

できる

dekiru PRO series

アパッチ

Apache Webサーバー



辻 秀典・渡辺高志・鈴木幸敏&
できるシリーズ編集部

改訂版

Version
2.4/2.2/2.0
対応

人気書籍がバージョンアップ

“とりあえず動く”から
“ちゃんとわかる”へ

CentOS/Ubuntu/Windows/Mac OS X を網羅!

インプレスジャパン

「できる」、「できるシリーズ」は、株式会社インプレスジャパンの登録商標です。
Linux は、Linus Torvalds の米国およびその他の国における登録商標です。
Red Hat は、Red Hat, Inc. の米国およびその他の国における商標です。
Microsoft、Windows は、米国 Microsoft Corporation の登録商標です。
その他、本文中の製品名およびサービス名は、一般に各開発メーカーおよびサービス提供元の商標または登録商標です。
なお、本文中には ™ および ® マークは明記していません。

Copyright © 2013 Hidenori Tsuji, Takashi Watanabe, Yukitoshi Suzuki and Impress Japan Corporation. All rights reserved.

本書の内容はすべて、著作権法によって保護されています。著者および発行者の許可を得ず、転載、複写、複製等の利用はできません。

まえがき

日本のインターネット利用者が1億人弱といわれ、ほとんど誰もがなんらかの形でインターネットを利用しています。しかも、その大半がWebに関わる利用ですから、Webページを提供しているWebサーバーは、実は私たちにとって、とても身近な存在なのです。

Apache HTTPサーバー（以降Apache）は15年以上もの間、世界で最も使われ続けられているWebサーバーのソフトウェアです。インターネット上の6割前後のシェアを占めているという統計結果があります。つまり、世界中にあるWebページの数多くがApacheによって提供されているということになります。

Apacheは機能が豊富な上に、柔軟に機能拡張もできます。1995年3月18日に公開されたApache 0.2は、もともとNCSAのWebサーバーのパッチ（修正モジュール）という位置づけでリリースされました。それが、時間の経過とともに改良され、新機能の追加と問題の修正だけでなく、さまざまなOSで利用できるようになりました。Apacheは「A patch」（パッチを当てた）という生い立ちと、アメリカの先住民族の部族であるApache族から名付けられたといわれています。

Apacheのさらなる特徴は、UNIX系OSやWindowsの複数のOSで動作するということと、ソースが公開されているオープンソースソフトウェアかつフリーソフトウェアとして提供されていることです。それに加えて、柔軟に新しい機能を追加して利用できるという、今も引き継がれている当初の思想が、Apacheが多くの人に受け入れられている理由でしょう。

本書は、このApacheを使ってWebサーバーを構築し、設定を行なって、Apacheのさまざまな機能を活用することを目的としています。Webサーバーは難しそう、Apacheは難しそうというイメージがあるかもしれませんが、しかし、本書は、よくあるリファレンス形式の書籍ではなく、一歩ずつApacheへの理解を深められるような作りになっていますので、より多くの方がApacheを利用するきっかけとなれば幸いです。

最後に、Apacheプロジェクトの開発者とプロジェクトを支えている方々、オープンソース文化とUNIX文化を支えている方々、そしてインプレスジャパンの出町さんと永山さん、編集者の高橋さんに感謝いたします。

2012年12月吉日 著者を代表して

辻 秀典

本書の読み方

本書は Apache HTTP サーバー (Apache) を使って、Web のサーバーを構築し、さまざまな設定によって動作を変更する方法を解説しています。

Web サーバーに必要な基礎知識やインストール方法、基本設定、起動と終了、コンテンツを見せるための設定、ユーザーごとのディレクトリを作って公開する方法、仮想ホストにより複数のサーバーに見せる方法、アクセス元やパスワードでアクセスを制限する方法、言語の自動切り替えや HTTP のリダイレクト、CGI のしくみと使い方、SSI の使い方、モジュールによって Apache を拡張する方法、SSL で安全に通信する方法、ログをチェックしカスタマイズしてアクセス状況を的確に把握する方法、サーバーの状態を把握して安定して運用する方法などを説明しています。設定方法を学ぶ中で、TCP/IP や HTTP による通信のしくみも理解できる構成になっています。

本書では Linux などのサーバー OS について、Apache の操作に必要な範囲で解説しています。ただし、インストールや基本操作、サーバー自身のインターネットへの公開方法などについては割愛しています。サーバー OS の基本操作については『できる PRO CentOS サーバー』などの入門書を読んでから本書に進むことをおすすめします。

本書の前提条件

本書ではルーターによって LAN が構成され、インターネットに接続されていることを前提にしています。インターネットに接続していない場合でも Apache をテストすることはできますが、Web サーバーを公開することはできません。また、ソフトウェアをダウンロードするにはインターネットに接続しているか、インターネットに接続したマシンからファイルを転送する必要があります。

本書では、Linux ディストリビューションの CentOS でパッケージとして用意されている Apache を使う場合を中心に、Ubuntu や Windows、OS X (Mac) の場合も併記して説明しています。環境によって操作や設定が異なる部分がありますが、補足説明を付けたので、読み替えて利用してください。

また、レンタルサーバーの Apache を設定する場合は、そのシステムごとに、基本設定やユーザーの設定できる範囲が異なります。各レンタルサーバーの仕様に合わせて利用してください。

コマンドの表記

本書では、コマンドラインでの操作方法を次のように表現しています。

操作の履歴はグレーで表現しています。
直前の操作の確認にお使いください。

入力しなければならないコマンドは太字で表現しています。

```
[root@server ~]# cd /usr/local  
[root@server local]# cd /  
[root@server //]#
```

コマンドプロンプトから記述しています。カレントディレクトリの確認やユーザーの確認にお使いください。

コマンドの最後に **↵** が表示されているときは、コマンドの入力後に **Enter** キーを押してください。

ご質問をお送りいただく前に

本書の内容についてご質問をお送りいただく前に、弊社の書籍紹介ページ<http://www.impressjapan.jp/books/3339>からとどれる「正誤表」をご確認ください。これまでに判明した正誤があれば、この書籍紹介ページに「ダウンロード・お詫びと訂正」ボタンが表示されます。このボタンをクリックすると、正誤表が表示されます。

本書の内容に関するご質問方法と免責事項

ご質問は返信用切手を同封した封書もしくはメールにてお受けいたします（電話、FAX でのご質問には対応していません）。本文に、書籍名、ご質問のページ番号、ご質問内容、パソコンのメーカー名および機種名、増設した周辺機器、ネットワーク環境をできる限り詳細にお書きの上、お送りください。

ただし、お客様固有の環境に依存するご質問や現象が確認できない場合、インターネット上に公開されていた情報が更新されていて代わりとなる情報が見当たらない場合など、確実な解決方法をご提示できないことがあります。また、回答作成のための調査に時間がかかることもあり、回答期限のお約束はできません。本書の内容およびご質問の回答は、お客様の問題解決を保証するものではないことを、あらかじめご了承ください。

