

移動先で同じユーザ環境を実現するための携帯ファイルサーバ

木村 哲郎, 木津 左千夫, 村永 哲郎

{kimura, skizu, muranaga}@isl.rdc.toshiba.co.jp

(株) 東芝 研究開発センター 情報・通信システム研究所

本報告では、自分が使うファイル全てを携帯機に格納して持ち歩き、オフィスや外出先でデスクトップ機が利用可能な時には、携帯機をファイルサーバとし、操作性の優れたデスクトップ機を使用する利用形態の提案と、携帯ファイルサーバのプロトタイプについて説明する。

Portable File Server for Providing the Same Personal Environment in Foreign Networks

Tetsuro Kimura, Sachio Kizu, Tetsuro Muranaga

In this paper, we propose a portable file server for providing same personal environment at any site. A user stores all personal files in a mobile host which is a file server in his home office. He can also get personalized environment at a desktop host in a foreign network, by connecting his mobile host which acts as his file server there.

1 はじめに

コンピュータの著しい小型軽量化により、気軽に持ち運べていつでもどこでもコンピュータを利用できるモバイルコンピューティングが可能となった。また二次記憶の大容量化により、ユーザの持つ個人ファイル全てを携帯機に格納することが出来るようになった。

しかし、携帯機の性能の飛躍的向上の一方で、小型化に伴う操作性の低下が問題となっている。狭い画面や小さいキーボードによる本格的な作業は大きなストレスとなる。そこで、外出先などでデスクトップ機が利用可能な場合にはデスクトップ機で作業をするといった使い分けが考えられる。この時、携帯機内データへのアクセスとともに重要な課題に、使い慣れたユーザ環境の再現がある。ユーザ環境がカスタマイズされていないデスクトップ機は、たとえ画面やキーボードが使いやすくても、やはりユーザにはストレスとなる。

そこで、ユーザの持つ個人ファイル全てを格納した携帯機にファイルサーバ機能を持たせ、デスクトップ機からそれらを利用するとともに、ユーザ環境も携帯機内のカスタマイズ情報を用いて構築する利用方法が有望となる。携帯機は自分のオフィスであっても、ユーザの個人ファイルを提供するファイルサーバとすることにより、オフィスで、移動途中で、そして外出先でと、全ての状

況で最新の情報を保持し、すぐにアクセスできるというメリットをもたらす。またレプリケーションしないため一貫性維持の問題からも解放される。

ここでの課題は、ネットワークドメインの移動に対応したファイルサーバ機能の実現である。とくにシームレスなファイルアクセスの実現とドメイン毎のアクセスコントロールが重要となる。本報告では、移動先のネットワークへの適応機構を持つ携帯ファイルサーバの提案と、設計と実装について述べる。

2 携帯ファイルサーバの提案

複数のドメインにアカウントを持つユーザにとって、各ドメインのファイル/データ管理は煩雑ではあるが非常に重要である。ドメイン間で共有するファイル/情報の一貫性維持は、失敗すれば貴重な情報を失う危険性がある。また、ユーザ環境のカスタマイズは作業効率に大きな影響を与えるため、ドメイン毎のシステム環境の違いを吸収し、徐々に変化するカスタマイズデータのー貫性を保つ作業もまた煩雑ではあるが重要である。

ユーザの個人ファイルやカスタマイズデータの管理方法は、利用環境とくにOSに大きく依存するため、上述の問題に対する解決方法も異なる。本報告では、UNIX環境を対象とした解決方法について提案を行う。UNIX

環境では、ユーザ環境を設定するカスタマイズデータは、ホームディレクトリ下にファイルとして保存するため、各ドメインのホームディレクトリ下の同期をとることにより、上述の問題は解決される。

そこで、通常オフィスのファイルサーバを持つホームディレクトリ下の全てのファイルを、携帯機のハードディスクに格納する。オフィスでデスクトップ機を利用する時も、携帯機をネットワークに接続しておき、デスクトップから携帯機のホームディレクトリをマウントして使用する。ユーザがオフィスで作成したファイルや受け取ったメール、スケジュールデータなどすべてが携帯機上に蓄積される。すなわち、ユーザが外出する時点で、携帯機上には最新の情報が格納されており、携帯機をネットワークから切り離す際に特別なダウンロードなどの処理を必要としない。

