

入門

河野 真治 著

Perl:



アスキー出版局

本書の範囲内するご質問は、小杜スーパーアスキー編集部まで、封書（返信用切手同封のこと）にてお願い致します。

電話によるお問い合わせには、応じられません。

なお、本書の範囲を越える質問に関しては、お答えできない場合もあります。

- 落丁・乱丁本は、送料当社負担にてお取り替え致します。

お手数ですが、小杜営業部までご返送ください。

入門Peri

1994年7月11日 初版発行

著 者 こうの しんじ 河野 真治

発行人 宮崎 秀規

編集人 土屋 信明

発行所 **株式会社アスキー**

〒151-24 東京都渋谷区代々木4-33-10

振 替 東京4-161144

大代表 (03)5351-8111

出版営業部 (03)5351-8194 (ダイヤルイン)

スーパーアスキー編集部 (03)5351-8125 (ダイヤルイン)

©1994 Shinji Kouno

本書は著作権法上の保護を受けています。本書の一部あるいは全部について（ソフトウェア及びプログラムを含む）、株式会社アスキーから文書による許諾を得ずに、いかなる方法においても無断で複写、複製することは禁じられています。

制 作 株式会社GARO

印 刷 モリモト印刷株式会社

編 集 小島 範幸／佐々木 三奈

ISBN4-7561-0292-1

Printed in Japan

入門Perl

河野真治 著

アスキー出版局

商 標

UNIX オペレーティングシステムは、UNIX System Laboratories, Inc. が開発し、ライセンスしています。

そのほか、本文で使用するシステム名、製品名は一般に各開発メーカーの登録商標です。なお、本文中では、TM、コピーライト表示は明記しておりません。

表紙デザイン：郷 啓子

表紙オブジェ制作：大崎吉之

表紙オブジェ撮影：山本耕司

はじめに

この本の読み方・使い方

Perl は、テキストのかたちをしたデータを処理するためのプログラミング言語である。知っている人はよく知っているだろうし、知らない人は名前も聞いたこともないに違いない。知っている人だったら Perl についてくるマニュアルで十分だし、『ラクタ本』と呼ばれるくわしい本も出ている。でも、それらの本をもってしても、ちょっと困るときがある。それは、Perl を知らない人に、Perl を使ってもらうときだ。この本は、手短かに Perl を知って使ってもらうための本である。

この本は Perl を知っている人が、Unix やプログラムの初心者に Perl を勧めるときに役に立つように書かれている。実際、この本には Perl のすべてが書いてあるわけではない。裏技があるわけでもないし、Perl の秘密が書いてあるわけでもない。でも、Perl をはじめて勉強する人に Perl のマニュアルを渡すのは可哀想だ。だから、この本は Perl のマニュアルを必要とところだけ読むことを考えて作ったつもりである。

Perlはすぐ使えるようになる

新しい言語を「さっさと」使う方法、使えるようにする方法はなんだろう。プログラム言語には作った人が考えた使い方があって、それにそって使うのがいちばんいい。でも、使い方はマニュアルを読んでもよくわからないものだ。いちばんいいのは短いプログラムを読むことである。短くて役に立つプログラムが、プログラム言語には絶対ある。だから、この本には短い例題をたくさん入れてある。最初は、それを少しずつ手直ししてみよう。必ず少しずつ変更しよう、そうでないと、動かなくなったときになにがいけなかったかよくわからない。それに慣れたら、その言語が持っている機能を実験してみよう。実験なしにプログラム言語をマスターすることはできない。それに、実験はおもしろい。とくに Perl の実験は軽くて早くて楽しい。Perl にはよいデバッガがついているから、Perl の動きを

簡単に調べることができる。いま、はやりのビジュアルなものではないけど。どうせ、Perl のもっとも強力な武器である正規表現にはビジュアルなんて関係ない。

Perlにできないこと

もちろん、Perl には向くこと向かないことがある。Perl はどんなものか？ UNIX の BASIC 言語だというのがぴったりだ。C や lex のようなコンパイラ言語よりは遅いけど、UNIX の機能を全部含んでいる。AWK、sed、grep のような特殊化したツールとしても使えるけど、普通のプログラム言語だからもっと柔軟だし、文字列処理用によく考えられているから制限が少ない。もし、巨大なアプリケーションを書きたいなら Perl はやめよう。もし、巨大なデータ (1M のログとか) を持っていて、それをちょっと操作したいなら Perl がぴったりだ。正規表現を使えば、そのなかから必要な部分を取り出すことが簡単にできる。