●封書でのご質問の場合

〒102-0075

東京都千代田区三番町20番地

株式会社インプレスジャパン できるシリーズ編集部

「できるPRO Apache Webサーバー 改訂版 Version 2.4/2.2/2.0対応」質問係

●メールでのご質問の場合

宛先：info@dekiru.impress.co.jp

件名：「できるPRO Apache Webサーバー 改訂版 Version 2.4/2.2/2.0対応」の質問

まえがき	3
本書の読み方	4

第1章 ApacheとWebを知ろう 15

1-1 Apache ってなに? <Apache HTTPサーバー>	16
一番使われているWebサーバー	16
バージョン2.0、2.2、2.4の違い	17
1-2 URLは何を表しているの? <URLの意味>	18
URLは複数の指定の集合体	18
WebブラウザはURLを分解してアクセスする	19
1-3 ホスト名ってなに? <名前解決>	20
ホスト名とドメイン名	20
ホスト名とIPアドレス	21
DNSで名前解決する	21
1-4 DNS ってなに? <DNSのしくみ>	22
DNSは階層構造で世界をカバーする	22
ダイナミックDNS	23
1-5 ネットワークのしくみを知ろう <インターネットとLAN>	24
LANはネットワークの最小単位	24
インターネットはネットワークの集合体	24
直接通信できるのはセグメントの中だけ	25
NATでLANとインターネットを中継	26
ルーターがつながって世界をカバー	27
1-6 サーバーとどう通信するの? <IPアドレス>	28
2種類のIPアドレス	28
IPv4のアドレス	29
IPv6のアドレス	30
URLをIPアドレスで指定する	30
セグメントはネットマスクで表す	31
1-7 IPアドレスはどう決めるの? <DHCP>	32
IPアドレスで互いを指定する	32
DHCPを使えば自動で割り当てられる	33
サーバーは固定IPでIPアドレスを指定する	33
STEP UP <よく使われているWebサーバー>	34

2-1	インストールする環境を用意しよう <必要なハードとソフト>	36
	必要なハードウェア	36
	OSと構成	37
	本書で使う環境	37
2-2	UNIX系OSにインストールするには <UNIX系OSへのインストール>	38
	本書で扱うUNIX系OS	38
	OSで用意されたものと自分でソースからコンパイルするもの	39
	CentOSへのインストール	39
	Ubuntuへのインストール	39
	Macでは標準でインストールされている	39
	OSで用意されているものをインストールする場合	40
	パッケージからインストールする (CentOSの場合)	40
	パッケージからインストールする (Ubuntuの場合)	41
	ソースからインストールする場合	42
	Apache 2.2や2.0をソースからインストールする	42
	Apache 2.4をソースからインストールする	46
2-3	Windowsにインストールするには <Windowsへのインストール>	52
2-4	Apacheを起動しよう <起動と停止>	56
	デーモンとして動く	56
	CentOSで用意されているものをインストールした場合	57
	Ubuntuで用意されているものをインストールした場合	57
	ソースからインストールした場合	58
	Macの場合	59
	Windowsの場合	59
2-5	アクセスしてみよう <リクエストとレスポンス>	60
	サーバーにリクエストを送る	60
	サーバーからレスポンスが返る	61
	クライアントからアクセスできるようにしてテストしよう	62
	STEP UP <Apacheのバージョンを最新に保つ>	64

3-1	設定ファイルを見てみよう <httpd.conf>	66
	テキストファイルで設定する	66
	設定ファイルは分割される	67

CentOSの場合	67
ソースからインストールした場合、Windowsの場合	67
Ubuntuの場合	68
Macの場合	68
設定ファイルを確認する	69
3-2 設定ファイルの構造を知ろう <セクションコンテナ>	70
指定の範囲を決める	70
設定の優先順位	71
ディレクトリ単位で指定する—<Directory>	72
ファイル単位で指定する—<Files>	72
パス名で指定する—<Location>	72
ほかの設定ファイルを読み込む—Include	73
3-3 初期設定をしよう <サーバー自身の情報>	74
エラー時に表示させるサーバーの情報が必要	74
CentOSで用意されているApacheの場合	75
Ubuntuで用意されているApacheの場合	76
ソースからインストールした場合、MacやWindowsの場合	78
3-4 HTMLファイルを置こう <ドキュメントルート>	80
ドキュメントルート以下にコンテンツを置く	80
ファイルを転送するには	83
3-5 設定を反映しよう <Apacheの再起動>	84
CentOSで用意されているApacheの場合	84
Ubuntuで用意されているApacheの場合	84
Macの場合	85
ソースからインストールした場合	85
Windowsの場合	85
3-6 アクセスの記録を確認しよう <ログファイル>	86
サーバーへのアクセスはログファイルに記録される	86
3-7 インデックスファイルを設定しよう <DirectoryIndex、Indexes>	90
index.htmがインデックスファイルに使われていないことを確認する	90
インデックスを設定する	91
3-8 ほかの場所にあるファイルを見せるには <Alias>	92
ドキュメントルートの中にあるファイルを見せる	92
3-9 文字コードの設定を変更しよう <AddDefaultCharset>	94
文字コードを指定するしくみ	94

第4章 公開ディレクトリを追加しよう 97

4-1 ユーザーごとの公開ディレクトリを用意しよう <ユーザーディレクトリ> 98

ユーザーごとのディレクトリを公開できる	98
ユーザーディレクトリの設定 (CentOSで用意されているApache)	99
ユーザーディレクトリの設定 (Ubuntuで用意されているApache)	100
ユーザーディレクトリの設定 (ソースからインストールしたApache 2.2/ 2.4)	101
ユーザーディレクトリの設定 (ソースからインストールしたApache 2.0)	102
ユーザーディレクトリを作る (UNIX系OSの場合)	103
ユーザーディレクトリを作る (Windowsの場合)	104
ユーザーディレクトリにアクセスする	105
ユーザーごとに許可や禁止をするには	105

4-2 ディレクトリごとの設定ファイルを使うには <.htaccessファイル> 106

4-3 複数のサーバーに見せるには <仮想ホスト> 108

1台のサーバーに複数のホスト名を付ける	108
名前ベースの仮想ホスト	109
IPベースの仮想ホスト	109
仮想ホストを設定する (CentOS、ソースからインストールしたApache 2.0)	110
仮想ホストを設定する (Ubuntu)	111
仮想ホストを設定する (ソースからインストールしたApache 2.2 / 2.4)	112
仮想ホストにアクセスする	113

4-4 仮想ホストに別名を付けるには <ServerAlias> 114

IPベースの仮想ホストに元のホスト名を付けるには	115
--------------------------	-----

4-5 ホスト名を指定しないブラウザに対応するには

<ServerPathによる仮想ホストの識別> 116

パス名で仮想ホストを判断	116
--------------	-----

STEP UP <HTTPの通信内容の確認方法> 118

第5章 アクセスを制御しよう 119

5-1 アクセス元を制限するには <アクセス制限> 120

アクセス元に対して禁止や許可を設定する	120
特定のアクセス元に対してアクセスを禁止する	121
特定のアクセス元にだけアクセスを許可する	122
アクセス元の指定方法	123

