OUKA
The University of Osaka
Institutional Knowledge Archive

Title	UNIXのネットワーク機能
Author(s)	村田, 正幸
Citation	大阪大学大型計算機センターニュース. 1989, 72, p. 41-53
Version Type	VoR
URL	https://hdl.handle.net/11094/65818
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

UNIXのネットワーク機能

大型計算機センター研究開発部　村田正幸

1　はじめに

前号で紹介したように[1]、阪大大型計算機センターにおいてUNIXワークステーション（SUN-3 5台、EWS-4800 8台）のサービスを行なっており、それらは、ネットワーク（Ethernet）を介して相互に接続されている。本稿では、これらのUNIXシステム上で実現されているネットワーク機能について紹介する。しかしながら、UNIXのネットワーク機能については、すぐれた解説書が多数あるので、ここでは、ネットワーク機能のあらましと、当センターのシステム設定環境に即した形でのネットワーク機能の利用方法に限定して述べていきたい。また、当センターのワークステーションがどのように運用されているかを知っていただくために、利用者には直接関係しないが重要なネットワーク機能（ネットワークファイルシステムを実現するNFS［第3章］とNFSに伴うアクセス権の制御を行なうYP［第4章］）についても紹介する。

2　UNIXの通信プロトコルファミリ

現在、UNIXが提供している通信プロトコルは、対象とする通信媒体、利用目的、過去の開発の経緯などによって、以下の3つのプロトコル群に大別できる。本章では、それらの通信プロトコルの概略と、プロトコルによって実現されているコマンドの使用法について述べる。

2.1　UUCPプロトコルファミリ

UNIX誕生当初から提供されてきたもので、通信媒体として電話回線やシリアル回線の利用を念頭に置き、通信速度は300bpsから9,600bps程度の低速通信を想定している。そのため、処理形態は会話的なものではなく、複数のコマンドを一括して処理するバッチ形式をとっている。特に、電話回線を使用する場合には通信路として電話回線を保持しなければならないので、会話的な処理を行なうにはコスト的にも問題がある。提供される機能としては、ファイル転送、リモートジョブ実行、電子メールの3つがあり、センターのワークステーションも参加しているJUNET[1]でもこの方式に基いてメール交換を行なっている。

UUCPコマンドとしては、uucp（ファイル転送）やuux（リモートジョブ実行）があるが、利用者が実際にこれらのコマンドを実行することは稀であると思われるので省略する。

2.2　DARPAプロトコルファミリ

DARPAプロトコルによって規定されているネットワーク機能をUNIX上で実現したもので、遠隔端末機能を提供するtelnetや、ファイル転送機能を提供するftpがある。このプロ

トコルは、UNIX 独自のものではなく、ARPANET に接続されている計算機間でデータをやりとりするために考えられたものである。センターにおいても、ポートセレクタ経由でターミナルサーバから UNIX ワークステーションにアクセスする場合には、ターミナルサーバ上で telnet のコマンドを用いてワークステーションに login するようになっている。また、ワークステーション上から ACOS-2000 にアクセスを実現する場合にもこの方式がもちいられている（この稿をまとめている段階では、計画中)。

ARPANET での host 名は、IP アドレスと呼ばれるネットワークアドレスで管理されており、それぞれの計算機上で「物理アドレスと IP アドレスの対応」および「IP アドレスと host 名の対応表 (/etc/hosts 、図 1 参照) 」を管理している。ホストを指定するには、IP アドレスを直接指定する方法と IP アドレスに対応した host 名を指定する方法の 2 通りがある。

```
#
# If the yellow pages is running, this file is only consulted when booting
#
# These lines added by the Sun Setup Program from server ccsun01
#
192.9.200.1     ccsun01 sun260 loghost
192.9.200.2     ccsun02 sun50a
192.9.200.3     ccsun03 sun50b
192.9.200.4     ccsun04 sun50c
192.9.200.5     ccsun05 sun50d
#
# NEC-EWS at Computation Center
#
192.9.200.11    ccews01
192.9.200.12    ccews02
192.9.200.13    ccews03
192.9.200.14    ccews04
192.9.200.15    ccews05
#
# NEC_EWS at Toyonaka Data Station
#
192.9.202.1     ccews08
192.9.202.2     ccews09
192.9.202.3     ccews10
```

