東京大学地震研究所技術報告，No. 5，29-38 頁，1999 年.
Technical Research Report, Earthquake Research Institute, University of Tokyo, No. 5, p. 29-38, 1999.

京都大学防災研究所技術室におけるサーバ再構築の試み
—UNIX の 2000 年問題に対応して—

平野憲雄*・吉田義則*・多河英雄*

Server Reconfigurations for UNIX Machines for Year 2000 Problems of the Division of Technical Affairs, Disaster Prevention Research Institute, Kyoto University

Norio HIRANO*, Yoshinori YOSHIDA* and Hideo TAGAWA*

Abstract

We have changed the operation systems (OS) of the Mailservers, WWW servers, and UNIX machines in order to solve Year 2000 problems.

In older version of the OS, the 4 digit datum of year is represented only by the last 2 digits, so the digit year "2000" will be recognized as the same as the digit, "1900". The time sequence for earthquake data will be troubled and all other applications using the value will be in error.

We have used commercial OS software to fix the problem for the newer machines. However, since some of the older machines could not run the new OS, the problem was solved with patches to the software. Finally, both cases the OS had to be re-installed, because the application software and program development tools were changed.

Key word : UNIX, sendmail, GNU, freeware, year 2000 (y2k) problem

はじめに

京都大学防災研究所技術室に設置してあるエンジニャリングワークステーション（EWS）でサーバの再構築をした結果を報告する．今まで使用していた EWS（Sun 4/50）が旧式となり故障多発の可能性が増したため，新しい機種（Ultra-5）を購入した．古い EWS はハードディスクを交換して代替サーバの役目をさせるようにした．このサーバは防災研事務部のメール，防災研の WWW（World Wide Web），メーリングリストの役目を担っている．

ネットワークの世界は情報の収集が可能になるのと引き換えに，自分のプライバシーを如何に守るかのせめぎ合いの世界でもある．迷惑な宣伝メール（商業用メールで発信者を隠して相手を選ばず送りつけるもの）や侵入したマシンの名前を語って送り出すスパムメールなどの攻撃を受けてディスクが満杯になったり，最悪の場合は第三者からの侵入で，パスワードを盗まれたり，大切なファイルを破壊されたりする．悪意を持った外部からの侵入者（クラッカーと呼ぶ）を防ぐには，常にマシンのログ情報を探り，対策された最新のソフトウェアを入れ直して対応するしかない．やがてそれをも乗り越えて侵入してくると，また入れ直して対策するなど“いたちごっこ”の世界である．この最新情報（セキュリティを上げる）について行くには柔軟な頭脳が必要であり，若手（35 才位まで）でなければ無理だと言われている．それを，50 過ぎの技術者が挑戦し悪戦苦闘した報告である．専門外の人達でも理解しやすい様に用語の説明も取り入れた．次の節では，年輩の技術者が不利だと思われる分野になぜ挑戦したかを述べる．

1999 年 7 月 29 日受付，1999 年 11 月 5 日受理.
* 京都大学防災研究所技術室
* Division of Technical Affairs, Disaster Prevention Research Institute, Kyoto University

新しい技術の導入

技術者の年令が進むにつれて平均的ではあるが，学ぶ意欲も薄れがちである．既存の知識にプラスアルファされるのならば良いが，全く新しい分野になると時間をかけねばマスターできなくなるからである．若い年代ならば，次から次へと新しい分野を学んでいけるが，図 1 に示すように一つのソフトウェアをマスターする時間は年令とともに長

