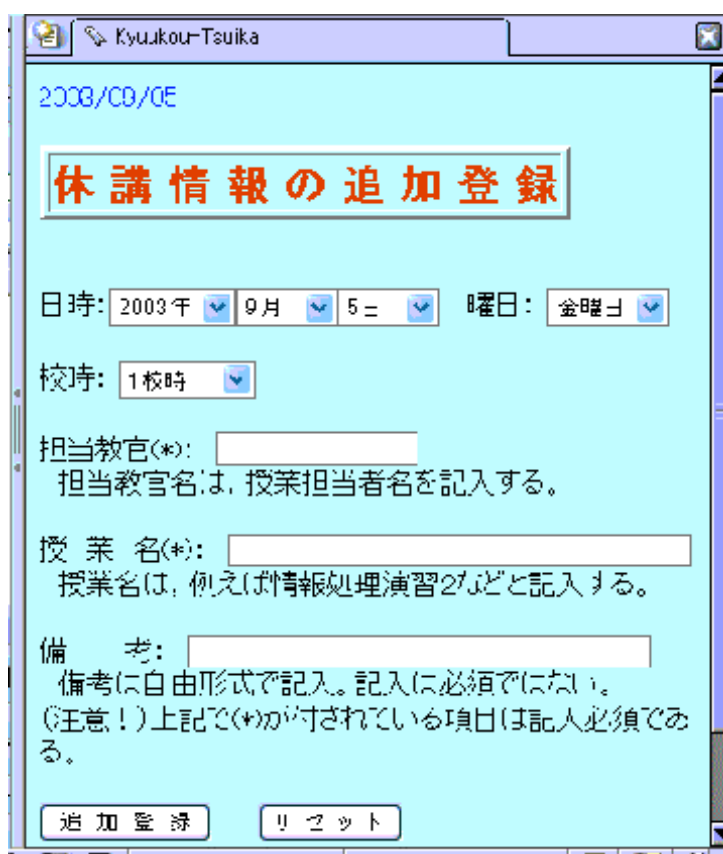
休講を通知するページに関して幾つかのサンプルを作成してみた。

#### 休講通知ページの編集

本学では、学部毎に休講通知の Web ページが提供されている。その編集はどのようになされているのであろうか。恐らくは、Web ページ作成ソフトウェアにより休講通知ページをその都度作成し、サーバにアップロードしているのではないかと想像する。何でもない作業のようでありながら、担当者にとっては面倒この上ないことであろう。Web ページ上で休講通知を編集する 2 つの Web ページを作成してみた。共に、仕組みは上述のシラバスの表示・編集ページ作成と同じである。

#### 休講通知情報の追加ページ

図 7 に Web ページ画面を示す。日時、曜日、校時はプルダウンメニューから選択するようにし、入力すべき項目を少なくして担当者の負担をできるだけ軽減している。



The screenshot shows a web browser window with the title 'Kyuukou-Tsuika'. The page content is as follows:

2003/09/05

**休講情報の追加登録**

日時: 2003年 9月 5日 曜日: 金曜日

校時: 1校時

担当教官(\*):   
担当教官名は、授業担当者名を記入する。

授業名(\*):   
授業名は、例えば情報処理演習2などと記入する。

備考:   
備考は自由形式で記入。記入は必須ではない。  
(注意!)上記で(\*)が付けられている項目は記入必須である。

追加登録      リセット

図 7 : 休講通知情報の追加ページ

#### 休講通知情報の削除ページ

休講通知情報も該当する日時が過ぎれば不要であり削除を要する。図 8 に、Web 上で不要な休講通知情報を削除する画面を示す。ここでは、削除すべき情報にチェックマークを施し

た上で送信する。このページへのアクセス時点で既に日時が経過した休講項目にはデフォルトでチェックマークが付されている。



<input type="checkbox"/>	日時	曜日	校時	担当教官	授業名
<input checked="" type="checkbox"/>	9月4日	木曜	1校時	香川大学	新大学論
<input type="checkbox"/>	9月9日	月曜	3校時	青木 昌三	ミクロとマクロの物理学 I
<input type="checkbox"/>	9月19日	金曜	3-4校時	情報 センタ	情報ネットワーク概論 II

図 8 : 休講通知情報の削除ページ

#### 休講通知の表示

休講通知情報の表示そのものは、これまで紹介した例に比べれば簡単である。休講の情報が記述された XML ファイルとそれを HTML 形式に変換する XSLT ファイルを用意するだけでこと足りる。ここでは、作成したページ例の XSLT ファイルにおいて、次の 2 点を考慮したことを付け加えておこう。

休講通知の表示にあたっては、日付の順にソートされたものを表示すること。

休講通知の情報を追加する場合、1 ヶ月以上も先のデータもあれば急を要する明日のデータもあり得よう。Web ページを HTML エディタで作成する場合にこれら情報をいちいち並び替えるのは結構手間のかかる作業である。表示の際に自動的にソートされることが望まれる。

日時の過ぎ去った不要な情報は、元となるデータに含まれていても表示しないこと。

不要となった休講通知情報の削除ページを作成したが、毎朝毎夕このページにアクセスして削除作業を行うのは時間の無駄である。不要な情報が削除された HTML ファイル作成を真夜中のある時刻に自動的に行うようスケジューリングすることも 1 つの方法ではあるが、ブラウザからのアクセスが時刻を問わないものであることを考えれば、アクセスに応じて臨機応変に表示画面を切り替えるアイデアも悪くはないであろう。

#### 携帯電話への対応

携帯電話の普及度を考慮すれば、提供する Web ページの携帯電話への対応は時代の要請であろう。シラバスはともかく、情報量が比較的少なく、そして「今すぐにでも見たい」というユーザ側からみて緊急度の比較的高い休講通知ページなどは、是非とも携帯電話に対応することが求められる。

Cocoon2 では、アクセスブラウザのユーザーエージェント名を判別することにより、同じ URL へのアクセスに対して異なったコンテンツを表示させることが可能である<sup>(\*)</sup>。この

Cocoon2 のもつ機能を，携帯電話からのアクセスに対して休講通知を表示させることに応用してみた<sup>(\*)</sup>。i-モード HTML シミュレータ<sup>(\*\*)</sup>と EZweb 用シミュレータ<sup>(\*\*\*)</sup>を用いてのアクセス表示画面を示す（図 9，図 10 参照）<sup>(\*\*\*\*)</sup>。

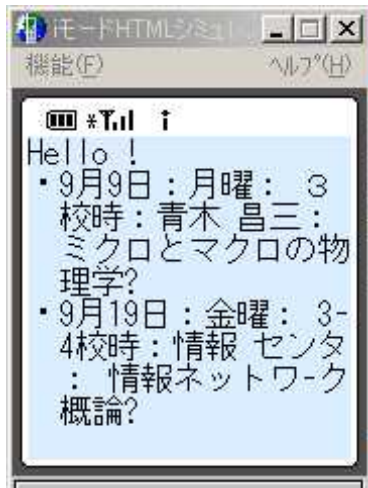


図 9：シミュレータによる表示（i-モード） 図 10：シミュレータによる表示（EZweb）

(\*1)この機能では，BrowserSelector コンポーネントが用いられる。例については，文献(3)を参照のこと。

(\*2)携帯電話への対応にあたって，cocoon¥sitemap.xmap に 2 つの変更を加えている。携帯電話では Shift\_JIS が用いられるようなので，名前が html-s で，文字エンコーディングが Shift\_JIS である HTMLSerializer を新たに登録した(以下参照)。挿入箇所はデフォルトで HTMLSerializer が記述されている部分の直ぐ後である。携帯電話対応 HTML ページについては，この Serializer を用いる。

```
<!--html2 -->
  <map:serializer logger="sitemap.serializer.html" mime-type="text/html" name="html-s" pool-grow="4" pool-max="32" pool-min="4" src="org.apache.cocoon.serialization.HTMLSerializer">
    <buffer-size>1024</buffer-size>
  <!-- <encoding>Windows-31J</encoding> -->
    <encoding>Shift_JIS</encoding>
  </map:serializer>
```

J-Sky Web と WZweb への対応として，ユーザーエージェントに関わる下記の記述を付け加えた。挿入箇所は，<map:selectors default="browser">と</map:selector>間である。

```
<browser name="jsky" useragent="J-PHONE"/>
<browser name="wap" useragent="KDDI"/>
```

BrowserSelector コンポーネントの使い方の紹介も兼ねて，携帯電話からのアクセスに関わる sitemap.xmap での記述を以下に示す。keiji5 へのアクセスに対して，携帯電話からのときには keiji-cell，それ以外は通常の keiji2 ヘリダイレクトさせている。

```
<map:match pattern="keiji5">
  <map:select test="browser">
    <map:when test="imode">
      <map:redirect-to uri="keiji-cell"/>
    </map:when>
    <map:when test="jsky">
      <map:redirect-to uri="keiji-cell"/>
    </map:when>
    <map:when test="wap">
      <map:redirect-to uri="keiji-cell"/>
    </map:when>
    <map:otherwise>
      <map:redirect-to uri="keiji2"/>
    </map:otherwise>
  </map:select>
</map:match>
```

(\*3)i-モード HTML シミュレータ Version6.0 による。

[http://www.nttdocomo.co.jp/p\\_s/application/](http://www.nttdocomo.co.jp/p_s/application/)を参照。

(\*4)Openwave SDK Universal Edition1.1 に附属するシミュレータによる。

<http://www.au.kddi.com/ezfactory/tool/ue/index.html>を参照。

(\*5)2つのシミュレータで、や が?と表示され、文字化けが起こっている。文字エンコーディングを Shift\_JIS とした Serializer を用いていることによっている。

#### メールの送信 - Java Mail の利用 -

シラバスの校正・編集や休講情報の追加・削除の作業は、許可された者のみが行うべきであろう。とすれば、Web ページへのアクセスに先立って、アカウントとパスワードによる認証が必要となる<sup>(\*)</sup>。

それでは、パスワードを忘れてしまった者へのケアはどうすればよいのであろうか。1つの解は、Web 上からの要請に応じて、忘れたパスワードを当人宛にメールで送信することである。こうしたパスワードを送信するといった、メール送信機能を持つページもまた、休講通知表示システムや Web ショッピングページ、所謂グループウェア的なプログラム等に欠くことのできないパーツ群の1つである。

メールを送信するプログラム作成は少々厄介だと思っても知れないが、Cocoon2 にはメール送信プログラムである Java-Mail を利用するためのロジックシート(タグライブラリ)が用意されていて、XSP ファイル中でのほんの 10 行程度の簡単なコーディングでメール送信機能を実現することができる。Java-Mail を利用するにあたってのロジックシートの使い方については文献(4)に詳しく紹介されているので、ぜひ同文献を参照してほしい<sup>(\*)</sup>。

文献(4)での記述に倣って、Web ブラウザからのアカウント名の送信を受けて、パスワードをメールで送信するページを作成してみたのが図 11 である。アカウント名、パスワード、メールアドレス等が記述されたファイルからパスワードを読み込んで、当該アカウントのアドレス宛にメールが間違いなく送信されることが確かめられた。

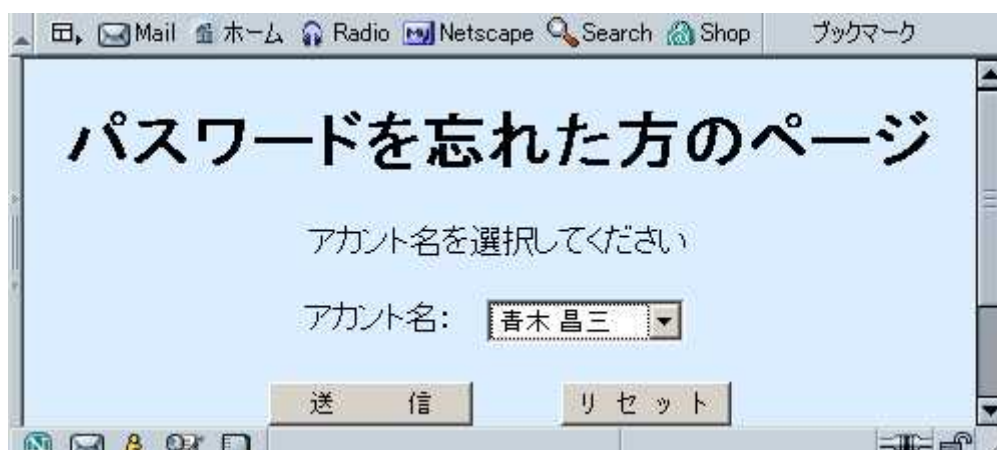


図 11 : パスワード送信のためのページ

- (\*1) アカウントとパスワードによる認証を受けてのアクセスについては、その後のセッション管理が必要となる。これについては、session ロジックシートを利用できる。
- (\*2) Java-Mail の利用は大変重宝する。文献(4)に従って sendmail ロジックシートが利用できるように設定した事柄をまとめておく。

\$CATALINA\_HOME¥common¥lib ディレクトリ内にある mail.jar と activation.jar の 2 つのファイルを \$CATALINA\_HOME¥webapps¥cocoon¥WEB-INF¥lib ディレクトリにコピーし格納する。

\$CATALINA\_HOME¥webapps¥cocoon¥WEB-INF¥cocoon.xconf ファイルを編集して、`<parameter name="prefix" value="input"/>` という記述があるロジックシート定義の後ろに (`<builtin-logicsheet> = = = </builtin-logicsheet>` の最後の項目として) 下記に示す内容を付け加える。

```

<!--2003.6.30 -->                この行はコメント，次行より次のコメント行までの内容を加える
<builtin-logicsheet>
  <parameter name="prefix" value="sendmail"/>
  <parameter name="uri" value="http://apache.org/cocoon/sendmail/1.0"/>
  <parameter name="href"
value="resource://org/apache/cocoon/components/language/markup/xsp/java
/sendmail.xsl"/>
  </builtin-logicsheet>
<!-- 2003.6.30 -->                この行はコメント

```

XSP ファイルには ,sendmail ロジックシートを用いるための名前空間の定義を記述する(以下参照)。

```

<?xml version="1.0" encoding="Windows-31J"?>
<!-- <?xml version="1.0" encoding="Shift_JIS"? -->
<!-- Author: M.Aoki "aoki@ed.kagawa-u.ac.jp" -->
<xsp:page language="java"
  xmlns:xsp="http://apache.org/xsp"
  xmlns:xsp-session="http://apache.org/xsp/session/2.0"
  xmlns:xsp-request="http://apache.org/xsp/request/2.0"
  xmlns:sendmail="http://apache.org/cocoon/sendmail/1.0"> この行を加える
<xsp:structure>

```

XSP ファイルにメール送信のためのコードを記述する。参考までに、作成した XSP ファイルの内容を付録 D に示す。

## おわりに

Cocoon2 による Web ページ作成の概要について、シラバスや休講通知の表示・編集といった身近な題材を採りあげてサンプルページを紹介した。折から、本学でもシラバスの Web 上での閲覧や編集が話題となっているようである。

誌面の都合上、採りあげたページについての詳細を割愛したがため、やや中途半端で荒削りな解説に終わってしまった感じではあるが、Cocoon2 の素晴らしさやその威力の一端を感じ取って頂けたのではないだろうか。

Cocoon2 は「XML をベースとした Web アプリ開発フレームワーク」へと発展しつつあるようで、Java による Web アプリ開発のキーテクノロジーの 1 つとして、Cocoon2 から暫く目を離すことができないように思われる。本稿を通して、Java 或いは Cocoon2 による Web アプリ開発に興味をもつ方が 1 人でも多くなることを、心から願う次第である。

## 参考文献

- (1)青木 昌三：「Java による Web アプリ開発の勧め ( ) - JSP , XSP 利用環境の構築 - 」  
香川大学情報処理センター Newsletter No.49 , June 16 , 2003
- (2)青木 昌三：「XML プログラミングによる HTML ファイルの作成 - シラバスファイル为例として - 」 香川大学情報処理センター年報 第 9 号 ( 2001 年 2 月 )
- (3)坂上 徹 他：「Apache Cocoon2 XML による Web サイト構築とコンテンツ管理」  
2002.9 ソフト・リサーチ・センター ( ISBN 4-88373-173-1 )
- (4)ラジヨス・モチャル：「Cocoon2 の威力を知る」  
JavaWorld 2003 April , アイ・ディ・ジー・ジャパン
- (5)原田 洋子 , 鯉崎 義之：「サーブレット&JSP トラブルシューティング」  
JavaWorld 2001 November , アイ・ディ・ジー・ジャパン
- (6)岡田 育枝：「Web アプリ開発のイロハ 第 2 回 JSP の基本 , サーブレットとの連携」  
JavaWorld 2003 April , アイ・ディ・ジー・ジャパン
- (7)風間 一洋：「CAFE BAVE Java プログラミング・ノート 2」  
UNIX MAGAZINE 1998.4 , アスキー
- (8)菊田 英明：「SE・プログラマ スタートアップテキスト JSP [基礎]」  
2002.2 , 技術評論社 ( ISBN 4-7741-1340-9 )

## 付録

### 付録 A : demo01.xml ファイルの記述

日本語を表示するデモページで用いた XSLT ファイル demo01.xml の内容は次の通りである。

```
<?xml version="1.0" encoding="Shift_JIS"?>
<!-- Author: M.Aoki "aoki@ed.kagawa-u.ac.jp" -->
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html>
      <head>
        <title>Sample XSP-demo01</title>
      </head>
      <body>
        <xsl:call-template name="ipc1"/>
        <center>
          <p>
            現在の時刻 : <xsl:value-of select="data/date"/>
          </p>
          <xsl:call-template name="username">
            <xsl:with-param name="s" select="'1'"/>
          </xsl:call-template>
          <xsl:call-template name="username">
            <xsl:with-param name="s" select="'2'"/>
          </xsl:call-template>
          <!-- <xsl:call-template name="username">
            <xsl:with-param name="s" select="'3'"/>
          </xsl:call-template> -->
        </center>
        <xsl:call-template name="ipc2"/>
      </body>
    </html>
  </xsl:template>

  <!-- Templates -->
  <xsl:template name="username">
    <xsl:param name="s"/>
    <xsl:choose>
      <xsl:when test="$s=1"><font size="+1">
        <xsl:value-of select="data/username"/>さん, こんにちは! <br/>
      </font></xsl:when>
      <xsl:when test="$s=2"><font size="+2">
        <xsl:value-of select="data/username"/>さん, こんにちは! <br/>
      </font></xsl:when>
      <xsl:otherwise><font size="+3">
        <xsl:value-of select="data/username"/>さん, こんにちは! <br/>
      </font></xsl:otherwise>
    </xsl:choose>
```

```
</xsl:template>
<xsl:template name="ipc1">
  <font color="#ff0000">
    <i>IPC of Kagawa University</i>
  </font>
</xsl:template>
<xsl:template name="ipc2">
  <center>
    <font color="#0000ff">
      <br/>
      <p>香川大学情報処理センター</p>
    </font>
  </center>
</xsl:template>
</xsl:stylesheet>
```



## 付録 B : files/030301.xml ファイルの記述

```
<?xml version="1.0" encoding="Windows-31J" ?>
<!-- <?xml version="1.0" encoding="Shift_JIS" ?> -->
<!-- Author: M.Aoki "aoki@ed.kagawa-u.ac.jp" -->
<syllabus>
  <subject code="030301" category="">
    <name> 物理学 A ( イ ) </name>
    <e-name>Physics Aa</e-name>
    <title></title>
    <e-title></e-title>
    <teacher>青木 昌三 ( 教育学部 ) </teacher>
    <unit>2 単位</unit>
    <term>前期</term>
    <weekly-daytime>火・1</weekly-daytime>
    <c-form></c-form>
    <contents>
      <outline> 本講義は物理学の基礎として、特に質点の力学、質点系および剛体の力学の分野から基礎基本的な項目を精選し、それらの内容、関連する物理的な概念や考え方を紹介する。先ずはこれら基礎基本的な事項についての基礎理解と関連する演習問題の解決能力の獲得を主眼とし、併せて物理学がどのように自然現象を取り扱うかといった物理学の手法に慣れ親しみ、物理学的な見方や問題取り扱いのセンスを養うことを目的としたい。時間に余裕が生じれば、熱力学分野の基礎的項目についても紹介する。 </outline>
      <system></system>
      <plan>第 1 回 ~ 第 3 回
        質点の運動の記述
          座標系，位置，速度，加速度，
          簡単な運動の例
        第 4 ~ 第 6 回
          運動の法則について
            力と運動，力と運動の例
        第 7 回，第 8 回
          力と仕事，エネルギー，ポテンシャル，
          力学的エネルギーの保存則，
          保存則を用いた運動の取り扱い例
        第 9 ~ 第 11 回
          質点系の力学の基礎
        第 12 回 ~
          剛体の力学の基礎事項，剛体の回転運動
          剛体のつりあい
        時間的に余裕がある場合：
          熱力学の基礎的事項
            温度と熱，気体の状態方程式，
            状態の変化，・・・，
            熱力学の第 1 法則，第 2 法則
            .....
      </plan>
      <caution></caution>
      <evaluation> 成績の評価は主として期末試験の成績による。レポートを課した場合に
```

```
は、その評価も加味する。</evaluation>
  <others> 物理学Aイ～A八の授業間で受講者数にアンバランスを生じた場合には、受講
者数の調整を行うことがある。</others>
  </contents>
  <text>原 康夫 著 「詳解 物理学」(東京教学社) (生協の書籍部にて購入)</text>
  <s-text></s-text>
  <r-text></r-text>
</subject>
</syllabus>
```

## 付録 C : edit-syllabus.xml ファイルの記述内容

```
<?xml version="1.0" encoding="Windows-31J"?>
<!-- Author: M.Aoki "aoki@ed.kagawa-u.ac.jp" -->
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="subject">
<html>
<head><title>Edit-Syllabus</title></head>
<body bgcolor="#e0ffff">
<br/>

  <TABLE border="3">
    <TR>
      <TH><font color="#0000ff" size="+2">シラバスの校正・編集</font>
    </TH>
  </TR>
</TABLE>
<br/>
<font color="#ff0000">(注意)科目コードは,半角数字で間違いのないよう必ず記入すること。</font>
<br/><br/>
<form action="edit-data" method="POST">

  授業科目:<input type="text" size="40" maxlength="50" name="name">
    <xsl:attribute name="value"><xsl:value-of select="name"/></xsl:attribute>
  </input>
  科目コード(*):<input type="text" size="16" maxlength="16" name="code">
    <xsl:attribute name="value"><xsl:value-of select="@code"/></xsl:attribute>
  </input>
<br/>
  授業科目(英文):<input type="text" size="34" maxlength="60" name="e-name">
    <xsl:attribute name="value"><xsl:value-of select="e-name"/></xsl:attribute>
  </input>
  担当者:<input type="text" size="24" maxlength="30" name="teacher">
    <xsl:attribute name="value"><xsl:value-of select="teacher"/></xsl:attribute>
  </input>
<br/>
  曜日・校時:<input type="text" size="8" maxlength="12" name="weekly-daytime">
    <xsl:attribute name="value">
      <xsl:value-of select="weekly-daytime"/></xsl:attribute>
  </input>
  学期:<input type="text" size="8" maxlength="12" name="term">
    <xsl:attribute name="value"><xsl:value-of select="term"/></xsl:attribute>
```

```

bute>
  </input>
  単位数:<input type="text" size="8" maxlength="10" name="unit">
    <xsl:attribute name="value"><xsl:value-of select="unit"/></xsl:attri
bute>
  </input>
  授業形式:<input type="text" size="8" maxlength="8" name="c-form">
    <xsl:attribute name="value"><xsl:value-of select="c-form"/></xsl:att
ribute>
  </input>
  <br/>
  授業の主題と目標・授業方法:<br/>
  <textarea name="outline" rows="8" cols="60" wrap="hard">
    <xsl:value-of select="contents/outline"/>
  </textarea>
  <br/>
  授業計画:<br/>
  <textarea name="plan" rows="18" cols="60" wrap="hard">
    <xsl:value-of select="contents/plan"/>
  </textarea>
  <br/>
  試験・成績評価の方法:<br/>
  <textarea name="evaluation" rows="4" cols="60" wrap="hard">
    <xsl:value-of select="contents/evaluation"/>
  </textarea>
  <br/>
  履修上の注意・その他:<br/>
  <textarea name="others" rows="4" cols="60" wrap="hard">
    <xsl:value-of select="contents/others"/>
  </textarea>
  <br/>
  テキスト・教材:<br/>
  <textarea name="text" rows="4" cols="60" wrap="soft">
    <xsl:value-of select="text"/>
  </textarea>
  <br/>
  <br/>
  <input type="submit" value="編集登録"/>
  <input type="reset" value="リセット"/>
</form>

</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

## 付録 D : メールを送信する XSP ファイルの例 ( sendmail02.xsp )

```
<?xml version="1.0" encoding="Windows-31J"?>
<!-- <?xml version="1.0" encoding="Shift_JIS"? -->
<!-- Author: M.Aoki "aoki@ed.kagawa-u.ac.jp" -->
<xsp:page language="java"
    xmlns:xsp="http://apache.org/xsp"
    xmlns:xsp-session="http://apache.org/xsp/session/2.0"
    xmlns:xsp-request="http://apache.org/xsp/request/2.0"
    xmlns:sendmail="http://apache.org/cocoon/sendmail/1.0">
<xsp:structure>
  <xsp:include>java.io.*</xsp:include>
  <xsp:include>java.util.*</xsp:include>
  <xsp:include>ipc_ku.Entry</xsp:include>
</xsp:structure>
<page0>
<xsp:logic>
  String dir_name = "c:/tomcat 4.1/webapps/cocoon/ipc_ku/ipc06/etc/" ;
  String f_name = "passwd" ;
  String nothing1 = "NO" ;
  String id = null ;
  String passwd = "" ;
  String user_auth = "YES" ;
  String name = "" ;
  String e_mail = "" ;
  String enc = "JISAutoDetect" ;
  String smtphost="snake.cc.kagawa-u.ac.jp" ;
  String from="aoki@cc.kagawa-u.ac.jp" ;
  String subject="From Admin";
  String body="Your Password is : " ;
  String charset="ISO-2022-JP" ;

  id=request.getParameter("account") ;
  if (id == null || id.length() == 0)
    user_auth = nothing1 ;
  else {
    String fname = dir_name + f_name ;
    Entry entry= new Entry(fname,enc) ;
    name = entry.getAccountName(id) ;
    passwd=entry.getAccountPasswd(id) ;
    e_mail = entry.getAccountE_mail(id) ;
    if (e_mail.length()==0 || passwd.length()==0 || e_mail.equals("")
|| passwd.equals(""))
      user_auth="NO" ;
  }
  if( user_auth.equals("YES")){ // 次行からメール送信のためのコード
    <sendmail:send-mail>
      <sendmail:charset><xsp:expr>charset</xsp:expr></sendmail:charset>
      <sendmail:smtphost><xsp:expr>smtphost</xsp:expr></sendmail:

```

SMTP サーバ名  
メール送信者名  
メールサブジェクト  
メール文の一部  
メールの文字コード

パスワードを得る  
メールアドレスを得る