図 1 . /etc/hosts の例

DARPA プロトコルファミリのコマンドには、telnet と ftp があり、これらは、次に示すように、よく似た内部コマンドを持つ。

1. telnet

   telnet は、現在使用している端末を他のホストの端末として使用するためのコマンドで、

   % telnet [host 名]

により起動され、host 名を指定した場合には自動的にそのホストへのコネクションが設定される。また、host 名を指定しない場合には、プロンプト

```
telnet>
```

が表示され telnet のサブコマンド受け付け状態になる。(センターでは、ポートセレクタを経てターミナルサーバに接続された段階で、このサブコマンド受け付け状態に入る。サブコマンドはかなりの数にのぼり、システムによって提供されているコマンドには少しずつ違いがあるので、ここでは一般的なものだけ紹介する。

- open host 名
  host 名で指定された host とのコネクションを設定する。host 名のかわりに IP アドレスを直接指定することも可能である。また、コネクションを設定した後でも、エスケープキャラクタを入力することにより、サブコマンドを入力することはできるが、コネクションは張られたままなので、セッションを終了する場合には必ず次の close か quit コマンドを実行する必要がある。センターのターミナルサーバ経由でワークステーションを利用する場合には、特にこの点に注意してほしい。なお、ターミナルサーバでは、BREAK によってセッションを中断した後、resume コマンドでセッションの再開ができるようになっている。
- close
  コネクションを解放し、セッションを終了する。
- quit
  コネクションを解放し、telnet コマンドを終了する。
- status
  現在のコネクションが設定されているホストを表示する。
- ?
  ヘルプを表示する。
- set argument value
  telnet の変数に値を設定する。以下のものがある。

  ```
  echo   escape   interrupt   quit   erase   kill   eof
  ```

  例えば、escape は telnet のコネクションをコマンドモードにするためのエスケープ文字が定義されているので、これを自由に変更できる。

図 2 に、センターのターミナルサーバ経由でワークステーションに login する場合の例を示す。

2. ftp
   ftp はファイル転送用のコマンドで、次に紹介する rcp コマンドに比較して、一度に複数のファイルが転送できるという点で便利なコマンドである。コマンドは

   ```
   % ftp [option] [host 名]
   ```

```
*** COMPUTATION CENTER OSAKA UNVERSITY ***
CLASS          S Y S T E M          BPS
  1          ACOS&SX(LEVEL0)       1200
  2          ACOS&SX(LEVEL0)       2400
  3          ACOS&SX(LEVEL0)       9600
  4          ACOS&SX(LEVEL2A)      2400
  5          WORKSTATION           1200
  6          WORKSTATION           9600
ENTER CLASS 5
GO

telnet> open ccsun01
Break-in character is BREAK
Trying to make connection...
[Open]

4.2 BSD UNIX (ccsun01)
login: a60000a
Password:
Last login: Thu Jan 12 18:51:06 from Terminal-Server
Sun UNIX 4.2 Release 3.4EXPORT (GENERIC) #1: Thu Apr 30 09:36:18 PDT 1987
        ***   Messages from Center   ***
You have mail.
ccsun01#/usr1/a60000a/1:hostname
ccsun01
ccsun01#/usr1/a60000a/2:^B
* connection 1 - ccsun01
telnet> status
TTY Port number:   2                  TTY Port speed: 1200
Local address:  192.009.200.010       Default echo mode: Remote

Connection 1: data transfer
Remote  host: ccsun01                 TCP state is ESTABLISHED
Receive window size is    384         Send window size is   4096
Data still to acknowledge   0         Smoothed round trip is 0.3 seconds
Foreign address:  192.009.200.001/23        Local port is  5115
Connection established for 1 mins 24 secs
telnet> resume
[Connection resumed]

ccsun01#/usr1/a60000a/2: logout
[Remote Close]
[Closed]
telnet> status
TTY Port number:   2                  TTY Port speed: 1200
Local address:  192.009.200.010       Default echo mode: Remote
No connections

telnet> quit
```

