

Mew 5.1 の新機能について

山本和彦
(株)インターネットイニシアティブ
kazu@iij.ad.jp

内容

- 検索機能
- セキュリティの強化
 - マスターパスワード
 - 宛先の警告
 - SMTP AUTH
 - 内容が同一であるメールの削除
- 改良されたコマンド
 - 本文の取得 ("I")
 - 添付ファイルの表示 ("C-cC-e")
 - 添付ファイルの分解 ("C-uB")

検索機能

背景

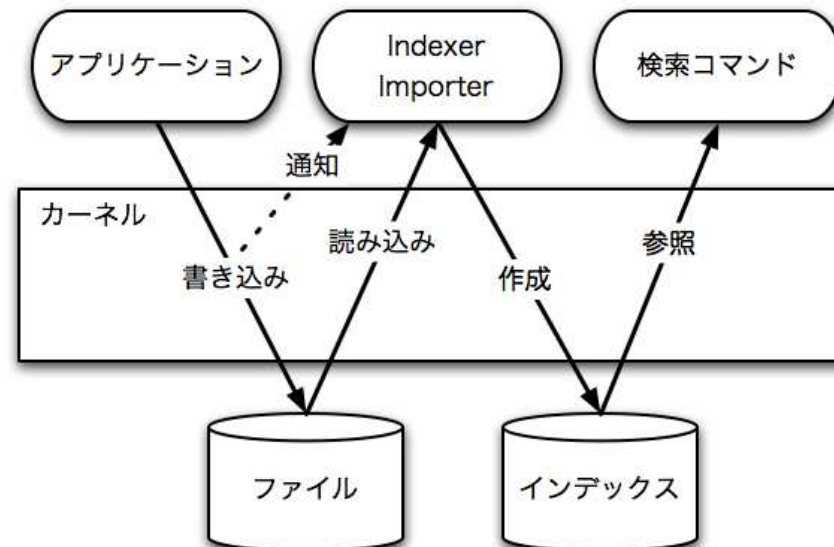
- Mew 4.2 までの検索機能
 - mewl を使ったヘッダ検索 (旧 "?")
 - あるフォルダの中で合致したメールに "*" マークを付ける
 - grep / mg を使った全文検索 (旧 "")
 - あるフォルダの中で合致したメールに "*" マークを付ける
 - Namazu を使ったインデックス検索
 - あるフォルダの中で合致したメールに "*" マークを付ける (旧 "k?")
 - 複数のフォルダの合致したメールから Virtual モードを作る (旧 "k")
- 新しいインデックス検索ツールの登場
 - Mac の spotlight
 - すべてのメールを対象にできる
 - Windows の Windows Desktop / Google Desktop
 - すべてのメールを対象にできる
- 新しいインデックス検索ツールを利用したい
 - すべてのメールに対して検索し、
キーワードに合致したメールで Virtual フォルダを
作りたい

インデックス検索ツール

- Namazu
 - UNIX, Mac, Windows
 - 1つのフォルダに対し、1つのインデックス
- Spotlight
 - Mac
 - すべてのメールに対し、1つのインデックス
 - データ型、あるいは拡張子
- Windows Desktop / Google Desktop
 - Windows
 - すべてのメールに対し、1つのインデックス
 - 拡張子がないとメールとして扱えない

Importer / Indexer

- インデックスの作成
 - Mac や Windows では、書き込みのイベントを他のアプリケーションに通知可能
 - 書き込まれたファイルの拡張子、あるいはデータ型に応じて、Importer / Indexer を呼び出す



実装の準備

- Spotlight
 - Importer は、井澤さんが実装
 - Emacs 22 からファイル型の設定
 - `mac_set_file_type()`
 - 非同期に設定するためは、C で実装する必要がある
 - David Reitter さん
 - 検索コマンドは "mdfind" を利用
- Windows Desktop
 - Indexer は Outlook Express 用を利用
 - 検索コマンドは、井上さんが "wdsgrep" を実装
- Google Desktop
 - Indexer は Outlook Express 用を利用
 - 検索コマンドは、北口さんが "gdsgrep" を実装

