

Mew 1.95 に向けて

山本和彦

IIJ 技術研究所

Emacs 上のメールリーダー Mew は、現在の stable バージョンが 1.94.2 であり、1.95 beta の開発が進められている。1.95 の最も大切な目標は、Emacs の機能を最大限に生かす多言語化である。本校では、Mew 1.95 beta の多言語機能について述べる。

1 Mew 1.94 の問題点

バージョン 1.94 までに、Mew では優れた多言語機能が提供されてきた。例えば、さまざまな文字コードで書かれたメールをカスタマイズなしに表示できるし、草稿の文字コードを自動判別し適切なラベルの付いたメールを作成することも可能である。しかしながら、いくつかの問題点があった。

- バックエンドとして IM (Perl) を利用しているので、新しい文字コードに対応する場合、Mew と IM の両方を変更する必要がある。
- 復号化 (文字コードから Emacs の内部表現に変換すること) と符号化 (その逆) に同じ文字コード・データベースを用いていたので、ヘッダと本文を異なった文字コードで符号化できなかった。
- メールを選択するための外部コマンドに渡す引数の符号化が静的に決っていた。デフォルトの EUC-JP 以外を利用したい場合は、ユーザがカスタマイズする必要があった。
- Mew は Emacs がどの文字コードに対応しているのか動的に判断する機能を持っていなかった。このため、対応していない文字コードを復号化し、画面を乱すことがあった。
- 8bit のメールは作成できなかった。これは、書きかけの草稿と作成したメールを書き出す

際に、ISO-2022-JP2 相当の文字コードを利用していただけである。

2 Mew 1.95 での改良点

Mew 1.95 では、IM の利用を止め、ほとんどを Emacs Lisp で実現した。速度を稼ぎ、非同期性を実現するため 2 つの C コマンドを利用するが、これらのコマンドでは言語処理をしない。そのため、多言語に関する処理は、Emacs Lisp のみで対処できるので、保守拡張が容易になった。

文字コード・データベースを復号化用と符号化用に分離した。このおかげで、例えばヘッダが EUC-KR、本文が ISO-2022-KR というメールを作成できる。

また、外部コマンド用の文字コード・データベースも用意した。これにより、ユーザはカスタマイズをしなくとも、外部コマンドへの引数を適切な文字コードに符号化可能になった。

Emacs がどの文字コードに対応しているか動的に判別する機能を実現した。このため、たとえば UTF-8 に対応していないのに、復号化し画面を乱すことがなくなった。

多言語性が必要なファイルの文字コードは、すべて Compound Text に統一した。これは、従来の Emacs が 8bit 空間に Latin-1 (ISO-8859-1 の右) 割り当てているためである。Compound Text を利用すると、ISO-8859-1 と ISO-2022-JP, KR などうまく共存でき、Emacs との上位互換性も保たれる。

このため書きかけの草稿は、Compound Text で保存される。メールを作成するときは、バッファの内部で符号化し、結果をバイナリとしてファイルに書き出す。このおかげで、8bit のメールを作成することが可能になった。