

多人数教育用計算機環境における システム管理の省力化の一方法

†齊藤明紀, †中西通雄, †安留誠吾, †重弘裕二, †西田 知博,
‡馬場健一, ††阿部広多, ††松浦敏雄
†大阪大学情報処理教育センター †高知工科大学
††大阪市立大学学術情報総合センター

概要 教育用の計算機システムは、一般に利用者数、および、管理すべき計算機の台数の両方が多いために、システム管理の負荷が大きい。さらに、利用者のうち初心者のおよむ割合が高いこと、利用者が必ずしも善意の利用者ばかりでないことなどの理由から、不注意や故意によってシステム運用に支障を与えるようなことも起こりうる。本稿では、このような多人数が利用する、多数の計算機をネットワークで接続したシステムの維持・管理を省力化するための方法として、多数の計算機の初期設定や設定変更等を集中管理で行なうパッチシステム、異常検出と再起動の自動化など、いくつかの方法を示し、その有効性を明らかにする。

Methods for Maintaining Campus-Wide Educational Network with Minimal System Administration

†Akinori Saitoh, †Michio Nakanishi, †Seigo Yasutome, †Yuuji Shigehiro,
†Tomohiro Nishida, †Ken'ichi Baba, ††Abe Kota, ††Toshio Matsura
†Education Center for Information Processing, Osaka University
†Kochi University of Technology ††Media Center, Osaka City University

Abstract As computer networks have come into wide use, distributed systems, in which many personal computers or workstations are connected via networks, become popular for campus wide education. However maintaining such systems imposes heavy workload to keep systems well-managed, such as initializing home directory, changing passwords, gathering and analyzing statistical information, monitoring network status,...and so on.

We have shown several methods for reducing such workload on system administration for campus wide educational systems.

1 はじめに

大学等における教育用計算機システムとして、多数のパソコンやワークステーションをネットワークによって接続した分散型のシステムが多く利用されるようになってきた。このような教育用計算機の環境を常に良い状態に保ってシステムを維持管理するには、一般に多くの手間がかかる。具体的な作業としては、利用者毎のホームディレクトリと初期ファイルの設定、パスワード変更に伴う作業、利用統計情報の収集と分析、ネットワークの運用状況の監視、システムの不具合の修正などがある。

大学等の教育用計算機システムでは、利用者数が多いこと、および、管理すべき計算機の台数が多いために、上記の作業負荷が大きくなる。さらに、これらの作業に加えて、環境の特殊性を考慮しなければならない。

まず、大学の場合は毎年、約1/4の学生が入れ替わるため、初心者のおよむ割合が常に多い。そのため、読み書き中のFDDを引き抜くなど単純操作誤りをおかすものの割合がシステム導入後年月を経ても減らない。また、計算機の不具合を見つけた、リハングアップさせたりしても管理者に報告しない事が多い。さらに、故意に(もしくは、不注意によって)、計算機をリセットしてしまったり、他組織に侵入を試みたり、偽造メールを出したりなどのイタズラをす

