# ディスクレス環境の教育用計算機システムに適したLinuxシステムの実装

桝田 秀夫†, 齊藤 明紀‡.
h-masuda@ime.cmc.osaka-u.ac.jp, saitoh@kankyo-u.ac.jp
† 大阪大学サイバーメディアセンター情報メディア教育研究部門
‡ 鳥取環境大学情報システム学科

**概要** 大学の情報基盤センターや総合情報処理センターなどの教育用計算機システムでは，可用性や安定性，セキュリティを保ちつつ，TCO(Total Cost of Ownership) を削減して運用を行わなければならない．本稿では，TCO の削減を目指し，近年注目されている，Diskless 構成をとるLinux をベースにしたシステムの設計と実装について述べる．端末として通常の Intel PC を利用した上で，オープンソースである Linux をベースとし，Diskless 構成の構築を，(1) 端末のローカルディスクは利用しない．(2) 基本的に単一イメージを利用して複数の端末を稼働できるようにする．(3) 既存のディストリビューションをできるだけ改変しない．(4) アプリケーションの更新を容易にする．(5) 単一の OS イメージで，できるだけ多くのハードウェア構成に対応できるようにする．(6) サーバは Linux に限定しない．(7) できるだけオープンソースで構成する．の点に着目して実施した．この方針に従い，Vine Linux 3.0 をベースに実装した結果，1 つのファイルを改変した上で，いくつかの設定をすることで，Diskless 構成をとることに成功した．

**キーワード** Linux, ディスクレス, 単一イメージ, 教育用計算機システム, Vine Linux

# Implementation of Linux system for educational computer system using diskless technology

Hideo Masuda[1] , Akinori Saitoh[2].
h-masuda@ime.cmc.osaka-u.ac.jp, saitoh@kankyo-u.ac.jp
[1] Infomedia Education Division, Cybermedia Center, Osaka University
[2] Department of Information System, Tottori University of Environmental Studies

**Abstract** To reduce the total cost of ownership of educational computer system is very important. To achieve it, we implement the Linux system for educational computer system using diskless technology. We assume that terminals are ordinary Intel PCs and seven points of view: (1) no local HDD (2) single OS image shared by all terminals (3) little changes for normal Linux distributions (4) easy updating the packages (5) various hardware of terminals (6) ordinary UNIX server (7) Open source Using Vine Linux 3.0, we establish the diskless environment modifying only one file and some settings.

**keywords** Linux, Diskless, Single image, Educational Computer System, Vine Linux

# 1. はじめに

大学の情報基盤センターや総合情報処理センターなど，大量の端末設備を有し，全学の学生が利用者であり，講義や演習を行う教室も提供するような計算機システムでは，可用性や安定性，セキュリティを保ちつつ，TCO(Total Cost of Ownership)を削減して運用を行わなければならない [1].

本稿では，TCO の削減を目指し，近年注目されている，Diskless 構成をとる Linux をベースにしたシステムの設計と実装について述べる.

Diskless 構成の特徴としては，

- 故障し易いハードディスクを利用せずに稼働するので，故障率が低い.
- ハードディスク交換後，再インストールの手間がない.
- サーバ側で OS などの更新をすれば良いため，更新時に端末が稼働している必要がない.

など，TCO 削減に寄与するとして期待が高い.

Diskless 構成をとる教育用計算機システムの例として，文献 [2] では，Mintwave 社の VID[5] を使った Windows と Linux の dual boot 環境を使用している例や，また文献 [3] では，Apple 社の Netboot[6] を使った MacOS X の環境を使用している例があるが，いずれも商用 OS の機構である．そのため，サーバの機種が自由に選べない，ライセンス費が発生するなど，コストメリットが薄くなってしまう．また，カスタマイズの自由度も低い．また，文献 [4] では，Linux カーネルをネットワーク経由で読み込んで動作するシステムを提案しており，HDD の障害時には一時的に Diskless で稼働できるようにすることに触れている．しかし，通常の利用環境としては HDD ありを前提としており，単一ハードウェア構成でのシステムである.

本稿では，端末として通常の Intel PC を利用した上で，オープンソースである Linux を元に，Diskless 構成の構築を以下の点に着目して行う.

1. 端末のローカルディスクは利用しない.
2. 基本的に単一イメージを利用して複数の端末を稼働できるようにする.
3. 既存のディストリビューションをできるだけ改変しない.
4. アプリケーションの更新を容易にする.
5. 単一の OS イメージで，できるだけ多くのハードウェア構成に対応できるようにする.
6. サーバは Linux に限定しない.
7. できるだけオープンソースで構成する.

