

DAVfs と Aufs を用いた組み込み Linux による シンククライアントシステムの試作

竹川知孝[†] 並木美太郎[†]

近年、情報家電や PC を家庭内でつなぐホームネットワークが普及している。情報家電の多機能化により OS が搭載された組み込み機器がネットワークで繋がる一方で、固有の設定は個別に対応しなければならない。そこで、本研究は DAVfs と Aufs を用いて組み込み機器を対象とする Linux シンククライアントシステムを試作し、クライアントの管理を容易にする。WebDAV サーバがクライアントのシステムファイル、ユーザデータを一元管理するので、クライアントの設定を容易になり、クライアントが使用するユーザデータの分散を防ぐ。評価では、クライアントのローカルに必要な `initrd` を約 6MB まで削減できた事を確認し、組み込み機器をクライアントに加える事を容易にした。

Prototyping of a Thin Client System Using the DAVfs and Aufs on Embedded Linux

Tomotaka Takekawa[†] Mitaro Namiki[†]

Home network constructed with appliances and PC become popular in recent years. While embedded systems equipped with OS are connected to the network, a setting should correspond individually. In this paper, we simplify setting of clients by making prototyping of a Linux thin client system that target embedded systems to use the DAVfs and Aufs. Because WebDAV server manages client's system file and user data, it becomes easy to set of clients and prevents spreading user data. As the result of evaluation, we confirmed that size of `initrd` was only about 6MB and simplified that embedded systems was associated with thin client system.

1. はじめに

近年、情報家電や携帯電話の高性能化に伴って組み込み機器にも OS を搭載するケースが増加してきた。OS を搭載する事で組み込み機器のアプリケーションが直接ハードウェアを制御する必要がなくなり、複雑化する組み込み機器の制御が可能になっている。市場には実際に組み込み機器向け OS を搭載した製品が出回る中で、最近は組み込み Linux を採用する製品が多くなってきている。組み込み Linux を採用する事で既に Linux 用に開発されている豊富なミドルウェアを利用できるようになる。この結果、高機能化した組み込み機器で使用するソフトウェアを全て作成するのではなく、既存のソフトウェアを基にして新たなソフトウェアを開発する事が可能になる。

また、ネットワーク機能を搭載した組み込み機器の普及によりセンサーネットワークなどの組み込み機器をクライアントに

したネットワークが普及してきた。今後はセンサーネットワークだけでなく、家庭において情報家電同士を接続するホームネットワークが普及すると考えられる。家電同士がネットワークで接続される事で家電同士の連携はもちろんネットワークを介しての操作などの新しい使い方が考案され普及していくと考えられる。しかし、ネットワーク機能を備える事で通信ができるようになる半面、クライアントが使用するユーザデータや管理情報が分散し、クライアントの管理が難しくなる。そこで本研究では組み込み機器を対象にした Linux シンククライアントシステムの作成を試みる。シンククライアントシステムの利点である情報の一元管理によって組み込み機器が使用するユーザデータ、管理情報を集約し、クライアントの管理を容易にする。

2. 既存のシンククライアントシステム

既存のシンククライアントシステムについて述べ、組み込み機

[†] 東京農工大学
Tokyo University of Agriculture and Technology

器向けのシンクライアントシステムを構築する上での問題点を示す。

2.1 シンクライアントシステムの分類

シンクライアントシステムはサーバとクライアントで構成され、クライアントが起動する前にカーネル、initrd（実行ファイル、ライブラリ、設定ファイルを含んだファイルシステム）をサーバに置くのか、クライアントに置くのか、さらに起動後どちらで実行するかによってシンクライアントシステムの方式が変わる。

● 起動前にクライアントに配置

クライアントにカーネル、initrdを置き、クライアントが実行して起動する。クライアントにカーネルとinitrdが保存されているので、起動時にサーバへ接続しブートに必要なファイルを取得する必要はなくなる。しかし、クライアントにROMを大量に用意する必要があり、シンクライアントとしての意味をなくしてしまう。また、新しいカーネル、initrdへの更新を試みても、クライアントごと個別に対応して更新しなければならない。

● 起動前にサーバに配置

この場合はシステム起動後、カーネルとinitrdをサーバで実行する方式と、クライアントに転送して実行する方式でシンクライアントシステムの方式が2種類ある。

▶ ネットワークブート方式

システムの起動時にクライアントはサーバからカーネル、initrdを読み込み、その後クライアントで実行する。具体的にはe-toolsのLanPC2²⁾が挙げられる。クライアントの数が増えた時もサーバはクライアントにカーネル、initrdを提供するだけなのでクライアント数の変化に柔軟に対応できる。しかし、ブートに必要なカーネルとinitrdを転送するため帯域の広いネットワークが配備された環境でないと使えない。また、ブート時にどんなデータをクライアントに置くのかをシンクライアントシステムの設計段階で決定する必要がある。