Apache 2.4でアクセスを制限する	124
Apache 2.4でのアクセス制御方法	125
5-2 Webブラウザの種類を制限するには <Apacheの環境変数>	126
リクエストのメッセージヘッダーを見て値を設定する	126
5-3 リンクを制限するには <リファラー>	128
リンク元の情報もサーバーに送られる	128
5-4 エラーメッセージを変更するには <ステータスコード>	130
レスポンスにはステータスコードが付く	130
5-5 パスワードをかけるには <基本認証>	132
登録ユーザーだけアクセスさせる	132
認証のしくみ	132
パスワードを指定する	134
5-6 認証してグループ単位で許可するには <ユーザーグループ>	136
5-7 アクセス制限と認証を組み合わせるには <Satisfy>	138
両方の条件に合ったときだけ許可する場合	138
どちらかの条件に合えば許可する場合	139
5-8 URLが変更になったときは <リダイレクト>	140
変更先のURLを返す	140
一時的にリダイレクトする場合	141
引っ越しのためリダイレクトする場合	142
URLのパターンでリダイレクトする場合	143
5-9 ページの言語を自動で切り替えるには <コンテンツネゴシエーション>	144
言語や文字コードの情報もサーバーに送られる	144
品質値で優先度を指定する	145
言語や文字コードを自動で切り替える	145
言語を切り替える	146
文字コードを切り替える	147
5-10 サーバーの種類を隠すには <ServerTokens、ServerSignature>	148
レスポンスにサーバーの情報が入っている	148
サーバーが作るページに表示される情報	149
5-11 データの種類を追加するには <AddType>	150
データの種類はMIMEタイプで表される	150
MIMEタイプを追加する	151
5-12 ロボットからのアクセスを制限するには <robots.txt>	152

収集されたくないディレクトリをrobots.txtで指定	152
HTMLのmetaタグでも指定できる	153
STEP UP <Apacheで使える認証方法>	154

第6章 CGIを使おう 155

6-1 CGI ってなに? <CGIの概要>	156
Apacheを通してプログラムを実行するしくみ	156
Perlなどでプログラムを作る	157
セキュリティに気を付けよう	157
6-2 CGIを使えるようにするには <ScriptAlias、AddHandler>	158
プログラムはダウンロードできない場所に置く	158
6-3 CGIを使ってみよう <CGIのスクリプト>	160
6-4 ユーザーディレクトリでCGIを使うには <ScriptAliasMatch>	162
6-5 状況に応じて動作させるには <環境変数の利用>	164
6-6 クライアントからデータを受け取るには <GETとPOST>	166
Webブラウザからデータを送る	166
GETメソッドとPOSTメソッド	167
CGIでデータを受け取る	168
6-7 クライアントにデータを覚えさせるには <Cookie>	170
メッセージヘッダーに値を付けて返す	170
CGIでCookieを使う	172
6-8 CGIでページにフィルターをかけるには <ActionによるCGIの呼び出し>	174
6-9 エラーになったCGIの通信を確認するには <ScriptLog>	176
6-10 SSIを使えるようにするには <AddOutputFilter>	178
ページの一部の内容を入れ替える	178
6-11 ファイルやコマンドの出力を組み込むには <include、exec>	180
ファイルの内容を組み込む — include file="～"	180
コマンドの出力を組み込む — exec cmd="～"	181
CGIの出力を組み込む — exec cgi="～"	182
条件で表示を変えるには	183
STEP UP <Perl以外のスクリプト言語>	184

第7章 モジュールで拡張しよう 185

7-1	モジュールってなに？ <Apacheのモジュール>	186
	モジュールで機能を追加できる	186
	モジュールの追加方法は2種類ある	187
	組み込まれているモジュールを調べる	187
7-2	モジュールを追加するには <モジュールの追加とコンパイル>	188
	インストールされているモジュールを有効にする場合 (CentOS)	188
	インストールされているモジュールを有効にする場合 (Ubuntu)	189
	OSで用意されているモジュールをインストールする場合 (CentOS)	190
	OSで用意されているモジュールをインストールする場合 (Ubuntu)	191
	DSOのモジュールをコンパイルする場合	192
	組み込みのモジュールを指定する場合	194
7-3	データを圧縮して送るには <mod_deflate>	196
	自動的に圧縮と復元が行なわれる	196
	mod_deflateを使う	197
	そのほかの圧縮設定	197
7-4	違うURLでアクセスさせるには <mod_rewrite>	198
7-5	PHPを使うには <mod_php>	200
	PHPをApacheの中で動かすmod_php	200
	OSで用意されているPHPをインストールする場合 (CentOS)	201
	OSで用意されているPHPをインストールする場合 (Ubuntu)	202
	PHPをソースからインストールする場合	203
	PHPの動作を試す	207
7-6	動的なコンテンツをキャッシュするには <mod_cache>	208
	コンテンツをキャッシュして高速化する	208
	PHPのコンテンツをキャッシュさせる	209
7-7	リバースプロキシを立てるには <mod_proxy>	210
	リバースプロキシの役割	210
	リバースプロキシを立てる	211
7-8	そのほかのApacheモジュールを知ろう <さまざまなApacheモジュール>	212
	mod_security : Webアプリケーションファイアウォール	212
	mod_ext_filter : 外部コマンドでコンテンツをフィルター	212
	mod_speling : ファイル名のスペルミスを修正	213
	mod_qos : リクエストのQoS制御	213
	mod_pagespeed : ページを最適化して表示を高速化	213
	STEP UP <ロードバランサで負荷を分散する>	214

第8章 SSLに対応しよう

215

8-1	暗号化されたWebサーバーを構築するには <SSL>	216
	SSL ってなに?	216
	SSLで高まるWebサイトのセキュリティ	217
	CAに認証してもらうしくみ	218
	本格的にSSLを利用したWebサーバーを運営するには	219
8-2	ApacheからSSLを使えるようにするには <mod_ssl>	220
	Apacheのインストール方法により異なる	220
	CentOSで用意されているApacheの場合	221
	Ubuntuで用意されているApacheの場合	222
	ソースからインストールした場合	223
8-3	テスト用の証明書を作成するには <サーバー証明書>	224
	サーバー証明書の作成とApacheへの組み込み	224
	証明書を作る (CentOSのOpenSSLを使う場合)	225
	証明書を作る (OpenSSLコマンドを使う場合)	227
	Apacheを設定する	229
	クライアントからアクセスする	230
STEP UP	<WebDAVを使ってみよう>	232

第9章 ログを活用しよう

233

9-1	ログをチェックしよう <ログのチェックポイント>	234
	誰が何にアクセスしているかをチェック	234
	怪しいアクセスに注意する	235
	ログの解析ツールもある	235
9-2	アクセスログの記録項目を変えるには <LogFormat、CustomLog>	236
	フォーマットの指定方法	239
9-3	指定した条件でログを分けるには <環境変数による制御>	240
9-4	エラーログの記録項目を変えるには <LogLevel>	242
9-5	ログ中のIPアドレスをホスト名に変えるには <logresolve>	244
9-6	Cookieでユーザーを追跡するには <mod_usertrack>	246
9-7	ログを1日単位で切り替えるには <ログのローテーション>	248
	OSの標準機能を使う場合	248
	rotatelogsコマンドを使う場合	249