Perl は UNIX または UNIX と同等の機能を持つ OS の上で使ってこそ意味がある。640KB のアドレス空間しかない MS-DOS で使ったって、本当の Perl なんてわからない。OS/2 でも Windows NT、あるいは DOS エクステンダでもいいけど、32bit 以上のポインタを使える OS じゃないとだめだ。1000 個とか 2000 個のデータ、あるいは、100KB のファイルをメモリに軽がると読み込んで処理する。それを他の Perl プログラムや、UNIX のツールと組み合わせる。10 分で組んで、10MB を 10 分で処理する。それが Perl のやり方。でも、大きなデータは、それなりに注意して扱おう。時間がかかったら、途中の状況を調べたり、ちゃんとバックアップを取る必要がある。

Perlは速くて効率がよい？

Perl は速い？ それはもちろん使い方による。たとえば、`cat*1` を perl で書いてみよう。200KB ある `/usr/dict/words` も UNIX ならキャッシュに入れば一瞬、0.2 秒で cat できる。Perl で考えずに簡単に書くと 9 秒かかる。でも工夫して書くと 0.7 秒で終わる。ちなみに時間は、Mach386 386SX 20MHz の上の Perl の場合である。書き方によってずいぶん違うのが Perl。でも、考えようによっては 9 秒だろうと 1 秒だろうと 8 秒しか差がないともいえる。だから、簡単に書きた

*1 1.1.2 cat.pl

いときは手を抜けばいいし（これがほかの言語ではなかなかできない）、よく使うプログラムなら力を入れてチューンナップすればよいわけだ。

使用したソフトウェアのバージョン

今回の執筆にあたり使用した Perl のバージョンは、4.036 である。読者が使用するものとバージョンが異なっている場合には、説明どおりに動作しないこともあるので注意願いたい。

参考図書

- 「Perl 付属のマニュアル」、近藤嘉雪氏が訳されたものがある。
- 高尾直弥氏による「日本語 Info」、Anonymous FTP、大手 BBS などにあるだろう。
- 「Perl プログラミング」、L.Wall and R.L.Schwartz 著、近藤嘉雪訳、ソフトバンク刊。ISBN4-89052-384-7。
本書で『ラクガ本』とっているのは、この本のことである。Perl の作者 Larry Wall による解説はきっと役に立つだろう。
- 「Perl 書法」、増井俊之著、アスキー出版局刊。ISBN4-7561-0281-6。
本書を読まれたあと、この本の Perl パワープログラミングを試すのももしろいだろう。

本書の表記方法について

本書では以下の表記法を使用している。

UNIX コマンド、ディレクトリ、スクリプト、コンフィギュレーションファイルなどはタイプライタ書体 (**Constant Width**) を使用している。

対話的なやりとり、ユーザーが実際に入力すべき文字などについては、太めのタイプライタ書体 (**Constant Bold**) を使用している。実際には以下のようなぐあいである。

```
% ls -F
foo  bar
```

また、キー入力については、**a**、**CTRL** のように表示している。具体的には、**CTRL** - **D** は CTRL キーを押しながら、D を押す。

謝 辞

この本を書く機会をくださった筑波大学の久野靖さん、そして、本書を手伝ってくれた千寿子さん、どうもありがとう。この場を借りてお礼申し上げる。

目 次

はじめに	3
この本の読み方・使い方	3
Perlはすぐ使えるようになる	3
Perlにできないこと	4
Perlは速くて効率がよい?	4
使用したソフトウェアのバージョン	5
参考図書	5
本書の表記方法について	5
謝 辞	6

1章 Perlの紹介————— 17

1.1 Perlを使ってみよう	17
1.1.1 ちょっと使ってみたいとき	18
1.1.2 もう少しプログラムらしいプログラム	19
1.1.3 簡単なプログラムでも複雑なことをしている	19
1.1.4 ちょっと試してみるにはデバグが便利	20
1.1.5 Perlのプログラムをコマンドにしたいとき	21
1.1.6 自分でつくったコマンドが動かなかったら 自分の環境を調べよう	23
1.1.7 マニュアルの読み方(manを忘れずに)	23
1.1.8 EmacsとPerl	25
1.1.9 EmacsのなかからPerlのデバグを立ちあげる	27
1.1.10 viとPerl	29

