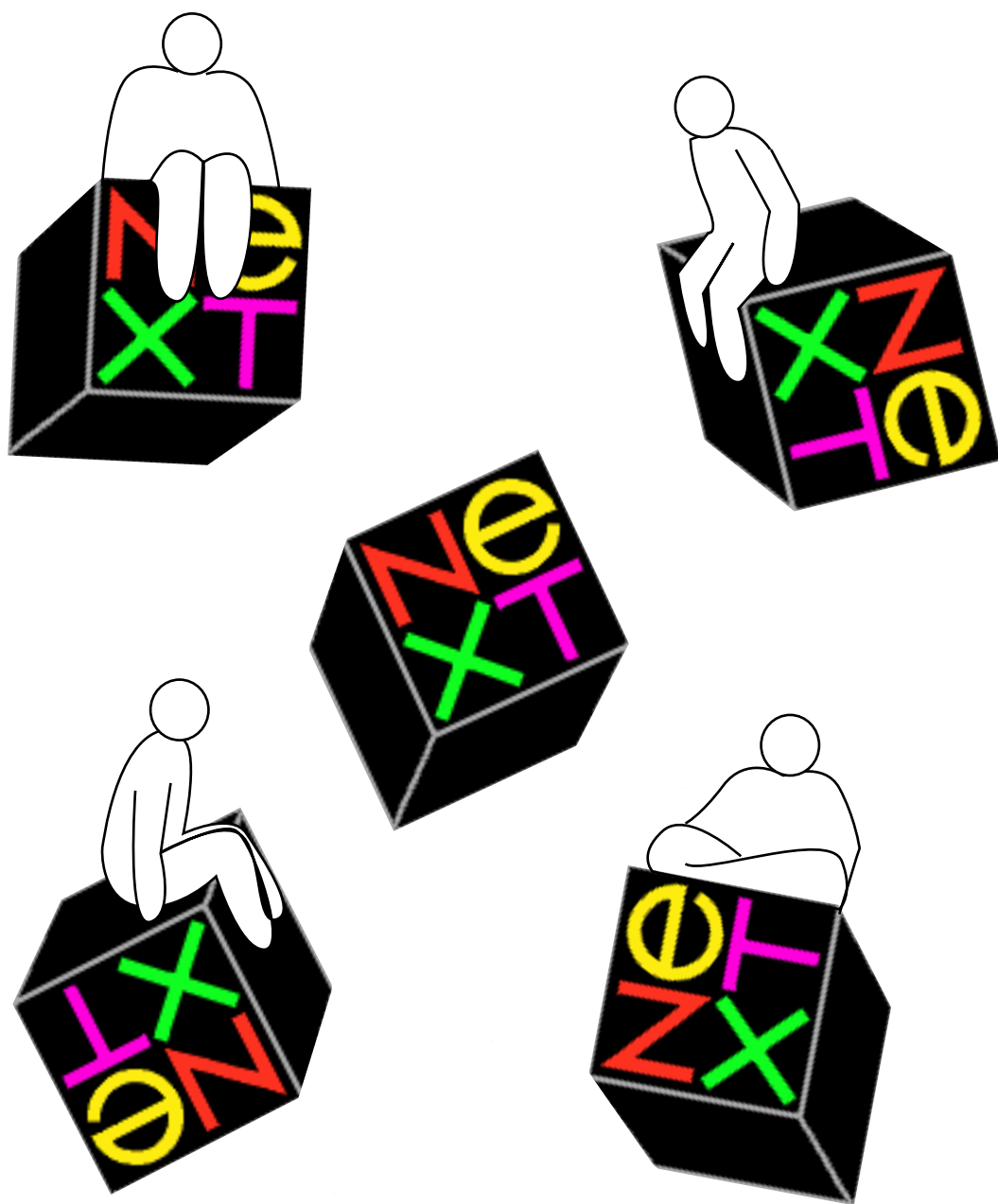


# OPENSTEP システム管理入門 (Ver. 2.1)

小田部 荘司、清永 明  
(九州工業大学 情報工学部 電子情報工学教室)

1997 年 3 月



# 目次

|  |    |
|--|----|
| はじめに   | 1  |
| 凡例および読み方   | 3  |
| 第 1 部 システムの起動、停止                                 | 4  |
| 1 スーパーユーザ (root) になる                             | 5  |
| 1.1 一般ユーザからスーパーユーザに移行したい (su)                    | 5  |
| 2 システムを起動する                                      | 5  |
| 2.1 シングルユーザモードでブートしたい                            | 5  |
| 2.2 システムを再起動したい (reboot)                         | 6  |
| 3 システムを停止する                                      | 6  |
| 3.1 システムを停止させたい (shutdown)                       | 7  |
| 3.2 異常時にシステムを停止させたい                              | 7  |
| 3.2.1 左右の Command キー + ~                         | 7  |
| 3.2.2 右の Command キー + ~                          | 7  |
| 3.2.3 電源スイッチ                                     | 8  |
| 第 2 部 システムの設定                                    | 9  |
| 4 NetInfo の設定をする                                 | 10 |
| 4.1 NetInfo を使いたい                                | 10 |
| 4.2 /NextAdmin の下の管理ツールで NetInfo を設定したい          | 12 |
| 4.3 NetInfoManager.app から NetInfo を設定したい         | 12 |
| 4.4 nidump, niload, niutil コマンドから NetInfo を設定したい | 12 |
| 5 システムの基本設定を行う                                   | 13 |
| 5.1 日付と時刻を設定したい                                  | 13 |
| 5.2 コマンドを自動的に立ち上げたい                              | 14 |
| 5.2.1 決まった時刻に定期的にコマンドを立ち上げたい (cron)              | 14 |
| 5.2.2 起動時に立ち上げたい (rc.local)                      | 15 |
| 5.3 スワップ領域の設定をしたい (swaptab)                      | 15 |
| 5.4 デフォルトデータベースを扱いたい (dread)                     | 16 |
| 6 ネットワークを設定する                                    | 16 |
| 6.1 ホストの IP アドレスを設定したい                           | 17 |
| 6.1.1 直接 /etc/hostconfig ファイルを書き換える              | 17 |
| 6.1.2 HostManager.app を使用する                      | 17 |
| 6.2 マシン名の設定をしたい                                  | 17 |
| 6.3 信頼できるマシンの設定をしたい (hosts.equiv)                | 18 |
| 6.4 ネームサーバの設定をしたい (resolve.conf)                 | 20 |
| 6.5 ネットワークプリンタを設定したい (PrintManager.app)          | 20 |
| 6.6 メールの設定                                       | 22 |

|                      |  |           |
|----------------------|--|-----------|
| 6.7                  | メールエイリアスを設定したい (aliases)                   | 23        |
| <b>7</b>             | <b>ファイルシステムを設定する</b>                       | <b>24</b> |
| 7.1                  | ハードディスクを初期化したい disktab                     | 24        |
| 7.2                  | ハードディスクをマウントしたい fstab                      | 25        |
| <b>8</b>             | <b>NFS を設定する</b>                           | <b>27</b> |
| 8.1                  | nidump, niload を用いて NFS の設定をしたい            | 28        |
| 8.1.1                | サーバ側の設定                                    | 28        |
| 8.1.2                | クライアント側での設定                                | 29        |
| 8.2                  | NFSManager.app を用いて NFS の設定をしたい            | 29        |
| 8.2.1                | サーバ側の設定                                    | 31        |
| 8.2.2                | クライアント側での設定                                | 32        |
| 8.3                  | ファイルシステムをマウントしたい                           | 33        |
| <b>9</b>             | <b>NIS を設定する</b>                           | <b>35</b> |
| 9.1                  | NIS サーバを設定したい                              | 36        |
| 9.1.1                | ドメイン名の設定                                   | 36        |
| 9.1.2                | NIS データベースのインストール                          | 36        |
| 9.1.3                | その他の設定                                     | 36        |
| 9.2                  | NIS クライアントを設定したい                           | 37        |
| 9.3                  | NIS の運用をしたい                                | 38        |
| 9.3.1                | パスワードの変更をおこないたい                            | 38        |
| <b>10</b>            | <b>ユーザを登録・変更・削除する (UserManager.app)</b>    | <b>39</b> |
| 10.1                 | ユーザアカウントを計画したい                             | 39        |
| 10.1.1               | ローカルアカウントまたはネットワークアカウント                    | 39        |
| 10.1.2               | ユーザ名                                       | 40        |
| 10.1.3               | ユーザ ID                                     | 40        |
| 10.1.4               | ホームディレクトリ                                  | 40        |
| 10.1.5               | ユーザグループ                                    | 41        |
| 10.1.6               | パスワード                                      | 41        |
| 10.2                 | ユーザアカウントの登録をしたい                            | 41        |
| 10.3                 | ユーザアカウントの削除やログインの無効にしたい                    | 44        |
| <b>11</b>            | <b>新しいマシンを追加する</b>                         | <b>44</b> |
| 11.1                 | クライアント機を接続したい                              | 44        |
| 11.2                 | スタンドアロンとして接続したい (SimpleNetworkStarter.app) | 46        |
| <b>12</b>            | <b>サーバ機を立ちあげる</b>                          | <b>46</b> |
| 12.1                 | OPENSTEP をインストールしたい                        | 48        |
| 12.2                 | サーバ機の環境設定をおこないたい                           | 48        |
| <b>第 3 部 システムの管理</b> |  | <b>52</b> |
| <b>13</b>            | <b>システムの状態を管理する</b>                        | <b>53</b> |
| 13.1                 | ホスト名を調べたい (hostname, hostid, domainname)   | 53        |

|           |   |           |
|-----------|---|-----------|
| 13.2      | グループを調べたい (groups)                          | 53        |
| 13.3      | 現在の利用者を表示したい (who)                          | 53        |
| 13.4      | 現在の利用状況を表示したい (w)                           | 54        |
| 13.5      | 個人情報を表示したい (finger)                         | 54        |
| 13.6      | ユーザのログイン情報を見たい (last)                       | 54        |
| 13.7      | プロセスの動作状態を確認したい (ps)                        | 55        |
| 13.8      | プロセスの優先度を変更したい (nice)                       | 55        |
| 13.9      | プロセスを終了させたい (kill)                          | 56        |
| 13.10     | ネットワークに接続されているホストの状態を調べたい (ping)            | 56        |
| 13.11     | マシン名から IP アドレスを知りたい (nslookup)              | 56        |
| <b>14</b> | <b>ネットワークの状態を管理する</b>                       | <b>57</b> |
| 14.1      | ネットワークの状態をチェックしたい (ifconfig)                | 57        |
| 14.2      | ネットワークをテストしたい (ping)                        | 58        |
| 14.3      | ネットワークのトラフィック状態を知りたい (netstat)              | 58        |
| <b>15</b> | <b>ファイルシステムの状態を管理する</b>                     | <b>59</b> |
| 15.1      | ファイルシステムの利用状況を確認したい (df)                    | 59        |
| 15.2      | ファイルシステムの整合性を検査したい (fsck)                   | 60        |
| 15.3      | ファイルの内容を検査したい (file)                        | 60        |
| 15.4      | ハードディスクの管理をしたい                              | 60        |
| 15.5      | ファイルを探したい (find)                            | 61        |
| 15.6      | ファイルの内容を 8 進数でみたい (od)                      | 62        |
| 15.7      | ファイルシステムをバックアップしたい (rdump, rrestore)        | 62        |
| 15.7.1    | テープの操作をしたい (mt)                             | 63        |
| 15.8      | ファイルシステムの自動バックアップをしたい                       | 64        |
| 15.9      | ファイルを移動したい (tar)                            | 64        |
| <b>16</b> | <b>コマンドを知る</b>                              | <b>65</b> |
| 16.1      | コマンドを man により調べたい                           | 65        |
| 16.2      | コマンドを Librarian.app により調べたい                 | 66        |
| <b>17</b> | <b>アプリケーションソフトをインストールする (Installer.app)</b> | <b>67</b> |
|           | <b>おわりに</b>                                 | <b>67</b> |

## 表目次

|   |                                      |    |
|---|--------------------------------------|----|
| 1 | /etc/fstab ファイルのエントリ構成 . . . . .     | 30 |
| 2 | /etc/fstab ファイルのマウントタイプリスト . . . . . | 30 |
| 3 | /etc/fstab ファイルのオプションリスト . . . . .   | 30 |
| 4 | mount コマンドのオプション . . . . .           | 35 |

## 図目次

|    |   |    |
|----|---|----|
| 1  | NetInfo の概念 . . . . .   | 10 |
| 2  | HostManager.app でローカルの情報を見る . . . . .   | 18 |
| 3  | HostManager.app でドメイン及びマシンを指定する。ここではドメインとしてネットワークドメイン‘/’を指定し、そこで登録されている aquarius11 というマシンを指定している。 . . . . . | 19 |
| 4  | HostManager.app でマシンを登録する。Ether アドレスはサーバからホスト名と Internet アドレスを自動的に取得するのに必要。 . . . . .                       | 19 |
| 5  | NetInfo.app でネットワークドメイン(‘/’)の /locations/resolver を見ているところ。 domain, nameserver が設定されている。 . . . . .          | 21 |
| 6  | PrintManager.app による設定 . . . . .  | 22 |
| 7  | NFSManager の画面 . . . . .  | 34 |
| 8  | UserManager の画面 . . . . .   | 42 |
| 9  | SimpleNetworkStarter.app によるスタンドアロンの設定 . . . . .  | 47 |
| 10 | SimpleNetworkStarter.app によるネットワーク情報の設定 . . . . .   | 47 |
| 11 | SNS.app によるサーバの設定 . . . . .   | 50 |
| 12 | SNS.app によるサーバのネットワーク情報の設定 . . . . .  | 50 |
| 13 | Librarian.app の画面 . . . . .   | 67 |

## はじめに

電子情報工学科に NeXT が入ったのは 1993 年のことです。学科の性格として半分は物性系また半分は情報系の研究教育をしているので、NeXT のように内部では UNIX で処理しているが、表面では NEXTSTEP のやさしい GUI が使えるというのはいい選択だったと考えております。実際、WS を使いなれなかった方々が NeXT では自在にいろいろな仕事できております。これは NEXTSTEP の一貫した GUI によるところが大きいものと考えられます。最近では GUI はほとんどの OS で取り込まれており、NEXTSTEP が早い段階で先進的な環境を提供していることがわかります。その後、NEXTSTEP は OPENSTEP としてデビューし直しました。1996 年 12 月には NeXT 社は Apple 社に買収されるというような出来事もありました。

OPENSTEP をいわゆる計算機管理の面からみると BSD 系の UNIX/OS であるので、その傾向が色濃くありますが、NetInfo という強力な管理システムを導入し、各種の設定は管理ツールをやはり GUI で効率良くできるように工夫されております。しかし学科ではいままで導入されている WS などのことから、業界で普及している NIS(Network Information System) を用いた管理も行っております。これは混成システムでは仕方の無いことですが、学科では OPENSTEP を用いながらもこのような特異なところがあります。

この解説の前には「NEXTSTEP 管理入門」なるものを書いておりました。本書はこれをベースに 1997 年の計算機の更新にあたり、NEXTSTEP が OPENSTEP に変わり、また NeXTstation(いわゆる黒 NeXT) から IBM-PC をベースとした計算機に変わって管理体制が若干変わったのを説明するために書いたものです。ですから本書では Intel ベースの OPENSTEP4.1J for Mach の管理について述べています。しかしハードに関わる場所以外は NeXTstation(NEXTSTEP3.3J) でも使えるはずですが。

本書では研究室に配属して、UNIX の管理をしなければならなくなった方々、つまり新しい root のために、OPENSTEP における計算機管理の仕方を解説しています。OPENSTEP のための計算機管理の方法はマニュアル [1] に全て書いてあります。OPENSTEP4.1J では日本語のオンラインマニュアルがあるので、Librarian.app を使って読むこともできます。したがってこれを注意深く読めば完全です。しかし UNIX につきものですが、いろいろな概念を必要とするので一つだけ読んだのではおそらく必要な情報が得られなく、他の参考書を読まなければならないと思われれます。またマニュアルは分厚く全部読むのには時間がかかり、また学科ではどのように設定しているかといった具体例はありません。ここでは学科の例を取り上げながら具体的にどのように設定しているかという話を載せてあります。

具体的に管理者がしなければならないことは次のようなことであり、ここではこれらについてまとめてあります。

- システムの起動、停止
- システムの設定
  - システムの基本
  - ネットワーク
  - NetInfo
  - NFS(Network File System)
  - NIS(Network Information System)

- ユーザの登録、削除
- システムの管理
  - マシン
  - ネットワーク
  - ファイルシステム

システムの設定はあらかじめ計算機委員会やいままでの管理者によってされていることでしょう。ですから、その様子を確認するというのが最初の仕事になります。システムを維持していくためには管理が不可欠です。また異常を見つけた時には対処しなければなりません。

本書を読んで記述の間違いや書いて欲しい事柄があったときには、是非連絡して下さい。次の版の時には反映させたいと思います。連絡先は [kiyonaga@cse.kyutech.ac.jp](mailto:kiyonaga@cse.kyutech.ac.jp) または [otabe@cse.kyutech.ac.jp](mailto:otabe@cse.kyutech.ac.jp) です。