第10章 サーバーを管理しよう 251

10-1 セキュリティに気を付けよう <セキュリティ対策> 252

インターネットのサーバーはスキャンや攻撃を受ける 252

Apacheで気を付けるべきこと 253

その他の危険 253

10-2 動作状況をWebブラウザで確認するには <mod_status、mod_info> 254

動作状況を確認する — mod_status 254

設定情報を確認する — mod_info 256

10-3 サーバーの動作状況を確認するには <OSの観察> 258

CPUの負荷の状態を見る 258

プロセスの状態を見る 259

ネットワークカードの状態を見る 260

ネットワークがつながっているかを確認する 260

ネットワークに関する状態を見る 261

OSのログを見よう 261

10-4 Webブラウザのキャッシュを制御するには <キャッシュ> 262

キャッシュのしくみ 262

サーバーからの制御 263

ETagと負荷分散の注意点 263

HTMLファイルによる制御 263

10-5 性能を向上させるには <パフォーマンスチューニング> 264

不要な設定を削れば性能は向上する 264

ベンチマークでApacheの性能を測定しよう 265

Keep-Aliveの調整 266

待ち受けプロセス数の調整 266

ApacheのMPMを選ぶ 267

Webページの構成を見直す 267

CGIを見直す 267

サーバーの構成を見直す 267

STEP UP <Web環境の変化を知ろう> 268

付録 設定項目リファレンス 269

用語集 279

索引 283

第1章 ApacheとWebを 知ろう

「Apache HTTPサーバー」は世界で最も使われているWebサーバーのソフトウェアです。この章では、Apache HTTPサーバーをインストールする前に知っておきたいWebサーバーのはたらきをはじめ、URLの意味やインターネットにおけるアドレスの扱いなど、Webサーバーを設定する上で必要となる基礎的な知識を整理します。

●この章の内容

- 1-1 Apache ってなに？ 16
- 1-2 URLは何を表しているの？ 18
- 1-3 ホスト名ってなに？ 20
- 1-4 DNS ってなに？ 22
- 1-5 ネットワークのしくみを知ろう 24
- 1-6 サーバーとどう通信するの？ 28
- 1-7 IPアドレスはどう決めるの？ 32

Apache ってなに？

Apache HTTPサーバー

Apache HTTPサーバーは世界で最も使われているWebサーバーのソフトウェアで、インターネット上の6割前後を占めているという統計結果があります。つまり、世界中にあるWebページの数多くがApache HTTPサーバーによって提供されているということになります。

1995年に開発が始まったApacheは、UNIX系OSやWindowsという複数のOSで動作するということと、ソースが公開されているオープンソースソフトウェアかつフリーソフトウェアとして提供されていることが特徴です。最近では主に、バージョン2.2系と2.4系が使われます。

一番使われているWebサーバー

Apache HTTPサーバーは世界で一番使われているWebサーバーのソフトウェアです。Netcraft社 (<http://www.netcraft.com/>) の「Web Server Survey」という統計によれば、インターネット上の6割前後のWebサーバーがApacheを使用している、という統計結果が公表されています。その割合は年と共に変化していますが、Apacheは10年以上も、おおむね6割前後のシェアを維持してきました。

現在、Apacheプロジェクトには、Webサーバーだけでなく、さまざまなソフトウェアの開発プロジェクトがあります。そのため、特にWebサーバーのことをApache HTTPサーバーと表記しています。なお、本書ではApache HTTPサーバーのことをApacheと表記しています。

◆ Apache HTTP Server project
<http://httpd.apache.org/>

The screenshot shows the Apache HTTP Server Project website. The main heading is "Apache HTTP SERVER PROJECT". On the left, there are navigation links: "Essentials" (with sub-links: About, License, FAQ, Security Reports), "Download" (with sub-links: From a Mirror), "Documentation" (with sub-links: Version 2.4, Version 2.2, Version 2.0, Using libssl), "Get Support", "Get Involved" (with sub-links: Mailing Lists, Bug Reports, Developer Info), and "Subjects". The main content area features several news items:

- The Number One HTTP Server On The Internet**: A paragraph stating that Apache httpd has been the most popular web server on the Internet since April 1996.
- Apache httpd 2.4.3 Released** (2012-08-31): A paragraph announcing the release of version 2.4.3 of the Apache HTTP Server.
- Apache httpd 2.2.22 Released** (2012-01-31): A paragraph announcing the release of version 2.2.22 of the Apache HTTP Server.

バージョン2.0、2.2、2.4の違い

Apacheは1995年に開発が始まり、1995年3月18日にバージョン0.2が初めて公開されました。その後、同年11月に1.0.0がリリースされてから、広く使われるようになりました。

また、2002年4月5日に2.0.35が正式にリリースされて以来、バージョン1系と2系の併存が長く続きました。Apacheでは主たる2つのバージョンを併存させることで、安定性重視と新機能重視、それぞれの要望に対応してきました。2000年代からApacheでWebサーバーを運用してきたサイトなどでは、互換性や、それまでのノウハウの継承などの理由により、1.3系が使われてきました。ただし、バージョン1.3系は2010年1月の1.3.42のリリースを最後にサポートが終了しているため、バージョンアップが推奨されています。

バージョン2系でも、これまで2.0系と2.2系の2つのバージョンが主に使われてきました。その後、2012年2月21日に2.4.1がリリースされたことにより、2.2系と2.4系の2つのバージョンの体制に移行しました。これにより2.0系は最低限のサポートとなったため、2.2系もしくは2.4系に移行することが推奨されています。

併存するバージョンのどちらを選択するかについては、どちらがよい、悪い、というわけではなく、状況に応じて使い分けることとなります。

Apache 2.4	<ul style="list-style-type: none"> ●正式リリースのバージョン 2.4.1 が 2012 年 2 月 21 日にリリースされた ●複数の MPM (Multi Processing Module) を動的に読み込み可能になった ●より一層のパフォーマンスの改善が行なわれた
Apache 2.2	<ul style="list-style-type: none"> ●正式リリースのバージョン 2.2.0 が 2005 年 12 月 1 日にリリースされた ●認証や承認のためのモジュールが見直され、機能ごとにモジュールを分割するなどの変更が行なわれた ●メモリキャッシュやディスクキャッシュなど、動的キャッシュ機能が強化された
Apache 2.0	<ul style="list-style-type: none"> ●正式リリースのバージョン 2.0.35 が 2002 年 4 月 5 日にリリースされた ●基本部分をモジュール化する MPM が導入された。これにより Apache 本体も詳細にモジュール化され、効率的な動作が可能となった ●POSIX スレッドに対応し、マルチプロセス、マルチスレッドで複数のモジュールを実行できるようになった ●SSL に標準で対応した ●Apache 2.4 系のリリースにより、2.2 系もしくは 2.4 系へのアップグレードが推奨されている
Apache 1.3	<ul style="list-style-type: none"> ●最初のバージョン 1.3.0 が 1998 年 7 月 6 日にリリースされた ●2010 年 2 月 3 日に公開されたバージョン 1.3.42 が 1.3 系の最後のリリースとなった ●セキュリティ対応やバグ修正を期待したいのであれば、2.2 系もしくは 2.4 系へのアップグレードが推奨されている

1-2

URLは何を表しているの？

URLの意味

URLはWebページのアドレスとして使われていますが、実はひとつの情報ではなく、複数の情報を指定しています。Webサーバーのアドレスだけでなく、通信の方式、Webサーバー上から取得するファイルなどの情報が含まれます。

