

# Let's Try ChangeLog Memo

---

Edition 0.1.2

by 安宅 正之 (Masayuki Ataka) [ataka@milkm.freemail.ne.jp](mailto:ataka@milkm.freemail.ne.jp)

---

## 目次

エッセイ風の序文 .....	1
<b>1 ChangeLog .....</b>	<b>2</b>
1.1 Emacs の ChangeLog モード .....	3
<b>2 ChangeLog メモ .....</b>	<b>4</b>
2.1 インライン日付 .....	4
2.2 引用文 .....	5
2.3 タグ .....	5
<b>3 clmemo — ChangeLog メモ モード .....</b>	<b>6</b>
3.1 clmemo のインストール .....	6
3.2 メモの入力 .....	6
リージョンをメモに追加する .....	7
3.2.2 タイトルの補完入力 .....	7
chalow カテゴリー風のサブ・タイトル .....	8
タイトルの自動修正 .....	8
3.2.3 曜日情報の追加 .....	8
3.3 メモの編集 .....	9
3.3.1 移動とマーク .....	9
3.3.2 行頭の TAB .....	9
3.3.3 インライン日付の編集 .....	10
インライン日付へのジャンプ .....	11
3.3.4 引用文の編集 .....	11
3.3.5 タグの編集 .....	11
3.3.5.1 Emacs-w3m とのコラボレーション .....	12
3.4 各エディタ上の clmemo .....	12
<b>4 clgrep — ChangeLog grep .....</b>	<b>13</b>
4.1 blgrep — Block grep .....	13
clgrep のインストール .....	13
4.2 clgrep の使い方 .....	14
AND 検索 .....	15
OR 検索 .....	15
4.3 clgrep の他の実装 .....	15
4.3.1 clgrep — Ruby での実装 .....	15
4.3.2 pickup — Perl での実装 .....	16
<b>5 関連ツール .....</b>	<b>17</b>
5.1 Chalow — CHangeLog On the Web .....	17
5.2 iimage.el .....	17
<b>参考文献およびウェブページ .....</b>	<b>18</b>
参考文献 .....	18
参考ウェブページ .....	18

付記 A  他の ChangeLog 形式 .....	19
古い ChangeLog 形式 .....	19
GNU Arch の形式 .....	19
変数索引 .....	20
関数・キー索引 .....	21
概念・プログラム索引 .....	22

## エッセイ風の序文

本テキストの目的は ChangeLog メモの紹介である。この題名にもなっている ChangeLog メモとは一体何か。一言でいへば、それは概念である。それも、コンピューター上のメモ、つまり電子メモのための概念である。アイデアといつても差し支えない。

大仰にいふのだから、難しいにちがひない。私なんかにはとても手が出まい、と思はれるむきもあらう。しかしさにあらず。プログラムの知識なんて要らない。レポート用紙に書くまでもない。実は、とても簡単なものである。それも、ただの一行で説明できる。かうである。

一つのファイルに ChangeLog 形式で全てのメモを取る

これだけである。なあんだそれだけか、と侮るなかれ。ここには二つの意味がある。

一つは、一つのファイルに全てのメモを取る といふ件である。これは、「超」整理法のポケット一つの原則—「必要なものは一つの収納場所にしまうべきである」といふ考え—をコンピューター上のメモに適用したものである。言ひ換えれば、「必要なメモは一つのファイルにしまうべきである」となるうか。しかしこの効能について、私が下らないことを書き連ねるよりも、「超」整理法から引用した方がよろしからう。

目的の書類は、本棚の中に必ず存在しているからだ。捨ててさえいなければ、必ずそこにある。ほかの場所を探す必要はない。つまり、「存在定理」が証明されているのである。だから、ここを探せば、必ず出てくる。搜索は、この場所に限定して行えばよい。

ここで、「目的の書類」は「目的のメモ」、「本棚」は「ファイル」と読み換えればよろしい。一般に、ないものをないと証明するのは難しい。ファイルを一つに限定することで、メモの検索効率が上がり、ないものを探しまわる手間が省けるわけである。ついでにいへば、ファイルの移動・編集も楽になる。たくさんのファイルと一つのファイルでは、どちらが扱い易いか自明であらう。

さて、ChangeLog メモ のもう一つの重要な意味を説明しよう。それは、ChangeLog 形式でメモを取る といふ件である。なぜ ChangeLog 形式なのか。それは、メモとメモの記入日が常に一緒になるからである (ChangeLog 形式の詳しい説明は本文にゆずる)。日付の入っていない日記が役にたたないのと同じやうに、何時書いたのかわからぬメモは価値が低い。もちろん、メモに記入日が一緒に入れば何でもよいのである。ChangeLog 形式は、たまたまその要求を満たしており、プログラムの世界では比較的有名で、特別なソフトに依存しておらず、わかりやすい形式であった。だから、我々は ChangeLog 形式を使っているにすぎない。

蛇足ながら付け加えると、メモをプレーン・テキストで取れることも嬉しい。というのも、特定のプログラム (例えば Word) に依存せずにすむからである。もしさういふことになれば、プログラムの寿命がメモの寿命になつてしまう。メモは一生ものであるから、できるだけプログラムに依存しないやうにしたい。いわゆる、データとアプリケーションの分離である。ChangeLog 形式は、さういふ意味でも電子メモとして優れている。

ChangeLog メモに興味を持つていただけたであらうか。よろしければ、この先をお読み頂きたい。ChangeLog メモの具体的な書き方、編集・検索用ツールの紹介、いくつかの有用なポインターがある。

最後に、ChangeLog メモの一例をもつて、序文を締めくくらせて頂く。

2004-09-06 Masayuki Ataka <ataka@milk.freemail.ne.jp>

\* clmemo: エッセイ風の序文を書いた。

# 1 ChangeLog

ChangeLog ファイルは、  
 プログラムが何時、何故変更されたのかという時系列記録を持つ。  
 記録は変更点を記述した連続物のエントリーで構成されている。  
 普通、その記録は ‘ChangeLog’ と呼ばれるファイルに保存されている...

A change log file contains a chronological record of when and why you have changed a program, consisting of a sequence of entries describing individual changes. Normally it is kept in a file called ‘ChangeLog’...