ここで管理の様子や初歩を学んで頂き、さらに興味を持たれていろいろな勉強をして、UNIX 管理者としてのエキスパートになることを期待しております。

1997年2月  
著者代表して 小田部 荘司

## 凡例および読み方

**ファイル名やコマンド名** ファイル名やコマンド名は /etc/passwd のように type writer フォントで表します。

**ファイル** ファイルの中身は

List 1 /etc/fstab

---

---

```
1 #
2 #      DO NOT DELETE THIS FILE, IT IS REQUIRED FOR BOOTING
3 #
4 #      This file contains information used to mount local hard disks.
5 #      Consult your Network and System Administration manual
6 #      for information on adding local disks.  Information on the format
7 #      of entries in this file can also be found in the fstab man page.
8 #
9 /dev/sd0a / 4.3 rw,noquota,noauto 0 1
```

---

---

のように表します。ただし、左の数字は行数を示すもので実際のファイルには書かれておりません。

**ターミナル画面** ターミナルでコマンドを入力し、出力がある様子は、

```
% su
Password: (スーパーユーザもしくは自分のパスワード)
#
```

のように四角で囲みます。

**読み方** 例題のほとんどは aquarius10 のドメインで行われております。ここではサーバ機として aquarius10、クライアント機として aquarius11, aquarius12 があるものと考えて下さい。

索引は設けておりません。目次を有効に使って下さい。本書では大きく分けてつぎのようになっています。

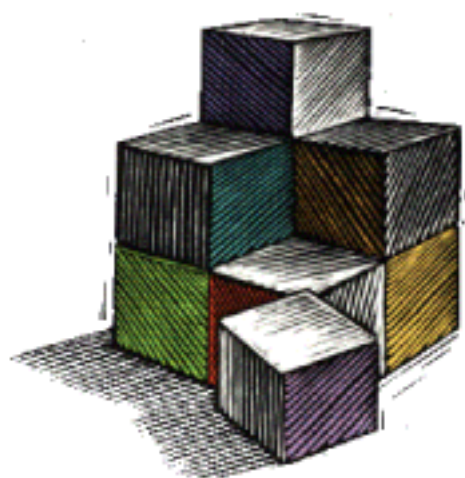
- システムの起動、停止
- システムの設定
- システムの管理

各章の題名は文章になっております。なになにをしたい、なになにをするとといったように簡単に内容を説明するようにしています。

計算機委員会での標準の設定をできるだけ各節の後に示すようにしましたので、参考にしてください。



# 第1部 システムの起動、停止



# OPENSTEP™

## 1 スーパーユーザ (root) になる

システム管理作業を行うためには、スーパーユーザの権限が必要です。スーパーユーザでログインするには、root というアカウントを使います。スーパーユーザの権限を得るには管理者 (ルート) の承諾を得て root のパスワードを教えてもらうか wheel グループに登録してもらう必要があります。スーパーユーザにはさまざまな特権が与えられています。しかし、普段からスーパーユーザ (root) としてログインしていると、ちょっとした不注意からシステムをクラッシュさせてしまうこともあります。したがって、ログイン画面により root でログインするのは /NextAdmin の下の管理ツールを頻繁に使うときのみに行いましょう。通常は Terminal.app から su コマンドを用いて一時的にスーパーユーザになります。また、/NextAdmin の下の管理ツールもルートのパスワードを要求するので一般ユーザから安全に使用できます。

### 1.1 一般ユーザからスーパーユーザに移行したい (su)

一時的にスーパーユーザの権限を得るためには、su コマンドを使用します。/bin/su は su.wheel にリンクされています。これによりスーパーユーザのパスワードを要求されますが、自分のアカウントが wheel グループに登録されていれば自分のパスワードを打ち込むことによって、スーパーユーザになることができます。この仕組みによりルートのパスワードを知らなくても、スーパーユーザの権限を得ることができます。従って、ルートのパスワードは職員が管理し、wheel グループに登録することにより、不要にパスワードが外部に知られないようにすることができます。アカウントの wheel への登録は管理者に行ってもらわなくてはなりません。(本書では特に断らない場合、“%” プロンプトを一般ユーザ、“#” プロンプトをスーパーユーザとして記述しています)

```
% su
Password: スーパーユーザもしくは自分のパスワード
#
```

## 2 システムを起動する

NeXT のシステムを起動させるときは、電源キーを押すだけで起動させることができます。しかし、接続機器がある場合はそれらの電源を先に入れておく必要があります。

### 2.1 シングルユーザモードでブートしたい

シングルユーザモードは、マルチユーザシステムとしての UNIX ではできないことをするためのモードです。シングルユーザモードに移行する時には、他の利用者がいないかどうかを確認し、迅速に、かつ作業は慎重に進めるようにして下さい。

電源を入れた時にメモリのチェック後、ディスクからモニタを読み込みにいきます。

```
boot:
```

表示がでて何も入力しなければ 10 秒後に通常通りのブートを行います。ここで '-v' を入力するとブートの様子を画面上で確認することができます。また '-s' と入力するとシングルユー

ザモードでブートすることができます。さらに `Configure.app` を使った設定に失敗したときなど、とにかく OPENSTEP を立ち上げなければならないときには `'config=default'` と入力すれば、白黒 2 階調 640 × 480 の画面で立ち上がります。

作業後にマルチユーザへの移行は `Control-d` を用います。

具体的なシングルユーザモードで作業する例として、たとえば、`/etc/fstab` に間違った設定をするとマウントができなくなり、マルチモードとして立ち上がりません。そのような時にシングルユーザで立ちあげて復旧します。

OPENSTEP の場合、管理情報は `NetInfo` で管理されています。シングルユーザではまだ `netinfod`(`NetInfo` デーモン) が立ちあがっていないので `nidump`, `niload` が使えません。これらのコマンドをシングルユーザモードでも使うためには

```
# sh /etc/rc&
```

として、`netinfod` を立ちあげます。そうすると `nidump` 等を用いた作業ができます。注意しないといけないのは、このように `rc` を立ちあげた時には作業が終了した時に `Control-d` で抜けてはならず、必ず `reboot` コマンドを用いて立ち上げ直します。

## 2.2 システムを再起動したい (reboot)

UNIX システムは、MS-DOS や Windows といったシステムと異なり、連続稼働で使用されることが多いシステムです。しかし、システムに新しいハードウェアを追加する場合やシステムの設定変更を有効にしたりする場合、また稀にシステムのハードウェアやソフトウェアの障害を復旧させる為に一旦システムを停止し、あらためて起動し直す必要が生じることがあります。その際、UNIX システムではバックグラウンドで常にいろいろなデーモンが動作している為、システムを再起動する場合には定められた手順に従って行う必要があります。

OPENSTEP でシステムの再起動を行いたい場合はログイン画面の再起動ボタンをクリックします。しかし、ログイン中、または何らかの障害の為に再起動ボタンが表示されていない状態でオペレーティングシステムを再起動したい場合には、スーパーユーザ権限で `reboot` コマンドを使用します。これで、ROM モニタの状態にまで戻らなくてもシステムを再起動させることができます。`reboot` コマンドは、`shutdown` コマンドを実行した後に `boot` コマンドを実行することと同様です。

```
# reboot
```

## 3 システムを停止する

クライアント機 (例: `aquarius11`) はログインパネルの電源 off を押せばシステムを安全に停止させることができます。サーバ機は Power キーにより停止しないように設定されています。この設定は `root` でログインして `Preferences.app` の “Login Window Preferences” で電源を落さないようにするように “Disable Power Off” にチェックします。これはターミナルから行うこともできます。`root` になって次のコマンドをいれます。`dwrite` コマンドについては 5.4 章を参照してください。

```
# dwrite loginwindow PowerOffDisabled yes
```

UNIX システムは MS-DOS のような単純なシステムと違い、表面的には誰も使っていないように見ても各種のサービスを実行しています。その為、サーバー機 (例:aquarius10) はめったに停止させることはありません。長期間にわたって利用しない場合や電源設備の法定定期点検によって停電となる日の前日等、どうしても停止させる必要がある時には、利用者がいないか、動作中のデーモンがないかを十分確認したうえで行うようにして下さい。

### 3.1 システムを停止させたい (shutdown)

システムをコマンドにより停止する時には、スーパーユーザになって `shutdown` コマンドを使用します。ただし、このときには必ずルートディレクトリにいないかならなければならないことに注意してください。下記のコマンドを実行することによりシステムの停止が実行されます。

```
# cd /  
# /etc/shutdown -hp now
```

この場合には `halt` した後に電源を落します。 `shutdown` コマンドを実行したとき、他のユーザがそのシステムを使用していた場合は、そのユーザに `shutdown` メッセージが送られます。

### 3.2 異常時にシステムを停止させたい

何らかの理由でシェルコマンドからシステムを終了できなくなった場合、キーストロークによりシャットダウンを行う方法があります。

#### 3.2.1 左右の Command キー + ~

左右両方の Command キー (Alt キーに割り当てられています) を押したまま、数値キーパッドにある ~ キー (Num Lock キーに割り当てられています) を押すことによりモニタにはいり、 `halt` と入力します。この場合にはファイルシステムは `sync` されるので、安全です。 '?' を入力すると簡単なヘルプがでます。ヘルプによると `reboot` をすると `sync` されないことになっています。

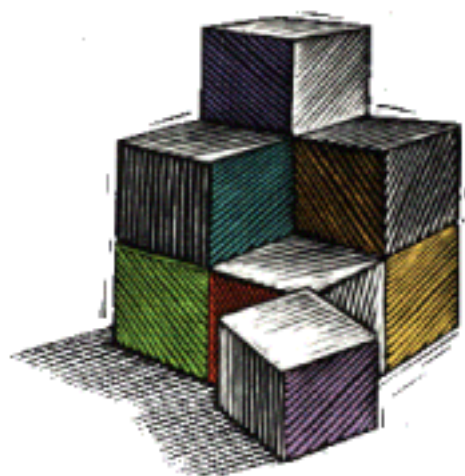
#### 3.2.2 右の Command キー + ~

上の方法でモニタに移行できないときには、右の Command キー + ~ を用います。このときには簡単な表示しかでてきません。 `r` でリブートします。 `h` でシステムを停止します。間違えて押した時には `n` でキャンセルすることができます。

### 3.2.3 電源スイッチ

前節の方法ができないときには、直接電源スイッチを切るしか方法がありません。この方法では sync されないので、次のブート時には fsck をかける必要があります (15.2章参照)。マルチユーザモードであれば、必ず fsck がかけられます。シングルモードでは fsck がかけられないのでそのままファイルの操作をするのは危険です。したがってマルチユーザモードで一度起動して、fsck をかけます。fsck がかけると 5-10 分ほどかかりますが、ほとんどの場合にはファイルの修復は成功します。その後自動的にまた再起動されます。まれにハードディスクが機械的に損傷をうけていると復旧できないこともあります。日頃のディスクのチェックとバックアップ作業が重要です (15.7章参照)。

## 第 2 部 システムの設定



# OPENSTEP™

## 4 NetInfo の設定をする

OPENSTEP ではネットワークで共有するほとんどの設定情報を NetInfo で管理しております。種々の設定の解説に入る前に、ここではこの NetInfo のことについて解説しておきます。

### 4.1 NetInfo を使いたい

NeXT 社で開発された NetInfo は NIS+(Network Information System+)(9章を参照) と同様に階層構造で情報を管理する事ができます。たとえば、通常の UNIX のマシンではユーザの情報は `/etc/passwd` に書かれております。これをネットワークにつながっている全部のマシンに書いておく事は無駄であり、管理が繁雑になります。NetInfo を使えば、サーバーに情報を書きおき、クライアントはその情報を参照する事ができます。情報を書き換える時にはサーバーの情報のみ書き換えれば、クライアントは特に何もする事はありません。

マシンでは最初に自分のローカルの NetInfo を参照して、さらに上位のマシンがあればそのマシンの NetInfo をネットワークを通じて参照します。通常の二階層構造の NetInfo では、サーバーはローカルな情報とは別にネットワークで共有する情報を書きおき、ネットワークに提供します。またクライアントではローカルの情報のみを書きおき、ネットワークから情報を取るよう設定します。

例えば図 1 ではサーバマシン earth はローカルの情報もあり、ネットワークにも情報を提供しております。

二階層構造の NetInfo は NIS と同様に思われるかも知れませんが、管理している情報がきめ細かいので、あらたにクライアントを追加するときには非常に便利になっております。NIS より増えている管理項目で重要なのは `/etc/printcap`, `/etc/fstab`, `/etc/ntpd.conf` などです。したがって、NetInfo が動いている OPENSTEP 群ではクライアントを追加した時にはクライアントの名前と IP アドレスを決めてサーバの `HostManager.app` で登録を行えばネットワークに関するほとんどの設定が終了します。その後必要なディレクトリを必要な場所にリンクを張れば完了です。(11章参照)

NetInfo の情報は `/etc/netinfo/local.nidb` にバイナリの形式で保管されており、さらにネットワークの情報がある時には、`/etc/netinfo/network.nidb` があります。さらに階層構造が増えている時にはそれにとまって `.nidb` が増えます。通常三階層構造で十分大規模のネットワークもまかなえます。

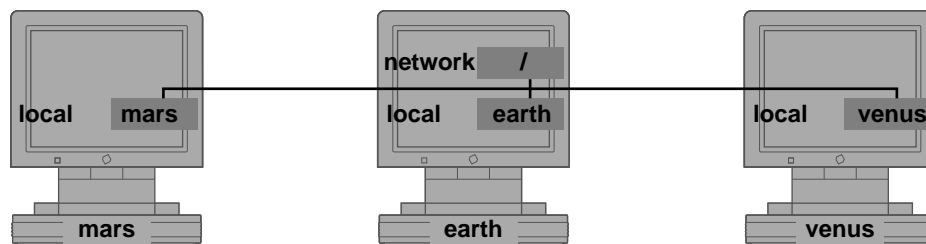


図 1: NetInfo の概念

NetInfo では情報を取り扱っている単位をドメインと読んでいます。この指定はディレクトリ構造のように指定する方法と、タグで指定する方法があります。図1の例ではネットワークドメインは earth が持っています。ネットワークはドメイン表記では '/' であり、タグ表記では 'network@earth' です。各マシンはローカルドメインを持っています。これらは '/earth, /venus, /mars' あるいは 'local@earth, local@venus, local@mars,' です。ローカルドメインはそのマシンであれば '.' で指定することができます。NetInfo の設定の際には必ずドメインを指定しなければなりません。ネットワークドメインとローカルドメインを間違えないように気を付けてください。当然ですが、ネットワークドメインが書き換えられないとネットワークの情報にはなりません。

保管されている情報は `nidump` コマンドを用いて今までの BSD 系の UNIX で使われてきたファイル構造と同じようにして取り出すことができます。たとえばパスワードファイルは通常は `/etc/passwd` にありますが、これと同じ情報が NetInfo にも記録されており次のようにして取り出すことができます。

```
# nidump passwd /aquarius11 (aquarius11 のローカル)
# nidump passwd . (動作させているマシンのローカル)
# nidump passwd / (ネットワークドメイン)
```

ここで、`nidump` コマンドの最後にある部分はドメインを指定しています。注意しなければならないのは、OPENSTEP でも `/etc/passwd` のようにファイルがありますが、このファイルを変更しても NetInfo の情報は変更されません。変更を及ぼさせる時には、`niload` コマンドを用いて NetInfo の情報の更新をする必要があります。しかし情報の更新は通常は `/NextAdmin` の下の管理ツールを使う方が一般的です。

```
# niload passwd . < /etc/passwd
```

NetInfo を設定するには、次のような方法があります。

1. `/NextAdmin` の下の管理ツールを用いる。
2. `NetInfoManager.app` を使う。
3. `nidump`, `niload`, `niutil` コマンドを用いる。

これらの方法により、`/etc/netinfo` の下の `local.nidb` などが設定されていきます。

万が一、NetInfo の管理情報の設定の仕方を間違えて、ブートしなくなってしまった時には、`/usr/template/client/etc` に `/etc` の初期のファイルがすべておいてありますので、シングルユーザで立ち上げて、そこからコピーして復帰します。

```
# cd /etc
# mv netinfo netinfo.old
# cp -r /usr/template/client/etc/netinfo .
```



## 4.2 /NextAdmin の下の管理ツールで NetInfo を設定したい

NetInfo は次で紹介する NetInfoManager.app によりすべての情報を設定する事ができます。しかし GUI を用いているために、操作が膨大になり場合により不便になることが多いです。niload, niutil はかなり NetInfo のことを習熟していないと直接操作は危険です。したがって、普通は /NextAdmin の下の専門の管理ツールにより設定を行います。

**HostManager.app** ホストの情報を管理します。/etc/hosts 相当, /etc/hostconfig を設定することができます。(6.2章参照)

**PrintManager.app** プリンタの設定ができます。/etc/printcap 相当の設定ができます。(6.5章参照)

**NFSManager.app** NFS の設定ができます。/etc/fstab, /etc/exports 相当の設定ができます。(8.2章参照)

**UserManager.app** ユーザの管理ができます。/etc/passwd, /etc/group 相当の設定ができます。(10章参照)

ここで相当と書いているのは、nidump した時の形がそれらのファイルの形になっているということです。

これらの管理ツールはお互いに連携しており、NetInfo でネットワークの情報を管理している時には威力を発揮します。たとえば、/etc/hosts の情報はあちらこちらでホストの情報を使う時に用いられます。

またこれらの管理ツールでは必要に応じて root のパスワードを要求してきます。ここでは su と違い、wheel に属しているユーザでもユーザのパスワードではだめです。

これらのツールは NetInfo で接続されているので、ドメインを指定して他のドメインを変更するのも容易にできます。つまり OPENSTEP の動いているところであればどこからでも情報の更新ができます。