URLで指定されている意味を解釈してWebサ

ーバーにアクセスし、Webページを表示するのはWebブラウザの役目です。WebブラウザはURLを分解し、そこで指定されている複数の処理を行なっています。Webブラウザでアクセスするといっても、それだけのことが行なわれているのです。

URLは複数の指定の集合体

URLはWebページのアドレスとして使われていますが、ひとつの情報を表すアドレスではなく、複数の情報を指定するようになっています。

最初の「http」部分がプロトコル（通信の方式）体系を指し示す部分で「スキーム名」といいます。ここが「http」の場合にはWebページへのアクセスを指し示しています。ホスト名はサーバーのインターネット上のアドレスを指定するものです。それに続くポートは、サーバー上で提供されている機能へのアクセス番号です。Webページの場合は80番を使用することになっています。スキーム名がhttpのときは「:80」を省略しても80番ポートへのアクセスとして処理されます。逆に80番以外を意図的に指定する場合に使うものと思ったほうがいいでしょう。それ以降の部分が、Webサーバー上のどのファイルにアクセスをするかを指定したパス名になります。

URLで使える文字は半角英数字です。ただし、URLに含まれている「:」「/」のほかに「;」「?」「@」「=」「&」「#」は特別な意味を持つ文字として解釈されますので、URL上の好きな位置に使えるわけではありません。例えば「#」はページ内リンクとしてURLのパス内で使われます。

ここではURLの概要を説明しましたが、URLの定義はインターネットの標準化文書RFCの1738番（<http://www.ietf.org/rfc/rfc1738.txt>）に記載されています。

<http://www.dekiru.jp:80/about/index.html>

プロトコル

ホスト名

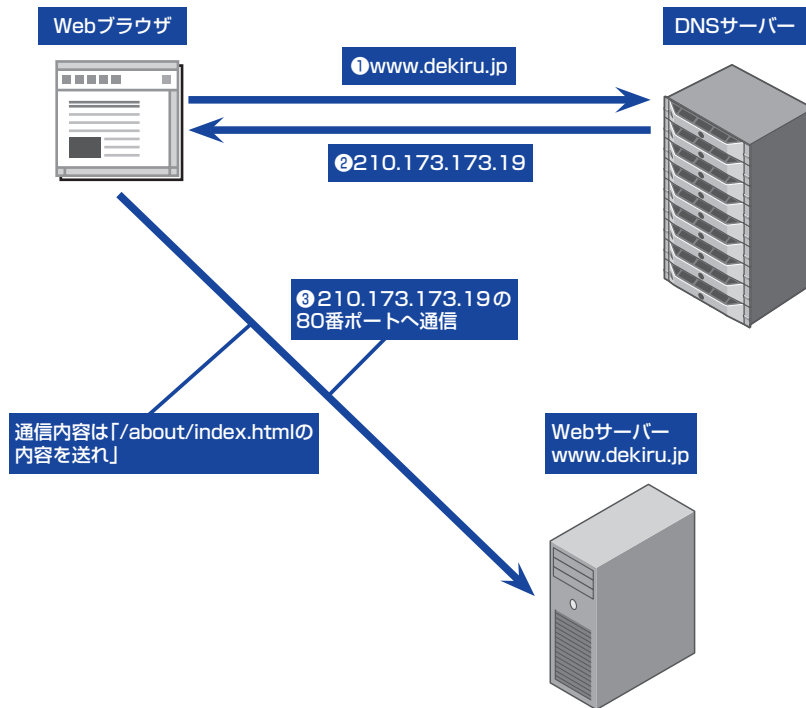
ポート

パス名

WebブラウザはURLを分解してアクセスする

「http」から始まるURLはWebページのアドレスを示しますが、そこで指定されている意味を解釈してWebサーバーにアクセスし、Webページを表示しているのはWebブラウザです。

WebブラウザはURLの定義にしたがってURLを分解し、そこで指定されている複数の処理を行なっています。「http://」と指定することでHTTPプロトコルで通信します。サーバー名を示すホスト名は、Webブラウザが動作しているOSを介してDNSサーバーに問い合わせをしてIPアドレスに変換し、そのIPアドレスの指定されたポート番号にhttpプロトコルにて通信を行ないます。そこでWebサーバーにパスで指定された情報を送信するように要求して、その結果を解釈して表示する、といった一連の動作を行なっています。



1-3

ホスト名ってなに？

名前解決

インターネット上にはIPアドレスのほかにホスト名というアドレスがあります。

ホスト名もIPアドレスもインターネット上の場所を指し示すものですが、通信を行なうときに使われるのはIPアドレスのほうです。しかし、IPアドレスは数字の羅列でしかありませんから、

IPアドレスとは別にホスト名という名前を持つことで、アドレスを分かりやすく表現します。

ホスト名でアドレスを指定しても、実際の通信ではホスト名はIPアドレスに変換されています。そのために、いつでもホスト名とIPアドレスを変換できるDNSというしくみがあります。

ホスト名とドメイン名

ネットワーク内にあるコンピュータや機器の所属を表すのがドメイン名です。そして、そのドメインのコンピュータなどに名前を与えるのがホスト名です。ホスト名は、FQDN (Fully Qualified Domain Name) とも呼ばれます。厳密にはFQDNのドメイン名の部分をドメイン部、コンピュータ名の部分をホスト部といいます。例えば「www.impress.co.jp」の場合、ドメイン名(ドメイン部)が「impress.co.jp」、コンピュータ名(ホスト部)が「www」です。

なお、Webサイトのホスト部が「www」なのは慣例で、Webサイトであることを人間に分かってもらうためでしかありません。ほかの名前でも問題はありません。

FQDN

www . impress . co . jp

ホスト部

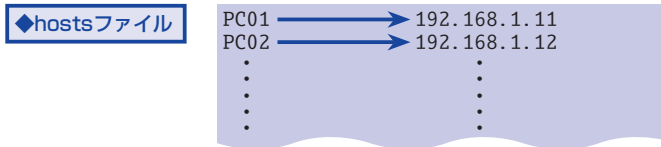
ドメイン部

ホスト名とIPアドレス

ホスト名もIPアドレスもインターネット上の場所を指し示すものですが、通信を行なうときに使われるのはIPアドレスです。しかし、IPアドレスは数字の羅列でしかなく、それ自体に分かりやすい意味を持たせることは難しいため、IPアドレスとは別に、ホスト名という名前を持たせることで、アドレスを分かりやすく表現します。

実際に通信を行なうときには、アドレスをホスト名で指定しても、これを内部でIPアドレスに変換して、IPアドレスを使って通信します。ホスト名とIPアドレスを対応付けるための方法は、いくつかあります。

一番シンプルな対応付けは、ホスト名とIPアドレスの対応をファイルに記載する方法です。これをhostsファイルといい、1行単位でホスト名とIPアドレスの対応を記述します。hostsファイルはWindowsでもUNIX系OSでも使われます。