— from *GNU Emacs Reference Manual*

本章では、ChangeLog メモが利用している ChangeLog のフォーマットを説明する。そもそも ChangeLog とは、(プログラムの) コードの変更履歴を書く為のフォーマットであった。そのためいくつかのバリエーションがあるが、ここでは GNU Emacs がサポートしている ChangeLog 形式を紹介する。なお、ChangeLog の用語は section “Change Logs” in *GNU Emacs Reference Manual* に依った。

最初に ChangeLog の例を示す。

```
1993-05-25  Richard Stallman  <rms@gnu.org>
```

```
* man.el: Rename symbols ‘man-*’ to ‘Man-*’.  
(manual-entry): Make prompt string clearer.
```

```
* simple.el (blink-matching-paren-distance):  
Change default to 12,000.
```

```
1993-05-24  Richard Stallman  <rms@gnu.org>
```

```
* vc.el (minor-mode-map-alist): Don’t use it if it’s void.  
(vc-cancel-version): Doc fix.
```

ChangeLog は、複数の「エントリー」によって構成されている。「エントリー」の一行目は「エントリー・ヘッダー」で、記入日と名前、そしてメール・アドレスから成る。エントリー・ヘッダーから次のエントリー・ヘッダーを一つのエントリーと数える。上の例では、‘1993-05-25 Richard Stallman <rms@gnu.org>’ と ‘1993-05-24 Richard Stallman <rms@gnu.org>’ の二行がエントリー・ヘッダーで、二つのエントリーがあることが分かる。

「エントリー・ヘッダー」に続けて、複数の「アイテム」が続く。ここが変更点を具体的に書く場所である。アイテムのルールは次の 2 つ。

- アイテムはタブでインデントされる。
- アイテムは ‘\*’ (アスタリスク) で始まり、空行で終わる。

アイテムは通常、‘\* ファイル名: ’ もしくは ‘\* ファイル名 (関数名 or 変数名): ’ という形を取る。

上の例では、最初のエントリーに 2 つのアイテム。次のエントリーに 1 つのアイテムがある。一番最初のアイテムは ‘man.el’ というファイルに関する変更履歴である: まず、‘man-\*’ で始まるシンボルを ‘Man-\*’ という形にリネームしたとある。続けて、‘manual-entry’ という関数のプロンプト文字列を分かり易くしたとある。二番目のアイテムは ‘simple.el’ というファイルの変更履歴だ。‘blink-matching-paren-distance’ という変数のデフォルト値を 12000 にしたとある。

これにエントリー・ヘッダーの情報を合わせれば、上記変更が 1993 年 5 月 25 日に Richard Stallman 氏によって行われた事が分かる。そして 1993 年 5 月 24 日の変更履歴が続くわけだが、もはや説明は不要であろう。

## 1.1 Emacs の ChangeLog モード

Emacs は標準で ChangeLog を編集する為のモードを備えている。通常 ‘ChangeLog’ という名前のファイルを開けば、自動的に ChangeLog モードに入る。ChangeLog モードは、次の機能を提供する。

- ChangeLog に応じた色付け。
- 改行で行頭への TAB 文字自動挿入。
- 行頭の TAB を考慮した fill。

しかし、Emacs ユーザーは ‘ChangeLog’ ファイルを直接、手で開いたりしない。編集中のファイルで `C-x 4 a` (`M-x add-change-log-entry-other-window`) と打つのである。すると Emacs は、適切な ‘ChangeLog’ ファイルを開き、エントリー・ヘッダーとファイル名と関数名 or 変数名を自動的に挿入する。この機能は、C でも Perl でも Ruby でも Fortran でも Java でも、ほとんどの言語でサポートされている。

なお、エントリー・ヘッダーに使われるユーザー名とメール・アドレスは、次の 2 つの変数で変えられる。

`user-full-name` [ユーザオプション]  
エントリー・ヘッダーのユーザー名

`user-mail-address` [ユーザオプション]  
エントリー・ヘッダーのメール・アドレス

もし、編集中の File が CVS などのバージョン管理下にあるなら、‘\*VC-Log\*’ バッファで `C-c C-a` してみよう。ChangeLog からログがコピーされる。

## 2 ChangeLog メモ

年寄りパイロットや勇敢なパイロットはいるが、  
勇敢な年寄りパイロットはいない。

There are old pilots, and there are bold pilots,  
but there are no old, bold pilots.

— from *chalow* を用いた *ChangeLog* 電子メモ術の極意

ChangeLog メモとは、ChangeLog 形式を利用して電子メモを取ろうというアイデアで、山下達雄氏によって 2000 年頃に提唱され、2002 年 12 月、Namazu で有名な高林哲氏が Unix Magazine の「横着プログラミング」で取り上げた事から広く知られるようになったものである。[序文], 頁 1 にある通り、ChangeLog メモファイルは一つのファイルで、メモに限らず、日誌、予定、家計簿などを書いておく。プレーン・テキストを利用した電子メモなので、

- grep やエディターのサーチといった既存のツールで検索可能
- sed や Perl, Ruby などですぐに加工可能

である。一方、画像を始めとするバイナリー・データは扱えない。

2005-03-11 Masayuki Ataka <ataka@milk.freemail.ne.jp>

\* タイトル (サブ・タイトル): ChangeLog の「アイテム」は一日のメモに、  
ファイル名はメモの「タイトル」に、  
関数名 or 変数名はメモの「サブ・タイトル」にあたる。

昔は空行をメモの区切りとしていたが、今はそうではない。  
空行の後に TAB\* があると新しいメモが始まったとみなす。

\* chalow [カテゴリ]: chalow では、サブ・タイトル部分に [...] を使う。  
ここをカテゴリと呼ぶ。

2004-07-22 Masayuki Ataka <ataka@milk.freemail.ne.jp>

\* pop-club: blgrep-0.1rc1 と clmemo-1.0rc1 をリリース。  
chalow ML にもリリースをメールした。  
pop-club は筆者のウェブページ。作業日誌としてメモを取った。

\* Command (grep): -v 結果の反転。マッチしなかった行を選択する。  
grep コマンドの使い方をメモした。  
もう man の網羅的な説明から -v オプションを探す必要はない。

ChangeLog メモは一生使える電子メモを目的としている。

「勇敢な (=格好いい) ツールを次から次へと現れるとしても、結局、最後はシンプル  
なものが生き延びる

と信じている。

### 2.1 インライン日付

インライン日付とは、年月日をハイフンで繋ぎ角カッコ「[...]」で囲んだものである。一桁の月や日は、「0」を補って二桁にする。

[2004-07-07]

インライン日付はアイテム中のどこにでも書けるが、下記のように行をまたいではない。

[2004-07-07]

インライン日付に対して、エントリー・ヘッダーにある日付を、エントリーの日付とか単に日付と呼ぶ。後に説明する `clmemo.el` や `clgrep` や `chalow` では、インライン日付はそのエントリーの日付へのハイパーリンクになる。

インライン日付では、アイテムを直接指定することもできる。以下のように、アイテムも指定したインライン日付をアイテム情報付インライン日付と呼ぶ。

[2004-07-07-3]

上の例は、2004 年 7 月 7 日のエントリーの下から 3 番目のアイテムを指す。アイテム情報付インライン日付では、アイテムの番号をエントリーの下から数える。これは、上から数えるとアイテムを追加した時に番号がずれてしまうためである。