以降，2. 節で本システムの設計について述べ，3. 節で実装について述べる．また，4. 節で実装したシステムについての評価について述べ，5. 節で今後の課題に触れる.

# 2. 設計

## 2.1 起動部分

まず，端末のハードウェアが，ローカルディスクの助けを借りることなく OS を起動する能力を持つ必要がある．本稿では，端末は PXE(Pre Execution Enviroment)[7] 機能を持つパソコンであるとする.

また PXE は，サーバとして，特殊な応答を返す DHCP サーバとブートローダや OS のカーネルイメージを提供する tftp サーバが，ネットワークで接続されている必要がある．DHCP サーバとしては，ISC の dhcpd[8] の実装を利用すれば設定可能であり，tftp サーバは多くの UNIX 系 OS で標準装備されているため，Linux 以外のサーバでも問題なくサービスできる．さらに，ブートセレクタ機能を提供する pxe サーバを稼働させれば，複数のカーネルを利用者に選択させることも可能になる．この pxe サーバも，UNIX 系 OS でコンパイル可能なコードが公開されている [9].

## 2.2 OS 稼働部分

Linux が Diskless 構成で稼働する構成として，

1. root パーティションを RAM disk 上にする
2. root パーティションを NFS 上にする

という二種類が考えられる．本稿では，教育用としてオフィススイートなどの比較的大きなアプリケーションの実行を考慮し，NFS のタイプとして実装する．この場合，稼働に際して OS イメージを提供する NFS サーバが必要になるが，ほとんどの UNIX 系 OS で NFS サーバ機能が提供されている為，問題はない[1].

NFS の場合，全てのアプリケーションがネットワークを経由してアクセスされるため，ネットワークに障害が発生すると，ほとんどすべてのアプリケーションが停止してしまい，端末がハングアップ状態になってしまう．この状態を回避する為に，software watchdog 機能を利用する．Linux カー

[1] どの UNIX 系 OS を選択するかで，性能上の問題が発生する可能性は残されている.

ネルには，Software Watchdog 機能が備わっており，特定のデバイス (/dev/watchdog) へのアクセスを一定時間 (標準では 60 秒) 実施しなかった場合，カーネルが reboot する．この機能を利用し，ネットワーク経由で読み込んだファイルを定期的にこのデバイスに書き込む専用プロセスを稼働させることで，ネットワーク異常時に自動的に reboot するといった仕組みが実現できる．

### 2.3 単一イメージ化部分

UNIX 系 OS では，/etc 以下にホスト毎の設定やシステムに対応する設定が配置され，/var 以下にホスト毎の状態やログといった情報が配置される為，この 2 つのディレクトリツリーは，ホスト毎に別々に準備する必要があるとされる．しかし，/etc 以下でも，例えば /etc/resolv.conf や /etc/nsswitch.conf といったシステム内では共通になる設定ファイルがあったり，また，ホスト毎の設定でも，例えば /etc/modules.conf といった同一ハードウェア構成なら同一となる設定も存在する．/var 以下でも，例えば /var/lib/rpm/ 以下にある導入されているアプリケーション情報といった，システム内で同一と考えられる情報が多く含まれている．そのため，単純に /etc, /var 以下をホスト毎に準備すると，無駄にコピーが発生したり，同一に保つ為に全ホスト分の更新を実施する必要が生じてしまう．このような部分が多くなると，更新にかかる時間が多くなってしまうため，教育用計算機システムのように，多くの台数を管理する必要がある場合に手間がかかってしまう．つまり，単一イメージにできない部分は，/etc の一部，/var の一部と，/dev, /tmp となる．また，それらを除いた部分は，通常の端末からは改変されるべきではないため，read-only で NFS export しておくべきである．

本稿では，各サブディレクトリツリーを以下の方針で構成する．

- /etc のうち，ホスト毎に異なるファイルは，RAM ディスクを作成し，起動時に自動生成するものへのシンボリックリンクとする[2]．
- /var は，ホスト毎に異なるディレクトリを NFS mount し，システム内で同一のファイルは，/.var 以下などに置いた上で，シンボリックリンクとする[3]．
- /tmp, /dev は，起動時に RAM ディスクを作成し，必要なファイルやディレクトリを自動生成する．

### 2.4 更新機能部分

Linux では，RPM や dpkg などのパッケージ管理システムをベースに，アプリケーションの追加・削除・更新などを行なうようになっている．単一のイメージによる構成をとる場合，いつ，誰が，どのようにパッケージの更新を実施するかが問題となる．通常，更新を実施しているタイミングでは，更新対象となるアプリケーションが稼働していると，実行ファイルや共有ライブラリが置き換わり問題になる場合が多いので，稼働中に置き換えることは難しい．