る可能性がある。これらの不正な利用を阻止し、大学として対外的な責任を果たしつつ、かつ、システムを常に授業ができる状態に保たなければならない。

本稿では、このような多人数が利用する、多数の計算機をネットワークで接続したシステムの維持・管理を省力化するためのいくつかの方法を示し、その有効性を明らかにする。

省力化すべき作業を分類すると以下ようになる。

- 導入、設置、初期設定
機種更新やOSのバージョンアップは2~5年に1度程度の頻度でしか起きないが、作業負荷は膨大である。
- 構成管理
ソフトウェアのバージョンアップや設定変更など。
- 利用者のアカウント作成および初期環境の作成
- 個々の計算機の障害対応
ハングアップ、ファイル溢れなど質的に困難な作業ではないが、量が問題である。
- 故障への対応
HDD(ハードディスク)の交換後の再設定など。
- 窓口業務
パスワード忘れ、利用申請などの対応の作業量はかなり大きい。
以降、それぞれの方法とそれによって軽減される

```

#!/bin/sh
#SINGLEUSER
# clear above line if this can do in MULTIUSER mode
VERSION=25
# by toyonaka!saitoh ttyp3 Apr 14 04:38 (tgray15)
# by Sat Apr 20 20:35:02 JST 1996
while read DIR file ;do
export DIR
(cd $DIR && tar xvfp -) < files.$VERSION/$file || exit 2
done <<EOF
/ develop.tar
/LocalAdmin/conf fstab.tar
/LocalAdmin hourly.tar
/tmp rc.tar
EOF
[ $? -eq 0 ] || exit 2
[ -f /tmp/rc ] || exit 2
mv /etc/rc /etc/rc.sav
mv /tmp/rc /etc/rc

case 'hostname' in
toyonaka*|ecipt[0-9]*)code=tc;; #toyonaka center
suita*|ecips*) code=sc;; #suita center
jinka*|igaku*|hoken*|sigaku*|yakugaku*|kougaku*)
code=sb;; #suita bunsan
*) code=tb;; #toyonaka bunsan
esac
cp /LocalAdmin/conf/fstab.$code /etc/fstab.new || exit 2
mv /etc/fstab /etc/fstab.sav || exit 2
mv /etc/fstab.new /etc/fstab || exit 2
ln -s /Net/toyonaka/home-toyonaka /home-toyonaka
ln -s /Net/suita/home-suita /home-suita
niload -d -r -t /mounts localhost/local < /dev/null
niload -t fstab localhost/local < /LocalAdmin/conf/fstab.neti.$code
chmod 751 /private/spool
#
sync; sync; sync
echo $VERSION > /LocalAdmin/conf/PATCHLEVEL
sync; sync; sync
fbshow -B -I "Configuration changed to $VERSION. Rebooting..." -z 47
reboot

```

図 1: パッチファイルの例

作業の種類について述べる。

2 パッチシステム

多くのホストを一元管理する場合、構成管理は重要な問題である。アプリケーションの追加や各種設定変更等は、台数に比例した作業量になりがちであり、多数の計算機を分散配置したシステムの維持管理作業量の多くを占める。

ここで紹介するパッチシステムは、以下の目的のために使用している。

- 多数の計算機のソフトウェアの設定を集中管理する
 - 設置後の各計算機毎の個別設定を行なう
 - 故障等で HDD を交換した時の再設定を行なう
- このようなパッチシステムを使用することによって得られる利点には以下のようなものがある
- 集中的な構成管理が容易
 - ローカルプリンタを持つホストなどの詳細な構成の決定が、発注時ではなく納入後に行なえる
 - きめ細かな設定が可能
 - クライアント機の故障交換作業がハードウェアエンジニアだけで可能。
 - 保守に携わるサービスエンジニアが、特権パスワード

ドを知る必要がない。

このパッチシステムの原型は筆者が大阪大学基礎工学部情報工学科の計算機演習室の集中的な構成管理のために開発したものである。

集中的な構成管理は、設定ファイルの自動コピー機能を用いることが多い。例えばサーバ/クライアントともに OS が UNIX であるばあいには、ファイルシステムの内容の同期させる rdist を用いる。rdist は、サーバに格納した設定ファイルの雛型と各被管理ホストの設定ファイル群を比べ、異なるファイルを転送する。

rdist は、ひとたび設定すると更新されたファイルを自動的に検出して転送するので運用が楽である。しかし、クライアントの構成が複数種類ある場合にはその種類毎に雛型を用意せねばならない。また、デーモン類などのプログラム・ファイルは上書き更新ができない (text file busy) という問題がある。

当センターでは各クライアント機には同一の部屋内のプリンタのみを登録など、きめ細かな設定を行ないたいという要求があった。そのため、rdist では多数の雛型を用意せねばならず、雛型の内容の管理の手間が心配された。

そこで、設定ファイルの内容ではなく、各種設定

を行なう手続き(シェルスクリプト)をサーバで提供し、各クライアントホストでそれを実行することで設定内容の更新を行なうことにした。これをパッチシステムと称している。