▶ 画面転送方式

本方式はカーネルからinitrdに含まれるファイルの実行まで全ての処理をサーバで実行し、クライアントはサーバから送られてくる処理結果の表示とユーザからの入力を担当する。具体的にはCitrixのPresentation Server²⁾やVNCを利用したシステム³⁾が挙げられる。クライアントは入出力だけなのでCPUとメモリはそれほど必要ない。しかし、サーバの性能に依存しているシステムであるため、あるクライアントからサーバに負荷のかかる処理が要求されると、システム全体のパフォーマンスが低下する。

従来のシンクライアントシステムはクライアントの対象をPCにする事が多かった。しかしPCを対象にしているため、クライアントに加えるために要求されるスペックが高く、PCより性能の低い組み込み機器をクライアントに加える事が難し

い⁴⁾。特に起動前にクライアントのローカルにカーネル、initrdを配置する場合とネットワークブート方式の場合、クライアントにカーネルとinitrdを配置するため、クライアントのメモリを圧迫する。また、各クライアントがinitrdを必要とするので、クライアントごとにinitrdを用意し、initrdにIPアドレスの指定などクライアント個別の設定を加える事ができる。しかし、クライアントの個数が少ない場合は個別に対応できるが、クライアントの個数が増えるとクライアントごとに個別の設定を施したinitrdを用意するのが煩雑になる。クライアントにinitrdを配置すると、クライアント個別の設定を加える事ができるが、その設定を一元管理できないのが問題である。

組み込み機器はその製品の目的を達成するために専用設計されている。PCは汎用コンピュータとして潤沢なハードウェアを搭載した環境を利用できる一方で、組み込み機器の場合は目的のための機能以外を機器に実装する事はない。そのため組み込み機器ではPCのようにディスプレイやキーボードなどユーザへの入出力が豊富とは限らない。入出力機器がないクライアントで画面転送方式を使うのは不可能である。

2.2 ネットワークファイルシステムの問題点

起動前にクライアントにカーネル、initrdを配置する場合とネットワークブート方式の場合、ファイルサーバに存在するシステムファイル、ユーザデータをクライアントに転送するためネットワークファイル共有システムを選択しなければならない。UNIXでは昔からNFSが使用されてきた。Linuxでもネットワークファイル共有システムとしてNFSが使用されている。NFSは枯れた技術であり、安定しているという利点があるが問題点も含んでいる。NFSはファイアウォールを介してファイル共有をしようとした場合、使用するポートの指定などサーバの設定をしなければならない。WANでの利用を想定した時、NFSではファイアウォールを介した通信が困難な点が問題である。

3. 目標

本研究は、クライアントに組み込み機器を加え、WebDAVサーバによってクライアント固有の設定を一元管理する事を目標にする。

WebDAVとはWeb-based Distributed Authoring and Versioningの略であり、http1.1をwebサーバ上のファイルやフォルダを編集できるように拡張したプロトコルである。従来のhttpはサーバが公開しているファイルをブラウザに送信するためのものであった。それに加えてWebDAVはファイルの送信や、サーバ上のファイル一覧の取得、ファイルの複製、移動、削除をhttpと同じ80番ポートを用いて実行できる。WebDAVはhttpの拡張であるため、以下のような特徴がある。

● http,httpsのポートだけを使用

WebDAVはhttpによるファイル共有はもちろん、httpsによる通信の暗号化も可能である。そのためWANでの使用に

において、NFS では困難であったファイアウォールの設定を容易にする。

- 特定の OS やハードウェアに依存しない

http は環境に依存しない。同じように WebDAV も特定の OS やハードウェアに依存しないプロトコルである。一般にファイルを扱うプログラムは、ファイル管理において OS の持つファイルシステムの機能の影響を大きく受けるが、WebDAV は http と同じように OS の影響を受けない。

クライアントはカーネルと必要最小限のファイルを持った `initrd` を TFTP サーバから取得してローカルに保存する。その他のシステムファイル、ユーザデータは WebDAV サーバから取得する。その後、クライアントはローカルにある `initrd` と WebDAV サーバから取得した `DAVs` を `Aufs` で統合し、クライアントのルートファイルシステムとして使用する。

`Aufs` はマウント時に `read only` と `read write` をそれぞれ指定した二つのディレクトリを統合することができる。つまり、統合して完成したディレクトリには指定した二つのディレクトリに含まれていたファイルが存在する事になる。このディレクトリに対する書き込みはすべて `read write` に指定したディレクトリに対して施されるので、`read only` に指定されたディレクトリには何も変更が加えられない。仮に `read only` に指定されたディレクトリに含まれていたファイルに対し変更を行っても、変更したデータは `read write` のディレクトリに隠しファイルとして保存される。`Aufs` を用いて `initrd` のディレクトリと `DAVs` のディレクトリを統合する事で、クライアントはローカルの `initrd` に存在するファイルと同じように WebDAV サーバにあるファイルを扱う事が出来る。