## 4.3 NetInfoManager.app から NetInfo を設定したい

NetInfoManager.app を用いると NetInfo の全ての情報を扱う事ができます。

最初に立ち上げると local の情報をブラウザ形式で表示します。内容を見るにはそのプロパティをダブルクリックします。プロパティの操作にはメニューから操作を選び操作します。このためどんな操作も可能ですから、注意が必要となります。

ネットワークの情報を扱う時にはドメインを指定して表示をします。

## 4.4 nidump, niload, niutil コマンドから NetInfo を設定したい

最後になりましたが、コマンドラインから NetInfo を設定するやり方を説明します。この方法は前にも述べたように NetInfo の仕組みをかなり習熟していないと危険です。

`nidump` コマンドを用いると従来の BSD 系 UNIX で用いられてきたファイルの形式になるので従来通りにエディタを用いて簡単に変更ができます。また変更した内容は `niload` コマンドで読み込ませれば NetInfo の情報も書き換えられます。したがって、ネットワークを介して設定を変更したり、またシェルスクリプトを作ってまとめて操作をしたい時には便利です。

変更を加える時にはもちろん `root` になる必要があります、`su` コマンドで移行することができます。管理ツールのように本当の `root` のパスワードを要求しませんので、`wheel` に属しているユーザは自分のパスワードで変更することができます。

実際に変更できそうなものは、NetInfo の `fstab`, `exports` 程度です。これらも `NFSManager.app` を使う方が間違いがありません。`/etc/passwd`, `/etc/group` は後述する `UserManager.app`(10章) を使う方が便利です。`/etc/printcap` は `PrintManager.app` を使います。`/etc/hosts` は `HostManager.app` を使わないで `niload` を使うと NetInfo どちらの接続の設定をしないので、`niload` は使わない方がいいです。

例えば、次の様にして操作します。

```
# nidump fstab . > /tmp/foo
# vi /tmp/foo (ここで必要な設定をおこなう。)
# niload fstab . < /tmp/foo
```

次に `niutil` コマンドにより、NetInfo の管理情報の全てを扱う事ができます。実際、`NetInfoManager.app` は `niutil` コマンドを通じて NetInfo とやりとりをし、その情報を GUI を用いて表示、操作をするものです。ですから、あまり理解していなくて操作するとブートできなくなるなどの危険な状態に陥る事もあります。`nidump`, `niload` と同様にコマンドラインからの操作ができるというところが便利です。`niutil` についてはここでは詳しく述べないことにします。

## 5 システムの基本設定を行う

UNIX システム上で作業を行う前に、基本的な設定をシステムに対して行っておく必要があります。例えば日付と時刻は、ファイルの作成・更新時刻を正しく維持する為だけではなく、電子メールをはじめとする数多くのツールから参照されることとなります。孤立したシステムならばともかく、ネットワーク上で複数のホストが相互にかかわって運用される環境では、これらが間違っていると、システム全体を安定して運用することが難しくなります。

環境設定の仕方によっては、バックアップの取得など、定期的に深夜無人で行いたいような作業を自動的に実行させることもできます。管理作業の省力化に活用して下さい。

### 5.1 日付と時刻を設定したい

NeXT において日付と時刻を設定する場合は、`Preferences.app` アプリケーションを使用します。また、日付変更を行う権限を `root` だけに限るには下記の作業が必要です。

```
# chmod 755 /usr/lib/Preferences/date
```

この制限を解除する時には次のようにします。

```
# chmod 6755 /usr/lib/Preferences/date
```

OPENSTEP では ntpd を用いて指定したマシンとの時計の同期を取ることができます。計算機委員会では scorpio を教室の ntpd サーバにしています。scorpio はさらに上位のサーバに接続されていて正確に時計を調整しています。OPENSTEP でネットワーク時間を利用するには HostManager.app の「ネットワーク」のメニューの下の「ネットワーク時間コンフィグレーション...」を使って設定をします。設定したら、reboot します。

**計算機委員会の標準設定** 標準設定では次のようにしています。

```
# niutil -destroy /aquarius10 /locations/ntp
# niutil -create /aquarius10 /locations/ntp
# niutil -createprop /aquarius10 /locations/ntp host scorpio
# niutil -destroy / /locations/ntp
# niutil -create / /locations/ntp
# niutil -createprop / /locations/ntp host aquarius10
```

この設定ではサーバ機である aquarius10 では scorpio を ntpd のホストとして利用します。クライアント機は aquairus10 と時間を一致させるようにします。

## 5.2 コマンドを自動的に立ち上げたい

### 5.2.1 決まった時刻に定期的にコマンドを立ち上げたい (cron)

バックグラウンドで自動的に起動するジョブを管理する時には、cron コマンドを使用します。ただし、cron は終了することのないコマンドですので、通常はブート時に一度だけ実行します。cron により定期的に行いたいコマンドを自動的に行うことができます。これは日常のネットワークの監視、ファイルの整理、バックアップ等に使用できます。

cron コマンドに実行させるジョブは、/etc/crontab, /etc/crontab.local ファイルに記述されています。ローカルの情報は /etc/crontab.local に書きます。

crontab, crontab.local ファイルの各行は、空白またはタブで区切られた6つのフィールドに分けられます。先頭から5つ目までの数字もしくは“\*”が書かれているフィールドは時刻の指定で、6番目以降の部分が実行すべきコマンド列となっています。5つの数字の列は、各々先頭から分、日、月、曜日となっていて、カンマで区切って複数の数字を並べることもできるし、“2-6”のように範囲を指定することもできます。

**計算機委員会の標準設定** 標準設定では次のように /etc/crontab.local を設定しています。

List 2 /etc/crontab.local

```
1 0 1 * * *      root    /bin/sh /usr/local/bin/checkmes >/dev/null 2>&1
```

```
2 02***      root    /usr/bin/calendar -
3 02**2      root    /bin/sh /usr/local/bin/backuphome >/dev/null 2>&1
```

---

---

上記の例では一日に一度ディスクのチェックをするシェルスクリプト `checkmes`(15.4章参照) と `calendar` コマンドを実行し、火曜日の朝にホームのバックアップをするプログラム (`backuphome`, 15.8章参照) を動かしています。

### 5.2.2 起動時に立ち上げたい (`rc.local`)

システムを boot して “login” のプロンプトが表示されるまでに、実行される “Run Command” シェルスクリプトです。ローカルで追加して起動時に必要なデーモンの立ち上げに利用します。次の例は `cannaserver` を立ち上げる `rc.local` の内容です。

List 3 /etc/rc.local

---

---

```
1 #
2 # canna server
3 #
4 rm /usr/spool/canna/lock/.CANNALOCK
5 (echo -n 'canna')
6 /usr/local/canna/bin/cannaserver
7 (echo '.')
```

---

---

## 5.3 スワップ領域の設定をしたい (`swaptab`)

UNIX では仮想メモリ空間を用いております。これは RAM 上の主記憶の他にディスク上に記憶空間を用意して、主記憶とのやりとりをすることで仮想的に記憶空間を増やすことができるものです。ディスク上の記憶空間をスワップ領域と言います。

仮想メモリ空間が小さい場合、ある程度以上のプログラムが起動できなくなったり、大規模なソースのコンパイルができなくなったりすることがあります。だからといって最初から非常に大きなスワップ領域を確保するというのは、ディスク容量とのからみもあり、余り現実的ではありません。

NEXTSTEP3.3J からディスク領域が減って 10Mbyte を切ると `Workspace.app` が警告を出すようになりました。あまり頻繁にこの警告が出る時には不要なファイルを整理して、空き容量を増やすべきです。

OPENSTEP では `/private/vm/swapfile` がスワップファイルであり、`/etc/swaptab` に設定があります。

List 4 /etc/swaptab

---

---

```
1 #
2 #      /etc/swaptab
3 #
4 # 16 Meg low water mark
5 # 32 Meg high water mark
6 /private/vm/swapfile    lowat=16777216,hiwat=33554432
```

---

一般にはスワップ領域は主記憶の2倍程度とされております。上記の例では16Mbyteからはじめて、32Mbyteまで使用を許可しております。OPENSTEPの場合、スワップファイルは増える一方で減少しません。また、プロセスを見るとPIDが-1のプロセスが増えていくのがわかります。このような時にはリブートするしか復旧の方法はないようです。

## 5.4 デフォルトデータベースを扱いたい (dread)

OPENSTEPではいくつかのアプリケーションに特有の個人の設定はデフォルトデータベース (defaults database) というものに記録します。たとえば、Edit.appでどのようなフォントを使うようになっているかといったような情報はこのデフォルトデータベースを利用しています。通常のUNIX OSではホームディレクトリの下のドットファイルを利用する場合があります。ファイルは`~/OpenStep/.NeXTdefaults.[DL]`です。呼び出す際には次のようにします。

```
% dread -l
```

書き出しはdwriteを使います。

**計算機委員会の標準設定** サーバの標準ではrootで次のように設定をしています。

```
# dwrite loginwindow HostName aquarius10
# dwrite loginwindow ScreenSaverEnabled Yes
# dwrite loginwindow TimeToDim 300
# dwrite loginwindow PowerOffDisabled yes
# dwrite Clare KatakanaAutoLearn YES
# dwrite Clare TankanAutoLearn YES
```

## 6 ネットワークを設定する

ネットワークを利用する為には、さまざまな設定を行う必要があります。OPENSTEPで重要なファイルは`/etc/fstb`、`/etc/hostconfig`、NetInfoの情報がある`/etc/netinfo/local.nidb`です。このなかで`hostconfig`はホストの基本の設定を行うのに非常に重要です。このファイルに間違った設定をすると他のネットワークに影響を与える可能性がありますので注意が必要です。通常は一度設定を済ませると、ほとんど変更することはありません。

## 6.1 ホストの IP アドレスを設定したい

/etc/hostconfig の設定を変更する方法には直接ファイルを書き換える方法と /NextAdmin/HostManager.app を使用する方法があります。

### 6.1.1 直接 /etc/hostconfig ファイルを書き換える

/etc/hostconfig ファイルには次のような情報が記述されています。

```
HOSTNAME=aquarius11
INETADDR=131.206.34.33
ROUTER=131.206.34.254
IPNETMASK=255.255.255.0
IPBROADCAST=131.206.34.255
NETMASTER=-NO-
YPDOMAIN=-NO-
TIME=-AUTOMATIC-
```

HOSTNAME はそのマシン名、INETADDR はそのマシンの IP アドレス、ROUTER はルーターの IP アドレス、IPNETMASK はネットマスクアドレス、IPBROADCAST はブロードキャストアドレス、NETMASTER はそのマシンがサーバーであるか否かの決定、YPDOMAIN は NIS ドメインの名前、TIME は時間の設定方法を表しています。ブロードキャストアドレスは SunOS4.1.x のころは 131.206.34.0 を使っていましたが、現在では主流である 131.206.34.255 を使っています。

### 6.1.2 HostManager.app を使用する

HostManager.app を使って /etc/hostconfig を変更するにはメニューのローカルを選択します。

1. HostManager.app を起動し、メインメニューから「ローカル」を選択します。
2. 設定を換えます。
3. 例を図に示します。
4. 「設定」をクリックします。

## 6.2 マシン名の設定をしたい

研究室内のマシン名や頻繁につかうマシン名は HostManager.app を用いて登録します。この際には niload は使わない方がいいです。特に NetInfo を用いるマシンは HostManager.app を使うべきです。図 3 にあるようにしてドメイン及びマシンを指定します。図の例ではドメインは '/' を選んでいますので、ネットワークドメインを指定しています。次に図 4 にあるようにマシンを登録します。ここで Ethernet アドレスはブート時にメッセージが出てくるものでそれぞ

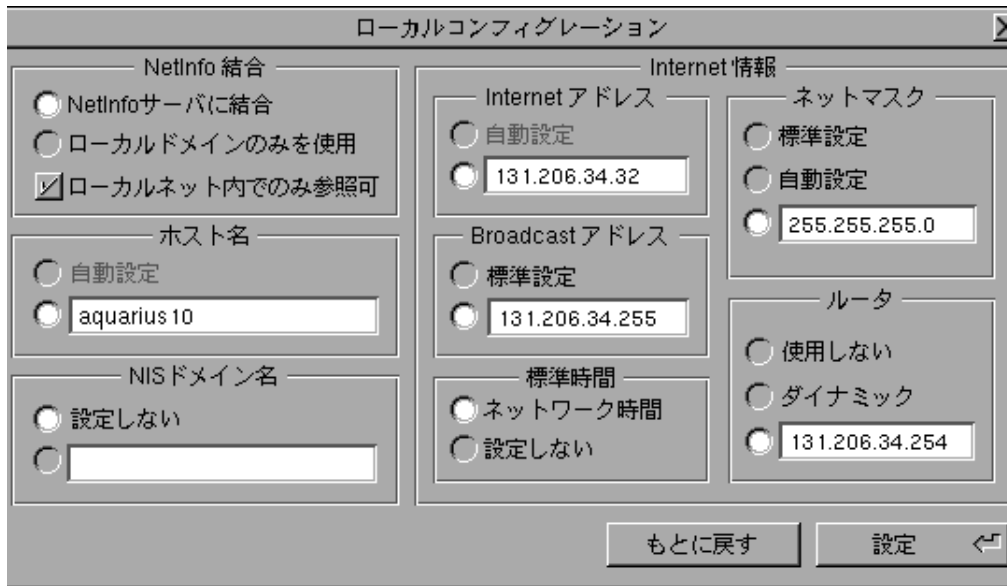


図 2: HostManager.app でローカルの情報を見る

れのマシンに固有です。また `/usr/adm/messages` により確認することもできます。この番号によりブート時にクライアントはサーバから自動的に名前と IP アドレスを得ることができます。

`scorpio, cse, cse.kyutech.ac.jp` を `131.206.43.1` に設定しておく必要があります。これは教官宛に `name@cse` で届くようにするためです。

注意 1 : `131.206.34.03` のように余分な 0 を入れてはいけません。誤動作を起こします。

注意 2 : たくさんのマシンを登録するときにはひとつひとつ「新規作成」をする必要があります。めんどくさいからといってひとつのウィンドウで操作すると一台のマシンを次々と変更しているだけで、一台しか登録されません。

### 6.3 信頼できるマシンの設定をしたい (`hosts.equiv`)

ドメイン内の信頼できるマシンは `hosts` の設定は `/etc/hosts.equiv` で行います。これにより `rlogin` のときにユーザ ID が一致していればパスワードを聞かれずリモートにログインできます。またプリンタをリモートから使いたいときにも設定しておく必要があります。

例 :

List 5 `/etc/hosts.equiv`

```
1 aquarius0a
2 aquarius0b
3 aquarius0c
4 aquarius10
5 aquarius11
6 aquarius12
```

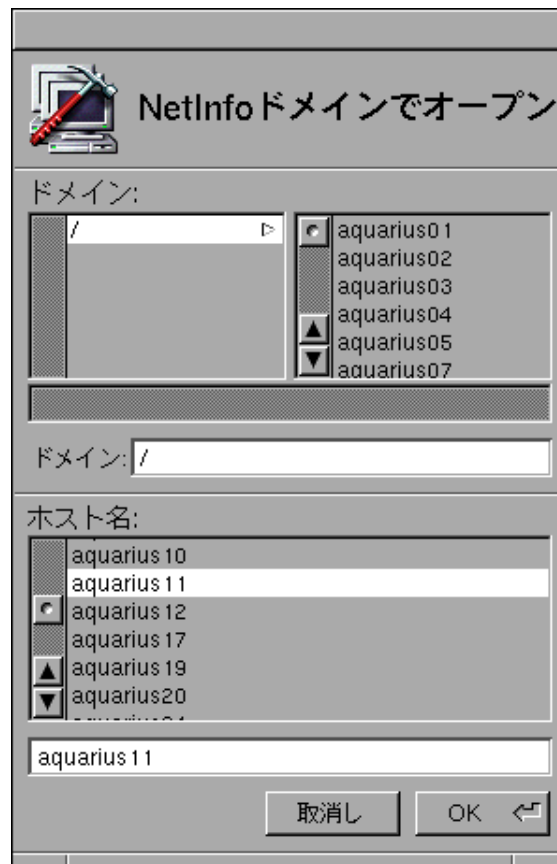


図 3: HostManager.app でドメイン及びマシンを指定する。ここではドメインとしてネットワークドメイン '/' を指定し、そこで登録されている aquarius11 というマシンを指定している。



図 4: HostManager.app でマシンを登録する。Ether アドレスはサーバからホスト名と Internet アドレスを自動的に取得するのに必要。



---

---

## 6.4 ネームサーバの設定をしたい (resolve.conf)

ネームサーバの設定は `/etc/resolve.conf` で行います。電子学科の計算機のネームサーバは `scorpio01(131.206.1.43)` になっています。この設定により、`telnet kyu-cc.cc.kyushu-u.ac.jp` のようになっている名前を実際の IP アドレスに変換してもらえます。またドメインを `cse.kyutech.ac.jp` しているので、`/etc/hosts` に記述されていないマシンはマシン名の後ろに `cse.kyutech.ac.jp` が付き、ネームサーバが IP アドレスに変換します。

List 6 `/etc/resolv.conf`

---

---

```
1 nameserver 131.206.1.43
2 domain cse.kyutech.ac.jp
```

---

---

NEXTSTEP3.3 からは `resolv.conf` の情報を NetInfo に入れることもできるようになりました。

**計算機委員会の標準設定** 標準の設定では NetInfo に直接書き加えています。

```
# niutil -destroy / /locations/resolver
# niutil -create / /locations/resolver
# niutil -createprop / /locations/resolver domain cse.kyutech.ac.jp
# niutil -createprop / /locations/resolver nameserver 131.206.1.43
```