2.1 パッチによる構成管理

各クライアント機は、「パッチレベル」の値をローカルファイル(/LocalAdmin/conf/PATCHLEVEL)に持つ。設置当初はパッチレベル0である。パッチファイルおよび、「現在のパッチレベル」情報はサーバから NFS で提供される。ファイル名は setup.{パッチレベル} となっており、setup.n には、クライアントのパッチレベルを n-1 から n にする手続きが格納されている。

クライアントは現在の自身のレベルとサーバに登録されたレベルを比較し、違っていれば対応する setup.n を順次実行する。このとき、何らかのエラーが起きればその時点で以降のパッチファイルの実行をとりやめる。エラーを起こしたレベルのパッチは、次のブート時あるいは定時チェック時刻(5分)に再度行なわれる。エラーが起こって中断された setup.n は次回再度実行されるので、それを考慮した書き方をせねばならない。

図1はパッチファイルの例(setup.25)である。第2行の SINGLEUSER は、この作業がリブートの過程(利用者がログインしていない時)にしか行なえないことを表す。setup.25 の場合は、fstab の更新にともない最後にリブートが必要となるためである。

ファイルの更新は、tar による。これは、ファイルの最終更新日時を、更新を行なった日時ではなくオリジナルのファイルの最終更新日時と同じにするためである。

シェルスクリプトを用いるため、設定の記述能力は十分にある。例えば、case 文や if 文を使うことで、個別に異なる設定が可能になる。この例でも、設置した地区(豊中、吹田)とセンター教室か分散端末室かを case 分で判断して、それぞれ異なる fstab を適用している。

このような設定変更は、リモートシェルを用いてサーバからクライアントを操作しても行なえる。しかし、サーバでの作業時に停電や故障で停止している端末機は、次に正常稼働するまで待たねばならない。本システムでは、次の再起動でクライアントが自動的に設定をダウンロードするので、クライアントの状態を気にせず、サーバにパッチを登録できる。

2.2 パッチによる初期設定

計算機の設置調整はもっとも集中的に労力を必要とする工程である。我々は、初期設定にもパッチシステムを使用することで設置調整を効率的に行なった。また、工場での生産過程での動作確認作業にも配慮した。

教育用の演習室では、すべての端末計算機で同様の環境を提供することを目的とする。そこで、各計算機(NEXTSTEP 3.3J を搭載した AT 互換機)の設定はほぼ同じであるが、ハードウェアレベルは、

- MO/CDROM の有無

- ローカルプリンタの有無、
- セカンド Ethernet カードの有無(ゲートウェイモデル)

という差異がある。ソフトウェア的には、ホスト名、IP アドレス、使用できるプリンタ、経路情報、使用するファイルサーバなどを個別に変更せねばならない。

個別設定作業を多く(大阪大学の場合は500台)について繰り返すのは非常に手間がかかる。そこで、工場では以下のような設定の端末計算機を生産するように依頼した。

- 遠隔設定を受け入れやすい設定である。
- ハードウェアの違いに関わらず、ハードディスクの内容は全く同じ。
- 当初設定(生産した状態)ではネットワークやサーバ計算機に依存せずにブートし、単独で動作確認が可能。

各教室に端末計算機を設置した時の個別作業は、以下のようなものである。

- ネットワークケーブルを接続せずに単体で起動することを確認し、
- ネットワークに接続して再起動し、
- 日時(カレンダー)情報とホスト名とハードウェアモデル種別を入力する。

これは非常に簡単な作業なので、OS(NEXTSTEP)に精通する必要がない。そのため、クライアント計算機の設置とケーブリングを担当するハードウェア部門のエンジニアでも容易に行なうことができる。

ホスト名を受けとるのはあらかじめ搭載した初期設定プログラムで、自機 IP アドレスを算出し、経路情報を設定する。クライアント機は初期設定後は稼働状態のまま(電源を切らずに)置かれ、サーバからの遠隔設定作業が可能な状態となる。ログイン可能なのはシステム管理用ユーザのみであり、ファイルサーバもマウントはしていない。この段階では、特権ユーザ(root)のパスワードを設置作業を行なうエンジニアに通知してある。

