

百万人のMutt

～Mutt活用講座～

滝澤隆史

tantaki@cyber.email.ne.jp

第9回 スコアリング



2001年12月にInternet Week 2001がパシフィコ横浜で開催されました。読者の中で参加された方もいるかと思いますが、その最終日に、日本UNIXユーザ会主催の「The Linux/BSD Day」というイベントにおいて、筆者はMUAのセッションでMuttの紹介を行いました。初っぱなから司会者がMuttをどう読むかわからなかったようで、「エム・ユー・ディー・ティ」などと言っていたので、筆者が「マットと読みます」と訂正した、というエピソードも含めてやはり日本国内では知名度が低いのだなぁと思ってしまいました。

そのセッション内で挙手によるアンケートを行ったのですが、Muttユーザーは約200人中6人(3%)と、やはり少ない結果となりました*1。一方、未だにmailコマンドやmhを使用しているユーザーが多かったのには驚きました。この結果は「単に昔から同じMUAを使い続けている人が多いだけなのだともいえます。とはいえ、Sylpheedのユーザーの方が多かったのですが。

一応、負け惜しみを言う、Muttは世界的に見れば日本国内におけるMewと同じくらい人気のあるMUAなのですが……。日本語に正式に対応する次の安定版1.4が出て、各ディストリビューションに標準で入ってから本領発揮するといったところでしょうか。

その1.4ですが、まだベータの状態でバージョンは1.3.25iです。この1.3.25iではリモートから攻撃される恐れのあるセキュリティホールに対する修正を行っているため、1.3系列を使っている人は1.3.25iにバージョンアップするようお願いします。

また、安定版の1.2.5iにも同様の問題があり、1.2.5.liにバージョンアップしました。しかし、この1.2.5.liはセキュリティホールの修正だけを行ったものであり、他のバグの修正は行われていません。そのため、特に1.2系列を利用し続けたい理由がなければ、より安定している1.3.25iの方をお使いください。

前置きはともかくとして、今回はスコアリングについて解説を行います。

スコアリング

スコアリングとは

スコアリングとは、メッセージにスコア(点数)を付ける機能です。この機能はニュースリーダを使っている人には馴染みの機能だと思います。投稿数の多いメーリングリストなどに対して、「特定の人の投稿だけ読みたい」、「特定のキーワードが含まれるメッセージだけ読みたい」といった場合に重宝します。またその逆に「この人のメッセージだけは読みたいくない」、「spamらしきメッセージを読みたいくない」という場合にも使えます。

設定変数

スコアリングに関する設定変数は表1のようなものがあります。

Muttでスコアリングを使うには`$score`を設定しますが、これはデフォルトで設定されていますから特に何もしなくてもいいでしょう。特定のフォルダのメッセージに対して、適応したい/適応したくない場合は`folder-hook`で設定したり解除したりしてください。例えば、「friends」というフォルダにスコアリングの機能を適応したくない場合には、次のように設定します。

```
folder-hook . 'set score=yes'
folder-hook friends 'set score=no'
```

スコアリングを設定すると、スコアの合計値と閾値の関係からメッセージに対して「削除フラグDを付ける」、「既読にする」、「重要フラグIを付ける」ことが自動的に行えます。

削除と既読に関しては閾値以下の場合に適応され、重要フラグの場合は閾値以上の場合

に適応されます。スコアの合計値が負の値になった場合は「0」とみなされます。

表1を見ても分かるように、削除と既読のデフォルト値は無効な閾値になっていますから、利用する場合には有効な閾値を設定し直す必要があります。例えば、「値が0以下の場合に削除する」、「49以下の場合に既読にする」、「100以上の場合に重要フラグを付ける」という場合には次のように設定します。

```
set score_threshold_delete=0
set score_threshold_read=49
set score_threshold_flag=100
```

これらの閾値は、次の項で説明する`score`コマンドと組み合わせで慎重に設定してください。不用意な設定を行うと、メッセージがすべて削除されたり、既読になってしまうようなことが起こります。特にそうする必要がなければ、削除の閾値はデフォルトのまま無効にしておいた方が良いでしょう。

スコアの付け方

スコアを付けるには設定コマンド`score`を使います。記述方法は次の通りです。

```
score < pattern > < value >
```

<pattern>にはパターンを記述します。ただし「`~h`」、「`~b`」、「`~B`」は利用できません。パターンについて詳しくは、Muttマニュアルの「4.2. パターン」(記事末のResource[1])を参照)を参照してください。また、本誌2001年8月号掲載「第2回パターン」でも説明を行っていますので、そちらもご覧ください。