図２．telnet の使用例

で起動され、host 名を指定した場合には、相手側計算機における login 名とパスワードを聞いてくるので、それに答えることによってコネクションが設定される。telnet と同様に、リソースの保護は相手側計算機のアカウントの有無で行なわれる。また、host 名を指定しない場合には ftp のサブコマンド open によってコネクションを設定する。

ftp のサブコマンドとしては、以下のものがある。

- コネクションの設定／解放
  - open [host 名]
    指定されたホストとのコネクション設定する
  - close (disconnect)
    設定されているコネクションを解放する
  - quit (bye)
    設定されているコネクションを解放し、ftp を終了する
- ファイル転送
  - get [remote-file 名] [local-file 名]
    リモートファイルをローカルファイルにコピーする。local-file 名を指定しない場合にはリモートファイルと同じ名前のファイルになる。また、両方指定しない場合には、それぞれのファイル名を聞いてくる。
  - mget [remote-files 名]
    get に同じ。ただし、同時に複数のファイルを転送できる。
  - put local-file [remote-file]
    ローカルファイルをリモートホストに転送する。remote-file 名を指定しない場合にはローカルファイルと同じ名前になる。
  - mput local-files
    put に同じ。ただし、同時に複数のファイルを転送できる。
- ファイルやディレクトリに関連したコマンド
  - pwd
    リモートホストでのカレントディレクトリを表示する
  - dir
    リモートホストでのファイル情報を表示する
  - ls [remote-directory 名] [local-file 名]
    リモートホストで ls を実行する。ローカルファイルを指定すると、ls の実行結果が指定されたファイルに書込まれる
  - lcd
    ローカルホストのカレントディレクトリを変更する
  - cd
    リモートホストでのカレントディレクトリを変更する
  - mkdir remote-files 名
    リモートホストでディレクトリを作成する

- rmdir directory 名
  リモートホストでディレクトリを削除する
- rename [from-name 名] [to-name 名]
  リモートホストでのファイル名を変更する
- mdelete [remote-files 名]
  リモートホストで指定されたファイルを削除する

この他、ファイル転送時のモード設定用にいくつかのコマンドがあるが、ここでは省略する。図3にftpの使用例を示す。

2.3 UNIX固有プロトコル

UNIX 4.2BSDによって提供されているもので、4.2BSD間でのファイル転送やリモートジョブ実行などの機能を実現している。従って、資源へのアクセス権のチェックなどもUNIXを基本にして考えられている。これらの機能は、4.2BSDだけではなく、SYSTEM Vでもネットワーク拡張機能として提供されている場合があり、EWS-4800でも提供されている。

提供される主な機能としては、仮想端末（コマンド名：rlogin）、遠隔実行（rsh）、ファイル転送（rcp）などがあり、いずれも、UNIX 4.2BSDで提供されるコマンドに"r"をつけたものであると考えてよい。これらのコマンドは、先のDARPAプロトコルによるtelnetやftpコマンドと異なり、（システムの適切な設定のもとで）リモートホストにアクセスするたびにlogin名やパスワードを入力する必要がない。そのためには、リモート側のシステムにおいて

1. /etc/passwdでアカウントが登録されている
2. /etc/hosts.equivでリモートコマンドのサービス要求があった場合にサービスしても良いhost名が登録されている
3. 2.が設定されていない場合には、~/.rhosts（ユーザのホームディレクトリの下のファイル.rhosts）で/etc/hosts.equivと同様の設定が行なわれている