```
smtphost>
  <sendmail:from><xsp:expr>from</xsp:expr></sendmail:from>
  <sendmail:to><xsp:expr>e_mail</xsp:expr></sendmail:to>
  <sendmail:subject><xsp:expr>subject</xsp:expr></sendmail:subject>
  <sendmail:body><xsp:expr>body</xsp:expr><xsp:expr>passwd</xsp:exp
r></sendmail:body>
  <sendmail:on-success>
    <p>Email successfully sent.</p>
  </sendmail:on-success>
  <sendmail:on-error>
    <p>An error occurred: <sendmail:error-massage/></p>
  </sendmail:on-error>
  </sendmail:send-mail>          // ここまでがメール送信のためのコード
}
</xsp:logic>
  <result_code><xsp:expr>user_auth</xsp:expr></result_code>
  <id><xsp:expr>id</xsp:expr></id>
  <name><xsp:expr>name</xsp:expr></name>
  <passwd><xsp:expr>e_mail</xsp:expr></passwd>
</page0>
</xsp:page>
```

# PC(パーソナルコンピュータ)を自分で組み立ててみよう

教育学部 黒田 勉

## 1. はじめに

昨今のパーソナルコンピュータ(以下 PC と略)の高速化には目覚ましいものがあり,1年前に(大枚を叩いて)購入した PC が,すでに陳腐化してしまうことが見受けられるようになっており,殊に,PC の動作クロックは 2003 年夏には 3.2GHz を記録するようになってきている.

2001 年には,Pentium 4 の 1.5GHz 前後が標準的な CPU だったことを考え合わせると,CPU の速度は倍,すなわち処理時間が半分,と言うことになる.一昨年末,筆者が Windows XP のインストール記を News Letter No.43 に執筆(発行・掲載は 2002 年 1 月末)してから,約 1 年半でこのような状態である.最も,Pentium 4 3.0GHz は 2003 年 10 月の価格改定以来 3 万円程度で売られているので,コストパフォーマンスは,CPU 価格が 2 万円を切る 2.6GHz 程度が良いものと考えられる.

この間,その他のデバイス(DVD-ROM,CD-R など)が劇的に速くなったかというところ,そう言う訳でもなく,高スペックなデバイスが,より安価になって買い求めやすくなった程度である.但し,HDD は別で,より高密度化が図られ,1GB 当たりの単価が半分以下になってきている.

また,主基板(メインボード)のチップセットに関しては,intel の 845 で基本的スペックは殆ど変わっていないのが実情である.いろいろ細かい点では改良が進んでいるが,ベースクロック 133MHz に対応しているか否か,DDR SDRAM に対応しているか否か程度である.

このような場合,CPU だけ載せ替えると言うのが,簡単かつ安全に高速化が図れる方法であるのは十分肯ける話になってくる.

本稿で述べている手法は,PC のパワーアップに有効な上記 CPU や VGA カードの交換にも応用が利くので参考にさせていただければ幸いである.

## 2. PC のスペック

計算機センターの予算の都合もあり,今回組立てた PC のスペックは,下記の通りである.

CPU: intel Pentium4 1.6AGHz (2003 年 4 月に 2.53GHz に交換)  
Memories: PC-2100 DDR SDRAM 512MB × 2, (Total 1GB)  
Mainbord: A-Open AX4B Pro 533A  
LAN, Sound, ATA-133 controler ON Board, USB2.0 Port × 6  
Graphic: Matrox Parhelia 128  
HDD: IBM IC35L060AVER07, (60GB, ATA133)  
IBM IC35L120AVVA07, (120GB, ATA133)  
DVD-ROM: MAtsushita SR-8588

CD-R: TEAC CD-W540E, (×40 Write)  
MO: Fujitsu MCM3130AP-S, (1.3G MO)  
Case: 星野金属 MT-PRO 1200 S-Limited  
Others: IBM 106 Keyboard (US), Logitech TrackMan Marble Wheel

大まかな市場価格は、ケースと MO、キーボード類を除けば 15 万円、Graphic カードを GeForce 系に、HDD を 1 台に変更すれば 10 万円程度になる。ケースはアルミ製の国産品で、冷却能力と組立易さに定評のある星野金属(Windy)のものを使用した。このケースは、市場価格でも 4 万円を切る程度であるが、最低でも 5 年は使用することを考慮に入れ、今後 CPU やグラフィックカードの発熱量増大に対応したものとした。ちなみに、筆者が現在使用している PC (Windows98 で使用) のケースも同社製で、使用を開始してからすでに 5 年以上経過しているが電源を交換したのみで、現役の仕事用 PC として活躍している。

OS は、今後の講義の参考資料作成の事も考えて Windows XP Upgrade 版をインストールするようにした。

### 3. 組立て詳細

#### 3.1. 準備物

工具類なしでは作業ができないので、工具を準備する。

筆者は、VESSEL の R2-200 という長さ 20cm の 2 号 (No.2 とも表示されている) + ドライバを利用している。このドライバーは芯が磁石になっており、ケース内に落ちたネジ等を拾うのに大変重宝している。ちょっと細めの 1 号のドライバーは、ネジ山を潰すことがあるので、注意が必要であろう。

また、希にネジ類が入っていない (入れ忘れ?) ケースがあるので、「パソコン用ネジ」をケースとともに購入しておけば万全。その他、- ドライバーも準備できればよいが、あまり頻繁には使わない。

作業をするためには周りに十分スペースのある作業台 (机) があると楽に作業ができる。目安としては、机の周りをしゃがんで一周できる程度。高さは 60 ~ 70cm がベスト。

コンセントが遠い場合は延長コードかテーブルタップをお忘れ無く。

テスターやハンダごてにお世話になることは、まず無いだろう。

#### 3.2. ケースの解体

組立て用 PC ケース殆どは、中に何も入っていない状態で組み立てられており、それが段ボール箱に梱包されている。ケースの箱の封を切って、ケースを外に出しビニール袋を取り除くと、ドライブ類が全くない「側だけ」の PC が出現する訳である。ケースによっては重量があったり、引っかかりがなかったりするので、机や作業台の上 (床の上でも可) にケースの箱をひっくり返して、箱を上を持ち上げるのが楽であると考えられる。

ケースを作業台の上に載せると作業開始である。このままでは何もできないので、サイドパネルを取り外し、ケース内にデバイスを取り付けられるようにしなくてはならない。ケースにも依るが大半のものは、側板が背面にネジ止めされている。高級なケースには、ネジ止



めが全く必要なく，クリックやボスで止めるものもある．親切なケースには解体方法が載った冊子（紙切れの場合もあるが）が同封されているので，それを参照すればよい．



図 1 . メインボード取り付けシャーシと背面ファン

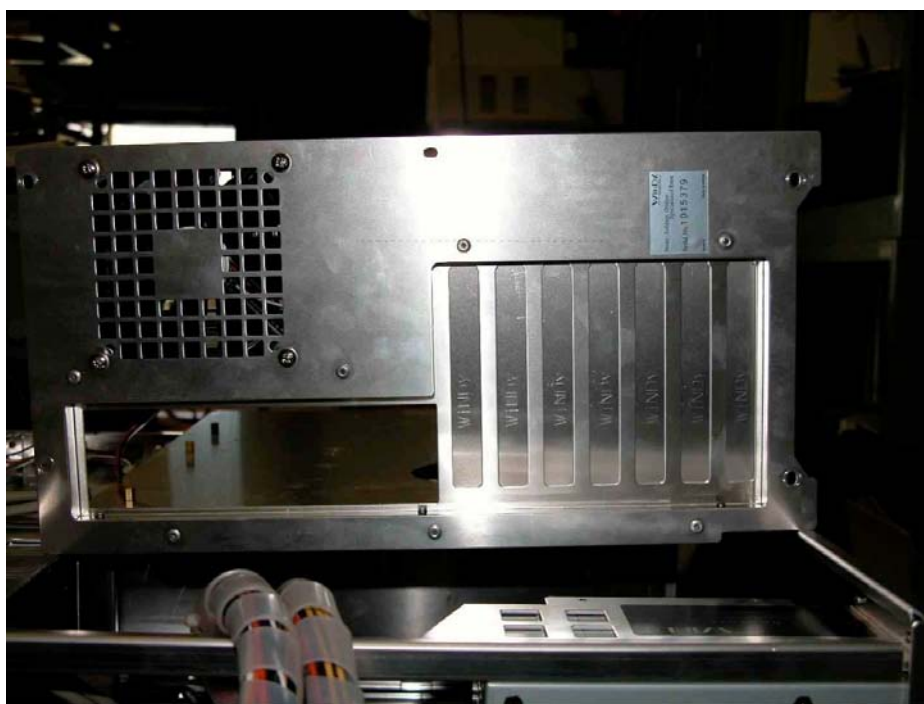


図 2 . 図 1 (シャーシ) の右側側面 (ケース背面)

解体作業が終わると，メインボードの取り付けに掛かる．図 1 は，メインボード取り付けシャーシと背面ファンの写真で，メインボード取り付けネジを取り付けた後の状態である．

ケースによっては、本体を横にしないとイケない場合があるので、適宜対処してもらいたいが、このようなケースは作業性が劣ると考えられる。逆に、メインボード取り付けシャーシが別にできるケースは、価格が比較的高いものが多い。

図2は、図1の右側側面で、ケース背面になる部分である。左方下側は「IOシールド」のための開口部であり、後述の主基板に付属してきたものをはめ込むようになっている。右側に並んでいる長方形の穴は、拡張スロットのための開口部で、拡張スロットにカードを挿入しない場合はそのまま塞いでおく。

### 3.3. フロントパネルの分解

このケースはフロントパネルの下に手を掛け、手前に引けばフロントパネルがはずれる構造になっている。また、3.5インチと5インチベイはそれぞれネジで止められており、必要に応じてネジを弛めて取り外すようになっている。今回は、図3の様に5インチベイに2台（DVD-ROMとCD-ROM）、3.5インチベイに2台（FDDとMOドライブ）のドライブ類を入れるので、カバーは図3中、右のように取り外した。

なお、再下段部分はUSBフロントアクセスベイとして用いる。



図3. フロントパネルを外した状態（左），フロントパネル表（中），同裏（右）

### 3.4. メインボードとIOシールド

主基板（メインボード，マザーボードとも呼ばれる）は，スペックの項目にも書いたようにAOpenのAX4BC Pro 533を用いた。図4は，同基板を背面側から写したもので，電源安定用コンデンサやコイルが多数見られることが判る。

さて，このメインボードは，ATX規格に準拠しているが，コネクタ類の位置が以前のものと大分変わっており，ケース付属のIO（あいおー）シールド板は殆ど使えない。このような場合，たいていメインボードの「箱」の中に付属品として入っている。この板も同様で，図5の上側に示すようなIOシールド板が入っていた。なお，図5の下側のものは，予めケースに

付属していたもので、図2では既に取り外してあり、そこには長方形の穴が開いている。

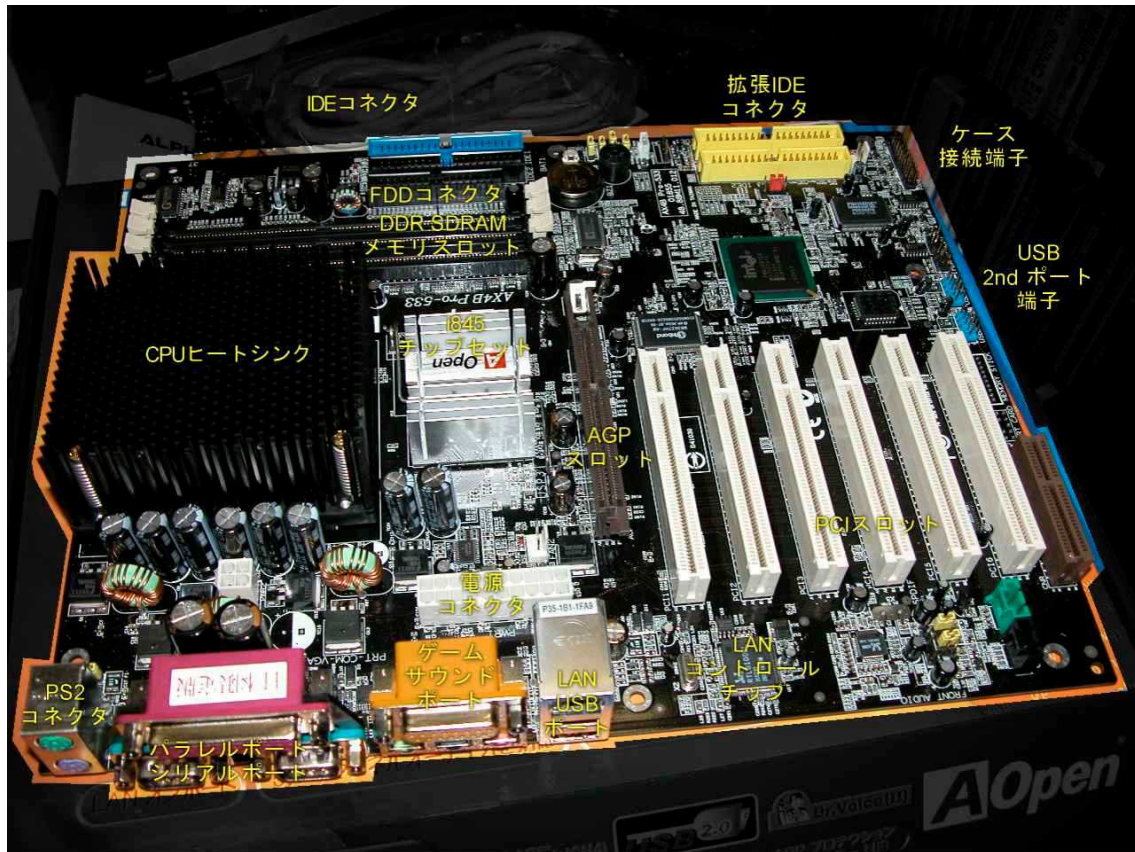


図4．主基板（メインボード）外観

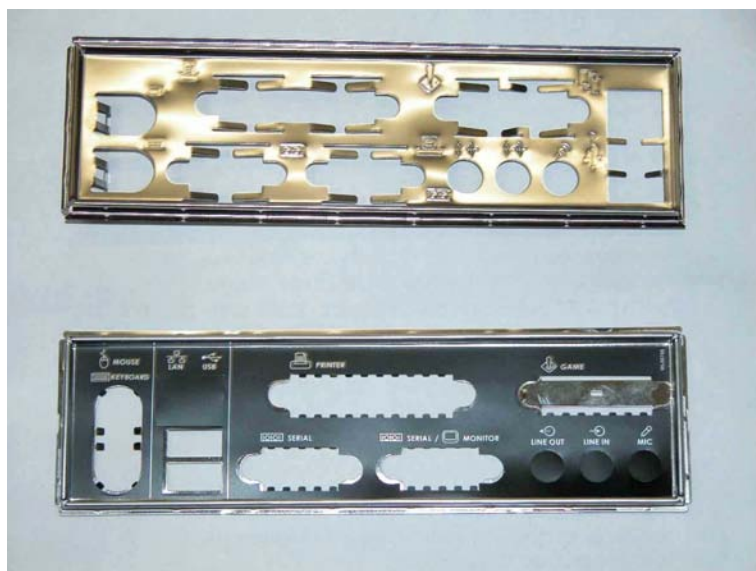


図5．IO シールド板，上がメインボード付属品，下がケース付属のもの





図 6 . IO シールド板をはめ込んだシャーシ背面

もし、他社のメインボードを購入して、ケース付属のものと穴の位置が合わないときは、必ず同じようなパーツが付属しているので、無いような場合は購入店に相談することをお勧めする。

殆どの IO シールド板は、固定用の出っ張りがあり、簡単に取り付けることができる。同様に取り外しも簡単にできるので、メインボード固定時には、注意が必要である。ガタが大きいときは、粘着テープ等で仮止めすると良いだろう。

図 6 は、図 2 に図 5 の IO シールド板をはめ込んだ状態を示したもので、今回の場合は、特に苦労もなくはめ込むことができた。ケースによっては、はめ合わせが巧くいかない場合がある（中には長方形の穴が開いていないものもある）ので現物合わせでよく確認してもらいたい。図 7 は横から見たところで、ぴったりとはめ込まれていることが判る。

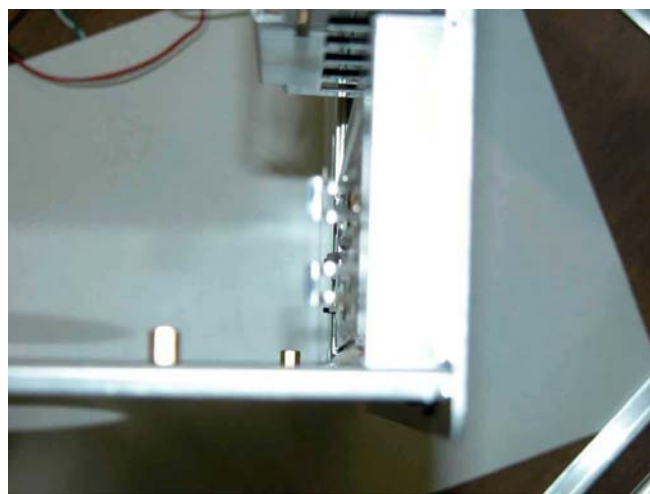


図 7 . IO シールド板をはめ込んだケース背面

### 3.5. メインボードのシャーシへの取り付け

いよいよ組み立て作業の第一段階に入る。CPU と CPU ファン、メモリをメインボードに取り付ける。本稿では、Pentium 4 付属のヒートシンクは冷却能力が奮わないので、Alpha 社製のヒートシンクとファンを用いて CPU 冷却能力を高めている。取り付け方法は、それぞれのマニュアルに詳しく掲載されているのでここでは割愛させていただく。

CPU、メモリを取り付けた主基板をシャーシの上に載せ、ネジ止めをしていく。このシャーシに使用されているネジは、殆どミリネジであるので、そちらを用いる。ケースにより、インチネジが用いられていたりセルフタップのネジが用いられていたりするので、現物合わせで使用する必要がある。

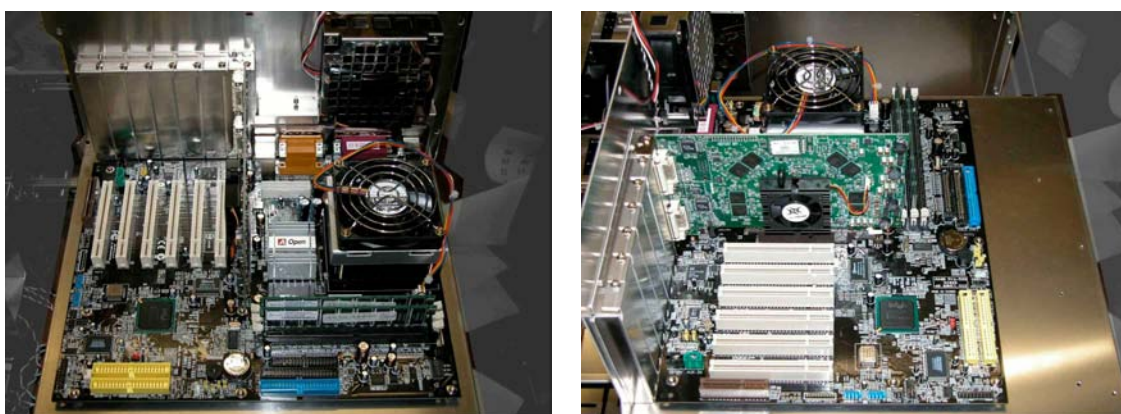


図 8. 主基板を固定したシャーシ、右側は横から撮影したもの

図 8 は、主基板とグラフィックカードを固定した状態のシャーシを示したもので、図 4 とは逆の位置からの撮影になる。

### 3.6. 動作チェック、その 1

この状態で電源ケーブルだけ接続し、主基板とグラフィックカードの動作を確認する。筆者は、動作チェックのためのスイッチボックスを秋葉原の PC パーツ店で購入した。

図 9 は仮接続した電源ケーブルの様子と、チェック用スイッチボックス（下の 2 個の白いスイッチが付いた黒い箱）を示している。左側の写真は、主基板へ結線したスイッチボックスからの配線で、電源 LED と RESET-SW、POWER-SW を結線している。HDD-LED は、IDE-HDD を接続していないこの場合は不要なので、結線していない。いずれもコネクタをピンに差し込むだけなので簡単ではあるが、それ故に抜けやすいため、取り扱い、特にスイッチ BOX を移動させるときには注意が必要である。また、主基板のメーカーによってピンの位置が異なるので、マニュアルをよく見て、間違いがないようにしておかなければならない。最近販売されている主基板では、接続先（どのケーブルを使うか）がよく判るように色分けされているものもある。

なお、この主基盤では、電源ケーブルを接続しただけで基板上の幾つかの赤い LED が点灯する。

同時に、AGP スロット（PCI のグラフィックカードであれば PCI スロット）にグラフィックカードを挿し、しっかり固定しておく。そして、グラフィックカードにはディスプレイを



接続しておき動作の確認を行う。

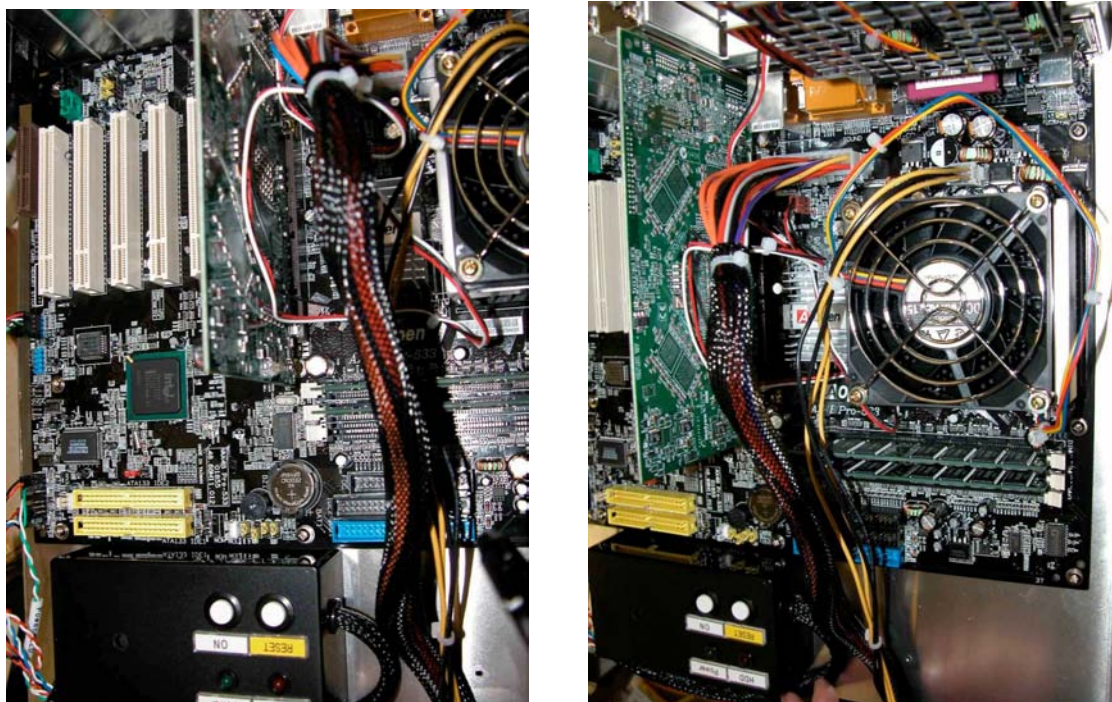


図9．仮接続した電源ケーブルと、動作チェック用 SW-BOX

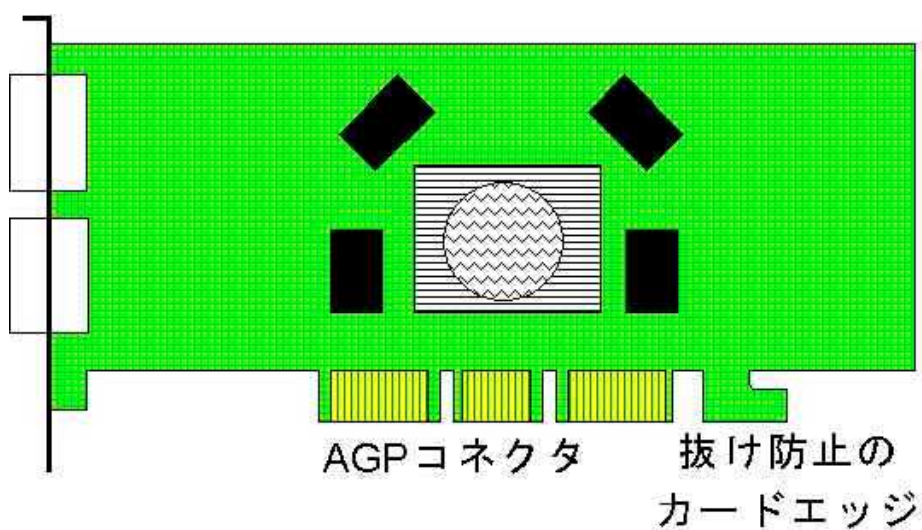


図10．AGP グラフィックカードの模式図

図10はAGP グラフィックカードの模式図を示したもので、左側にディスプレイ接続コネクタがある状態を示している。この方向から見てAGP コネクタの右側にあるL字型のカードエッジが、抜け防止のためのロックタブに嵌り込むようになっている。今回使用した主基板ではシーソー式になっているが、メーカーによってはピンを押し込み固定するものもある。

POWER-SW を押すと電源が入り、起動画面が表示されれば特に問題はないが、「ピー、ピ、ピ」と警告音が鳴る（殆どのメーカーで同じ鳴り方をする）場合、AGP スロットに巧く刺さっておらず、どこかがずれているので、きちんと挿し直す必要がある。図11の左の写真のよ

うに抜けロックが「カチッ」と音がするまではめ込んでやれば殆どミスはないと考えられる。図 11 の右側が今回使用したグラフィックカードで、奥のロットが AGP スロットで、向かって右側のロックタブが機能していることが判る。

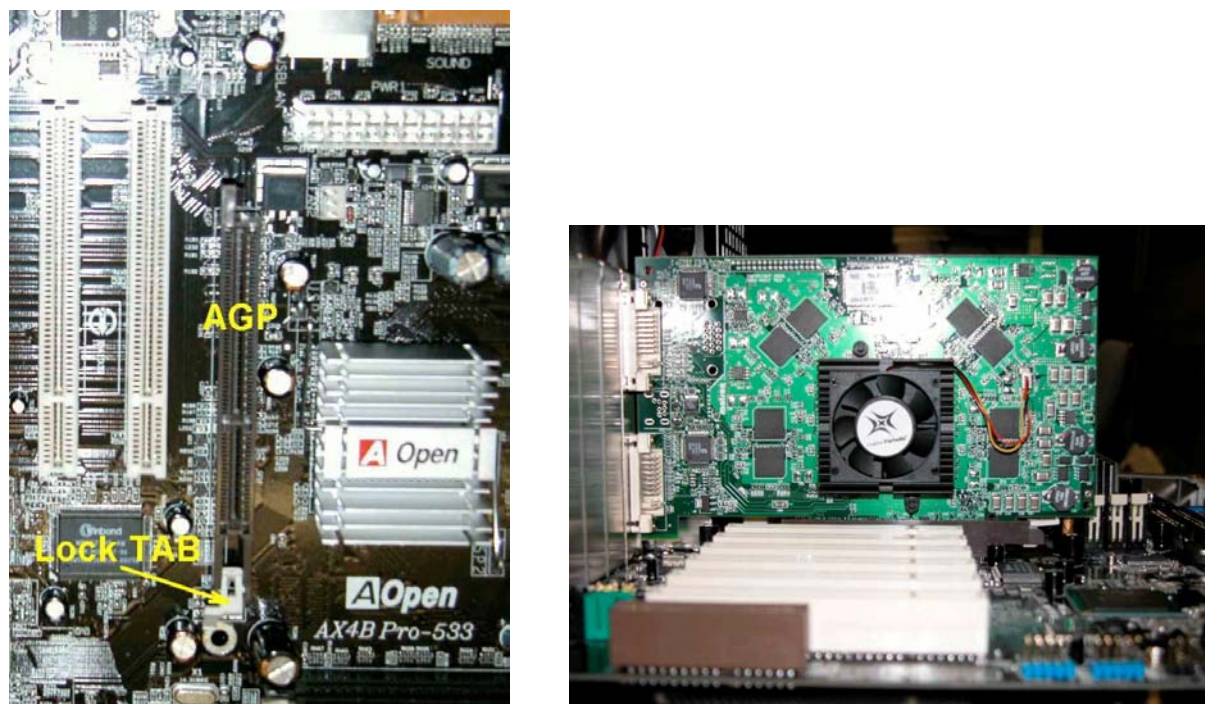


図 11 . AGP スロット (左) とグラフィックカード (右)

### 3.7. ドライブ類のシャーシへの取り付け

メインボードのチェックが終わったら、ドライブ類をケースに取り付けていく。このケースは奥行が十分にあり、また、メインボードよりメインボード取り付けシャーシの方が大きいので、先にドライブ類を取り付けても問題はないが、Micro-ATX や、新システムの教育用 PC の様な Book タイプのケースであれば、メインボード>メインボードのテスト>ドライブ類取り付け>総合テスト>OS インストールとなる。

ドライブ類のベゼル(プラスチック製の所謂「側」)が、ケースの開口部より微妙に大きい場合は外から挿入、そうでない場合は内側から取り出すように取り付けると、ドライブやケースのフロント部に傷が付かない。逆に無理をすると、確実にどちらかに傷を付けてたり塗装を剥がしたりしてしまい、後悔することになる。

また、CD-ROM や DVD-ROM , FDD を取り付ける場合、3mm の「ミリねじ」を用いるが、HDD (ケースに固定する電源を含む) を取り付ける場合、「インチねじ」と呼ばれる「ミリねじ」よりピッチの荒いものを用いなければならない。このドライブ取り付け規格の違いは、HDD の設計がインチがスタンダードのアメリカでなされたものに対して、FDD , CD-ROM , DVD 等のドライブの設計がメートル法を採用している日本やヨーロッパ諸国で行われたことに起因している。いずれにしても、ねじの種類を間違えるとねじ山を潰してしまい、取り付け・取り外しが困難になるので注意を要する。

閑話休題、筆者が作成した PC には、DVD-ROM , CD-R/RW , FDD , MO の 4 種のドライブ



