6S-01　ディレクトリサービスを用いたOS混在環境における ユーザアカウントの一元管理*

倉前 宏行　　島野 顕継　　木村 彰徳　　松本 政秀　　亀島 鉱二
大阪工業大学 工学部

1. 緒言

情報処理教育の増大および学科独自の教育内容に柔軟に対応するため，著者らはこれまで，Windows NTとPC-UNIXのデュアルブート環境を提供する学科専用のPCクラスタシステムを構築してきた [1–3]．このシステムは，コンピュータリテラシ教育からプログラミング，数値実験などの演習，さらには研究利用や授業時間外のオープン利用まで，のべ1000名もの学生がさまざまな授業演習等で利用する．よって，ユーザはコンピュータに初めて触れる者から，研究のためのシステムソフトウェア開発を行う者まで，スキルやその利用形態はきわめて幅広く，これに対応した管理・運用が求められる．

一方でこのシステムは，学科専用のものであるため，情報処理センターのようなシステム管理・運用のための専門組織を持たない．したがって，本システムでは，あらかじめセキュリティ対策を万全にしておくことや，システムの運用管理をできる限り自動化することが特に必要となる．

本システムを運用管理する教員ユーザの視点から見ると，自動化のポイントは学生ユーザアカウントデータの全学的な一元管理にある．そこで，本学情報センターで一括管理されている学生ユーザ情報と連携させるとともに，ディレクトリサービスNDS (Novell Directory Services) を用いてWindows NTとUNIXとのパスワード情報の一元化を実現した．これにより，本システムにおいてはユーザアカウント管理を一切することなく，情報センターの計算機利用のアカウント1つで大学共通施設と本システムとがシームレスに利用できるようになった．

2. OS混在環境におけるユーザアカウントの一元化

UNIXとWindowsの混在環境におけるユーザの認証は，パスワードの暗号化など認証方式が異なるため，図1に示すように，NIS (Network Information Service) とPDC (Primary Domain Controller) により，それぞれユーザアカウントのデータベースを作成保持しなければならない．

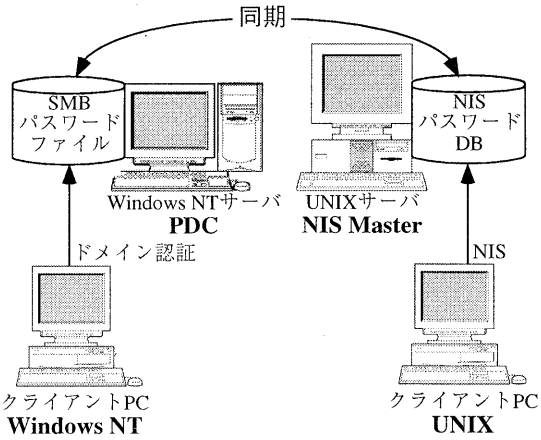

図1 OS混在環境におけるユーザ認証

この2つのユーザアカウントを一元化し，それぞれのOSを同じユーザ名とパスワードで利用できるようにする方が，ユーザにとって便利である．しかし，パスワードの暗号化をはじめとするユーザ認証方法は，セキュリティ上，OSにとっては最も機密を保持しなければならない機構であることから，異なるOSのユーザ認証方法を一元化することは容易ではない．

3. ディレクトリサービスによる一元管理

3.1 NDS (Novell Directory Services)

全学生のユーザアカウントを複数OS上で一元管理するため，NDS Corporate Editionを導入した．NDSはもともとNetWare 4.x用に開発されたディレクトリサービスであるが，今回導入したCorporate EditionからはIPに対応し，図2に示すように，Windows NTにおけるSAM (Security Account Manager) 認証，および，Solaris，Linuxなどの UNIXにおけるPAM (Pluggable Authentication Modules) 認証とNSS (Name Service Switch) フレームを提供することができる．NDSでは，アカウント名，パスワード，ログインスクリプトなどをはじめとする50以上のユーザ情報を一元管理することができる．なお，NDSにおけるパスワードはRSA暗号方式を採用している．

3.2 NDSのレスポンス性能

NDSを導入したLinuxクライアントは，表1に示すように，動作レスポンスが大幅に低下した．同表は，1台のLinuxクライアントでパスワードを入力しGNOMEデスクトップが起動するまでに要した時間，

*User Account Management Unification on Multiple OS using Directory Service, Hiroyuki KURAMAE, Akitsugu SHIMANO, Akinori KIMURA, Masahide MATSUMOTO, Kohji KAMEJIMA, Osaka Institute of Technology, 5-16-1, Omiya, Asahi-ku, Osaka 535-8585, Japan.