## 2.2 引用文

ChangeLog メモでは、引用を表すのに次の二つの表記を用いることが多い。一つは、行頭に ‘>’ もしくは ‘|’ といった記号を付ける方法である。

2005-03-12 (Sat) Masayuki Ataka <ataka@milk.freemail.ne.jp>

```
* clmemo: 短い引用のやり方
> メール引用記号のようなノリで書ける。
> ただし、長い引用には向かない。
| 引用記号は、‘>’ でも ‘|’ でもよい。
| 引用記号は TAB のすぐ後に来ることが期待される。
| 従って、これは引用ではない。
```

もう一つの方法は、引用の始まりに ‘>>’ と書いた一行を、引用の終わりに ‘<<’ と書いた一行を入れる方法である。

2005-03-12 (Sat) Masayuki Ataka <ataka@milk.freemail.ne.jp>

```
* clmemo: 長い引用のやり方
>>
長い引用は ‘>>’ と ‘<<’ の間に書く。
行頭に引用記号を入れなくて済むので楽だ。
せっかくの長い引用の例なのに、短くてスマン。
短い引用に使ってもよいということで...
<<
```

後述の `chalow` は、引用部分を `blockquote` 要素で囲う。

## 2.3 タグ

タグとは、一行指行のキーワードである。

(tagname: text)

後に説明する `clmemo`, `clgrep` では、`url` タグはそのアドレスへのハイパーリンクになる。  
`chalow` では対応していない?



## 3 clmemo — ChangeLog メモ モード

Emacs ユーザーは幸せです。  
何故なら、C-x 4 a と押すだけで、  
簡単に ChangeLog ファイルを作れちゃうからです。

— from ex-webpage

clmemo.el は、ChangeLog メモの入力を支援するコマンドやマイナー・モードを提供する。

### 3.1 clmemo のインストール

最新版の clmemo.el は以下の場所にある。

- <http://pop-club.hp.infoseek.co.jp/emacs/clmemo-1.0.tar.gz>

ダウンロード後の作業は、以下の通り。make 前に必要に応じて、‘Makefile’を編集する。

```
$ tar xzvf clmemo-1.0.tar.gz
$ cd clmemo-1.0
$ make
# make install
```

続けて、‘.emacs’に以下のコードを追加する。

```
(autoload 'clmemo "clmemo" "ChangeLog memo mode." t)
;; あなたの ChangeLog メモファイルへのパス
(setq clmemo-file-name "~/personal/memo/clmemo.txt")
;; 好きなキーへバインド
(global-set-key "\C-xM" 'clmemo)
```

ここでは、ChangeLog メモが ‘~/personal/memo/clmemo.txt’にあるとしている。また、関数 ‘clmemo’を C-x M にバインドした。以降の説明は、上の設定に従うものとする。もし、違う設定を施した場合は、適宜読み替えて頂きたい。

最後に、ChangeLog メモファイルの末尾に以下のコードを追加する。

```
^L
Local Variables:
mode: change-log
clmemo-mode: t
End:
```

### 3.2 メモの入力

C-x M あなたの「ChangeLog メモ」を開き、メモのタイトルを補完入力する。

C-u C-x M あなたの「ChangeLog メモ」を開く。or 過去のメモを追加。

C-u C-u C-x M  
あなたの「ChangeLog メモ」を検索する。

例として「メモ」というタイトルのメモを書いてみよう。M-x clmemo とすると、タイトルの入力を促すプロンプトが出るので、‘メモ’と入力する。

```
clmemo title: メモ(RET)
```

ウィンドウが二つに分かれて、片方のウィンドウに ChangeLog メモファイルが現れる。

```
2004-09-06 Masayuki Ataka <ataka@milk.freemail.ne.jp>
```

```
* メモ: -!-
```

‘-!-’ は、カーソルの位置を表す。エントリー・ヘッダーに挿入されるユーザー名・e-mail アドレスの変更方法は See section “Emacs の ChangeLog モード” in *GNU Emacs Reference Manual*.

もし ChangeLog メモファイルの中から clmemo を実行した場合は、ウィンドウを二つに分けない。また、タイトルの入力プロンプトで C-g を打つと、clmemo はメモの入力を中止する。

C-u M-x clmemo のように数引数をつけて clmemo を呼び出すと、ChangeLog メモファイルの外にいる場合、ChangeLog メモファイルを開く。ChangeLog メモファイルの内にいる場合、カーソルを含む日付にメモを追加する。過去の日付にメモを追加するには、その日付に移動してから C-u M-x clmemo とする。

clmemo を呼び出す前の状態に戻るには、C-c C-q (clmemo-exit) とする。

## リージョンをメモに追加する

‘.emacs’ に次のコードを追加すると、リージョンがある場合、C-x M はリージョンをメモに追加するようになる。

```
(setq clmemo-buffer-function-list
      '(clmemo-insert-region))
```

### 3.2.2 タイトルの補完入力

タイトルの入力には補完が使える。補完リストは変数 *clmemo-title-list* で定義されている。例えば次のように設定すると、‘idea’、‘computer’、‘購入’ に対して補完入力ができるようになる。

```
(setq clmemo-title-list '(("idea" "computer" "購入")))
```

補完リストには、別名を定義する事もできる。その場合は、‘(別名 . 本体)’ という形で書く。例えば、上の設定では ‘購入’ の補完はいかにも面倒そうだ (補完するのに日本語入力を ON にしなければならない!) そこで、別名に ‘buy’ を定義してみよう。

```
(setq clmemo-title-list '(("idea" "computer" ("buy" . "購入"))))
```

タイトルに ‘buy’ と入力すると、自動的に ‘購入’ へと展開される。

サブ・タイトルの入力も続けて行うには、タイトルの末尾に ‘(’ を置く。サブ・タイトルの入力でもタイトルと同様の補完が効く。

上の *clmemo-title-list* 設定を使って、「CD の購入」というようなメモを作ってみよう。まず、C-x M でタイトル入力画面を出し、‘buy(’ と入れる。

```
clmemo title: buy(RET)
```

‘buy’ の後ろに ‘(’ がついているので、サブ・タイトルの入力を促すプロンプトが出る。ここで、‘buy’ は自動的に ‘購入’ へと置き換えられる。

```
subtitle for ‘購入’: CD(RET)
```

‘CD’ というサブ・タイトルを入力した。これで、次のようなメモを得る。

```
2005-03-12 (Sat) Masayuki Ataka <ataka@milk.freemail.ne.jp>
```

```
* 購入 (CD): -!-
```

clmemo-title-list [ユーザオブション]

タイトル入力用の補完リスト。

デフォルト値は ‘’ (“idea" "computer")’。

clmemo-subtitle-char [ユーザオブション]

タイトルの末尾につける事で、サブ・タイトルの入力を促す文字。

デフォルトは ‘(’。

clmemo-subtitle-punctuation-char [ユーザオブション]

サブ・タイトルの前後に、自動的に挿入される文字。

デフォルトは ‘( “( . ")’。

## chalow カテゴリー風のサブ・タイトル

変数 *clmemo-subtitle-char* と *clmemo-subtitle-punctuation-char* を次のように変更することで、chalow 風のカテゴリーが入力される。

```
(setq clmemo-subtitle-char "["
      clmemo-subtitle-punctuation-char '("(" "[" . "]))
```

なお、chalow では次のように複数のカテゴリーを入力する場合がある。

\* うまい! [酒] [コンビニ]: 昨日、コンビニで見つけた...