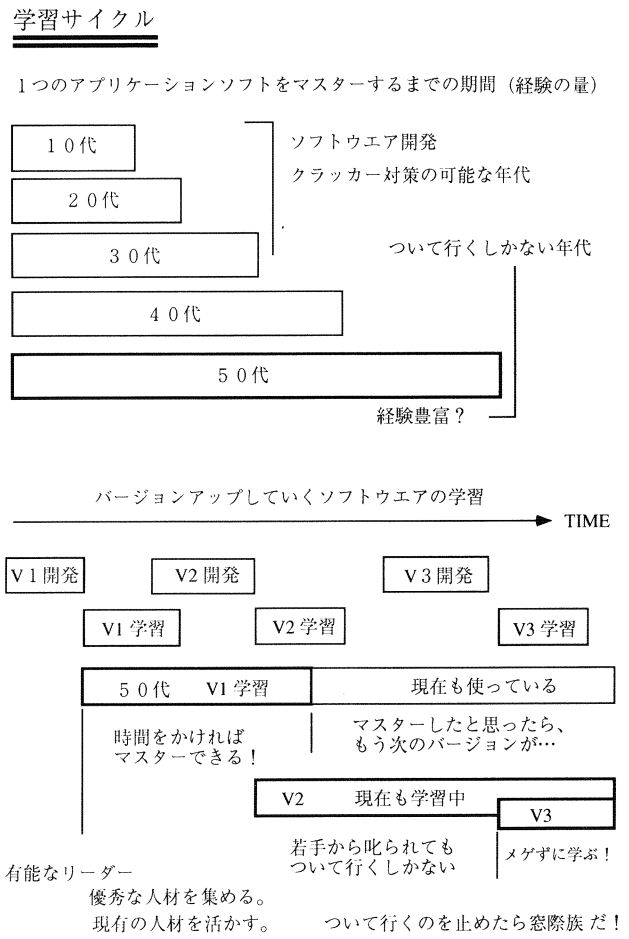

図 1. 学習サイクル（バージョンアップについていくために）

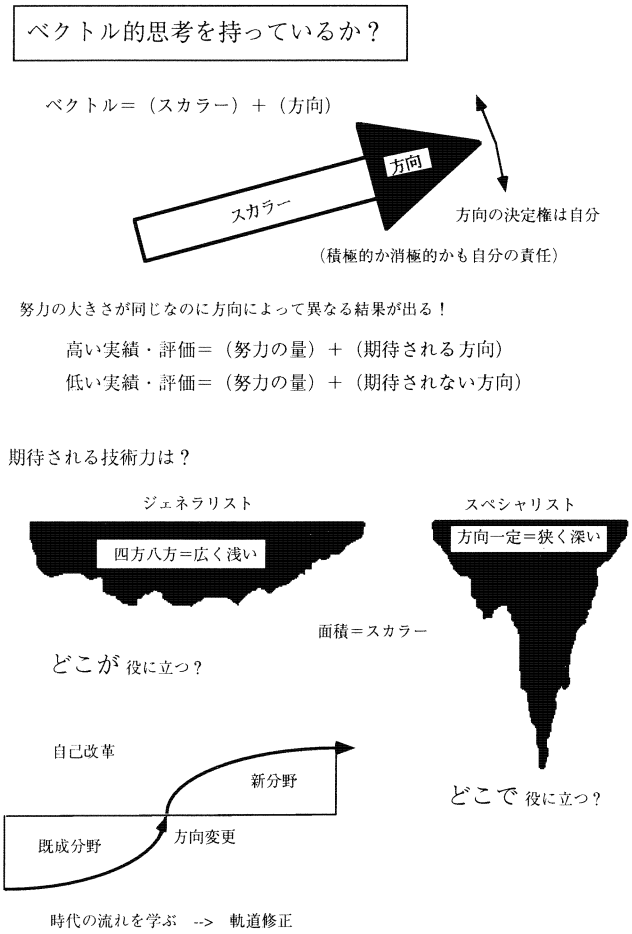

図 2. ベクトル的思考を持って自己改革をしよう．我々技術者は井の中の蛙になりやすい．

くなっていくものである．ようやく一つをマスターしたと思ったら，それはもう古い技術になっているほど技術革新の速度が早い．新しいソフトウェアが便利と判っていても現在の古いソフトウェアが問題なく動いているならば，多少の不便を覚悟してなんとか使っていく姿勢になる．

やがて，最新の機械やソフトウェアが世の中の主流になると，古いデータの処理が困難になってくる．そのデータを完全に使い切り御用済みになったならば，古い解析ソフトウェアを改良することもなかろう．問題なのは，御用済みになる以前にクラッカーなどの侵入が発生する場合である．大切なデータを破壊されるような事態になるのであれば，侵入を許さない最新のソフトウェアへと切り替えていかざるを得ない．はじめに述べたようにクラッカー侵入の防御は一時しのぎであり，常に最新のソフトウェアへと切り替える体制が必要である．

ネットワークの大きさや重要性が増すにつれて，安全性（セキュリティ）を高めるための専門の人員が新たに必要になってくる．その人員に求められるものは，苦労して築いたシステムであっても，次の最新のシステムに切り替える勇気（必要条件）と最新のソフトウェアで再構築する能力（十分条件）が要求される．

コンピュータ関係では永年築き上げた技術が無用となる場合が多い．例えば，和文タイプの国家試験に受かった人々とその台数は限られたものであったが，いまでは電子メールで誰もがワープロ技術を身につけて当然と思われる時代になっている．我々技術者にとって技術の蓄積は最大の義務でもあり誇りでもあるのは変わらないが，技術を単に積み上げたものを蓄積とせず，最新の技術の中から選別する見識と，選別した技術をどの分野に融合させていくかの努力がさらに必要になったのである．努力の量は時間をかければ大きくなるがその方向を誤ると役にたたない技術となってしまう．図２はベクトル的思考を持って努力する方向を決めねばならない事を示している．待遇に違いがあった場合の疑問に「同じ努力をしているのになぜ差が出る？」がよく聞かれる．努力の大きさだけでは判断できないのだ．その努力の方向は期待に応える方向だったのか，それとも反対の方向かあるいは方向に無頓着であったのか

が評価の違いとなって現れただけなのである．何が求められているかを察知し，その方向への努力は高い実績と評価に結びつくものと思われる．

我々技術官には，努力する方向は一定であり狭い分野で深い技術力を持っている者が多いであろう．この技術がどこで「場所」またいつまで「将来」役に立つのかを真剣に考え，さらに磨くのはどの部分「方向」かを見きわめることが大切である．もし将来性がないと判断したならば方向転換する勇気も必要であろう．いまさらそんなことはできないと，そのままの姿勢で古い技術のままでいれば，時代の流れから取り残された存在となるのは必定である．

サーバの再構築をしたと言うことは，サービス向上の義務とセキュリティ確保の責任とが生じるのである．サービス向上とセキュリティのガードを強めることは相反する方向であるが，データ保護を努力しつつ情報を利用する立場を優先したいものである．管理業務では民間業者と比較して早さでは勝負にならないが，ユーザ側の立場に立ちきめ細かい対応を売り文句にして，老骨にむち打って時代の流れについて行くしかないのである．年輩の技術者が新しい技術に挑戦すれば，ぼけ防止に役立つことは確かであろう．