この設定を NetInfo.app で見ると図 5 のようになります。

## 6.5 ネットワークプリンタを設定したい (PrintManager.app)

OPENSTEP でのプリンタの設定は通常は PrintManager.app を使って行います。プリンタをつないでいるマシンで PrintManager.app を動かして設定します。繋げるプリンタはメニューにより与えられているので GUI で簡単に設定できます。キャノンのプリンタについてはドライバも出ているので、安いバブルジェットプリンタを購入し PS カラープリンタとして使うこともできます。ネットワークプリンタとして使う時にはドメインを指定することによりできます。

クライアントのマシン名はサーバの `/etc/hosts.equiv`(6.3章参照) に設定されている必要があります。そうでないと、接続を拒否されます。

次に、OPENSTEP 以外の計算機につながっているプリンタをリモートで使いたいときには次のように `/etc/printcap` を準備して `niload` で直接 NetInfo に書き込みます。

List 7 `/etc/printcap` の内容

---

---

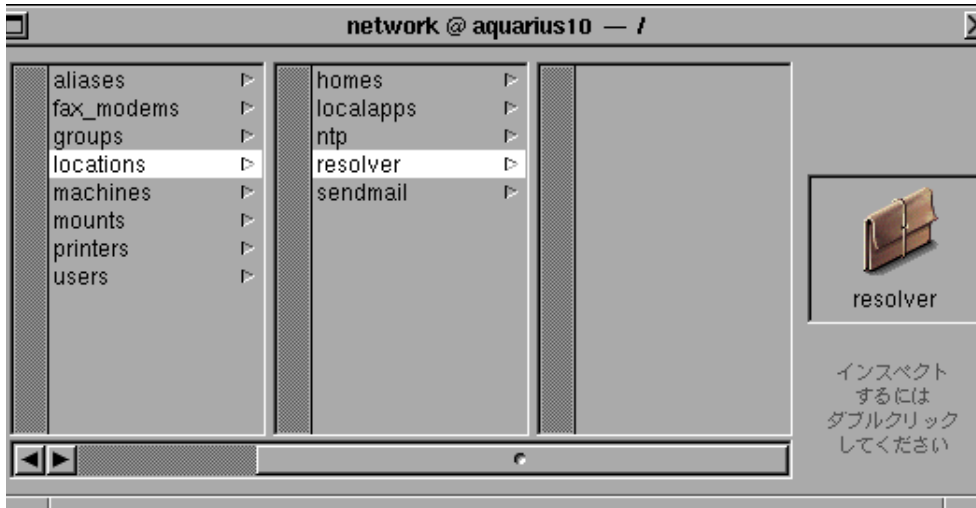


図 5: NetInfo.app でネットワークドメイン ('/') の /locations/resolver を見ているところ。domain, nameserver が設定されている。



図 6: PrintManager.app による設定

```

1 rp: \
2      :rm=aquarius07:rp=lp:lp=:ty=Printer on Sparc Station: \
3      :note=Matsushita Labo.:

```

```
# niload printcap . < /etc/printcap
```

rm はリモートプリンタのマシンを指します。rp はそのマシンでのプリンタの名前です。

**計算機委員会の標準設定** 図6に設定の例を示します。この例では“Canon LSER SHOT LBP-730PS with 2 fonts”を“HP JetDirect”を通じて接続しています。利用範囲は「個人利用」ですが、この理由はネットワーク利用にすると黒 NeXT から利用できるように見えるが、実際には問題が生じることと、フィルタプログラムを起動するので結局ローカルで処理した方が速いと考えられるからです。個々のマシンでの設定がめんどろであれば利用範囲をネットワークに設定するという方法もあります。

## 6.6 メールの設定

メールは非常に重要であり、また設定が難しいので、トラブルの際には経験の積んだ管理者に頼んで変更してもらう方がいいです。計算機委員会では標準の設定をそのまま使っています。

ここでは基本的な仕組みのみ説明します。

メールを受けたり送ったりするのは /usr/lib/sendmail であり、 /etc/rc で自動的に立ち上がります。 /usr/lib/sendmail のパーミッションは

```
-r-sr-x--x  1 root      98304 May  1  1995 /usr/lib/sendmail*
```

のように s ビットが立っている必要があります。そうしないと一般ユーザのメールを受け付けなくなります。現在、 sendmail は NeXT 社から来たものをそのまま使っておりますが、場合によっては自分で用意した sendmail に変えることもできます。

メールの変換規則を書いているのが sendmail.cf です。位置はデフォルトでは /etc/sendmail/sendmail.cf です。このディレクトリには 3 つの sendmail.cf が準備されており、 sendmail.cf にリンクされています。これは NetInfo の情報によって変えることもできます。

```
% niutil -read / /locations/sendmail
name: sendmail
mailhost: aquarius10
sendmail.cf: /etc/sendmail/sendmail.sharedsubsidiary.cf
```

上の例ではネットワークでは /etc/sendmail/sendmail.sharedsubsidiary.cf を利用することとしています。 sendmail.cf を書き換えるのはかなり知識が必要ですし、間違えると動かなくなるので注意が必要です。

メールサーバでは /etc/sendmail/sendmail.mailhost.cf を /etc/sendmail/sendmail.cf にリンクします。あるいは NetInfo に書き込みます。

クライアントでは、 mailhost が hosts として登録されていることが必要です。

```
% nidump hosts / | grep mail
131.206.34.32  aquarius10 mailhost
```

次に /etc/sendmail/sendmail.sharedsubsidiary.cf を /etc/sendmail/sendmail.cf にリンクします。あるいは NetInfo に書き込みます。

## 6.7 メールエイリアスを設定したい (aliases)

以前のシステムでは学科ではどのマシンからも To: フィールドに kiyonaga を書けば kiyonga@zodiac20.cse.kyutech.ac.jp に届くように設定されておりました。これはメールのエイリアス (別名) を利用しています。つまりサーバには kiyonaga は kiyonaga@cse と別名登録されました。しかし教官の転出や転入が多く、別名登録されている名前は UserManager.app で登録することができず不便だったので、今回からは限られた別名登録しかしていません。教官宛は name@cse のように書いてください。

OPENSTEP の場合にはエイリアスは /etc/sendmail/aliases に書き込みます。計算機委

員会では次のようにシェルスクリプトを準備しています。

List 8 /etc/sendmail/aliases.update

---

---

```
1 #!/bin/sh
2
3 cat aliases.local aliases.cse > aliases
4 niload -d aliases . < aliases
5 newaliases
6 #/etc/pushinfo
```

---

---

aliases を直接書き換えずに、ローカル情報は aliases.local に書きます。書き換えたら aliases.update を実行します。aliases.local と aliases.cse をあわせて aliases に書き込み、これを NetInfo に書き込みます。newaliases によりエイリアスのデータベースを更新して、そして NIS のデータベースを更新します。最後に NIS サーバで /etc/pushinfo を実行して aliases はクライアントにも配布されます。

## 7 ファイルシステムを設定する

UNIX OS のファイルシステムはルートディレクトリ (/) にすべてがツリー構造で見えるようになっており、ユーザは実際のファイルがどここのデバイスにあるか気にしなくてもいいようになっています。ファイルを格納しているデバイスをファイルシステムに繋げる作業をマウントと呼びます。マウントするファイルシステムとしてはハードディスク、CD-ROM、フロッピー、またネットワークに繋がっているファイルシステムなどがあります。ネットワークにあるファイルシステムをマウントする方法は次章の NFS の章で詳しく述べます。ここでは主にハードディスクをマウントしてファイルシステムを設定する方法について述べます。

### 7.1 ハードディスクを初期化したい disktab

OPENSTEP では次節で述べる /etc/fstab に書かれていないデバイスは自動的にルートディレクトリ (/) にマウントしようとします。そのときに初期化されていないときにはパネルが現れて初期化することができます。

しかし 2GByte を越えるようなファイルシステムはうまく初期化できないことがあります。このときには /etc/disktab にディスクの記述を行い、マニュアルで初期化を行います。

1. scsimodes コマンドを用いてパラメータを得ます。

```
# scsimodes /dev/rsd1a
SCSI information for /dev/rsd1a
Drive type: SEAGATE ST32151N 053000
512 bytes per sector
126 sectors per track
```

```
8 tracks per cylinder
4177 cylinder per volume (including spare cylinders)
10 spare sectors per cylinder
16 alternate tracks per volume
4197404 usable sectors on volume
```

2. この情報をもとにして /etc/disktab に以下の記述を加えます。

List 9 /etc/disktab の一部

```
1 ST32151N|st32151n:\\
2 :ty=fixed_rw_scsi:nc#4177:nt#8:ns#126:ss#512:rm#7200:\\
3 :fp#320:bp#0:ng#0:gs#0:ga#0:ao#0:\\
4 :os=mach_kernel:z0#64:z1#192:hn=localhost:ro=a:\\
5 :pa#0:sa#4194304:ba#8192:fa#1024:ca#8:da#4096:ra#10:oa=time:\\
6 :ia:ta=4.3BSD:
7
```

OPENSTEP の場合、1 パーティション = 2GB(2,147,483,648byte) 以下でなければならないために、sa#(パーティションサイズ) の値は、4194304(=2,147,483,648byte/512byte) となります。

3. disk コマンドで初期化を行います。

```
# disk -t st32151n -i /dev/rsd1a
```

4. 最後に実際にマウントして確認しておきます。

```
# mount /dev/sd1a /Disk
# df
Filesystem            kbytes    used  avail capacity  Mounted on
/private/vm/swapfile 1820988  392200 1246689    24%  /private/vm/swapfile.front
/dev/hd0a             1820988  392200 1246689    24%  /
/dev/sd1a             2026271     9 1823634     0%  /Disk
```

## 7.2 ハードディスクをマウントしたいfstab

いわゆる IBM-PC 互換機では Windows 95, Windows NT, OS/2, FreeBSD, Linux, OPENSTEP, NEXTSTEP 等様々な OS を動かすことができます。そこでハードディスクを OS ごとのパーティションに分ける作業の後、それぞれの OS をそれぞれのパーティションに入れて適当なブートマネジャーを使っていくつかの OS を持った PC とすることができます。

一般に UNIX OS では OS ごとのパーティションの中でさらにいくつかのパーティションに分けてファイルシステムを管理します。その理由は、スワップ領域は特別なフォーマットを行っ

て効率を上げたり、バックアップする領域ごとに分けるなどがあります。しかし OPENSTEP では基本的には一つのパーティションにすべてのファイルシステムを入れるようになっています。スワップも特別なファイルとして管理されています。

ブート時にマウントしたいデバイスは `/etc/fstab` に記述しておきます。この記述は非常に重要で間違えるとファイルシステムが正しくマウントされないためにブートできなくなることもあります。

ここで主なディレクトリについて述べておきます。

`/LocalLibrary` NeXT 社が提供しないライブラリ群、`/home/local/LocalLibrary` からのリンク

`/LocalApps` NeXT 社が提供しないアプリケーション群、`/home/local/LocalApps` からのリンク

`/Net` ネットワークのファイルシステムのマウント先

`/NextApps` NeXT 社の提供するアプリケーション群

`/NextAdmin` 管理用アプリケーション群

`/NextDepeloper` 開発者用のツールなど

`/NextLibrary` NeXT 社の提供するライブラリ群

`/bin` 基本的なバイナリ

`/dev` デバイスドライバ

`/etc` 設定ファイル

`/me` me のホームディレクトリ

`/tmp` 一時的なファイルを置くディレクトリ

`/usr/adm` 管理情報が置かれるディレクトリ

`/usr/local` `/home/local` からのリンク

`/usr/spool` 一時的にファイルを保管するディレクトリ

`/home/aquarius10` ユーザのホームディレクトリ

`/home/local` ユーザが準備したアプリケーションなど

これらの中で `/home` 以外は OPENSTEP の CD-ROM からインストールすることができます。

計算機委員会の標準設定 標準の設定では次のようになっています。

List 10 /etc/fstab

---

```
1 #
2 #      DO NOT DELETE THIS FILE, IT IS REQUIRED FOR BOOTING
3 #
4 #      This file contains information used to mount local hard disks.
5 #      Consult your Network and System Administration manual
6 #      for information on adding local disks.  Information on the format
7 #      of entries in this file can also be found in the fstab man page.
8 #
9 /dev/hd0a / 4.3 rw,noquota,noauto 0 1
10 /dev/sd1a /home 4.3 rw,noquota 0 2
11 /dev/sd2a /home/backup 4.3 rw,noquota 0 2
```

---

/dev/hd0a は内蔵の EIDE であり、これをルートディレクトリにマウントしています。  
/dev/sd1a は外部の SCSI ハードディスクであり、これを /home にマウントしています。また  
もう一つの SCSI ハードディスクは /home/backup にマウントしています。

この方法を行うと NeXT 社が提供したものはすべてルートディレクトリ (/) にまとめ、  
自分たちで準備したものはすべて /home にまとめておくことができます。 /usr/local,  
/LocalApps, /LocalLibrary については /home/local からリンクを張ることにします。次の  
章で述べる NFS ではクライアント機はサーバ機の /home/aquarius10 と /home/local を自分  
のファイルシステムにマウントする方法をとっています。こうすることによりネットワーク上の  
すべてのマシンに同じソフトをインストールする必要がなくなります。

またこの方法であるとディスクのバックアップが楽になります。ディスクにはクラッシュが  
つきもので、日頃からのバックアップが重要です。サーバ機のディスクは特別なことがない限り  
常に起動し続け熱を持つので、空冷がきちんとできるようにファンの口がふさがれていない場  
所に設置し、また夏場に空調が切れて暑くなる南側の部屋に置くことは避けます。上の例では  
/dev/hd0a は一度設定が済んでしまうとファイルはほとんど変化がありませんし、クラッシュし  
ても再インストールすれば復旧は困難ではありません。したがってこの領域の rdump コマンド  
を用いたバックアップは頻度は少なくてもいいです。(15.7章参照)

一方 /dev/sd1a はユーザのホームがありますので、毎日のように内容が変化しますし、非常  
に重要なファイルがあるのでバックアップが欠かせません。 /dev/sd2a が準備されているのは  
/home/aquarius10, /home/local のバックアップをとるため、 cron を利用して週に一度く  
らい自動でバックアップをするようにすると便利です(5.2.1章参照)。もしディスクがクラッシュ  
しても復旧が楽になります。

## 8 NFS を設定する

NFS(Network File System) はネットワークを通じて他のシステムのファイルシステムを使  
うことができる便利な機能です。ここでは NFS の使いかたについて説明をします。



NFS を設定するときには、サーバ側とクライアント側の双方で設定を行う必要があります。ここで説明する設定作業は、NFS を使用するマシン間でネットワークの環境が整い、相互に `rlogin` などの LAN を使うコマンドが使える前提になっています。

通常の UNIX OS では次のような方法をとります。

1. サーバ側でエクスポートの設定 (`/etc/export`)
2. サーバ側でファイルシステムをエクスポート (`exportfs`)
3. クライアント側で NFS マウントのための設定 (`/etc/fstab`)
4. クライアント側でマウントポイントの作成 (`mkdir`)
5. クライアント側で NFS マウントの実行 (`mount`)

OPENSTEP では `NetInfo` で管理されているので、`NetInfo` を書き換えてそれぞれのコマンドを実行することになります。しかし通常は `NFSManager.app` を用いる方がはるかに便利です。

ここでは `nidump`, `niload` を使って設定する場合と `NFSManager.app` を使う場合を説明します。`nidump`, `niload` の場合にはリモートで入って設定をすることができます。`NFSManager.app` の場合にはドメインを指定することによりリモートで設定をすることができます。

## 8.1 `nidump`, `niload` を用いて NFS の設定をしたい

### 8.1.1 サーバ側の設定

サーバ側では、クライアントが資源をマウントするのを許可します。これをエクスポート (`export`) と言います。この手順を次に示します。

1. サーバのマシンにログインし、スーパーユーザになります。
2. `/etc/exports` 相当のファイルを作成します。たとえば `/home` を `aquarius13` というクライアントマシンからアクセスできるようにする場合は、次のように編集します。OPENSTEP の場合には `/etc/exports` は形はありますが、実際には参照されませんので `nidump`, `niload` を使って適当なファイルを用いて編集します。

```
# cd /tmp (安全な directory に移動する)
# nidump exports . > exports
# vi exports (ここで編集する)
    /home      -access=aquarius13 を追加
# niload exports . < exports
```

3. `exportfs` コマンドを実行します。

```
# /usr/etc/exportfs -a
```

-a は NetInfo 中の情報を有効にするオプションです。

### 8.1.2 クライアント側での設定

NeXT の場合、`/etc/fstab` はブート時にローカルのハードディスクをマウントするための情報が記述されています。なぜ NetInfo に書けないのかというと、ブートの時には NetInfo は使えないからです。このファイルにはそれ以外のマウントの情報は書かず、NetInfo に書きます。`/etc/fstab` はブート時に非常に重要なファイルですので、書き間違えないように注意します。通常 NFS の情報は NetInfo で管理されています。

NetInfo の都合のいいところはネットワークドメインに `fstab` の情報を書き込めば、ネットワークにつながっているすべてのマシンに対して有効になることです。通常 UNIX では一台一台の `/etc/fstab` に書き込む必要があります。