ことが必要になるが、センターの場合には、1.、2.の設定を既に行なっているので利用者側での設定は不要である。

具体的なコマンドの使用方法を、次に示す。

- rlogin ホスト名 [-l ユーザ名]
  "–l ユーザ名"を指定しない場合には、現在のユーザ名でリモートシステムにloginする。指定するとローカルシステムで使っているユーザ名と異なるユーザ名でloginできる。図4に、その使用例を示す。telnetと異なり、同一ユーザ名でloginする場合には、パスワードの入力が不要であることに注意してほしい
- rsh リモートホスト名 [-l ユーザ名] コマンド
  リモートホストでコマンドを実行する。通常の > により出力をリダイレクションするとコマンドの実行結果はローカルシステム上に書き込まれ、一方、">"によりリダイレクションするとリモートホスト上のファイルにコマンドの実行結果が書き込まれる。

```
ccsun01#/usr1/a60000a/41: ftp
ftp> open ccews01
Connected to ccews01.
220 ccews01 FTP server (Version 4.84 Fri Mar 25 13:33:29 JST 1988) ready.
Name (ccews01:a60000a): a60000a
Password (ccews01:a60000a):
331 Password required for a60000a.
230 User a60000a logged in.

ftp> get
(remote-file) rfile1
(local-file) lfile1
200 PORT command successful.
150 Opening data connection for file2 (192.9.200.1,1749) (71214 bytes).
226 Transfer complete.
72863 bytes received in 1.13 seconds (62 Kbytes/s)

ftp> dir
200 PORT command successful.
150 Opening data connection for /bin/ls (192.9.200.1,1750) (0 bytes).
total 65
-rw-r--r--   1 a60000a  other      71214 Jan 12 18:06 rfile1
226 Transfer complete.
91 bytes received in 0.23 seconds (0.34 Kbytes/s)

ftp> mput lfile*
mput lfile1? rfile2
200 PORT command successful.
150 Opening data connection for file2 (192.9.200.1,1752).
226 Transfer complete.
72863 bytes sent in 1.05 seconds (67 Kbytes/s)
mput lfile2? rfile3
200 PORT command successful.
150 Opening data connection for file4 (192.9.200.1,1753).
226 Transfer complete.
1512 bytes sent in 0.01 seconds (78 Kbytes/s)
ftp> dir
200 PORT command successful.
150 Opening data connection for /bin/ls (192.9.200.1,1750) (0 bytes).
total 147
-rw-r--r--   1 a60000a  other      71214 Jan 12 18:06 rfile1
-rw-rw-rw-   1 a60000a  other      71214 Jan 12 18:22 rfile2
-rw-rw-rw-   1 a60000a  other       1476 Jan 12 18:19 rfile3
226 Transfer complete.
169 bytes received in 0.98 seconds (0.45 Kbytes/s)
ftp> quit
221 Goodbye.
ccsun01#/usr1/a60000a/42:
```

図3．ftpの使用例

```
ccsun01#/usr1/a60000a/40: whoami
a60000a
ccsun01#/usr1/a60000a/41: hostname
ccsun01
ccsun01#/usr1/a60000a/42: rlogin ccews01
ccews01> hostname
ccews01
ccews01> logout
Connection closed.

ccsun01#/usr1/a60000a/43: rlogin ccews01 -l a60001a
Password:
ccews01> hostname
ccews01
ccews01> pwd
/usr/user1/a60001a
ccews01> logout
Connection closed.
```

図4．rloginの使用例

- rcp from-file 名 to-file 名
  rcp [−r] file 名1 file 名2 ... directory 名
  ファイル名やディレクトリ名に“リモートホスト名:”というプレフィックスをつけることによりリモートシステム上のファイル名やディレクトリ名を表す。‘−r”オプションは、リカーシブに指定されたディレクトリの下のすべてのファイルとサブディレクトリをコピーする。図5に、rsh、および、rcpの使用例を示す。

```
ccsun01#/usr1/a60000a/46: hostname
ccsun01
ccsun01#/usr1/a60000a/47: rsh ccews01 hostname
ccews01
ccsun01#/usr1/a60000a/48: rsh ccews01 ls
rfile1  rfile2
ccsun01#/usr1/a60000a/49: ls
lfile1
ccsun01#/usr1/a60000a/50: rcp ccews01:~/rfile2 lfile2
ccsun01#/usr1/a60000a/51: ls
lfile1   lfile2
```

図5．rsh、rcpの使用例

これら以外にも以下のコマンドが使える。

- ruptime
  ネットワークに接続されている計算機の稼働状態、負荷情報などが示される。

- rwho
  ネットワークに接続されている計算機に現在 login している利用者の情報が示される。

- talk 相手側ユーザ名@ホスト名
  ネットワークに接続されている他の計算機に現在 login しているユーザと画面上で会話をする。

図6に、出力例を示す。

```
ccsun01#/usr1/a60000a/44: ruptime
ccews01        up      5:12,      2 users,   load 1.13, 1.07, 1.04
ccews02        up      2:06,      0 users,   load 1.00, 1.00, 1.00
ccews03      down      2:12
ccews04      down      2:12
ccsun01        up  3+09:16,      2 users,   load 0.00, 0.00, 0.00
ccsun02      down 71+08:10

ccsun01#/usr1/a60000a/45: rwho
a60000a  ccews01:ttyp0 Jan 12 01:32 :24
a60000a  ccews01:ttyp1 Jan 12 01:57
a60000a  ccsun01:ttyp2 Jan 12 19:03
x60630a  ccsun01:ttyp0 Jan 12 18:36
```