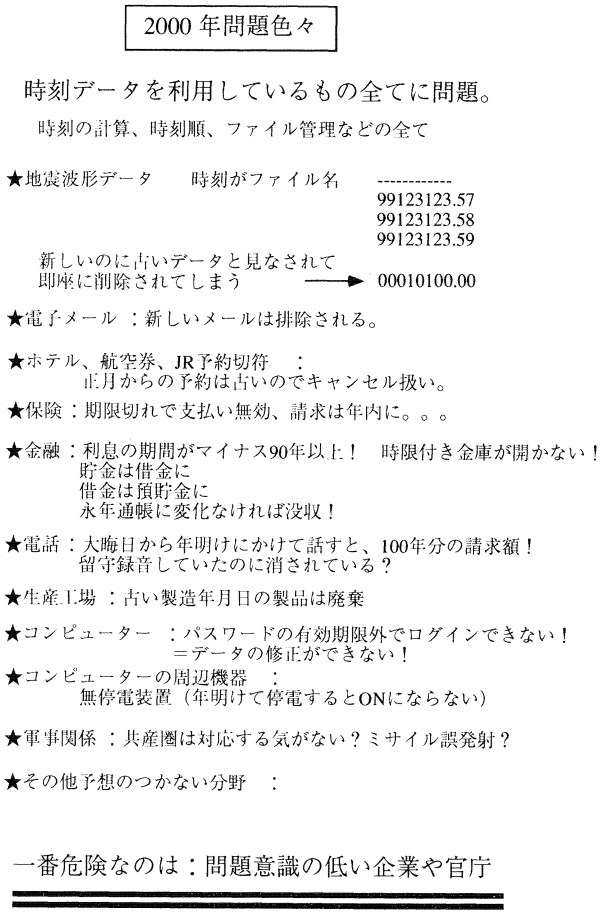

図 3.　2000年問題とは？予想される諸問題

2000年問題とは

コンピュータの2000年問題は西暦4桁の数を下位2桁で計算させているのが発生原因である．100の位が変わらない間は2桁のみの計算でも問題はなく，無条件に1900年が加算された年数になっている．2000年を超えても正しい4桁表示にするためだけに，現在使用のソフトウェアに改善と投資をしても出力結果が同じである間は割りに合わない．従って，実際の弊害が出るまで問題解決を先延ばしにしてきた理由の一つでもある．

2桁計算の場合は，2000年になると99から00になる．時刻データからさまざまなデータを作るソフトウェアでは00になる時点で狂うことになる．例えば，図3のように地震データは時刻からファイル名を作っており，2000年1月1日には最新のデータが100年前の1900年のデータになってしまう．利息だとマイナスの利子がついて一夜にして借金まみれ？になったり，金持ちになったり？，ホテルの予約が古すぎてキャンセルされたりするであろう．図3に予想される問題を列挙した．中には問題が起きてから解決する方法もあるが，パスワードの期限を設定してあるとログインができなくなり修復不可能の場合もある．でき得る限り年内に解決しておかねばならない．

2000年問題の対応

OS（Operatiing System）の場合は早くから予想されており，対応済みOSが発売されている．また古いOSの場合は対応パッチソフト（OSの部分的な修正用ソフトウェア）も用意されている．本当の問題は時間を利用してデータのスケジュール管理をしているアプリケーションソフトである．時計装置の時刻を，あるいはOSの用意した時刻を入力して工場の生産ラインの制御，データファイルの新旧の整理に始まって，契約事（利息や予約や保険期間など）を処理するなど種々雑多なものが運用されている．このように時間を利用している全てのソフトウェアには2000年問題発生の可能性を持っている．ハードウェアの側で4桁年数の時計出力を装備すれば済むのではない．年数を入力するのに西暦の下2桁しか読まないソフトウェアが問題を発生させるのである．

未対応のOSでは2桁データに1900を加算しているのが普通である．パッチを当てる処理の内容は，OSの用意した時刻データを各アプリケーションソフトに渡す場合は，時刻出力の手直しで済む．例えば，図4のように70以上は1900加算，以下は2000加算をすれば，2000年を前後する演算に問題なく対応できる（中村，1998）．一方，直接時計からデータをもらうアプリケーションソフトの場合は，それ自身の書き替え（専用のパッチ当て）が必要である．内部時計を4桁のLSIに代えても4桁を読むソフト

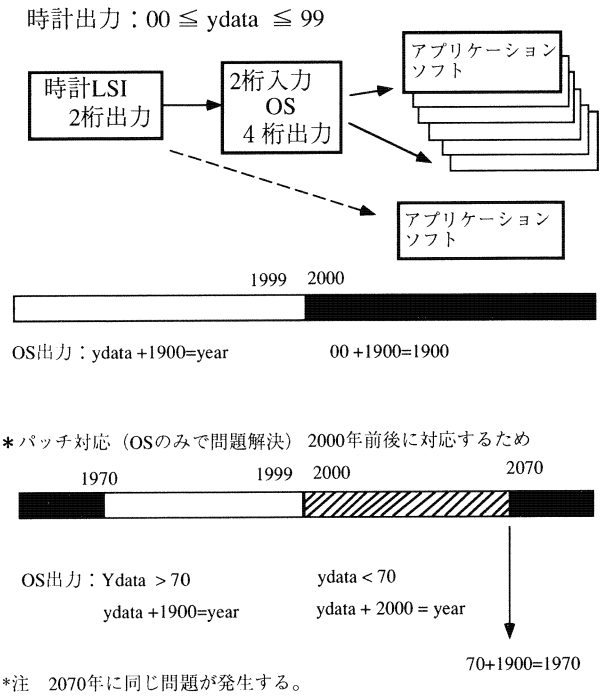

図 4. 2000 年問題の対応策
ソフトウェアだけの対策が一番安価である.

ウェアに入れ替えねばならないから，結局 OS にパッチソフトのみを当てる方が安価である．注意すべきは，安価ではあるが，問題を 2070 年後へと先送りにする（実はその前の 2038 年に 64 ビットを超える問題もある）だけで，4 桁の時計出力を利用しない限り同じ問題を抱えている．一方，時刻とは無関係のシステム（注意！メールの送受信は関係あります！）ならば，内部時計を 1980 年など過去の時刻に設定する方法もある．あるいは，2000 年を過ぎてから，始めて OS を作動させる方法もあるが，いずれも 1900 年代の世界のままであり不便であろう．

問題を深刻にさせているのは，現在運用しているソフトは 2000 年問題を抱えているかどうか不明の場合が多いことである．運用年数の長いソフトウェアになる程，実績があるため返ってソフトウェアの更新には消極的となり，対応するにしても開発者が不明であったり，忘却していたりして判断を遅らせている．また，運用する側の問題意識の希薄と，ソフトウェアの入れ替えに多額の予算を必要とすることも対応を遅らせている原因になっている．残念ながら 2000 年になる前に問題発生のソフトウェアを全て発見し対応するのは不可能と思われる．とは言え 2000 年になってからの障害を極力少なくするために，できる限り対応しておく必要がある．また，対応内容が完全でない場合（2000 年は例外的に閏年である）も予想される．従って，2000 年の 1 月 1 日から時刻やデータ異常の有無を監視しておく態勢が必要である．

対応内容は，OS とアプリケーションソフトの両方に処理をせねばならない．これから新たに導入したり開発する場合は，2000 年問題対応済みの OS やアプリケーションソフトを入れれば良い．一方，現在稼動中の OS をそのまま使用する場合は，対応パッチをあて，対応済みのアプリケーションソフトを再インストールする．