2章 Perlの要素 31

2.1	これだけは	31
2.2	スカラ	32
2.2.1	まずは変数を使ってみよう	32
2.2.2	比較するときは文字列と数値を区別したい	33
2.2.3	Perlで繰り返しをプログラムするには	35
2.2.4	Perlで数値計算?!	36
2.2.5	\$_を使って簡単にプログラムする	38
2.2.6	ファイルに含まれるいちばん長い行の長さ	38
2.2.7	このプログラムを動かしてみる	39
2.2.8	デバッガから動かしてみる	40
2.2.9	スカラ変数にはいろいろなものが入る	41
2.2.10	Perlには便利な文字列の演算子がたくさんある	42
2.2.11	UNIXのファイルをMS-DOSファイルに変換する	42
2.2.12	引用	44
2.2.13	デバッガで文字列の引用をいろいろ試してみる	44
2.2.14	引用を使ってPerlを電卓に使う	45
2.3	パターン(正規表現)	46
2.3.1	文字列とパターン	46
2.3.2	Perlのパターンにできることをデバッガで調べてみる	47
2.3.3	正規表現をPerlのマニュアルで調べてみる	48
2.3.4	正規表現を使ってファイルから制御記号を取り除く	49
2.3.5	UNIXではめんどくさいことに少しずつ違った 正規表現が使われている	51
2.3.6	Perlで書いたgrepとegrepどっちが速い?	52

2.4	ファイルからデータを取り出す	54
2.4.1	ファイルをオープンする	55
2.4.2	デバッガからファイルをオープンしてみる	56
2.4.3	ほかのコマンドの出力や入力をオープンすることもできる ..	57
2.4.4	PerlでPerlのマニュアルをみる	58
2.4.5	UNIXはディレクトリに気をつけよう	59
2.4.6	複数のファイルの一行目をみる	60
2.5	配 列	61
2.5.1	ユーザーが与えた引数の配列: @ARGV	61
2.5.2	配列はコンテキストに気をつけて使う	62
2.5.3	文字列から配列をつくる	64
2.5.4	Perlで表計算	65
2.5.5	クロス集計をやってみる	66
2.5.6	配列をどんどん大きくするとどうなる?	68
2.5.7	配列をスタックやキューとして使う	70
2.5.8	スタックとキューを使ってファイルを逆順に表示する	71
2.5.9	スタックを使ってTeXの括弧の対応を調べる	72
2.5.10	grepで前後数行を表示する	73
2.5.11	配列の要素全部に対して操作するには	74
2.6	配列と引用を組み合わせる	75
2.6.1	まず出力例を考える	76
2.6.2	変数部分だけを先に用意する	76
2.6.3	行指向の引用を使っていっきに表示する	77
2.7	連想配列	79
2.7.1	連想配列を集合として使う	80
2.7.2	ユーザーのログインの回数を調べてみる	80

3章 Perlプログラミング 83

3.1 条件分岐は最適化される	84
3.2 最適化のようすをデバッグオプションで調べる	86
3.3 ループにもいろいろなかたちがある	87
3.4 配列の中身をループを使って表示してみる	89
3.5 ブロックだけでループをつくる	90
3.6 サブルーチンとモジュールを勉強する	91
3.7 サブルーチンに引数を渡すには	93
3.8 Perlのライブラリでは引数をどう渡しているの？	94
3.9 ライブラリのbigint.plを使ってみる	95
3.10 Perlに向かないこと：字句解析	99
3.11 Perlに向かないこと：GUI	103

4章 UNIXプログラミング————109

4.1	UNIXとPerl	109
4.2	ファイルを探しながら(find)	111
4.2.1	Perlとfindの組み合わせ	111
4.2.2	find2perlなら、もっと複雑なこともできる	112
4.2.3	findとfind2perlとどちらが速い	113
4.3	Infoを検索する	114
4.3.1	InfoのファイルをPerlで読めないか調べてみる	116
4.3.2	Infoファイルを検索して表示する	120
4.4	UNIXのシステムインタフェース	122
4.4.1	日付と時間	122
4.4.2	サーバークライアントとデーモン	124
4.4.3	fingerdを手動で試す	125
4.4.4	chat2.plを使ってfingerdと通信する	126
4.4.5	直接fingerdと通信する	127
4.4.6	別なウィンドウからマルチプロセスでPerlをデバッグする ..	128
4.5	デバッグ	133
4.5.1	デバッグのコマンド	133
4.5.2	戦略的にデバッグする	135
4.5.3	デバッグの実際	136