シンククライアントシステムの方式はネットワークブート方式とする。このためシステムの起動前にクライアントは何もデータを持っていないので、起動時に TFTP サーバから起動に必要なブートローダ、カーネル、`initrd` を送ってもらう。この時、クライアントに転送する `initrd` を共通にする。クライアントに転送する `initrd` は共通なので、個別の設定は WebDAV サーバで設定できる。このためクライアント個別の設定を一元管理する事ができる。

WebDAV サーバに接続するために必要な実行ファイルと、`init` スクリプトを実行するのに必要なシェル、それらを動かすのに必要な共有ライブラリを中心に構成された必要最小限のファイルを持った `initrd` をクライアントは使用する。残りのシステムファイル、ユーザが使用するデータは WebDAV サーバに保存して、必要になった時にクライアントがサーバに接続して使用できるようにする。WebDAV サーバに配置するファイルとクライアントに配置するファイルに分け、クライアントに配置するファイルを最小限にする事でクライアントに要求されるメモリを減らす事が可能となる。

4. 全体構成

本シンククライアントシステムのクライアントは `initrd` と `DAVs` を統合してルートファイルシステムとして使用する。

本来、クライアントが使用するファイルは1つのファイルシステムに存在するが、本システムはクライアントが使用するファイルを `initrd` と `DAVs` に分け、`DAVs` を取得するためのクライアント共通のファイルは `initrd` にクライアント個別のファイルは `DAVs` に置く。ルートファイルシステムに含まれるファイルをクライアント共通のファイルとクライアント個別のファイルに明確に分ける事で、シンククライアントシステムのために、サーバで一元管理しなければならないファイルを明確にする。そして、そのファイルを本シンククライアントシステムは WebDAV サーバで管理している。

本シンククライアントシステムは WebDAV サーバ、TFTP サーバ、DHCP サーバ、クライアントから構成される。図 1 に本シンククライアントシステムの概略図を示す。

WebDAV サーバはクライアントが利用するシステムファイルとユーザデータを格納するファイルサーバであり、クライアントからリクエストされたファイルを転送する。WebDAV サーバにクライアントにマウントされるディレクトリを提供する `DAVs` が置いてある。

TFTP サーバはクライアントがローカルで使用するカーネル、`initrd` を起動時に転送する。DHCP サーバは PXE(Preboot eXecution Environment)によるネットワークブート時に使用される。

ブート前のクライアントはブートローダ、カーネル、`initrd` を保持していない。WebDAV サーバ、TFTP サーバに接続してネットワークブートする。

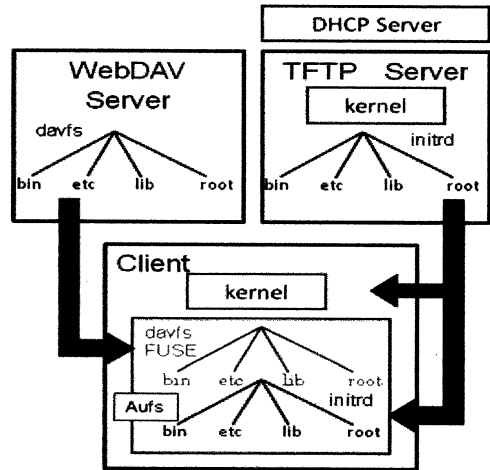


図 1 シンククライアントシステムの概略図

従来のネットワークブート方式のシンククライアントシステムはクライアントにカーネル、`initrd` を取得していた。つまりクライアントが取得したファイルを実際に使用するかに関わらず、`initrd` に含まれる全ての実行ファイル、設定ファイル、ライブラリファイル、ユーザデータの全てをクライアントのローカルに保存する事になる。本シンククライアントシステムでは全て取得するのではなく、これらのファイルの配置場所をクライアントとサーバに分けてクライアントは使用する。そのためにクライアントのブートを2段階に分割して考える。

