仮想計算機を用いたWindows/Linuxを同時に利用できる教育用計算機システムとその管理コスト削減

安倍広多[†] 石橋勇人[†]
藤川和利[††] 松浦敏雄[†]

Linux と Windows を同時に利用できる教育用計算機システムの一実現法を，大阪市立大学での事例をもとに紹介する．このシステムでは，利用者端末の OS として Linux を採用し，その上で仮想計算機（VMware）を使用して Windows2000 を動作させる方式を採用した．本稿では，仮想計算機上のハードディスクやネットワーク構成，セキュリティ等，このようなシステムを実現するときの問題点と解決法を明らかにしている．仮想計算機の性能向上のため Linux にはハードディスクを内蔵するが，ハードディスクの管理コストを削減するために，(1) ハードディスクが空でもブートできるように Linux はネットワークブートする，(2) 利用者端末上の Linux と Windows のハードディスク内容は，ブート時にサーバ上の雛形を使用して自動的に更新する，等の工夫も行っている．この点についても詳述する．また，利用者端末の起動時間やハードディスク内容の更新時間等も実用的な範囲内であることを示す．

A Virtual Machine Solution of Windows/Linux Co-existing Computer System for Education

Kota Abe,[†] Hayato Ishibashi,[†] Kazutoshi Fujikawa[††] and Toshio Matsuura[†]

The purpose of this paper is, on the basis of our experience at Osaka City University, to introduce a method to implement an educational computer system whose users can simultaneously use both Linux and Windows operating system. In our system, Linux is used as the operating system of a user terminal and Windows2000 runs on a virtual machine (VMware) which runs on Linux. In this paper, we show the issues and solutions of implementing such system, including virtual machine's hardware configuration and security. In such system, as a hard disk is necessary for user terminals to enhance the performance of the virtual machine, we also demonstrate our method to reduce administration cost of the hard disk. For example, (1) Network boot procedure is used to boot Linux kernel so that it can run even if the disk is left uninitialized or out of order, (2) The filesystem for both Linux and Windows are automatically updated using a template on a server. Also, we experimentally confirmed that both booting a user terminal and updating its filesystems can be done within practical time.

1. はじめに

近年，大学の情報処理教育センターのような，様々な学部で共同利用する情報処理教育用のシステムでは，利用者からの要求の多様化にともない，利用者端末の OS として UNIX 系の OS と Windows 系の OS の両方を提供するケースが増えてきている[1)~3)]．

UNIX と Windows の両方を利用できるような利用者端末を構築するためには，大きく分けると，(1) UNIX 端末，Windows 端末を物理的に両方用意する，(2) 電源投入時にどちらをブートするか選択する（デュアルブート），(3) 1 つの端末上で，両方の OS を，同時に利用できるようにする（なんらかの方法で），という 3 つの方法があるが，教員や学生の多様な要求に最も幅広く対応できる方法は，両方の OS を同時に利用可能な方法 (3) であると考えられる．

従来から，方法 (3) を実現するいくつかの方法が利用されてきた．これらは利用者端末に何を使用するかによって，およそ以下のように分類できる．

(a) 端末として Windows を使用する．
(b) 端末として UNIX を使用する．

† 大阪市立大学学術情報総合センター
Media Center, Osaka City University
†† 奈良先端科学技術大学院大学情報科学センター
Information Technology Center, Nara Institute of Science and Technology

表 1 方法 (3) を実現する従来の方法
Table 1 Conventional methods to implement a UNIX/Windows co-existing system.

端末	Windows の利用	UNIX の利用
(a) Windows	ローカル Windows	リモート UNIX
(b) UNIX	Windows ターミナルサービス	ローカル UNIX
(c) シンクライアント	Windows ターミナルサービス	リモート UNIX

(c) 端末として Windows-based Terminal[4]，X 端末，Sun Microsystems 社の Sun Ray[5] のようないわゆるシンクライアント（thin client）を使用する．

(a) の場合，UNIX を利用するためには Windows 用の X サーバソフトウェアを利用し，リモートの UNIX サーバに接続する．(b) の場合，Windows を利用するためには UNIX 上で Windows ターミナルサービス☆1のクライアントを実現するソフトウェア（Citrix 社の MetaFrame と NCD 社の WinCenter for MetaFrame の組合せ等）を使用し，ネットワーク上の Windows サーバを利用する．(c) の場合，Windows には Windows ターミナルサービスを使用し，UNIX の利用には X サーバを利用してリモートの UNIX サーバを使用する（**表 1**）．

しかし，(a) のように Windows を利用者端末とした場合，遠隔からの一括管理が難しいという問題がある．また，内部が公開されていないため，細かいチューニングを行うことが難しく，問題が発生した場合の原因追求も困難である．(b)，(c) のように Windows ターミナルサービスを利用する場合，(1) 利用する Windows アプリケーションは Windows ターミナルサービスに対応している必要があるため，必ずしもすべての Windows アプリケーションが動作するわけではない（個々のアプリケーションごとに動作検証が必要），(2) Windows サーバとしてメモリや CPU に余裕を持った強力な計算機を多数（利用者端末 10〜20 台あたり 1 台程度[1]）用意する必要がある，といった問題がある．