1. クライアントのマシンにログインし、スーパーユーザになります。
2. `nidump` でファイルを作り、その最後に次のようなエントリを追加します。そして `niload` で NetInfo に書き込みます。

```
# cd /tmp (安全な directory に移動、/etc では fstab を書き換える  
           恐れがあるので注意 !!)  
# nidump fstab / > fstab.netinfo (ネットワークドメインを対象)  
# vi fstab.netinfo  
   aquarius10:/home /home nfs rw 0 0 を追加  
# niload fstab / <fstab.netinfo
```

NFS サーバーのディレクトリ名とマウントポイントのディレクトリ名は、必ず絶対パス名で指定します。タイプフィールドは “nfs” となります。オプションはカンマ (、) で区切って複数指定できます。nfs の場合、ダンプ周期とパスフィールドはどちらも 0 にします。`/etc/fstab` ファイルの各行は空行で区切られた 6 つのフィールドから構成されています。形式を表にまとめます。

3. `mount` コマンドを実行します。

```
# mount -a
```

ここで、`-a` は `fstab` の情報にしたがってマウントを行うことを意味します。

## 8.2 NFSManager.app を用いて NFS の設定をしたい

ここでは `/NextAdmin` の下の管理ツールの一つの `NFSManager.app` を使って NFS を設定するやりかたについて述べます。GUI を使っているのでわかりやすいですが、作業量が多くなる可能性もあります。また root のパスワードを要求されます。

|         |  |
|---------|--|
| フィールド 1 | マウントされるでデバイスの論理ディスク名。<br>NFS ではサーバーとなるマシン名とディレクトリ名                             |
| フィールド 2 | マウントするディレクトリ名 (マウントポイント)   |
| フィールド 3 | マウントタイプの指定 (次図参照)  |
| フィールド 4 | マウントのオプション (次図参照)。カンマにより複数指定   |
| フィールド 5 | マウントとは直接関係ない。dump コマンドで使用  |
| フィールド 6 | マウントとは直接関係ない。fsck と df コマンドによって使われるフィールドで、立ち上げ時に fsck をかける順番、df コマンドで表示する順番を示す |

表 1: /etc/fstab ファイルのエントリ構成

|      |                   |
|------|-------------------|
| 4.2  | 通常のローカルなマウントを指定   |
| nfs  | NFS のリモートマウントを指定  |
| swap | mount コマンドでは使われない |

表 2: /etc/fstab ファイルのマウントタイプリスト

|         |                              |
|---------|------------------------------|
| quota   | クォータをかけるファイルシステム             |
| noquota | クォータをかけないファイルシステム            |
| rw      | リード・ライトできるファイルシステム           |
| ro      | 読み出し専用でマウントする                |
| suid    | SetUID ビットプログラムの実行を許す        |
| nosuid  | SetUID ビットプログラムの実行を許さない      |
| grpuid  | ファイル作成 BSD 方式でグループ ID をセットする |
| noauto  | 自動マウントしない                    |

表 3: /etc/fstab ファイルのオプションリスト

### 8.2.1 サーバ側の設定

NFS サーバをコンフィグレーションするには、サーバに対して共有するディスクの一部をネットワークにエクスポートするように指示しなければなりません。

1. ファイルサーバとなるコンピュータに正規のアカウントを使用してログインします。
2. /NextAdmin に格納されている NFSManager.app を起動します。ローカルドメインに対する「インポートディレクトリ」ウィンドウと「エクスポートディレクトリ」ウィンドウをキーウィンドウにします。
3. 「エクスポートディレクトリ」ウィンドウ内をクリックし、「エクスポートディレクトリ」ウィンドウをキーウィンドウにします。
4. エクスポートするディレクトリを追加するために、「エクスポートディレクトリ」ウィンドウの右上の「追加」ボタンをクリックします。「ディレクトリをエクスポート」パネルが表示されます。
5. ブラウザを使用してディレクトリを選択し、「OK」をクリックします。ローカルドメインに対する root パスワードの入力を要求するパネルが表示されます。
6. root パスワードを入力し、「ログイン」をクリックします。エクスポートしたディレクトリがリストに表示されます。
7. エクスポートしたディレクトリに対し、特定のグループをリードオンリアクセスに指定する場合は、「リードオンリ」コラムの下のテキストフィールドをクリックします。ホスト名を入力し、テキストフィールドの下の「追加」ボタンをクリックします。リードオンリアクセスに指定するホストが他にもある場合は、同じ方法を繰り返して下さい。
8. エクスポートしたディレクトリに対し、特定のグループをリードライトアクセスに指定する場合は、「リードライト」コラムの下のテキストフィールドをクリックします。ホスト名を入力し、テキストフィールドの下の「追加」ボタンをクリックします。リードライトアクセスに指定するホストが他にもある場合は、同じ方法を繰り返して下さい。リードオンリのリストとリードライトのリストに同じホスト名を指定することはできません。
9. 「リードオンリ」及び「リードライト」のどちらにも指定しなかったホストに対しデフォルトのアクセスを設定するために、「下記のリストにないホストの標準アクセス権」と記された右のポップアップリストからアクセス権を選択します。これによって、上記のいずれにも指定しなかったネットワーク上のすべてのホストに対し、選択した標準アクセス権が適用されます。標準アクセス権に対する制限を以下に示します。
  - 「リードオンリ」欄にホスト名が指定されている場合は、標準アクセス権は「なし」のみとなります。
  - 「リードライト」欄にホスト名が指定されていて、「リードライト」欄にはホスト名が1つも指定されていない場合は、標準アクセス権は「リードオンリ」と「なし」となります。

- 上記の2つの欄のいずれにもホスト名が指定されていない場合は、デフォルトアクセスは、「リードライト」、「リードオンリ」、「なし」となります。この場合、「なし」を選択すると、どのユーザもアクセスできないディレクトリをエクスポートすることになるので、ほとんど意味がありません。
10. ホストに対し root アクセスを許可する場合は、「ルートアクセス」コラムを使用します。コラムの下のテキストフィールドをクリックし、ホスト名を入力し、「追加」をクリックします。root アクセスを許可するホストが他にもある場合は、同じ方法を繰り返して下さい。
  11. 未登録のユーザに対し、エクスポートしたディレクトリへのアクセスを許可しない場合は、“未登録ユーザのアクセスを許可”と記されたスイッチをクリックして、チェックマークを消します。
  12. 未登録のユーザに対し、エクスポートしたディレクトリへのアクセスを許可し、未登録のユーザを nobody 以外のユーザとして扱う場合は、「nobody ユーザとして」と記されたボタンをプレスし、表示されるポップアップリストから“指定 ID のユーザとして”を選択します。そして、テキストフィールドにユーザ ID(ユーザ名ではありません)を入力します。
  13. 「OK」をクリックします。デフォルトアクセスの設定がその他のアクセスの設定と重複している場合は、設定の変更を求めるパネルが表示されます。変更する場合は、「OK」をクリックします。
  14. `exports -a` を実行します。

### 8.2.2 クライアント側での設定

エクスポートされたディレクトリをインポートするには、以下の方法で操作します。

1. ネットワークに接続されているコンピュータに、正規のアカウントを使用してログインします。
2. `/NextAdmin` に格納されている `NFSManager.app` を起動します。ローカルドメインに対する「インポートディレクトリ」ウィンドウと「エクスポートディレクトリ」ウィンドウが表示されます。
3. メインメニューから「インポート」を選択します。
4. 「NetInfo ドメインを選択」というパネルが出るので正確にドメインを指定します。ネットワークの時には `/` です。ネットワークに設定すれば NetInfo で接続されているすべてのマシンで設定が有効になります。
5. マウントポイントはデフォルトでは `/Net` の下にマシン名、ディレクトリ名というようになっています。マウントポイントとして `/Net` を指定することは、OPENSTEP では自動マウントが行われることを示します。マウントポイントを `/Net` 以外のディレクトリに指定したい場合は、「マウントポイント」テキストフィールドにマウントポイントをフルパス名で指定します。「選択」をクリックして「マウントポイントを選択」パネルを使用することによっても、マウントポイントを設定することができます。

6. デフォルト以外のマウントオプションを指定する場合は、フィールドを以下のように設定します。

- ホストディレクトリをリードオンリでマウントする場合は、「マウントモード」のポップリストから「リードオンリ」を選択します。
- ディレクトリをバックグラウンドでマウントする場合は、「マウント方法」のポップリストから「バックグラウンド」を選択します。
- “「サーバーの応答がない場合」”のポップアップリストの3種類の選択肢の中から1つを選択します。
- `setuid` ファイルを通常のユーザとして実行する場合は、「`setuid` ファイルの取り扱い」のポップアップリストから「`setuid` ビットを無視」を選択します。
- 「エキスパート」場尾単が、追加オプションとしてパネルに表示されています。通常は、すべてデフォルトの状態のままにしておきます。

7. 「OK」をクリックします。

8. `mount -a` を実行します。

以上によって、リモートディレクトリをマウントするための設定は終了しました。ディレクトリを `/Net` にマウントするように設定した場合は、自動的にマウントが行われます。それ以外の場所にマウントする場合は、クライアントマシンの次のブート時(一度クライアントマシンが Off になった後)にマウントが行われます。(export, mount コマンドがうまくいけばこのときにマウントされます)

**計算機委員会の標準設定** 標準の設定の例を図 7 に示します。上の図ではネットワークドメイン (`/`) のインポートディレクトリを示します。このように NetInfo では `fstab` の情報をネットワークで共有できるので便利です。ここにはマウントしなければいけないすべてのディレクトリをあげておくことができます。通常 `/Net` にオートマウントでマウントします。それから必要な場所にリンクを張っています。具体的にはクライアントではリンクは次の位置を行っています。

```
# ln -s /Net/aquarius10/home/aquarius10 /home/aquarius10
# ln -s /Net/aquarius10/home/local /usr/local
# ln -s /Net/aquarius10/home/local/LocalApps /LocalApps
# ln -s /Net/aquarius10/home/local/LocalLibrary /LocalLibrary
```

図にはその他に `aquarius0a`, `aquarius0d` 等が黒 NeXT としてあげられています。

図 7 の下の図はサーバ (`/aquarius10`) でのエクスポートディレクトリを示しています。リードライトは指定した方がセキュリティ上いいです。

### 8.3 ファイルシステムをマウントしたい

NFS のところでも使った `mount` コマンドについて解説します。



図 7: NFSManager の画面

| オプション | 内容   |
|-------|--|
| -p    | 現在のマウント情報を /etc/fstab のフォーマットで表示する               |
| -a    | /etc/fstab にリストされているすべてのファイルシステムをマウントする。         |
| -t    | マウントするファイルシステムのタイプを指定する。                         |
| -lo   | ローカルなファイルシステムを別のディレクトリにマウントする<br>ループバック・ファイルシステム |
| -r    | 読み出し専用でマウントする。                                   |

表 4: mount コマンドのオプション

ファイルシステムのマウントには mount コマンドを使います。mount コマンドの使い方は次のようになります。

```
mount (オプション) (デバイス名) (マウントポイント) % mount
現在のマウントの状態を表示します。
% mount aquarius10:/home /home
aquarius10 の /home をこのマシンの /home にマウントする。
% mount -a
/etc/fstab と Netinfo の fstab に従った内容通りにマウントを試みます。
```

デバイス名を指定しないでマウントポイントだけを指定すると、fstab の該当するマウントポイントの指定に従ってマウントします。また、デバイス名もマウントポイントも指定しない場合には、現在のマウントの状態を表示します。

マウントに失敗した時にはエクスポートされていない場合やデバイスが無い場合などがあります。

mount コマンドで使用できる主なオプションは表のとおりになります。

## 9 NIS を設定する

NIS(Network Information Services) は NFS に匹敵するくらい非常に重要な、ネットワーク上で有効な機能です。NIS, NFS とともに Sun が開発したもので、初期の SUN が売れた原因の一つです。NIS は当初 YP(Yellow Page) と呼ばれておりましたが、ATT とのトレードマークの問題があり、NIS という名前では呼ばれております。NIS により、NetInfo と同様にネットワーク上で共有する管理情報を一元管理する事ができるようになります。

NIS は Solaris では NIS+ と進化して階層構造の管理をすることができるようになっており、ほとんど業界標準です。当学科では OPENSTEP を使っておりますので、前に述べている NetInfo を用いる事により NIS 以上の情報を共有管理する事ができますが、NetInfo は他のメーカーのマシンではほとんど採用されておられません。したがって、他のメーカーのマシンに情報を提供するのに NIS を使うことにします。



## 9.1 NIS サーバを設定したい

ここでは NIS サーバを OPENSTEP 上で実現する方法を示しますが、1997 年 2 月現在では障害のため Intel 機の上では NIS サーバとして運用することはできません。したがって以下のことは NEXTSTEP で行なえるというように読んで下さい。

### 9.1.1 ドメイン名の設定

最初にドメイン名の設定が必要です。/etc/hostconfig に

List 11 /etc/hostconfig の一部

---

```
1 YPDOMAIN = aquarius10
```

---

のような行をいれます。/etc/rc はこれが書かれていると boot の際に

```
domainname $YPDOMAIN >/dev/console 2>&1
```

を実行します。これにより NIS のドメイン名を設定ができます。これは HostManager.app により設定することもできます。(図 2)

### 9.1.2 NIS データベースのインストール

NIS は以前 YP と呼ばれていたもので、コマンドやファイルの名前に yp がつく事が多いです。NIS の関連ファイルは /etc/yp の下にあります。

```
# /etc/yp/ypinit -m
```

を実行する事により、ドメイン名に沿った NIS データベースのインストールを行います。途中でドメイン名を指定するように聞いて来ますから、キーボードから入力します。

NIS を実行するには NIS 情報を発信する ypserv やそれを受け取る ypbind が立ち上がっている必要があります。ypinit をした後は reboot をして rc により自動的に関連のデーモンを立ち上げます。

### 9.1.3 その他の設定

NIS で管理している情報は主に、

- /etc/passwd
- /etc/group

- /etc/hosts
- /etc/sendmail/aliases

です。これらのファイルの情報は NIS データベースに書き出されてネットワーク上で参照されます。これらの情報が変更になった時には NIS データベースを更新する必要があります。通常は /etc/yp の下の Makefile にしたがって更新が行われますので、(/etc/yp; make) をするだけで済みます。しかし OPENSTEP では本来は NetInfo で情報を管理しておりますので、NetInfo に情報を蓄えております。したがって、一旦 NetInfo の情報を /etc/ に下のファイルに書き出して、それを NIS データベースに移してネットワークに提供する事になります。そのための実行形式のファイルを /etc/pushinfo として準備しております。

---

#### List 12 /etc/pushinfo

---

```
1  #! /bin/sh
2  # /etc/pushinfo
3  #
4  # NetInfo -> plain text (niload/nidump)
5  # aliases bootptab bootparams exports fstab group
6  # hosts networks passwd printcap protocols rpc services
7  #
8  # NIS
9  # passwd group hosts ethers networks rpc
10 # services protocols netgroup aliases
11 #
12 /usr/bin/nidump passwd / > /etc/passwd
13 /usr/bin/nidump group / > /etc/group
14 /usr/bin/nidump hosts / > /etc/hosts
15 /usr/bin/nidump aliases / > /etc/sendmail/aliases
16 cd /etc/yp
17 make
```

---

ユーザを登録した時、メールエイリアスを変更した時などのように NIS の情報に変更があった時にはこの /etc/pushinfo を最後に実行します。

## 9.2 NIS クライアントを設定したい

NIS クライアントの設定は /etc/hostconfig に

---

#### List 13 /etc/hostconfig の一部

---

```
1  YPDMAIN = aquarius10
```

---

---

と書きます。これにより、ブート時に `yplibind` が `rc` により起動されます。

次に、`/etc/passwd`, `/etc/group` の一番下の行に

---

---

List 14 `/etc/passwd`, `/etc/group` の下

---

---

1 +:

---

---

を追加します。これにより、これらのファイルに書かれている情報にプラスして NIS からの情報を利用する事になります。

OPENSTEP では `lookupd` デーモンを用いて管理情報の優先度を定めることができます。たとえば `/etc/hosts` の情報は、フラットファイル (`/etc/hosts` 自身、`NetInfo`, `NIS`, `DNS` から得ることができます。NIS の情報を使うためにはその旨記述する必要があります。

```
# niutil -destroy . /locations/lookupd
# niutil -create . /locations/lookupd
# niutil -createprop . /locations/lookupd hosts CacheAgent NIAgent YPAgent DNSAgent
# niutil -createprop . /locations/lookupd users CacheAgent NIAgent YPAgent FFAgent
# niutil -createprop . /locations/lookupd groups CacheAgent NIAgent YPAgent FFAgent
# niutil -createprop . /locations/lookupd aliases CacheAgent NIAgent YPAgent FFAgent
```

これらの設定が終わったら `reboot` をします。

## 9.3 NIS の運用をしたい

設定が一通りすめば、クライアントではサーバーの管理情報を手にいれる事ができるようになり、クライアントに `login` 名が記録されていなくても、サーバに記録されていれば `login` ができるようになります。

例えば、ユーザを登録した時のように、NIS で管理しているファイルを変更した時には、前述の `/etc/pushinfo` を実行します。

```
# /etc/pushinfo
```

これにより、`NetInfo` から管理情報を `/etc/` の下の `/etc/passwd` のようなファイルに一旦移し、それから `/etc/yp` の下で `make` を実行します。

### 9.3.1 パスワードの変更をおこないたい

`NetInfo` で全体を管理している時には `Preferences` を用いてユーザはパスワードを簡単に変更できます。しかし NIS が動いている状態では個人のパスワードの変更は `yppasswd` コマンドを