拡張子の導入

- Mew のメールを表す拡張子 ".mew"
- Mac
 - データ型 "MewX" に対し Importer を登録できる
 - Emacs 22 は、データ型を設定できる
 - Emacs 21 は、データ型を設定できない
 - Emacs 21 では、拡張子を利用
- Windows Desktop / Google Desktop
 - Outlook Express 用の Indexer を利用したい
 - 拡張子がないと Indexer が登録できない

拡張子の設計と実装

■ 設計

- "123" と "124.mew" が共存可能
- 好きなときに切り替え可能
 - (setq mew-use-suffix t)
- 内部形式は変えない
 - これまで通り数字のみ
 - Summary モードを作り直す必要がない

■ これまでの実装

- (mew-expand-folder "+inbox" "1")
→ "/home/kazu/Mail/inbox/1"

■ 新しい実装

- (mew-expand-folder "+inbox")
→ "/home/kazu/Mail/inbox"
- (mew-expand-file "+inbox" ".mew-summary")
→ /home/kazu/Mail/inbox/.mew-summary
- (mew-expand-msg "+inbox" "1")
→ /home/kazu/Mail/inbox/1.mew # "1.mew" が存在すれば
→ /home/kazu/Mail/inbox/1 # そうでなければ
- (mew-expand-new-msg "+inbox" "1")
→ /home/kazu/Mail/inbox/1.mew # mew-use-suffix が t なら
→ /home/kazu/Mail/inbox/1 # そうでなければ

検索書式

- mewl の検索書式から各ツールの検索書式へ
 - subject=mew-dist from=kazu
 - Linux ! Unix
- Namazu
 - +subject:mew-dist and +from:kazu
 - Linux not Unix (Linux and not Unix はダメ！)
- Spotlight
 - kMDItemTitle == "mew-dist"wc && kMDItemAuthors == "kazu"wc
 - kMDItemTextContent == "Linux"wc && kMDItemTextContent != "Unix"wc (動かない)
- Windows Desktop / Google Desktop
 - Linux -Unix

検索書式の変換

- 字句解析
- 構文解析
 - 各ツールで許される論理的な構文へ

```
(defun mew-pick-parse-and/or ()  
  (setq mew-inherit-pick-mewlp t)  
  (let ((cur (car mew-inherit-pick-tokens)))  
    (cond  
      ((or (stringp cur) (listp cur))  
       (unless mew-inherit-pick-omit-and  
         (setq mew-inherit-pick-ret (cons 'and mew-inherit-pick-ret)))  
       (mew-pick-parse-elements))  
      ...  
    )  
  )
```

- 検索書式の変換
 - 各ツールの検索書式へ

検索ツールの UI

- "k?"
 - あるフォルダ内でマッチしたメールに "*" マークを付ける
- "k/"
 - マッチしたすべてのメールで Virtual モードを作る
- "kc"
 - 検索ツールの切り替え

	"k?"	"k/"	インデックス作成
Spotlight	○	○	○
Windows Desktop	○	○	○
Google Desktop	○	○	×
Namazu	○	検討中	手動

これまでの検索コマンドの問題点

■ "*"マークを付ける

- "?"
 - `mewl` を呼び出す
- "C-u?"
 - リージョンに対し `mewl` を呼び出す
- "/"
 - `grep /mg` を呼び出す
- "'"
 - 呼び出すコマンドを指定できる

■ Virtual モードを作る

- "/"
 - `mewl` を呼び出す
- "C-u/"
 - `grep /mg` を呼び出す

■ 問題

- コマンド体系に統一感がない
 - "C-u?" vs "C-u/"
- リージョンに対して Virtual モードが作れない
 - 逆に "/" は、複数のフォルダーを対象にできる。。。

検索コマンドの改良

■ アイディア

- 入力キーワードによって `mewl` と `grep / mg` を切り替える
- キーワードに空文字が入力されたらコマンドを訊く
- `"C-u"` はリージョン
 - `"/` は、複数のフォルダを対象にできることをあきらめる

■ 新しいコマンド体系

- `"?"`
 - `"**"` マークを付ける
- `"C-u?"`
 - リージョンのメールに `"**"` マークを付ける
- `"/`
 - 1つのフォルダから Virtual モードを作る
- `"C-u/`
 - 1つのフォルダのリージョンのメールから Virtual モードを作る