次に，対応済み OS の環境下で動作確認をする．現在稼動中のシステムで，既存のデータとの混在が新たな問題を発生させる場合は，可能ならば模擬システムで試験をしてみる．

インストール

インストールとはソフトウェアをコンピューターに記憶させて動くようにすることを言う．我々が対応したのは，UNIX マシンである．OS そのものが対応済み（Solaris 2.6 以上）ならば，その環境下でコンパイルしたアプリケーションソフトは正常に動作すると思われる．しかし，コピーするだけで動くパッケージソフトのアプリケーションソフトを直接インストールした場合は不明である（確実に問題ないと言い切れない）．古いバージョンの OS（Solaris -2.5, SunOS-4.1.3 など）をそのまま使う場合は，対応パッチを OS に当て，アプリケーションソフトの再インストールをした．

ハードディスクの交換

UNIX マシンのハードディスクは休みなく回転させるため寿命が短い．我々は古い機種の OS 立ち上げ用に新品のハードディスクを入れ替えた．ここで注意すべきは，UNIX の歴史の産物との説明であるが，ハードディスクの SCSI 番号を 3 にしておかないと OS の管理番号のゼロに対応しないことである．他の番号はそのままハードディスクの SCSI 番号と OS の管理番号は対応している．この違いが最初に判らなくて，OS が認識してくれず解決まで長い時間を浪費した．

開発環境ソフトウェアのインストールから

アプリケーションソフトの代表的なものにメールのサーバをする sendmail がある．購入時の OS に附属としてついてくるが，セキュリティ対策がなくスパムメールの被害を受けることは確実である．そこで対策済みのソフトウェアを入手して，新しい技術導入の章で述べたように最新版への入れ替えが必要である．また，最新版ほど改良が適宜

されておりsendmail用のパッチファイルも入手する必要がある．このsendmailをコンパイルすればいいのだが，さまざまな関連するソフトウェアも同時に組み込んでおく，すなわちソフトウェアの開発環境も整備しておかねばならない．もちろんそれらのソフトウェアも2000年問題の対応がなされていなければならない．これにはGNU (Gnu is Not Unix) で提供されているフリーウェアがお勧めである．

インストールは図5のように，ftp (File Transfer Protocol) サイトからソースファイルを入手し，makeツールを使ってテキストファイルになっているソースプログラムを実行可能なファイルに翻訳する（井上，1998）．コンパイルツールのmakeなどは，一番最新のオブジェクトファイルを採用するのでmakeそのものもコンパイルし直さなければならない．どれが安全かは判別できないので関係すると思われる全てをコンパイルした．

巷で流行っているFree-BSD (Free-Berkelay Software Distribution) は安いハードウェアを使い，パッケージ（実行ファイルになっているソフトウェア）のインストールで簡単に済み，今後の方向であろう．しこしことコンパイルするのはダサい？かもしれないが，Sun OSシリーズが一番安定しており，実績があるので挑戦した．

フリーウエアのダウンロード

まず，開発する任意の個人アカウントを作り，専用の作業用ディレクトリを用意しておくことをお勧めする．代表的なftpサイトは，

sunsite.sut.ac.jp/pub/archive/gnu
ftp.cs.titech.ac.jp/pub/gnu
ftp.kyoto.wide.ad.jp/mail/sendmail

などがある（井上，1998）．実際の手続きは，カレントディレクトリをファイルのコピー先のディレクトリにしておきftpコマンドでftpサイト名をつけて起動する．

%cd user-dir
%ftp sunsite.sut.ac.jp
name : anonymous
passwd :

目的のサイトにたどり着いたら，名前を尋ねてくるのでanonymousまたはftpと答え，パスワードは自分のメールアドレスを答えれば入れる．ログインに成功するとlogin okの表示とともにプロンプトがftp>になる．後はコピー元のディレクトリに移り，binコマンドでバイナリーモードにしてからgetコマンドとファイル名を付けて入手する．例えば，

ftp> cd pub/archive/gnumake
ftp> bin
ftp> get make-3.77.tar.gz
ftp> quit

で入手できる．ファイル名の拡張子tarは複数のファイルをtarコマンドで一つのファイルにアーカイブ（保存）されたもので，拡張子gzがついているのはGNUのgzipコマンドで圧縮されたファイルであることを意味している．

以下は，インストールしたソフトウェアの一覧でGNUのフリーウエアである．数字はバージョンの番号を示しており，大きい値ほど新しい．

ファイルの圧縮 (gzip-1.2.4)，複数ファイルのアーカイブ (tar-1.12)，プログラム作成ツール (make-3.77)，コンパイラ生成ツール (bison-1.25)，オンライン情報 (texinfo-3.9)，GNUのCコンパイラ (gcc-2.8.1)，ソースファイルにパッチをあてる (patch-2.5)，Perl言語インタプリタ (perl-5.005_04)，sendmail用ライブラリ (db-2.6.4)，DNSサーバプログラム (bind 812)，ネットワークのアクセス制御とlog出力 (tcp - wrappers _7.6)，メール送受信 (sendmail.8.9.3)，sendmail.cfファイルの生成ツール (CF-3.7 Wp 12)，PCへのメール送信 (qpopper 2.53)，WWWサーバ (apache_1.3.6)，メーリングツール (fml-2.2 B-snapshot 3)，FTPサーバ (wu-ftpd-2.4.2)．