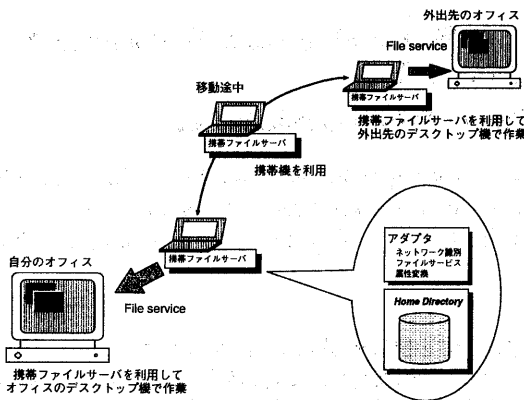


図1: 携帯ファイルサーバの利用イメージ

ユーザは、常に手にしている携帯機内に最新のファイルがあるため、外出途中では、携帯機を利用してファイルの参照や更新が可能である。

さらに移動先のオフィスや自宅などデスクトップ機が利用可能なときには、携帯機をネットワークに接続し、デスクトップ機に対してホームディレクトリを提供するファイルサーバとする。

このように携帯機を、いつでもどこでもホームディレクトリを提供するファイルサーバとして利用する事により、レプリケーションを行わない一元管理が可能となり、オフィスや自宅、移動先といった利用場所によらない一貫したユーザ環境が実現可能となる。

3 携帯ファイルサーバの設計

3.1 携帯ファイルサーバに求められる機能

さて、このような利用形態を実現するための要件について考える。携帯機は複数のネットワークドメインを渡り

歩くことになるが、携帯機をネットワークに接続したり切り離す時に必要となるシステム管理を自動化し、短時間でそれが終了することが重要である。携帯機を移動先のネットワークに接続し、そこでファイルサーバとして機能させるためには、以下の課題を克服しなければならない。

- ネットワークコンフィグレーションの変更

移動に伴うネットワークからの切り離しと再接続を検出し、移動先のネットワークで有効な IP アドレスの取得しなければならない。

- ネットワークドメインへの適合

異なるユーザ管理ドメインに移動した際に、ファイルのアクセス制御のキーとなるユーザ ID やグループ ID の変換が必要となる。

- ホームディレクトリ設定

移動先のデスクトップで login をした際に、ホームディレクトリが携帯機内のホームディレクトリに設定されなければならない。

以下、各項目について説明する。

3.1.1 ネットワークコンフィグレーションの変更

携帯機を移動先のネットワークに接続する場合、まず最初に IP アドレスを取得しなければならない。一般的には IP アドレスはシステム管理者に申請して取得するが、移動ホストはネットワークへの接続が一時的であるため、IP アドレスの自動割り当てを利用するのが一般的である。本システムでは広く普及している DHCP[1] を利用している。

ただし、DHCP では、システムが稼働中にネットワークを移動することが考慮されていないため、移動先でのネットワーク接続後も IP アドレスの再取得をせず無効な IP アドレスをそのまま利用してしまう。このため、稼働中のネットワーク移動を検知し、ネットワークに接続した際に再度 IP アドレスを取得し直す機能を追加する必要がある。

3.1.2 ネットワークドメインへの適合

異なる管理体系にあるネットワークは、ユーザ ID やグループ ID などの管理情報が各ドメインで独立して管理されるため、あるユーザに与えられるユーザ ID はネットワーク毎に異なるのが一般的である。

携帯機がドメインを跨って移動する場合、接続するドメインに適合するように携帯機内部の管理情報を変換するメカニズムが必要になる。特に、携帯機をファイルサーバとする場合、個々のファイルに設定されているオーナーやパーミッション情報を変換しなければならない。

たとえば、携帯機内のユーザ ID やグループ ID などの管理情報をオフィスのドメインに合わせるとする。携

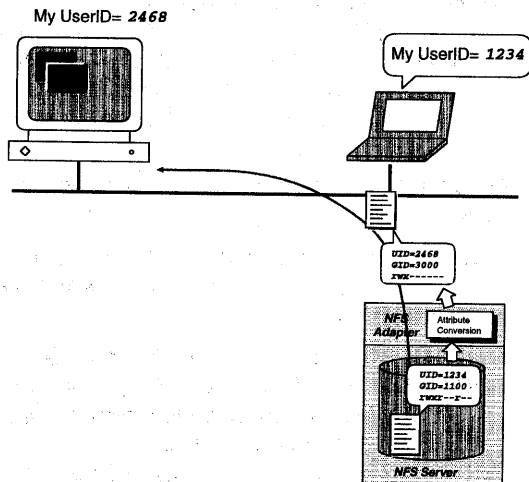


図 2: 外出先のネットワークメインに適合したファイルサービス

携帯機の持ち主は、移動先ではそのドメインに属するデスクトップ機に login し、そのデスクトップ機から携帯機のファイルをアクセスすることになる。このときファイルアクセスは移動先の管理体系に基づいたユーザ ID で行われるが、携帯機内のファイルはオフィスの管理体系に合わせてあり、ファイルのオーナー ID とアクセスするユーザの ID が異なるため、自分のファイルとして扱うことができない。