と、60GB、120GBの2台のHDDが内蔵されている。このケースには、5個の5インチベイ、2個の3.5インチベイが外部からアクセスできるように開口部があり、ほかに4台のHDDを内蔵できる隠し3.5インチベイが準備されている（最下段のベイを使用しないときは6台まで可）。最も、3台以上のHDDを内蔵する必要がある場合は、外付けATAカードを増設したりRAIDを組むような特殊な例であり、通常の使用では、システム用とデータ用にそれぞれ1台ずつ計2台までで十分であろう。

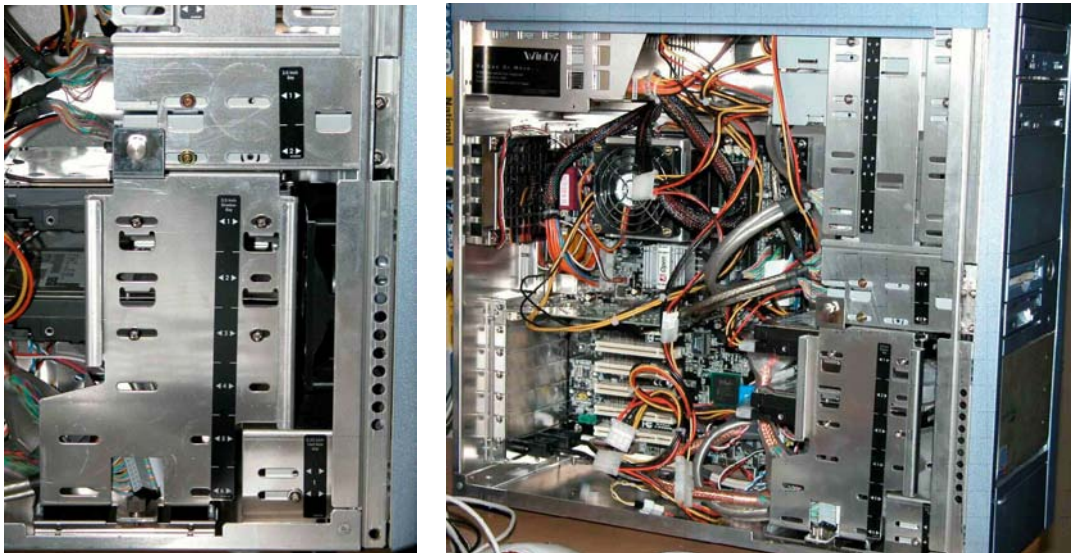


図 12 . 3.5 インチベイ（左）と稼働中の本 PC の内部（右）

図 12 は、ドライブベイにドライブを固定した様子を示したもの（原稿を書いているときにケースの側板を開けた状態）である。上の方に DVD-ROM、CD-RW ドライブ、中程に MO と FD のドライブ、その下に HDD を取り付けている。

### 3.8. ドライブ類の設定

IDE の規格では各ポート（ケーブル）に 2 台以下のデバイスを接続できることになっており、通常のメインボードには Primary と Secondary の 2 ポートがあり、それぞれに Master、Slave の設定がある。通常、Primary の Master には HDD を、Secondary の Master には CD-ROM（DVD-ROM）を接続する。最近のメインボードでは Primary が Ultra-ATA 133 に対応しているので転送速度を要求する HDD の接続に利用する。また、「起動は Primary の Master から」が暗黙の了解であるので、そのように接続するのが一般的である。また、MO や CD-R などの比較的転送速度が遅いデバイスでは、ATA-33 で十分間に合うので Secondary を使用する。

本稿で使用しているメインボードは、これらのデバイスの他に ATA-133 コントロールチップが搭載されており、このコントローラの機能で Ultra-ATA 133 の転送速度を持つデバイス（HDD）が 4 台接続できる。筆者は、このコントローラの 2 個のチャンネルに 1 台ずつ HDD を接続している。60GB のドライブは起動ドライブとするため 1 番チャンネルに、120GB のドライブは 2 番チャンネルにそれぞれ Master として接続した。また、メインボードの Primary の Master には DVD-ROM ドライブを、Secondary の Master には CD-R/RW ドライブ、Slave には



MO ドライブを接続している。

余談ではあるが、SCSI 接続の場合、ID 番号を設定するだけで 7 台～15 台のデバイスが同時に接続できる。

### 3.9. 電源ケーブル類の配線

ドライブ類のシャーシへの固定が終わったら、電源とデータのケーブル類の配線を行う。ケースに固定されている ATX 電源から、「赤、黒、黒、黄」がまとまって配線されているケーブルがドライブ用の電源ケーブルである。これを、各ドライブのコネクタに差し込めば電源の配線は終わりになる。メインボードへは、メインボード用電源コネクタと、ATX -12 と呼ばれる黄と黒が各 2 本まとめられたケーブルを挿すようになっている。メインボードによっては、さらに「P2 電源」と呼ばれる 5 本の線が纏まったコネクタも接続する必要がある。なお、ATX12 が必要な CPU は Intel の Pentium 4 のみで、AMD の CPU を使用する場合は接続は不要である（というよりコネクタがない）。

そのほかに、メインボードに 3P の白いコネクタがいくつかあり、そこには「FAN」と印刷されている。ここに、ケースに固定されているファンや CPU ヒートシンクに付属のファンの電源ケーブルを差し込む。

ケースから出ている電源 LED、Reset-SW、PPOWER-SW、HDD-LED のコネクタをそれぞれメインボードの該当するピンに差し込んでいく。ピンの位置に関しては 3.6 の動作チェック用 SW-BOX の項目を参照されたい。

### 3.10. データ転送用ケーブルの配線

HDD のデータをメインボード上の CPU に伝送するデータ用のケーブルの配線を行う。は ATA ケーブル (IDE ケーブルとも呼ばれる) を使用する。本稿の場合では、全て IDE (ATA) 接続であるので、ATA ケーブルを使用する。もし、SCSI 機器を内蔵する場合、SCSI カードと SCSI 機器を結ぶ SCSI ケーブルが必要になる。

さて、本稿で使用したデータ転送ケーブルは、FD 用、ATA 用共にメインボード付属のリボン状の「フラットケーブル」ではなく、「スマートケーブル」と呼ばれるものを使用している。スマートケーブルは、フラットケーブルに比較して専有面積が狭いため、ケース内を広くとることができる上、ケースに取り付けられた全面・背面ファンの空気の流れを阻害しないという利点がある。

図 13 は、IDE フラットケーブルとスマートケーブルの例を示したもので、中のものは ATA-33 ケーブル、下のものは ATA-133 ケーブルである。ここには示していないが、ATA-133 フラットケーブルや、SCSI、Ultra160-SCSI のスマートケーブルも存在する。

ATA-133 ケーブルと ATA-33 ケーブルの違いは、ツイストペアになっているか、単なるケーブルの集合かの違いで、ATA-133 ケーブルの方が高速なデータ通信に対応している。そのため、ATA-133 ケーブルの方が価格が高い。また、コネクタは黒、青、灰の 3 色に色分けされている。これは、青がメインボードのコネクタ、黒が Master、灰が Slave に設定されるようになっており、HDD や各種ドライブ類のジャンパピン設定を「Cable Select」に設定すると、

つないだコネクタの位置で、Master と Slave を自動設定してくれる。

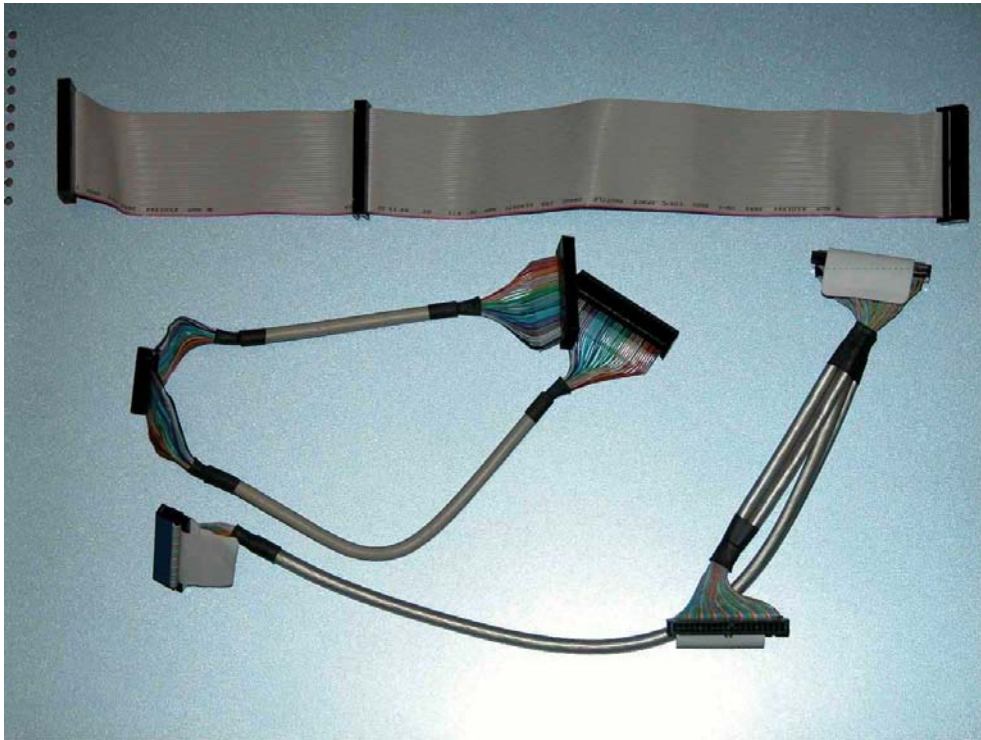


図 13．フラットケーブル（上）とスマートケーブル（中，下）

### 3.11. 動作チェック，その2

ケーブル類が全て接続でき，結線を確認し大丈夫であれば，まず，のコンセントからの電源ケーブルをケース電源のコネクタに差し込み，スイッチを「I」にする．正しく接続できていれば，幾つかのメインボード基板上のLEDが点灯する．そして，ケースの電源スイッチを押し，電源を入れる．

これまで間違いがなければ，「ポッ（ピとかプーの場合もある）」とブザー音がし，BIOS 起動画面が表示されればハードウェアの動作チェック完了である．

万が一，電源を入れたときに「ボスッ（バチッ）」と音がし，青い火花が飛んだ場合，すぐに電源を抜き，音がした部品を点検する．このようになった場合，ハンダ付けの不良でショートしている事がほとんどである．当然のことであるが，部品を購入した店（業者）に症状を報告し，該当部品を交換してもらわなければならない．その部品を取り外して再度電源を入れたとき，BIOS 画面が表示されなければ，メインボードまで壊れた可能性が高いので，やはり，交換してもらおうしか手はないだろう．

筆者の場合は，ケース用ファンのハンダ付け不良が原因で火花が飛び，電源を取っていたメインボードが破壊された．このときは幸いにも保証期間中であったため，メインボード交換だけで済んだ．しかし，交換までの約2週間，そのPCに対して何もできず，空しい時を過ごす羽目になってしまった．不良品だったファンは，ハンダ付けの箇所を修正したら無事使えるようになったが，本来ならクレームと損害賠償の対象になるだろう．

### 3.12. ケースの組立

さて、無事起動することが確認できたら、いよいよケースの組立である。といってもケースを分解したときと逆の手順で、両側の側板をはめ込み、ネジ止めするだけである。

これで、自作 PC ハードウェア部分は完成したことになる。

### 3.13. その他

最近、青色や白色の LED が安価になってきて PC ケースにも採用されるようになってきたり、自分で交換できるパーツが販売されてきたりしている。このような LED を使用するとき、そのコネクタ間の電圧に注意しなければならない。

従来の LED は 1.5 ~ 2V で点灯しているが、青や白の LED は 3.6V の電圧がないと点灯しない。メインボードによっては、POWER ピン間に 3V 程度しか出さないものがあり、このような場合、正しい方向に LED を挿しても点灯しない場合がある。もし、電源 LED が青（白）で正しく接続して点灯しない場合、LED の故障を疑う前に電圧不足を考慮する必要がある。

## 4. ソフトウェアのインストールと設定

これまで主にハードウェア的の組立を行ってきたが、いよいよこれからメインボードの BIOS 設定、OS やアプリケーションのインストールを行っていく。

### 4.1. BIOS の設定と OS (Windows XP) のインストール

BIOS の設定は BIOS 設定プログラムを起動させて行う。一般的なメインボードでは「DEL」キーを押しながらリセットボタンを押して PC を再起動すれば、BIOS 設定画面が表示され各種 BIOS 設定が行うことができる。



図 14. BIOS 設定画面（左：起動時の画面，右：終了時の画面）

図 14 は、BIOS 設定の起動時と終了時の画面を示したもので、ほとんどのメインボードでは、青い画面に黄または白の文字が表示される。本稿で使用した AOpen の AX4B Pro-533 では、左側に設定項目が集中している。上から順に Standard CMOS Features, Advanced BIOS Features, Advanced Chipset Fratures, Integrated Peripherals, Power Management Setup, PnP/PCI Configurations, PC Health Status, Frequency / Voltage Control となっている。右側は、BIOS の設

定を元に戻す，BIOS 設定画面に入るためのパスワードをかける，BIOS 設定画面終了時のオプション設定等ができるようになっている．

以下，各設定項目について概説していく．なお，このセクションで使用している BIOS の設定画面は，同一メインボードを使用した別の PC であり，デバイス名等が異なっていることを予めお断りしておく．

#### 4.1.1. Standard CMOS Features

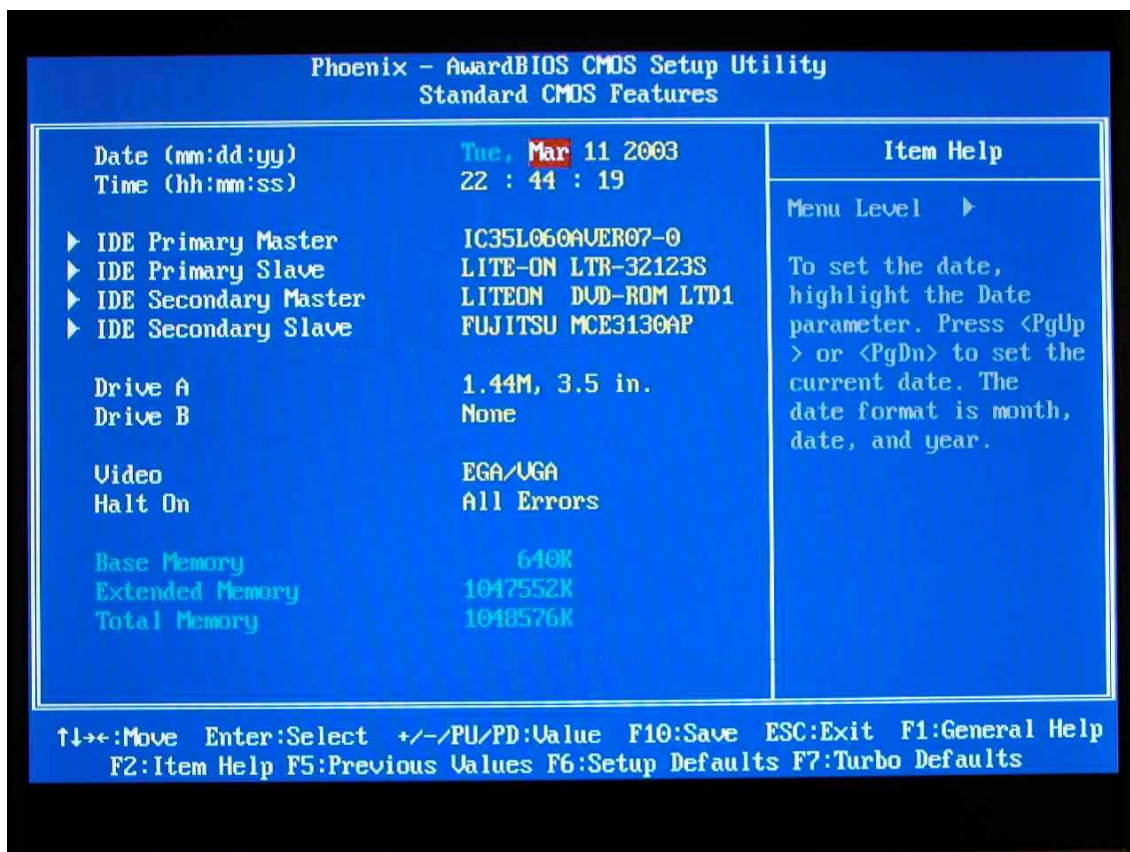


図 15 . Standard CMOS Features

図 15 は，Standard CMOS Features の設定画面である．この項目では，このメインボードに接続されているドライブ類のデバイス名，接続されている FDD の種類，BIOS カレンダの持つ日付，時刻の設定，エラーが起きたときの処理手順等が表示される．下部には使用するベースメモリ 640KB と 1047552KB の拡張メモリが表示されている．この数字は，MS-DOS の時代の 640KB のメインメモリとそれ以上の拡張メモリに分けて管理していた名残である．ちなみに，このメインボードには 1GB (1000KB) のメモリが搭載されており，もはや「640KB」という単位は，最早過去の遺物であろう．

#### 4.1.2. Advanced BIOS Features



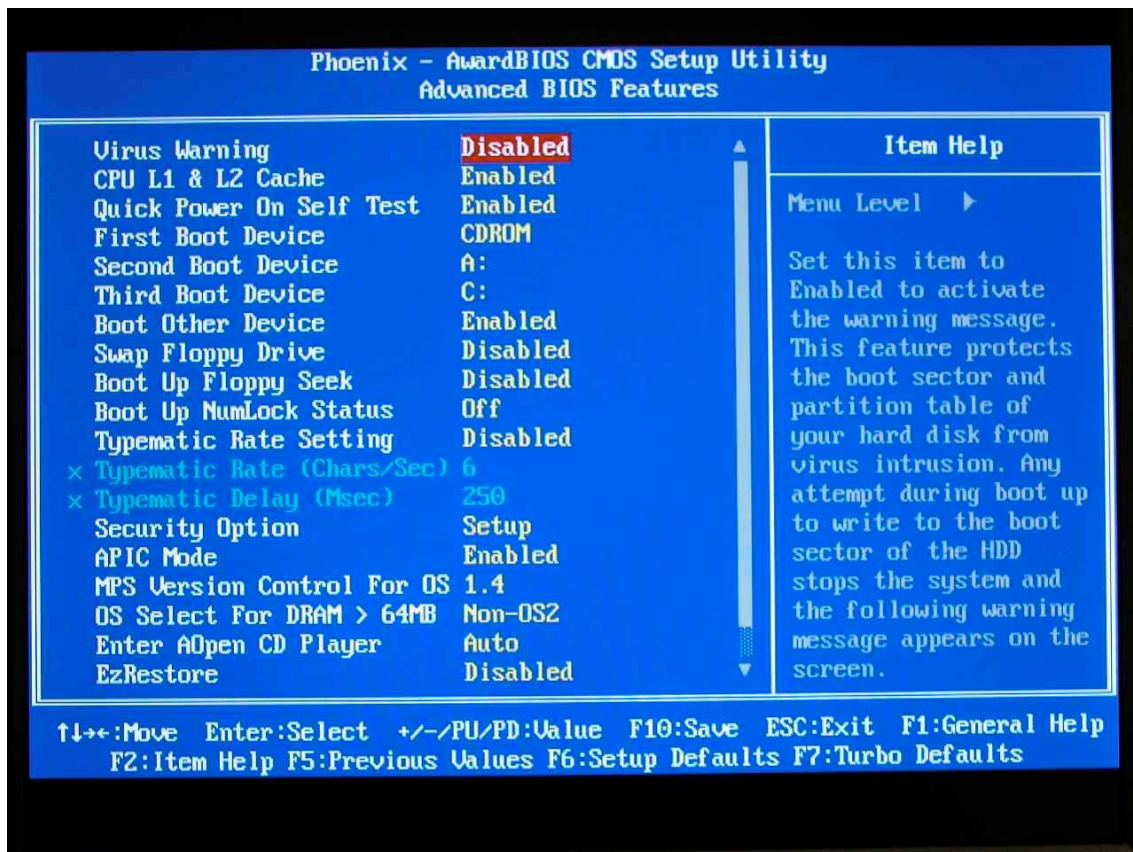


図 16 . Advanced BIOS Features

図 16 は、Advanced BIOS Features の設定画面である。上から BIOS によるウイルスチェック、L1&L2 キャッシュの設定、電源自己テスト、ブートデバイス（1 番目から 3 番目、その他）の設定、FD のドライブ名交換の設定、電源投入時に FDD をシークしに行くかの設定、Num Lock（テンキーを数字キーとして認識させる）の設定、以下、その他の機能の設定になっている。

BIOS によるウイルスチェックは、ウイルスチェックのパターンが時代にそぐわないものとなってしまう、現実的に機能できない、Disabled の設定にしている（特に必要のない限り OFF でかまわない）。また、L1, L2 キャッシュを切る(Disabled にする)ことは、Pentium 4 CPU の性能をほとんど殺すことになるので、必ず Enable にしておく必要がある。この PC では、ブートデバイスは、IDE 接続の CD-ROM、FDD、IDE 接続のプライマリのマスター-HDD の順にしている。ブート可能な SCSI 機器が接続されている場合、SCSI を選択することもできる。また、通常電源投入時には、キーボードの Num Lock 機能（キーボード右側のテンキーが数字のキーとして機能させること）は機能しない。これをやめ、テンキーを矢印キー等ではなく数字キーとして使うため、Boot Up NumLock Setting で Num Lock キーを押す設定を行う。

これ以外の設定の変更は、ほとんど必要ないので省略する。

#### 4.1.3. Advanced Chipset Features

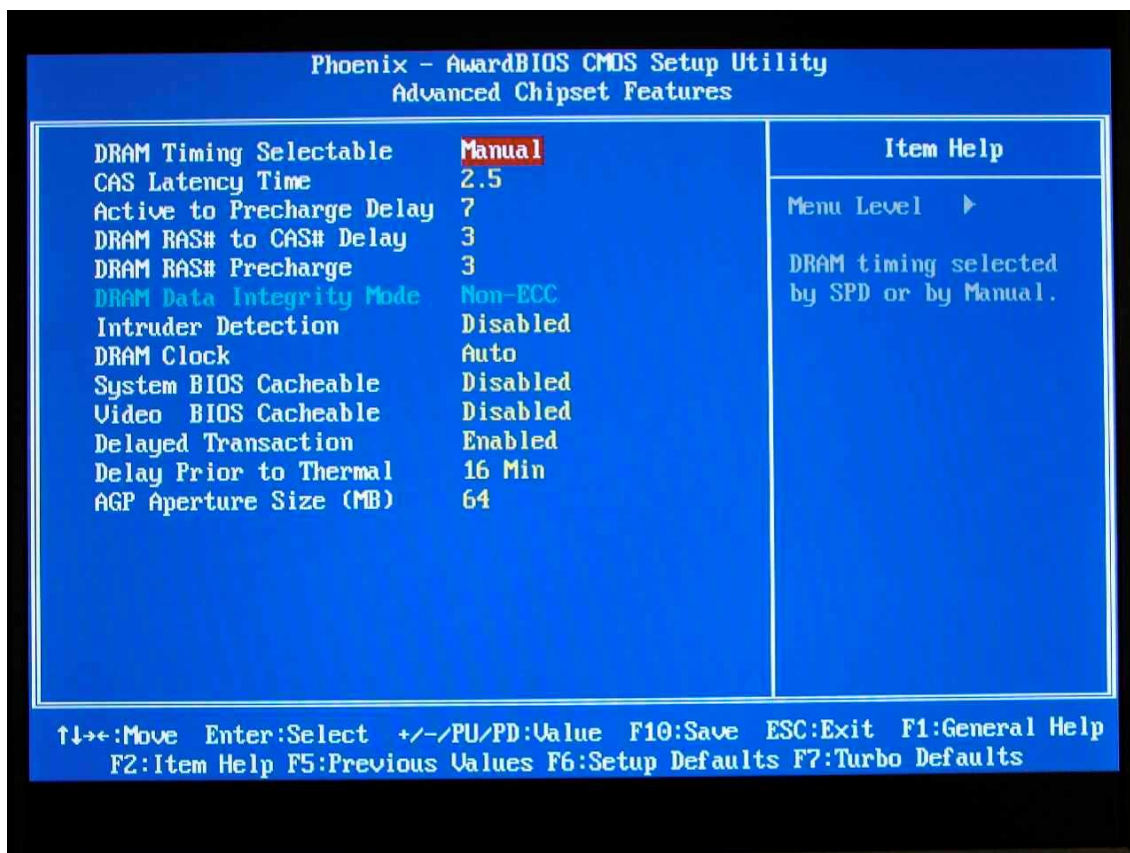


図 17 . Advanced Chipset Features

図 17 は、Advanced Chipset Features の設定画面である。ここでは、RAM のタイミングや AGP バスの設定を行う。

この PC では、マニュアルで設定している DRAM Timing は、「By SPD」とする方が一般的である。「By SPD」と設定したとき、CAS、Precharge Delay、CAS# Delay、RAS# Delay は自動で設定される。また、DRAM Clock の設定は、PC-2700 DRAM を PC-2100 の設定で使用するときなどに使用する。

#### 4.1.4. Integrated Peripherals

図 18 は、Integrated Peripherals の設定画面を示したもので、メインボード上に搭載されている各種チップのセットアップを行う。

基本的に Auto または Enable と設定すればよいが、IDE 接続デバイスが Primary だけだったり、IDE 接続のデバイスが全くない全てのドライブを SCSI 接続で構成した場合などでは On-Chip Primary (Secondary) PCI IDE は Disable にしておくと、割り込み IRQ が空き、別のカードの IRQ 要求に応えることができる。

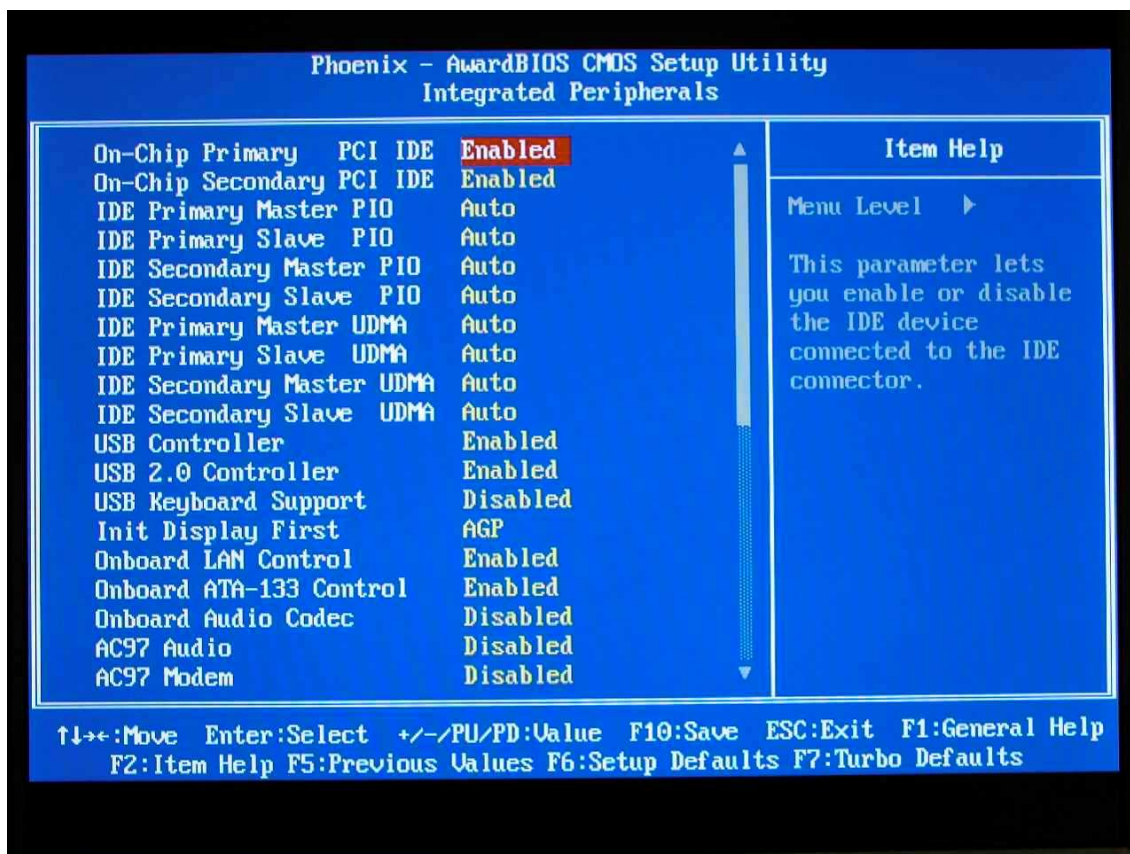


図 18 . Integrated Peripherals

USB ,USB 2.0 コントローラは USB 機器を使用するために必要であるため ,Enabled と設定 ,  
 ただし USB キーボードは使用しないので ,ここは Disabled に設定した .

Init Display First の項目は ,グラフィックカードを AGP スロットに挿しているものと PCI  
 スロットに挿しているものがあるとき ,どちらを優先するかを選択である .通常は AGP スロ  
 ットにグラフィックカードを 1 枚挿すので ,AGP に選択しておけばよい .また ,オンチップ  
 でグラフィック機能が提供されている場合も同様である .

LAN と ATA-133 のコントローラはいずれもオンボードコントローラの機能を使用するので  
 Enabled としておく .もし ,不要 (使用しない) であれば Disabled にしておく .

また ,この PC にはサウンドカードを増設しているため ,オンボードのサウンド機能は必要  
 ないので Disabled にしている . Modem 機能も使用していないので Disabled に設定している .

この画面以外にもゲームポートや MIDI 音源の割り当ての設定が用意されているが ,いず  
 れもサウンドカードの機能を使用するため Disabled にしている .もし ,オンボードの機能を  
 使う場合 ,いずれの設定も Enabled に変更しなければならないのはいうまでもない .



#### 4.1.5. Power Management Setup, PnP/PCI Configurations



図 19 . Power Management Setup ( 左 ) と PnP/PCI Configurations ( 右 )

図 19 は Power Management Setup と PnP/PCI Configurations の画面である . Power ~ は電源の管理を PnP ~ はプラグアンドプレイ , PCI スロットの設定を行っている . どちらも , 特に必要なければ初期設定で問題はない .

#### 4.1.6. PC Health Status, Frequency / Voltage Control

PC Health Status は現在のメインボード上の CPU 温度やケース内部温度 , ファン回転数などを表示するもので , 特に高温な場所 ( 例えば直射日光の当たる南側の窓際など ) に PC を設置しない限り , 参照する必要はない . また , Frequency / Voltage Control では , ベースクロック周波数や , CPU にかかるコア電圧の設定を行うことができる . この項目も , 普段使用するに当たっては特に変更する必要はない .