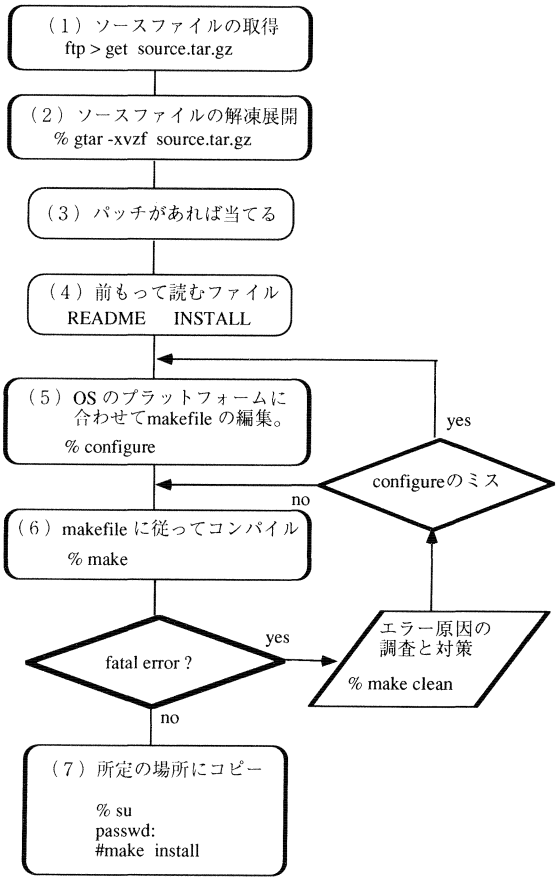

図5. ソースファイルの入手，コンパイル，インストールまでの流れ

コンパイル

GNU フリーウエアから入手したソースファイルのコンパイルは，図5のように(1)ソースファイルの収得，(2)gtarコマンドで圧縮ファイルの解凍と自動的に新しいディレクトリが作られたその下に展開，(3)もしパッチファイルが附随していれば，展開されたソースファイルに patch コマンドでそれを当てる．パッチファイルは，ソースファイルが実際に運用されだし不具合がある場合はソースファイルの訂正箇所だけを記述したものが提供されるている．(4)コンパイルの手順などの情報である README や INSTALL のテキストファイルを読んでおく，(5)自分のマシンの OS に合わせるために configure を実行して makefile の編集，(6) make コマンドで makefile の記述に沿ってコンパイラ等の実行，(7)出来上がった実行ファイルを所定の場所に make install でインストール，の順番で行う．忘れてならないのは，UNIX コマンドを自由に使えるために PATH を通しておくことである．実はこの PATH を通しておくのが不完全だったためコンパイルの途中でエラーを出し問題解決に多大な時間を浪費して大変苦労した．一般的に make を失敗したら原因を調べて対策をして make clean を実行してから(5)に戻る．もし configure の段階のミスがあれば，README のチェックと修正後(5)か(6)に戻るが，それでもダメな場合は，(1)のソースファイルの入手時の失敗も考えられるから，振り出しの ftp からやり直すこともある．バージョン番号の変なソフトを取り込んで失敗したこともある．その場合は展開したファイル全てを削除する．成功したらスーパーユーザの権限で /usr/local/bin のディレクトリに保存しておく．以上が基本であるが，make に他のオプションがついていたり，configure と make を build コマンドでまとめて行う場合もある．

gcc-2.8.1 のコンパイルをする際に，make 処理で「v7…ディレクトリーがない.」とのエラーが出て理解できないままでいたら，無視しても良いとの判断が経験者から指摘された．fatal エラーでない限り make 処理は成功したものとするという知識がなければお手上げの状態であった．参考文献だけではインストール作業は進まず，豊富な知識を持つ経験者がいなければならない事を学んだ．エラーの処理には経験の豊かな人に尋ねれば即刻解決するが，試行錯誤で努力する方が勉強になる．ただし，時間がかかるので怠けているかのように誤解される．

ここでは代表的な WWW サーバである apache_1.3.6 の場合を図6に示す．モジュール組み込みのために configure の実行が1回多い．また，httpd. conf のファイル (Laurie *et al.*, 1998) をサービスする環境（ホームページの場所，ポート番号，閲覧制限など）に合わせて編集して実際の運用方法を決定する．さらには，不特定多数のユーザーが入ってくるから，確実に動作するかを確認した後で一般ユーザに解放せねばならない．

```
1）ソースコードを入手
   www のユーザー名で login
     % ftp  ftp サイト
     ftp> get apache_1.3.6.tar.gz
   解凍、展開
       % cd  /home/www
       % tar  -xvzf apache_1.3.6.tar.gz

2）コンパイル
  2－1）モジュール組み込み用の configuration.apaci 作成
       % cd  /home/www/apache_1.3.6
       % configure
  2－2）モジュール組み込み
       % cd  /home/www/apache_1.3.6/src
       % vi configuration.apaci          編集
       % Configure -file  configuration.apaci
       % make
       % cd  ../
  2－3）コンパイルとインストール
       % make
       # make  install

3）環境と機能調整
  3－1）httpd.conf の編集
       % cd  /usr/local/apache/conf
       % vi  httpd.conf
            ホームページの場所、ポート番号、閲覧制限、etc.
  3－2）起動
       # /usr/local/apache/bin/apchectl  start
  3－3）動作確認
       ホームページの開設は？制限が効いている？
  3－4）おかしい場合はhttpd.conf の再編集
          停止
       # /usr/local/apache/bin/apchectl  stop
         httpd.conf の再編集
         再起動
       # /usr/local/apache/bin/apchectl  restart

注意！バージョンによって編集内容が変る。
```

図 6. WWW サーバ（apache）のインストール手順例

ネットワーク

複数のコンピュータ間でデータ交換をする連絡網をネットワークという．インターネットの代表的なネットワークプロトコル（protocol）に TCP/IP（Transmission Control Protocol/Internet Protocol）があり4つのネットワーク階層に分けられる．ユーザに近いところから述べると，遠隔ログイン（telnet）やファイル転送（ftp）などのネットワークアプリケーションに関するアプリケーションプロトコル，TCP や UDP（User Datagram Protocol）のトランスポートプロトコル，IP とそれに付属する形でエラー通知を行う ICMP（Internet Control Massage Protocol）のインターネットプロトコル（ちなみに ping は ICMP を利用したコマンドである.），そしてハードウェアに近いところで実際に通信を行う足まわりとしてのイーサーネット（Ethernet），X 25，専用線，ATM などの個別ネットワークに合した個別ネットワークプロトコルがある（村井ほか，1996）．特にイーサーネットは米ゼロックス社によって開発された LAN（Local Aria Network）のため