そこで，図 1 のように，パッケージ管理システムによる更新を実施する端末を一台のみとし，その端末からのみ，オリジナルのディスクイメージを更新する[4]．その更新した内容に対して，スナップショットを作成し，新たな OS イメージとして，NFS サーバ上で公開する．スナップショットを作成した時刻以降に起動する端末に対して，新しい OS イメージを NFS でマウントして起動するように指示し，古い OS イメージで起動している端末に対しては，利用者が居なくなったタイミングで再起動するようにすれば，稼働している端末が参照している OS イメージが更新中の状態に陥ることはない．また，NFS mount client リストを監視し，古い OS イメージを参照している端末が無くなった時点で，その OS イメージは削除可能とする．

## 3. 実装

2. の設計に従い，Vine Linux 3.0[10] を元に実装した．Vine Linux は，

- コンパクトで安定指向
- 日本語の環境に強い

といった特徴を持っており，教育用計算機システムで利用する OS に適する．以降，実装上の注意点・工夫点について述べる．

[2] NetBSD などに実装されている union fs が使えれば，小さな RAM ディスクを union mount で上からかぶせることにより，シンボリックリンクにしておく必要も無くなる．

[3] 再起動されても残しておくべきログ情報などを，syslog を使ってログサーバに直接送付し，それ以外は RAM ディスクとしておくことも考えられるが，/var/tmp といった相当量を必要とするディレクトリが含まれるため，NFS サーバ上で提供するものとする．

[4] この端末は固定的に決める必要はなく，同時に一台のみが実施すれば，異なる端末からの更新であっても構わない．

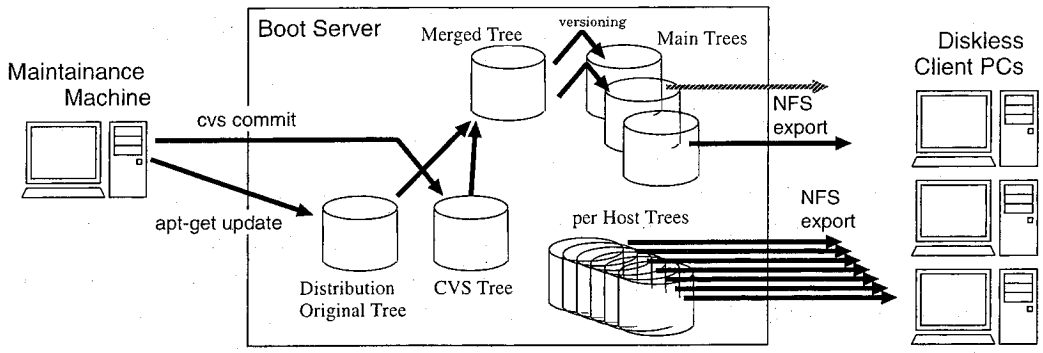


図 1: 更新システムの構成図

### 3.1 カーネル

NFS を root パーティションとして起動するため，カーネル設定に対して，

- CONFIG_ROOT_NFS=y, CONFIG_IP_PNP=y, CONFIG_IP_PNP_DHCP=y を追加
- 必要なイーサネットデバイスを，m(モジュール) から y(組み込み) に変更

を実施し，さらに，SOFTWARE WATCHDOG 機能を使うため，

- CONFIG_WATCHDOG=y, CONFIG_SOFT_WATCHDOG=y を追加

を実施した上で，カーネルを再構築した．カーネルのネットワーク版ブートローダには，Vine Linux が採用している lilo/syslinux のネットワーク版ともいえる pxelinux を利用した[5]．

### 3.2 ディストリビューションの改変

起動時に /tmp や/dev 以下を生成したり，/etc と/var を構成したりする必要がある．Linux では，カーネルの起動直後には/sbin/init が呼び出され，init は，/etc/inittab の内容に従って動作する．Vine Linux や Vine Linux がベースとする Red Hat Linux や Fedora Project では，システム起動時には，まず最初に/etc/rc.d/init.d/rc.sysinit を起動スクリプトとして実行するように記述されている．調査の結果，この部分で各種ファイルシステムの mount が行われているので，この起動スクリプトを改変することとした．