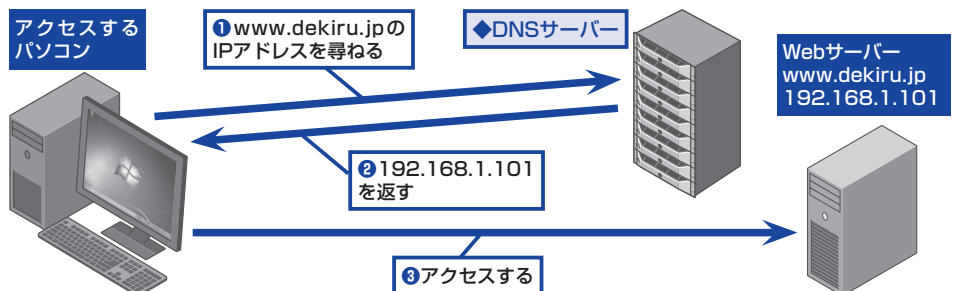


DNSで名前解決する

hostsファイルはシンプルですが、そこで定義されている情報をマシン間で共有するのは大変です。そこで、インターネットにはDNSというしくみがあります。

DNSの役割はホスト名とIPアドレスを関連付けてくれるデータベースです。ネットワーク経由で、ホスト名とIPアドレスのどちらかを問い合わせたとき、対応するもう一方を教えてくださいます。

ホスト名をIPアドレスに変換することを「名前解決」と呼びます。また、DNSに問い合わせることを「DNSを引く」と呼びます。名前解決にあたるホスト名からIPアドレスへの問い合わせを「正引き」、その逆のIPアドレスからホスト名を問い合わせることを「逆引き」と呼びます。



1-4

DNS ってなに？

DNSのしくみ

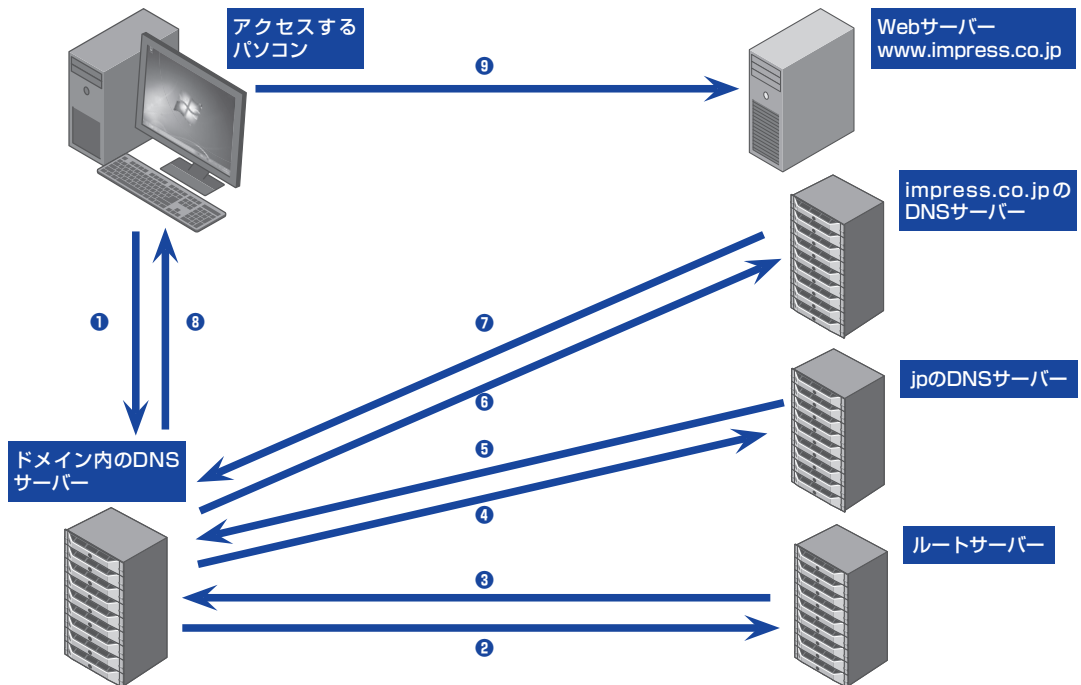
いつでもホスト名とIPアドレスを変換できるしくみがDNSであることを、**レッスン1-3**で説明しました。DNSはホスト名とIPアドレスを関連付けるデータベースですが、インターネットは世界中に広がる巨大なネットワークですので、それを1か所で管理するのは、ほぼ不可能です。

そこで実際には、それぞれのドメインを管理するDNSサーバーどうしが協調しあって動作するしくみになっています。

また、IPアドレスを固定的に使っていないインターネット環境でも名前解決ができる、ダイナミックDNSというしくみもあります。

DNSは階層構造で世界をカバーする

ホスト名とIPアドレスの変換は、それぞれの変換に対応する表をDNSサーバーが持つことで行なわれます。ただし、インターネットの世界は広大ですので、すべてのホスト名とIPアドレスの対応表を1か所で管理することは大変です。さらに、ホスト名とIPアドレスの対応は、いたるところで頻繁に更新されるものなので、これも1か所



ですべてを管理するのはとても無理な話です。実際には、各ドメインのDNSサーバーが、自分の管理するドメインのホスト名とIPアドレスの対応表を持っており、それらが階層構造をもって、互いに情報を交換し合うことで名前解決を行なっています。

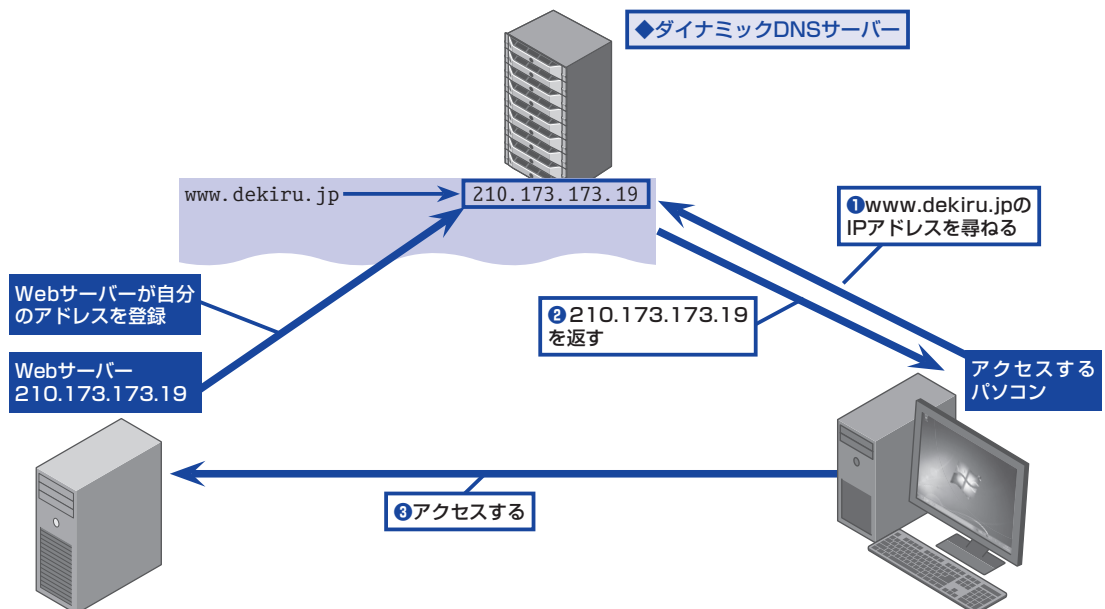
例えば、www.impress.co.jpにアクセスするとき、www.impress.co.jpのIPアドレスを問い合わせるために、「impress.co.jp」というドメインのDNSサーバーのIPアドレスを、「.jp」を管理するDNSサーバーに問い合わせます。これを行なうためには、まず最初に、DNSのルートサーバーといわれるDNSサーバーに「.jp」のDNSサーバーのIPアドレスを問い合わせます。このように階層化することで、DNSの分散管理を実現しています。