このようなカテゴリーを入力するには、カテゴリー入力の末尾に ‘[’ を加える。上の例を入力するには... C-x M でタイトルを入力 (末尾に ‘[’ を忘れないように)、

```
clmemo title: うまい![(RET)
```

最初のカテゴリーが聞かれるから、それに答えて (末尾に ‘[’ を忘れないように)、

```
subtitle for ‘うまい!’: 酒[(RET)
```

もひとつカテゴリーが聞かれるから、答えてやる。

```
subtitle for ‘うまい! (酒)’: コンビニ(RET)
```

すると、複数カテゴリー付のメモができる。

```
2005-03-13 (Sun) Masayuki Ataka <ataka@milk.freemail.ne.jp>
```

\* うまい! [酒] [コンビニ]: -!-

## タイトルの自動修正

変数 *clmemo-title-format-function* を定義する事で、例えば、タイトルを常に大文字にするとか、小文字にするとかといった自動修正ができる。

```
‘(setq clmemo-title-format-function ‘upcase-word)’
```

入力したタイトルを常に大文字にする

```
‘(setq clmemo-title-format-function ‘capitalize-word)’
```

入力したタイトルの一文字目を常に大文字にする

```
‘(setq clmemo-title-format-function ‘downcase-word)’
```

入力したタイトルを常に小文字にする

### 3.2.3 曜日情報の追加

clmemo のデフォルトのエントリー・ヘッダーは次のような形をしており、メモを取った日の曜日が分からない。

```
2004-09-06 Masayuki Ataka <ataka@milk.freemail.ne.jp>
```

変数 *clmemo-time-string-with-weekday* を ‘t’ にすると、新規エントリー・ヘッダーに曜日情報を加えるようになる。

```
2004-09-06 (Mon) Masayuki Ataka <ataka@milk.freemail.ne.jp>
```

clmemo-time-string-with-weekday [ユーザオプション]

‘non-nil’ なら、新規エントリー・ヘッダーに曜日情報を付加する。

デフォルトは ‘nil’。

また、既存のエントリー・ヘッダーを編集するために、次の 2 つのコマンドが用意されている。

clmemo-format-header-with-weekday [関数]

リージョン内のエントリー・ヘッダーに曜日情報を付加する。

clmemo-format-header-without-weekday [関数]

リージョン内のエントリー・ヘッダーから曜日情報を削除する。

### 3.3 メモの編集

本章では、ChangeLog メモの編集にかかわるコマンドを紹介する。

#### 3.3.1 移動とマーク

メモ単位の移動コマンドが、いくつか用意されている。下記のコマンドは、全て数引数を取ることで複数年・月を移動することができる。

`C-x [` [関数]  
`clmemo-next-year` [関数]  
 次の年へ

`C-x ]` [関数]  
`clmemo-previous-year` [関数]  
 前の年へ

`C-c }` [関数]  
`clmemo-previous-month` [関数]  
 前の月へ

`C-c {` [関数]  
`clmemo-next-month` [関数]  
 次の月へ

マーク用の関数もある。

`C-x C-p` [関数]  
`clmemo-mark-year` [関数]  
 カーソルのある年をマークする。

`C-c C-h` [関数]  
`clmemo-mark-month` [関数]  
 カーソルのある月をマークする。

#### 3.3.2 行頭の TAB

ChangeLog メモの欠点はメモの先頭に常に TAB が入ることだ、とよく言われている。本節では行頭の TAB と仲良くつきあうためのコマンドを紹介する。

なお `clmemo.el` では、TAB (`C-i`) に `clmemo-next-inline-date` が割り当てられている。そのため、TAB で TAB 文字が入力できない。TAB 文字を入力するには、`C-q TAB` もしくは `C-q C-i` とする。ただし、行頭に TAB を挿入するだけなら、`clmemo-indent-region` を使う方が楽である。

`C-c TAB` [関数]  
`C-c C-i` [関数]  
`clmemo-indent-region` [関数]  
 リージョンを TAB でインデントする。

`C-c C-w` [関数]  
`clmemo-kill-ring-save` [関数]  
 行頭の TAB を除いて、`kill-ring-save` (`M-w`) する。

`C-c C-y` [関数]  
`clmemo-yank` [関数]  
 行頭に TAB を加えて、`yank` (`C-y`) する。ただし、`yank-pop` (`M-y`) には対応していない。

### 3.3.3 インライン日付の編集

インライン日付入力モードは、Section 2.1 [インライン日付]、頁 4 の入力を少し (?) 楽にする挿入コマンドである。インライン日付を挿入したいと思ったら、*C-c C-d* (clmemo-insert-inline-date) としてインライン日付入力モードに入る。この時点で、まだインライン日付は挿入されない。入力したい日付のエントリーに移動して *RET* を押す。すると、インライン日付入力モードに入った場所へカーソルが戻り、そのエントリーの日付がインライン日付として入力される。インライン日付入力モードから抜けるには *q* である。

具体的な例を見てみよう。例えば、次のようなメモを書いていたとする。

```
2003-09-15 (Mon) Masayuki Ataka <ataka@milk.freemail.ne.jp>
```

```
* 映画鑑賞: 「X-Men 2」 on DVD
X-Men 2 (2003)
-!-
```

この映画の前作は、以前見たことがあるのでリンクをはっておきたい。上の ‘-!-’ の位置で、インライン日付入力モードに入る (*C-c C-d*)。前作の題名は「X-Men」であるから、*C-s X-Men* として X-Men について書いてあるメモを探す。次のようなメモが見つかった。

```
2003-05-11 (Sun) Masayuki Ataka <ataka@milk.freemail.ne.jp>
```

```
* 映画鑑賞: 「X メン」 on TV
X-Men-!- (2000)
- 借りてきた (IT 以外の) DVD よりも、このテレビでやっていた X メンの方が面白い。
  今、X-Men II をやっているから、ブームに乗ろうというテレビ企画か?
- マグニート (敵) 役でイアン・マッケランが出演。
  次作では、X-Men 達と手を組んで活躍するというので楽しみだ。
```