#### 4.2. OS ( Windows XP ) のインストール

BIOS の設定が終了すると OS のインストールに取りかかることができる . 本稿では Windows XP Professional アカデミックアップグレード版 ( 以下 , XP と記す ) をインストールしていく .

CD ( DVD ) -ROM ( RW ) ドライブに Windows XP の CD-ROM をセットしリセットボタンを押すか , 電源を入れた直後にトレイを開けて CD-ROM をセットする . すると , このドライブからデータを読み込み始め , XP のインストーラが起動する . 通常の IDE ポートに接続した HDD や , SCSI 接続の HDD から起動する場合はあまり問題にならないが , 本稿で使用している Ultra ATA-133 チップが搭載されていて , これに接続したドライブを起動ドライブとするためには , デバイスドライバを予め FD にコピーしておく必要がある .

<http://www.aopen.co.jp/tech/faq/mb/serialata.htm> に起動ドライブとして使用する方法 , データドライブとして使用する方法が記載されている .

この方法をメインボードのマニュアルにある説明と上記 Web ページより転載したものを掲載する .



ULTRA-ATA133 チップドライバFDの作成法：

- ・ Windows でフォーマットしたFDを1枚用意する。
- ・ 別のPCを準備してメインボードに添付されているCDをエクスプローラ等で開く。
- ・ CD-ROMのドライブ名:¥Driver¥Promise¥ATA133¥Driver¥WinXP (Windows2000の場合は~¥WIN2000)のフォルダの内容を全てFDにコピーする。
- ・ コピーし終わったらFDを取出す。

以下にドライバのインストール方法を、<http://www.aopen.co.jp/tech/faq/mb/serialata.htm>より転載したものを示す。なお、Webページ上の図は省略し、読みやすいように文章を少し修正している。

1. Windows2000/XPのインストールCDを挿入し、CD起動にてコンピュータを起動させてください。
2. インストール画面の下方に「Press F6 if you need to install a third party SCSI or RAID driver...」と表示されたら、「F6」キーを押してください。
3. その後、「S」を押してください。
4. FD要求メッセージが表示されたら添付のドライバフロッピーディスクを挿入し「ENTER」キーを押してください。
5. 選択が表示されますので、WindowsXPの場合「WinXP Promise Sata(tm) Controller」を選択し「ENTER」キーを押してください。(デバイス名はバージョンによって異なる場合がございます。)
6. Windows2000の場合はキーボードの十字キーにて下方へスクロールしていただき、「Win2000 Promise Sata (tm) Controller」を選択し、「ENTER」キーを押してください。(デバイス名はバージョンによって異なる場合がございます。)
7. 下記画面が表示されたら「ENTER」キーを押してください。(デバイス名はバージョンによって異なる場合がございます。)
8. あとは通常と同じようにインストールが開始されます。

以上、転載終了。ULTRA-ATA133チップを起動ドライブにすることは、自作を初めてする人にとって敷居が少々高いように思われる。また、デュアルOSブートも可能になるので、BIOSの設定方法も同社のWebページから転載(読みやすいように修正済み)する。

関連BIOS設定項目について：

複数のハードディスクを装着する場合はBIOS設定にて起動優先順位を指定する必要があります。起動順位の設定は初期状態では下記のようになっております。

Advanced BIOS Features >

First Boot Device > CD-ROM

Second Boot Device > A

Third Boot Device > C

例えば下記のように HDD を接続した場合、BIOS 設定を変更しなければなりません。

ATA133 ソケット(黄) Windows XP インストール済みハードディスク

ATA100 ソケット(青) Windows Me インストール済みハードディスク

Windows XP を起動させるなら

Advanced BIOS Features >

First Boot Device > CD-ROM

Second Boot Device > SCSI

Third Boot Device > C

Windows Me を起動させるなら

First Boot Device > CD-ROM

Second Boot Device > C

Third Boot Device > SCSI

つまり ATA133 ソケットに接続したハードディスクから起動させたい場合は、「SCSI」の起動順位を「C」よりも上にしてください。ATA100 ソケットに接続したハードディスクから起動させたい場合は、「C」の起動順位を「SCSI」よりも上にしてください。

<http://www.aopen.co.jp/tech/faq/mb/mbglo.htm> に用語解説があるので、これも参考にさせていただきたい。

さて、CD から BOOT すると、最初にハードディスクを特別な (Windows XP に標準で搭載されない) デバイスドライバに接続しているかどうかの旨のメッセージが表示される。本稿の場合、上記手法で作成したドライバを使用するため、「特別なデバイスドライバが必要な場合」に相当する。このとき、指定されている時間内に F6 キーを押してドライバが必要なことをプログラムに知らせる必要がある。最後のデバイスドライバを読んだ後、「FDD にデバイスドライバの FD を入れるように」というメッセージが表示されるので、そのデバイスドライバをコピーした FD を入れ、「S」キーを押す。すると、該当するドライバが読み込まれ、Windows XP のインストールが続行される。

これ以降は、News Letter43 に書かれている通りであるのでそちらを参照していただければ幸いである。

#### 4.3. ドライバのインストール

最新のグラフィックカードを購入して取り付けると、Windows XP に標準添付されているド

ライバでは十分に能力を発揮できない。そのため、ほとんどのビデオカードにはCD-ROM ディスクが添付されており、「専用デバイスドライバ」や様々なアプリケーションソフトが供給されている。また、製造元が Web ページを作成して最新のドライバを供給していることも多い。

本稿で使用している PC のグラフィックカードは、Matrox 社製 Parhelia 128 であるが、<http://www.matrox.com>のサイトの support にある Latest Drivers の項目に最新のドライバが登録されている。最新のドライバは、様々なバグフィックスが行われる他、隠し機能が有効になっていたり、描画が速くなっていたりすることが多い。

ダウンロードした圧縮されたドライバのファイルを解凍後、インストールプログラムを起動し、データがインストールされた後再起動すれば、最新のデバイスドライバに置き換えることができる。

また、サウンド関係やネットワーク等のメインボード上に搭載されたチップのドライバは、メインボードの会社の Web ページに登録されていることが多い。

#### 4.4. アプリケーションのインストール

##### 4.4.1. IME, ワードプロセッサ等の市販アプリケーションのインストール

ほとんどのアプリケーションソフトは、マニュアルの手順通りに進めていけば、インストールの失敗はない。大抵の場合、「すべて」、「限定」、「カスタマイズ」のインストール区別があり、必要に応じて選択できるようになっている。

本稿では、InstallShield を基に作られているインストーラの例(Justsystem の IDisk マネージャ、Netnote)を挙げておく。



図 20 . セットアップの起動画面

図 20 が起動画面で , Autorun 機能を有効にした CD(DVD)-ROM(R/RW)のドライブに CD または DVD を挿入すると表示される .



図 21 . コピー開始のメッセージ画面

図 21 の画面の時にインストール時の設定の変更を行えるソフトもある .



図 22 . コピー先のフォルダの設定

図 22 は、コピー先のフォルダ設定画面である。殆どのソフトはデフォルトで C:\Program Files の下に作成されるようになっているが、インストール先のドライブを変更したり、フォルダ名を変更する場合、この画面の「参照(R)」ボタンを押し、適宜変更すればよい。

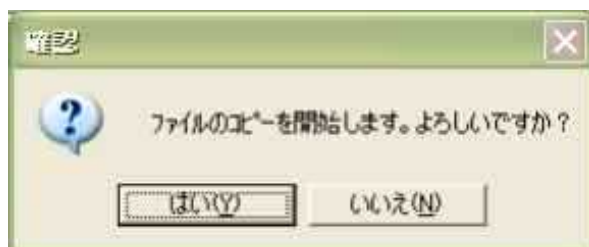


図 23 . 開始確認画面

思い通りの設定になっていれば、図 23 のコピー開始確認画面が表示されるので、「はい」を押せば、ファイルのコピーが開始され、図 24 の状況バーが表示される。ファイルがアーカイブされていて、インストール時に展開するときには、バーが 2 段で表示されることもある。



図 24 . インストール実行状況を知らせるバー

図 24 のバーが右端(100%)になり、インストールが終了すると、セットアップ完了画面(図 25)が表示され、インストールが終了する。アプリケーションソフト、特にデバイスドライバや Windows の設定を変更するようなソフトでは、この後、再起動するか否かを尋ねる場合があるので、起動中のソフトを終了してから、「完了」ボタンを押すようにすればよい。



図 25 . セットアップ完了画面

筆者注：図 20 から図 23 は IDisk，図 24 と図 25 は Netnote の画面のハードコピーである。

#### 4.4.2. ユーティリティ等のアプリケーションのインストール

基本的には 4.4.1 と同じであるが，ソフト（オンラインで流通しているフリーソフト等が殆ど）によっては，自分でショートカットを作成し，スタートメニューの登録しなければならないものがある。

この場合，まず該当するプログラムファイルのアイコン上でマウスの右ボタンを押し，メニュー（いわゆる右クリックメニュー）を表示させ（図 26，図 27 に拡大画面），この中の「送る」を選択すると，様々な送り先（"C:\Documents and Settings\Default User\SendTo" や "C:\Documents and Settings\ユーザー名\SendTo" フォルダに設定されている内容）のアイコンが表示される。このときに「デスクトップ（ショートカットを作成）」を選択すれば，デスクトップに該当する実行ファイルのショートカットが作成できる。

次に，スタートメニュー > 設定 > タスクバーと [スタート] メニューを選択し，「スタートメニュー」のプロパティ画面（図 28）を表示させ，カスタマイズのボタンを押し，カスタマイズ設定画面（図 29）を表示させる。

その後，「メニューの詳細設定ボタン」を押すと，図 30 に示すような画面が表示される。ここに，先ほどデスクトップ上に作ったショートカットアイコンを移動すれば，スタートメニューに該当アプリケーションソフトが登録される。

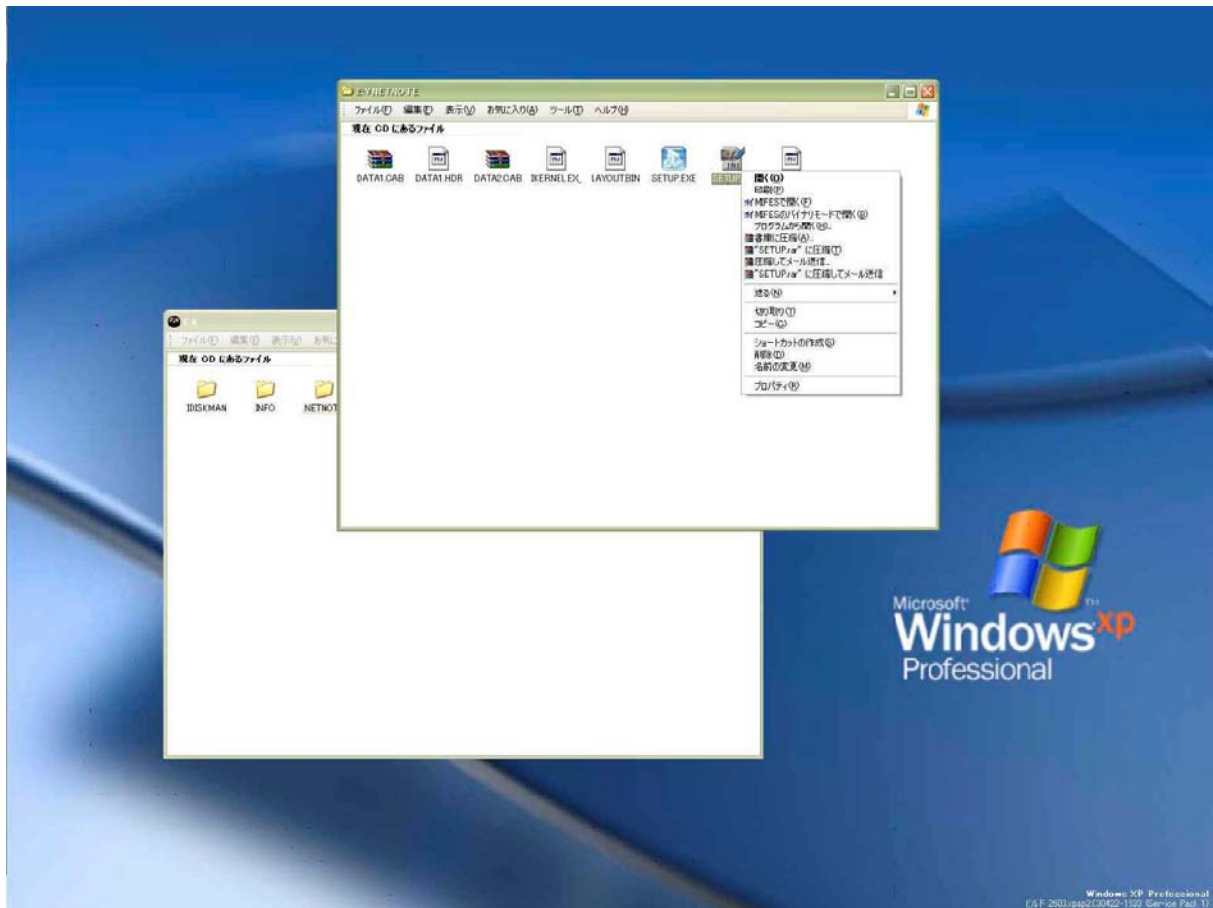


図 26 . 右クリックメニューを表示させたデスクトップ



図 27 . 右クリックメニューの拡大図





図 28 . スタートメニューのプロパティ画面



図 29 . スタートメニューのカスタマイズ画面



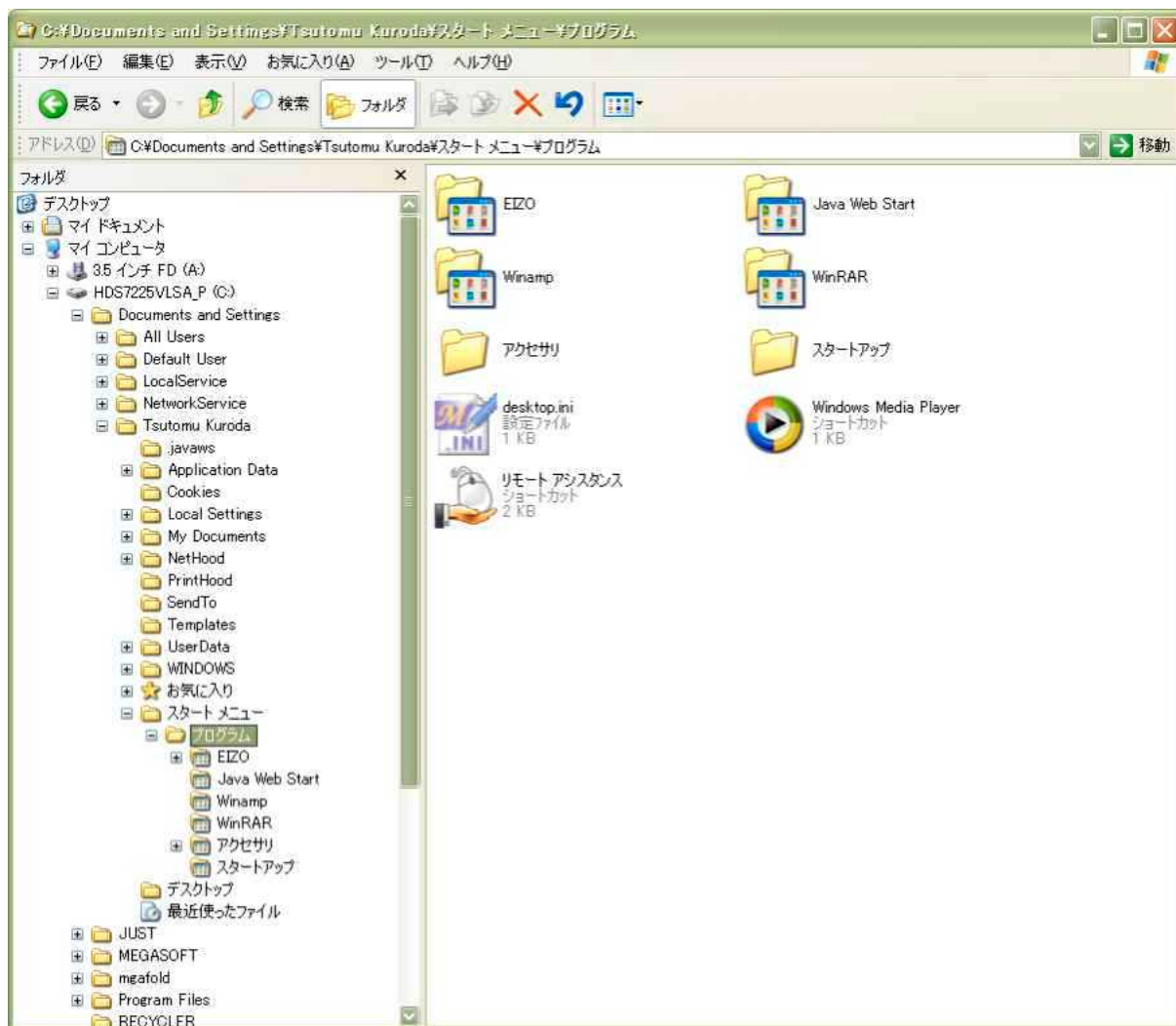


図 30 . プログラムフォルダの詳細

ちなみに、マイコンピュータから"C:\Documents and Settings\ユーザ名\スタートメニュー\プログラム"を選んでフォルダを展開させたものと同様である。

#### 4.4.3. マルチディスプレイの設定

本稿で使用した Matrox Parhelia 128 (後で発売された Millenium P750 も同様) は、ディスプレイを 3 台同時に使用するトリプルディスプレイの設定も可能である。しかし、このような環境が必要になるのは稀であり、トリプルディスプレイにするには、アナログ接続ディスプレイ 3 台、もしくはデジタル接続ディスプレイ + アナログ接続ディスプレイ 2 台になるため、よほど広い部屋と机がないとアナログ接続ディスプレイ 3 台は難しい。また、LCD の様にデジタル接続ができるディスプレイでは 2 台同時に使用したデュアルディスプレイの設定のみが可能である。

本稿では、デジタル接続 LCD 2 台を接続したデュアルディスプレイの設定について述べておく。VGA カードによっては、1 台目がデジタル接続、2 台目がアナログ接続になっているものや、両方ともアナログ接続のものもあるが、デュアルディスプレイでの設定は、ほぼ同

じである。

デスクトップの何も無いところ（アイコン以外の場所）にマウスカースルを持って行き右ボタンをクリックすると図 31 の「画面のプロパティ」が表示される。



図 31 . 画面のプロパティ

デュアルディスプレイの設定が ON になっていると、ここで、2 つの画面の設定ができる。ここで「識別」ボタンを押すと画面に「1」、「2」の数字が大写しになる。「詳細設定」ボタンを押すと VGA カードとディスプレイのプロパティが表示され、種々の設定が可能になる。VGA カードメーカーのユーティリティがインストールされている場合、ここにメニューが追加されているはずである。本稿の場合、Matrox が提供している「Power Desk」のメニュータブが追加されている（図 32）。

ここで、「Matrox PowerDesk-HF」のボタンを押すと、に示す画面が表示され、ドライバの詳細な設定が行えるようになる。このプログラムを起動するには、個々に述べた方法のほか、タスクバー（図 34）の該当プログラムのアイコンをダブルクリックする、コントロールパネ

ルにある該当プログラムのアイコンをダブルクリックする方法がある。

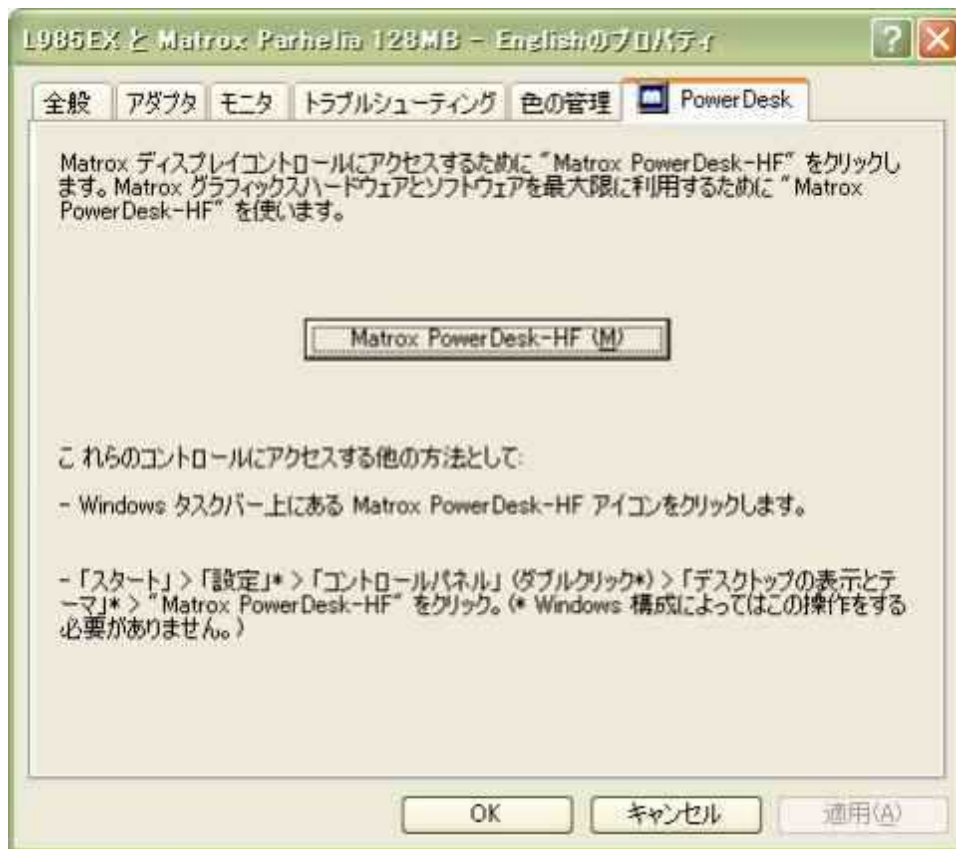


図 32 . ドライバのプロパティの詳細



図 33 . メーカー作成の設定プログラム



図 34 . タスクバーのアイコン (右側, 左から 3 番目が設定プログラムのアイコン)

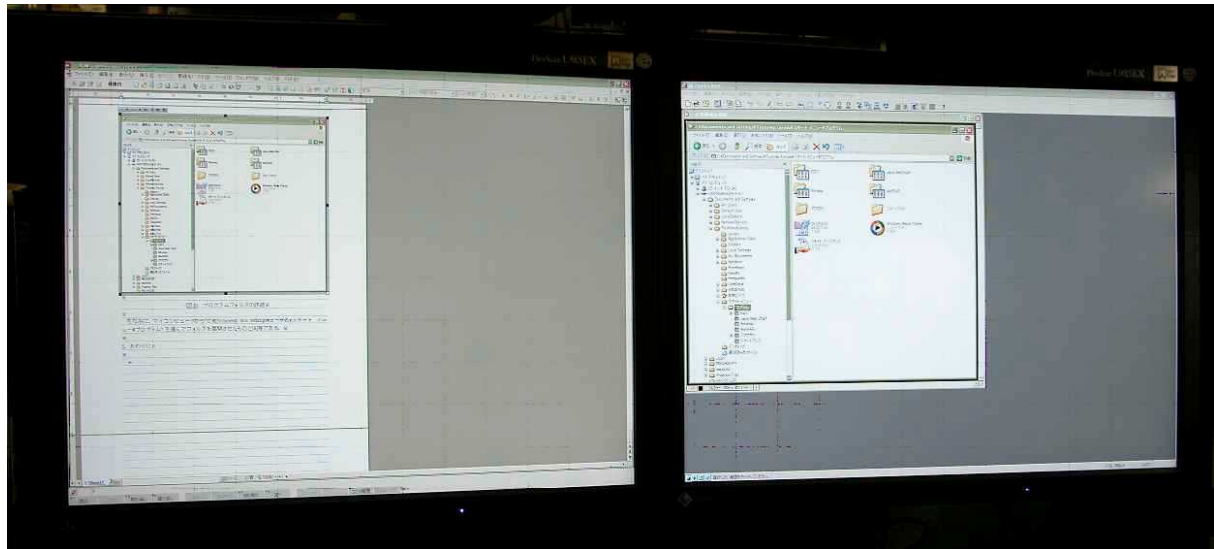


図 35 . デュアルディスプレイの表示例

図 35 は, デュアルディスプレイの表示例 (3200×1200 ドット, 32bit Full Color) であるが, デジタル接続が 1600×1200 ドットより上の解像度をサポートしていないので, LCD を用いた最高解像度ということになる.

## 5. おわりに

初めて組み立てパソコンに挑戦してみようという方はもとより, 2 台目を組んでみたけどどうも..., という方でも参考にできるように, なかなか参考資料が見つからない BIOS や初期設定関係に力を入れたつもりであるが, 筆者の拙い文章力ではこれが限界である. これを機に, 何かアップグレードを図りたい, 設定をいじってみたいという方が一人でも増えれば幸いであると考えている.

また, 判りづらい点があったり, 事実誤認等を発見した場合, kuroda@ed.kagawa-u.ac.jp に連絡をいただければ幸いです.

# ウイルスチェックサーバによるアラートメールの集計

青木 昌三<sup>\*</sup> 瀬野 芳孝<sup>\*\*</sup>

曽根 計俊<sup>\*\*</sup> 高橋 多恵<sup>\*\*</sup>

## 1. はじめに

2003年3月の旧香川大学ネットワークシステム更新に伴い、2台のウイルスチェックサーバが導入され、2003年5月半ば過ぎよりその運用が開始された<sup>(\*)</sup>。2003年6月より同12月までの間にウイルスチェックサーバにより除去されたメール件数やウイルスの種類について集計した結果を報告する。

(\*)ウイルスチェックサーバの設定等の詳細については、本年報の別掲記事「香川大学におけるウイルス対策メールシステム」を参照のこと。

## 2. 管理者向けアラートメールの集計

2台のウイルスチェックサーバは学内および学外から送られてくるメールを事前にスキャンし、もしウイルスがあればその除去等を行うと共にメールの送信者、受信者、管理者にアラートメールを送信する。この管理者に送信されるアラートメールを解析し、集計用データを蓄積すると同時にその内容を表示する管理者向けWebページを自動作成する環境を構築した。

### (1) 管理者向けアラートメールの書式変更

ウイルスチェックサーバ導入時点での管理者に送信されるアラートメールは次のような書式・内容であった。