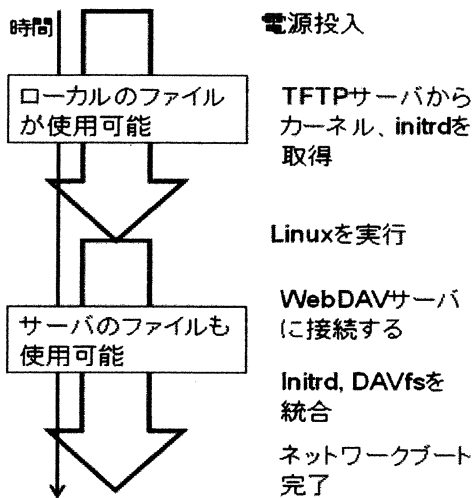


図 2 クライアントの2段階ブート

図 2 に示すように本システムのクライアントは電源投入後から Linux を起動するまでと、ネットワークブートが完了するまでの2段階でブートする。

以下にクライアントのブートの詳細を記す。

1. 電源投入後、TFTPサーバからブートに必要なカーネル、initrd を転送してもらう。TFTPサーバから転送されたカーネル、initrd を使って Linux をブートする。その後、クライアントは initrd の init プロセスを実行する。
2. クライアントは init プロセスを実行し、クライアントの設定を行う。最初に WebDAV サーバのマウントを行う。クライアントは WebDAV サーバをマウントして DAVfs を取得する。
3. クライアントは DAVfs を取得してからは図 1 に示すとおり、Aufs を使って TFTP サーバから送られてきた initrd と DAVfs によって提供されるディレクトリをひとつのルートファイルシステムに統合してブートを完了する。

TFTP サーバが転送する initrd には WebDAV サーバをマウントするための実行ファイルと実行ファイルを動かすための共有ライブラリ、busybox を置く。残りのシステムファイルとユーザが使用するデータは、WebDAV サーバマウント後、取得した DAVfs を用いて WebDAV サーバに保存してあるファイルを使用する。この結果、クライアントがローカルに置かなければならないファイルを最小限にする事ができる。

本システム的设计ではクライアントに置くファイルの機能を吟味し、クライアントに置くファイルを可能な限り削減する。そして、initrd のファイルサイズを少なくし、クライアントが組込み機器でも本システムに加えられるようにする。

5. クライアントとサーバ

本システムのクライアントとサーバ的设计について述べる。

5.1 クライアント

本システムのクライアントは NIC を搭載し、Linux が動作する機器を想定している。また、PXE に対応した機器では、PXE によるネットワークブートも可能である。ディスプレイ、キーボードといった入出力を備えている機器が望ましいが、クライアントが TFTP サーバから取得する initrd の init スクリプトで設定をして、ユーザの操作を必要とせずアプリケーションを起動させる事もできる。

本システムはネットワークブート方式のシンクライアントシステムであるため、ブート前のクライアントはカーネル、initrd を保持しない。クライアントは電源投入後、TFTP サーバからカーネルと initrd を取得し initrd 内の init スクリプトを実行する。この時点でクライアントが実行できる機能はカーネルと TFTP サーバから転送される initrd に用意された機能のみである。つまり、init スクリプトを実行する過程で WebDAV サーバをマウントするには、TFTP サーバから転送されるカーネルと initrd だけで WebDAV サーバをマウントできるようにカーネルと initrd を設計しなければならない。クライアントは TFTP サーバから取得した initrd 内の init スクリプトを実行し、WebDAV サーバにある DAVfs をマウントする。マウントするのは /mnt/server 内のディレクトリであり、bin には実行ファイル、etc には設定ファイル、lib には共有ライブラリ、root にはユーザデータが格納される。その後、DAVfs をマウントしたファイルシステムと initrd に存在するファイルシステムを、Aufs を用いて統合する。本システムのクライアントは TFTP サーバからカーネルと initrd を取得して、initrd 内の init プロセスの実行過程で WebDAV サーバから DAVfs を取得する。つまり、クライアントのローカルに必要なファイルは TFTP サーバから送られるカーネルと initrd のみである。以下では TFTP サーバから転送されるカーネルと initrd の詳細な設計について述べる。

5.1.1 カーネルの構成

クライアントが使用するカーネルは必要な機能を取捨選択するために再構築を行った。クライアントが使用するファイルシステムに含まれるファイルを少なくするため、必要な機能はロードダブルカーネルモジュールではなく静的にカーネルに取り込む事にする。以下にカーネルに取り込んだ機能を列挙する。

● FUSE(Filesystem in UserSpace)

FUSE[®]は、ユーザ空間のプログラムにファイルシステムを実装するためのインタフェースを提供する。本来、ファイルシステムはカーネル空間に存在し、ファイルシステムを作成するにはカーネルプログラミングと、VFS などのカーネル内で利用されている技術に関する知識が必要であった。

FUSE はファイルシステムのユーザ空間での構築を可能にし、アプリケーションとしてファイルシステムを作る事ができるようになる。本システムではクライアントが WebDAV サーバに接続する時に実行する `davfs22)(mount.davfs)` で FUSE が利用される。

● NIC のドライバ

多くの Linux ディストリビューションは NIC のドライバをカーネルモジュールで提供し、起動時にクライアントが読み込む設計になっている。しかし、組込み機器でいくつものカーネルモジュールを用意するのは、多くのメモリを必要とするため好ましくない。そこで、本システムではクライアントを限定して、その機器に合った NIC のドライバをカーネルに静的に取り込む。