/dev については，Linux の kernel には，devfs と呼ばれるダイナミックにデバイスファイルを生成する仕組みが存在するが，多くの Linux ディストリビューションでは標準で使われておらず，またアプリケーションによってはうまく連携しない場合もある．そこで，起動直後に 1Mbyte の RAM disk 上に生成して，必要なデバイスファイルを作成した上で，mount -bind という機能を利用して /dev として利用できるようにした．この機能は，NetBSD で/sbin/init 読み込み直後に/dev 以下が存在しない時に memory filesystem を自動的に作成して MAKEDEV を実行する機能を参考にした．さらに，rc.sysinit スクリプト中の devfs を起動させようとする部分に hook を埋め込み，この機能を別スクリプト (/etc/rc.MAKEDEV) とすることで，選択的に利用できるようにした．

また，/etc，/var を必要な形に加工して mount する為のスクリプト (/etc/rc.diskless) を作成し，起動スクリプト中に hook を埋め込むことで，この機能も選択的に利用できるようにした．

つまり，この 2 つのスクリプトを呼び出さなければ，元々のディストリビューションとなんら変わらない状態になるように工夫することで，ベースとなったディストリビューションの更新をトラッキングする手間を極限まで減少させている[6]．

#### /etc 以下で共有できないもの

- fstab: リムーバブルメディアの有無など
- modules.conf: ハードウェア (サウンドカードなど) に依存
- murasaki/murasaki.preload: ハードウェア (USB,IEEE1394 など) に依存

[5] pxegrub を利用してもよい

[6] 起動スクリプト中には，root パーティションが NFS であることを考慮していない部分がありパッチを適用する必要があったが，これは，元々のディストリビューションのある意味でバグともいえる為，フィードバックを検討中である．

- X11/xorg.conf: ハードウェア (グラフィックカードなど) に依存
- mtab: マウント状況に依存 (→/proc/mounts への symbolic link とする)
- sysconfig/: ホストやシステム毎に異なる可能性のある設定一般
- printcap: 利用できるプリンタ (設置場所など) に依存
- cups/printers.conf: 利用できるプリンタ (設置場所など) に依存

/var 以下で共有できるもの

- cache/man/whatis: マニュアルページの whatis データベース. 導入されているパッケージのマニュアルページにしか依存しない.
- lib/rpm/: 導入されているパッケージデータベース.

### 3.3 端末ハードウェアに対する自動構成

Vine Linux では, kudzu と呼ばれるハードウェア自動構成支援ツールが採用されており, 起動時にハードウェア構成が変更されていれば, 自動的に対応する設定を/etc 以下に施すようになっている. しかし, 教育用計算機システムの場合, 端末の種類は事前に分かっていると考えられるため, それなりに時間の掛かる自動構成を毎回実施する必要はない. そこで, /etc/rc.diskless 内で, カーネルの出力を元にハードウェアの種別を判定し, それに応じた設定を /etc 以下に施すようにした. その上で, 未知のハードウェア種別であった場合は, 通常通り kudzu を起動するようにすれば, 多くのパソコンに対応できると考えられる.

また, /var 以下はホスト毎に別々に持っているが, OS イメージの更新に応じて, ディレクトリ構造が変更されたりする可能性がある. そこで, OS イメージのスナップショット作成時に, /var 以下の変更状況を抽出しておき, 各端末は起動時の /etc/rc.diskless 内で, その変更状況を適用するようにした.

### 3.4 OS イメージのスナップショット

図 1 のように, OS イメージはスナップショットを作成することにより, 稼働中の端末には元の OS イメージを提供しつつ, スナップショット作成後にブートする端末には, 新しい OS イメージを提供することが可能となる. このとき, 単純にスナップショットを作成すると, 更新部分が少ない場合に無駄が多い. そこで, スナップショット作成時に, 更新されていないファイルに対しては元のファイルに対してハードリンクを作成することで, 無駄なコピーを発生させず, ディスク上の領域も無駄使いしないようにする. この仕組みは, pdumpfs[11] をベースにしている.

## 4. 評価

### 4.1 起動時間

本稿で実装したシステムの起動にかかる時間を測定した. 使用した評価環境は, 表 1 の通りである.

| | |
|---|---|
| サーバ | SGI Origin300<br>R10000 600MHz x4, 4GB,<br>500GB(FC,RAID5),<br>1000baseSX |
| クライアント 1 | IBM Intellistation<br>PentiumIII 500MHz, 128MB,<br>100baseTX(Intel PRO/100) |
| クライアント 2 | DELL Optiplex GX260<br>Pentium4 2.6GHz, 512MB,<br>1000baseT(Intel PRO/1000) |
| ネットワーク | Summit1i<br>Summit24(学内 VLAN) |

表 1: 評価環境

起動時間として, 電源投入後の POST が終了した後, PXE の DHCP リクエストを出していることを示すメッセージが表示されてから測定を開始し, X が起動した上でログインパネルが出るまでとした.