以降は、パッチシステムを用いて順次個別設定を進めてゆけばよい。一般利用者のログインを許すレベルに設定段階が進むと、root のパスワードを運用段階のものに変更しているが、これもパッチシステムを用いて実現している。

ハードディスクの初期設定をすべて同じにしたため、工場製造時のクライアント計算機のモデル数は3種類(通常、ゲートウェイ、SCSIあり)のみとなった。特別なモデルは、各部屋あたり高々1台づつしかない。このため、工場での生産だけでなく出荷・輸送・搬入作業も効率化された。

2.3 パッチによる設定回復

ハードディスクの故障や復旧不可能なファイルの損傷で、端末機の OS のインストールからやり直さねばならないことがある程度起こるのは避けられない。通常の保守契約ではハードウェアの交換とせいぜいファームウェアレベルの再設定までしか行われぬ。ソフトウェアの細かい再セットアップは

大学側の作業となる。これは煩雑な作業であるだけでなく作業の発生が予測できないため、運用上の負担は大きい。

パッチシステムを前提とすると、初期インストールの OS が 1 種類で良いため、あらかじめ OS を搭載済みのハードディスクをあらかじめ用意しておいて交換するだけで修理を済ませることができる。その後は当初の設置時と同じ時刻とホスト名を入力してしばらく待てば、その時点で正しい状態にまで自動的に設定される。

2.4 運用結果

図 2 は、大阪大学におけるパッチシステムの利用回数の経時変化である。薄い色の部分は、setup スクリプトの作成ミスにより増えたパッチレベルである。

2 月 29 日検収、4 月 8 日開館というスケジュールであったため、初期設定を順次行なった 2~4 月の利用回数が増えている。1996 年度 4 月 9 日~6 月のパッチは、NFS のマウントパラメータなどパフォーマンスチューニングが主である。1997 年 4 月には、ネットワークの構成変更 (VLAN 導入) およびファイルサーバの構成変更を行なったため利用回数が増えている。その他の時期の変更は、ソフトウェアのバージョンアップや発見されたセキュリティホールへの対応などである。

意図した通り、パッチシステムにより、全端末を巡回しての作業はほとんどなくなった。また、全台数への設定変更が数時間で終了するため、不具合を発見してから対処が完了するまでの時間が短縮された。

HDD 交換時も、予定した通りセンター職員への負担は皆無となった。

3 IP アドレスの設定方式

多数の計算機の集中管理を行う場合、IP アドレス等のネットワーク設定は個別には持たせず、bootp, rarp, DHCP などで集中管理することがよく行われている。しかし、我々はローカルファイルに持たせる方式を取った。

ネットワークブートを行うにはブロードキャストドメイン内に必ず 1 台以上のコンフィグレーションサーバが存在するような構成にする必要がある。数台規模の分散端末室を多く置かねばならない構成では、それぞれにサーバを置くのは難しい。

また、据え置き型計算機では管理の都合上、IP アドレスの割り当ては固定的に行うことが望ましい。このため、サーバにはクライアントの MAC アドレスを登録せねばならない。

設置当初に端末機毎の MAC アドレスを調べてサーバに登録するのは、それぞれにホスト名を与えるよりも手間がかかる。また、Ethernet カードの交換によって MAC アドレスが変わったときに、サーバへの MAC アドレス変更作業等が必要になってしまう。