■ 比較

- `"k?"`
 - `"**"` マークを付ける
- `"k/`
 - すべてのフォルダから Virtual モードを作る

セキュリティ強化

マスターパスワード

- マスターパスワード
 - 一つのマスターパスワードで他のすべてのパスワードを守る
 - 他のパスワードは、マスターパスワードで暗号化しファイルへ保存
 - (setq mew-use-master-passwd t)
- Mew の起動時
 - ユーザーはマスターパスワードを入力
 - 暗号ファイルをマスターパスワードで復号化し、パスワードをメモリへ格納
 - 暗号ファイルがなければ、新規にマスターパスワードを二回尋ねる
- Mew の終了時
 - メモリ中のパスワードをマスターパスワードで暗号化しファイルへ保存

宛先の警告

- 草稿を書く際、社外のアドレスを赤く表示
 - 例) "ij.ad.jp" 以外のアドレスを警告する
 - (setq mew-safe-domains '("ij.ad.jp"))
- 詳細
 - アドレスが 'mew-safe-addresses' で指定されていれば警告しない。
 - アドレスが 'mew-warn-addresses' で指定されていれば警告する。
 - ドメインが 'mew-safe-domains' で指定されていれば警告しない。
 - ドメインが 'mew-warn-domains' で指定されていれば警告する。
 - 'mew-warn-addresses' か 'mew-safe-domains' が定義されていれば警告する。
 - 警告しない。

SMTP AUTH

- SMTP サーバのユーザ認証
 - ユーザ認証に対応している場合、EHLO に対して AUTH ケーパビリティを返す

旧)

- mew-smtp-user を設定すると、パスワードを送る
- (setq mew-smtp-user "alice@example.jp")

新)

- SMTP サーバが、AUTH ケーパビリティを返すと自動的にパスワードを送る
 - ユーザ名は自動生成
 - ユーザ名のデフォルトは、From: のメールアドレスと同じ
- AUTH ケーパビリティを返すが、パスワードを受け付けないサーバが存在
 - ケーパビリティに AUTH があっても無視するオプション
 - (setq mew-use-smtp-auth nil)

内容が同一であるメールの削除

- 本文の先頭が同一である場合、"D" マークを付ける
 - (setq mew-summary-form-mark-spam t)
 - inbox フォルダのみ
- 先頭とは？
 - 引用などを除いた mew-scan-body-length 行分 (10行)
 - この MD5 チェックサムを取る
 - MD5 チェックサムは、scan/inc 中に一時的に作られる

改良されたコマンド

本文の取得 ("I")

- 本文を取得する

- メールが大きすぎて一部しか取得されていない ("T" マーク)
- ヘッダのみ取得した

旧)

- スレッドでは使えなかった
- 本文を取ってきても表示しなかった

新)

- スレッドでも利用できる
- 本文を取ってきたら表示する

添付ファイルの表示 ("C-cC-e")

旧)

- "C-cC-e" は、設定されている外部コマンドを起動
 - "C-uC-cC-e" で、ユーザがデータ型を指定できる
- "C-cC-t" は、ユーザが指定したデータ型に応じてバッファに表示
 - Outlook Express は、なんでもかんでも App/Oct
 - ファイル名の拡張子に頼る
 - 拡張子からデータ型を推測し、ユーザヘデフォルトの値として提示
- "C-cC-y" は、ユーザが指定した外部コマンドを起動

新)

- "C-cC-e" は、設定されている外部コマンドを起動
 - CT: が App/Oct ならユーザにデータ型を尋ねる
 - ユーザが Text を指定すると、バッファに表示
 - 必要に応じて外部かバッファかを尋ねる
- "C-uC-cC-e" は、ユーザが指定した外部コマンドを起動
 - CT: が App/Oct ならユーザにデータ型を尋ねる
- "C-cC-t" は破棄
- "C-cC-y" は破棄

添付ファイルの分解 ("C-uB")

旧)

- "B" は、本文に添付された「メール」の分解
- "C-uB" は、そのパートの「メール」を分解

新)

- メール上で起動された "B" は、本文に添付された「メール」の分解
- パート上で起動された "B" は、そのパートの「メール」を分解
- "C-uB" は、あらゆる添付ファイルを分解