```
Have detected a virus (WORM_SOBIG.A) in mail traffic. Action: quarantined.
```

上記の内容ではウイルス名と処理された方法しか分からないことから、アラートメールのオプションをすべて選択し、次の内容・書式とするよう設定を変更した<sup>(\*)</sup>。

```
The uncleanable file CAGXI78D.scr is moved to
/etc/iscan/virus/virFfqR7V ;quarantined ;06
/13/2003 14:35:36 ;CAGXI78D.scr ;abc@aaa.ccc.jp ;WORM_KLEZ.H ;zzz@xx.kagawa-
u.ac.jp
;/etc/iscan/virus/virFfqR7V ;SMTP ;dragon.kagawa-u.ac.jp
```

(\*)この書式・内容の例では、送信者・受信者名を適宜編集している。

### (2) 管理者用Webページの作成

ウイルスチェックサーバが処理した週間のウイルス情報を管理者用のWebページとして自動作成するスクリプトを作成した。このWebページは管理者用のため、閲覧するにはアカウントとパスワード入力が必要なように設定している。以下に、管理者を対象とした「週間ウィ

---

\* 教育学部

\*\* 総合情報基盤センター幸町分室

ルスレポート」Web ページを示す（図 1 参照）。

年/月/日	ウイルスの件数	備考
2004/01/06	16	<a href="#">詳細</a>
2004/01/05	19	<a href="#">詳細</a>
2004/01/04	23	<a href="#">詳細</a>
2004/01/03	9	<a href="#">詳細</a>
2004/01/02	9	<a href="#">詳細</a>
2004/01/01	11	<a href="#">詳細</a>
2003/12/31	19	<a href="#">詳細</a>

図 1 : 「週間ウイルスレポート」の Web ページ画面

図 1 で、「[詳細](#)」をクリックすると、次のような書式でより詳しい内容を表示するようになっている(図 2 をも参照)。

ウイルスレポート (SMTP)						
日付 01/04/04						
日時	ウイルス名	送信元	送信先	処理	感染ファイル名	
01:46:44	WORM_KLEZ.H	tymer@mx.mmd.co.jp	s00z000@xxx.kagawa-u.ac.jp	quarantined	head_bk	
.....						

ウイルスレポート(SMTP)						
日付 01/06/04						
日時	ウイルス名	送信元	送信先	処理	感染ファイル名	
00:24:59	WORM_KLEZ.H	tymer@mx.mmd.co.jp	s00z000@xxx.kagawa-u.ac.jp	quarantined	head_bk	

図 2 : 「[詳細](#)」クリックにより表示される画面（一部）

### (3) ウィルスデータの集計

2 台のウィルスチェックサーバから送られてくるアラートメールの書式・内容は(1)の  
に示した通りであるが、スクリプトにより、これを更に集計し易いよう 1 日 1 回自動的に整  
理し、月毎のファイルとするようにしている。作成されたデータをもとにした集計結果を以  
下に示す。



(a) ウィルス除去数ベスト 10 (2003 / 6 ~ 12 月)

2003 年 6 月から 12 月までの間でのウィルスチェックサーバにより除去されたウィルス数について、7 ヶ月間の合計数が多い順に上位 10 個のウィルス (その他を含む) をまとめたものをリスト 1 に、関連したクラブを図 3 ~ 図 6 に示す。

リスト 1 : ウィルス除去数ベスト 10

	6	7	8	9	10	11	12	計
WORM_KLEZ.H	165	360	409	301	277	355	646	2513
WORM_SWEN.A	0	0	0	361	422	196	180	1159
WORM_MIMAIL.I	0	0	0	0	0	0	987	987
PE_BUGBEAR.B-O	76	187	90	244	99	6	19	721
WORM_MIMAIL.A	0	0	247	11	57	40	92	447
WORM_SOBIG.F	0	0	127	124	1	0	0	252
WORM_MIMAIL.F-1	0	0	0	0	0	224	27	251
PE_BUGBEAR.DAM	0	69	38	4	12	2	2	127
EMail_Flaw_MIME_Tag_Overflow	7	14	15	5	16	5	17	79
その他	102	74	56	25	18	98	67	440
計	350	704	982	1075	902	926	2037	6976

ウィルス除去数全体推移(2003/6 ~ 12)

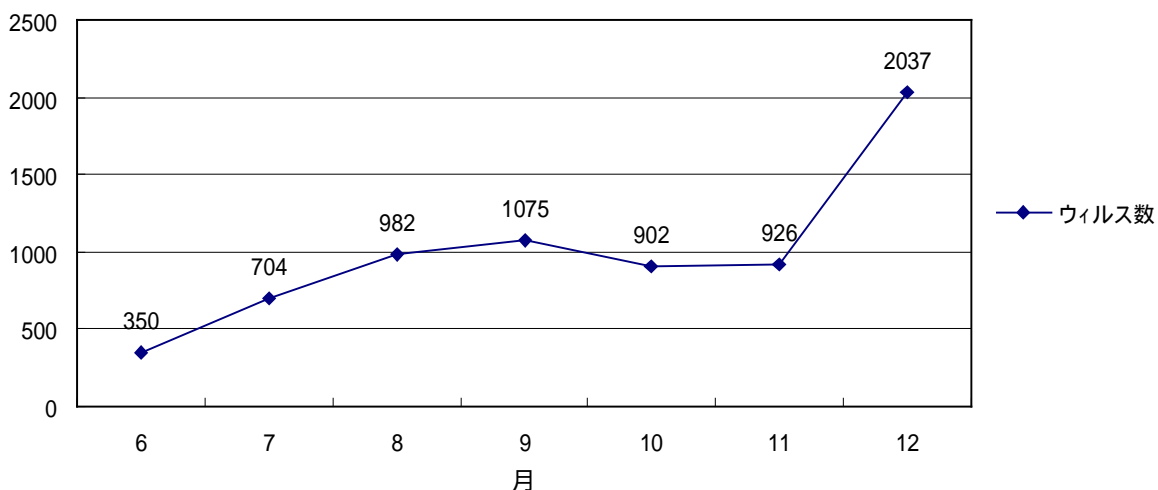


図 3 : ウィルス除去数全体の推移(2003 / 6 ~ 12)

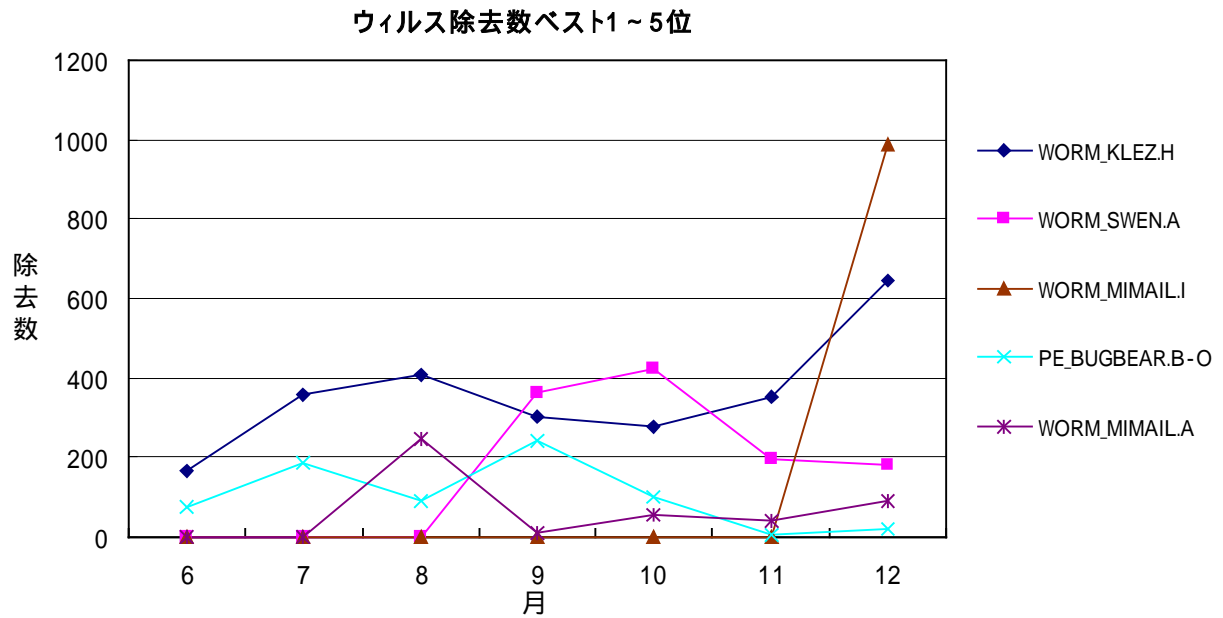


図4：ウイルス除去数ベスト10 - その1 -

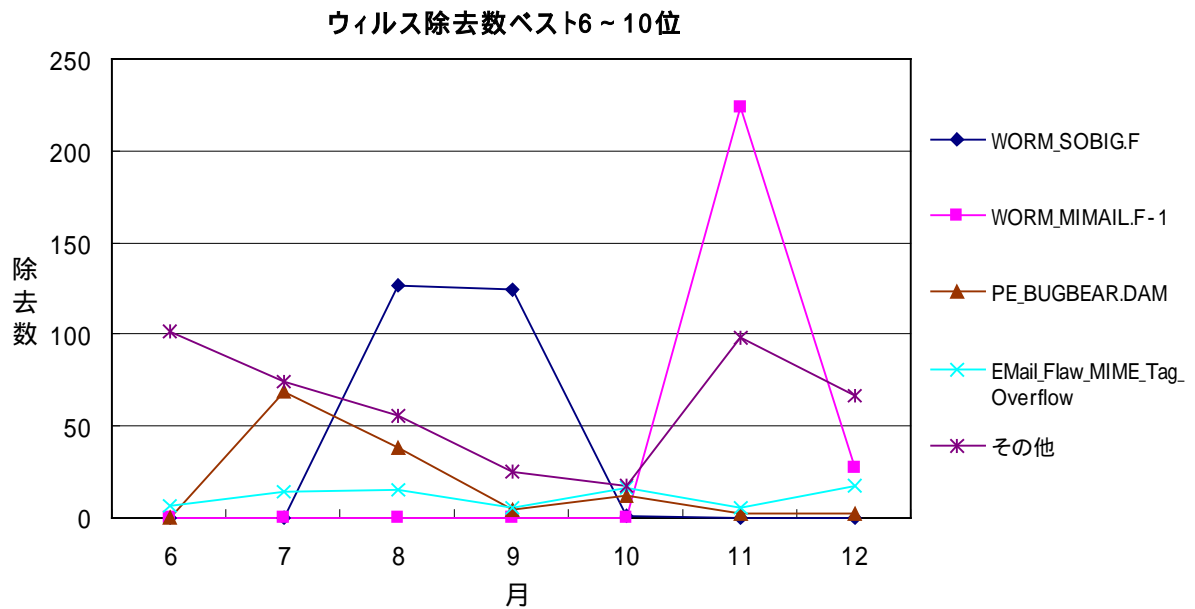


図5：ウイルス除去数ベスト10 - その2 -



ウイルス削除数ベスト10(2003/6~12)

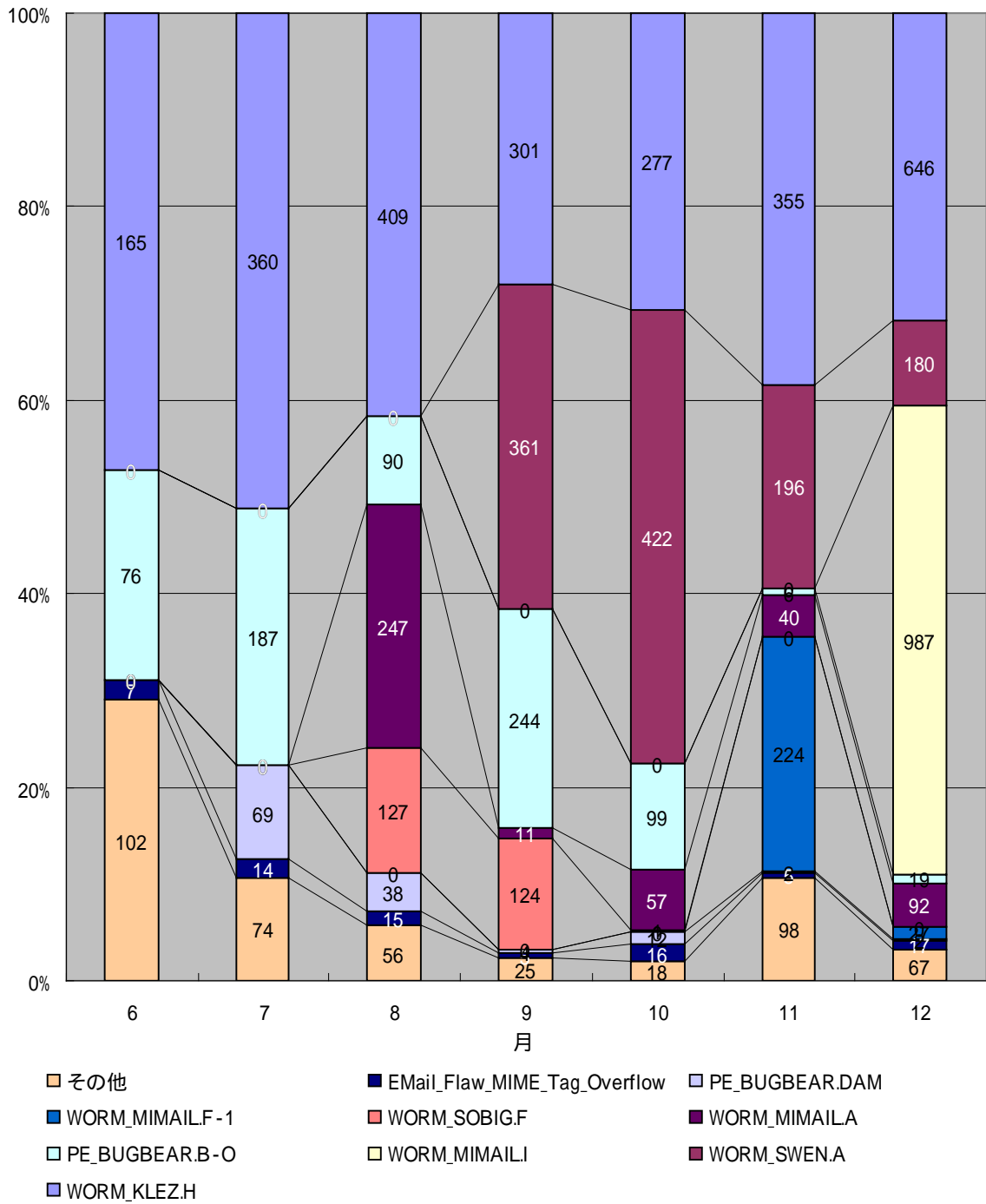


図6：各ウイルスの全体に占める割合（2003/6~1）

(b) 「その他」のウイルス

リスト1に示された「その他」に含まれるウイルスの内訳は次のリスト2の通りである。

リスト2：「その他」ウイルスの内訳

	6	7	8	9	10	11	12	計
WORM_MIMAIL.C-1	0	0	0	0	0	60	17	77
PE_CIH.1003, WORM_KLEZ.H	31	15	2	0	0	0	0	48
ADW_HOTPRIV.A	0	0	0	0	0	24	20	44
WORM_SOBIG.A	21	4	12	4	0	0	0	41
PE_MAGISTR.DAM	38	2	0	0	0	0	0	40
VBS_REDLOF.A	6	6	1	0	4	1	7	25
WORM_BUGBEAR.A	2	14	3	2	0	0	0	21
WORM_SOBIG.F.DAM	0	0	14	6	0	0	0	20
WORM_YAHA.G	1	5	4	9	1	0	0	20
WORM_SOBIG.E	0	15	0	0	0	0	0	15
WORM_KLEZ.E	0	0	12	0	0	0	0	12
PE_Magistr.B	0	2	2	0	5	1	0	10
WORM_MIMAIL.J	0	0	0	0	0	5	4	9
VBS_REDLOF.A-1	0	0	0	0	1	0	7	8
WORM_BADTRANS.B	0	0	1	1	5	1	0	8
WORM_FIZZER.A	2	5	0	0	0	0	0	7
WORM_HYBRIS.B	1	1	0	2	2	1	0	7
PE_DUMARU.A	0	0	0	0	0	0	6	6
JS_FORTNIGHT	0	2	3	0	0	0	0	5
JS_FORTNIGHT.M	0	0	0	0	0	0	2	2
VBS_REDLOF.A-11	0	0	0	0	0	0	2	2
WORM_MIMAIL.G-1	0	0	0	0	0	2	0	2
WORM_MIMAIL.M	0	0	0	0	0	0	2	2
JS_FORTNIGHT.E-2	0	1	0	0	0	0	0	1
PE_Mtx.A, WORM_MTX.A	0	0	0	1	0	0	0	1
VBS_REDLOF.A	0	0	0	0	0	1	0	1
VBS_REDLOF.A.GEN	0	0	1	0	0	0	0	1
WORM_GIBE.B	0	1	0	0	0	0	0	1
WORM_KLEZ.DAM	0	0	1	0	0	0	0	1
WORM_MIMAIL.E-1	0	0	0	0	0	1	0	1
WORM_MIMAIL.H-1	0	0	0	0	0	1	0	1
WORM_SOBIG.D	0	1	0	0	0	0	0	1
計	102	74	56	25	18	98	67	440

### 3．おわりに

設定した2台のウィルスチェックサーバは順調に動作し予定した通りの成果をあげている。ユーザが利用するPCについては、セキュリティに対する意識の高まりを受け、個々にウィルス除去プログラムやパーソナルファイアウォールの導入が進んでいるところであるが、本報告に見られるように、大学全体としてのウィルスチェックサーバの果たす役割は依然として大きいものと思われる。

サーバからのアラートメールを受け取ったユーザから、「不信なメールを受け取った」とか「どう取り扱ったらよいのか」といった質問が時折寄せられる。関係するメールに含まれていたウィルスを除去した旨の連絡であることを冷静に受け止めて、もしも発信者側の立場であるならば、利用しているPCに潜むウィルスを除去するなどの対策に役だ立てて頂きたい次第です。



# 事業報告

(旧香川医科大学情報メディアセンター)



## 旧香川医科大学情報メディアセンター事業報告

旧香川医科大学情報メディアセンターでは、平成 13 年度からセンター所轄事業として計算機システム及び学内ネットワークの更新、増強がなされている。当初同センターの事業運営方針として、学内情報基盤システムが実際に大学の成すべき研究・教育に反映され、評価に値するものであるか、また使用する教職員・学生の視点に立った利用価値の高いインフラを提供できるものであるか、を挙げた。この目標を達成するために、センターでは広く学内に達成すべき事業を募り、その中から以下に挙げた 15 の事業をセンター委員会議において審査し、情報メディアセンター事業として承認するに到ったものである。それぞれの事業は代表責任者を中心に展開されており、ここでは事業内容とともに、現在までの利用状況、評価などについて報告する。

また、旧香川医科大学情報メディアセンター所管機器等を事業別に記載した。大学統合にあたり医学部外の教職員・学生にもこれら機器の利用が可能となっている。これらの機器、システム類が大学全体の利益に繋がり一層効果の高いものになることを期待するものである。

### 旧香川医科大学情報メディアセンター承認事業

- ( 1 ) 学内ネットワーク基盤整備事業
- ( 2 ) 情報リテラシ教育設備の更新と導入
- ( 3 ) マルチメディア教育支援事業
- ( 4 ) 学生生活支援事業
- ( 5 ) チュートリアル教育支援事業
- ( 6 ) 学内動画配信システムの整備
- ( 7 ) 国家試験合格支援設備の整備
- ( 8 ) 実験実習機器センター機器予約システム
- ( 9 ) 学術データ発表、保全支援システム
- ( 10 ) 生体分子の立体構造に関する教育・研究支援事業
- ( 11 ) 機器センターにおける「体験教室」の実施及び、研究効率の向上に向けてのサーバー化
- ( 12 ) 放射線教育支援システム
- ( 13 ) バーチャル PCR 実習設備の整備
- ( 14 ) 院内感染防止対策支援事業
- ( 15 ) 地域医療情報ネットワークの形成を促進するための研究開発



事業名：(1) 学内ネットワーク基盤整備事業

担当者：総合情報基盤センター 河内 一芳

## 1) ネットワークサーバ関連

### ・ DNS サーバ

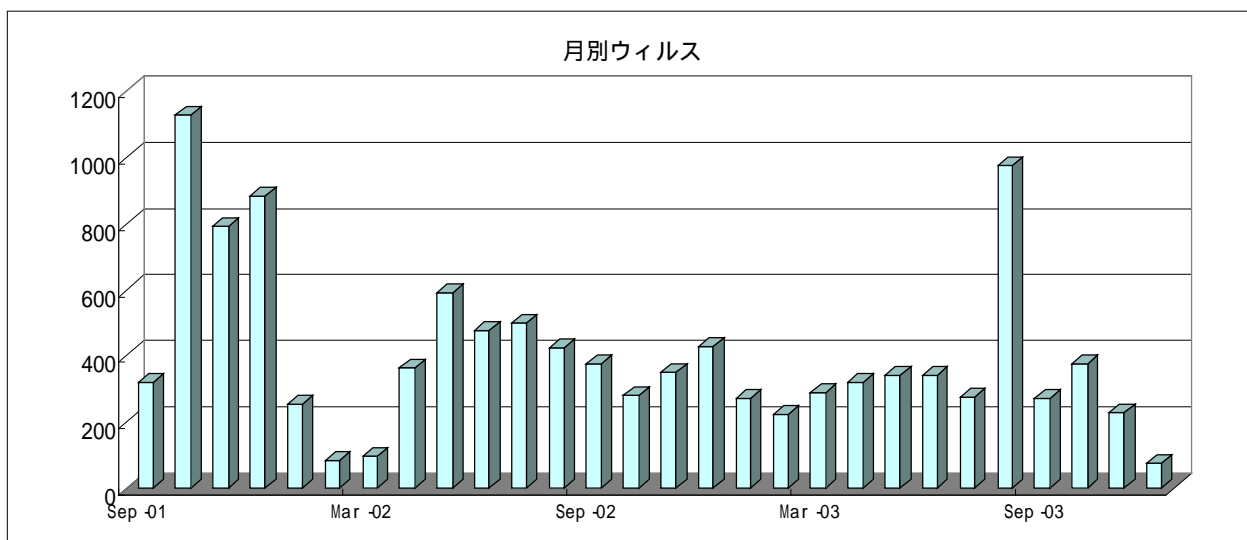
平成6年の旧香川医科大学情報ネットワーク(KMSnet)整備当初から稼働し続けているDNSサーバを2台とも更新し、現在も順調に稼働中です。

### ・ Web サーバ

医学部の広報の一翼を担うWebサーバを更新しました。現在2台で同じ内容を保つためミラーリングをしています。稼働状況は医学部ホームページを参照してください。

### ・ メールサーバ

医学部用メールサーバにウィルス対策ソフト(InterScanVirusWall)を導入しました。



上図は月別のウィルス対策ソフトにより検出・削除されたウィルスの数ですが、導入直後から効果を上げています。

## 2) 接続基盤

### ・ ギガビット L2 スイッチ

旧香川医科大学情報ネットワーク(KMSnet)の基盤部分に32ポートのギガビットスイッチを11台増設しました。情報コンセントもギガビットに対応させるため新たに73個増設し、これにより一部ではありますが Client to Server あるいは Client to Client でギガビットの通信が可能になっております。

### ・ エッジスイッチ

図書館及び看護学科教育研究棟に24ポートのエッジスイッチを合計5台整備し「マルチメディア教育支援」事業及び「情報リテラシ教育設備の更新と導入」事業で設置された学生用パソコンをネットワークに接続しています。

事業名: (2)情報リテラシ教育設備の更新と導入  
担当者: 医学部看護学科環境保健科学 真鍋芳樹  
医学部基礎放射線学 立川敏明、寺中正人

今回の情報メディアセンターの予算措置により、2001年まで使用してきた情報リテラシ教育設備の更新を行った。そのシステム概要、ならびに利用状況、成果等について報告する。

### 【システム概要】



マルチメディア実習室



クライアント PC と中間モニタ

看護学科教育研究棟2階マルチメディア実習室内に本教育設備を導入した。医学科・看護学科双方の学生数や授業形態を考慮し、クライアントPCを72台設置した。機器導入に関して、クライアント・サーバモデルに基づく運用を行うことを念頭にシステム構築をおこなった。

教材はサーバに置き、クライアントからの要求に応じて配信している。また、教育用ソフトの一部では、プログラム本体をサーバに置き、クライアントからの起動命令によってソフトを起動するように設定し、メンテナンスの省力化を図った。さらに、ウィルス対策ソフトでは、サーバが最新のパターンファイルを自動的に取得し、クライアントのログイン時に最新のパターンファイルがクライアントに配信されるようにも設定した。

クライアントPC2台に1台の中間モニタを設置し、クライアントPCのモニタと中間モニタに同一のあるいは別々の映像を配信できる映像・音声配信システムを導入した。

教師側ではモニタ画面にマーキングできる機能を有しており、必要に応じて、重要ポイントを指示できるようになっている。

クライアントPCのハードディスク内容が書き換えられても初期状態に戻すために、また、最新のOS状態へのアップデート、応用ソフトのアップデート、新規応用ソフトのインストールのために、システム更新/復元機能を取り入れた。このことによって、クライアントPCのメンテナンスに要する労力を大幅に削減することが可能となった。

### 【利用状況】

今回導入・整備したシステムを用いて、看護学科の「情報科学」「データとデータ解析」「疫学」、大学院の「看護研究方法特論」「地域看護学特論」「地域看護学演習」、医学科の「情報科学」「分子生物学入門」「生命科学講義」の授業を行っている。さらに、医学科の臨床実習前試験(CBT)、付属病院看護部の看護研究研修会、図書館の文献検索講習会等にも利用されている。

### 【成果等】

今回導入のシステムでは中間モニタを配置したことによって、高い教育効果が得られた。従来は教師のモニタ画面をプロジェクタで投影するという方法が用いられており、投影時には室内照明を減光する必要があった。さらに高精細画像の投影には適さなかった。しかし、今回中間モニタを配置することにより、従来の課題を払拭することが可能となり、室内の減光の必要もなく、高精細画像や動画、ビデオ映像なども配信できることが教育効果につながったと考える。

教育面において、一部の授業では毎年学生からの授業評価を取り入れているが、その結果、非常に高い評価を得ている。教材も従来の紙ベースに加え、マルチメディア教材を提供しているからであろうと思われる。教材作成には労力が必要であるが、最近はマルチメディア教材も比較的安価に購入できるようになり、このシステムでの利用可能性が広がっている。さらに、教材はサーバからの配信であるため、教材のメンテナンスを非常に簡単に行うことが可能になっている。

また、システムメンテナンス面では、クライアントPCの更新/復元システムを導入したため、システムメンテナンスの大幅な省力化を図ることが可能となった。

事業名：(3) マルチメディア教育支援事業  
 担当者：情報図書課医学系整理係 斉藤 純代

マルチメディア教育支援事業は、PC17台、プリンタ2台、ビデオデッキ16台、プロジェクタ1台を主な機器構成として、附属図書館1階のマルチメディア室に設置された。平成14年2月の機器導入以来、新入生のガイダンス、看護学科の授業、学外者向けのデータベース講習会等に有効活用している。また、個人単位の利用では、医学部内で学生が24時間PCを利用できる唯一の場所として高く評価されていることが、利用統計からも伺い知ることができる。

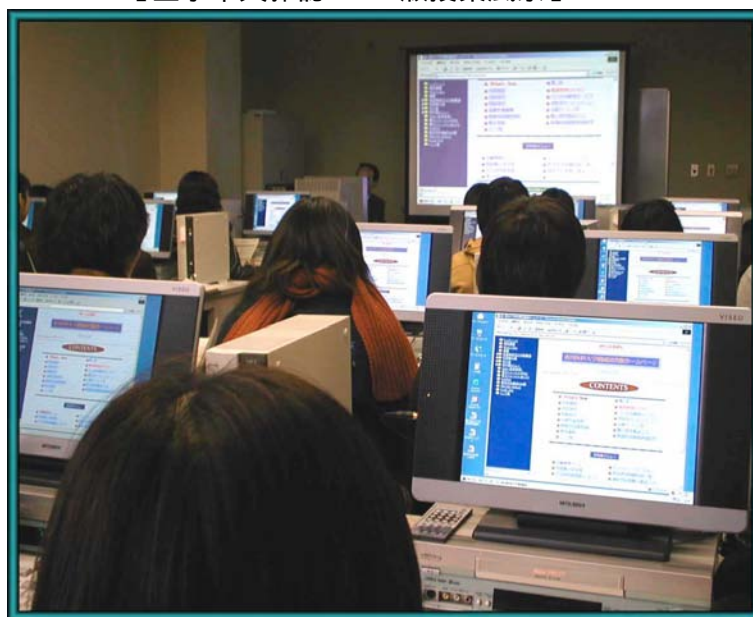
【医学中央雑誌 Web 版授業風景】

主な利用目的は、レポート作成、インターネットでの情報収集、文献検索、ビデオの閲覧等である。

平成15年4月には、ステッドマン医学大辞典を全PCに導入した。これにより、Web上からのワンタッチ翻訳が可能となり、学生の英語文献検索の利便性を向上させた。

現状における問題点は、プリンタのトラブルとハードディスクの障害が多いことが挙げられる。

今後の課題は、少人数のグループ単位のガイダンスがほとんど実施されていないので、回数を増やしていきたい。



マルチメディア室利用統計(2002.4.1 ~ 2003.9.30)

		教職員	学部学生(医)	学部学生(看)	大学院生(医)	大学院生(看)	研究生	合計
平成14年度	利用者数	1,006	19,478	6,322	9	28	52	26,895
	(1日平均)	(2.79)	(54.11)	(17.56)	(0.03)	(0.08)	(0.14)	(74.71)
平成15年度 上半期	利用者数	584	12,274	4,069	65	26	2	17,020
	(1日平均)	(3.19)	(67.07)	(22.23)	(0.36)	(0.14)	(0.01)	(93.01)

マルチメディア室講習会等実施記録

年月日	講習会等名	対象者	参加者数
H.14.1.22	医学中央雑誌Web版検索実習(母性看護学、看護研究の授業として)	看護学科学部学生	57
H.14.2.16	PubMed講習会(インターネット無料データベースの学外向け講習会)	県内医療機関職員	10
H.14.4	新入生図書館ガイダンス(看護学科)	看護学科学部学生	25
H.14.4	新入生図書館ガイダンス(医学科)	医学科学部学生	60
H.14.4	大学院文献検索ガイダンス(医学科)	医学科大学院生	15
H.14.4	大学院文献検索ガイダンス(看護学科)	看護学科大学院生	3
H.14.6.18	医学中央雑誌Web版検索実習(母性看護学、看護研究の授業として)	看護学科学部学生	65
H.15.4	新入生図書館ガイダンス(看護学科)	看護学科学部学生	68
H.15.4	新入生図書館ガイダンス(医学科)	医学科学部学生	60
H.15.4	大学院文献検索ガイダンス(医学科)	医学科大学院生	19
H.15.4	大学院文献検索ガイダンス(看護学科)	看護学科大学院生	4
H.15.12.6	将棋セミナー(将棋部のサークル活動の一環として)	本学将棋部学生	13

事業名：( 4 ) 学生生活支援事業

担当者：総務学務課学生係長 金谷 尚明

学生の教育支援・国家試験合格支援・就職支援・学内情報伝達等学生生活全般にわたる情報提供を行う目的により、平成14年1月から看護学科教育研究棟2階のマルチメディア自習室にパソコン19台（Windows12台・Mac7台）、プリンター2台を8時30分から19時まで、また平成14年4月から医学部会館1階の演習室（1）にパソコン18台（Windows9台・Mac9台）、プリンター2台を8時30分から20時までの間、医学部学生の利用に提供しています。

事業開始後、マルチメディア自習室は毎月半月程度の利用と比較的利用頻度が少なく、演習室（1）は医学部会館が開館している間は、ほぼ毎日利用されています。

コンピュータウイルスによるトラブル等に対して多くの方々の御協力により利用環境を整えていただいて今日に至っております。学生がパソコンを利用する機会は、医師の卒後臨床研修義務化に伴うマッチングあるいは共用試験（CBT）の実施等と日を追うごとに多くなってきております。

今後とも学生の皆さまに大いに利用していただき、勉学等の効果を上げてもらいたいと願っております。





事業名：( 5 ) チュートリアル教育支援事業

担当者：医学部第一内科 永井 雅巳

医学部では、人間性と創造性の豊かな医師、医学研究者を育成するために、医学に関する問題を的確に把握し、その問題を自主的、積極的かつ総合的に解決するために必要な基本的知識、技術の修得のための教育手法が必要とされてきました。また、それらの自己学習法は、学生期間のみならず、生涯にわたって続けられる態度・習慣の養成を意図するものであります。従来の伝統的な授業形態は、多人数の学生を対象とした講義でしたが、この方式では、知識の伝達が一方的なものであり、また記憶することが中心で、考える機会が少なく、学生の学習意欲を喚起しにくいものでした。これらを改善するために、当学部では、平成12年度1年次生より、チュートリアル教育を導入しました。

チュートリアル教育は、6-8名の学生の小グループに1名のチューターが担当する少人数教育(small group-learning)であり、セミナーのように教官が講義する受動的なものではなく、学生が自主的に進める能動的参加型(self-directed learning)の授業です。学生は実際に遭遇するであろうと思われる症例や状況が課題として与えられ、討論を通じて、その課題の中から、自ら問題点の抽出を行います。少人数の討論により個々に自己表現の場が与えられるので、参加意欲も高められます。また、抽出された問題点については、次回までに、その問題点を解決するために必要な知識を、教科書、論文、リソースパーソンとの討論、インターネットなど IT 機器を利用した情報収集などにより自己学習 (problem-based learning) します。次のチュートリアル授業で、学生は自己学習してきた知識に基づいてさらに討論し、問題点を解決していきます。問題の解決を通じて、実際に医学を学んでいきますので、知識が実践的でしかも思考能力が身につく、さらにこの過程を通じて、生涯に亘る自己学習する習慣付けと、医師にとって必要な集団の中での人間関係を保つ資質の育成も期待されます。平成15年度からは、3-4年生の統合講義の中でもチュートリアル教育が導入されました。

本教育手法を有効に働かせるためには、チューターとなる教官の養成とともに、課題内容の充実が必要でした。自学自習するためには、教材や資料が必要なときに手元にあること、また個々のグループでインターネットなどを自由に利用できることが要求されます。これを解消するために、各グループ単位に無線で利用可能なPCを配布し、課題作成者は課題資料として必要なX線フィルムや病理組織像、動画を含めた超音波画像、患者の動きなどを端末から利用可能なウェブとして作成し、いつでも参照できるようにしました。また図書館で整備された図書だけでは不十分な活きた情報がインターネットで得られ、リソースパーソンである教官との討論もメールを利用できるようになりました。さらに、今後は出席や授業に対する教官・学生の評価などについても今までのようなペーパーではなく、PCを利用して行えるようにし、課題などに対する評価が速く、また適切になされるような情報機器の活用法についても検討中です。

事業名：(6) 学内動画配信システムの整備

担当者：医学部附属病院医療情報部 岡田 宏基

#### 1. 使用機器

医学部内のネットワークを使用しての高精細動画配信のために、以下の機器を整備している。

- ・ 電送装置：DVCPRO (Panasonic) 1 対向
- ・ ビデオカメラ：AJ-D215 および付属品 2 式
- ・ デジタルビデオカメラ NV-MX3000 および付属品 2 式：
- ・ マトリックススイッチャー：MS-343DQ 2 台
- ・ その他：ミキサー、DV デッキなどを上記機器と併せて 2 台のラックにそれぞれ収納している。

#### 2. 医学部内ネットワーク (LAN) の調整

元のネットワーク環境では、病院 2 階会議室と遠隔診断室等との動画配信が不能であったため、原因を調査し、不具合を改修した。これにより、医学部内においてはどの 2 点間においても、動画配信が可能となった。

#### 3. 使用実績

本システムはこれまでのところ、JGN (Japan Gigabit Network：通信・放送機構) を介しての遠距離動画配信に利用してきている。以下その実績を紹介する。

- 1) 「ギガビットネットワーク・シンポジウム 2001」(2001.11.19-21；通信・放送機構主催) のスペシャルセッションにて、会場となった沖縄県万国津梁館と香川医大とを JGN で結び、香川医大からリアルタイムで香川県の遠隔医療についてプレゼンテーションを行った。