クライアント機の故障交換作業に連係したサーバでの作業が不要であるので、故障修理に関しては職員の作業負担はなくなる。

また、保守エンジニアが、クライアント、サーバ

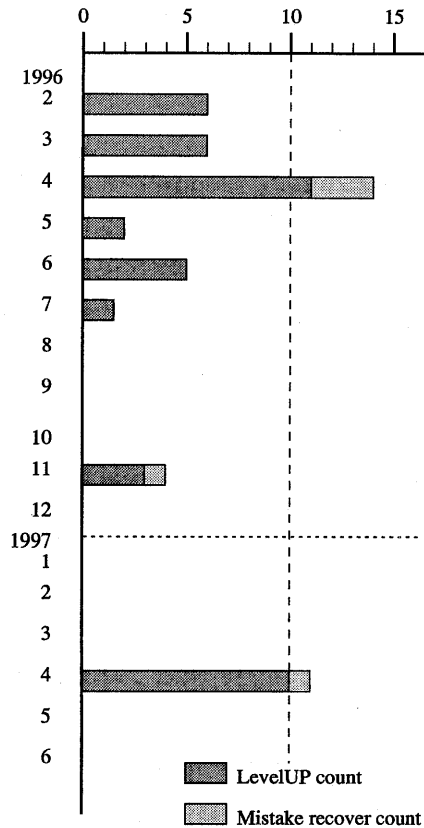


図 2: パッチシステム利用回数

ともに (運用段階での) 特権パスワードを知る必要がないので、大学側もベンダー側も互いに安心である。

4 起動方式

一般に UNIX 系のオペレーティングシステムでは、保守作業はシングルユーザモードで、設定ファイルをエディタで書き換えることで行なえる。しかし、マルチユーザモードで GUI 経由でない設定できない項目も最近では見られるようになった。

現在では、ネットワーク管理やサービスに NIS (Network Information Service) などを使用することが通例であるが、ネットワークやネットワークインターフェースカードの故障時にはシステムがハングアップしてしまいマルチユーザモードに起動しない。これでは、工場やネットワーク工事中の教室での設定作業や動作確認作業が困難である。また、ネットワーク系の設定変更作業も手間がかかって不便である。

保守作業を行なうためにネットワークに依存せず起動できる設定に変更し、作業後もとに戻すのは手間がかかりすぎる。

そこで、本センターのクライアント機はサーバにアクセスできない場合は自動的にスタンドアロン状態で起動するようにしてある。クライアント機は

ブート過程で、まずサーバのクライアントの設定情報の部分だけを NFS でマウントすることを試みる。これがタイムアウトや手動操作による割り込みで失敗した場合は、NIS 機能を使用しない。そのため、ハングアップせずにマルチユーザモードに移行する。

これにより、ネットワークが工事中であったり、ネットワークや中央サーバに障害が存在する状況でも管理者だけは普通にログインして作業を行なうことができる。

何らかの障害によりスタンドアロン状態で起動したあと放置されたクライアント機は、第5節で述べる自己診断により、サーバへのアクセスが可能になると自動的に再起動し、一般利用者へのサービスを再開する。

サーバ等管理用の機器の IP アドレス等はローカルファイルで持たせているので、スタンドアロン状態のクライアントに対するリモートメンテナンスが可能である。

このようにローカルディスクに置く情報を増やすことはトラブルに対する耐性を高める効果がある反面、各ホストに分散した設定情報の管理/更新のコストが増大する。これに関しては、2節に述べるパッチシステムで解決した。

5 定時自己診断

端末機の障害回復の手段としてよく行なわれる手段は再起動である。サーバ機では停止の与える影響が大きいため、再起動はできるだけ避けるが、クライアント機では障害の追求ために人手をかけることができないため、再起動を第一に試みることが多い。

ネットワークごしの遠隔操作での再起動は常に可能だとは限らない。また、常時監視を行なうには担当者が多数必要である。そこで、簡単な自己診断を各ホストが定期的(1時間に1回)に行なって、必要に応じて再始動するようにしている。

調査項目は以下の通りである。

- ファイルサーバにアクセスできるか?
- スタンドアロンモードで動作中か?
- パッチシステムでのエラーがおきているか?
- 前回の調査プロセスがハングアップしているか?
- ログイン記録(wtmp)やプリンタアカウントのサーバへの転送は成功したか?