の技術で，もっとも一般的に利用されている．イーサーネット上を流れるデータはすべてイーサーネットフレームと呼ばれるパケットに入れられ，そのヘッダには上位層のマルチプロトコル環境を実現するために，世界唯一の番号からなるMAC（Media Access Control）アドレスが用いられている（國安ほか，1998）．

WWWでもメールでも基本はデータ交換でありネットワークに接続してこそ機能する．小さなネットでは情報の範囲も小さいが，全世界のインターネットに接続すれば全世界と情報交換ができる．このようにインターネットのプロトコルで作られたソフトがWWWや電子メールなどである．

DNS（Domain Name Server/ Domain Name System）

インターネットに接続したマシンには固有の32ビットのマシンアドレス（IPアドレス）がついており，IPアドレスを指定すれば互いに連絡できる．このIPアドレスだけでは覚えるのが困難で使いにくい．ドメインネームはネットワークの領域（ドメイン）や領域の中の部分領域（サブドメイン）やホストを指定するためのものである．このドメインネームでマシンのホストネームを指定すればIPアドレスに変換してくれる機能を持たせたものがDNSである（下山，1993）．

メールを配信するにはこのドメインネームを使ったアドレスになっており，基本的にDNSが機能しているネットワークに所属したマシンがメールサーバになりうる．例えば防災研のドメインネームはdpri.kyoto-u.ac.jpである．ドットで区切られた単語毎に階層構造をしておりそれぞれの階層でDNSが働いている．防災研の内部だけならばdpriにぶらさがるホストを管理するDNSマシンが対応する．いずれのDNSマシンにも登録されてないホスト名は間違っており，接続すべき相手は存在せずエラーメール扱いになる．このようにDNSを基礎にして始めてメールサーバ機能が生きてくる．従って，全世界と連絡を取りたければ連絡用のホストマシンを登録せねばならない．だたし，技術室ではDNSの管理をしておらず，専門の職員のいる京大大型計算機センターに任せている．本報告では詳しく述べない．

メールサーバ

メールサーバは不特定多数のユーザを相手にメールの送受信や中継する機能を持つ．原則として受信を拒否しないためにスパム攻撃を受け易い．防御には，送受信の窓口を1本（ハブマシン）に限定してガードを固めるのが好ましい．またメールを保存するマシン（スプールマシンでsendmailが動いている．）を用意しても，いったんハブマシンを経由する構成が望ましい．図7に構成例を示す．UNIXのメールサーバのソフトにはsendmailが使われており，実績があるため広く利用されている．現時点（1999年8月）での最新バージョンは8.9.3である．このバージョンは中継機能を制限したり，内部と外部のメールを識別して配送したり，ブラックリストに載っているマシンからのメールを拒否したりする機能を持つ．

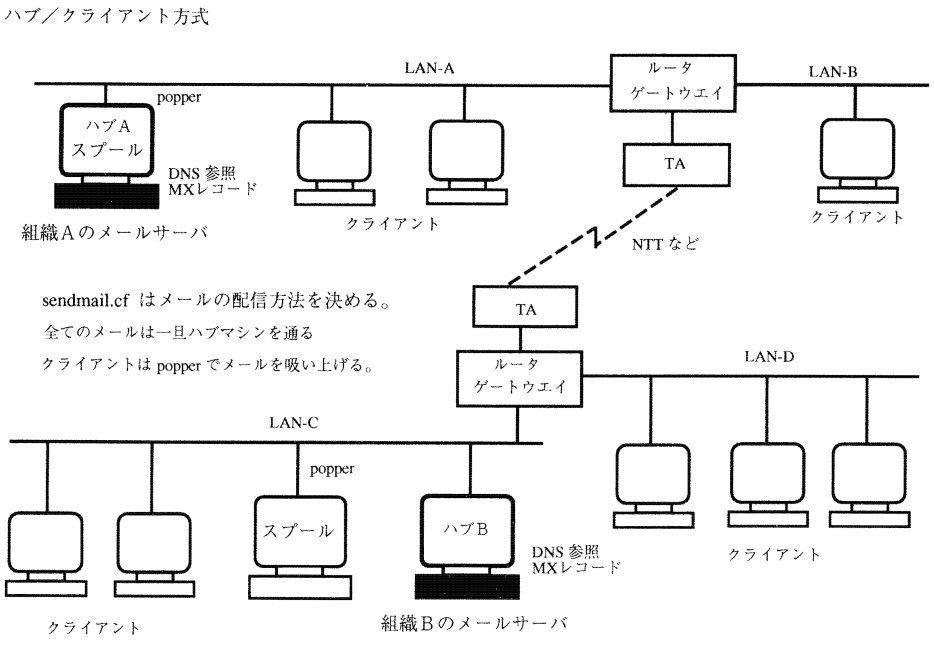

図 7.　ハブ/クライアント方式のメールサーバ構成例

スパムメールは無防備（低いバージョンのsendmailが動いている）のマシンに入り，そのマシンを踏み台にして短時間に大量のメールをまき散らす．一番の被害者はメール管理者である．瞬時にディスクのデータ領域がゴミメールで満杯になったり，本来のサーバ機能が低下したり，同じく被害を受けた他のメール管理者からの苦情を受けたりする．管理者が怠けていたらそのマシンの所属するIPアドレスがブラックリストに載せられ，最悪の場合はそのホストを含むネットワーク（組織＝所内，社内，学内）に所属する全てのマシンが全世界のメールの窓口から無条件に閉め出されることになる．