<value>にはスコアの値を記述します。スコアの値は、各`score`コマンドで適応されたスコアの合計値を正または負の整数で指定しま

【表1】スコアリングの設定変数

設定変数	機能	デフォルト値
score	スコアの適応	yes
score_threshold_delete	削除フラグ適応の閾値	-1
score_threshold_read	既読適応の閾値	-1
score_threshold_flag	重要フラグ適応の閾値	9999

*1 質疑のメモは<http://www.jus.or.jp/bsd-linux-day/>にあります。

す。ただし「=50」のように「=」を付けて値を設定した場合には、その値が適用されるようになります。

scoreコマンドで設定したスコアのエントリを削除する場合にはunscore コマンドを使います。記述方法は次の通りです。

```
unscore <pattern> [ <pattern> .....]
```

<pattern> には、score コマンドで設定したパターンのうち、解除したいパターンを記述していきます。すべてのエントリを解除するには「*」とだけ記述します。

スコアの初期値は「0」です。削除フラグや既読フラグの閾値を「0以上」に設定している場合、すべてのメッセージに削除フラグや既読フラグが付くようになります。しかし、このような動作が好ましくない場合には、すべてのメッセージに一致するパターン「~A」を使い、設定した閾値よりも大きい値を設定します。例えば、削除の閾値「0」、既読の閾値「49」とした場合、「削除したくはないが既読にしてもよい」ときは、次のようにすべてのメッセージに「40」を加えるようにします。

```
score ~A +40
```

特定のドメインのメールアドレスからのメッセージや、Subject に好ましくない特定の単語が含まれているメッセージを削除したい場合には、次のように負の値や「0」を指定します。

```
score '~f freemail\.example\.org$' -40
score '~s "(お得|副収入)"' =0
```

特定の人が投稿したメッセージや、特定のSubject を含むメッセージを既読にしない場合には、次のように既読の閾値より大きい値にするようにします。

```
score '~f foo@example.org' +10
score '~s announce' +20
```

パターン「~A」で加える値を「50」に変える場合には、次のように一度解除してから設定をし直します。これは、フォルダごとにスコアの設定を変える場合などに使用します。

```
unscore ~A
score ~A +50
```

スコアの表示

リスト1のように、設定変数\$index_format のフォーマット中にスコアを示す「%N」(ここでは3桁表示させるため「%3N」)を記述すると、リスト2のようにインデックスにスコアを表示することができます。

スコアのパターン

パターン「~n MIN-MAX」で特定の範囲のスコアに対して操作を行うことができます。例えば、スコアの値が50以上のメッセージだけを

【リスト1】インデックスに %N を使って3桁のスコアを付ける

```
set index_format="%4C %Z (%3N) %{b %d} %-15.15L %s"
```

【リスト2】リスト1の結果(インデックスに3桁のスコアを付ける)

```
3442 s ( 40) Nov 30 Prahlad Vaidyan patch-1.3.23.cd.edit_threads-6[C.YA] not
3443 Ns ( 50) Nov 30 Thomas Roessler [Announce] mutt-1.3.24i is out (BETA).
3444 N ( 50) Nov 29 Dairy Wall Lime +->Re: [Announce] mutt-1.3.24i is out (BE
```

【リスト3】特定の人の新着メッセージのみを開くようにする

```
# (1) デフォルト score なし、maillist フォルダ score あり
folder-hook . 'unset score'
folder-hook . 'reset score_threshold_read'
folder-hook maillist 'set score'
folder-hook maillist 'set score_threshold_read=49'

# (2) 特定の人を +50
folder-hook maillist 'unscore *'
folder-hook maillist 'score "~f foo@example\.com$" +50'
folder-hook maillist 'score "~f bar@example\.com$" +50'

# (3) maillist フォルダ未読メッセージ積み込みなし
folder-hook . 'set collapse_unread'
folder-hook maillist 'unset collapse_unread'
folder-hook maillist 'push \eV'
```

表示する場合は「l」(limit) の入力後に次のパターンを入力します。

```
~n 50-
```

スコアによるソート

設定変数 \$sort あるいは \$sort_aux に「score」や「reverse-score」を設定することによって、インデックス画面でスコアによるソートが可能になります。また、インデックス画面に置いて「o」(sort-mailbox) や「O」(sort-reverse) を入力し「c」(score) を入力することでもスコアによるソートが可能です。