UNIX のファイルシステムでは、ファイルアクセスは、オーナーユーザ、グループ、アザーで制御される。グループ ID によるアクセスコントロールでは、オフィス内で定義されたグループに対応するグループが移動先に無い場合が一般的であり、単純な ID の変換だけでは対応できない。また、アザーに対してアクセス許可が設定されている場合でも、オフィス内でのアザーに適合するユーザ群は、移動先のユーザ群とは母集団が全く異なるため、注意が必要である。

このため、携帯機の接続しているネットワークメインに合わせた、より柔軟なアクセス制御が求められる。

3.1.3 ホームディレクトリ設定

UNIX 環境では、さまざまなプログラムが起動時に参照する設定ファイルをホームディレクトリ下に置くのが慣例である。このため、各種設定ファイルを格納している携帯機のファイルシステムをホームディレクトリとしてマウントすることにより、オフィスや外出先のデスクトップ機のユーザ環境を携帯機内で集中して管理できるため大変有用である。

動的に IP アドレスが割り当てられる携帯機内にホームディレクトリを設定するためには、ユーザが移動先のデスクトップ機に login した際に、ネットワークに接続した携帯機を探しだし、携帯機のエクスポートするホームディレクトリを持つファイルシステムをマウントするメカニズムが必要となる。

ユーザが login し、ホームディレクトリをアクセスした時に、オンデマンドでファイルシステムをマウントする自動マウント機構（オートマウンタ）は、広く普及している。しかし、移動先のネットワークに携帯機を接続し、動的に IP アドレスを取得する場合、オートマウンタがマウントすべき携帯機のアドレスが分からないため、マウントができない。このため、接続した携帯機に与えられた IP アドレスを調べるメカニズムとオートマウンタの協調動作が必要となる。

3.2 携帯ファイルサーバの構成

ユーザが行く先々に携帯ファイルサーバを利用できるためには、携帯ファイルサーバ特有の機能をオフィスや外出先にあるネットワーク側にできる限り持たせずに、携帯機側に集約することが好ましい。携帯ファイルサーバはこの方針に基づき設計を行った。

オフィスや外出先などの、携帯機を接続するネットワーク環境については以下の項目を仮定した。

- DHCP による IP アドレスが取得可能
- 分散ファイルシステム NFS[3] が利用可能
- 自動マウント機構 amd[4] によるユーザのホームディレクトリのマウントが可能

図 3 に携帯ファイルサーバの構成を示す。

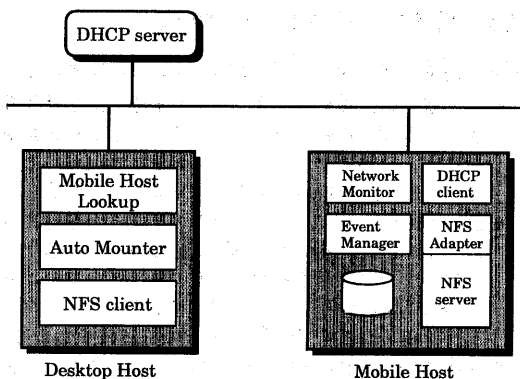


図 3: 携帯ファイルサーバの構成

まず携帯機上のモジュールについて簡単に説明する。

DHCP clinet : 接続したネットワークの DHCP サーバから IP アドレスを取得する。IP アドレスが割り当てられると、EventManager にアドレスを通知する。

Network Monitor : ネットワークを監視し、携帯機のネットワークからの切り離しと再接続を検知し、EventManagerに通知する。

Event Manager : Network Monitor および DHCP client から通知されるイベントに応じて、他のモジュールに指令を送る。

NFS Adapter : 携帯機の提供するファイルサービスを、接続しているネットワークドメインに適合させるモジュール。デスクトップ機上の NFS client と携帯機上の NFS server の通信を仲介し、ファイルの管理情報を変換する。

次にデスクトップ機上のモジュールについて説明する。

NFS client : 広く普及している分散ファイルシステム NFS のクライアントであり、デスクトップ機から携帯機内への透過なファイルアクセスを実現するためのモジュール。携帯ファイルサーバのための拡張は必要ない。