5.1.2 `initrd` 中のファイル

TFTP サーバから送られてくる `initrd` は WebDAV サーバに接続するための最小限のファイルだけが配置され、残りのファイルは WebDAV サーバに接続して利用する。WebDAV に接続するための最小限のファイルとして、WebDAV をマウントする `mount.davfs`、`mount.davfs` の実行に必要な共有ライブラリ、`busybox` が考えられる。これら最小限のファイルを格納するディレクトリには実行ファイルを格納するための `/bin`、共有ライブラリを格納するための `/lib`、設定ファイルを入れるための `/etc`、デバイスノードを格納するための `/dev`、可変的なシステムファイルを格納する `/var` が必要になる。また、`proc` ファイルシステムをサポートしようとする `/proc` も必要になる。以上の事を踏まえて TFTP サーバからクライアントに転送される `initrd` のディレクトリ構成を考える。

WebDAV サーバのファイルが含まれる DAVfs は `/mnt` 以下のディレクトリにマウントされる。しかし、WebDAV サーバをマウントした時点では TFTP サーバから転送された `initrd` 内のディレクトリと、WebDAV サーバのマウントによって利用できるようになったディレクトリが別々のディレクトリで存在し、カーネルがこれらのファイルを利用しやすい状態とは言えない。そこで、図 1 に示したように `Aufs` によって、TFTP サーバから送られた `initrd` と WebDAV サーバをマウントして得た DAVfs が統合されたルートファイルシステムとして利用できるようにする。表 1 にクライアントが使用する `initrd` と DAVfs の内容を示す。

表 1 クライアントが使用する `initrd` と DAVfs

	<code>initrd</code>	DAVfs
<code>bin</code>	<code>mount.davfs busybox</code>	クライアント個別の実行ファイル InterfaceKit-simple (センサーの値を読むのに使用)
<code>dev</code>	デバイスノード	×
<code>etc</code>	<code>mtab init.d/rcS</code>	クライアント個別の設定ファイル <code>sysconfig/network</code>

<code>lib</code>	<code>mount.davfs</code> の実行に必要なライブラリ <code>ld-2.3.6.so</code> <code>libc-2.3.6.so</code> <code>libcrypt-2.3.6.so</code> <code>libdl-2.3.6.so</code> <code>libneon.so.26.0.4</code> <code>libxml2.so.2.6.27</code>	クライアント個別のライブラリ <code>libphidget21.so</code> (<code>InterfaceKit-simple</code> の実行に必要)
<code>mnt</code>	<code>server/bin,etc,lib,root</code>	×
<code>proc</code>	<code>proc</code> ファイルシステムをマウント	×
<code>root</code>	何も保存しない	ユーザデータ
<code>sbin,usr</code>	<code>busybox</code> へのシンボリックリンク	×
<code>var</code>	<code>cache run</code>	×

(× : WebDAV サーバで提供されない)

● `/bin`

`/bin` に含まれるコマンドは `busybox` と `mount.davfs` の二つである。`busybox` はクライアントが使用する主なコマンドをまとめた実行ファイルであり、`mount.davfs` は WebDAV サーバをマウントするための実行ファイルである。`/bin` は WebDAV サーバをマウントした `/mnt/server/bin` と `Aufs` によって統合される。統合後はローカルに存在する `busybox` と `mount.davfs` だけでなく、WebDAV サーバ上のコマンドも実行できるようになる。

● `/dev`

デバイスノードが保存されている。つまり、`/dev` にはデバイスにアクセスするためのファイルが格納されている。クライアントに接続されたデバイスを使用するには、このディレクトリに保存してあるデバイスノードを通して利用する。ファイルが実体としてあるわけではないので、ファイルサイズは 0 になる。使用用途を考慮するとこのディレクトリを WebDAV サーバに置く事はできない。

● `/etc`

Linux の各種設定ファイルが保存される。TFTP サーバから得られる `initrd` 内の `/etc` には `mtab` が配置されている。`mtab` にはマウントされたファイルシステムの一覧が書き込まれるので、`mount.davfs` によってマウントされたディレクトリの情報が書き込まれる。`mount.davfs` で利用されるので、TFTP サーバから転送される `initrd` に含めなければならない。また、`/etc/init.d/rcS` には起動時に実行される `init` スクリプトが記述されている。

● `/lib`

システムが起動するのに最低限必要な共有ライブラリが入っている。起動するのに最低限必要な共有ライブラリとは `bin` に置いてある `mount.davfs` が動作するのに必要な共有ライブラリである。また、`/mnt/sever/lib` は WebDAV サーバをマウントするためのディレクトリである。`/lib` と `/mnt/server/lib` を、`Aufs` を用いて統合することで WebDAV