図6．ruptime、rwho の使用例

3 NFS (Network File System)

一般に、ネットワークファイルシステムというと、ネットワークに接続されている計算機間にまたがったファイルシステムのことをさす。UNIX を基本として考える場合には、それぞれの計算機のファイルシステムは木構造になっており、この木構造をネットワーク上でどのように再構成して、ネットワーク上のファイルシステムを構築するかするかが問題となる。そのためにいくつかの方法が考えられてきたが、ここでは、Sun Microsystems 社による NFS (Network File System) を紹介する。NFS は元来 Sun 上で動作するものであるが、現在、他の UNIX 計算機でも広く採用されており、EWS-4800 でも提供されている機能である。

NFS では、まず、ある計算機がファイルシステムの一部（具体的には、あるサブディレクトリ以下のサブツリー）をネットワーク上の他の計算機からアクセスすることを許可する。具体的には、/etc/exports にどのサブツリーをどの計算機に対してアクセスを許すかという対応表を書く。このローカルなファイルシステムの一部をネットワークに対して提供する側を NFS サーバと呼んでいる。一方、他の計算機（NFS クライアント）は、この提供されたサブ

ツリーを自分のローカルなファイルシステムの一部として接続し、全体で一つのファイルシステムを再構成する（図7参照）。

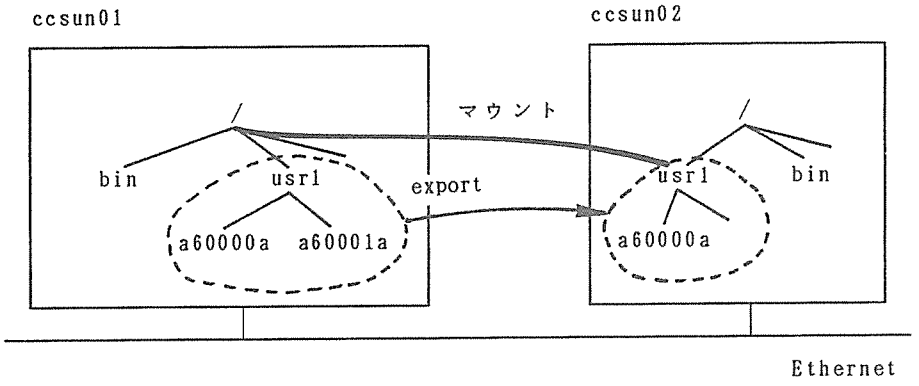

図7．NFS によるネットワークファイルシステムの実現例

これは、リモートマウントと呼ばれているものであるが、これは、単一の UNIX システムで実現されている「マウント」[1] のネットワークへの拡張版である。

リモートマウントにより、クライアント側では、サーバのファイルシステムを仮想的に自分のファイルシステムとしてアクセス可能になる。最初に、クライアントがサーバのファイルシステムをマウントする場合には、「サーバがどの計算機でどこにあるか？（具体的には、ネットワークアドレスはいくつか？）」を意識する必要があるが、いったんマウントした後は、利用者はそのファイルがどの計算機が所有し管理しているものであるかは意識しなくてすみ、その意味でのネットワーク上での位置透過性が実現されている。

センターのワークステーションの場合には、この NFS の機能を利用してユーザディレクトリを一元管理している。具体的には、ccsun01 が NFS サーバとなってユーザディレクトリの実体を持ち、他のワークステーション（ccsun02、ccsun03、ccsun04、ccsun05）は、このディレクトリをリモートマウントしている。従って、利用者はどの Sun ワークステーションに login しても同じディレクトリにアクセスできる。EWS-4800 についても同様に、ccews01、ccews08 がそれぞれユーザディレクトリのサーバとなり、他の計算機（ccews01 に対しては、ccews02、ccews03、ccews04、ccews05、ccews08 に対しては、ccews09、ccews10）がそれぞれクライアントとなってサーバのユーザディレクトリにアクセスしている。