まさに、これこそ求めていたメモだ (X-Men は TV で見たのだなあ)。*RET* をおすと、もといいた場所に ‘[2003-05-11]’ と日付が挿入される。

```
2003-09-15 (Mon) Masayuki Ataka <ataka@milk.freemail.ne.jp>
```

```
* 映画鑑賞: 「X-Men 2」 on DVD
X-Men 2 (2003)
[2003-05-11]-!-
```

最終的には、こんなメモができた。

```
2003-09-15 (Mon) Masayuki Ataka <ataka@milk.freemail.ne.jp>
```

```
* 映画鑑賞: 「X-Men 2」 on DVD
X-Men 2 (2003)
[2003-05-11] に観た X-Men の続編。
- ヒュー・ジャックマン、ハル・ベリー、イアン・マッケラン、パトリック・スチュワート
  と豪華なメンバーが再び大活躍。
- SFX と相まって、とても面白い出来に仕上がっていた。
```

上の例ではインライン日付を挿入したが、アイテム情報付インライン日付を入力したければ、*RET* のかわりに *C-u RET* とする。

インライン日付入力モードは、(View モードのように) 文字の編集を受けつけない代わりに、*space* や *backspace* に移動用のコマンドを割り当てている。

*RET*           カーソルのあるエントリーの日付をもといた位置に挿入する。

*C-u RET*   カーソルのあるエントリーの日付とアイテム情報をもといた位置に挿入する。

*q*           インライン日付入力モードを抜ける。

*s (r)*       前方 (後方) インクリメンタル・サーチ

*n (p)*       次の (前の) アイテムへ移動する。

*f (b)*       次の (前の) エントリーへ移動する。

*{ (})*       次の (前の) 月へ移動する。

*space (backspace)*  
          前の (後ろの) ページへ移動する。

## インライン日付へのジャンプ

*TAB* [関数]  
*clmemo-next-inline-date* [関数]  
  次のインライン日付へジャンプ

*Shift-TAB* [関数]  
*clmemo-previous-inline-date* [関数]  
  前のインライン日付へジャンプ

### 3.3.4 引用文の編集

*C-c >* [関数]  
*clmemo-quote-region* [関数]  
  リージョンの行頭に引用記号を追加する。  
  数引数付で呼び出すと (*C-u C-c >*)、使用する引用記号を尋ねてくれる。

*clmemo-quote-prefix* [ユーザオプション]  
  関数 *clmemo-quote-region* で挿入される引用記号。  
  デフォルトは `">"`。

### 3.3.5 タグの編集

本節では、Section 2.3 [タグ], 頁 5 の設定方法と使い方を説明する。

タグは変数 *clmemo-tag-list* で設定する。デフォルトでは、2 つのタグ (`'url'` と `'file'`) が用意されている。電話番号用のタグ `'tel'` とメール・アドレス用のタグ `'e-mail'` を作ってみよう。

```
(setq clmemo-tag-list
      (cons '("tel" "e-mail") clmemo-tag-list))
```

タグを入力するには *C-c* ( *clmemo-tag-insert-quick* ) とする。

tag:

とタグを聞いてくる。ここで、タグの頭文字を入れるとタグが入力される。例えば、`'url'` タグを入力したい場合は *u* と入れればよい。頭文字が同じタグを設定していた場合は、*list* の中で一番最初に定義されているものが挿入される。二番目以降のタグを入力するには、*space* を打つ。すると、補完が効くようになる。

タグの上で *C-c RET* すると、タグに関連づけられた関数が呼ばれる。これを、タグ・ジャンプと呼ぶ。デフォルトでは、`'url'` タグには *browse-url-at-point* が、`'file'` タグには *find-file-at-point* が呼ばれる。

タグ入力時、タグ・ジャンプ時に関連を関連付けるには、変数 *clmemo-tag-list* を次の書式で設定する。

(タグ名 ジャンプ用関数 入力用関数)

*clmemo-tag-list* のデフォルト値は、次の通り。

```
(defvar clmemo-tag-list
  '(("url" browse-url-at-point)
    ("file" find-file-at-point clmemo-read-file-name))
```

### 3.3.5.1 Emacs-w3m とのコラボレーション

*Emacs-w3m* (<http://emacs-w3m.namazu.org/index-ja.html>) とは、Emacs 上で動作するウェブ・ブラウザである。Emacs-w3m には独自のブックマーク機能があるが、本節では、ブックマークを ChangeLog メモで取る方法を説明する。なお、Emacs-w3m のインストール方法および使用法は、Emacs-w3m のページを参照されたい。

まず、次のコードを `‘.emacs’` に入れる。

```
(setq clmemo-buffer-function-list
  '(clmemo-tag-insert-url-from-w3m))
```

次に、Emacs-w3m でウェブ・ページをブラウズする。ここでは、clmemo.el を配布しているウェブページを覗いてみることにしよう。ウェブページのアドレスは <http://pop-club.hp.infoseek.co.jp/emacs/clmemo.html> である。さて、このページはよいページなので (もちろんだとも!) ChangeLog メモにブックマークとして保存したくなった。C-x M でメモを開こう。まず、title が聞かれる。‘URI’ という名前にしよう。

```
clmemo title: URI RET
```

普段ならここでメモの編集ができるようになるが、Emacs-w3m の中から clmemo を呼び出した時は少し違う。ウェブページの「タイトル」を入れるかどうか聞いてくる。

```
Insert ‘Let’s Try ChangeLog Memo’ as title? (y or n)
```

ここは ‘y’ と答えよう。次にウェブページの URL を入れるかどうか聞いてくる。

```
Insert URL? (y or n)
```

更に ‘y’ と答える。すると、次のようなメモができる

```
2002-01-15 (Tue) Masayuki Ataka <ataka@milk.freemail.ne.jp>
```

```
* URI: Let’s Try ChangeLog Memq!-
(url: http://pop-club.hp.infoseek.co.jp/emacs/clmemo.html)
```

url タグの中で C-c RET とすると browse-url-at-point が呼ばれ、(おそらく Emacs-w3m が立ち上がって) ウェブページが見れる。

## 3.4 各エディタ上の clmemo

Vim            ChangeLogMemo Tips (<http://apollo.u-gakugei.ac.jp/~yoshiki/clmemo/cltips.php>)

秀丸

TeraPad

WZ エディタ

Windows ユーザーのための電子メモ術 (<http://blogger.main.jp/changelog/yayadoc/memolit..>)

xyzzzy        changelogmemo (<http://www5e.biglobe.ne.jp/~fukafuka/xyzzzy/script/scripts.html>)