5章 アプリケーション ————— 143

5.1 Perlのアプリケーション	146
5.2 Cops	152
5.3 Sybperl	153
5.3.1 なぜSybperlなのか	154
5.3.2 リレーショナルデータベースとSQL	155
5.3.3 Sybperlの動作例	157
5.3.4 Sybperlはどうやってライブラリをリンクしているの?	158
5.4 ファイル変換の実際	160
5.5 Tcl/Tkの生成	171
5.6 Perlで書いたtip	181
5.7 Installの仕方	198

図一覧

- 図 1-1 Perl オンラインマニュアル
- 図 1-2 Emacs から Perl のマニュアルを読む
- 図 1-3 info
- 図 1-4 目次
- 図 1-5 Perl のデバッガ
- 図 1-6 デバッグの実行

- 図 2-1 「`~`」の中での変数の展開
- 図 2-2 Perl でつくったヘノンアトラクタ
- 図 2-3 強調する `grep`

- 図 3-1 ループからの脱出
- 図 3-2 `Package` の二つの使い方
- 図 3-3 X Window と Perl
- 図 3-4 Tcl/Tk の widget
- 図 3-5 ボタンをつけた widget

- 図 4-1 ファイルハンドルとプロセス
- 図 4-2 `texinfo-format-buffer` をかける
- 図 4-3 あとはセーブするだけ
- 図 4-4 X Window でデバッグ
- 図 4-5 マルチウィンドウを使ったマルチプロセスのデバッグ

- 図 5-1 Perl のアプリケーションの構造
- 図 5-2 Sybase のインタフェース
- 図 5-3 リレーショナルデータベースの例
- 図 5-4 `tip.pl` の構造
- 図 5-5 `tip.pl` のデバッグ

リスト一覧

リスト 1-1	cat.pl	入力をそのままコピーする cat	19
リスト 2-1	henon.pl	ヘノンアトラクタの計算	36
リスト 2-2	ll.pl	行の最大長をみつける	38
リスト 2-3	grepf.pl	強調する grep	50
リスト 2-4	grep.pl	普通の grep	52
リスト 2-5	manperl.pl	Perl でマニュアルを検索	58
リスト 2-6	files.pl	複数のファイルの一行目を見る	60
リスト 2-7	argtest.pl	@ARGV のテスト	62
リスト 2-8	data	cross.pl のテストデータ	65
リスト 2-9	cross.pl	表計算 (クロス集計)	67
リスト 2-10	tac.pl	ファイルを逆に表示	71
リスト 2-11	cbcheck0.pl	LaTeX の bigen、end を調べる	72
リスト 2-12	grepw.pl	前後を表示する grep	73
リスト 2-13	llv.pl	grep を使った行の最大長をみつけるプログラム	75
リスト 2-14	crtex.pl	クロス集計を LaTeX で印刷	77
リスト 2-15	logincount.pl	last からの情報を使ってログイン回数を調べる	81
リスト 3-1	cases.pl	条件文の最適化を調べるプログラム	84
リスト 3-2	flush.pl	Perl のライブラリ	91
リスト 3-3	bigint.pl	Perl の任意桁整数演算ライブラリ	95
リスト 3-4	tgifl.pl	tgif 用字句解析プログラム	101
リスト 3-5	簡易版 wish.pl	Tck/Tk のコマンド生成プログラム	104
リスト 3-6	wafehello.pl	Perl から Wafe を使う	105
リスト 3-7	tclhello.pl	Perl から Tcl/Tk を使う	106
リスト 4-1	/usr/local/lib /nemacs/info/dir	info のルートファイル	116
リスト 4-2	info0.pl	info を Perl で検索	121
リスト 4-3	mailtime.pl	メールのタイムスタンプを計算する	123
リスト 4-4	.perlodb	マルチプロセスと X Window を使ってデバッグする (デバガ側)	129

リスト 4-5	<code>perldbc.com.pl</code>	マルチプロセスと X Window を使っ てデバッグする (ウィンドウ側)	130
リスト 4-6	<code>pxcal.pl</code>	Xcalendar を表示 (バグ入り)	136
リスト 4-7	<code>pxcal.pl</code> (完成版)	Xcalendar を表示	140
リスト 5-1	<code>FORMAT2.1</code>	xfig のフォーマット	161
リスト 5-2	<code>xfig2idraw.pl</code>	xfig を idraw に変換	162
リスト 5-3	<code>idraw2tcl.pl</code>	idraw を Tcl/Tk に変換	173
リスト 5-4	<code>tip.pl</code>	Perl で書いた通信プログラム	184