Auto Mounter(amd) : NFS によるファイルアクセスを行う場合、利用開始時にまず mount しなければならない。amd は、最初のファイルアクセスが行われたときに、オンデマンドに mount を行うためのモジュールであり、広く普及している。携帯ファイルサーバのための拡張は必要ない。

MobileHost Lookup : デスクトップ機が携帯機のファイルシステムをマウントするためには、携帯機に割り当てられた IP アドレスとパス (マウントされるディレクトリ) を知らなければならない。このモジュールは、ネットワーク上で携帯機を検索し、携帯機の IP アドレスとパスの情報を獲得し、amd に通知する。

3.3 携帯ファイルサーバの動作

以下、携帯ファイルサーバの動作について説明する。

- ユーザが携帯機をネットワークに接続する。携帯機上の DHCP client がネットワーク上にアドレス要求を送出し、DHCP サーバからのアドレス割り当てを受け、EventManager にアドレス獲得を通知する。
- EventManager は、DHCP クライアントから通知された IP アドレスから、携帯機が接続しているネットワークドメインを認識し、そのドメインでのユーザ ID およびグループ ID の変換情報とファイルパーミッションの変換情報を NFS Adapter に通知する。
- NFS Adapter は変換情報を元に、内部に変換テーブルを構築し、そのネットワーク上でのファイルサービスの準備をする。
- 携帯機をネットワークに接続したユーザは、デスクトップ機に向かい、login する。デスクトップ機上の login プロセスは login 名とパスワードからユーザ認証を行い、正しい場合には、続いてホームディレクトリにアクセスする。

- amd(自動マウンタ) は、ホームディレクトリアクセスにより起動される。amd はまず、MobileHost Lookup を起動してホームディレクトリを提供する携帯機を検索し、携帯機の IP アドレスとホームディレクトリのパスの情報を獲得する。IP アドレスとパスにより NFS マウントを行う。

以上の手続きで、携帯機内のファイルシステムが、ユーザのホームディレクトリとしてマウントされる。

次に、デスクトップ機上で実行されるプロセスがホームディレクトリ下のファイルをアクセスした場合の、動作について説明する。

- デスクトップ機上でのファイルアクセスは、通常の NFS プロトコルに従って処理される。たとえば、ホームディレクトリ下にある .cshrc をアクセスする場合について、その動作をみると、まずホームディレクトリの filehandle を元に、サーバに Lookup(.cshrc) リクエストが携帯機上の NFS サーバに送出される。lookup リクエストを受け取った携帯機上の NFS サーバは、ホームディレクトリ内を検索し、.cshrc に対する filehandle とファイル属性を用意する。ファイル属性は、NFS Adapter により現在のドメインに対応する属性値に変換された後にクライアントに返信される。デスクトップ機上で、得られたファイル属性を元にアクセス権チェックを行い、アクセス可能であれば引き続き、NFS.READ リクエストにより、ファイルを読む。

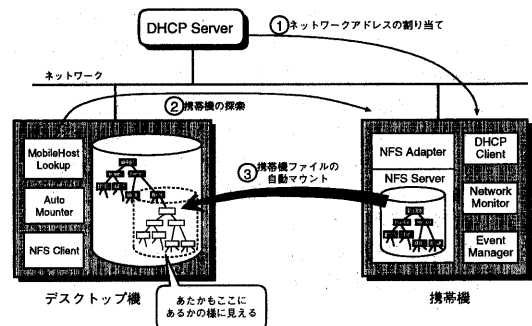


図 4: 携帯ファイルサーバの接続からマウントまでの動作

4 実装

本節では、FreeBSD が稼働する携帯機 (東芝 Libretto) に実装した携帯ファイルサーバのプロトタイプについて説明する。

4.1 NFS Adapter

前節で述べたように、NFS Adapter は、デスクトップ機上の NFS Client と携帯機上の NFS Server の通信を仲介し、ファイル属性 (ユーザ ID /グループ ID/パーミッション) を変換する。FreeBSD では、NFS Server はカーネル内に実装されておりパケット受信からファイルサービスに至るまですべてカーネル内で処理される。このため、NFS Adapter もカーネル内に実装した。

NFS Server procedure の中で、ファイル属性を扱うのは、getattr, setattr, lookup, write, create, symlink, mkdir である。NFS Adapter はこれらの要求パケットとその応答パケットに対してファイル属性の変換を行う。