サーバ上のライブラリも利用できるようになる。

- /mnt

WebDAV サーバをマウントするディレクトリが収められている。具体的には bin, etc, lib, root が /mnt/server に用意されており、これらのディレクトリが起動時に init スクリプトの実行で WebDAV サーバをマウントする。

- /proc

このディレクトリには proc ファイルシステムがマウントされ、カーネル内部の情報、実行しているプロセスにアクセスするためのファイルが格納される。このディレクトリに格納されるファイルは /dev 以下のファイルと同じように実体のないものなのでシンボリックリンクを除いてファイルサイズは 0 である。proc ファイルシステムは init スクリプトの実行過程でマウントされるので、TFTP サーバから転送される initrd 内の proc は内部に何もファイルが入っていない。

- /root

このディレクトリは WebDAV サーバをマウントした /mnt/server/root と aufs で統合される。このため TFTP サーバから転送された直後は何も入っていない。統合後、/root にはアプリケーションで使用するユーザデータを保存するために使用される。

- /sbin, /usr

/sbin にはシステム管理のためのコマンド、/usr には bin ディレクトリが入っている。本システムのクライアントが利用する主なコマンドは busybox で提供される。このため、これらのディレクトリには busybox へのシンボリックリンクが収められる。

- /var

本システムでは内部に cache, run の二つのディレクトリがある。PXE サーバから送られてきた時点では、これらのディレクトリも空である。その後、cache はアプリケーションのキャッシュデータが保管され、run には起動後にシステムデータが格納される。

5.2 WebDAV サーバ

WebDAV によって接続されるファイルサーバである。initrd には DAVfs を取得するためのファイルが格納されるのに対して、initrd に格納する必要がなかった InterfaceKit-simple や sysconfig/network などのシステムファイルとユーザデータを保存するために本システムでは利用する。そして、これらのファイルを一元管理するために用いられる。

各クライアントに TFTP サーバから転送される initrd 内のディレクトリ /mnt/server に存在する bin, etc, lib, root を、クライアントが init スクリプト実行して WebDAV サーバにマウントする。この結果、データの実体は WebDAV サーバに保存されているファイルとなり、ユーザデータの書き込みも WebDAV サーバに対して行う。TFTP サーバから転送される initrd 内の /mnt/server にあるディレクトリが WebDAV サーバをマウントするため、サーバにはそれぞれに対応したディレクトリが用意されている。

5.3 TFTP サーバ

ネットワークブート型のシンクライアントシステムを実現するためのサーバである。TFTP サーバで公開しているファイルをクライアントが取得して起動する。表 2 に TFTP サーバが公開するファイルを記す。

表 2 TFTP サーバで公開されるファイル

名前	詳細
uImage	クライアントが使用するカーネル
initrd	クライアントが使用する initrd
pxelinux.0	PXE によるネットワークブート時に使用されるブートローダ
pxelinux.cfg	PXE のための設定ファイルが格納される

5.4 DHCP サーバ

PXE によるネットワークブート時にクライアントの電源投入直後、クライアントの IP アドレス取得のため使用される。

6. 実現

本システムはサーバとクライアントをそれぞれ PC と玄箱 PRO の両方で実現した。表 3 にそれぞれのハードウェアの詳細を記す。

表 3 実現に用いた機器の仕様

PC(クライアント)	Core2Duo1.66GHz, メモリ 1GB, 1GbpsEthernet
PC(サーバ)	Celeron 2.5GHz, メモリ 512MB, 1GbpsEthernet
玄箱 PRO	ARM9 互換 Marvell 製 88F5182 400MHz, メモリ 128MB, 1GbpsEthernet

6.1 クライアント

クライアントが WebDAV サーバに接続するために davfs2 を利用した。davfs2 は FUSE を利用して WebDAV サーバをローカルのファイルシステムにマウントするためのプログラムである。プログラムをコンパイルする事で mount.davfs を得る。このプログラムを実行する事によってローカルのファイルと同じようにシェルによるファイル操作が可能になる。通常の mount.davfs は initrd 上で動作するように設計されていない。そこで、プログラムに変更を加えて initrd 上でも動作するようにする。

- クライアント PC

クライアント PC は PXE を用いてネットワークブートする。このためクライアント PC は BIOS で PXE によるネットワークブートを選択しなければならない。

- クライアント玄箱

クライアント玄箱は内蔵するブートローダである U-Boot でブートする。U-Boot は TFTP サーバを利用したネットワー

クブートに対応しているため、TFTP サーバの IP アドレスを指定しクライアント玄箱はネットワークブートする。

(単位: 秒)