クライアント 1 は, 現在の大阪大学サイバーメディアセンターで利用されている端末であるが, 約 **1 分 50 秒**で起動した. このうち, 約 **25 秒**は /dev の構成に掛かっていた. OS のバージョンが異なる (Vine Linux 2.6r4) ので単純には比較できないが, 現在の稼働状態であるローカルの HDD から起動した場合は, 約 **1 分 15 秒**で起動しているため, /dev の構成にかかる時間を除けば, ほぼ遜色のない時間で起動している.

クライアント 2 は, 比較的最近のスペックであるが, こちらは, 約 **1 分 30 秒**で起動した. また, クライアント 2 で, ローカル HDD に同一構成の

OSイメージを入れた上で起動すると，**約1分10秒**で起動した．

クライアント1と2の差は，/devの構成に掛かる時間の差と考えられる．そのため，ある程度CPUスペックの劣るパソコンに対して，/devをより高速に生成する仕組みを検討する必要がある．クライアント2は，GbEで接続されているが，起動時間に関してはあまり大きな速度の向上は見られなかった．これは，起動時は，デバイスプローブやネットワークサーバとのやりとりなど，CPUやファイルの転送を伴わない処理が多いためではないかと推測される．

### 4.2 改造量

3.節で述べた通り，ディストリビューションに対して加えた改変は，基本的に/etc/rc.d/init.d/rc.sysinitだけであり，変更部分は極めて少なくできた．

改造点を考慮すれば，別のディストリビューションへの実装もそう難しくないと考えられる．

しかし，少なくともFedoraのような有名なLinuxディストリビューションであっても，作りがアドホックな部分が多々みられるため，動作に問題が発生する可能性は否定できない．本稿で実装時に分かっている点は，以下の通りである．

- ルートディレクトリに書き込みができないことや，NFSであることを考えていない部分がある(例えば，/etc/rc.d/init.d/rc.sysinit内の.autofsckファイルの作成条件など)．
- /etc/mtabを/proc/mountsへのシンボリックリンクとする場合，一般利用者からmount/umountを許可するuser mount機能が働かない．
  これは，sudoを利用したり，supermount機能を利用したりすることで比較的容易に回避できる．
- /devがRAMディスクを含む別のパーティションになっている場合，devfsであると決めうちになっているRPMが存在する(した)．

## 5. まとめ

本稿では，大学などの教育用計算機システムに適し，TCO削減に寄与するとされているDiskless構成をとることが可能なLinuxをベースにしたシステムの設計と実装について述べた．Vine Linuxをベースに，元のディストリビューションの構成をほとんど変更することなく実現したことで，実質的に新たなディストリビューションになってしまうことを避け，OS自体のメンテナンスコストが上昇することを軽減している．

今後の課題として，サーバ性能に関するサイジング情報の蓄積や，サーバの性能やコストをどこまで下げられるか，実際の運用環境に適用した上で，どの程度のTCO削減に繋がるかを調査する，などが挙げられる．

## 謝辞

有益な意見や協力を戴いた，有限会社ヴァインカーヴの鈴木大輔さん，やまだあきらさん，松林弘司さん，に感謝します．

## 参考文献

[1] 桝田他："大規模分散ネットワーク環境における教育用計算機システム"，情処学会会誌，pp.225–281，Vol.45，No.3，2004.

[2] 江藤，田中，松原，渡辺，渡辺，只木："演習用Windows端末群のディスクレスによる安定運用"，情処論，pp.2–11，Vol.45，No.1，2004.

[3] 安東，田中："教育用計算機環境の事例 - Mac OS X 編"，情処学会会誌，pp.243–246，Vol.45，No.3，2004.

[4] 安倍，石橋，藤川，松浦："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"，情処論，pp.3468–3477，Vol.43，No.11，2002.

[5] Mintwave社，VID，
http://www.mintwave.co.jp/tc/vid.html.

[6] Apple社，NetBoot，
http://www.apple.com/jp/server/macosx/netboot.html.

[7] Intel社，Preboot Execution Environment (PXE) Version 2.1，
ftp://download.intel.com/labs/manage/wfm/download/pxespec.pdf/.

[8] ISC，DHCPD，
http://www.isc.org/sw/dhcpd/.

[9] Tim Hurman，PXE daemon，
http://www.kano.org.uk/projects/pxe/.

[10] Vine Linux，http://www.vinelinux.org/.

[11] pdumpfs，
http://www.namazu.org/~satoru/pdumpfs/.