一方，最近ではソフトウェア的に IBM-PC 互換機をエミュレートするソフトウェア（仮想計算機ソフトウェア）が使用できるようになってきた．このようなソフトウェアを用いると，UNIX 上で仮想計算機を構築し，その上で Windows を動作させることが可能となる．この方法では Windows ターミナルクライアントに起因する問題は存在せず，また強力な Windows サーバ計算機は必要としない．

今回，筆者らの所属する大阪市立大学学術情報総合センターの教育用計算機システム（以下本システムと呼ぶ）の機種更新にあたり，この方法を用いたシステムを実現した．本稿では，今回のシステム構築によって得られた，仮想計算機を用いて UNIX と Windows を提供する教育用計算機システムの管理運用上の問題点とその解決法を明らかにする．また，本方式では 4.2 節で述べるように性能上の理由で利用者端末内にハードディスクを内蔵する必要があるが，UNIX ベースの利用者端末のハードディスク管理コストを削減するための仕組みを構築したので，あわせて述べる．

以下，2 章でシステム設計の基本方針を述べた後，3 章で利用者端末のハードディスク管理方法，4 章で仮想計算機上の Windows の運用方法について述べる．5 章で導入したシステムの概要を説明し，6 章で評価を与える．

2. システム設計の基本方針

仮想計算機を実現するソフトウェアとしては，システム設計時に利用できた唯一の安定した仮想計算機ソフトウェアであった VMware 社の VMware Workstation 2.0 を用いることにした☆2．VMware はいわゆる IBM-PC 互換機をエミュレートする．VMware を動作させている OS を**ホスト OS**，VMware の仮想計算機上で動作する OS を**ゲスト OS** と呼ぶ．本システムでは，ホスト OS として VMware がサポートする唯一の UNIX 系 OS である Linux を採用し☆3，ゲスト OS になる Windows は，マルチユーザに対応したファイルシステム（NTFS）が利用できる Windows NT 系列の最新版であった Windows2000（Professional 版）とした．Windows の管理には Active Directory を使ってドメイン環境を構築する．Active Directory のサーバ（ドメインコントローラ）には，実計算機上の Windows2000 Server を使用する．

教育用の端末では，(1) ソフトウェアのインストールや更新，設定変更等が頻繁に行われる☆4，(2) 利用者数および端末数が多いにもかかわらず管理者の数は少ない，(3) 故障で利用できないと授業に支障がある，(4) 利用者が突然電源をオフにする等の理由でハード

☆1 Windows のアプリケーションはすべてサーバ側で実行し，クライアント側は画面表示と，キーボードやマウスからの入力処理を行う方式．

☆2 本稿執筆時点では VMware のほかに Connectix 社の VirtualPC という選択肢もあるが，システムの設計段階ではまだリリースされていなかった．

☆3 ディストリビューションとしては Redhat Linux 7.1 日本語版を採用した．

☆4 大阪市立大学の旧システムでは 5 年間で 51 回の設定変更を行っている．

ディスクの内容が壊れやすい，といった特徴がある．ハードディスクに起因するトラブルを避けるために，ハードディスクを内蔵せず，すべてのファイルアクセスをネットワーク経由で行う，いわゆるディスクレス方式を採用するという選択肢もあるが，VMware を利用するうえでハードディスクが内蔵されていないと十分な性能が得られない（4.2 節で述べる）という制約がある．このため，ハードディスク内蔵を前提とし，そのうえで以下の基本方針を立てた．

- 初期導入時や，故障時のハードディスク交換の管理コストを下げるため，端末のハードディスクが空の状態からでもブートし，管理者が介入することなくハードディスク上にシステムを自動的に構築できること．
- ハードディスク故障時にも授業への影響を最小限にするため，ハードディスクが故障している場合はディスクレス端末として利用できること．
- ソフトウェアのインストールや更新，設定変更のたびに管理者が端末の 1 台 1 台を手動で変更することは現実的ではないため，何らかの方法で端末へのファイル配布処理を自動化すること．

これらの実現法は次の 3 章で述べる．

3. 利用者端末のハードディスク管理

多数の利用者端末を管理するうえで，端末のハードディスク内容をどのように一括管理するかということは重要な問題である．この問題に対する解決策として，近年，分散配置された端末のハードディスクに，ネットワークを介して OS やソフトウェアを配布するための仕組みが実用化されている．主なものとして，IBM 社の LCCM（LAN Client Control Manager）[6]，Intel 社の LCM（LANDesk Configuration Manager）[7]，Symantec 社の Symantec Ghost [8] 等があり，なかには Linux のファイルシステムに対応しているものもある．しかし，これらの製品は端末のハードディスクに OS やソフトウェアを配布することが前提であり，また商用ソフトウェアであるため内部が公開されていない．今回のシステムではハードディスク故障時にも端末を利用したいという要求があるため，ディスクレス端末としての動作を考慮していないこれらの製品を利用することは困難であると判断し，新たに Linux が動作する利用者端末のハードディスク管理を省力化するための仕組みを構築した．本章ではこの仕組みについて述べる．