用いて変更します。このためにはサーバーで yppasswdd(yppasswd デーモン) が動いている必要があります。また yppasswdd が変更の情報を得た時には (cd /etc/yp; make) が実行されますが、その時に NetInfo の情報も書き代わるように変更しなければいけません。これらの設定は以下ようになります。

/etc/rc に追加

List 15 /etc/rc に追加

---

```
1 if [ -f /usr/etc/rpc.yppasswdd ]; then
2     /usr/etc/rpc.yppasswdd /etc/passwd -m passwd &&
3     (echo -n ' rpc.yppasswdd')>/dev/console
4 fi
```

---

/etc/yp/Makefile に追加

List 16 /etc/yp/Makefile に追加

---

```
1 cat /etc/passwd | niload -d passwd . ; \
2 echo 'update netinfo'; \
```

---

以上で yppasswd によりパスワードを変更することができるようになります。

## 10 ユーザを登録・変更・削除する (UserManager.app)

NeXT でユーザのアカウントを登録・変更・削除するには /NeXTAdmin/UserManager.app を使用します。/etc/passwd を書き換え、NetInfo に登録し、ユーザのホームディレクトリをするのはコマンドラインからもできますが、UserManager.app を利用する方法がはるかに便利です。NEXTSTEP3.3J からは UserManager.app のバルク機能で一度に多人数のユーザを作ることできるようになりました。

### 10.1 ユーザアカウントを計画したい

新たなユーザアカウントを追加する前に、アカウントをどのようにコンフィグレーションするかについて計画してみましょう。ユーザアカウントについての計画を立てるには、以下の点を考慮します。

#### 10.1.1 ローカルアカウントまたはネットワークアカウント

ローカルアカウントは、特定のコンピュータ上だけで使用できるアカウントです。ネットワークアカウントは、ネットワーク上の複数またはすべてのコンピュータから使用できるアカウントです。ユーザを1台のコンピュータしか使用できないように制限するのか、あるいは、複数

台のコンピュータにログインできるようにするのを決定します。各ユーザアカウントは、1つのNetInfoドメインにのみ存在します。例えば、kiyonagaに対するユーザアカウントをルートドメインに追加し、さらに、ローカルドメインに追加するといったことは避けて下さい。このようにユーザアカウントが重複すると、kiyonagaがどちらのコンピュータにログインしているかによって、パスワード、ハードディレクトリ、ファイルのオーナーシップが異なってしまいます。ですからローカルで管理するのは、rootなどの特別なユーザのみです。一般ユーザはネットワークで使えるようにサーバに登録します。

### 10.1.2 ユーザ名

ユーザ名は、ユーザがログインするときに使用する名前です。ユーザ名には、文字、数字、アンダースコア(\_)を使用することができ、文字数は8文字までです。一般的に、ユーザ名は大文字を使用しません。また、学科内の教官の名前と重複するようなアカウント名は以前はメールアドレスとしてエイリアスで登録されていて登録することはできませんでしたが、これからは研究室内で重複するユーザ名でなければ大丈夫です。

### 10.1.3 ユーザID

ユーザIDは、システムが各ユーザアカウントを識別するために使用する番号です。UserManagerは、新たに追加されたユーザアカウントに自動的にユーザIDを割り当てます。ファイルのパーミッションやオーナーシップは、ユーザ名ではなくユーザIDに基づいて管理しています。従って、複数のユーザが同じユーザIDを持たないことが重要となります。たとえば同じユーザIDが2人のユーザに割り当てられたとすると、一方のユーザがどのようなパーミッションであったとしても、もう一方がファイルをリード、ライトしたり、ファイルを作成することができてしまいます。ネットワークに接続されたマシンでは、NetInfoドメイン階層全体にわたって、ID番号はユニークでなくてはなりません。学科のシステムでは研究室ごとにID番号が割りふられています。例えば、capricornは1000番台、aquariusは2000番台が割り振られています。また学生は通常は1100番や2100番以降に振ってあります。卒業などでユーザがいなくなった時でも一度使ったユーザIDは使わない方がいいです。nidumpコマンドを用いてあらかじめ番号を確認しておくのがいいでしょう。

```
# nidump passwd /
```

### 10.1.4 ホームディレクトリ

ホームディレクトリは、ユーザが自分のファイルを格納するディレクトリです。アカウントをセットアップすると、UserManager.appによってホームディレクトリが作成されます。ネットワークアカウントを作成した場合は、ユーザのホームディレクトリは、ユーザがログインできるすべてのコンピュータから利用できるようにしなければなりません。これは、ホームディレクトリは共有ファイルシステム上になければならないことを意味します。従って、ネットワークアカウントを作成する場合は、あらかじめホームディレクトリサーバをセットアップしておかなければなりません。学科では/homeをクライアントはサーバからマウントすることになっています。またユーザのホームディレクトリは/home/zodiac20/kiyonagaのようになっています。

### 10.1.5 ユーザグループ

ユーザグループは、お互いのファイルの一部(または全部)をリード、ライトできるようにセットアップされたユーザの集まりです。各ファイルやディレクトリは、関連する1つのグループとグループのメンバに適用されるパーミッションを持っています。ここでのパーミッションとは、グループ外のユーザに与えられパーミッションとは異なったものです。ユーザはいくつものグループのメンバになることができ、各ファイル(またはディレクトリ)ごとに、いずれかのグループを指定することができます。

UserManager.app は、新たなユーザアカウントにデフォルトのグループを1つ割り当てます。新たなユーザのホームディレクトリが作成されると、そのホームディレクトリにはデフォルトが割り当てられます。ユーザが新たなファイルやディレクトリを作成すると、それらには、その親のディレクトリに割り当てられていたのと同じグループが割り当てられます。これによって、ユーザのすべてのファイルにデフォルトのグループが割り当てられることになります。

ユーザアカウントを新たに追加する前にユーザグループを作成しておき、新たなユーザを作成しておいたユーザグループに割り当てることができます。

学科では基本的には同じ小講座の二つの研究室が一つのグループを共有しております。たとえば capricorn は 1000, aquarius は 2000 のように振られております。自分の属しているグループをデフォルトにします。

これらのユーザ ID、ホームディレクトリ、ユーザグループの規則は学科を通して重ならないように計算機委員会で決めています。これは数値計算を行う scorpio 系で各研究室のホームディレクトリをマウントした際にぶつからないようにするためです。かならず守ってください。

### 10.1.6 パスワード

各ユーザアカウントには、パスワードを設定すべきです。アカウントにパスワードを設定しておかないと、そのアカウントを使用して誰でもがログインできてしまいます。パスワードは8文字までとし、覚えやすくかつ解読しにくいものにします。小文字、大文字、数字を組み合わせると、解読しにくくなります。新たなアカウントの作成時にパスワードを設定しない場合は、ユーザーに対し、Preferences.app を使用してなるべく早くパスワードを設定するよう通知して下さい。

## 10.2 ユーザアカウントの登録をしたい

ユーザをコンピュータやネットワークにアクセスできるようにするには、新たなアカウントを作成します。この節では、 UserManager.app の NewUser ウィンドウの詳細形式を使用して、ユーザアカウントを追加する方法を説明します。

1. 新たなユーザのホームディレクトリを作成するコンピュータにログインします。ローカルアカウントの場合はローカルのコンピュータにログインし、ネットワークアカウントの場合はホームディレクトリサーバにログインします。従って、通常のユーザの登録の場合はサーバ機にログインします。



図 8: UserManager の画面

2. /NextAdmin に格納されている UserManager.app を起動します。
3. root としてログインしていない場合は、このアプリケーションを実行するにはスーパーユーザのステータスが必要であることを示すパネルが表示されます。root のパスワードを入力し、「ログイン」をクリックします。
4. ここで、「ユーザレコード」メニューから「標準値...」を選択して、標準の値を設定しておくとう便利です。たとえば、aquarius の場合には「ID 番号の最小値」を 2100、「標準グループ番号」を 2000、「標準ホーム」を /home/aquarius10 のようにしておけば、手間が省けます。
5. 「ユーザレコード」メニューから「新規ユーザ」を選択します。「ユーザタイプ」パネルが表示されます。
6. ユーザアカウントのタイプを選択します。
7. 「新規ユーザ」ウィンドウが開いたら「簡易形式」になっているクリックをプルダウンして「詳細形式」に変更します。
8. 最初の 3 つのフィールド、「フルネーム」、「ユーザ名」、「パスワード」に適切な値を入力します。
9. 「標準グループ」の設定は上部の「詳細形式」クリックをプルダウンして、「グループ」を選択して行います。ドメインの下のグループの中から適切なグループを「登録」して、「標準を作成」をクリックします。通常は aquarius のようなグループのみで結構ですが、root の権限を与える時には wheel にも入ってもらいます。
10. 「ユーザ ID」各ユーザアカウントには、ユニークなユーザ ID 番号 (UID) が割り当てられます。システムはこの番号を使用して、ユーザを識別します。UserManager.app は、利用できる番号を決定し、自動的にアカウントにその番号を割り当てます。自動的な割当てが間違っていた時には、この番号が使われていない番号を用いて設定します。
11. 「ホーム」ではユーザのホームディレクトリを設定します。新たなユーザ名で終了するフルパス名で入力してください。
12. 「ログインシェル」はユーザが Terminal ウィンドウをオープンしたとき、起動するプログラムを設定するフィールドです。また、ユーザが Terminal ウィンドウを通してリモートからログインした場合も、ここで設定したプログラムが起動します。通常は、この内容は変更する必要はありません。別のシェルをここで設定した場合は、設定したプログラムを /etc/shells に記述しなければなりません。
13. 「ユーザレコード」のメニューから「セーブ」を選択し、新たに設定したユーザアカウントをセーブします。
14. パスワードを確認するためのパネルが表示されるので、設定したパスワードを入力します。
15. 新たなユーザアカウントに関する情報が記載されたアラートパネルが表示されます。ここで「yes」をクリックするとディレクトリを作成するかどうかを確認するアラートパネルが表示されます。ここで「yes」をクリックすればアカウントの作成が始まります。この時の



元となるドットファイル(.cshrc 等)は /usr/template/user.Japanese です。このディレクトリの下を注意深く設定すればドメインの設定をデフォルトにすることもできます。「新規ユーザ」ウィンドウのクローズボタンがバツ印になれば終了です。

- 最後に NIS を更新する操作をします。NeXTstation で NIS サーバを運用している際には /etc/puhsinfo を実行します。

```
# /etc/pushinfo
```

### 10.3 ユーザアカウントの削除やログインの無効にしたい

ユーザが卒業などでいなくなった時にはユーザアカウントを削除する必要があります。しかし、ユーザ ID やホームディレクトリを残す必要がある時にはログインを無効にすることもできます。削除するにはユーザをオープンしてメニューから削除を選びます。ログインを無効にするにはログイン無効を選びます。これによりパスワードフィールドに \* がつきます。ユーザ ID やホームディレクトリはそのまま残りますがログインはできなくなります。

## 11 新しいマシンを追加する

### 11.1 クライアント機を接続したい

NeXTstation が安価で配布されたので、研究室によっては何台もの NeXTstation を接続することになるかも知れません。この章では具体的なクライアント機の接続の方法について述べます。

#### サーバ機での設定

- サーバ機で必要なディレクトリがエクスポートされているか確かめます。サーバ機はホームディレクトリ (/home/aquarius10) とバイナリディレクトリ (/home/local) がエクスポートされている必要があります。また Intel と Motorola ではバイナリは違うのでそれぞれのサーバ機で設定されている必要があります。
- サーバ機でネットワークドメインにおいてホームディレクトリとバイナリディレクトリがマウントができるように設定されているか確かめます。たとえば次の例では aquarius10 からはホームディレクトリと Intel のバイナリディレクトリ、及びメールディレクトリをマウントするようになっています。また aquarius0a からは Motorola のバイナリディレクトリをマウントするようになっています。

```
% nidump fstab /
aquarius10:/usr/spool/mail /usr/spool/mail nfs rw,bg,intr,noquota 0 0
aquarius10:/home/aquarius10 /Net nfs bg,intr,net,timeo=6 0 0
aquarius10:/home/local /Net nfs net 0 0
aquarius0a:/home/local /Net nfs net 0 0
```

3. 同様にサーバ機でローカルドメインにおいてホームディレクトリとバイナリディレクトリがエクスポートされているか確かめます。

```
% nidump exports .  
% nidump exports /aquarius10
```

## クライアント機での設定

1. クライアント機をシングルユーザで立ちあげて /etc/hostconfig と /etc/netinfo/ を標準の状態に戻します。(1章参照)

```
# cd /etc  
# mv hostconfig hostconfig.old  
# mv netinfo netinfo.old  
# cp /usr/template/client/etc/hostconfig .  
# cp -r /usr/template/client/etc/netinfo .
```

2. クライアント機のブートの情報から Ether address(例: 00:60:97:02:7c:ca) を書き取って、サーバで HostManager.app を使ってマシンの登録をします。(6.2章参照) これでクライアントはサーバから IP アドレスとホスト名を受け取り、ブートすることができるようになります。
3. Control-d を入力することによりブートを開始します。正常にサーバから情報が得られないときには Control-c か c を入力することを要求されますのでもう一度 reboot してやり直します。
4. 正常に立ち上がると、キーボードのタイプを聞かれるので選択します。
5. 最初は me のパスワードがないので me のユーザ ID で立ち上がるので me のパスワードと root のパスワードを設定します。
6. いくつかのリンクするファイルを設定します。

```
# ln -s /Net/aquarius10/home/aquarius10 /home/aquarius10  
# ln -s /Net/aquarius10/home/local/LocalLibrary /LocalLibrary  
# ln -s /Net/aquarius0a/home/local /usr/local  
# ln -s /Net/aquarius0a/home/local/LocalApps /LocalApps
```

7. その他の設定をします。たとえば /etc/shells, /etc/hosts.equiv, /etc/swaptab など

```
# dwrite loginwindow HostName namehost
```

以上で設定は終わりです。意外と簡単だったのではないのでしょうか。OPENSTEP 同士であるとネットワークに関する設定はほとんどサーバで行っているため、メール、プリンタなどは全く設定する必要がありません。その分の管理の負担が非常に軽くなっていることが分かります。

## 11.2 スタンドアロンとして接続したい (SimpleNetworkStarter.app)

NeXT をスタンドアロンとして設定するときには SimpleNetworkStarter.app を用います。

1. クライアント機をシングルユーザで立ちあげて /etc/hostconfig と /etc/netinfo/ を標準の状態に戻します。(1章参照)

```
# cd /etc
# mv hostconfig hostconfig.old
# mv netinfo netinfo.old
# cp /usr/template/client/etc/hostconfig .
# cp -r /usr/template/client/etc/netinfo .
```

2. Control-d を入力することによりブートを開始します。ネットワークから物理的に切り離されている時にはそのまま立ち上がります。ネットワークにつながれている時にはサーバから情報を得ようとするので Control-C か c を入力して立ち上げます。
3. 正常に立ち上がると、キーボードのタイプを聞かれるので選択します。
4. 最初は me のパスワードがないので me のユーザ ID で立ち上がるので me のパスワードと root のパスワードを設定します。
5. /NextAdmin/SimpleNetworkStarter.app を立ち上げます。図 9 に初期画面を示します。設定には大きく分けて次の 3 つがあります。
  - スタンドアロンでネットワークを用いる。
  - クライアントとして設定する。
  - サーバーとして設定する。

スタンドアロンで使うときには一番上を選択します。

6. 次にネットワークの設定として「ネットワークオプション...」ボタンを押すと図 10 が現れるのでそれぞれ必要な値を設定します。最後に「コンフィグレーション」ボタンを押すと設定を開始します。

これでネットワークは使うが、スタンドアロンになります。

## 12 サーバ機を立ちあげる

この章ではこれまでの設定の知識をすべて使ってサーバ機を立ちあげるについて述べたいと思います。細かいことまですべて触れていくと膨大になるので、ここでは概要について述べます。