スコアリングの利用例

スコアリングの利用例を2つほど紹介します。

利用例1

非常に投稿数の多いメーリングリストがあるとして。しかも、あまり意味のないメッセージが非常に多く、すべてのメッセージを読み切ることではできず、しかも読む価値がない。しかし、ごく一部の人のメッセージに関しては、読んだり議論したりする価値があります。このような場合、新着メッセージの中で、特定の人のメッセージ以外を既読にして、新着メッセージとしては拾わないように設定してみます。

まず、このメーリングリストのフォルダにのみスコアリングを適用する場合はリスト3の(1)のように設定します。このとき既読フラグの閾値を「49」とします。続いて、特定の人のメッセージを既読にしないために、リスト3の(2)のようにその人からのメッセージに対して「+50」を加えます。最後に、リスト3の(3)のように設定して、フォルダを開いたときに新着メッセージがあるスレッド以外のスレッドを積み込むようにします。なお、デフォルトでスレッド表示する設定が行われているとします。

ここで、\$collapse_unread は未読のメッセージがあるスレッドを積み込む設定で、「push \eV」はフォルダを開いたときに「Esc V」キーを入力することを表しています。「Esc V」は、すべてのスレッドを積み込み込むかを開くかを切り

【画面1】リスト3の実行結果(特定の人の新着メッセージのみを開く)



替える collapse-all 機能のキーバインドです。

以上のように設定すると、フォルダを開いたときに、特定の人のメッセージの新着メッセージのあるスレッドのみが開き、その他のスレッドが畳み込まれて画面1のような状態になります。これで余計なメッセージを読まなくて済むようになります。

利用例2

spam対策用にスコアを設定してみましょう。次に挙げる条件に一致するメッセージを削除したり、既読にしたり、別のフォルダに移動したりします。

- ・特定のメールアドレス
- ・特定のドメイン
- ・メーリングリスト宛でも自分宛でもない
- ・Subject に特定の単語を含む

このような条件設定なら、procmailやmaildrop

などのフィルタリング可能なMDAでもできることですし、使い勝手もその方が良いでしょう。しかし、Muttのスコアリング機能には次のような利点があります。

- ・削除前の確認が行えるため、設定ミスによるメッセージ紛失の危険性が少ない
- ・日本語のSubjectを条件にすることが可能
- ・MDAを使わなくてもMuttだけで可能

そこで、ここで説明する利用例では、MDAによるフィルタリングを行わず、スプールフォルダのメッセージに対するスコアリングを行ってみましょう。条件を表2、設定例をリスト4に示します。

まず、リスト4の(1)のように、デフォルトでは既読と削除のスコアリングを適用しないように設定しておきます。これは、スコアリングで削除フラグを付ける場合、設定を誤るとすべてのメッセージを削除してしまう可能性があ

るからです。

続いて、削除の閾値を「0」、既読の閾値を「49」に設定し、通常のメッセージのスコアは既読の閾値以上の値(ここでは50)になるようにします。設定はリスト4の(2)のようになります。なお、削除せずに「spamコレクション用のフォルダに移動させたい」という物好きな人は、\$score_threshold_deleteを設定せずに削除フラグの閾値はデフォルトのままにしておいてください。

条件(a)に一致する場合は確実にspamであると分かっているの、スコアを「0」にして削除するのは問題ないでしょう。設定例はリスト4の(3)のようになります。条件(b)~(d)はspamと断定はできないので、とりあえずスコアを減らして既読にし、新着メッセージとして拾わないようにします。この中で、2つ以上の条件に一致する場合はspamである可能性は高いと考えられます。そこで該当メッセージを削除するために、2つの条件でスコアが「0」になるように、それぞれのスコアを「-25」とします。設定例はリスト4の(4)~(6)のようになります。

以上のように設定すると、スプールフォルダを開いたときには、疑わしきメッセージは既読になり、spamと判断されたメッセージには削除フラグが付きま

また、削除しないで、spamコレクションとして他のフォルダに移動させる場合には、リスト5のようなマクロを記述してみると良いでしょう。

最後に

実を言うと、筆者はこの記事を書きおきながらスコアリング機能はほとんど使っていません。今回の記事を書くにあたって利用例を考えてみたのですが、利用例1に関しては、実際に使ってみてこれは結構重宝すると思いました。利用例2に関しては、個人的にはmaildropでのフィルタリングで十分に満足しているため使わないでしょう。「spam避け」という意味ではヘッダに記述されていることがあるX-Mailerフィールドなどでフィルタリングするのが効果的です。これは大抵のspamが特定のバルクメーラーで送られているからです。しかしMuttのスコアリングでは、特定のヘッダに対してしかパターンが使えないため、X-Mailerフィールドでパターンを一致させるということができません。そのため、maildropなどのMDAでフィルタリングをした方が効果的です。