4 YP (Yellow Page)

ネットワークファイルシステムを構築するには、ファイルに対するアクセス権の制御ををどうやって実現するかがひとつのキーポイントになる。NFS では、ファイルの所有権はそのファイルに対するユーザ ID で確認している。すなわち、ネットワークを介してファイルにアクセ

[1] UNIX では、ディスクは複数のパーティションと呼ばれる論理的な区分に分けられており、それぞれが1つのファイルシステムを構成している。システム立ち上げ時にそのうちの1つがルートディレクトリになり、残りのファイルシステムはマウントという操作によってルートファイルシステムに接続され、全体として1つのファイルシステムを再構成する [3]

スしてきた時には、アクセスを要求してきたリモートユーザのIDと（そのリクエストを受けた計算機上の）ローカルファイルの所有者のユーザIDとを比較して、一致していればアクセスを許可するわけである。ところが、UNIXシステムでは、ユーザ名とユーザIDの対応は、それぞれのUNIX計算機上で/etc/password.に書いて管理している。そのため、異なる計算機上で異なるユーザ名に同一のユーザIDを割当てている場合には、全くの他人にアクセスを許してしまうという問題が生じてしまう。また、異なる計算機上で同一のユーザ名が同一のユーザIDで登録されていても、同一のユーザを指すとは限らない場合もある。全く同じことが、グループ名とグループID（これらは、/etc/groupで管理されている）についてもいえる。一つの解決法として、NFSを利用しているすべての計算機上で全く同じの/etc/passwordファイルを保持する方法が考えられる。しかしながら、この方法では、データの一貫性は保たれるが、管理が大変になる。

そこで、Sunでは、これらデータの一貫性を保つための分散型データベースとしてYP（Yellow Page）が考えだされた。YPでは、ネットワーク上で唯一つの計算機がYPマスターサーバとして元データの管理をする。データの更新はすべてこのYPマスターサーバ上で行なう。しかしながら、データを保持する計算機がネットワーク上でただ1台しかない場合にはその計算機にデータ検索要求が集中し応答時間の劣化を招くことになる。そこで、YPではデータのコピーを保持する計算機（YPスレーブサーバ）を複数台置くことができる。マスターサーバで更新されたデータは、一定時間毎にスレーブサーバに送ることによってデータの一貫性を保っている。一方、データを持たない計算機（YPクライアント）上でデータへの検索要求が発生するとその計算機は、ネットワーク上でデータを持っているサーバを探しだし、データ検索を依頼する（図8参照）。

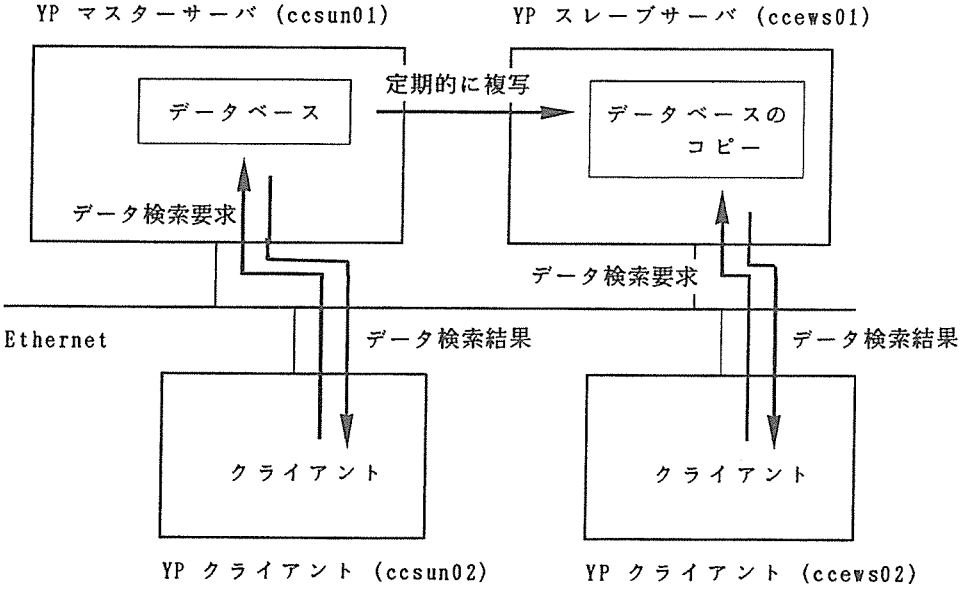

図8．YPによるネットワーク情報の一元管理

当然のことながら、/etc/password をネットワーク上でデータベース化して保持することにより、ユーザのパスワードも一元管理される。従って、ある計算機上でパスワードを変更すると、(YP で結合されている) 他の計算機でも変更後の新しいパスワードで login 可能になる。