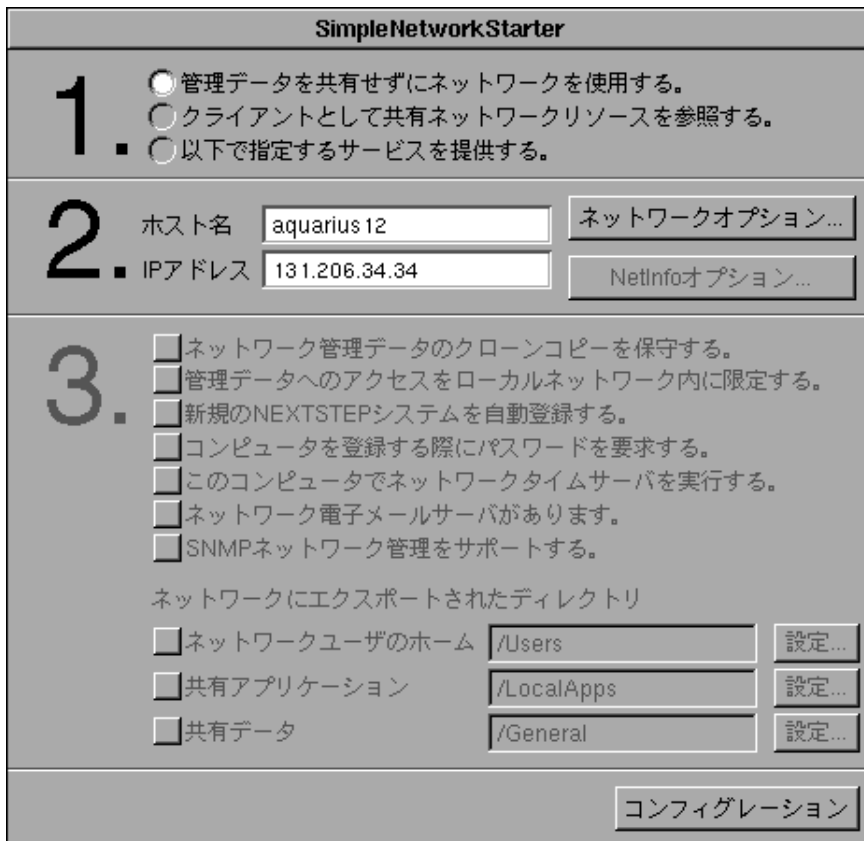


図 9: SimpleNetworkStarter.app によるスタンドアロンの設定



図 10: SimpleNetworkStarter.app によるネットワーク情報の設定

## 12.1 OPENSTEP をインストールしたい

OPENSTEP を Intel ベースの PC にインストールする方法については「OPENSTEP インストールとシステム設定」を参照しながら作業を行います。

1. ハードウェアの接続を完了させます。
2. 電源を入れてインストール用のディスクと CD-ROM をセットします。
3. OPENSTEP が順調にあがるとパーティションの設定をします。OPENSTEP では 2Gbyte を越えるパーティションを割り当てることができませんので、パーティションの大きさを 2Gbyte 以下になるように設定します。
4. OPENSTEP をインストールし終わると一度再起動します。
5. システム設定を行います。Configure.app を用いてディスプレイカード、マウス、ネットワークカード、SCSI カード、音源カード、その他をそれぞれ設定していきます。
6. 基本的なソフトのインストールをおこなう画面になり、インストールを行います。その後再起動します。

## 12.2 サーバ機の実環境設定をおこないたい

次に下準備が必要となります。

1. 最初には me のパスワードがないのでそのまま me で立ち上がります。そこで me と root のパスワードを付けます。
2. 古いマシンがありその管理情報を引き継ぎたいときには、それらのファイルを集めます。考えられるのは以下のファイルです。
  - /etc/passwd
  - /etc/group
  - /etc/exports
  - /etc/shells
  - /etc/hosts
  - /etc/hosts.equiv
  - /etc/sendmail/aliases

これらは nidump を利用して準備したりします。これらをそれぞれ passwd.old のように呼ぶことにします。

3. 古いマシンのユーザホームディレクトリがあればそれをコピーします。コピーの方法はネットワーク経由、あるいはハードディスクをつないで行う方法が考えられます。いずれにしろ次のようになります。

```
# cd sourcedir;
# gnutar cf - aquarius10 | (cd /home; gnutar xf -)
```

4. 同様に古いマシンにメールディレクトリ (/usr/spool/mail) があればそれをコピーします。このときには上流からメールが届かないように、サブドメインを切り離して行う方がベターです。具体的にはブルータの電源を落としておけばよろしいでしょう。

いよいよ設定にかかります。

1. まず SimpleNetworkStarter.app を用いてネットワークの初期の設定を行います。図 11, 12を参照してください。
2. 古いマシンの管理情報を設定してきます。

```
# niload -d passwd / < passwd.old
# niload -d group / < group.old
# niload -d group / < group.old
# niload -d aliases / < aliases.old
# cp shells.old /etc/shells
# cp hosts.equiv.old /etc/hosts.equiv
# cp aliases.old /etc/sendmail/aliases
```

3. su の安全性をあげる設定をします。(1.1章参照)
4. ネットワーク時間の設定をします。(5.1章参照)
5. crontab.local を設定します。(5.2.1章参照)
6. dwrite で設定できるものを設定します。(5.4章参照)
7. ネームサーバの設定をします。(6.4章参照)
8. HostManager.app を用いてドメイン内のマシンを登録していきます。この際には hosts.old が参考になりますが、niload で設定するのは危険です。niload だけでは NetInfo のサービスをクライアントが受けられなくなります。NetInfo の情報を受けるときには HostManager.app でセーブするときに「ホストの親ドメインとして設定する」にチェックをする必要があります。(6.2章参照) 具体的には一度ホストを開いて「別ドメインにセーブ」を用いてネットワークに「ホストの親ドメインとして設定する」をチェックしてセーブし直します。
9. NFSManager.app を用いて NFS の設定を行います。NFS の仕方にはいろいろな考え方がありますが、ネットワークに必要なディレクトリをすべて /Net に automount しておき、個々のマシンで必要なディレクトリをリンクするようにする方法が良いと思われます。(8.2章参照)
10. 必要なリンクを張ります。

**SimpleNetworkStarter**

**1.**  管理データを共有せずにネットワークを使用する。  
 クライアントとして共有ネットワークリソースを参照する。  
 以下で指定するサービスを提供する。

**2.** ホスト名  ネットワークオプション...  
IPアドレス  NetInfoオプション...

**3.**  ネットワーク管理データのマスタコピーを保守する。  
 管理データへのアクセスをローカルネットワーク内に限定する。  
 新規のNEXTSTEPシステムを自動登録する。  
 コンピュータを登録する際にパスワードを要求する。  
 このコンピュータでネットワークタイムサーバを実行する。  
 ネットワークの電子メールサーバを保守する。  
 SNMPネットワーク管理をサポートする。

ネットワークにエクスポートされたディレクトリ

ネットワークユーザのホーム  設定...  
 共有アプリケーション  設定...  
 共有データ  設定...

コンフィグレーション

図 11: SNS.app によるサーバの設定

 **その他のネットワーク情報**

—— ルータ ——

使用しない  
 ダイナミック

—— ネットマスク ——

標準設定  
 自動設定

—— NISドメイン名 ——

使用しない

—— Broadcastアドレス ——

標準設定

ローカル NetInfoデータの参照をローカルネットワーク内に限定する

取消し 設定 ←

図 12: SNS.app によるサーバのネットワーク情報の設定

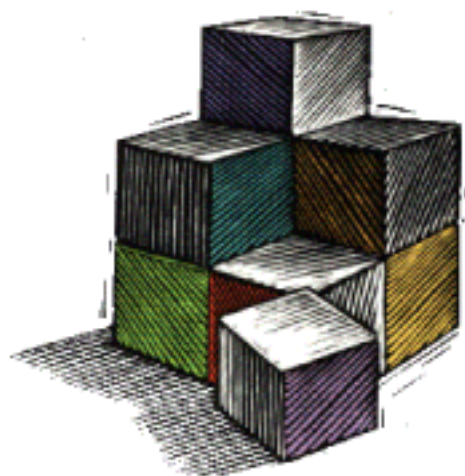
11. PrintManager.app を用いてプリンタの設定を行います。(6.5章参照)

12. 必要に応じて NIS の設定を行います。(9章参照)

以上で基本的な設定は済みました。後は OPENSTEP 上で動くソフトのインストールなどの作業があります。



## 第3部 システムの管理



# OPENSTEP™

## 13 システムの状態を管理する

現在稼働している OS のバージョンやシステムのハード構成情報をはじめとしたシステムの固有情報や、利用者の状況、そのシステムで現在起動しているプロセスの種類やその実行の状態等システムを維持管理するために必要とされる情報を取得するためのコマンドについて説明します。

### 13.1 ホスト名を調べたい (hostname, hostid, domainname)

現在のホスト名、ホスト ID、ドメイン名を調べたい時は `hostname`, `hostid`, `domainname` コマンドを使います。

```
% hostname
aquarius10
% hostid
0x1022390
% domainname
aquarius10
```

### 13.2 グループを調べたい (groups)

その人が含まれているグループ名を知りたいときには、そのユーザ名を指定して `groups` コマンドを使用します。 `groups` コマンドの実行結果は、以下のようになります。

```
% groups kiyonaga
wheel other zodiac
```

### 13.3 現在の利用者を表示したい (who)

現在の利用者とその利用状況を知りたい時には、 `who` コマンドを使用します。 `who` コマンドは `/var/adm/utmp` ファイルを検索して、ログイン名、端末名、ログイン時刻を出力します。

```
% who
kiyonaga console Feb 16 13:11
kiyonaga ttyt1 Feb 16 14:41
```

`who` コマンドで `"am i"` オプションを使用すると、コマンドを発行したユーザの情報を出力します。

`state` 情報は、 `-T` オプションで出力されます。 `idle` 情報及び `pid` 情報は、 `-u` オプション及び `-l` オプションで出力されます。

## 13.4 現在の利用状況を表示したい (w)

現在の利用者とその利用状況を知りたい時には、`w` コマンドを使用します。

```
% w
4:47pm up 1 day,  6:13,  2 users,  load average: 0.20, 0.09, 0.09
User      tty      login@  idle   JCPU   PCPU   what
kiyonaga console  1:11pm162days      -
kiyonaga ttyt1    2:41pm                -csh
```

ロードアベレージについては `rup` コマンド (13.10章) を参照して下さい。

## 13.5 個人情報を表示したい (finger)

ローカルユーザとリモートユーザに関する情報を知りたい時には、`finger` コマンドを使用します。オプションを使用せずに `finger` コマンドを実行すると、そのローカルホストに現在ログインしているユーザ情報を出力します。出力結果は、以下のようになります。

```
% finger
Login      Name          TTY Idle   When      Office
kiyonaga Akira KIYONAGA      co 164d Mon 14:38
kiyonaga Akira KIYONAGA      p1      Mon 14:39
```

また、`finger` コマンドに“ユーザ名@ホスト名”の形式で引数を指定すると、指定されたユーザに関する情報を照会していきます。

```
% finger kiyonaga@zodiac20
Login name: kiyonaga          In real life: Akira KIYONAGA
Directory: /home/zodiac20/kiyonaga  Shell: /bin/csh
On since Feb 19 10:34:49 on ttyt3 from :0.0
3 hours 9 minutes Idle Time
No Plan.
```

## 13.6 ユーザのログイン情報を見たい (last)

ユーザのログイン情報を見たい時には、`last` コマンドを使います。

```
% last
kiyonaga ttyt1          Fri Feb 23 14:03  still logged in
kiyonaga console zodiac23        Fri Feb 23 12:57  still logged in
reboot    ~                Fri Feb 23 12:16
```

## 13.7 プロセスの動作状態を確認したい (ps)

システム上で起動されているプロセスの情報を知りたい時には `ps` コマンドを使います。下の例では `ps` コマンドに `a,x,g` という3つのオプションを与えています。これは、自分以外のユーザが起動したプロセスの情報 (`a`) や、デーモンプロセスなどのターミナルから起動されていないプロセスの情報 (`x`)、ゾンビ状態のプロセスの情報 (`g`) もあわせて表示させる指定です。この3つのオプションを同時に指定することによって、現在のシステムで起動しているすべてのプロセスの情報を表示させることができます。

```
% ps axg
 0 ?  R N 203:42  (kernel idle)
 1 ?  SW   0:00 /usr/etc/init -xx
 3 ?  SW   0:01 /usr/etc/kern\_loader -n
22 ?  S    0:00  (nmserver)
89 ?  SW   0:00 /usr/etc/syslogd
96 ?  SW   0:00 /usr/etc/portmap
101 ? SW   0:00 /usr/etc/nibindd
102 ? SW   0:01 /usr/etc/netinfod local
106 ? SW   0:00  (ypbind)
```

またフラグ `u` を使うことによりユーザを含むより詳細な情報を表示することができます。

```
% ps axu
USER      PID  %CPU %MEM  VSIZE  RSIZE  TT  STAT  TIME  COMMAND
root      515   8.6  4.5  2.21M  744K  p2  R N   0:00  ps axu
root       0   7.9 14.0 16.6M  2.23M  ?   R N 435:43  (kernel idle)
.....
```

## 13.8 プロセスの優先度を変更したい (nice)

使用しているプロセスの優先度を変更したい時には、`nice` コマンドを使用します。一般的にはプロセスの優先順位を下げるために使用しますが、スーパーユーザ権限の時だけは優先順位を上げることができます。

```
% nice who
# nice --10 date (root はマイナスの値を使うことができる)
```

## 13.9 プロセスを終了させたい (kill)

```
1.kill 1234
2.kill -9 1234
3.kill -KILL 1234
```

現在使用しているプロセスを強制的に終了させていときには、`ps` コマンドの出力からそのプロセス番号を得て `kill` コマンドで終了させます。そのプロセスの所有者でないユーザが `kill` コマンドを使用しても無効になりますが、スーパーユーザ権限ではすべてのプロセスに対して終了させる権限を持っています。最初の例は一般的な使用方法ですが、この方法で強制終了できない場合は、`kill` コマンドの 9 番目のシグナル (KILL) を指定して実行します。2 番目の例と 3 番目の例は同意です。

## 13.10 ネットワークに接続されているホストの状態を調べたい (ping)

LAN に接続されているホストの状態を調べる時には、`rup` コマンドを使用します。表示される数値は、このコマンドを実行した時点から見て過去 1 分間、5 分間そして 15 分間における負荷の平均値 (ロードアベレージ) 並びにシステムが起動してからの経過時間です。ロードアベレージの数値が大きければ大きいほど、そのシステムは負荷の高い状態、すなわち CPU の稼働状態が高いことを意味しています。

`rup` コマンドは、RPC (Remote Procedure Call: リモートプロシジャークール) を使用して各々のシステムの状態を調べています。従って、`rstatd` が起動できないシステムから表示データの取得が不可能なため、そのようなシステムの状態を表示することはできません (通常、`rstatd` は `inetd` から起動されるようになっています)。

```
% rup
zeta.cse.kyu      up           38 mins,  load average: 0.10, 0.05, 0.04
  zodiac20       up  1 day, 13:10,  load average: 0.20, 0.03, 0.02
zodiac23.cse     up           1:24,  load average: 0.06, 0.09, 0.09
beta.cse.kyu     up           22 mins,  load average: 0.38, 0.32, 0.21
zodiac02.cse     up  7 days, 12:27,  load average: 0.00, 0.01, 0.01
macross.cse.     up           21:19,  load average: 0.01, 0.01, 0.02
```

このように `rup` コマンドにオプションを指定しないと、ブロードキャストを利用して `rup` コマンドを実行するシステムが接続されている LAN (のセグメント: 同一支線) に収容されている全システムのデータを収集・出力します。

また `-l` オプションを指定した時は負荷の平均値を、`-h` オプションではホスト名で、`-t` ではシステムが起動してからの経過時間でソートします。

## 13.11 マシン名から IP アドレスを知りたい (nslookup)

`nslookup` を使って名前から IP アドレスへの変換ができていることを確かめることができま

す。

```
% /usr/etc/nslookup
Default Server:  scorpio.cse.kyutech.ac.jp
Address:  131.206.1.43

> kyu-cc.cc.kyushu-u.ac.jp
Server:  scorpio.cse.kyutech.ac.jp
Address:  131.206.1.43

res_mkquery(0, kyu-cc.cc.kyushu-u.ac.jp.cse.kyutech.ac.jp,
1, 1)
(省略)
-----
Name:  kyu-cc.cc.kyushu-u.ac.jp
Address:  133.5.9.1

>
```

## 14 ネットワークの状態を管理する

電子メールなどのコマンドや NFS によるファイルの共有などを見てもわかるように、UNIX システムはネットワークに依存して動作します。したがって、ネットワークの稼働状態は UNIX システムを利用するユーザの作業に大きな影響を与えますので、ネットワークの管理は大変重要となります。ここでは、ネットワークの状態を調査するためのコマンドについて解説します。

### 14.1 ネットワークの状態をチェックしたい (ifconfig)

システムのネットワークインターフェイスの状態をチェックしたり、ネットワークの動作にかかわる各種パラメータの設定値を確認する時には、ifconfig コマンドを使用します。

以下の出力例では le0 は動作中 (“UP”, ”RUNNING”) で、その inet アドレス (IP アドレス) は “131.206.45.38” にネットマスクは “ffffff00” に、ブロードキャストアドレスは “131.206.45.255” にそれぞれが設定されていることなどがわかります。

```
% /usr/etc/ifconfig -a
en0: flags=63<UP,BROADCAST,NOTRAILERS,RUNNING>
      inet 131.206.45.38 netmask fffffff0 broadcast 131.206.45.255
lo0: flags=869<UP,LOOPBACK,NOTRAILERS,RUNNING>
      inet 127.0.0.1 netmask ff000000

%
```

## 14.2 ネットワークをテストしたい (ping)

特定のホストに対しての接続をテストしたいときには、ping コマンドを使用します。ping コマンドは、特定のシステムに対して “ICMP ECHO\_REQUEST” を送信して、送信後、そのシステムからの応答の有無 (“ICMP ECHO\_REPLY” の返送の有無) を一定時間待った後に報告します。ping コマンドに対する応答がない場合には、その指定したシステムが現在停止しているか、あるいはそのシステムまでの途中のどこかにおいてネットワークの障害が発生している可能性が考えられます。

```
% ping zodiac20
PING zodiac20: 56 data bytes
64 bytes from 131.206.45.35: icmp\seq=0. time=19. ms
64 bytes from 131.206.45.35: icmp\seq=1. time=1. ms
64 bytes from 131.206.45.35: icmp\seq=2. time=1. ms
----zodiac20 PING Statistics----
6 packets transmitted, 6 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 1/4/19
```