- 2) 「ギガビットネットワークシンポジウム in やまぐち」(2003.3.20；山口大学) において、香川医大から JGN を介して、香川県の遠隔医療についてリアルタイムプレゼンテーションを行った。
- 3) 「医療分野における最新動向セミナー in 高松」(2003.5.15；MEDIS 主催) において、山口大学より JGN を介して山口県の遠隔医療についての講演をリアルタイムで受信し会場に放映した。
- 4) 香川大学には MINCS の受信設備がないため、厚生労働省のからの MINCS 映像を山口大学 JGN - 香川医大という経路で、2 回にわたり受信し (2003.6.9 および 6.23) 病院内 (病院 2 階会議室) でリアルタイムに放映した。
- 5) 平成 15 年度香川県医学会 (2003.11.16) において、山口大学と香川大学医学部 (臨床講義棟 2 階) 間を JGN で接続し、山口大学から山口県の遠隔医療ネットワークについて、リアルタイムプレゼンテーションを行っていただいた。

これらの映像は、いずれも極めて鮮明であり、DVCPRO を用いた遠隔講義、あるいは遠隔会議は十分に実用的であると評価できた。

#### 4. 今後の予定

医学部内での画像送信は、問題なく行えることがわかっているが、学部間のネットワークは現在の段階では帯域が十分でないので、ネットワーク整備にて十分な帯域が確保できれば、学内の遠隔会議、遠隔講義にも使用して行きたい。今年度末で試験運用が終了する JGN については、来年度から新たに敷設される新高速ネットワークを利用して、MINCS 映像の山口大学経由の受信や、高精細映像を利用した広域医療連携を今後も模索してゆきたいと考えている。

事業名：(7) 国家試験合格支援設備の整備

担当者：医学部薬理学 木村 正司

### 事業の内容

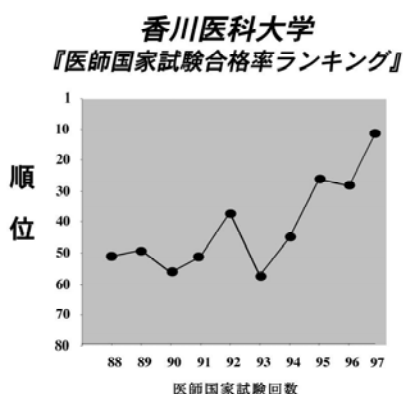
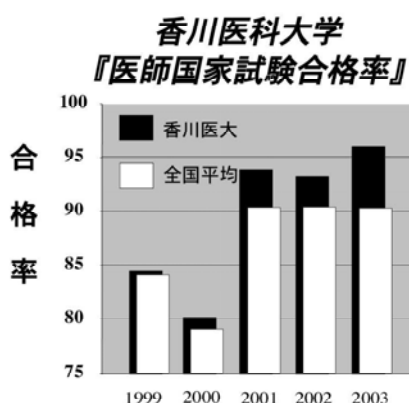
医学部修了時の資格取得試験は、医学科における医師国家試験、及び看護学科における看護師・保健師国家試験である。いずれも医学部卒業対象者のほぼ全員が受験するものであり、本学部教育の最終評価となっている。本事業は、臨床科目講義、実習に携わる教官への国試情報の提供と、学生の個人あるいはグループ学習効率を高めるために、医師国家試験過去問題を収録し、解説を加えられたパソコンシステムを活用利用し、以て当該国家試験合格率の向上を図るものである。

### 事業の中間評価

以下に平成15年までの過去5年の国家試験合格率の推移を以て評価する。

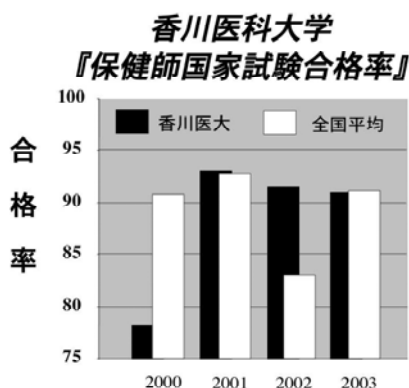
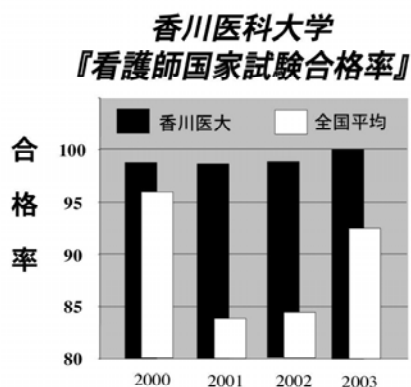
#### 【医師国家試験】

香川医大における本事業開始前までの同試験合格率は、ほぼ全国平均を推移していた。事業開始後平成13年(2000)以降は、合格率の全国平均の上昇が認められるも、香川医大の合格率は、さらにそれを上回るものである。平成15年に到っては大学別合格率ランキングにおいて11位であり、近畿以西の医学科併設大学のうち最高位であった。



#### 【看護師・保健師国家試験】

香川医大看護学科は平成12年より卒業生を輩出している。同年保健師国家試験合格率は77.8%と低調であったが、事業開始後90%以上を堅持している。看護師国家試験においては、ほぼ100%の合格率である。





事業名：( 8 ) 実験実習機器センター機器予約システム

担当者：総合生命科学実験センター実験実習機器部門 永井由紀子

本システムは実験実習機器センターの機器を利用する研究者がネットワークを通じて各講座からオンラインで予約できるよう便宜を図るものであります。当機器センターでは、機器予約システムの運用を平成14年10月より開始し、予約可能な機器は約80種で、現在のユーザー登録数は300です。これまで機器センターでは、機器の予約は「機器予約表」への記入により行っていました。機器使用の際には、機器センター事務室まで来て他の研究者の予約が入っていないか予約表を確認し、予約を書き込んで、その後各講座に帰るという作業に、平均5分かかっていました。しかし、予約システムではパソコンで予約を確認し、即自分の予約をいれることができるので大幅に研究者の時間を短縮することができました。

さらに、この予約システムは下記の性能を備えているため、簡単な入力でいろいろな予約が可能となりました。

1. ユーザーの登録数が無制限であり、また同時アクセス可能人数も無制限である。
2. 一人の利用者が、同一の日付及び時間帯に複数の機器の予約ができ、かつ日をまたいだ予約も可能である。
3. 10分単位での予約が可能である。
4. 繰り返し予約の登録が可能である。
5. 機器の登録可能数が無制限である。
6. 予約の日表示、週表示、及び月表示ができる。
7. 年間延べ最大10000回の予約が可能である。

#### 《機器予約ページ》

	1(木)	2(金)	3(土)	4(日)	5(月)
クリオスタット	●	●	●	●	●
顕微鏡用ミクロトーム	●	●	●	●	●
メディカルフィルムプロセッサ(1)	●	●	●	19:00-20:00 大森実	●
メディカルフィルムプロセッサ(2)	●	●	●	●	●

事業名：(9) 学術データ発表、保全支援システム  
(研究成果のデジタルプレゼンテーション・パブリケーション支援)  
担当者：医学部分子細胞機能学 西 望  
総合生命科学実験センター実験実習機器部門 宮中宏  
医学部附属病院検査部 多田達史

利用状況及び経過報告(平成15年12月現在)

#### 入力支援設備内容

A3フラットベッドスキャナ、ドキュメントスキャナ、X RAYフィルムスキャナ、フィルムスキャナ



利用者数413人

#### 編集支援

ノンリニアビデオ編集、デジタル文書作成システム、画像解析システム



利用者数135人

#### データ解析支援

イメージアナライザ



利用者数768人

#### 出力支援

フィルムレコーダ、写真タイププリンター、B0プリンタ、A3レーザープリンタ(モノクロ・カラー)



利用者数945人

#### 参考データ

利用講座数 43講座

利用者発表学会

(日本細菌学会、生理学会、日本法医学会、日本小児放射線学会、日本法中毒学会、解剖学会、Neuropathology学会、日本消化器学会、日本生化学会、呼吸器学会、日本泌尿器学会、毒素シンポジウム、感染症シンポジウム、小児外科学会、分子生物学会、Clostpath Conference 2003、国際泌尿器科学会議、その他)

事業名: (10-1) 生体分子の立体構造に関する教育・研究支援事業

担当者: 医学部生体情報分子学 徳光 浩(担当分)

[教育分野]

情報メディアセンター事業で導入された機器を用いて、医学部学生の研究教育を行い、筆頭著者(医学科5年生)として国際学術雑誌に以下の論文が掲載されました。

Identification and Characterization of Novel Components of a  $Ca^{2+}$ /Calmodulin-dependent Protein Kinase Cascade in HeLa Cells. : Y. Ishikawa, H. Tokumitsu, H. Inuzuka, M. Murata-Hori, H. Hosoya, and R. Kobayashi *FEBS Letters* 550, 57-63, 2003

[研究分野]

事業導入機器を利用した下記の学術論文が受理されました。

- 1) A single amino acid difference between and  $Ca^{2+}$ /calmodulin-dependent protein kinase kinase dictates sensitivity to the specific inhibitor, STO-609. : H. Tokumitsu, H. Inuzuka, Y. Ishikawa, and R. Kobayashi *The Journal of Biological Chemistry* 278, 10908-10913, 2003
- 2) Postsynaptic density-95 promotes calcium/calmodulin-dependent protein kinase II-mediated Ser<sup>847</sup> phosphorylation of neuronal nitric-oxide synthase. : Y. Watanabe, T. Song, K. Sugimoto, M. Horii, Araki N, H. Tokumitsu, T. Tezuka, T. Yamamoto, and M. Tokuda *Biochemical Journal* 372, 465-471, 2003
- 3) Hsp90 is a direct target of anti-allergic drugs, disodium cromoglycate (DSCG) and amlexanox. : M. Okada, H. Itoh, T. Hatakeyama, H. Tokumitsu, and R. Kobayashi *Biochemical Journal* 374, 433-441, 2003
- 4) Neuronal calcium sensor proteins are targets for the insulinotropic agent repaglinide. : M. Okada, D. Takezawa, S. Tachibanaki, S. Kawamura, H. Tokumitsu, and R. Kobayashi *Biochemical Journal* 375, 87-97, 2003
- 5) Regulatory Mechanism of *Dictyostelium* Myosin Light Chain Kinase A. : H. Tokumitsu, H. Hatano, H. Inuzuka, Y. Ishikawa, T. Q. P. Uyeda, J.L. Smith, and R. Kobayashi *The Journal of Biological Chemistry* 2003 in press
- 6) S100A1 is a novel molecular chaperone and a member of the Hsp70/Hsp90 multichaperone complex. : M. Okada, T. Hatakeyama, H. Itoh, N. Tokuta, H. Tokumitsu, and R. Kobayashi *The Journal of Biological Chemistry* 2003 in press

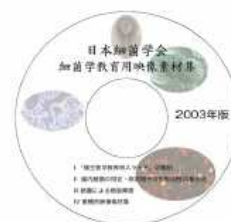
[広報分野]

ホームページ (<http://www.kms.ac.jp/~chemist/indx.html>) を作成いたしました。

事業名: (10-2) 生体分子の立体構造に関する教育・研究支援事業  
担当者: 医学部分子微生物学 松下 治(担当分)

[ 教育分野 ]

1) 情報メディアセンター事業で導入された機器を用いて、日本細菌学会教育用スライドや新しい教育用映像(計650 葉以上)を電子化し、これらを収録したCD-ROM(450 部)を作成して配布いたしました。このような活動を通じて我が国の微生物学教育の高度化をはかることができました。



2) 医学科2年生を対象にした生命科学講義(医学系分子生物学I)において、「タンパク質の構造と機能」に関する講義を担当しました。情報メディアセンター事業用機器を、既設の無線LAN および液晶プロジェクターの各設備と連携して使用することにより、学生にタンパク質の構造をリアルに説明することができ、医学導入教育の質を高めることができました。

「コーディネータの意見」

無線LAN とインターネットを介してタンパク質構造データベースにアクセスする様子を学生に実演いただきありがとうございました。従来の方法では教科書やプリントなどで平面的に学生に示説せざるをえなかったタンパク質の構造が、教室のスクリーン上でリアルタイムに三次元表示され、学生の理解度も深まったことと思います。

佐藤忠文

[ 研究分野 ]

事業導入機器を利用した下記の学術論文が受理されましたので、ご報告申し上げます。

- 1) Kaji, M., O. Matsushita, E. Tamai, S. Miyata, Y. Taniguchi, S. Shimamoto, S. Katayama, S. Morita, and A. Okabe. A novel type of DNA curvature present in a *Clostridium perfringens* ferredoxingene: characterization and role in gene expression. *Microbiology* 149:3083-3091. (2003)
- 2) Tamai, E., T. Ishida, S. Miyata, O. Matsushita, H. Suda, S. Kobayashi, H. Sonobe, and A. Okabe. Accumulation of *Clostridium perfringens* epsilon-toxin in the mouse kidney and its possible biological significance. *Infect. Immun.* 71:5371-5375. (2003)
- 3) Wilson, J.J., O. Matsushita, A. Okabe, and J. Sakon. A bacterial collagen-binding domain with novel calcium-binding motif controls domain orientation. *EMBO J.* 22:1743-1752. (2003)

また汎用機器につきましては、院生研究棟7階セミナー室に設置した液晶プロジェクターが学位審査などで広く活用されるようになりました。また昨年に引き続き法医解剖にデジタル・カメラをご利用いただくなど、事業担当者以外にも幅広くご活用いただいております。好評を博しております。

[ 広報分野 ]

ホームページ([www.kms.ac.jp/.microbio/index.html](http://www.kms.ac.jp/.microbio/index.html))を作成いたしましたので、ご報告申し上げます。

事業名:(10-3)生体分子の立体構造に関する教育・研究支援事業

担当者:医学部分子細胞機能学 西 望(担当分)

[教育活動]

1. 本事業で導入された機器を用いて、県内の高校生を対象とした体験学習\*用スライドを作製し、機器操作体験前後の講義・説明に使用した。

\* : 附属実験実習機器センター主催(2003年1月7日実施)

2. 本学医学科3年生を対象とした内分泌学講義(「ホルモン受容体」, 「膵内分泌 ~」)において、自作の教材および「Molecular Biology of The Cell, 4th Edition」付属のCD-ROM(生体高分子の構造と機能に関する動画を含む)を、教育効果を高めるための補助的手段として用いた。この際、本事業で導入された機器と既設の液晶プロジェクターを利用することにより、講義(教材の準備を含む)を効率よく進めることができた。また、本学医学科3年生を対象とした内分泌学実習において、実習内容の説明および実験操作マニュアルをまとめた「実習の手引き」を学生に配布した。この「実習の手引き」の作製に本事業で導入された機器を利用した。



[研究活動]

研究とその成果の公表(学会発表、論文作成)に本事業で導入された機器を利用し、下記の学術論文を発表した。

1. Nishi, N., Shoji, H., Seki, M., Itoh, A., Miyataka, H., Yuube, K., Hirashima, M. and Nakamura, T. Galectin-8 modulates neutrophil function via interaction with integrin  $\alpha$ M. *Glycobiology* (2003) 13:755-763
2. Pelletier, I., Hashidate, T., Urashima, T., Nishi, N., Nakamura, T., Futai, M., Arata, Y., Kasai, K., Hirashima, M., Hirabayashi, J. and Sato, S. Specific recognition of *Leishmania major* poly-beta-galactosyl epitopes by galectin-9: possible implication of galectin-9 in interaction between *L. major* and host cells. *J. Biol. Chem.* (2003) 278:22223-22230
3. Shoji, H., Nishi, N., Hirashima, M. and Nakamura, T. Characterization of the *Xenopus* galectin family. Three structurally different types as in mammals and regulated expression during embryogenesis. *J. Biol. Chem.* (2003) 278:12285-12293

事業名：(11) 機器センターにおける「体験教室」の実施及び、研究効率の向上に向けての  
サーバー化

担当者：総合生命科学実験センター実験実習機器部門 夕部 幸一、漆原 正英

実施日時：2003 年度 1 月 7 日火曜日

香川県立北高校 理学部部員 5 人（引率 1 人）の参加による機器センター主催の体験実習  
を行いました。

#### 講義及び実習の内容

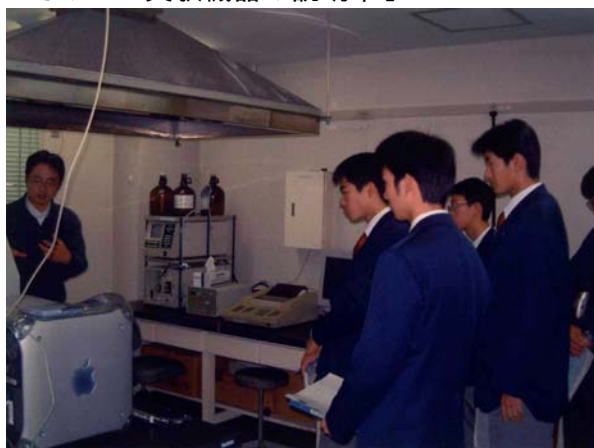
講義内容：DNA の構造、蛋白質の構造、遺伝子から蛋白質についてのセントラルドグマ、  
その他にオーダーメイド医療、ゲノム情報の取り扱いの問題点等、プロジェクターを用い  
て解説。

実習内容（DNA シーケンサー）：ピペッターの取り扱い方法、PCR の原理及び DNA  
シーケンサーの原理等の説明。標準品を用いて実際にサンプルを調整し、DNA シーケン  
サーを用いてサンプルを測定する迄の操作を機器の操作の説明も交えて学生に体験してもら  
う。

実習内容（プロテインシーケンサー）：分子模型を用いてアミノ酸の説明、分析原理と  
してエドマン分解の説明をした。ピペッターの操作方法を指導し ラクトグロブリンを用  
いて分析機器にサンプルをアプライするま迄の実習を指導した。又、データの解析法、ア  
ミノ酸の種類と一字表記方法、単位の表し方等を説明した。

#### 体験教室の風景

その 1 「実験機器の説明中」



その 2 「講義中」





事業名：(12) 放射線教育支援システム

担当者：放射性同位元素実験施設（医学部地区） 須田博文

本施設では、放射線業務従事者に義務付けられている定期の教育訓練をウェブ上で実施することを検討しています。使用するコンテンツは現在開発途中ですが、これまでのところ、放射線の人体に与える影響とその防護について解説した動画コンテンツと、放射線やR I（放射性同位元素）の基本的性質、それらの安全取り扱いや関連する法令等に関する質問とそれへの回答および達成度を測るインタラクティブコンテンツの試作版があります。

現在は各コンテンツとも試験段階ですが、他の施設との協同も検討しつつ、新たなコンテンツの拡充を図るとともに、現在開発中のものについても、その内容や形式、操作性などの充実と向上をはかる予定です。

以下に、上述したコンテンツの一部画像を掲載いたします。

図1と図2は動画コンテンツからのキャプチャー画面。タイトル画面（図1）と放射線の人体への作用の解説部分（図2）

図3と図4は受講者に対してなされる放射線の安全取り扱い等に関する質問画面と回答画面。

図 1



図 2

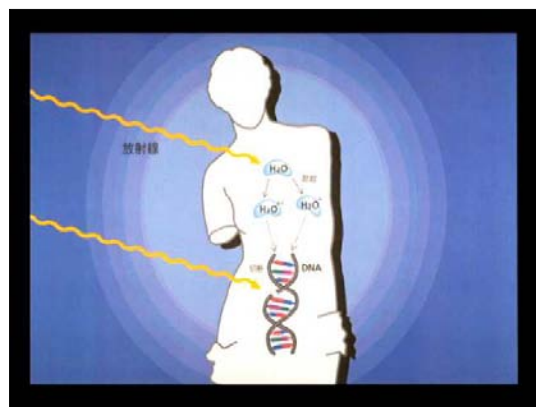


図 3

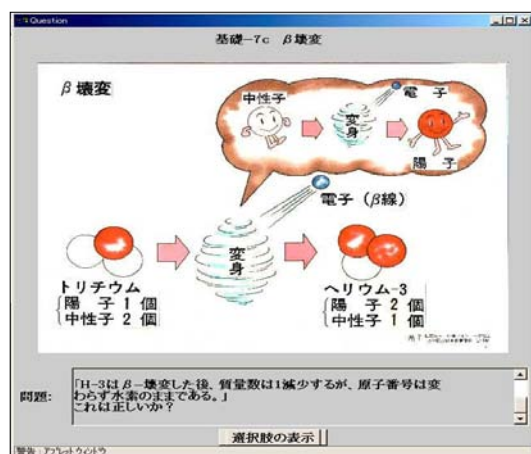
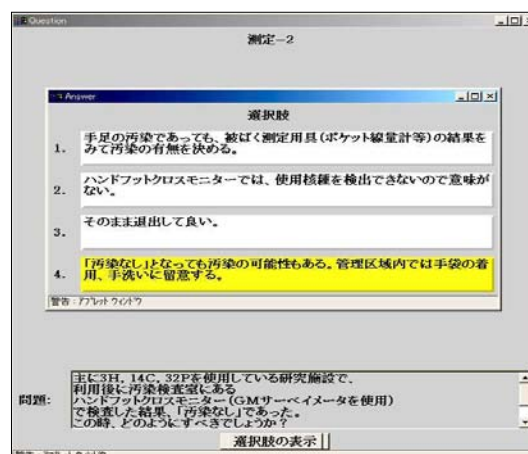


図 4





事業名：(13) バーチャルPCR実習設備の整備

担当者：総合情報基盤センター 佐藤 忠文

医学部3年次生の分子生物学実習の一項目としてDNAの抽出、酵素処理、増幅がある。受講生約30名を6班に分けヒト血球などからDNAを抽出・精製した後PCR(Polymerase Chain Reaction)によって特定塩基配列を増幅している。

「バーチャルPCR」は鋳型DNA分子が時間とともに倍々に増幅してゆく様子をin silicoにシミュレートするものである。

“水系物採り”実習の実施が半日から一日を要するのに対して、コンピュータによるシミュレーション実習は2-3時間の所要と随所性、随時性の利点を有する。

以下は2002年度と2003年度の試作ソフトによるレポートの中の一例である。

バーチャルPCR実習レポートの一例  
(2003年度、3年次生 3081 F君の場合)

図1 バーチャルPCRによるDNAの増幅曲線

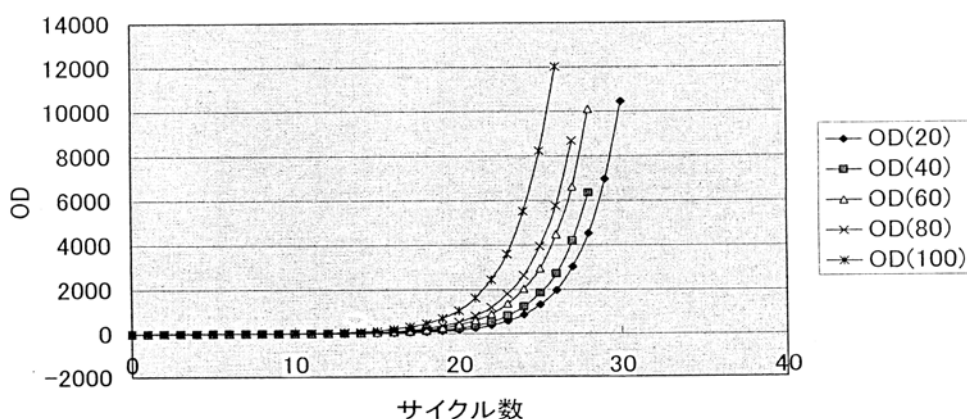
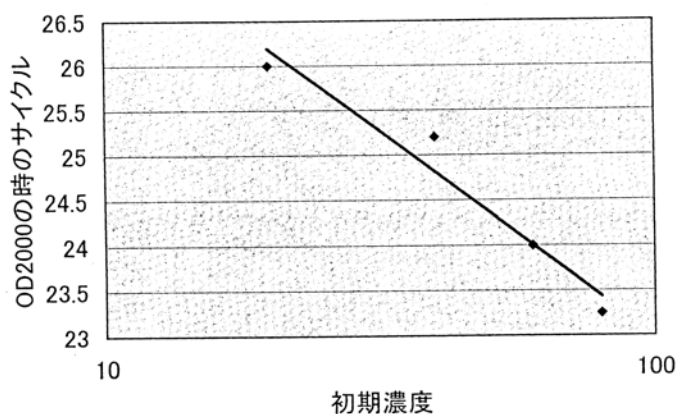


表1 仮想DNA量、OD=20, OD=40, OD=60, OD=80, OD=100 サンプルの増幅

サイクル数	OD(20)	OD(40)	OD(60)	OD(80)	OD(100)
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	1	0	1
3	0	0	1	1	1
4	0	0	1	1	2
5	0	1	1	1	2
6	1	1	1	2	4
7	1	1	2	3	5
8	1	2	3	4	8
9	2	3	5	6	10
10	3	4	7	9	19
11	4	6	10	14	28
12	6	10	15	20	41
13	10	14	23	31	63
14	14	21	35	47	94
15	21	32	52	69	145
16	31	49	79	105	220
17	47	73	118	157	324
18	71	108	178	242	712
19	109	165	264	367	1055
20	163	249	406	540	1629
21	252	368	612	820	2451
22	376	537	897	1201	3606
23	579	820	1363	2637	5512
24	864	1232	2030	3951	8227
25	1330	1848	2954	5767	12020
26	1982	2725	4471	8685	
27	33038	4190	6613		
28	4530	6341	10098		
29	6964				
30	10447				

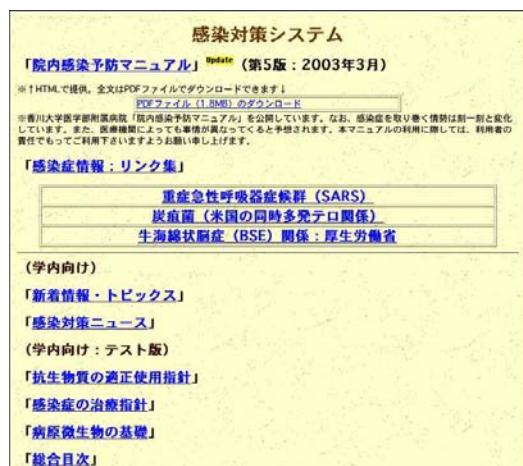
図2 未知DNA濃度算出のための標準曲線



初期DNA量	OD2000
80	23.25
60	24
40	25.2
20	26

事業名：(14) 院内感染防止対策支援事業  
担当者：医学部附属病院感染対策室 樋口 和子

## [ 現状報告 ]



感染対策室では、院内感染防止など感染症の適切なコントロールに努めています。そのためには、感染症の制御に必要な情報について、医療関係者同士の情報の共有や、地域医療機関の院内感染防止の支援、地域住民への感染症に関する正しい情報の提供が大切です。これらは、大学病院の感染対策室の重要な責務だと考え、感染症に関する情報の収集、整理、提供に努めております。今回、情報メディアセンター事業で導入された機器により、以下の感染症情報の提供を行っています。

### ( 感染対策室 )

1. 感染対策システムをインターネットで提供しています。同システムでは、「香川大学医学部附属病院院内感染マニュアル」(第5版、平成15年3月改訂)の提供を開始しました。また、PDFファイルでの提供を追加し、他の施設でも容易にダウンロードでき、利用しやすくなりました。
2. イン트라ネットでは、社会へ公開された情報を元に、抗生物質の適正使用指針、感染症の治療指針、病原微生物の基礎などを、学内向けに試験的に提供しています。

### ( 細菌検査室 )

3. 細菌検査報告書にグラム染色画像を添付するようにし、医師の適切な判断を支援しています。