NFS Adapter で変換するファイル属性の変換情報を与える EventManger はユーザプロセスとして実装するためには、ユーザプロセス側からカーネル内の NFS Adapter に変換情報を設定するインタフェースが必要となる。

ここでは、4.4BSD で導入されたカーネル内のデータアクセスの標準インタフェースである sysctl を拡張し、変換情報設定用 MIB (Management Information Base) を用意した。これにより、root 権限をもったユーザプロセスが、sysctl インタフェースを用いて、NFS Adapter の変換テーブルを設定する。

4.2 Network Monitor

Network Monitor は、携帯機がネットワークから切り離されたかどうかを検出するための情報収集と EventManager への通知を行う。

携帯機がネットワークから切り離されたかどうかをハードウェア的に検知する方法が不明であったため、本プロトタイプではカーネル内の通信統計情報から切り離しを推測する方法を採用した。

カーネルが管理する通信統計情報には、ネットワークに送信したパケット数と受信したパケット数の累計がある。Network Monitor は、一定時間間隔ごとにこの送信パケット数と受信パケット数の変化をモニタし、変化が無い場合にネットワークからの切り離しが生じた可能性があるとして、EventManager に通知するものである。

ただし、変化が無いことの原因として、ネットワークからの切り離し以外に、単に通信を行うプロセスがなかった場合もあるため、EventManager は、Network Monitor からの通知を受けて、DHCP Client にネットワークの切り離しや移動が無かったかどうかのチェックを依頼する。

4.3 DHCP Client

DHCP クライアントの動作の概略は、起動時に DHCP サーバから IP アドレスを獲得し、割り当てられた IP アドレスのリース時間が残り少なくなったところで DHCP サーバにリース時間の延長を申請する。

DHCP クライアントはリース期間内にネットワークを移動することを仮定していないため、リース期間の延

長を申請する時刻になるまでは sleep している。このため sleep 中にネットワークを移動してもアドレスの再取得が行われない。

プロトタイプ実装で採用した WIDE 版 DHCP [2] は、ネットワーク移動に伴うアドレスの再取得を考慮し、verify リクエストが実装されている。これは、DHCP Client が獲得した IP アドレスがまだ有効であるかどうかをチェックするもので、ネットワーク移動が生じたと推測される場合に、このリクエストを発行し獲得したアドレスの有効性を調べる。ネットワークから切り離れたあと同じネットワークに再接続した場合には、獲得した IP アドレスは有効であるため、verify は成功する。一方異なるネットワークへ移動した場合には、verify に失敗するため新たなアドレス獲得が行われる。

しかし、この verify 機能はアドレスを確かめる手段に過ぎず、ネットワークからの切り離しや再接続は検出できない。そこで、本システムでは、NetworkMonitor がその機能を担当している。

また、DHCP Client は、割り当てられた IP アドレスが変更された場合に、それを EventManager に通知する機能を持たない。このため、DHCP クライアントにアドレス変更通知機能を付加した。

4.4 Event Manager

EventManager は、NetworkMonitor や DHCP Client から通知されるイベントと、その時点での携帯機の状態とから、次にとるべきアクションを決定し、対応するモジュールに指示を与える。

EventManager が扱うイベントとそれに対応するアクションは以下の通り。

- NetworkMonitor からの切断イベント
DHCP Client に verify リクエストの発行を指示する
- DHCP Client からのアドレス変更イベント
新しいアドレスからネットワークドメインを特定し、そのドメインに対応するファイル属性変換情報を NFS Adaptor に与える
- DHCP Client からの Giveup-verify イベント
携帯機がネットワークから切り離され、スタンバイ状態におかれている場合、DHCP クライアントが発行する verify リクエストに対して、何の返答も得られない。この場合、DHCP クライアントは一定時間待った後、verify を諦め、EventManager に通知する。
この場合、ネットワークへの再接続に備えて、再度 DHCP クライアントに verify を要求する。

4.5 MobileHost Lookup

ユーザがデスクトップ機に login する場合、ユーザのホームディレクトリをマウントするために、携帯機に割り当てられた IP アドレスとホームディレクトリのパスを調べ

なければならない。本システムではデスクトップ機がブロードキャストメッセージを送出して携帯機を探す方法を採用した。

デスクトップ機は、ユーザを一意に特定する情報をブロードキャストメッセージにのせて発信する。ネットワークに接続している複数の携帯機が各々このブロードキャストメッセージの中のユーザ特定情報をチェックし、該当する携帯機のみが、応答メッセージにホームディレクトリのパスを添付して返す。デスクトップ機は、応答メッセージの送信ホストアドレスとメッセージ中のパスにより、ユーザのホームディレクトリをマウントするのに必要な情報を獲得する。