また，これから述べる Linux のブートシーケンスの概略を**図 1** に示す．

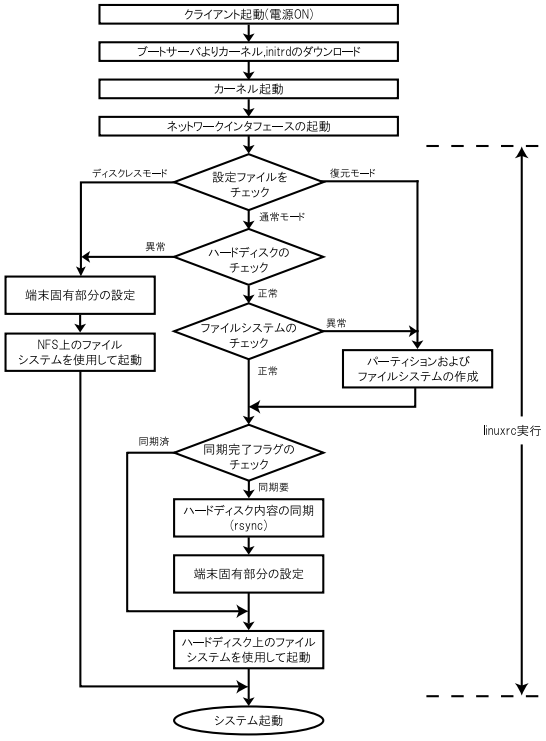

図 1 ブートシーケンスの概略
Fig. 1 Overview of the boot sequence.

3.1 Linux カーネルのロード方法

Linux は内蔵ハードディスクを利用するが，端末のブートデバイスをハードディスクにしたのでは，ハードディスクに何もインストールされていない，あるいはハードディスクが故障しているような状態では端末を起動できない．このため，Linux カーネルはハードディスクに頼らず，BIOS の PXE（Preboot eXecution Environment）[9] 機能を用いてネットワーク経由でロードする．Linux カーネルをサーバ上で一括管理できるため，カーネルの更新も容易に行うことができる．カーネルダウンロードの際にはマルチキャスト TFTP [10] を用いている．

PXE を使用するため，端末の IP アドレスは DHCP [11] で配布することになる．

3.2 ファイル配布作業の省力化

端末のハードディスクへファイルを配布する作業を省力化するために，文献 12) ではパッチシステムを使用している．この方法は，システムの変更履歴を残すことができる優れたシステムであるが，ハードディスクに OS そのものをインストールするためには用いることができない．本システムでは，次のような方法でファイルの配布を行う．

Linux には，initrd（initial ramdisk）と呼ばれる

RAM ディスクの機能がある．この RAM ディスクの内容はカーネルと一緒にロードされ，カーネル起動直後にルートファイルシステムとしてマウントされる．また，RAM ディスク中に linuxrc というファイルが存在する場合，それが自動的に実行される．この機能を使用すると，ハードディスク上に Linux ファイルシステムが存在しない場合でもネットワークブートした Linux 上で任意の処理を実行することができる．本システムでは，linuxrc はシェルスクリプトとし，以下の処理を行うようにした．

まず，linuxrc でハードディスクが利用可能かどうかをチェックする☆．利用できないと判断した場合は 3.3 節で述べる処理に移る．使用できると判断した場合は，続いてファイルシステムのチェック（fsck）を行う．fsck で自動的に回復できないエラーがある場合には，**復元モード**に入る．復元モードでは自動的にハードディスクにパーティションを作成し（fdisk コマンドに相当），ファイルシステムを構築（mkfs）する．なお，サーバ上の設定ファイルで指定することにより強制的に任意の端末を復元モードでブートさせることも可能である．

こうしてエラーのないファイルシステムがハードディスク上に存在することを確認した後，引き続き linuxrc を使ってハードディスク上のファイルシステムの内容を最新の状態に更新する（同期処理）．これは，端末のファイルシステム内容をそのままサーバ（ブートサーバと呼ぶ）上に置いておき，端末起動時にサーバ上のファイルと端末上のファイルを比較し，差分転送することで実現した．UNIX 上でこのようなディレクトリツリーの差分転送を行うためのプログラムとして，rdist および rsync[13] が一般的であるが，本システムでは以下の理由で rsync を採用した☆☆．

本システムの環境では，利用者端末のハードディスクが空の状態でも同期処理を行える必要があるため，利用者端末側で同期のために必要な仕掛けをなるべく単純なものにしたいという要求がある．rdist，rsync は良く似たプログラムではあるが，rdist はファイルの配布元（今回はブートサーバ）からしか実行できないのに対し，rsync はファイルの配布先（利用者端末）からも実行できるという違いがある．rdist を利用する場合は，利用者端末からブートサーバに対して何らかの形で rdist 開始要求を送る必要があり，さらに，ブートサーバの rdist が利用者端末側と通信するために，利用者端末にブートサーバから rsh でログインできるようにしなければならない．rsync を利用する場合はこのような手順は不要である．このため，今回は利用者端末側での導入コストの低い rsync を採用することにした．利用者端末の rsync は，ブートサーバで実行中の rsync サーバ（`rsync --daemon` で起動）と接続して動作する．rsync は低速な通信路でも効率良く同期できるように，転送元と転送先のファイルの差分を計算して必要な部分のみを転送する差分転送機能を備えているが，今回の環境では `--whole-file` オプションを指定することでこの機能をオフにして使用している（サーバの CPU に負担をかけ，結果的に転送速度を悪化させるため）．