6.2 サーバ

それぞれのサーバは Linux 上でデーモンを起動する事で実装する。WebDAV サーバは WebDAV モジュールを読んだ Apache を起動する。

- サーバPC

Fedora6 上でそれぞれのサーバを起動する。使用するソフトウェアは Apache2.2, DHCP Server V3.0.5, tftp0.42 である。それぞれの設定ファイルを環境に合わせて設定する。

- サーバ玄箱

debian4.0 で Apache2.2, DHCP Server2.0p15 tftp0.17 を実行しシンクライアントサーバとする。

7. 評価

構築したシンクライアントシステムの評価を行う。評価項目はサーバ、クライアント間の転送速度、作成した initrd のサイズである。

7.1 NFS と WebDAV の転送速度の比較

本シンクライアントシステムはネットワークファイル共有システムに WebDAV を利用している。UNIX では同じようにファイルを転送するプロトコルとして昔から NFS が利用されてきた。本システムの評価として WebDAV と NFS の転送速度の比較を行う。

計測方法は 2.5MB, 5MB, 7.5MB, 10MB の null 文字が大量に記載されたダミーファイルを dd コマンドで転送し、実行時間を計測する。転送するファイルの最大値が 10MB なのはクライアントが玄箱の場合、使用しているカーネルが RAM ブロックデバイスを 32MB までしか認識しないためである。転送したユーザファイルのサイズが大きくなると、カーネル、initrd に含まれるファイル、転送するユーザファイルの合計が 32MB を超えてしまう。したがって転送するダミーファイルのサイズは、クライアントが玄箱の場合のユーザファイルの最大値である 10MB とする。

本システムが起動した後に、クライアントで以下のコマンドを実行して計測する。

```
time dd if=/root/testfile of=/result
```

コマンドを WebDAV と NFS で実行した時のそれぞれの結果を表 4, 表 5 に、転送時間から算出した転送速度を図 3, 図 4 に示す。

表 4 WebDAV 使用時の計測結果

サーバ	クライアント	2.5MB	5MB	7.5MB	10MB
PC	PC	0.13	0.32	0.40	0.55
	玄箱	0.84	1.69	2.45	3.4
玄箱	PC	0.19	0.35	0.62	0.77
	玄箱	0.67	1.35	1.97	2.76

表 5 NFS 使用時の計測結果

サーバ	クライアント	2.5MB	5MB	7.5MB	10MB
PC	PC	0.28	0.39	0.51	0.73
	玄箱	0.71	1.59	2.29	3.26
玄箱	PC	0.21	0.79	0.82	1.22
	玄箱	1.87	1.67	1.79	3.46

(単位: 秒)

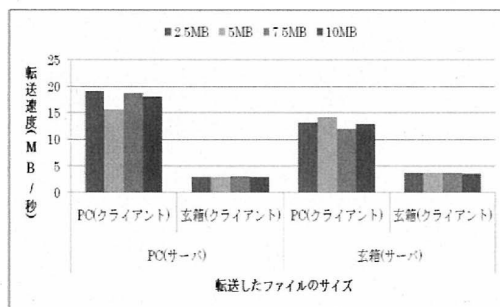


図 3 WebDAV 使用時の転送速度

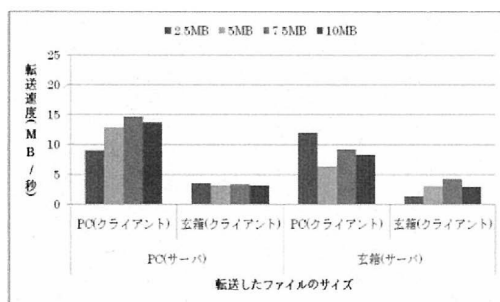


図 4 NFS 使用時の転送速度

転送速度から導いた WebDAV と NFS の平均転送速度を表 6 に示す。

表 6 平均転送速度

サーバ	クライアント	WebDAV の平均転送速度(MB/秒)	NFS の平均転送速度(MB/秒)
PC	PC	17.9	12.5
	玄箱	2.98	3.25
玄箱	PC	13.1	8.89
	玄箱	3.72	2.85

7.2 initrd のサイズ

TFTP サーバから転送される initrd の容量を確認する。initrd は圧縮して保存されており、使用する時に解凍する。圧縮をしていない状態で initrd 内のディレクトリのサイズと合計を PC と玄箱のそれぞれで調べる。また、比較対象として fedora の起動時に使用される initrd のサイズも掲載する。

表 7 使用する initrd のサイズ

名前	PC の initrd(KB)	玄箱の initrd(KB)	fedora の initrd(KB)
bin	1440	632	4336
dev	8	13	8
etc	4	6	24
init	4	×	4
lib	5536	5367	492
mnt	×	9	×
proc	4	1	4
root	4	1	×
sbin	×	1	1
usr	2192	3	3
var	12	5	5
合計	9204	6038	4877