4. 感染対策委員会での耐性菌の発生状況の報告に活用しています。また、本院の感受性情報を集積し、学内への提供準備をしています。
5. 染色画像を蓄積し、本院独自の病原微生物の基礎データの提供準備をしています。

### [ 今後の予定 ]

現在、学内向けに提供している、抗生物質の適正使用指針、感染症の治療指針、病原微生物の基礎などを、本院のデータに基づいて作成、編集し、インターネットまたはイントラネットで公開する準備を進めています。

事業名：(15) 地域医療情報ネットワークの形成を促進するための研究開発

担当者：医学部附属病院医療情報部 原 量宏

## 1. 使用機器

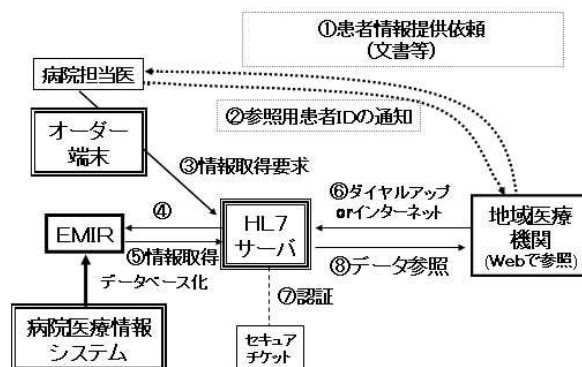
- ・HL7 サーバ（センター設備品）
- ・セキュアチケット用コンピュータ（医療情報部で準備）

## 2. 事業概要

厚生労働省が平成13年に提言した「保健医療分野の情報化にむけてのグランドデザイン」では、医療機関への電子カルテの普及とそれを利用した診療情報ネットワークの形成とが期待されている。しかしながら、平成15年現在での現状を見ると、中～大規模病院での電子カルテの普及は徐々に進んでいるものの、診療所での普及は著しく遅れている。このため、電子カルテを介した診療情報のネットワーク形成には、なお相当の時間を要するものと予測される。このような過渡期にあっては、診療所に電子カルテがない状態でも、中核病院の診療情報が高度なセキュリティの元にインターネットで継続して参照できるシステムが囑望される。単発的な診療情報提供・参照システムは、香川医科大学（当時）の遠隔診断システムを発展させた形で、香川県が「かがわ遠隔医療ネットワーク：K-MIX」（<http://www.m-ix.jp/>）を平成15年6月から稼働させた。これに対して、診療情報を継続して参照できるシステムとして、われわれは、病院医療情報システムに蓄積された診療情報を、高度なセキュリティの元に、インターネットを介して地域医療機関から参照できるシステムを、HL7 サーバを利用して開発した。なお、病院診療情報は、既に稼働している診療情報二次利用のためのデータベースである EMIR から取得することとした。

## 3. システムの概要

患者情報の参照を希望する地域医療機関医師（登録制）は、当院の担当医にどの患者のどのデータを参照したいかを文書等で申請する。その内容に基づいて、当該患者の参照用 ID が発行され、それを希望医師に通知する。担当医は、医療情報端末で、当該患者の当院での ID を入力し、参照させるデータ種別を選択する。この操作に伴って、HL7 サーバは参照用医療情報データベースである EMIR にアクセスし、



当該データを取得し、HL7 の形式でサーバ上にデータを保存、データベース化する。地域医療機関からは インターネットまたはダイヤルアップで HL7 サーバにアクセスし、セキュアチケット（認証ツール）で認証された後に、データ参照が可能となる。

## 4. 稼働予定

平成15年末にプロトタイプはほぼ完成。まず、EMIR との通信試験を行い、次いで院内からのネットワーク接続試験を行う。これらの後、セキュアチケットの設定、遠隔診断用ファイアウォールの設定を行い、インターネット（STCN：STNet 利用）経由での院外からの接続を可能とする。利用者の認証は当面セキュアチケットを用いるが、四国電力で稼働させている地域認証局も将来的に利用したいと考えている。

なお、本システムについては、第23回医療情報学連合大会で岡田が発表し、座長推薦演題として、第8回日本医療情報学会春季学術大会演題に推挙された。

## 旧香川医科大学情報メディアセンター支援事業配布機器一覧

### (1) 学内ネットワーク基盤整備事業

DNS サーバ	2 台
Web サーバ	2 台
パソコン (Mac)	1 台
パソコン (Windows)	1 台
ギガビット L2 スイッチ 32 ポート型	11 台
ギガビット L2 スイッチ 8 ポート型	1 台
エッジスイッチ 24 ポート型	5 台
ギガビットスイッチ	1 台

### (2) 情報リテラシ教育設備の更新と導入

管理用サーバ	3 台
教材提供用サーバ	2 台
パソコン (Windows)	72 台
教官用パソコン (Windows)	4 台
管理用パソコン (Windows)	3 台
カラーレーザープリンタ	3 台
モノクロレーザープリンタ	5 台
スキャナ	5 台
デジタルビデオカメラ	4 台
デジタルカメラ	2 台
パソコン (Mac)	1 台
デジタルビデオレコーダ	2 台
デジタルビデオ編集機	1 台
フィルムレコーダ	1 台
ノートパソコン (Mac)	1 台
ノートパソコン (Windows)	1 台

### (3) マルチメディア教育支援事業

管理用サーバ	1 台
パソコン (Windows)	17 台
プロジェクタ装置	1 台
ビデオデッキ	1 台
プリンタ	2 台

### (4) 学生生活支援事業

パソコン (Windows)	23 台
パソコン (Mac)	17 台
モノクロレーザープリンタ	4 台

### (5) チュートリアル教育支援事業

パソコン (Mac)	3 台
ノートパソコン (Mac)	30 台
AirMac ベースステーション	15 台
プロジェクタ装置	1 台

### (6) 学内動画配信システムの整備

電送装置	2 台
DVCPRO デスクトップレコーダ	4 台
DVCPRO カメラレコーダ	2 台
液晶デジタルカメラ	2 台
デジタルビデオカセットレコーダ	2 台

### (7) 国家試験合格支援設備の整備

パソコン (Mac)	16 台
パソコン (Windows)	13 台
ノートパソコン (Mac)	23 台
ノートパソコン (Windows)	14 台

### (8) 実験実習機器センター機器予約システム

サーバ	1 台
-----	-----

### (9) 学術データ発表、保全支援システム

A3 対応スキャナ	2 台
Film スキャナ	2 台
立体物スキャナ	1 台
デジタルスキャナ	2 台
デジタルカメラ	2 台
デジタルビデオカメラ	3 台
パソコン (Windows)	5 台
パソコン (Mac)	4 台
ノートパソコン (Windows)	2 台

ノートパソコン(Mac)	2台
画像処理用コンピュータ(Windows)	2台
画像処理用コンピュータ(Mac)	2台
動画編集システム	2式
スライド作成システム	2式
カラーコピー機	1台
B0版プリンタ	1台
カラーレーザプリンタ	2台
プロジェクタ装置	1台
AirMac ベースステーション	1台

(14) 院内感染防止対策支援	
ファイルサーバ(Mac)	1台
パソコン(Mac)	1台
カラーインクジェットプリンター	1台

(15) 地域医療情報ネットワークの形成を促進するための研究対策	
地域医療情報ネットワーク用システム	1式
ファイアーウォール	1台

(10) 生体分子の立体構造に関する教育・研究システム

デジタルカメラ	1台
カメラサーバ(動画入力装置)	1台
パソコン(Mac)	6台
プロジェクタ装置	1台

(11) 機器センターにおける「体験教室」の実施及び研究効率の向上にむけてのサーバ化

DNA Sequencer 専用パソコン(Mac)	4台
ノートパソコン(Mac)	1台
カラーインクジェットプリンター	3台
プロテイン Sequencer 専用パソコン(Mac)	1台

(12) 放射線教育支援システム

サーバ(Windows)	1台
--------------	----

(13) バーチャル PCR 実習設の整備

サーバ(Windows)	1台
サーバ(Mac)	1台
ノートパソコン(Windows)	1台
デジタルプロジェクタ	1台
デジタルカメラ	1台
カラーインクジェットプリンター	1台
スキャナ	1台

# 報 告

(旧香川大学情報処理センター)





## 情報処理センター日誌

2002年(平成14年)

- 4 . 1 (月) 回線速度変更 (SINET 10Mbps)  
5 (金) 回線速度変更 (附属幼稚園 128Kbps)  
19 (金) 第124回センター会議  
平成14年度業務分担、次期システム、平成14年度教育基盤設備充実経費の要求、平成14年度一般営繕要求、報告、その他について協議。  
第120回NEC連絡協議会  
メーカー側より前回協議会以降のシステム整備状況について作業報告を受け、センターとしてより具体的に不具合な状況や要望を伝えた。
- 5 . 7 (火) 情報処理センターコンピュータシステム仕様書案説明会  
17 (金) 第125回センター会議  
次期システム、ニューズレターNo.45の発行、SINETノード、各部署への緊急時の連絡体制、情報処理体制検討分科会、ネットワーク障害、報告、その他について協議。  
第121回NEC連絡協議会  
メーカー側より前回協議会以降のシステム整備状況について作業報告を受け、センターとしてより具体的に不具合な状況や要望を伝えた。
- 24 (金) 情報処理センターコンピュータシステム仕様策定委員会 (第5回)
- 6 . 5 (水) 情報処理センターコンピュータシステム仕様策定委員会 (第6回)  
21 (金) 第126回センター会議  
次期システム、平成13決算・平成14予算(案)、運営委員会の開催、Piafs Dialupの申請、平成14年度購入雑誌、DHCPサービス、報告、その他について協議。  
第122回NEC連絡協議会  
メーカー側より前回協議会以降のシステム整備状況について作業報告を受け、センターとしてより具体的に不具合な状況や要望を伝えた。
- 27 (木) 第17回情報処理センター協議会総会 (電気通信大学)
- 7 . 4 (木) 第1回情報処理センター運営委員会  
平成13年度決算及び平成14年度予算(案)、次期システム技術審査職員の任命、報告、その他について審議。  
8 (月) 情報処理センターコンピュータシステム入札説明会

- 19 (金) 情報処理センターコンピュータシステム仕様策定委員会 (第7回)  
 第127回センター会議  
 次期システム、教育用PCの再インストール、ニュースレターNo.46の発行、ファイル交換ソフト等による不正利用、報告、その他について協議。  
 第123回NEC連絡協議会  
 メーカー側より前回協議会以降のシステム整備状況について作業報告を受け、センターとしてより具体的に不具合な状況や要望を伝えた。
- 8.14 (水) 夏季休業 (16日まで)  
 21 (水) 情報処理センターコンピュータシステム入札締切
- 9.17 (火) 情報処理センターコンピュータシステム開札  
 20 (金) 第128回センター会議  
 次期システム、教育用PCの再インストール、年報の発行、入退室管理システム、パケット監視、報告、その他について協議。  
 第124回NEC連絡協議会  
 メーカー側より前回協議会以降のシステム整備状況について作業報告を受け、センターとしてより具体的に不具合な状況や要望を伝えた。
- 10.15 (火) 平成14年度大型計算機利用大阪地区(第6地区)協議会総会 (大阪大学)  
 18 (金) 第129回センター会議  
 次期システム、法人化に向けての意見聴取、報告、その他について協議。  
 第125回NEC連絡協議会  
 メーカー側より前回協議会以降のシステム整備状況について作業報告を受け、センターとしてより具体的に不具合な状況や要望を伝えた。
- 11.15 (金) 第130回センター会議  
 次期システム、平成16年度概算要求、SINETノード、ニュースレターNo.47の発行、非常勤職員(夜間)の雇用、報告、その他について協議。  
 第126回NEC連絡協議会  
 メーカー側より前回協議会以降のシステム整備状況について作業報告を受け、センターとしてより具体的に不具合な状況や要望を伝えた。
- 12.4 (水) 平成14年度学術情報ネットワークノード担当者会議 (学術総合センター)  
 20 (金) 第131回センター会議  
 次期システム、予算執行状況、新入生用パンフレット、年末年始のシステム運用、報告、その他について協議。  
 26 (木) 第126回NEC連絡協議会

メーカー側より前回協議会以降のシステム整備状況について作業報告を受け、センターとしてより具体的に不具合な状況や要望を伝えた。

27(金) 年末年始休業(1月5日まで)

2003年(平成15年)

1. 6(月) SINET接続回線変更(岡山大学ノード 香川大学ノード)

17(金) 第132回センター会議

次期システム、平成16年度概算要求、平成15年度建物新営に伴う設備要求等、平成15年度営繕関係経費等の要求、学術情報ネットワーク等の接続経費、運営委員会の開催、ファイルサーバ(daisy)の運用、報告、その他について協議。

第127回NEC連絡協議会

メーカー側より前回協議会以降のシステム整備状況について作業報告を受け、センターとしてより具体的に不具合な状況や要望を伝えた

2. 18(金) 第133回センター会議

次期システム、サブドメインの取得、キャンパス間接続システム、ニュースレターNo.48の発行、平成15年度センター開館、教育用PCおよび学生用メールアカウントの年度更新、ファイアウォールの継続等の確認、平成15年度の利用申請、平成15年度のPCルーム等の利用調査、報告、その他について協議。

第128回NEC連絡協議会

メーカー側より前回協議会以降のシステム整備状況について作業報告を受け、センターとしてより具体的に不具合な状況や要望を伝えた

3. 17(月) 管理者対象利用者説明会(部局教職員用サーバ)

18(火) 第2回情報処理センター運営委員会

入退室管理システムの導入と来年度の土日運営、新システムの導入経緯・特徴、総合情報基盤センターについての経緯・現状について審議。

19(水) 管理者対象利用者説明会(部局教職員用サーバ)

第134回センター会議

次期システム、入退室管理システム、報告、その他について協議。

第129回NEC連絡協議会

メーカー側より前回協議会以降のシステム整備状況について作業報告を受け、センターとしてより具体的に不具合な状況や要望を伝えた。

平成14年度利用申請一覧（研究・事務）

	所 属	氏 名	研 究 課 題	備 考
1	教育学部	青 木 昌 三	物理学に於けるいくつかの問題のシミュレーション	
2	教育学部	高 井 忠 昌	立方格子における分散曲線	
3	教育学部	土 屋 盛 茂	論理実証主義およびポパー哲学の研究	
4	教育学部	黒 田 勉	研究	
5	教育学部	松 島 欣 哉	New England Transcendentalists,特に Henry David Thoreau の研究	
6	教育学部	平 篤 志	多民族社会における少数派コミュニティの特性に関する研究	
7	教育学部	佐 藤 明 宏	国語科教育に関する研究	
8	教育学部	藤 元 恭 子	幼児の身体表現に関する研究	
9	教育学部	西 岡 圭 子	教育方法に関する研究	
10	教育学部	加 藤 みゆき	茶の風味に関する研究	
11	教育学部	森 征 洋	風の局地性に関する研究	
12	教育学部	森 征 洋	宇宙・地球科学データベース	
13	教育学部	時 岡 晴 美	生活様式の継承と変化に関する実証的研究	
14	教育学部	安 西 一 夫	情報交換とフーリエ解析の応用	
15	教育学部	深 石 博 夫	数学の研究及び通信	
16	教育学部	内 藤 浩 忠	整数論	
17	教育学部	島 田 泰 子	各種データベース類のオンライン利用	
18	教育学部	小 椋 博	社会化の理論と余暇研究	
19	教育学部	野 崎 武 司	スポーツマネージメント	
20	教育学部	植 村 典 昭	体育関係情報収集（電子メール他）	
21	教育学部	松 下 文 夫	マルチメディアの教育効果	
22	教育学部	小 川 育 子	親水性ポリマーの膨潤	
23	教育学部	宮 崎 英 一	計算機による光学パターンの解析	
24	教育学部	山 本 茂 喜	国語科単元学習における評価法の研究	
25	教育学部	日 野 陽 子	PPP ダイアルアップ	
26	附属幼稚園	塩 田 知 子	インターネットと幼稚園教育	
27	附属坂出小学校	西 浦 雅 弘	情報教育におけるインターネットの研究	
28	附属幼稚園	高 井 忠 昌	幼児教育におけるインターネットの利用	
29	教育学部	松 村 雅 文	光散乱の計算と、その星間現象への応用	
30	教育学部	小 池 和 男	素粒子の世代構造の研究	
31	教育学部	村 山 聡	経済分析	
32	教育学部	田 北 晋 一	図形処理	
33	教育学部	山 下 直 子	日本語・日本事情教材の作成の資料収集・連絡のため	
34	教育学部	高 橋 尚 志	Photoemission spectrum data の解析と理論計算	
35	教育学部	高 橋 龍 夫	近現代文学に関するデータ検索	
36	教育学部	秋 山 智	グラフィックデザインにおけるコンピュータ処理	
37	教育学部	青 山 夕 夏	情報交換	
38	教育学部	岡 田 順 直	数学の研究及び通信	
39	教育学部	小 方 朋 子	障害児教育における地域ネットワーク作り	
40	教育学部	松 本 康	教科教育に関する調査研究	
41	教育学部	安 東 恭一郎	研究	

	所 属	氏 名	研 究 課 題	備 考
42	教育学部	石 川 雄 一	学内 LAN への PPP 接続	
43	教育学部	岡 田 みゆき	家庭科教育における教科書の変遷	
44	教育学部	高 倉 良 一	法学教育に関する情報収集	
45	教育学部	Paul Batten	Ethnography of Compensatory Communicative Strategies	
46	附属高松中学校	日 詰 裕 雄	社会科教育におけるインターネットの利用の在り方	
47	教育学部	服 部 兼 敏	心理測定	
48	教育学部	佐々木 信 行	化学データベース作成	
49	教育学部	佐 藤 明 宏	国語科教育に関する研究	追加申請
50	教育学部	永 尾 智	英語助動詞の歴史的研究	
51	教育学部	小 椋 博	マルチメディアリテラシーの授業研究	追加申請
52	法学部	松 本 タ ミ	法学における情報処理	
53	法学部	木 下 麻奈子	法学における情報処理	
54	法学部	神 江 伸 介	選挙研究	
55	法学部	池 端 忠 司	インターネットをはじめとするマルチメディアの法制度研究	
56	法学部	上 田 信太郎	刑事訴訟法の研究(証拠法関係)	
57	法学部	湯 山 智 之	国連国際法委員会の国家責任条文草案の起草作業に関	
58	法学部	松 井 康 浩	地域国際化に関する動向	
59	法学部	尾 崎 壽 子	データ処理	
60	法学部	松 尾 邦 之	法律関係情報の収集	
61	法学部	石 井 一 也	開発論・平和研究	
62	法学部	小 桜 純	年金総合研究センターとの連絡	
63	法学部	上 村 貞 美	名誉毀損法の実証的研究	
64	法学部	高 橋 正 俊	研究	
65	法学部	石 橋 可奈美	国際環境法の研究	
66	法学部	堤 英 敬	選挙等に関するデータ等の収集	
67	経済学部	久 松 博 之	計量経済モデルの推定と検定	
68	経済学部	増 井 久 之	情報リテラシー、経営情報システム、電子政府電子商取引、ビジネスモデル	
69	経済学部	長 山 貴 之	日本の予算に関する実証分析	
70	経済学部	岡 田 徹太郎	経済政策	
71	経済学部	雲 和 広	地域経済成長と地域間生産要素移動に関する研究	
72	経済学部	安 藤 博 子	UNIXの利用	
73	経済学部	三 谷 宗 子	UNIXシステム管理	
74	経済学部	原 直 行	経済研究	
75	経済学部	興 梶 紀 子	設計品質に関する研究	
76	経済学部	姚 峰	経済モデル解析	
77	経済学部	高 木 文 夫	研究	
78	経済学部	斉 藤 和 也	西洋古典学関係の情報収集	
79	経済学部	山 田 勇	モデム接続用	
80	経済学部	中 村 邦 彦	情報処理技法の研究	
81	経済学部	安 井 修 二	産業循環の分析	
82	経済学部	板 倉 宏 昭	研究	
83	経済学部	井 上 貴 照	国際経済分析	
84	経済学部	大 野 拓 行	日本経済の分析	
85	経済学部	大 野 拓 行	計量経済分析	

	所 属	氏 名	研 究 課 題	備 考
86	経済学部	本 田 道 夫	知的情報処理の研究	
87	経済学部	木 村 住 枝	経済分析	
88	経済学部	王 維	データ処理	
89	経済学部	尾 崎 万 枝	経営分析	
90	経済学部	原 真 志	産業集積に関する研究	
91	経済学部	清 水 真 志	政治経済学(経済原論)の研究	
92	経済学部	佐 藤 公 一	西洋思想関係の情報検索	
93	経済学部	三 原 麗 珠	人文・社会・自然科学の研究	
94	経済学部	佐 藤 幹 子	日本経済分析	
95	経済学部	西 尾 富美子	地域社会分析	
96	経済学部	渡 辺 己	データ処理	
97	経済学部	曾 道 智	コンフリクトのゲーム理論解析	
98	経済学部	藤 村 和 宏	研究(サービス・マーケティングの研究)	
99	経済学部	梶 脇 裕 二	研究	
100	経済学部	星 野 良 明	経済理論の研究	
101	経済学部	金 澤 理恵子	教育のサポート	
102	経済学部	金 東 吉	経営資料検索	
103	経済学部	平 井 健 之	Eメールの利用	
104	経済学部	西 山 一 郎	地方自治体の予算編成過程の調査	
105	経済学部	藤 井 宏 史	マクロ経済のシミュレーション	
106	経済学部	最 上 英 明	電子メールの利用	
107	経済学部	関 義 雄	設計品質に関する研究	
108	経済学部	宮 脇 秀 貴	現代企業の会計情報システム分析	
109	経済学部	喜 田 恵津子	財務分析	
110	経済学部	小 宮 一 高	研究にともなう資料収集のため	
111	経済学部	細 川 滋	経済分析	
112	経済学部	大 賀 睦 夫	E-Mail, NetWareの利用	
113	経済学部	松 岡 久 美	経営関連情報の検索のため	
114	経済学部	水 野 康 一	e-mailの利用、英語教育関係の情報収集	
115	経済学部	高 橋 明 郎	台湾関係資料の収集	
116	経済学部	井 上 信 一	日系企業に於ける管理会計の実態	
117	経済学部	井 上 信 一	日系企業の原価管理システムの研究	
118	経済学部	井 上 信 一	グローバル企業の原価管理システムの研究	
119	経済学部	井 上 信 一	在英日系企業の原価管理	
120	経済学部	井 上 信 一	在英日系企業の管理会計の調査	
121	経済学部	井 上 信 一	利益管理情報と原価管理情報に関する調査研究	
122	経済学部	崔 康 植	研究	
123	経済学部	岡 田 ツヤ江	経営分析	
124	経済学部	横 山 佳 充	文章処理・統計処理	
125	経済学部	守 矢 信 明	フランス文化圏の資料収集	
126	経済学部	佐 藤 保 久	研究及び情報収集・検索	
127	経済学部	植 木 英 治	Eメール、掲示板	
128	経済学部	藤 原 敦 志	金融論の検証	
129	経済学部	RALPH DEGEN	授業用及び研究資料のオンライン検索	
130	経済学部	山 口 博 幸	企業行動の多変量解析	

	所 属	氏 名	研 究 課 題	備 考
131	経済学部	井 上 善 弘	研究	
132	経済学部	山 下 隆 資	情報収集	
133	経済学部	LRONG. LIM	経営・統計分析	
134	経済学部	佐 藤 忍	国際労働力移動の研究	
135	経済学部	ラナデ . R . R	数理経済学	
136	経済学部	穴 戸 榮 徳	オペレーションズ・リサーチの応用	
137	経済学部	穴 戸 榮 徳	コンピュータネットワークの有効利用に関する研究	
138	経済学部	穴 戸 榮 徳	地域経済学に於ける実証研究	
139	経済学部	高 塚 創	地域成長の数値解析	
140	経済学部	田 中 嘉 穂	原価管理の調査データ分析	
141	経済学部	古 川 尚 幸	設計品質に関する研究	
142	経済学部	山 村 能 郎	地価予測システムに関する研究	
143	経済学部	朴 鏡 杓	管理会計研究	
144	経済学部	辻 川 尚 紀	研究	
145	経済学部	桑 原 正 行	会計情報検索のため	
146	経済学部	安 井 敏 晃	米国の保険学研究誌の論文、記事入手	
147	経済学部	阿 部 文 雄	経済理論の研究	
148	経済学部	堀 啓 造	マーケティングの心理的研究	
149	経済学部	渡 邊 英 夫	フランス現代文化研究のための Email とおしての資料収集	
150	工学部	郭 書 祥	マイクロ能動ガイドワイヤカテーテルシステム	
151	工学部	和 田 隆 広	研究	
152	工学部	寺 林 優	地球科学データベース構築	
153	工学部	秦 清 治	学外からメールサーバへのアクセスのため	
154	工学部	岩 本 直 樹	研究	
155	工学部	松 原 行 宏	感性工学と製品開発に関する基礎的研究	
156	工学部	高 津 義 典	研究	
157	工学部	香 川 考 司	研究	
158	工学部	FINN, William Daniel	Foundation Geodyuamics	
159	工学部	堀 川 洋	パターン認識に関する研究	
160	工学部	服 部 哲 郎	パターン情報の処理と認識に関する研究	