以後，ブートサーバ上にある端末のファイルシステム内容を**端末イメージ**と呼ぶことにする．同期処理には一定の時間を要するため，利用者端末ごとに同期完了フラグを用意し，端末イメージが変更されたときのみ同期処理を行うようにした（利用者のファイルはファイルサーバ上に存在するため，利用者が端末上で書き込み可能なディレクトリは/tmp 等，一部しか存在しない．このため，Linux のファイルシステムが利用者によって書き換えられることを心配する必要はない．なお，/tmp 等はログアウト時に自動的に消去するようにしている）．

すべての利用者端末のハードディスク内容はほぼ同一であり，異なる個所はホスト名や教室ごとに異なるプリンタの設定等，ごく一部にすぎないため，端末イメージは全端末で共通とする．端末ごとに異なる個所を設定するため，同期処理完了後に linuxrc でホスト名やプリンタの設定ファイルを上書きするようにした．これにより，1 つの端末イメージを更新し，個々の端末をリブートするだけで，全端末のハードディスク内容を更新することができる☆☆☆．なお，同期処理はブートサーバにかかる負荷が大きいため，サーバを複数用意して負荷分散できるようにした．端末ごとにあらかじめ決められたサーバを使用するが，耐故障性を向上させるため，決められたサーバが ping に応答しない場合は自動的に次のサーバを選択するようにしている．

☆ 現在は sfdisk コマンドを用いて，ハードディスクのパーティションテーブルにアクセスできない，またはパーティションテーブルに異常があると判断した場合はハードディスク利用不能と判断している．この方法では，パーティションテーブルの内容が壊れただけでハードディスク利用不可と見なしてしまうため，今後改善する予定である．

☆☆ sup[14] という選択肢もあるが，利用者端末の Linux，およびブートサーバの Solaris での動作がサポートされていないため，候補から外した．

☆☆☆ 実際には，端末イメージは複数持つことが可能で，端末ごとにどの端末イメージを使用するかを設定できるようにしている．

3.3 端末のハードディスクが利用できない場合の動作

端末のハードディスクは故障しやすい部位であるが，授業で利用することを考えると，ハードディスク故障中でも端末が利用できることが望ましい．このため，端末のハードディスクが利用できない場合には，一時的にディスクレス構成で動作できるようにした．ディスクレス構成でのブートは，端末起動時にハードディスクが利用できないと判定した場合か（前ページ脚注参照），サーバ上の設定ファイルで指定された場合に行う．

ディスクレス構成で起動する場合，ファイルシステムとしてサーバ上の端末イメージをそのまま NFS マウントして利用する．ただし，NFS で共有できないファイル（教室ごとに異なるプリンタの設定ファイル等）に関しては次の方法で対処する．端末イメージ上で，共有できないファイルを格納するためのディレクトリ (`/ramdisk`) を作成し，共有できないファイルはそのディレクトリの下のファイルにシンボリックリンクを張る．ディスクレスで起動する場合は，`/ramdisk` に RAM ディスクをマウントし，動的に必要なファイルを生成する．`/tmp`，`/var` 等の共有できないディレクトリについても RAM ディスクを利用する．

3.4 端末イメージの更新

サーバ上の端末イメージの更新は以下のように行う．

(1) 新しい端末イメージを作成するため，管理者が利用者端末上でソフトウェアのインストールや設定変更等を行う．この端末を**マスター端末**と呼ぶ．

(2) rsync を用いてマスター端末のファイルシステムをサーバ上の端末イメージに転送（アップロード）する．ただし，転送中に他の端末がそのサーバと同期処理を行うと，完全でない端末イメージをダウンロードしてしまうことになるため，マスター端末からのアップロード中は端末の同期処理を行わないように排他制御を行っている（同様に，端末の同期処理中はマスター端末からアップロードできないようにしている）．

(3) 端末がブートする際に同期処理を行うようにするため，全端末の同期完了フラグをクリアする．

3.5 ブートシステムのセキュリティ

今回採用した Linux のブートシステムではネットワークブートを使用しているため，ネットワークブートに共通するセキュリティ上の懸念がある．たとえば，悪意を持った利用者は，偽の DHCP サーバや TFTP サーバをネットワークに接続することで端末を支配することが可能である．これを防ぐためには，たとえば BIS（Boot Integrity Services）[15] のような，ダウンロードしたブートローダを検証する仕組みを採用すればよい．

4. VMware 上の Windows2000 の運用

次に，VMware 上の Windows の運用について述べる．なお，ここでは VMware 上で Windows2000 を動作させることにより生じるテーマを主に扱い，実計算機上でも適用できる管理手法については触れない．

4.1 VMware について

2 章で述べたように，VMware は仮想的な IBM-PC 互換機を提供する．ここでは，システム設計上重要な仮想ハードディスクと仮想ネットワークインタフェースについて説明する．

VMware は仮想計算機上でゲスト OS が利用できる仮想ハードディスクを提供する．仮想ハードディスクの実体はホスト OS 上のファイルである（本稿では**仮想ハードディスクイメージ**と呼ぶ）．