## 14.3 ネットワークのトラフィック状態を知りたい (netstat)

ネットワークのトラフィック状態を知りたい時には、netstat コマンドを使用します。netstat コマンドは、ネットワーク関連データの構造の内容を表示するもので、代表的なオプションの使用例は、以下のようになります。

```
% netstat -i
(システムが起動してからのインターフェイスにおけるパケット転送の状態を表示)
Name Mtu Network Address Ipkts Ierrs Opkts Oerrs Coll
en0 1500 131.206.45 zodiac23.cse.ky 16272 0 12384 0 0
lo0 1536 loopback localhost 2008 0 2008 0 0
en0 1500 none none 16270 0 12386 0 0

% netstat -m (ストリームの情報を表示)
78/256 mbufs in use:
    2 mbufs allocated to packet headers
    67 mbufs allocated to socket structures
    3 mbufs allocated to routing table entries
    2 mbufs allocated to socket names and addresses
    2 mbufs allocated to interface addresses
    2 mbufs allocated to <mbuf type 15>

0/8 mbuf clusters in use
8 interface pages allocated
104 Kbytes allocated to network (70% in use)
```

```
0 requests for memory denied
0 requests for memory delayed
0 calls to protocol drain routines
```

% netstat -r (使用しているルーティングテーブルの状態を表示)

Routing tables

| Destination | Gateway            | Flags | Refs | Use  | Interface |
|-------------|--------------------|-------|------|------|-----------|
| localhost   | localhost          | UH    | 2    | 595  | lo0       |
| default     | 131.206.45.254     | UG    | 0    | 5283 | en0       |
| 131.206.45  | zodiac23.cse.kyute | U     | 17   | 6468 | en0       |

## 15 ファイルシステムの状態を管理する

何らかのトラブルが発生したことにより、ファイルシステムの整合性が損なわれてしまったり、最悪の場合ファイルシステムが完全に破壊されてしまうことが考えられます。このような場合、ファイルシステムの整合性を修復したり、破壊されたファイルシステムを一時的にシステムから切り離して修復作業を実施した後、再びシステムに追加して使用できるようにするための操作を行う必要があります。また、ファイルシステムがデータでパンク状態になるのを未然に防ぐためにファイルシステムの利用状況を監視したり、ファイルの種類を検査したりする必要があります。ここでは、これらの作業に利用されるコマンドについて解説します。

システムを使い続けていると、作業に必要なファイルの数やサイズがしだいに増加してきます。また、システムを利用する頻度が高ければ高いほど、システム障害によってデータが破壊された時のダメージが増してきます。ここではデータのバックアップやリストア、ディスクの検査といった維持に関するコマンドも解説します。

### 15.1 ファイルシステムの利用状況を確認したい (df)

ファイルシステムのディスク使用状況 (空き容量や使用量など) を知りたい時には、df コマンドを使用します。

```
% df
Filesystem          kbytes    used   avail capacity  Mounted on
/dev/sd0a            235953  189542   22815    89%    /
/private/vm/swapfile 235953  189542   22815    89%    /private/....
```

空き容量が少なくなった時には、システムがハングアップしたりすることがあるので、不要なファイルを他のファイルシステムに移したり、削除したりして空き容量を確保します。



## 15.2 ファイルシステムの整合性を検査したい (fsck)

ファイルシステムの整合性のチェックや不整合の修復を対話的に行う時には、`fsck` コマンドを使用します。一般的に、`fsck` コマンドを使用しなければならない状況に遭遇することはほとんどありません。システムがクラッシュ場合であっても、たいていの場合は起動時に自動的に修復されます。もし自動的に修復が不可能な場合であっても、コンソールに表示されるメッセージに従って `fsck` コマンドを実行するだけで修復が完了する 경우가ほとんどです。

| オプション例               | 意味   |
|----------------------|--|
| <code>fsck</code>    | UNIX ファイルシステムのチェックを行い、不整合が発見された場合には、指示を得てその修復を行います。          |
| <code>fsck -y</code> | UNIX ファイルシステムの整合性のチェックだけを行い、修復します。                           |
| <code>fsck -n</code> | コマンドがどのように実行されるのか、その実行過程の表示を行います。コマンドそのものは実際には実行されることはありません。 |

## 15.3 ファイルの内容を検査したい (file)

ファイルシステム中のファイルタイプを調べたい時には、`file` コマンドを使用します。

```
% file etc
etc:  directory
% file foo
foo:  ascii text
```

## 15.4 ハードディスクの管理をしたい

よくあるのはハードディスクのトラブルです。ハードディスクの調子が悪くなって来ると、カーネルは `/usr/adm/message` に ERROR を記録していきます。`message` を頻繁に確認すればいいのですが、ここでは `cron` を使って一日に一回チェックすることができる方法を紹介しましょう。

まず、`/usr/local/bin/checkmes` というシェルスクリプトを準備します。

List 17 `/usr/local/bin/checkmes`

---

```
1  #!/bin/sh
2  #
3  #      checkmes      check /usr/adm/message inculde ERROR or warning
4  #
5  #                  by E. S. Otabe
6  #                  st Sep 14, '93
7  #
8
9  messages=/usr/adm/messages
```

```
10 tmp=/tmp/checkmes.$$
11
12 if egrep 'ERROR|error|WARNING|warning' $messages > $tmp
13     then mail -s 'error on messages' root < $tmp
14 fi
15 rm -f $tmp
```

---

---

次に /etc/crontab.local に次の行を追加します。

List 18 /etc/crontab.local の一部

---

---

```
1 0 1 * * *      root    /bin/sh /usr/adm/checkmes >/dev/null 2>&1
```

---

---

これで、root 宛にチェックした内容が送られます。もちろん root のメールエイリアスは管理者にしておきます。

List 19 /etc/sendmail/aliases.local の一部

---

---

```
1 root: otabe
```

---

---

## 15.5 ファイルを探したい (find)

ある特定のディレクトリの下に存在するすべてのサブディレクトリにわたって、探索対象として指定されたファイルが存在するかどうかを調べる時には、find コマンドを使用します。さらに、ファイルが存在した場合の動作を指定することができますので、パス名を表示させたり、ファイルを削除することができます。

```
find (ディレクトリ名) -name (探索ファイル名) -print
find . -print -name (ディレクトリ名) -prune
find . -mtime +14 -print
find (ディレクトリ名) -name (探索ファイル名) -exec rm {} \;
find . -name "*.Z" -exec uncompress {} \;
find . -name \*.tex -exec grep -i kiyonaga {} \; -print
```

最初の例は、指定したディレクトリ以下に探索すべきファイルが存在するかどうか調べ、存在した場合には、そのパス名をすべて表示します。

2 番目の例は、カレントディレクトリ以下のファイル名をすべて表示しますが、指定したディレクトリ中のものは除外します。

3 番目の例は、15 日以上、ファイルの内容が変更されていないものをすべて表示します。

4 番目の例は、指定したディレクトリ中の探索ファイルを削除します。ここで、{ } \; は見付かったファイルを表しています。

5 番目の例は、圧縮されているファイルをすべて復元します。

6 番目の例は、grep を使って kiyonaga という語 (大文字、小文字区別無し -i) を検索しています。

## 15.6 ファイルの内容を 8 進数でみたい (od)

ファイルの内容の一部または全部を 8 進数でみたい時には od コマンドを使います。od コマンドはオプションにより 10 進数、16 進数の表示も可能になります。

| オプション名 | 意味             |
|--------|----------------|
| -b     | 各バイトを 8 進数表示   |
| -c     | 各バイトを ascii 表示 |
| -d     | 各語を 10 進数として表示 |
| -o     | 各語を 8 進数として表示  |
| -h     | 各語を 16 進数として表示 |

```
% od -ha table.dvi | head
0000000  f702  0183  92c0  1c3b  0000  0000  03e8  1b20
          w stx soh etx dc2  @  fs  ; nul nul nul nul etx  h esc  sp
0000020  5465  5820  6f75  7470  7574  2031  3939  362e
          T e X sp  o u t p u t sp  1 9 9 6  .
0000040  3033  2e30  383a  3135  3233  8b00  0000  0100
          0 3  .  0 8  :  1 5  2 3 vt nul nul nul soh nul
0000060  0000  0000  0000  0000  0000  0000  0000  0000
          nul nul nul nul nul nul nul nul nul nul nul nul nul nul nul
```

## 15.7 ファイルシステムをバックアップしたい (rdump, rrestore)

NeXT のファイルシステムをバックアップする方法の 1 つとして、2F 計算機室の 8mm ストリーマを使用する方法を示します。

1. まず、計算機委員会のスタッフ (staff@cse) に scorpio22 のアカウントを作成してもらいます。その際、スタッフにパスワードフィールドを用意しておくこととアカウントの作成が容易になります。
2. アカウントができれば 8mm テープを持って 2F 計算機室に行きます。ストリーマ (EXBYTE) の電源を入れた後、NeXT の電源を入れ、そして自分のアカウントで入ります。
3. rlogin で自分の dump したいマシンに入ります。そこから rdump をつかって scorpio22 に dump します。

例

```
% rlogin zodiac20
% su
# df
# rdump 0usf 500000 scorpio22:/dev/rxt0 /dev/sd1a
```

4. 必要であれば `rrestore` を使って確認します。

```
% cd (安全なディレクトリ)
% rrestore ivf scorpio22:/dev/rxt0
```

5. バックアップ作業が終わったら、NeXT の電源を落として EXBYTE の電源も落とします。

また、バックアップしたファイルを戻したい時には次のようにします。

1. scorpio22 からシステムを戻したいマシンに入ります。

```
% rsh zodiac20
```

2. ディレクトリを移動させて、`rrestore` コマンドを使って、バックアップファイルに戻します。

```
% cd /home
% rrestore rf scorpio22:/dev/rxt0
```

### 15.7.1 テープの操作をしたい (mt)

`mt` コマンドを用いてテープの巻き戻しや、早送り、取り出しを行うことができます。EXABYTE の場合にはデバイス名は以下の二つがあります。

`/dev/rxt0` 最初の `r` はリワインド (rewind、巻き戻し) を表しており、処理が終ると必ず巻き戻しをしてテープの最初に戻します。

`/dev/nrxt0` 最初の `nr` はリワインド (no-rewind、巻き戻しなし) を表しており、処理が終っても巻き戻し動作はしません。

主な mt コマンドの使い方は以下のとおりです。

```
% mt -f /dev/rxt0 fsf 1 (まちがい)
% mt -f /dev/nrxt0 fsf 1 (正しい)
% mt -f /dev/nrxt0 bsf 2
% mt -f /dev/rxt0 rewind
% mt -f /dev/rxt0 offline
```

一番最初は間違いの例です。-f はデバイスを指定します。fsf 1 は一つのファイルを飛び越して進めるコマンドです。たしかにテープは早送りされて一つファイルを進めますが、デバイスが rxt0 ですから、コマンドが終了すると巻き戻しをしてテープの最初に行ってしまいます。

二番目は正しいコマンドで、一つのファイルを飛び越します。

三番目は二つ前のファイルに移動します。

四番目は巻き戻しをします。

五番目はテープを巻き戻して、取り出します。

詳しくは man mt でマニュアルページを参照して下さい。

## 15.8 ファイルシステムの自動バックアップをしたい

ディスクに余裕があるときにはパーティションを分けておいて、非常に重要なファイルを cron(5.2.1節) を使って自動的にバックアップすることができます。たとえば次の例では /home/aquarius10 以下のファイルを /home/backup/aquarius10 の形でバックアップします。gnutar を使うのは 255 文字を越える長いファイル名を取り扱うことができるからです。

List 20 /usr/local/bin/backuphome

---

```
1 #!/bin/sh
2 #
3 # backup /home/aquarius10 to /home/backup/aquarius10
4 #
5
6 rm -rf /home/backup/aquarius10
7 cd /home
8 gnutar cf - aquarius10 | (cd /home/backup; gnutar xf -)
```

---

## 15.9 ファイルを移動したい (tar)

tar コマンドは本来磁気テープにファイルを読み書きするためのコマンドですが、複数のファイルを一つのファイルに固める時に使うことが多いコマンドです。この tar コマンドを使ってファイルを移動させるのに利用することができます。

```
% cd fromdir; tar cf - . | (cd todir; tar xf -)
```

最初の tar で fromdir の下のファイルを固めて標準出力に出します。次のところでは todir に移ってから tar で標準入力から取り込み展開します。ファイルの移動は cp -r も考えられますが、リンク情報を扱えない、日付がコピーした日付につけ変わってしまうなどの問題があり、通常使いません。

標準の tar は 255 文字を超えるファイル名を取り扱うことができません。付属の gnutar は取り扱うことができますし、同時に圧縮をかけることもできます。

```
% gnutar cfz foo.tar.gz foo
```

この例では foo ディレクトリをまとめ、圧縮します。

## 16 コマンドを知る

UNIX の管理には実に膨大な知識を必要とします。またコマンドも多岐にわたっていてそれぞれの機能も非常に多く覚えるのは大変です。通常の UNIX では man コマンドを多用してコマンドの内容を知ります。OPENSTEP ではこの他に Librarian.app があり、それによって調べることができるので非常に便利です。この章ではこの二つについて解説します。

### 16.1 コマンドを man により調べたい

あるコマンドの動作がわからない時には man コマンドにより調べるのが一般的です。

```
% man who (who コマンドについて調べる)
Reformatting page.  Wait... done

WHO(1)                UNIX Programmer's Manual
WHO(1)

NAME
  who - who is on the system

SYNOPSIS
  who [ who-file ] [ am I ]

DESCRIPTION
  Who, without an argument, lists the login name,      terminal
  name, and login time for each current UNIX user.
```

Without an argument, who examines the /etc/utmp file to obtain its information. If a file is given, that file is examined. Typically the given file will be /usr/adm/wtmp, which contains a record of all the logins since it was created. Then who lists logins, logouts, and crashes since the creation of the wtmp file. Each login is listed with user name, terminal name (with '/dev/' suppressed), and date and time. When an argument is given, logouts produce a similar line without a user name. Reboots produce a line with 'x' in the place of the device name, and a fossil time indicative of when the system went down.

With two arguments, as in 'who am I' (and also 'who are you'), who tells who you are logged in as.

#### FILES

/etc/utmp

#### SEE ALSO

getuid(2), utmp(5)

%

この例では who コマンドについて調べてあります。(1) というのは man の章の番号です。最初に文法が書いてあり、説明があります。ヒントとして重要になるのは最後の “FILE” や “SEE ALSO” のところです。これらで関連のファイルやコマンドを知ることができます。

## 16.2 コマンドを Librarian.app により調べたい

Librarian.app を使うことによって、フォルダ内のファイル名やファイル内の単語やフレーズを検索することができます。さらに /NextLibrary/Documentation の下の ManPages(man コマンドで参照される場所) や NextAdmin(管理マニュアル) や NextDev(開発環境マニュアル) をターゲットとすることでオンラインマニュアルとしても使用できます。図で「NextAdmin」というような本の絵が入っていますが、これは Workspace で /NextLibrary/Documentation/NextAdmin を表示させて、これをドラッグしてこの Bookshelve に近付けると張り付けることができ、ターゲットの一つになります。Bookshelve はセーブします。/NextLibrary/Bookshelves にもいくつか便利な Bookshelve が準備されているのでそこからコピーするのもいいでしょう。OPENSTEP4.1J からは日本語のオンラインマニュアルがかなり充実して便利になりました。また他のアプリケーションからもサービスで Librarian.app を立ち上げて検索できるので便利です。図の例では、ManPages をターゲットにして ftp を調べています。ftp を検索すると ftp を含む語を全部検索してくれます。したがって、ftpd 等のような関連のコマンドを知ることができます。Librarian.app は Index のついたターゲットについては大変高速に検索することができます。



図 13: Librarian.app の画面

## 17 アプリケーションソフトをインストールする (Installer.app)

NeXTのアプリケーションソフトをインストールする時は、/NextAdmin/Installer.appを使用します。インストールしたいアプリケーションソフトがfoo.pkgのようにパッケージファイルになっていればInstaller.appに対応しており、これを用いることにより容易にインストール作業が行うことができます。大抵は付属のマニュアルにインストールの詳細が書いてありますが、pkgファイルをダブルクリックしてInstaller.appを立ち上げて、インストールするだけです。

インストールされているアプリケーションを削除するのは、他のOSでは大変な作業の一つでした。OPENSTEPでは削除するにもInstaller.appを使います。正規にインストールされたパッケージは/NextLibrary/Receipts/にインストールされたことを記録しています。ここにあるpkgファイルをダブルクリックしてInstaller.appを立ち上げ、極めて安全にインストールされていない状態に戻すことができます。

### おわりに

ここで説明してある管理方法でおおよその設定ができるようになると思います。時間の無い方はこの文書だけで十分かも知れません。しかしまだ管理にはいろいろなことがあります。それらは多数出版されている本を参考にして下さい。/NextLibrary/Documnetsの下には日本語になった文書も多いのでよく読まれるといろいろと重要な情報を得ることができるでしょう。



また計算機の世界は日進月歩で進歩が非常に早いのでどんどん変わっていきます。ネットワークを通じた情報を注意深く読んで最新の情報を知っておくことが重要になります。

本書が計算機管理の一助になれば幸いです。

## 参考文献

- [1] OPENSTEP ネットワーク & システム管理, NeXT Computer Inc.  
この本をしっかりと読めばシステム管理は問題無いはずですが、学科特有の問題もあるので注意が必要です。
- [2] SUN システム管理, 下山 智明他, アスキー出版社, 1991 年  
SunOS4.1.x の管理について非常にこと細かく記述しており、同じ BSD 系ということで参考になります。
- [3] The Unix Hyper Text 上下, 技術評論社