161	工学部	澤 田 秀 之	研究	
162	工学部	角 田 直 人	高分解能電子顕微鏡像の計算機シミュレーション	
163	工学部	佐々木 孝	数値流体解析	
164	工学部	佐々木 孝	環境システム工学	
165	工学部	佐々木 孝	海洋データ処理	
166	工学部	佐々木 孝	水システム解析	
167	工学部	山 崎 敏 範	パターン情報処理	
168	工学部	山 崎 敏 範	感性情報処理の研究	
169	工学部	古 川 善 吾	ソフトウエアの分散開発方式の研究	
170	工学部	加 藤 大志朗	画像処理・ネットワークソフトに関する研究	
171	工学部	吉 田 秀 典	大規模地下空洞掘前解析とその評価	
172	工学部	吉 村 英 徳	学外からメールサーバへのアクセスのため	
173	工学部	富 永 浩 之	情報科学の教育研究	
174	工学部	仲 谷 英 夫	地球科学データベース構築	



	所 属	氏 名	研 究 課 題	備 考
175	工学部	生 越 重 章	モバイルマルチメディアの研究	
176	工学部	吉 野 文 雄	研究	
177	工学部	中 西 俊 介	非線型光学効果の数値計算	
178	工学部	今 井 慈 郎	計算機ネットワーク環境の実証的研究	
179	工学部	今 井 慈 郎	ネットワークセキュリティに関する基礎的研究	
180	工学部	石 川 浩	信頼性工学に関する研究	
181	工学部	守 田 秀 則	土地利用モデルの開発	
182	工学部	呉 景 龍	研究	
183	工学部	井 面 仁 志	学外からのメールサーバへの接続	
184	工学部	白 木 渡	学外からのメールサーバへの接続	
185	工学部	末 永 慶 寛	海域環境変化の数値シミュレーション	
186	工学部	井 面 仁 志	学外からのメールサーバへの接続	
187	工学部	石 井 知 彦	スピン分極相対論電子状態の解析	
188	農学部	山 本 弘 幸	病害抵抗性の発現機構に関する研究	
189	農学部	川 浪 康 弘	有機化合物の構造解析	
190	農学部	片 岡 郁 雄	果樹生産分野における研究打ち合わせと資料収集	
191	農学部	早 川 茂	食品蛋白質の生理的・加工的特性の解析	
192	農学部	亀 山 宏	環境政策シミュレーションCGEモデルの構築	
193	農学部	竹 川 薫	分裂酵母の塩基配列及び遺伝子構造の解析	
194	農学部	佐 藤 正 資	生理活性試験結果のデータ処理	
195	農学部	高 村 武二郎	園芸植物の種苗生産に関する研究	
196	農学部	永 瀬 雅 啓	生体防御機構に関する研究	
197	農学部	多 田 邦 尚	E-mailによるパソコン通信,海洋環境データ処理	
198	農学部	麻 田 恭 彦	担子菌類の分子生物学	
199	農学部	秋 光 和 也	e-mail	
200	農学部	古 本 敏 夫	情報検索等	
201	農学部	豊 田 正 範	作物学における統計解析	
202	農学部	別 府 賢 治	果樹生産分野における資料収集	
203	農学部	田 村 啓 敏	食品化学に関する緒研究	
204	農学部	松 村 伸 二	土地被覆分類の統計解析	
205	農学部	石 田 智 之	農地工学における数値計算	
206	附属図書館	岩 澤 尚 子	学生用PCの管理	
207	附属図書館	藤 谷 栄	図書受入目録業務	
208	附属図書館	竹 島 雅 美	メディアプラザのメンテナンス	
209	附属図書館	櫛 橋 一 雅	図書館システム管理 メディアプラザ管理	
210	附属図書館	龍 満 馨	メディアプラザPC利用・管理	
211	附属図書館	鵜 川 多歌子	他機関のデータベース等利用・メディアプラザ管理	
212	附属図書館	片 山 恒 信	図書館システムのメンテナンス	
213	附属図書館	鈴 木 美智子	メディアプラザPC利用・管理	
214	附属図書館	池 田 喜美代	他機関のデータベース利用のため・利用者用PC管理	
215	附属図書館	山 本 博 子	メディアプラザPC利用・管理	
216	保健管理センター	鎌 野 寛	学外データベース利用、研究データ処理	
217	遺伝子実験施設	清 末 知 宏	高等植物の遺伝子解析	
218	遺伝子実験施設	清 末 知 宏	高等植物の遺伝子解析	取消申請
219	大学教育開発センター	稲 永 由 紀	研究のため	

# 平成14年度情報処理センター利用状況

## 情報処理センターPCルーム1, 2, 3

		月	火	水	木	金
8:50	前				都市地域分析 FW 消費者調査法	
	後		経済統計			
10:20	前	データ管理論	都市地理学基礎 演習	マルチメディアリテラシ	経営財務論 産業情報システム論 A	産業情報システム特殊講義
	後				経営環境システム論	産業情報システム論 B 経済統計 統計学
12:00	前	経営財務論	情報処理入門イ	情報処理入門ハ		教養ゼミナール
	後					
13:00	前	経営財務論	情報処理入門イ	情報処理入門ハ		教養ゼミナール
	後					
14:30	前	演習 演習	演習 演習		経済統計 A 特殊講義	コミュニティア・イングリッシュ
	後	演習 演習	演習 空間計量分析			
16:10	前	情報処理基礎 計算機基礎	情報処理基礎	情報処理入門ロ	演習	コミュニティア・イングリッシュ
	後	人間環境教育基礎演習	演習		演習 演習	演習
16:20	前	情報処理基礎 計算機基礎	情報処理基礎	情報処理入門ロ	演習	コミュニティア・イングリッシュ
	後	人間環境教育基礎演習	演習		演習 演習	演習
17:50	前				産業情報システム 特別講義	
	後	主題科目				統計学
19:30	前				産業情報システム 特別講義	
	後	主題科目				統計学
19:40	前					
	後	主題科目				
21:10	前					
	後	主題科目				
備考		集中講義:生活情報処理、生活情報処理演習、大学院演習 法学部情報処理ガイダンス、経済学部情報処理ガイダンス、生涯学習センター講義(経営学への招待、 電子メール・WWW 入門講座、法学部における「IT革命」)				

# 平成14年度情報処理センター利用状況

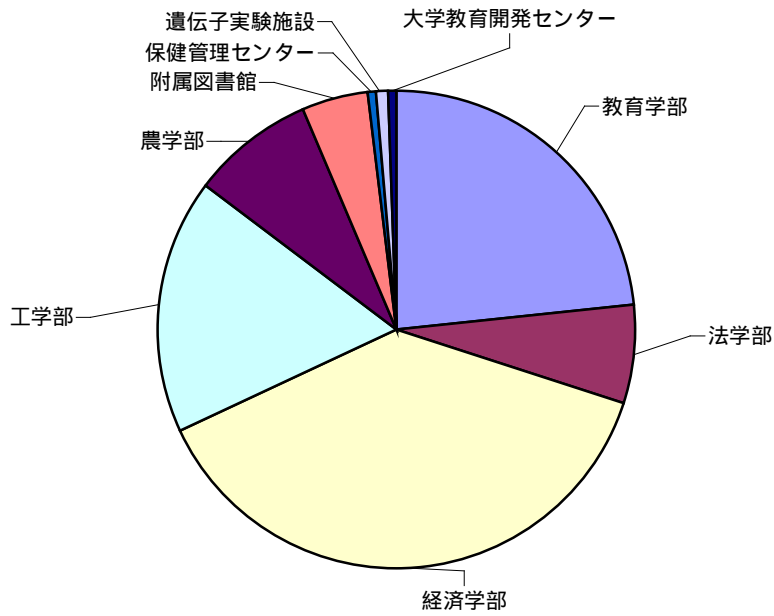
## 教育学部情報処理教室1, 2

		月	火	水	木	金
8:50	前					
	後			マルチメディアリテラシ		
10:20						
10:30	前	数学科教育論		マルチメディアリテラシ		
	後					
12:00						
13:00	前		数学情報論			
	後					
14:30						
14:40	前					
	後		マルチメディアリテラシ		マルチメディアリテラシ	コミュニケーション・イングリッシュ
16:10						
16:20	前	情報処理基礎	マルチメディアリテラシ			
	後	マルチメディアリテラシ				コミュニケーション・イングリッシュ
17:50						
18:00	前					
	後					
19:30						
19:40	前					
	後					
21:10						
備考	集中講義等: 数学特別講義、英語学特講					

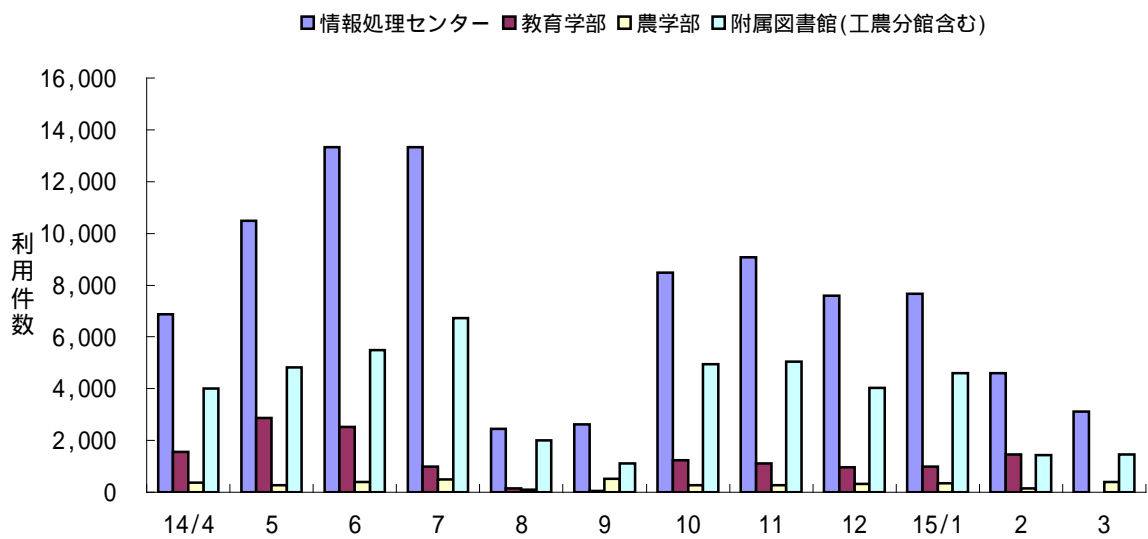
注:この表に記載の授業名は情報処理センターに通知のあったものである。

# 計算機利用統計

## 平成14年度利用申請



## 平成14年度パソコン室利用状況



## 平成14年度パソコン室利用統計

### 情報処理センター

#### 利用件数

	14/4	5	6	7	8	9	10	11	12	15/1	2	3	計
教育学部	168	157	528	867	288	191	539	453	413	511	447	159	4,721
法学部	1,639	2,495	2,366	2,377	579	690	2,266	2,066	1,398	1,699	795	452	18,822
経済学部	4,955	7,672	10,266	9,751	1,481	1,611	5,445	6,383	5,631	5,304	3,255	1,587	63,341
工学部	7	15	50	150	27	26	73	68	61	55	49	31	612
農学部	99	142	128	134	50	107	116	84	61	64	40	304	1,329
その他	3	2	3	54	7	4	35	33	32	26	20	583	802
計	6,871	10,483	13,341	13,333	2,432	2,629	8,474	9,087	7,596	7,659	4,606	3,116	89,627

#### 延べ利用時間(分)

	13/4	5	6	7	8	9	10	11	12	15/1	2	3	計
教育学部	6,234	7,105	27,863	49,276	18,751	12,969	30,153	28,362	25,443	39,020	28,075	11,304	284,555
法学部	83,334	123,688	123,058	122,508	35,627	42,904	103,805	106,892	77,484	109,333	43,241	34,663	1,006,537
経済学部	256,552	441,685	607,379	599,678	94,910	107,475	278,794	402,572	365,864	333,683	191,340	94,731	3,774,663
工学部	214	958	2,503	7,947	1,054	1,556	3,345	4,010	3,620	2,347	2,334	918	30,806
農学部	6,103	11,949	9,420	11,209	5,046	8,843	9,859	7,177	5,422	5,612	4,226	4,906	89,772
その他	30	4	146	2,607	216	199	1,487	2,076	1,962	1,334	1,274	15,892	27,227
計	352,467	585,389	770,369	793,225	155,604	173,946	427,443	551,089	479,795	491,329	270,490	162,414	5,213,560

### 教育学部情報処理教室

#### 利用件数

	14/4	5	6	7	8	9	10	11	12	15/1	2	3	計
教育学部	1,376	2,283	1,904	643	100	40	1,089	903	755	824	1,075	0	10,992
法学部	29	61	30	4	18	2	62	49	19	25	9	0	308
経済学部	131	478	573	342	29	2	60	145	172	126	354	0	2,412
工学部	13	7	9	0	3	2	27	0	1	1	2	0	65
農学部	1	24	13	1	2	0	0	0	0	0	3	0	44
その他	0	0	0	0	0	0	0	0	0	0	0	0	0
計	1,550	2,853	2,529	990	152	46	1,238	1,097	947	976	1,443	0	13,821

#### 延べ利用時間(分)

	14/4	5	6	7	8	9	10	11	12	15/1	2	3	計
教育学部	81,508	142,759	119,825	44,712	8,024	3,438	67,925	72,189	67,974	69,537	76,067	0	753,958
法学部	1,322	3,541	1,949	122	852	96	3,017	3,100	1,304	1,660	432	0	17,395
経済学部	7,777	32,011	45,477	26,175	1,023	120	2,892	8,133	9,257	7,176	17,351	0	157,392
工学部	734	492	504	0	260	213	740	0	7	32	137	0	3,119
農学部	8	1,566	695	29	102	0	0	0	0	0	78	0	2,478
その他	0	0	0	0	0	0	0	0	0	0	0	0	0
計	91,349	180,369	168,450	71,038	10,261	3,867	74,574	83,422	78,542	78,405	94,065	0	934,342

## 農学部情報処理教室

### 利用件数

	14/4	5	6	7	8	9	10	11	12	15/1	2	3	計
教育学部	0	0	0	1	0	0	0	0	0	0	0	0	1
法学部	0	0	0	2	0	0	0	0	0	0	0	0	2
経済学部	0	0	0	8	1	0	0	0	0	0	0	0	9
工学部	0	0	0	0	0	0	0	0	0	0	0	0	0
農学部	357	271	397	479	84	510	262	265	322	334	131	379	3,791
その他	0	0	0	0	0	0	0	0	0	0	0	8	8
計	357	271	397	490	85	510	262	265	322	334	131	387	3,811

### 延べ利用時間(分)

	14/4	5	6	7	8	9	10	11	12	15/1	2	3	計
教育学部	0	0	0	61	0	0	0	0	0	0	0	0	61
法学部	0	0	0	159	0	0	0	0	0	0	0	0	159
経済学部	0	0	0	687	40	0	0	0	0	0	0	0	727
工学部	0	0	0	0	0	0	0	0	0	0	0	0	0
農学部	26,153	23,336	31,680	42,443	5,910	36,945	17,909	19,916	22,487	30,443	13,370	50,763	321,355
その他	0	0	0	0	0	0	0	0	0	0	0	904	904
計	26,153	23,336	31,680	43,350	5,950	36,945	17,909	19,916	22,487	30,443	13,370	51,667	323,206

## 附属図書館(工学部及び農学部分館含む)

### 利用件数

	14/4	5	6	7	8	9	10	11	12	15/1	2	3	計
教育学部	696	526	614	1,175	321	73	896	916	767	899	171	171	7,225
法学部	825	902	842	933	240	139	763	727	573	737	136	180	6,997
経済学部	1,511	1,702	2,168	2,437	477	257	1,584	1,780	1,343	1,533	309	340	15,441
工学部	194	367	466	537	128	51	355	425	365	381	140	36	3,445
農学部	789	1,327	1,396	1,640	821	583	1,355	1,206	992	1,044	667	516	12,336
その他	0	0	0	0	0	0	0	0	0	0	0	207	207
計	4,015	4,824	5,486	6,722	1,987	1,103	4,953	5,054	4,040	4,594	1,423	1,450	45,651

### 延べ利用時間(分)

	14/4	5	6	7	8	9	10	11	12	15/1	2	3	計
教育学部	32,798	25,159	28,536	54,394	14,260	4,363	42,488	48,948	43,942	52,106	9,333	8,508	364,835
法学部	34,611	37,055	38,134	36,149	11,746	7,005	32,295	37,652	26,575	37,463	6,901	7,242	312,828
経済学部	66,804	79,963	102,400	105,352	23,799	13,455	69,450	85,037	70,192	77,614	15,009	17,339	726,414
工学部	7,760	11,336	14,309	19,447	5,770	2,231	10,573	14,660	10,036	10,541	3,984	1,462	112,109
農学部	35,034	63,177	65,243	80,086	39,212	27,847	59,851	61,375	50,535	58,374	42,762	31,803	615,299
その他	0	0	0	0	0	0	0	0	0	0	0	11,969	11,969
計	177,007	216,690	248,622	295,428	94,787	54,901	214,657	247,672	201,280	236,098	77,989	78,323	2,143,454





# 資 料



## 香川大学総合情報基盤センター規則

### (趣旨)

第1条 この規則は、香川大学学則第10条第2項の規定に基づき香川大学総合情報基盤センター（以下「センター」という。）に関し必要な事項を定める。

### (目的)

第2条 センターは、香川大学（以下「本学」という。）における学内共同教育研究施設として情報通信技術（マルチメディア技術及びネットワーク技術を含む。以下同じ。）を活用した基盤の管理運用及びその支援を行い、教育研究及び大学運営に供するとともに、情報通信技術及びその応用に関して高度な研究開発を行い、その発展に寄与すると同時に本学における情報環境の高度化を推進し、また、学内外への情報発信基盤として学内及び地域の連携強化を進めることを目的とする。

### (業務)

第3条 センターは、次の各号に掲げる業務を行う。

- (1) キャンパス情報ネットワーク、センターのサーバ類及び情報処理教育システムの構成、維持管理及び利用に関すること。
- (2) 学術研究のための情報通信技術の研究に関すること。
- (3) 学術研究の情報通信技術の活用に関すること。
- (4) キャンパス情報ネットワークを活用した分散システムの構築及び連携に関すること。
- (5) 学内外への情報通信技術を用いた情報システムの構築、提供及び利用に関すること。
- (6) 情報通信技術の教育に関すること。
- (7) 情報通信技術を活用した教材作成に関すること。
- (8) 国立情報学研究所、全国共同利用の大型計算機センター及びその他学外の情報ネットワークとの連携に関すること。
- (9) その他大学の情報化推進に関すること。

### (部門)

第4条 センターに情報基盤システム部門、情報応用システム部門及び情報教育システム部門を置く。

- 2 情報基盤システム部門は、キャンパス情報ネットワーク及びセンターのサーバ類の構成及び維持管理並びに情報通信技術の基盤となる技術の研究
- 3 情報応用システム部門は、情報通信技術を応用した各種データベースの構築、利用及び支援並びに情報通信技術の応用に関する研究
- 4 情報教育システム部門は、教育用計算機システムの構築及び運用、教育用教材の開発並びに情報通信技術の教育に関する研究

### (分室)

第5条 センターは、幸町キャンパス、林町キャンパス及び三木町農学部キャンパスに、分室を置く。

2 分室に関し必要な事項は、別に定める。

(職員)

第6条 センターに、次の各号に掲げる職員を置く。

- (1) センター長
- (2) 部門長
- (3) 専任教員(以下「センター教員」という。)
- (4) その他必要な職員  
(センター長)

第7条 センター長の選考は、本学専任教授の中から部局長会議の議に基づき、学長が行う。

2 センター長は、センターの業務を掌理する。

3 センター長の任期は2年とし、再任を妨げない。ただし、センター長が辞任をした場合又は欠員となった場合の後任者の任期は、前任者の残任期間とする。

(センター長の選考時期)

第8条 センター長の選考は、次の各号の一に該当する場合に行う。

- (1) 任期が満了するとき。
- (2) 辞任を申し出たとき。
- (3) 欠員となったとき。

2 センター長の選考は、前項第1号の場合には、任期満了の1月以前に、同項第2号又は第3号の場合には、速やかに行うものとする。

(部門長)

第9条 部門長の選考は、センター教員の中からセンター長の推薦に基づき、学長が行う。

2 部門長は、部門の業務を掌理する。

(センター教員の選考)

第10条 センター教員の選考は、部局長会議の議に基づき、学長が行う。

(運営委員会)

第11条 センターの管理及び運営に関する重要事項を審議するため、運営委員会を置く。

2 運営委員会に関し必要な事項は、別に定める。

(事務)

第12条 センターの事務は、部門が関係する学部事務部の協力を得て、学術研究協力部研究協力課において処理する。

(雑則)

第13条 この規則に定めるもののほか、センターに関し必要な事項は、別に定める。

附 則

この規則は、平成15年10月1日から施行する。

## 香川大学総合情報基盤センター運営委員会規則

### (趣旨)

第1条 この規則は、香川大学総合情報基盤センター規則第9条第2項の規定に基づき、香川大学総合情報基盤センター運営委員会(以下「運営委員会」という。)に関し必要な事項を定める。

### (任務)

第2条 運営委員会は、総合情報基盤センター(以下「センター」という。)の円滑な運営を図るため、次の各号に掲げる事項を審議する。

- (1) センターの運営の基本方針に関する事。
- (2) センターの事業計画に関する事。
- (3) センターの予算及び施設・設備に関する事。
- (4) その他センターの管理運営に関する重要事項

### (組織)

第3条 運営委員会は、次の各号に掲げる委員をもって組織する。

- (1) センター長
- (2) センターの部門長
- (3) 各学内共同教育研究施設(センターを除く。)から選出された教員各1人
- (4) 各学部及び医学部附属病院から選出された教員各1人
- (5) 保健管理センターから選出された教員1人
- (6) 企画部長
- (7) 学術研究協力部長
- (8) 学務部長

2 前項第3号から第5号までの委員は、学長が任命する。

3 第1項第3号から第5号までの委員の任期は、2年とし、再任を妨げない。ただし、委員に欠員が生じた場合の補欠の委員の任期は、前任者の残任期間とする。

### (委員長)

第4条 運営委員会に、委員長を置き、センター長をもって充てる。

2 委員長は、運営委員会を招集し、その議長となる。

3 委員長に事故があるときは、あらかじめ委員長が指名した委員が、その職務を代行する。

### (議事)

第5条 運営委員会は、委員の過半数の出席がなければ議事を開くことができない。

2 議事は、出席委員の過半数をもって決し、可否同数のときは、議長の決するところによる。

### (委員以外の者の出席)

第6条 運営委員会は、必要があるときは、委員以外の者の出席を求め説明又は意見を聴取することができる。

### (専門委員会)

第7条 運営委員会に、専門の事項を調査検討するため、専門委員会を置く。

2 専門委員会に関し必要な事項は、別に定める。

(事務)

第8条 委員会の事務は、学術研究協力部研究協力課において処理する。

(雑則)

第9条 この規則に定めるもののほか、運営委員会に関し必要な事項は、運営委員会が別に定める。

附 則

1 この規則は、平成15年10月1日から施行する。

2 この規則の施行後最初に任命される第3条第1項第3号から第5号の委員の任期は、第3条第3項の規定にかかわらず、平成17年3月31日までとする。

## 編集後記

「総合情報基盤センター 年報 第1号」を発刊する運びとなりました。この年報の発行については、昨年10月からの総合情報基盤センター発足後の運営体制が未だ十分には整わない中、また、4月からの国立大学法人化を控えて予算執行等々が急がれる状況下での準備・編集でありましたが、関係者のご協力を得てほぼ予定した通りの作業日程で発行に漕ぎつけることができました。

これまでの旧香川大学情報処理センター年報のスタイルを踏襲してはおりますが、掲載記事については、旧香川大学情報処理センターのスタッフ、旧香川医科大学情報メディアセンター関係者から寄せられた内容のものとなっています。新しい総合情報基盤センターへの理解を深めて頂く端緒となり得れば編集者としてはこの上ない喜びです。

総合情報基盤センターにおいては、今年7月頃には専任教官の就任を得て充実したスタッフのもとでの運営が本格的に始まることと思われます。この「年報」も、専門家集団としての専任スタッフによるより一層充実した日頃の活動に基いた、豊富な内容に富む新しいスタイルでの年報へと引き継がれるものと強く期待しているところです。

本年報の編集にあたっては、センター関係者の方々には日々何かと慌しく忙しい中であって、編集者からの無理なお願いにもかかわらず快く原稿執筆をお引き受けいただきました。また、センター幸町分室の瀬野 芳孝氏、曾根 計俊氏および高橋 多恵氏には日常の業務極めて繁多な中であって、様々な仕事を引き受けていただきました。最後になりましたが記して心から感謝申し上げます。

平成16年2月  
(平成15年度年報編集担当)

---

---

年報 第1号

2004年2月発行

編集 香川大学総合情報基盤センター

発行 香川大学総合情報基盤センター

〒761-0793 木田郡三木町池戸1750-1

087-891-2266(ダイヤルイン)

〒760-8523 高松市幸町2-1

087-832-1292(ダイヤルイン)

---

---