VMware は，仮想計算機上で仮想ネットワークインタフェース（仮想 NIC）も提供する．仮想 NIC には Bridged と Host Only の 2 つのモードがある．Bridged の場合，仮想 NIC はホスト OS が接続された実ネットワークと直接つながっているように見える（VMware によってパケットが中継される）．Host Only の場合，仮想 NIC は，ホスト OS 内に存在する仮想的なネットワークに接続されている．この仮想的なネットワークは，ホスト OS からは vmnet と呼ばれるネットワークインタフェースを経由してアクセスできる．Host Only の場合，ゲスト OS はホスト OS としか直接通信できないため，実ネットワーク上の他の計算機と通信する場合はホスト OS がルーティングする必要がある．

4.2 仮想ハードディスクのモード

実計算機上で Windows を利用者端末として教育用システムを運用する場合，ハードディスクに対する書き込みをリセットし，容易に元の状態に復元できるような製品を導入して運用コストの低減を図る方法がある．Windows のセキュリティホールやウイルス，あるいは利用者の故意等の原因でハードディスク内容が破壊されても容易に復旧できるため，教育用計算機システムには好都合である．VMware の仮想ハードディスクにも，同様の機能を実現する nonpersistent mode があるため，これを利用することにした．このモードでは，書き込み内容は仮想ハードディスクイメージとは別のファイル（REDO ログと呼ばれる）に記録さ

れ，VMware 終了時に破棄されるようになっている．

仮想ハードディスクイメージや REDO ログを NFS サーバ上に置いた場合，十分な性能が得られないため☆，端末にはハードディスクを内蔵することにした．

仮想ハードディスクイメージは，Linux 上のファイルであるため，各端末へは 3.2 節で述べた仕組みによって配布する．

4.3 Windows の起動時間短縮

VMware は，ゲスト OS の実行を中断（サスペンド）し，次に VMware を起動したときにゲスト OS の実行を再開（リジューム）できるという機能（サスペンド機能）を備えている．ゲスト OS の実行状態はファイル（本稿では**サスペンドイメージ**と呼ぶ）に書き出される．

VMware 上の Windows2000 を普通に起動すると時間を要するため，サスペンド機能を利用して高速化することを考えた．Windows のログオンパネルの状態でサスペンドし，そのときのサスペンドイメージを端末に配布することで，利用者が VMware を起動すると，いきなりログオンパネルの状態から Windows を使用することができる．リジュームに要する時間は短いので，Windows を利用可能になるまでの時間を大幅に短縮することができる．サスペンドイメージはリジュームするたびに消去されるため，サスペンドイメージのオリジナルは保存しておき，実際に使用するサスペンドイメージはコピーしたものを用いる．利用者が Linux からログオフするたびに Windows2000 が使用されたかをチェックし，使用された場合はサスペンドイメージをコピーする．このコピーには 34 秒程度要するが，バックグラウンド処理で行っているため，ログオフ後にすぐに次の利用者がログインし，かつ Windows2000 を起動しない限り問題にならない．また，Windows2000 を起動しようとした時点でサスペンドイメージのコピーが終了していなかった場合には，コピー終了を待つようにしている（コピー完了フラグファイルの有無をチェックしている）．

4.4 仮想ネットワークの構成

サスペンドイメージからリジュームする場合，すべての端末上の Windows で，MAC アドレス，コンピュータ名，SID☆☆等，ハードウェア環境，ソフトウェア環境のすべてが，文字どおりまったく同一となってしまう．

MAC アドレスが同一のコンピュータが複数同一ネットワーク上に存在することを回避するため，VMware の仮想ネットワークは Host Only 構成とした．Windows からはネットワーク環境もまったく同一に見える必要があるため，仮想ネットワークには，すべての Linux ホストで同一のプライベートアドレスを与え，Windows 側の仮想 NIC，Linux 側のネットワークインタフェース（vmnet）ともに固定の IP アドレスを割り当てた．また，Windows から実ネットワークを利用するために，ホスト OS の Linux を NAT ルータとして設定した．なお，Windows では NAT 変換が入ると利用できなくなるようなソフトウェアは利用していない．

4.5 SID 問題

Active Directory 環境では，計算機を識別するために SID を使用する．本システムでは，VMware 上のすべての Windows2000 はコンピュータ名や SID が同一となるため，ドメインコントローラからは 1 台のクライアント計算機として認識される．Windows のドメイン環境では，同一の SID を持つクライアント計算機が複数あっても問題は生じないとされているが，以下のことに注意する必要がある．

クライアント計算機に付与された SID は定期的に（Windows2000 では 30 日）に乱数で変更され，ドメインコントローラに通知されるようになっている．このため，Windows2000 をインストールして 30 日が経過すると，実行中の Windows2000 の 1 つ（仮に x と呼ぶ）が SID を更新し，ドメインコントローラは新しい SID を受け取ることになる．この新しい SID は当然それまでの SID とは異なるが，x 以外の Windows はそれまでの SID を使用しているため，結果として x 以外の Windows は Active Directory の管理下から外れてしまう（また，nonpersistent mode を利用しているため，x 自身も仮想計算機の電源を切れば SID は元に戻り，結局 Active Directory の管理下から外れてしまう）．