## 4 clgrep — ChangeLog grep

clgrep は ChangeLog メモ専用の grep である。

まず、ChangeLog メモは「アイテム」単位のメモである。そして、(我々がよく利用する) grep は「行」単位の検索ツールである。従って、grep は検索が成功しても「メモの一部」しか表示しない。一部で十分な場合もあるが、「メモの全体」を俯瞰できると便利である。そこで、検索結果をアイテム単位で出力するように拡張された grep ツールが求められた。それが clgrep なのである。

次の出力は、‘M-x occur’<sup>1</sup> を使い、私の ChangeLog メモで「弟子」を検索した結果である。

```
2 matches for "弟子" in buffer: clmemo.txt
```

```
4653:      * Quotation: 弟子が尋ねた。
```

```
17978:      が鼻祖となり、その弟子 S. I. ハヤカワらが世に広めた新しい言語学体系
```

メモの一部分を抜き出しているため、何を書いてあるのかよく分からない。一方、clgrep を使って検索すると、次のように検索にヒットしたメモ全体が出力される。

```
2004-06-24 (Thu) Masayuki Ataka <ataka@milkm.freemail.ne.jp>
```

```
* Quotation: 弟子が尋ねた。
```

```
「先生、私は先生がカッコをまるで魔術師のように  
扱っているのを常々敬服しています。
```

```
どうすれば先生のようになれるのでしょうか？」
```

```
師「えっ？ カッコ？ あ、そうか。そんなものもあったな。いやあ、  
すっかり忘れておったわ」
```

```
(author: 竹内 郁雄)
```

```
(cite: 竹内郁雄: やっぱり Lisp だね, bit, Vol. 32, No. 5., 2000.) pp.33-39
```

```
2003-01-13 (Mon) Masayuki Ataka <ataka@milkm.freemail.ne.jp>
```

```
* 本: 「非 A の傀儡」読了。
```

```
The Pawns of NULL-A (1956) p.352
```

```
<author; A. E. van Vogt> A. E. ヴァン・ヴォークト
```

```
- 解説に「非 A」の簡単な紹介がある。
```

```
>>
```

```
「非 A」即ち「非アリストテレス」主義はアルフレッド・コージブスキー (1879-1950)  
が鼻祖となり、その弟子 S. I. ハヤカワらが世に広めた新しい言語学体系  
「一般意味論」の別称である。
```

```
<<
```

### 4.1 blgrep — Block grep

clgrep.el は Emacs 上で動作する clgrep で、blgrep パッケージに含まれている。blgrep パッケージは EmacsLisp で書かれた grep コマンド集で、以下のサイトからダウンロードできる。

- blgrep のページ (<http://pop-club.hp.infoseek.co.jp/emacs/blgrep>)
- <http://pop-club.hp.infoseek.co.jp/emacs/blgrep/blgrep-0.2.tar.gz>

#### clgrep のインストール

ダウンロード後の作業は、以下の通り。clgrep のコンパイルには clmemo.el が必要なので、clmemo.el を clgrep.el のあるディレクトリーにコピーする。

<sup>1</sup> 乱暴に言えば grep を EmacsLisp で実装したコマンド



```
$ tar xzvf blgrep-0.2.tar.gz
$ cd blgrep-0.2
$ cp path/to/clmemo.el ./
$ make
# make install
```

インストールに成功したら、`.emacs` に以下のコードを追加する。

```
;
; clgrep
;
(autoload 'clgrep "clgrep" "ChangeLog grep." t)
(autoload 'clgrep-item "clgrep" "ChangeLog grep." t)
(autoload 'clgrep-item-header "clgrep" "ChangeLog grep for item header" t)
(autoload 'clgrep-item-tag "clgrep" "ChangeLog grep for tag" t)
(autoload 'clgrep-item-notag "clgrep" "ChangeLog grep for item except for tag" t)
(autoload 'clgrep-item-nourl "clgrep" "ChangeLog grep item except for url" t)
(autoload 'clgrep-entry "clgrep" "ChangeLog grep for entry" t)
(autoload 'clgrep-entry-header "clgrep" "ChangeLog grep for entry header" t)
(autoload 'clgrep-entry-no-entry-header "clgrep" "ChangeLog grep for entry except entry h" t)
(autoload 'clgrep-entry-tag "clgrep" "ChangeLog grep for tag" t)
(autoload 'clgrep-entry-notag "clgrep" "ChangeLog grep for tag" t)
(autoload 'clgrep-entry-nourl "clgrep" "ChangeLog grep entry except for url" t)
(add-hook 'clmemo-mode-hook
  '(lambda () (define-key clmemo-mode-map "\C-c\C-g" 'clgrep)))
```

## 4.2 clgrep の使い方

ChangeLog メモファイルの中に入り、`M-x clgrep` を実行する。検索文字列を聞いてくるので答える。検索には正規表現が使える。ここではシンプルに「clmemo」を検索してみよう。

Query: clmemo

検索結果はアイテムごとに日付の新しい順に並ぶ。

2004-09-17 (Fri) Masayuki Ataka <ataka@milk.freemail.ne.jp>

\* Idea (idate): 日付入力用ツール。  
clmemo.el で clmemo-inline-date-convert としているコードを改良。

2004-09-06 (Mon) Masayuki Ataka <ataka@milk.freemail.ne.jp>

\* pop-club (clmemo): clmemo-1.0rc3 リリース。

以下続く...

数引数を取ると (`C-u M-x clgrep`)、検索結果を逆順に表示する。

clgrep-item-header [関数]  
検索対象をアイテム・ヘッダー (アイテムの一行目) だけに絞って clgrep を実行する。

clgrep-item-tag [関数]  
検索対象をタグだけに絞って clgrep を実行する。

clgrep-item-notag [関数]  
検索対象からタグを除いて clgrep を実行する。

clgrep-item-nourl [関数]  
検索対象から URL と思われる文字列を除いて clgrep を実行する。

clgrep-entry-\*

[関数]

出力単位を (アイテムから) エントリーにした clgrep。

## AND 検索

clgrep は Google などの検索エンジンでよくみかける AND 検索をサポートしていない。「A」と「B」と「C」の AND 検索をする場合は、まず「A」で clgrep して、その結果に対して更に「B」で clgrep して、そして「C」について clgrep する。

例えば、「指輪物語」と「本」の両方にマッチするメモだけ表示させるには、最初に「指輪物語」で検索をかける。するとカーソルは \*clgrep\* バッファに移動して検索結果が表示される。そこで更に `M-x clgrep` とするか `g` を押すと、\*clgrep\* バッファに対して検索がかけられる。検索文字に「本」を指定して、「指輪物語」と「本」の両方にマッチするメモだけを得る。

## OR 検索

OR 検索は、正規表現を使う。

例えば、「foo」か「bar」にマッチするメモを検索するには

```
Query: foo\|bar
```

とする。ここで「\|」が正規表現にあたり、「\|」より前のものと後のものにマッチする。つまり「\|」の前の「foo」と後の「bar」にマッチするのである。

正規表現の詳しい説明は See section “Syntax of Regular Expressions” in *GNU Emacs Reference Manual*.