ダイナミックDNS

通常のインターネット接続環境では、インターネット接続に利用するIPアドレスが変化する、ダイミックIPが基本です。

ダイナミックIPの問題点は、IPアドレスが変化するために相手を特定できないことです。しかし、変化するIPアドレスを特定のホスト名と結びつけて管理できれば、IPアドレスが変化しても相手を特定できるようになります。

これができるようにするのが「ダイナミックDNS」というサービスです。ダイナミックDNSに登録したホスト名と変化するIPアドレスの同期を取ることで、ホスト名で相手を特定できるようにします。ダイナミックDNSを利用すれば、ダイナミックIP環境でもサーバーを公開できます。



1-5

ネットワークのしくみを知ろう

インターネットとLAN

ネットワークを使えば、機器間で情報の共有や管理ができます。自分が構成もしくは参加している特定の範囲で構成される最小単位のネットワークのことを、LAN (Local Area Network) といいます。

この最小単位のネットワークを組み合わせ

いくことで、より広い範囲でネットワークを利用できます。インターネットでは、TCP/IPという決まりのもとに、無数のネットワークが組み合わせることで、世界レベルのネットワークを構成しています。このときに重要になるしくみが、ルーティングです。

LANはネットワークの最小単位

ネットワークとは、コンピュータや、プリンタ等の機器を接続して、情報の共有や管理を行なうためのものです。ひとことにネットワークといってもいろいろな意味を持ちますが、特定の範囲で機器を接続して形作るネットワークのことをLAN (Local Area Network) といいます。LANは、構築や管理の際の単位となる基本的なネットワークのことを指します。

LANは限定された範囲で構成されるネットワークの最小単位ですが、自分が接続しているLANからインターネットを利用できるかどうかは、そのLANがインターネットに接続 (参加) しているかどうかによります。

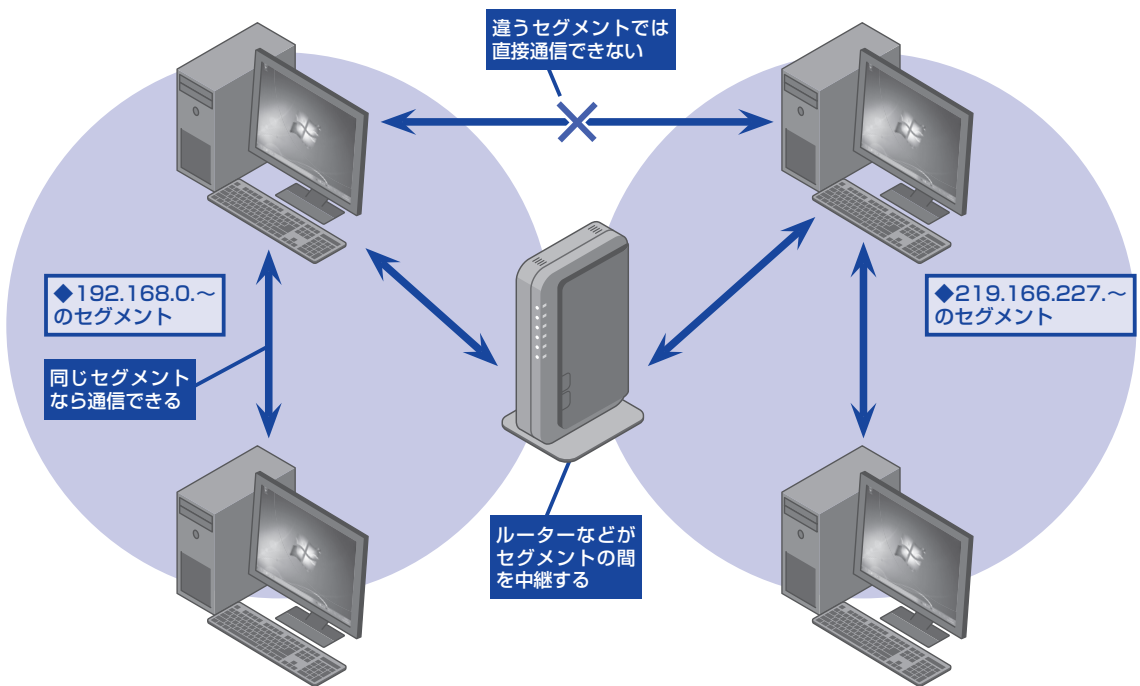
インターネットはネットワークの集合体

LANという特定の範囲のネットワークを組み合わせることで、より大きなネットワークを作ることもできます。そのようにして、数多くのネットワークを一定の決まりのもとで接続し、世界規模でひとつの巨大なネットワークを構成しているのが、インターネットと呼ばれるネットワークです。LANをインターネットに接続すると、そのLANはインターネットを形作る一部になります。

直接通信できるのはセグメントの中だけ

IPアドレスのネットワークは、複数の連続するIPアドレスをまとめてひとつのネットワークとして管理します。これをセグメントと呼びます。セグメントはTCP/IPネットワークの基本単位になります。セグメントどうしを結びつけて、より広い範囲で情報交換を行なうには、セグメント間の中継点となるポイントで情報をリレーすることで、異なるセグメントに情報を伝達させます。中継点となるポイントで通信を中継することをルーティングと呼び、これを行なう機器をルーターと呼びます。

インターネットの特徴は、世界中に数多くあるセグメント単位の小さなネットワークが集まり、ルーティングによってどこにでもアクセス可能なネットワークを構成していることです。このように、ルーティングは、インターネットにおける非常に重要な機能のひとつです。