この問題は，SID の変更頻度を 30 日から最大の 100 万日に増やすことで対処した．

4.6 端末固有の設定

各端末の Windows2000 の設定はほぼ同一であるが，Linux 同様，プリンタの設定などで，一部設置する教室ごとに異なるような設定がある．すべての Windows で同一の構成でリジュームするため，このような設定は利用者がログオンした後に，Windows2000 のログ

☆ 本システムの端末を使用し，仮想ハードディスクイメージや REDO ログを NFS サーバ（Sun Blade 1000）に配置した場合，Windows2000 の起動に 2 分 10 秒，ログオンしてデスクトップが表示されるまでに約 5 分を要した．端末のハードディスク上に配置した場合のデータは表 4 参照．

☆☆ セキュリティ識別子．Windows システムをネットワーク上で一意に識別するために使用される．

オンスクリプトを用いて行うようにした．

Windows2000 側は自力では自分がどの端末上で動作しているのかを判定できないため，仮想ネットワークを経由して Linux から ftp を使ってホスト名を取得し，ログオンスクリプト内でホスト名に応じた処理（たとえばプリンタの設定）を行うようにしている．

4.7 セキュリティ対策

VMware 上の仮想計算機に対しても，セキュリティ対策が必要である．

4.7.1 Windows 環境の保護

利用者が仮想ハードディスクイメージを Linux 側から読むことができると，セキュリティ上問題となる可能性がある．このため，VMware を実行するための特別なユーザ（vmuser）を作成し，VMware はこのユーザの権限でのみ実行できるようにした☆．Windows の仮想ハードディスクイメージには vmuser からしかアクセスできないように設定する．

また，VMware 上の仮想計算機は利用者が簡単にリセットさせることができるが，仮想計算機はリセットされると，VMware の初期設定ではフロッピーをブートデバイスとして起動するため，利用者が適当なブートフロッピーを挿入すれば仮想計算機に対するすべてのアクセス権が得られてしまう．このため，仮想計算機の BIOS 設定でブートデバイスは仮想ハードディスクに設定し，さらに BIOS 設定を変更できないように，BIOS パスワードを仕掛けている．

4.7.2 別の OS のインストール防止

利用者が VMware 上に勝手な OS をインストールされると，仮想ネットワーク上のトラフィックを盗み見される可能性があるため，セキュリティ上好ましくない．

上で述べたように VMware を利用者の権限で動作させないようにしたため，利用者が VMware 上で新たな OS をインストールするためには，Windows2000 を起動するために立ち上げる VMware を使う必要があるが，VMware は，ゲスト OS 実行中は設定変更ができないようになっている．このことを利用し，VMware を起動する際に，`-x` オプションと，使用する VMware のコンフィギュレーションファイルを指定することで，VMware が起動するとすぐにゲスト OS の実行が開始されるようにした．また，`-q` オプションも指定することで，ゲスト OS が終了すると VMware も終了するようにした．こうすることで，利用者が勝手な仮想ハードディスクイメージを実行したり，設定を変更したりできないようにしている．

☆ sudo コマンドを利用して VMware を vmuser の権限で実行させている．

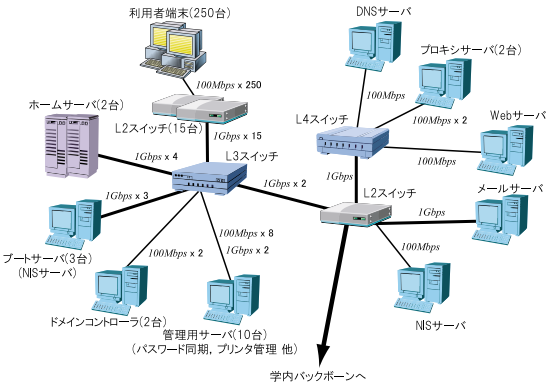

図 2 システム概要
Fig. 2 Overview of the system.

表 2 緒元
Table 2 Specifications.

種別	備考
利用者端末（250 台）	IBM-PC 互換機 CPU：PentiumIII 1 GHz メモリ：512 MB ネットワーク I/F：100 Base-TX
ブートサーバ（3 台）	Sun Blade 1000 (Solaris8) メモリ：1 GB ネットワーク I/F：1000 Base-SX
ホームサーバ（2 台）	Network Appliance NetApp 840 (350 GB/1 台)
その他 UNIX 系サーバ	メール，DNS，Web，プロキシ，NIS 等
その他 Windows 系サーバ	ドメインコントローラ，パスワード同期等

5. システムの構成

導入したシステムの概要を**図 2** に，主な諸元を**表 2** に示す．このシステムは 2001 年 10 月にサービスを開始した．

DHCP を使用する関係上，利用者端末とブートサーバはすべて同一のサブネット（サブネットマスク 22 ビット）上に配置している．

利用者端末の物理メモリ 512 MB のうち，VMware 実行時にはゲスト OS 側に 256 MB を割り当てた．Windows の仮想ハードディスクイメージの大きさは約 3 GB（C ドライブ約 2 GB，D ドライブ約 1 GB）である．

利用者のアカウントは，Linux，Windows をそれぞれ NIS と Active Directory で管理しているが，利用者からはどちらも同じユーザ名・パスワードで利用できるようにしている．これは利用者登録や削除を行

表 3 ブート時間
Table 3 Boot time.