## 4.3 clgrep の他の実装

本節では、blgrep 以外の clgrep を紹介する。

なお、コマンドラインから呼び出すタイプの clgrep は (clgrep に限らず、普通の grep でもだが) ファイルの文字コードとプロセスの文字コードが合っている必要がある。この問題は、特に日本語を検索するときに重要になる。

例えば、もし ChangeLog メモファイルを Shift\_JIS で書いていたとして、clgrep をコマンドラインから実行したらどうなるか。あなたの端末が Shift\_JIS ならよいが、それ以外 (ex. EUC-JP, ISO-2022-JP, UTF-8) だったなら文字コードが違うので、検索に失敗してしまう。そういう場合は、ファイルを端末の文字コードに変換しておくといよい。

### 4.3.1 clgrep — Ruby での実装

clgrep の Ruby による実装。

コマンド名

clgrep

作者 高林哲さん

ウェブページ

Unix Magazine 連載: 横着プログラミング (<http://namazu.org/~satoru/unimag/>)

clgrep の最初の実装。

以下の機能をサポートしている。

- 検索結果の逆順表示
- 大文字・小文字の無視

### 4.3.2 pickup — Perl での実装

clgrep の Perl による実装。

コマンド名

pickup

作者 藤部修平さん

ウェブページ

メモの活用 (<http://shika.aist-nara.ac.jp/member/shuheif/research/myres/pickup.html>)

以下の機能をサポートしている。

- 検索結果の逆順表示
- 検索文字列の強調表示
- アイテムごとの出力一時停止

## 5 関連ツール

本章では、ChangeLog メモ関連ツールの紹介をする。

### 5.1 Chalow — CHAngeLog On the Web

*chalow* は ChangeLog メモ を HTML に変換する Perl スクリプトである。

作者 山下達雄

ウェブページ

<http://nais.to/~yto/tools/chalow/>

手元に `clmemo.txt` という名前の ChangeLog メモ ファイルがあるとしてこれを HTML に変換するには

```
$ chalow -o sample clmemo.txt
```

とする。ここで `-o sample` は、`sample` というディレクトリーの中にできた HTML ファイルを入れる、という意味である。`sample/index.html` ファイルをウェブ・ブラウザで開いてみよう。あなたの ChangeLog メモファイルが HTML になっているのが見れるはずだ。

詳しくは、chalow 付属の `README` をお読み頂きたい。

### 5.2 iimage.el

*iimage* は Emacs 上で画像を表示するマイナー・モードである。下にあるような「画像へのパス」を、実際の画像へと置き換えてくれる。

```
file://foo.png
'file://foo.png'
[[foo.gif]]
<foo.png>
'foo.JPG'
```

作者 小関吉則

ウェブページ

<http://www.netlaputa.ne.jp/~kose/Emacs/iimage.html> (英 語)  
<http://www.netlaputa.ne.jp/~kose/Emacs/200402.html> (日本語)

作者のページから `iimage.el` をダウンロードして *load-path* に置き、`.emacs` に以下の内容を追加する。

```
(autoload 'iimage-mode "iimage" "Support Inline image minor mode." t)
(autoload 'turn-on-iimage-mode "iimage" "Turn on Inline image minor mode." t)
```

ChangeLog メモに入って、*M-x iimage-mode* とすると画像が表示される。もう一度 *M-x iimage-mode* すると画像は非表示になる。

デフォルトの設定では、画像をカレント・ディレクトリーだけからしか探さない。画像を他のディレクトリーに入れて管理している場合は、`image-dir/foo.jpg` というように、ディレクトリーも指定しなければならない。しかし、この方法はいかにも患わしいし、ディレクトリーの名前を変えた時の処理が大変である。変数 *iimage-mode-image-search-path* にディレクトリー名を指定する事で、ディレクトリー名を省略できるようになる。

```
(setq iimage-mode-image-search-path '("image-dir"))
```

*clgrep* の検索時に *iimage* を自動 ON するには、次の設定を `.emacs` に加える。

```
(add-hook 'clgrep-mode-hook 'turn-on-iimage-mode)
```

## 参考文献およびウェブページ

### 参考文献

section “Change Logs” in *GNU Emacs Reference Manual*

GNU 形式の ChangeLog の説明。ChangeLog メモの用語も、このマニュアルに則っている。

「超」整理法 [中公新書 1159]

副題「情報検索と発想の新システム」。「ポケット一つの原則」や「押し出しファイリング」を世に広めたベストセラー。タイム・マネージメントを扱った続刊 続「超」整理法・時間編、捨てる技術を扱った続々刊「超」整理法 3 あり。

横着プログラミング 第 1 回: *UNIX のメモ技術* [Unix Magazine 2002/01]

連載 横着プログラミング では、Namazu や migemo で有名な高林哲氏が、横着した Unix ライフを送るためのツールを紹介した。ChangeLog メモは、その第 1 回を飾った。

横着プログラミング 第 12 回: [Unix Magazine 2002/12]

連載 横着プログラミング は、Unix Magazine に一年間続いた。最終号は、締め括りとして連載分を振り返る形を取った。第 1 回で取り上げた ChangeLog メモについては、Emacs 版 clgrep が紹介された。

書き散らかし指向メモツール *howm* [UNIX USER 2004/02] pp.50–63

副題「一人お手軽 Wiki もどき」。ChangeLog メモとは一味違うメモツール *howm* の解説。2 部構成で、第 1 部は *howm* の創作者、平岡和幸氏による *howm* の概略と Emacs との連携について。第 2 部は、七尾功一氏による vim 用 *howm-mode.vim* の解説。

*chalow* を用いた *ChangeLog* 電子メモ術の極意 [UNIX USER 2004/08] pp.54–61

ChangeLog メモの創作者、山下達雄氏による *chalow* の解説。

### 参考ウェブページ

たつをのホームページ (<http://www.nais.to/~yto/>)

ChangeLog メモの創作者、山下達雄氏のウェブページ。ChangeLog メモから HTML を作成するツール *chalow* も公開されている。

Unix Magazine 連載: 横着プログラミング (<http://namazu.org/~satoru/unimag/>)

高林氏が Unix Magazine 誌に連載した横着プログラミングの記事が公開されている。連載の栄えある第 1 回こそ、ChangeLog メモ の特集であった。