一般ユーザの認識不足から知らない間に踏み台にされている場合もある．UNIXを購入してそのまま起動すると，OSは付属の古くて無防備のsendmailも起動するため標的にされるのである．対応は，付属のsendmailを自動立ち上げしないか最新のものをインストールするかである．もし使わなければ低いバージョンのSunOS 4では起動ファイル/etc/rc.localのsendmailの行をコメントアウト（行の先頭の文字を＃文字にする）したり，SunOS 5以上のバージョンでは/etc/rc2.dにある起動ファイル名SS88sendmailの頭文字をS以外の文字に変えておけば自動起動はしない（Winsor, 1998）．外に対して安全性の高いメールサーバにするためには，ハブ/クライアント方式（図7）にすることである．これには，DNS管理ファイルにMXレコードを記述して各クライアントからの問い合わせに対してハブの存在を知らせる記述にしておき，一つの組織で扱う全てのメールはガードをしっかり設定した一つのメールサーバ（ハブマシン）を経由するように制限する（Costales *et al.*, 1998）．これは，クライアントのマシンのsendmailを起動せず，すなわち無防備の窓口（sendmailの起動）を組織の内部に作らないことになる．また，受信メールを留めておくsendmailを起動しているマシン（スプールマシン）は必ずハブマシンを経由して送受信し，sendmailを持たないクライアントマシンではスプールマシンからpopperのプログラムでメールを吸い上げる方法になる．これだとメール管理者の主な監視はハブマシンのみで済む．管理する仕事は，/var/log/syslogのファイル内容（送受信や中継）を調べたり，lastコマンドで外部からのアクセスを調べたりなど，定期的な監視が大切である（下山ほか，1993）．また，1998年の秋にバージョン8.8.8に更新したのに，半年後にはさらなる更新をしなければならないほど，クラッカーとのいたちごっこを続け，遅れを取らないことが，安定したメールサーバの役目をはたすのである．

メールの配信先，保持日数，内部受け入れマシン名，スパム対策用ブラックリスト名など組織に応じた設定を決めるものにsendmail.cfのファイルがある（小山，1998）．この内容は難解な記号が並んでおり，メール管理の初心者泣かせであるが，sendmail.cf自動作成用のソフトCF-3.7 Wp12がある．これは判りやすい語句で記述してあるファイルsendmail.defを編集すれば自動的にsendmail.cfを作成してくれる．出来上がったsendmail.cfを試験的に起動し任意のメールアドレスを与えて動作の確認をする．例えば，自分宛てのメールは自分自身のマシンへ，その他の全てのアドレスはハブマシンを経由するようになっているか？などである．試験に合格すれば実際に運用する．できればだが，外部の組織にアカウントを持っているならば外部から送信してみると完璧である．sendmail.cfの設定を間違えるとエラーメールを上位メールサーバ管理者にたくさん出すから慎重に試験をしておく．

サーバは動いていて当然？

メールであれWWWであれネットワークを利用している一般ユーザにとっては，サーバは動いて当然と思っている．もちろんそのように思いこませるネットワーク管理者は優秀である．管理する立場になって始めて判ったことがある．電話や水道などは使用量に応じて負担しており，管理も専門家たちが運営をしている．一方，大学のネットワークは利用度に応じた個人的な負担もなく組織化された運営でもないのに，ユーザからは機能向上が常に望まれている．そして管理作業の重要さは理解されつつあるが，ネットワーク管理の多くはボランティアでなされている現状で，実体を伴った具体的な高い評価がされていないことである．

多くのネットワーク管理者は，自分自身がネットワークをもっと快適に利用できるように管理の仕事も止むを得ずしているのが実情である．そして，ついでにボランティアで組織のネットワーク管理もしているのである．ところが一般ユーザやクライアントマシンが増え，データ量も巨大になると管理作業が増大して本業ではない人たちにとっては苦痛になってくる．そしていったんトラブルが発生すると，動いて当然と思っている人たちから「管理がなっとらん」との苦情が寄せられる．図8のように，ネットワーク管理者は他の組織の管理者達と連絡を取りながらスムーズな運用を心掛けている．トラブルには，ハードエラーだけでなく，クラッカーの侵入や内部ユーザの不注意がある．また2000年対応の処理に多大な労力を注ぎ込んでもネットワークの機能は同じである．これが本業ならば対応の方法も対策も組織的にできるが，趣味の延長では責任を負うわけにはいかない．

近年，目に見えるハードウェアはもちろんだが，見えにくいソフトウェアにも資本や労力を投入するようになってきた．同じようにネットワーク管理も多大な資本や労力の必要性・重要性が望まれている．現状は，管理者の仕事は

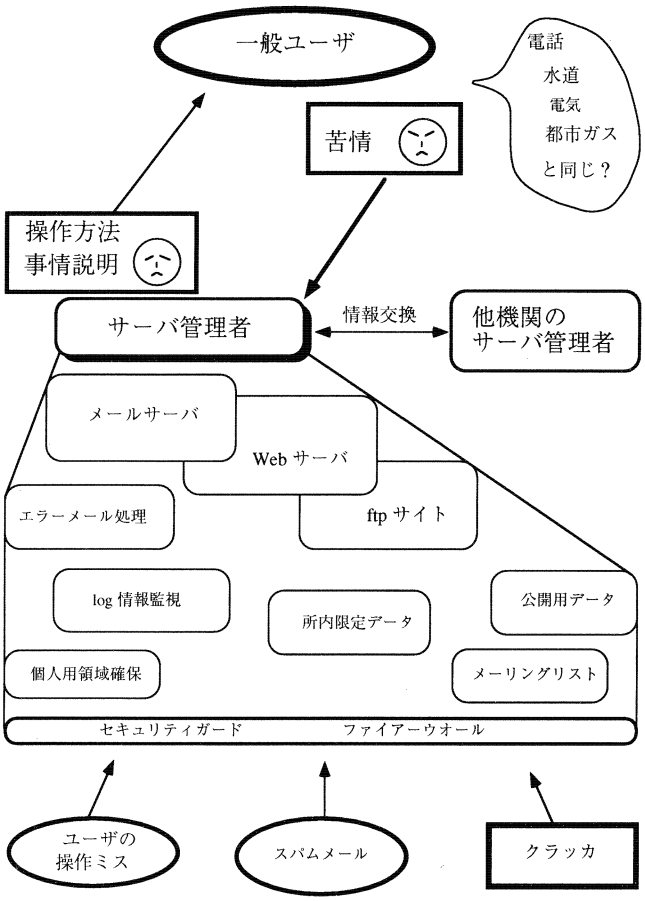

図 8.　サーバは動いていて当然か？

理解されつつあるが，専用の人材のために予算化されてはおらず，残念ながらコンピュータに堪能な人に頼っているのが実情である．論文を書いていくばくかの実績になる職種の人がネットワーク管理に振り回されていては浮かばれない．いきおい研究職ではなく技術職の人達に期待がかかるのである．もうひとつ悲しいかな，その技術は常に変化していることと，対応する技術者の平均年令が高いので，十分な対応ができないのも現実である．この現実を少しでも打破しようと奮起し挑戦したのがこの報告である．