起動種別	時間
ネットワークブート	1 分 40 秒
ハードディスクブート	1 分 10 秒

うアカウントの管理プログラムや，利用者によるパスワードの変更処理を，両方のアカウントに対して働くようにすることで実現した☆

6. 評　　価

6.1 Linux のブート時間

まず，ネットワークブートのオーバヘッドを測定するため，利用者端末の電源を投入してから利用可能になるまでの時間を，ハードディスクからブートした場合の時間と比較した．端末のハードディスクは同期済みで，同期処理（rsync）は必要ない場合の値である．結果を**表 3** に示す．ネットワークブートによるオーバヘッドは 30 秒程度であり，特に問題になる時間ではない．なお，本システムの実際の運用では，利用者端末は利用者が使用可能な時間帯にはつねに起動されているため，実際には利用者は Linux の起動を待つ必要はない．

6.2 ファイル配布時間

次に，利用者端末のハードディスクにファイルを配布する場合の端末起動時間を測定した．測定は，以下の場合について行った．

実験 1 Linux にインストールされている Web ブラウザ（Mozilla）をバージョン 0.9.3 から 1.0 に更新する（転送サイズ約 69 MB）．

実験 2 Windows2000 に新たに Mozilla（バージョン 1.0）をインストールする（転送サイズ約 3 GB）．

実験 3 復元モードで起動（転送サイズ約 6 GB）．

ブートサーバは 1 台とし，端末の電源を投入してから全端末で起動が完了するまでの時間を測定した．結果を**図 3** に示す．X 軸は同時に起動する端末台数，Y 軸は所要時間（分）である．

Linux の更新では，変更したファイルのみを転送するため，50 台に同時配布しても 10 分程度で終了している．Windows の更新の場合，わずかの変更でも仮想ハードディスクイメージすべてを転送することになるため時間を要する．グラフより，Windows を変更した場合でも，ブートサーバ 1 台で 1 時間に利用者端末 20 台にファイル配布できる．これを 1 時間ごとに

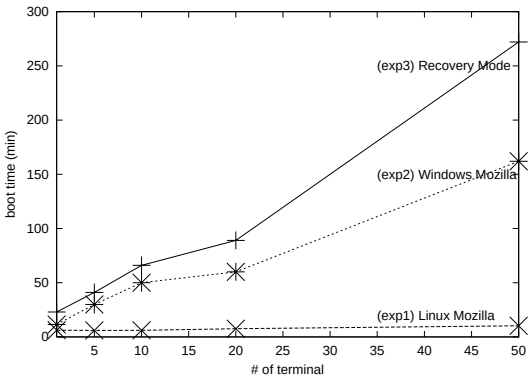

図 3 ファイル配布時の端末台数と起動時間の関係
Fig. 3 Relation between # of terminal and time to distribute files.

表 4 サスペンドイメージからの起動による効果
Table 4 Effects of booting from suspend images.

起動方法	起動時間	ログオン時間
通常の起動	1 分 36 秒	1 分 15 秒
サスペンド状態からの起動	3 秒	45 秒

繰り返す（端末を 20 台ごとのグループに分割し，グループごとに 1 時間ずらしながら起動させる）ことで，夜間（8 時間）で 160 台の端末にファイルを配布することが可能である．

復元モードは，システムの導入時，ファイルシステムに修復不能なエラーが発生した場合，およびハードディスクを交換した場合にのみ使用される．多数の端末を同時に復元するにはかなりの時間を要するが，このような場合はシステム導入時だけであり，システム導入時には時間の余裕があるため実用上問題は少ない．日常的には復元モードで同時起動する端末数はたかだか 1〜2 台程度であり，復元時間は実用の範囲内であると判断している．

6.3 Windows が利用可能になるまでの時間

4.3 節で述べた工夫により，利用者が Windows2000 を使用するために必要な時間がどの程度短縮されたかを，次の方法で測定した．

VMware 上で Windows2000 を普通に起動した場合と，サスペンド状態から起動した場合とで，起動時間（VMware 起動から Windows のログオンパネルが表示されるまでの時間）と，ログオンしてからデスクトップが表示されるまでの時間を測定した．結果を**表 4** に示す．

サスペンド状態から起動することによって 2 分程度早く Windows を使用できている．両者の間でログオン時間にも違いが見られる原因は，Windows の起動方法の違いが Linux 側のバッファキャッシュに影響を

☆ 利用者によるパスワード変更のためには，富士通北陸システムズ社製のスルー PASS2000 を利用している．

与えるためではないかと推測している．

7. おわりに

本稿では，Linux ベースの利用者端末の管理コスト削減の一手法と，その上で VMware を使って Windows 環境を提供する場合の管理運用の方法を述べた．大阪市立大学ではすでに 9 カ月間運用しているが，利用者からは最初にアプリケーションを起動する際に若干遅いという声があるものの，他に大きな苦情は寄せられていない．また，VMware を利用したことによる管理上の問題も生じていない．nonpersistent mode を使うことで Windows 環境を毎回リセットできるため，Windows の管理に要するコストは非常に低くなっている．運用を開始してから半年間で，端末のハードディスクの故障は 3 回発生したが，単に新しいものと交換するだけで済んでいる．