## 付記 A 他の ChangeLog 形式

### 古い ChangeLog 形式

バージョン 20.1 前の Emacs では、日付情報に違うフォーマットを使っていた。

(20.1 前): Fri May 25 11:23:23 1993 Richard Stallman <rms@gnu.org>

(20.1 後): 1993-05-25 Richard Stallman <rms@gnu.org>

旧形式のタイム・フォーマットから現在のフォーマットに変換するには、旧形式のタイム・フォーマットを使っている ChangeLog ファイルに入り、‘M-x change-log-redate’ とすればよい。これは Emacs の標準関数である。

旧形式のタイム・フォーマットは、Ruby の ChangeLog など、まだ見る事ができる。

### GNU Arch の形式

バージョン管理システムの一つである GNU Arch (<http://www.gnu.org/software/gnu-arch/index.jp.html>) では、一風変わった ChangeLog 形式を用意している。以下に示すのは、GNU Arch 1.2 に附属のマニュアルに載っている ChangeLog の例から一つのエントリーを抜粋したものである。

```
2003-01-30 GMT Tom (testing) Lord <lord@emf.net> patch-2

Summary:
  merge from mainline sources
Revision:
  hello-world--candice--0.1--patch-2

Patches applied:

  * lord@emf.net--2003-example/hello-world--mainline--0.1--patch-3
    added copywrong statements

  * lord@emf.net--2003-example/hello-world--mainline--0.1--patch-2
    commented return from main

new files:
  @{arch@}/ [...] /hello-world--mainline--0.1 [...] /patch-2
  @{arch@}/ [...] /hello-world--mainline--0.1 [...] /patch-3

modified files:
  hw.c main.c

new patches:
  lord@emf.net--2003-example/hello-world--mainline--0.1--patch-2
  lord@emf.net--2003-example/hello-world--mainline--0.1--patch-3
```

ChangeLog から、「アイテム」がなくなっている。GNU Arch では GNU 形式の ChangeLog も併用できるので、私は (Arch 上では) 両方の ChangeLog 形式を使っている。

## 変数索引

### C

clmemo-buffer-function-list ..... 7, 12  
clmemo-file-name ..... 6  
clmemo-quote-prefix ..... 11  
clmemo-subtitle-char ..... 7, 8  
clmemo-subtitle-punctuation-char ..... 7, 8  
clmemo-tag-list ..... 11  
clmemo-time-string-with-weekday ..... 8  
clmemo-title-format-function ..... 8

clmemo-title-list ..... 7

### I

iimage-mode-image-search-path ..... 17

### U

user-full-name ..... 3  
user-mail-address ..... 3

## 関数・キー索引

## A

add-change-log-entry-other-window ..... 3

## C

C-c ( ..... 11

C-c &gt; ..... 11

C-c { ..... 9

C-c } ..... 9

C-c C-d ..... 10

C-c C-h ..... 9

C-c C-i ..... 9

C-c C-w ..... 9

C-c C-y ..... 9

C-c TAB ..... 9

C-q C-i ..... 9

C-q TAB ..... 9

C-x [ ..... 9

C-x ] ..... 9

C-x C-p ..... 9

capitalize-word ..... 8

change-log-redate ..... 19

clgrep-entry-\* ..... 15

clgrep-item-header ..... 14

clgrep-item-notag ..... 14

clgrep-item-nourl ..... 14

clgrep-item-tag ..... 14

clmemo-exit ..... 7

clmemo-format-header-with-weekday ..... 8

clmemo-format-header-without-weekday ..... 8

clmemo-indent-region ..... 9

clmemo-insert-inline-date ..... 10

clmemo-insert-region ..... 7

clmemo-kill-ring-save ..... 9

clmemo-mark-month ..... 9

clmemo-mark-year ..... 9

clmemo-next-inline-date ..... 9, 11

clmemo-next-month ..... 9

clmemo-next-year ..... 9

clmemo-previous-inline-date ..... 11

clmemo-previous-month ..... 9

clmemo-previous-year ..... 9

clmemo-quote-region ..... 11

clmemo-tag-insert-quick ..... 11

clmemo-tag-insert-url-from-w3m ..... 12

clmemo-yank ..... 9

## D

downcase-word ..... 8

## I

iimage-mode ..... 17

## K

kill-ring-save ..... 9

## O

occur ..... 13

## S

Shift-TAB ..... 11

## T

TAB ..... 11

turn-on-iimage-mode ..... 17

## U

upcase-word ..... 8

## Y

yank ..... 9

yank-pop ..... 9



## 概念・プログラム索引

### \*

\*VC-Log\* ..... 3

### A

alias ..... 7  
AND search ..... 15  
Arch, GNU ..... 19  
asterisk ..... 2

### B

blgrep ..... 13  
Block grep ..... 13  
blockquote ..... 5  
bookmark ..... 12  
browser, web ..... 12

### C

category, chalow ..... 4, 8  
chalow ..... 4, 5, 8, 17, 18  
ChangeLog on the Web ..... 17  
clgrep ..... 5, 13, 15  
clgrep, Perl ..... 16  
clgrep, Ruby ..... 15  
clmemo ..... 5, 6  
completion, title ..... 7  
CVS ..... 3

### D

day of week, entry header ..... 8

### E

e-mail address ..... 2  
Emacs ..... 2  
Emacs, GNU ..... 3  
Emacs-w3m ..... 12  
entry ..... 2  
entry header ..... 2  
entry header, day of week ..... 8  
EUC-JP ..... 15

### G

GNU Arch ..... 19  
GNU Emacs ..... 2, 3  
Google ..... 15  
grep ..... 13

### H

header, entry ..... 2  
Hidemaru ..... 12  
howm ..... 18  
HTML ..... 18

### I

iimage ..... 17  
image, inline ..... 17  
inline date ..... 4, 10  
inline date, item number ..... 5  
Inline image ..... 17  
ISO-2022-JP ..... 15  
item ..... 2  
item number, inline date ..... 5

### K

keyword, one line ..... 5

### L

Log ..... 3

### N

Namazu ..... 4

### O

old-style ..... 19  
one line keyword ..... 5  
OR search ..... 15

### P

Perl ..... 16, 17  
pickup ..... 16  
plain text ..... 1, 4

### Q

quotation ..... 5, 11

### R

region ..... 7  
Regular Expressions ..... 15  
Revision Control System ..... 3  
Ruby ..... 15, 19

### S

search, AND ..... 15  
search, OR ..... 15  
Shift\_JIS ..... 15  
subtitle ..... 4, 7, 8

### T

TAB ..... 2  
tag ..... 5, 11  
TeraPad ..... 12  
title ..... 4, 6  
title, alias ..... 7

title, completion ..... 7

## U

user name ..... 2

UTF-8 ..... 15

## V

Version Control System ..... 3

view-mode ..... 10  
vim ..... 12, 18

## W

web browser ..... 12

wiki ..... 18

WZ-editor ..... 12

## X

xyzyy ..... 12