【リスト5】spamコレクション用マクロ

```
macro index \eS 'T~n <50~m;s=spam~my'
```

【表2】spam対策用スコアの設定

条件	設定内容	設定例
(a) 特定のメールアドレス	あらかじめ spam と分かっている特定のメールアドレスから来たメッセージのスコアを「0」にする	リスト4の(3)
(b) 特定のドメイン	特定のドメインから来たメッセージのスコアを「25」減らす	リスト4の(4)
(c) メーリングリスト宛でも自分宛でもないメール	スコアを「25」減らします。メーリングリスト宛のメッセージのスコアを「+25」にしているのは自分宛でないメッセージで「-25」になるのを相殺するため	リスト4の(5)
(d) Subject に特定の単語を含むメール	Subject に特定の単語が含まれているメッセージのスコアを「25」減らす	リスト4の(6)

【リスト4】spam対策用スコアの設定

```
# (1) デフォルト score なし
folder-hook . 'unset score'
folder-hook . 'reset score_threshold_delete'
folder-hook . 'reset score_threshold_read'

# (2) 閾値と初期値を設定
folder-hook '!' 'set score'
folder-hook '!' 'set score_threshold_delete=0'
folder-hook '!' 'set score_threshold_read=49'
folder-hook '!' 'unset *'
folder-hook '!' 'score ~A 50'

# (3)(a) 特定のメールアドレス
folder-hook '!' 'score "~f spam@example\.org$" =0'

# (4)(b) 特定のドメイン
folder-hook '!' 'score "~f freemail\.example\.org$" -25'

# (5)(c) メーリングリスト宛でも自分宛でもない
folder-hook '!' 'score "!~p" -25'
folder-hook '!' 'score "~C mutt-users@mutt\.org$" +25'
folder-hook '!' 'score "~C mutt-dev@mutt\.org$" +25'

# (6)(d) Subject に特定の単語を含む
folder-hook '!' 'score "~s \"(お得 | 副収入)\\"" -25'
folder-hook '!' 'score "~s \"(bulk|shop|market)\\"" -25'
```

単なる Mutt ユーザーの na6ce です。

今まで執筆された方々と違い、Muttの発展には何も貢献していない私ですが、なぜかお声が掛りました。開発関係者以外でもこんな使い方ができるんだ、程度に思っていただければ幸いです。

Mutt との出会い

私がMuttと出会ったのは「Kondara MNU/Linuxに付いていた」それだけ。……ではあまりにもひどすぎるので、そもそもKondaraを使い始めた経緯からお話しますね。

私がLinuxを始めたきっかけはPerlでした。理由は、単純に「タダで開発環境が手に入る」程度でした(後に、私が使っているプロバイダではCGIを許していないことが分かって泣いたのですが)。最初はVine Linuxを使っていたのですが、残念ながらXが起動しません。いろいろ試しても動かず、Windowsを起動して情報収集してみると、使っているビデオカードはVineでサポートされていないものだったのです。さらに調べてみると、Red Hat Linux 6.0ではサポートされている模様。そこで、当時1.0リリース前のKondaraを使い始めました。

でもLinux自体の知識がないので、Webやメールなんてできません。今はダイヤルアップ設定ツールなどがあるので、ほとんどの人はそんな経験はないでしょうが、それどころか、起動はできてもshutdownコマンドを知らなかったので24時間ブローンリ。当時は情報収集にWindowsを使っていたのですが、マシン1台でHDDを差し替えてWindowsとLinuxを使い分けていたので、shutdownコマンドを知らない私は、

本体の電源ブチ(!!)