今後の課題としては，利用者端末へのファイル配布を，マルチキャストを利用する等の方法で高速化することがあげられる．

謝辞 システムの設計にあたって貴重なご助言をいただいた京都大学総合情報メディアセンターの丸山伸先生，東京大学情報基盤センターの安東孝二先生に感謝する．また本システムの実装に協力していただいた富士通株式会社に感謝する．

参考文献

1) 田中哲朗，安東孝二，吉岡　顕：複数 OS 環境におけるユーザ管理，情報処理学会研究報告，Vol.99, No.98, pp.49–54 (1999). (99-DSM-16).
2) 丸山　伸，北村俊明，藤井康雄，中村順一：5 年間使えるシステム作り，分散システム/インターネット運用技術シンポジウム 2002 論文集，情報処理学会シンポジウムシリーズ，Vol.2002, No.5, pp.60–74 (2002).
3) 京都産業大学情報処理教室. http://www.kyoto-su.ac.jp/information/
4) Microsoft: Microsoft Windows-based Terminals. http://www.microsoft.com/japan/windows/serverappliance/wbt/default.asp
5) Sun Microsystems: Sun Ray Integrated Solutions. http://www.sun.com/products-n-solutions/hardware/infoappliances.html
6) IBM: LCM (LANClient Control Manager) マニュアル．http://www-6.ibm.com/jp/pc/desktop/lccm/docs.html
7) Intel: LANDesk Configuration Manager. http://support.intel.com/support/landesk/configmgr/index.htm
8) Symantec: Symantec Ghost Corporate Edition 7.5. http://enterprisesecurity.symantec.com/content/ProductJump.cfm?Product=%3&PID=na&EID=0
9) Intel: Preboot Execution Environment (PXE) Specification Version 2.1. ftp://download.intel.com/labs/manage/wfm/download/pxespec.pdf
10) Emberson, A.T.: TFTP Multicast Option, RFC2090 (1997).
11) Droms, R.: Dynamic Host Configuration Protocol, RFC 2131 (1997).
12) 斎藤明紀：教育用大規模計算機システムにおける管理の省力化手法，情報処理学会論文誌，Vol.41, No.12, pp.3198–3207 (2000).
13) Tridgell, A.: Efficient Algorithms for Sorting and Synchronization, Ph.D. Thesis, The Australian National University (2000). http://samba.org/~tridge/phd_thesis.pdf
14) Shafer, S. and Thompson, M.: The SUP Software Upgrade Protocol. (1989). ftp://ftp.cs.cmu.edu/afs/cs/project/mach/public/doc/unpublished/sup.ps
15) Intel: Intel Boot Integrity Services (BIS). http://www.intel.com/labs/manage/wfm/tools/bis/
16) 安倍広多，石橋勇人，藤川和利，松浦敏雄：VMware を用いた Linux/Windows2000 が共存する教育用計算機システムの構築，平成 13 年度情報処理教育研究集会論文集，pp.343–346 (2001).

(平成 14 年 4 月 1 日受付)
(平成 14 年 9 月 5 日採録)

安倍　広多（正会員）

平成 4 年大阪大学基礎工学部情報工学科卒業．平成 6 年同大学大学院博士前期課程修了．同年 NTT 入社．平成 8 年大阪市立大学学術情報総合センター助手．平成 12 年同講師．博士（工学）．オペレーティングシステムの設計，実装に興味を持つ．電子情報通信学会会員．

石橋 勇人（正会員）

昭和 62 年京都大学大学院工学研究科修士課程情報工学専攻修了．平成元年同大学院博士後期課程情報工学専攻退学．同年京都大学大型計算機センター助手．平成 10 年より大阪市立大学学術情報総合センター講師．博士（情報学）．高速ネットワーク，ネットワーク管理システム等に関する研究に従事．人工知能学会，電子情報通信学会，IEEE，ACM 各会員．

藤川 和利（正会員）

平成 2 年大阪大学大学院基礎工学研究科博士前期課程修了．平成 3 年同後期課程中退．同年大阪大学基礎工学部助手．大阪大学大型計算機センター助手，奈良先端科学技術大学院大学情報科学研究科助手を経て，平成 10 年 4 月大阪市立大学学術情報総合センター講師，平成 14 年 4 月奈良先端科学技術大学院大学情報科学センター助教授．博士（工学）．広域分散環境におけるマルチメディアシステム技術に関する研究に従事．電子情報通信学会，IEEE，ACM 各会員．

松浦 敏雄（正会員）

昭和 50 年大阪大学基礎工学部情報工学科卒業．昭和 54 年同大学大学院基礎工学研究科（情報工学専攻）後期博士課程退学後，同大学基礎工学部助手．平成 4 年同大学情報処理教育センター助教授，平成 7 年大阪市立大学教授．工学博士．ユーザインタフェース，マルチメディア，情報教育等に興味を持つ．ACM，IEEE，電子情報通信学会各会員．