パッチシステムがエラーとなるのは、setup スクリプトの異常終了あるいは再起動が必要なパッチの存在時である。

何らかの障害が発見された場合はさらに以下の点を調べる。

- ファイルサーバおよびネットワークが正常か?
rup コマンドでサーバが確認できない状況では、再起動よりもサーバ(への通信)の回復を待つ方が適切である。
 - 障害の深刻度は?
 - 障害が何回(何時間)連続して起こったか。
 - 現在ログインしている利用者はいるか。
- これらを総合して、以下の3種の対処のいずれか

を行なう。

- エラー回数のカウンタを進める
- 現在の利用者がログアウトした際にリブートするよう予約する
- 再起動を行なう (shutdown -r)

例えば、NIS が動作していないという重度の障害があり、サーバも動作している時は、1回目の検出で再起動する。ログインしている利用者がある状態で新たなパッチが発行されたという軽度の再起動要請がある場合には、3回めで(3時間後)、「ログアウト時再起動」を予約し、利用者がログアウトしない場合には5時間待って再起動する。

以上の作業は1時間に1回ずつ各ホストで行なわれる。多くのクライアントが同時にサーバにアクセスすることは望ましくないため、ホスト名や IP アドレスを用いて簡単なハッシュ計算の結果で定期チェックの時刻(のうち分)を決めている。

6 利用者環境の初期化

利用者登録に際しては、適当な初期設定のホームディレクトリも作成せねばならない。しかし、他人の個人環境を操作するのは面倒である。そこで、テンプレート用のユーザアカウントをもちいて設定を行ない、テンプレートユーザのホームをコピーすることで初期環境を書き込んでいる。

また我々は、利用者が初めてログインするときに、個人ディスク領域の初期化を行うという方式を取っている。利用者アカウントを作成する際には空のホームディレクトリだけを作成する。これには、以下のような利点がある。

まず、年度始めに数1000人分のホームを作成することは手間がかかる。特に日本語システムの場合、仮名漢字変換の個人辞書ファイルが大きいため、個人ディスク領域は大きくなりがちである。そこで、ホームの一括作成(約250KB, 18ディレクトリ, 21ファイル×3000人)の間、数時間~数十時間にわたってファイルサーバが過負荷になりレスポンスが悪化してしまう。大阪大学等のように学生全員にアカウントを作成する方式をとっている場合、利用しない学生や2年生になってはじめて利用しはじめる学生が20~30%存在する。本方式により、サーバの不可を分散化し、かつディスクスペースを節約することができる。

アプリケーションのバージョンアップ等で、ユーザ・テンプレートを変更することは学期途中で何回か起こり得る。しかし、すでに作成した利用者環境のある項目だけを変更することは手間がかかる。本方式を用いると、利用者が初めてログインした時点での最新の設定が用いられる。

利用者ファイルのうち電子メールのフォルダなど、初期保護モードが他人からの読み出しを禁止しているものは多く存在する。そのため、テンプレート・ユーザのホームを一般利用者の権限で読み出すことはできない。しかし、端末機側の特権ユーザの権限がファイルサーバにまで及ぶように設定することは保安上問題がある。そこで、1日に2回、テンプレ

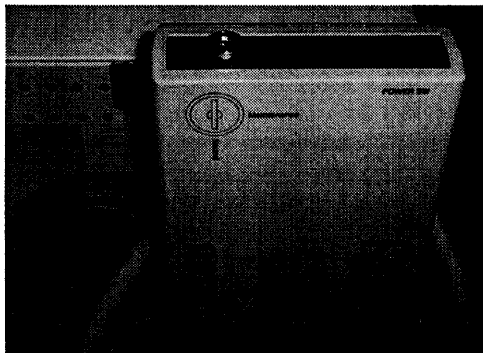


図 3: セキュリティボックス

トのホームを tar(gtar) でアーカイブして一つのファイルにまとめ、そのファイルの保護モードを一般読み出し許可としてファイルサーバに登録している。

7 セキュリティ

学生によるいたずらや誤操作によるハードウェア・ソフトウェアの損傷を避けるため、教育環境、特に自由自習に供するシステムではセキュリティ面での配慮が重要である。特に、IBM PC-AT 互換機等の PC を用いたシステムでは、ハードディスクの初期化ツールが誰にでも容易に入手できてしまう。また、不必要かつ安易なハードウェアリセットや電源断も安定運用のためには避けねばならない。

7.1 ハードウェアの保護

我々のシステムでは、フロッピーディスク、CD-ROM や MO からの起動を禁止する措置を講じている。これは、ファームウェア (BIOS) の機能や NEXTSTEP システムの改造などの手段で実現されている。また、BIOS セットアップやシングルユーザーシェルはパスワードで保護している。