(× : 使用しない)

8. 考察

クライアントに必要なファイルをサーバとローカルに分ける事で、組込み機器向け Linux シンククライアントシステムを構築した。クライアントに置くファイルと WebDAV サーバに置くファイルを選別する事で、クライアントのローカルに必要なメモリを少なくした。この結果、組込み機器のように必要最小限のメモリしか搭載できない機器でもシステムに加える事を容易にした。また、クライアントは Linux を起動するため既存のソフトウェアを利用、またはシステムに適したアプリケーションを開発し、用途に応じた Linux シンククライアントシステムを構築する事が可能である。

クライアントが利用するシステムファイル、ユーザデータは WebDAV サーバの DAVs、TFTP サーバの initrd で管理している。特に WebDAV サーバの DAVs はクライアント個別のファイルを格納し、クライアントの要求に応じて、必要なファイルを提供している。このため、システムファイル、ユーザデータを個々のクライアントが保持しシステムファイル、ユーザデータを分散させるのではなく、WebDAV サーバによる一元管理を行っている。WebDAV サーバのファイルを変更するだけで、クライアントの IP アドレスなどの個別の設定の変更できるようにし、本シンククライアントシステムはクライアントの管理を容易にした。

評価では、WebDAV と NFS の転送速度の比較を行い、本システムのファイル共有システムとして使用する WebDAV の特性を調べた。評価結果は表 6 より PC をサーバ、玄箱をクライアントにした場合を除いて PC(サーバ)-PC(クライアント)で 43%、玄箱(サーバ)-PC(クライアント)で 47%、玄箱(サーバ)-玄箱(クライアント)で 30%NFS より WebDAV が早かった。WebDAV と NFS の違いとして、WebDAV が http, https を用いた通信を可能とするだけでなく、転送速度においても違いがある。

本システムに必要なファイルを取捨選択し玄箱が使用する initrd を約 6MB まで小さくすることができた。これは PC で

使用するディストリビューションのルートファイルシステムより大幅にサイズを削減している。このため玄箱のような資源が潤沢でない組込み機器でも本システムのクライアントに加える事が可能になった。また、表 7 で玄箱の initrd の内訳を確認すると lib が大部分を占めていることがわかる。これは mount.davfs を動作させるのにたくさんのライブラリファイルが必要としたためである。一方で、PC の initrd が玄箱の initrd よりサイズが大きくなっている。これは PC の initrd を fedora の initrd を基盤に作成したので、玄箱の initrd よりサイズが大きくなった。

9. おわりに

本研究では、組込み機器を対象にした Linux シンククライアントシステムの作成を試みた。結果、クライアントに必要な initrd を 6MB まで削減し、WebDAV と aufs を用いた組込み機器による Linux シンククライアントシステムを作成した。また、クライアントに必要なユーザデータ、設定ファイルを WebDAV サーバで一元管理することでクライアントの管理を容易にした。

今後は、クライアントの管理を容易にする本システムの利点を生かし、組込み機器によるクライアントが多数必要なセンサネットワークなどの応用システムを作成したい。

参考文献

- 1) LanPC2. <http://www.lanpc.jp/index.html>.
- 2) Citrix. <http://www.citrix.co.jp/>.
- 3) 高橋一志, 笹田耕一, 竹内郁雄 : WebVDI のための VNC Proxy, 情報処理学会研究報告, 2008-OS-109, pp. 85-92(2008).
- 4) 高橋竜男, 高橋修, 水野忠則 : モバイル向けシンククライアントシステムの検討, 情報処理学会論文誌 Vol. 45, No. 5, pp. 1417-1431(2004).
- 5) aufs. <http://aufs.sourceforge.net/>.
- 6) FUSE(Filesystem in Userspace). <http://fuse.sourceforge.net/>.
- 7) WebDAV Linux File System (davfs2). <http://dav.sourceforge.net/>.
- 8) 宮本久仁男, 田中英彦 : シンククライアントアーキテクチャをベースにしたセキュアクライアントの検討, 情報処理学会研究報告, 2007-CSEC-038, pp. 305-310(2007).
- 9) 金井遵, 須崎有康, 八木豊志樹 et al : HTTP-FUSE-KNOPPIX-BOX によるモバイルシンククライアントシステムの実現, 電子情報通信学会論文誌「ユビキタス時代の情報基盤技術」特集号, Vol. J90-D, No. 6, pp. 1383-1393(2007).
- 10) Albert M. Lai, Jason Nieh et al : Improving Web Browsing on Wireless PDAs Using Thin-Client Computing, International World Wide Web Conference archive Proceedings of the 13th international conference on World Wide Web, pp. 143-154(2004).
- 11) 濱田正博:シンククライアントのすべてがわかる, 日経 BP 企画, ISBN4-86130-202-1(2006).