サーバの影武者

ハードディスクは四六時中回転をしているので寿命が短い．ある日突然にサーバが動かなくなるのはハードディスクのクラッシュが原因と言っても過言ではない．クラッシュほどではない多少の不調であれば，シングルユーザーモードになってfsckコマンドで回復できるが，不調の回数が増えれば新品と交換せねばならない．冷却ファンの故障で機器全体の熱が上がり，他の部品の故障を誘因させることもあるが，やはり稼動部分の寿命が短い．従って，故障の際にはすぐに対応するのだが，1台のみでは回復までに数時間から数日の停止を覚悟しなければならない．技術室には古いEWSがあり，ハードディスクの交換でまだ使用に耐え得るので，サーバの代替機の役目をさせている．普段はサーバと別のホスト名で動かしており，適宜サーバのファイルをコピーしているのである．当然だが，ネットワーク内に同じホスト名とIPアドレスを持つ複数のマシンの存在は許されない．もし同じ名前のまま稼動するならばネットワークケーブルをはずして，オフラインでのコピー操作をしなければならない．WWWのファイルやメーリングリストなどは別のパソコンで編集作成してからサーバに登録しているから，バックアップは既にあることになる．メール利用者の登録やメーリングリストの追加や削除は時々更新するのでバックアップしやすい．刻々と変化するのは受信したメールスプールの中身である．サーバ稼動中のコピーは，いったんWWWやsendmailを止めてファイルの更新をなくする．コピー終了後にWWWやsendmailを再起動すれば，正常にサービスを再開する．

サーバの交代は，影武者だったマシンの名前をサーバの名前に変更しなければならない．稼動中のサーバを止めた（あるいはダウンしている）状態にして，次のファイルを変更する．/etc/hostsのIPアドレス，/etc/hostname.hme0のホスト名，/etc/nodenameのホスト名である（細川，1992）．変更したらリブートで影武者だったホストがサーバの名前で立ち上がりサービスを開始する．このように同じ機能を持つサーバをもう1台用意しておく方法はサーバの停止期間を非常に短くできる．

おわりに

失敗談を二つ述べる．影武者を構築中でpasswdをコピーする際にどう言うわけか/etc/passwdファイルをなくしてしまった．知らずにリブートすると，OSが立ち上がらない状態になってしまった．OSの始めからインストールか！と恐れたが，CD-ROMのOSで立ち上げておき/etc/passwdを編集して対応する方法が見つかり安堵した．もう一つは，ftpサーバのインストールの際に処理を間違えてしまい，スーパーユーザでもコマンドを受け付けない状態になってしまったことである．どうもOSそのものが記録されているハードディスクの/，/usr領域が壊れたようだ．これはディスククラッシュと同じ症状である．つまり復帰が不可能である．最後のソフトをインストールする段階だったので，今までのインストール作業時間が無駄に終わったのだ．結局，泣く泣く振り出しに戻りOSのインストール（城谷，1998）から始めねばならなかった．復帰の作業に延べ2日もかかった．

システム点検など障害以外のサーバ停止は，ユーザアクセスの少ない時間帯で行うのが好ましい．一般にはその時間帯はほとんど深夜になるし，ソフトウェアの入れ替えな

ど時間のかかるものは休日を利用することになる．ネットワーク管理を専門にするならば夜勤と休日出勤の職種にしたいものだ．日中常勤者には少し辛いお仕事である．いずれにしても管理者には，落胆に負けない気力と徹夜できる体力が必要であることも理解した．

ネットワーク管理はコンピュータの好きな学生が担当している場合が多い．地震予知研究センターからは自動で決めた地震情報をメールで流しているのだが，受け手のマシンが故障するとメール配達不能のエラーが発生する．相手に連絡すると担当の学生がいないなどの返事が返ってくる．責任のない学生に頼る組織が多いのである．業者に委託するところも増えてきているが休日までは対応できない状態である．

常に快適なネットワーク環境を目ざす場合は，組織内部にネットワーク管理に堪能でサービス精神旺盛な人材が必要である．ボランティアに頼るのではなく，ネットワーク管理の分野をもっと積極的に価値ある分野として認知されることを切に望むものである．

謝　辞：最後にハードウェアからソフトウェアまで指導していただいた大見士朗博士に感謝いたします．また，メール管理関係では院生の妻井康幸氏（1999年7月から静岡大学情報学部情報社会学科）に貴重な意見をいただきました．お二人の豊富な経験と指導がなければもっと時間がかかったであろうと思われます．記して感謝いたします．

文　献

Costales B. and E. Allman , 1998, Sendmail システム管理, 鈴木克彦訳, (株)オライリー・ジャパン, 185-198.

細川昭男, 1992, UNIX システム管理の方法, HBJ 出版局, 136-144.

井上亜潮, 1998, Solaris 2.6 サーバー構築, (株)秀和システム.

小山洋一, 1998, *UNIX MAGAZIN*, (株)アスキー, **13**, No. 12, 25-37.

國安和廣・秀和システム出版編集部, 1998, フリー UNIX で作るネットワークサーバ, (株)秀和システム, 3-20.

Laurie B. and P. Laurie, 1998, Apache ハンドブック, 三代川信義訳, (株)オライリー・ジャパン, 60-65.

村井　純・吉村　伸監修, 1996, インターネット参加の手引き, 共立出版(株), 7-10.

中村　眞, 1998, *UNIX MAGAZIN*, (株)アスキー, **13**, No. 12, 12-15.

下山智明・城谷洋司, 1993, SUN システム管理, (株)アスキー, 105-126, 219-236

城谷洋司, 1998, *UNIX MAGAZIN*, (株)アスキー, **13**, No. 2, 11-30.

Winsor J., 1998, Solaris システム入門, (株)インプレス, 329-331.