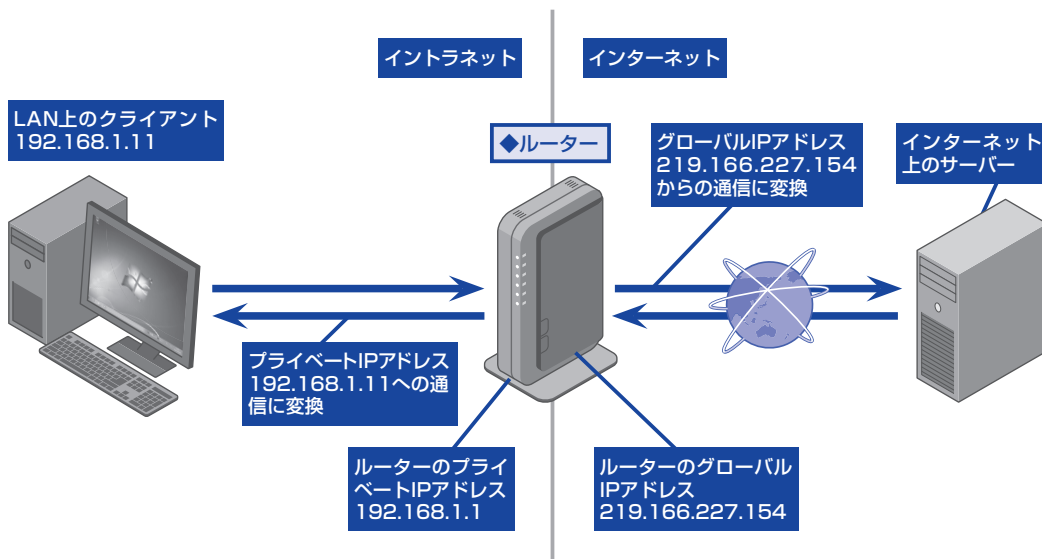
次のページに続く

NATでLANとインターネットを中継

インターネットに接続せず、プライベートアドレスを使って構築されるネットワーク（LAN）のことを、「イントラネット」と呼びます。しかし、プライベートアドレスが割り当てられたパソコンなどから、インターネットにアクセスできている場合も多いでしょう。それを実現しているひとつの方法として「NAT」（Network Address Translation）というしくみが使われています。

インターネットで通信を行なうためには必ずグローバルアドレスが必要になります。NATを使うと、プライベートアドレスからのアクセスを、代表となるグローバルアドレスからのものに変換してアクセスします。NATを使ってインターネットにアクセスする場合には必ず、NATの機能を持った機器（ルーターなど）はイントラネット用のプライベートアドレスとグローバルアドレスの両方が割り当てられています。

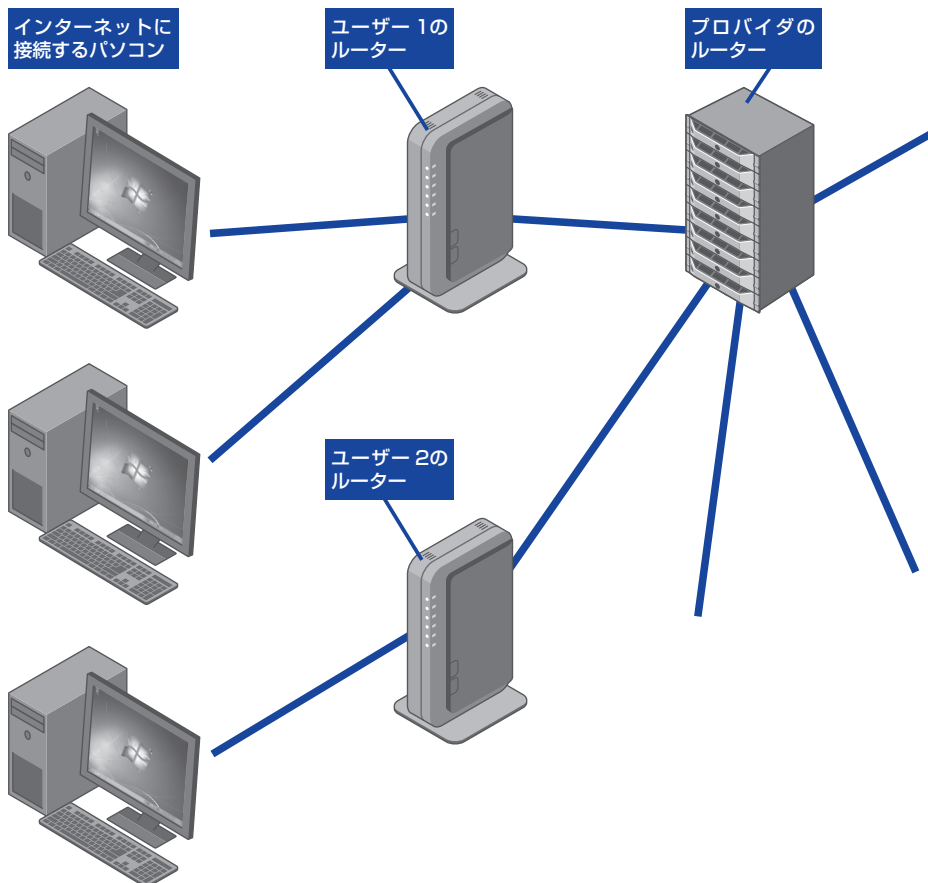
NATのもうひとつのメリットは、LANの中をインターネットから隠すことができる点です。アクセスするときにプライベートアドレスはすべて代表のアドレスに変換されますので、インターネット側からはLANの中を知ることはできません。あくまでLANからのアクセスの応答のみが、NATを経由してLAN内の元のアドレスに戻ることができます。



ルーターがつながって世界をカバー

セグメント間の中継点となるポイントで通信を中継することを、ルーティングと呼ぶと説明しました。ルーティング (routing) とは、道を作るという意味です。これは、インターネット上で自分の位置から目的地に到達するための道を作るということです。

インターネットはネットワークが蜘蛛の巣のような形でつながっています。ですから目的地に直接つながっているとは限りません。そのため、自分がつながっているネットワークに対して、目的地の書かれた情報を渡します。受け渡されたネットワークは、同様に情報の目的地を見て、自分がつながっている別のネットワークに情報を渡します。このようなパケットリレーを繰り返すことで情報が目的地に到達します。ルーティングとは、目的地を見て適切なネットワークを通じて情報を送り届けることをいいます。ルーティングのおかげで、インターネットは柔軟に世界中で通信ができるわけです。



1-6

サーバーとどう通信するの？

IPアドレス

ネットワークの通信では、どこからどこへ、という互いの場所を示すアドレスがあってはじめて通信が成り立ちます。その実際の通信で使われるアドレスがIPアドレスです。インターネットを利用する機器には、自分という存在を特定してもらうために、必ずIPアドレスが必要です。

IPアドレスには、従来より広く使われているIPv4と、IPv4アドレスの枯渇に対応したIPv6の2種類があります。現在のところ、インターネットで一般的に扱われているのはIPv4です。そのため、本書でIPアドレスといえば、特に断りのない限り、IPv4であると思って読み進めてください。

2種類のIPアドレス

よく目にするIPアドレスは、0～255の4つの数字の組を「.」で区切って表現され「0.0.0.0」に始まり「255.255.255.255」に終わる、4,294,967,296個（約43億個）のアドレスです。従来より使われているこのアドレス体系を、IPv4（IP version 4）といいます。

IPv4のアドレスは全部で約43億個とはいえ、全部使えるわけではありません。また、世界中にインターネットが普及した現在では、アドレスが全然足りていません。そのため、**レッスン1-5**で解説したNAT等の技術を駆使して対処しています。

このIPv4の枯渇問題を解決する新しいアドレス体系として、IPv6（IP version 6）があります。IPv6では、IPv4の約43億の4乗分という、想像できないほど膨大なアドレスが使えるようになります。しかし、すべての機器をIPv6に対応しなければ利用できないため、簡単には移行できず、今のところ限られた範囲での利用にとどめられています。

	長所	短所
IPv4	今現在、どのネットワークでも利用されている	アドレスが足りないため、NATなどを組み合わせて利用する必要がある
IPv6	アドレスが膨大なため、気軽にアドレスを割り当て利用できる	今のところ、限られたネットワークやサービスでしか利用できない