HDDを入れ替えてWindowsを起動

プリントアウト

Windowsを終了

HDDを入れ替えてLinuxを起動

なんてスゴイことをやっていました。

さて、そんなこんなで少しずつコマンドを覚えていき、それなりにLinuxが使えるようになった頃です。PPxPなんて素晴らしいツールが(自分のHDDの中に)存在する驚愕の事実を知ったのは……、と思

い返すと恥ずかしいことばかりやっていました。

WebブラウザにはNetscape Navigatorを使っていたので、しばらくはNetscapeのメーラを使っていました。しかし当時はとにかく落ちまくるし、アドレス補完の問題もあって、日常使うにはちょっとためられるものがありました。そこで「自分に合ったメーラを探そう」と探しはじめて早3年(ウソ)、Muttというメーラ(正確にはMUAですが)を見つけました。

All mail clients suck, this one sucks less.

この自信に打たれました。これまた、自分のHDDの中にはすでにMuttが存在する事実にも驚きながらも、プロンプトから「mutt」と打ち込んでみました。しかし、メールを送るところが読むことすらままなりません。最初の感想。「使えねー」。

ズルズルと引き込まれる

第一印象は激悪だったMuttですが、私にしては珍しく「メールの送受信ができるまでやってみよう」と思いました。幸い、英文を読むのに不便はなかったもので、docやFAQを読んでいくにつれ、少しずつMuttの仕組みが分かってきました。

Mutt中毒になるコツを1つ。メーラで「やりたいこと」を持つことです。Muttなら、ほぼ確実にその「やりたいこと」が実現できるでしょう。私の「やりたいこと」は、メーリングリストから来たメールを自動的に専用のフォルダに入れることでした。これはすぐ実現できました。厳密にはMuttでなくprocmailですが。

その後は無茶ばかりしていました。Perlスクリプトでキーバインドをランダムに変更して(.muttrcの

書き換え)。押すとシャットダウン(しかもsudoで)してしまう地雷スクリプトを作ったり、ゴリ押しでspamを排除するスクリプトを作ったり……。これらは私のページに置いてあります([2])。かつての私にとって、メーラとは「メールの読み書きを行うもの」という認識だけだったので、これだけ無茶できるとなったら、もう離れられません。もっとも今は落ち着いていますけど……。

今、一番使う機能はパターンマッチです。Tキーでタグ付けモードにし、「~d 01/01-31/01」とすると、dateフィールドが1月1日~1月31日までのメールにすべてタグを付けることができます。その後、タグ付きメールに対して;キーでtag-呼び出し、sキー(save)でまとめて保存しています。

私はMaildirを使っていますが、メールのアーカイブを取るときにはmbox形式で保存しています。追記がありません。1つのファイルだと検索が楽です。昔はこれをマクロでF5キーに割り当てていました*2。

あと、dキー(delete)を押すと即座に削除してしまうので、リストAのような設定ではなく、deletedというフォルダを作成し、削除候補のメールはいったんこのフォルダに移してから削除するようにしました(deletedに関するfolder-hookはデフォルト指定のものより後に書かないとエラーになります)。これで誤ってメールを削除することがなくなりました。

と、このような感じでMuttを使っています。今挑戦(&停滞)中なのは、spamメールのデータベース化。MySQLと連動してspamを排除しようとしています(が、いつまで経っても完成しません)。暇があったら私のページを覗いてみてください。何かまた無茶しているかもしれません。(na6ce)

【リストA】選択中のメールをdキーで消してしまう設定

```
folder-hook . 'set delete=yes'
folder-hook . 'macro index d "s=deleted/ny" [delete]'
folder-hook . 'macro pager d "s=deleted/ny" [delete]'
```

【リストB】delete フォルダに退避してからメールを削除する設定

```
folder-hook \!/deleted/ 'set delete=ask=yes'
folder-hook \!/deleted/ 'bind index d delete-message'
folder-hook \!/deleted/ 'bind pager d delete-message'
```

*2 興味のある方は私のページ(<http://home.att.net/~na6ce/archive.html>)を参照ください。

そういえば、昨年の夏ごろから韓国からたくさんspamが来るようになりました。HTMLなメールの上に韓国語で書いてあるので読めません。無駄な努力はいり加減やめてほしいものです。もっとも、読もうと思えば読めないこともないのですが。ちなみに、本文中に書いたspamコレクション用のフォルダに移動させている物好きな人とは筆者のことです。

Resource

[1] Japanese Mutt Manual

<http://ha4.seikyou.ne.jp/home/Kohtaro.Hitomi/mutt/>

[2] Mutt王 Mutt マスターを目指す (2002年1月10日現在、休止中です)

<http://home.att.net/~na6ce/mutt.html>

・ Mutt Japanese Edition

<http://www.emallab.org/mutt/>