大阪大学のシステムでは、電源スイッチを本体後面に移し、安易に押せないようにしている。また、リセットスイッチは配線を取り外すことで、無効化している。さらに、筐体に南京錠をつけ、ケーブルを絡ませることで、本体内部の部品やキーボード、マウス等の周辺機器の盗難を防止している。

大阪市立大学のシステムでは、図 3 のようにセキュリティBOX と呼ぶ追加ハードウェアを取り付けている。これは電源断の禁止 (投入はできる)、リセットスイッチの無効化、フロッピーブートの禁止を行うものである。セキュリティBOX はキースイッチをそなえており、メンテナンスモード側に切り替えれば上記制限は解除される。

7.2 ソフトウェアの保護

教育用環境では、コンソール利用者は一般利用者であり、特権操作はすべて禁止せねばならない。ところが、小型計算機用の OS は、コンソール利用者がシステム管理者であるという前提で作られていることが多い。たとえば、NEXTSTEP では、システムの時計やタイムゾーンの設定などを一般利用者でも

行うことができる。そこで、不必要な設定用アプリケーションの削除や setuid の解除を行った。また、ホームのファイルシステムは nosuid(no setuid) モードでマウントしてある。

利用者個人のファイル使用量の制限 (quota) が厳しい環境では、ローカルディスク上の一時利用領域 (/tmp, /usr/tmp) は、利用者の作業用領域として提供することが必要である。しかし、無制限にファイルを書き込むことによりファイルシステムが一杯になると、以降のシステムの正常な稼働が妨げられる。自由自習や講義を行う演習室では数 10 分から 1~2 時間で順次利用者が交代するので、ファイルシステムのあふれを放置すると以降の利用者の迷惑である。

そこで、ログイン時に前回と異なる利用者であれば /tmp を空にするようにしている。ログアウト時に消去しないのは誤操作や何らかのトラブルでログアウトしてしまった場合も同一端末に再ログインすれば作業が継続できるようにするためである。

また、各クライアント機には ident サービスを実装して、利用者認証を行なっている。これにより、他人を装ってサーバへアクセスすることを予防している。例えば From を偽造して記事をニュースサーバに投稿しても、本名を含むフィールドが自動的に追加されるようになっている。

8 まとめ

本稿では、多人数が利用する、多数の計算機をネットワークで接続した分散型システムの維持・管理を省力化するための方法を提案した。特にバッチシステムを採用したことによる管理の手間の削減が大きかった。大阪大学、大阪市立大学、高知工科大学の新システムの運用開始時期は、それぞれ、平成 8 年 4 月、平成 8 年 10 月、平成 9 年 4 月であった。バッチシステムのお蔭で大阪大学での設置調整も比較的簡単に行えたが、そこで経験を活用できたので、大阪市立大学、高知工科大学での設置調整作業はさらにスムーズであった。今回提案した管理の省力化手法の定量的な評価は今後の課題である。

謝辞

本研究の一部は文部省科学研究費補助金 基盤研究 No.(C)(2)08680376、および、電気通信普及財団の研究助成金の援助による。

参考文献

- [1] 齊藤明紀: “大阪大学情報処理教育センターのシステム構築 (1)~(6)”, UNIX MAGAZINE, 1996 年 5,8,12 月号, 1997 年 1,2,3 月号、アスキー。
- [2] 齊藤明紀, 原田章, 山井成良, 馬場健一, 安留誠吾, 松浦敏雄: “教育用計算機システムの運用および授業支援のためのシステムの現状と今後”, 情報処理学会分散システム運用技術研究グループ資料, DSM-9501031, 平成 7 年 1 月。
- [3] 齊藤明紀, 原田章, 重弘裕二, 馬場健一, 安留誠吾, 中西通雄, 山井成良, 松浦敏雄: “情報教育のための授業支援環境”, 情報処理学会第 52 回全国大会分散システム運用技術研究グループシンポジウム論文集, 平成 8 年 3 月。