4.6 Auto Mounter(amd)

amd は、オンデマンドにファイルシステムをマウントするためのメカニズムを提供する [4]。

amd は、様々な種類のファイルシステムをマウントできる様に設計されている。NFSでマウントする場合には、NFS用の設定 (`filesystem_type = nfs`) を利用することにより簡単にマウントが行える。ただし、NFS用の設定には予めファイルサーバの IP アドレスが分かっているなければならないが、携帯ファイルサーバの IP アドレスは DHCP により動的に割り当てられるため、NFS用の設定は利用できない。

一方 amd には、標準でサポートされない様々なファイルシステムのマウントのために `program filesystem(filesystem_type = program)` と呼ばれる設定がある。これは、ファイルシステムのマウントに特殊な mount プログラムを利用するためのオプションである。そこで MobileHost Lookup 機能を mount プログラムに付加した専用のマウントコマンドを用意し、`program filesystem` として登録することにより、携帯機上のファイルシステム (ホームディレクトリ) の自動マウントを実現した。

5 関連研究

携帯機用のファイルシステムとしては、Coda filesystem[5] が有名である。Coda は、広域分散ファイルシステム AFS をベースに携帯機用に拡張したファイルシステムで、ネットワーク接続を絶たれた状態でのファイルアクセスを可能にする `disconnected operation` を実現している。

Coda は、ファイルサーバが管理するデータのレプリカ (ネットワーク接続中はキャッシュ) を積極的に活用する方針に基づいている。しかしレプリカを扱うために一貫性維持の問題を抱えている。一方、携帯ファイルサーバでは扱う対象をユーザのホームディレクトリに限定することにより、オリジナルデータの携帯を可能とし、レプリカをつくらないことにより一貫性維持の問題から解放されている。もう一つの大きな違いは、移動中/移動先における携帯機の扱い方であり、Coda ではユーザは携帯機だけを利用することを念頭においているのに対し、携

帯ファイルサーバでは外出先にあるデスクトップ機との連携に主眼を置いているため、外出先での快適な作業環境を提供する。

ただし、携帯ファイルサーバは対象をホームディレクトリに限定しており、プロジェクトで共有するファイル群に対しては適用できない。これらのファイルに対しては、`disconnected operation` をサポートする携帯機用ファイルシステムを併用し、ホームディレクトリとは異なるファイルシステム領域を割り当てるのが妥当であろう。

6 おわりに

本報告では、携帯機上に個人ファイルや各種設定ファイルなどをすべて格納して携帯し、外出先にあるデスクトップ機が利用可能な時には、携帯機をファイルサーバとし、操作性の優れたデスクトップ機を使用する利用形態の提案と、携帯ファイルサーバのプロトタイプについて説明した。

携帯ファイルサーバプロトタイプでは UNIX 環境を対象として、携帯機をネットワークに接続した際に、そのネットワーク内でファイルサーバとして機能させるためのメカニズムを実現した。外出先のネットワークとして、そこにアカウントがあり携帯機の Lookup 機能があれば、携帯機をネットワークに接続しログインするだけで、そのデスクトップでのホームディレクトリが携帯機内のファイルシステムに設定され、オフィスでのユーザ環境と同一の環境が再現される。

我々は、Libretto 上にプロトタイプ実装し試用を通して携帯ファイルサーバの有効性を確認した。一般に携帯機はファイルサーバとして利用するには性能的に不十分であるが、携帯ファイルサーバでは携帯機所有者一人分のホームディレクトリのみを提供するため、実用に耐えうるとの感触を得た。今後、試用を重ねてシステムの備えるべき要件を洗いだし改良を加えていく予定である。

参考文献

- [1] R. Droms, "Dynamic Host Configuration Protocol," Internet Engineering Task Force, RFC 2131, March 1997.
- [2] A.Tominaga, O.Nakamura, F.Teraoka, J.Murai, "Problems and Solutions of DHCP - Experiences with DHCP implementation and Operation -," Proceedings of INET'95, April 1995.
- [3] Sun Microsystems, Inc. "NFS: Network File System Protocol Specification," RFC 1094
- [4] Jan-Simon Pendry, "Amd - The 4.4 BSD Auto-mounter"
- [5] M.Satyanarayanan, "Mobile Information Access," IEEE Personal Communications, Vol.33, No.1, Feb.1996.