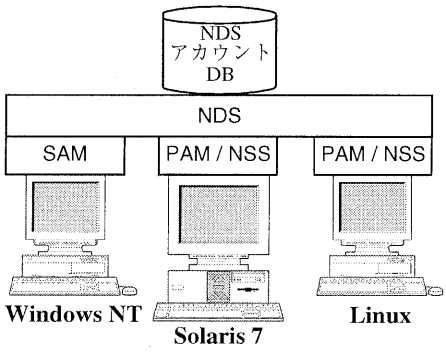

図 2 NDS Corporate Edition の機能

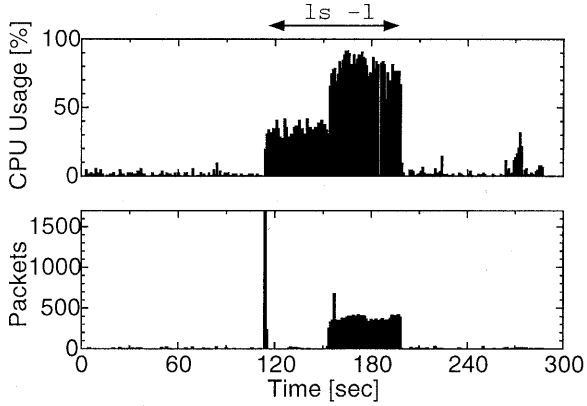

図 3 Linux クライアント動作中のレプリカサーバとネットワークの負荷

`ls -l` コマンドは所有者が全て異なるディレクトリ名を表示させるのに要した時間をそれぞれ実測したものである.

表 1 NDS を導入した Linux 環境のレスポンス性能

	チューニング前	チューニング後
ログイン時間	3 分 15 秒	23 秒
`ls -l` (250 個)	3 分 45 秒	8.49 秒
`ls -l` (935 個)	2 時間 27 分 28 秒	1 分 19 秒
`ls -l` (8000 個)	N/A*	2 時間 33 分 30 秒

*一晩かかっても終了せず

NDS の性能を向上させるため, メーカ/ベンダの協力のもとユーザオブジェクトツリー上の検索経路に関するチューニングを行ったものの, 同表に示すように実用的なレベルへ改善させることはできなかった. こうしたレスポンス低下は, `ls -l` や `ps` コマンドなどを実行した際, ファイル (ディレクトリ) やプロセスの所有者名・グループ名を NSS を経由して NDS へ参照する際に時間を要しているのが原因である.

図 3 は, 1 台の Linux クライアントにおいて 935 個のディレクトリ名を表示する `ls -l` コマンド実行時のレプリカサーバの CPU 利用率とネットワークトラフィックを示している. 同図より, たった 1 台のクライアントからの NSS 参照要求によってもレプリカサーバの CPU 利用率を大きく押し上げ, ネットワークにも大量のパケットが連続的に流れることがわかる.

以上のように, 現時点における NDS の Linux モジュールは実用に耐えがたいものである.

3.3 NDS と NIS を併用した運用

Linux 環境の動作レスポンスを抜本的に改善するため, 演習室内に NIS サーバを設置し, パスワード以外のユーザ情報を NIS によってクライアントへ提供するようにした. すなわち, PAM 認証のみに NDS を利用し, パスワード照合以外の NSS 経由のユーザ情報は NIS を参照するようにした.

この仕組みより, 動作レスポンスは表 2 に示すように十分実用に耐えられるものとなり, レプリカサーバの負荷も下げることができた.

表 2 NIS を併用したシステムのレスポンス性能

	NDS のみ	NIS の併用
ログイン時間	23 秒	18 秒
250 個の `ls -l`	8.49 秒	0.27 秒
953 個の `ls -l`	1 分 19 秒	1.19 秒
8000 個の `ls -l`	2 時間 33 分 30 秒	20.86 秒

4. 結言

Windows NT と UNIX との OS 混在環境において, ユーザアカウントを全学的に一元化するために, ディレクトリサービス NDS を導入した. これにより複数の OS を 1 つのユーザアカウントでシームレスに利用できるようになり, 学生の利便性が向上するとともに, システム管理が省力化された. しかしながら, 現時点での NDS の Linux モジュール機能に大きな問題があることがわかり, これを解決するため NIS を併用したシステムを構築した結果, 実用的なシステムを完成させることができた.

参考文献

[1] 倉前, 島野, 松本, 亀島, ネットワーク型ソフト実験のための PC クラスタシステムの設計と構築, 平成 11 年度情報処理教育研究集会講演論文集, 文部省・東北大学, pp. 139–142, (1999).

[2] 島野, 倉前, 松本, 亀島, ネットワーク型ソフト実験のためのネットワークシステムの設計と構築, 平成 11 年度情報処理教育研究集会講演論文集, 文部省・東北大学, pp. 143–146, (1999).

[3] A. Shimano and H. Kuramae, Design and Construction of Educational Computer System Using Self-mentenance System for Files and User Identification Agent, Proc. of 9th IEEE International Workshop on Robot and Human Interactive Communication, pp. 23–28, (2000).