また、YP によって管理されるデータは、/etc/password、/etc/group 以外に /etc/hosts、/etc/networks、/etc/services などがある。

センターでは、ccsun01 を YP マスターサーバに、ccews01、ccews08 を YP スレーブサーバに、他のワークステーションはすべて YP クライアントに設定している。スレーブサーバ上でのデータの更新はシステム側で自動的に行なわれているが、利用者が意識しなければならないことが2つある。

- パスワードの変更は、従来の password コマンドではなく、yppasswd コマンドで行なうこと。これにより、ローカルの/etc/password ではなく、YP マスターサーバのデータベースが更新される。

- YP マスターサーバから YP スレーブサーバへデータが転送され、YP スレーブサーバ上でデータが更新されるのは一定時間毎なので、一時的に、ネットワーク上でデータの一貫性が保たれなくなる。現在、データの更新がより頻繁に行なわれると考えられる /etc/password については、1時間に一度、より時間制約の緩いと考えられる他のデータについては、1日に一度、あるいは、二度のデータ更新を行なうように設定している。従って、YP スレーブサーバ上でデータが更新されるまでの時間は、計算機によっては古いデータにアクセスする場合があるので注意を要する (特にパスワード)。

5 参考文献の紹介

[1] は、当センターの UNIX ワークステーションの全体の構成を紹介したものである。[2]–[5] は、UNIX のネットワーク機能を詳述したもので、NFS を含めたネットワークファイルシステムの紹介等が行なわれている。[6]–[8] は、UNIX のネットワーク機能およびそのコマンド群が平易に紹介されている。[9] は、現在連載中のもので、UNIX の通信機能、特にその管理手法について詳しい。[10]–[15] は当センター備え付けのマニュアルのうちネットワークに関連するもので、[10]–[12] は Sun-3 用、[12]–[15] は EWS-4800 用のものである。

参考文献

[1] 村田, "大阪大学大型計算機センターにおける UNIX ワークステーションの運用について、" 大阪大学大型計算機センターニュース, Vol.18, No.3, pp.67-76, 1988 年 11 月.

[2] 村井他, "UNIX のネットワーク機能," 最新 UNIX (bit 臨時増刊) 第 VI 章, 1987 年 5 月, 共立出版.

[3] 村井他, UNIX ワークステーション I <基礎技術編>, アスキー出版局, 1987 年.

[4] 村井他, プロフェッショナルUNIX, アスキー出版局, 1987年.

[5] "コマンド探検隊－ネットワークのコマンドたち(1)－," UNIX MAGAZINE, Vol.3, No.5, pp.38-44, 1988年5月号.

[6] "コマンド探検隊－ネットワークのコマンドたち(2)－," UNIX MAGAZINE, Vol.3, No.6, pp.68-73, 1988年6月号.

[7] "コマンド探検隊－ネットワークファイルシステムとNFS－," UNIX MAGAZINE, Vol.3, No.7, pp.53-70, 1988年7月号.

[8] 斎藤, 山口, "Unix Communications Note," UNIX MAGAZINE 連載中.

[9] "Sun Network Tutorial," 伊藤忠テクノサイエンス株式会社.

[10] "Networking on the Sun Workstation," 800-1324-03, Sun Microsystems, Inc.

[11] "System Administration for the Sun Workstation," 800-1323-03 、Sun Microsystems, Inc.

[12] "やさしいネットワーク," EWS50-1, 日本電気株式会社.

[13] "EWS-UX/V 利用の手引　ネットワーク説明書," EWS52-1, 日本電気株式会社.

[14] "ネットワーク説明書," EWS01-3, 日本電気株式会社.