ML のアーカイブを用いた FAQ 自動生成システム

02H080 西川一美

目次

1 はじめに 1

2 メーリングリスト 2

3 シェルスクリプト 2
- 3.1 シェルについて . . . 2
- 3.2 bash (Bourne Again Shell・bash) . . . 2

4 システム 3
- 4.1 システムの目的 . . . 3
- 4.2 システムの概要 . . . 3
- 4.3 プログラムの流れ . . . 3

5 プログラムの検証 4
- 5.1 検証方法 . . . 4
- 5.2 検証結果 . . . 4
- 5.3 グラフ . . . 10
- 5.4 問題点 . . . 10
- 5.5 今後の課題 . . . 10

6 考察とまとめ 11
- 6.1 考察 . . . 11
- 6.2 まとめ . . . 11

付録 A プログラム 12
- A.1 プログラムソース . . . 12

付録 B サンプルメール 15
- B.1 方法 1 疑問文よりメール本文抜粋 . . . 15
- B.2 方法 1 質問文よりメール本文抜粋 . . . 17

1　はじめに

現在 IT 産業の発展により、インターネットを利用したサービスも数多く生まれた。その中でよく利用されているのは電子メールである。最近では携帯電話を使って、いつでもどこでもメールが利用できたり、フリーのメールアドレスを簡単に取得できるようになった。そのため、メールは身近なものになってきている。今日では、電子メールは様々な世代、場面 (趣味、仕事) で活用されている。この電子メールを使ったシステムの一つとして、メーリングリストがあげられる。

メーリングリストは、複数の人に同時に電子メールを配信する仕組みのことで、ML と略されるシステムである。このシステムによって、今まで直接会って情報のやり取りをしていたのが、離れていても瞬時に情報を送れようになった。また、待っていても情報が入ってくるようになった。

本研究では、メーリングリストのアーカイブを用いた分別システムの構築を行う。大学という限られた空間で学生が利用するものを対象にする。目的は、授業で役立つシステムの構築である。メールを利用したレポートの提出をメーリングリストを使うことで、学生は自分以外のレポートを閲覧することができる。それにより、よりよいレポートを作成するのに役立てることができる。他にも、メーリングリストを使ったシステムが作れるだろう。その為には、学生から見て、今何が必要とされるのかを考えてみるとよいと考える。

本研究で構築する機能とは、過去のメーリングリストのデータを使い、質問メールとそれ以外を分別するシステムである。

最終的には、取り出した質問メールのデータをリスト化して、疑問を解決できる機能を実装化させることが目標である。

2　メーリングリスト

メーリングリストとは、電子メールを使って、特定のテーマについての情報を特定のユーザの間で交換するシステム。複数のユーザを 1 つのグループとしてメールサーバに登録し、情報を同時配信することにより実現している。インターネット上には無数の ML が開設されており、グループの性格によって、広く参加者を募る公開型と、特定の集団の内部利用に限定された閉鎖型に大別される。

参考文献の [7] から抜粋

3　シェルスクリプト

本研究のシステムを構築するのにシェルスクリプトを使う。シェルスクリプトとは、シェルで実行される Script(台本の意) です。コマンドラインで使用するシェルに対して、ユーザがコマンドを入力する代わりにファイルから命令を読み込んで実行させるしくみである。

参考文献の [5] から抜粋

3.1　シェルについて

「シェル」という言葉には、広い意味と狭い意味の 2 つの意味がある。

広い意味でのシェルは、コンピュータを操作するときに我々が使用する基本的なソフトウェア全般をさしている。最も基本的で、かつ頻繁に使用されるソフトウェアであるため、ときにはオペレーティングシステムそのものと捉えられる場合もある。我々はシェルを通じてコンピュータに指示を与えて、ファイルやディレクトリを操作したり、さまざまなプログラムを実行したりすることができます。広義のシェルに含まれるものとしては、Windows の「Explorer」や Mac OS の「Finder」などがある。

狭い意味でのシェルは、UNIX を操作するときに使用するコマンドラインインターフェースを指している。ユーザはシェルに対してキーボードからコマンドを入力することで UNIX を操作することができる。操作の目的自体は広い意味でのシェルと同様、ファイルやディレクトリの操作やプログラムの実行を行います。狭義のシェルに含まれるものとしては「B Shell」や「C Shell」、「Bourne Again Shell (bash)」などがある。

参考文献の [5] から抜粋

本研究では、後者の狭義のシェル「bash」を使う。

3.2　bash (Bourne Again Shell・bash)

bash は B Shell に対して機能を拡張する形で作られた B Shell 互換のシェルである。GNU Project で開発が行われている。

おもな拡張機能としてはコマンドラインに入力した文字列を編集する機能や無制限のコマンドライン履歴などがある。Linux では bash が標準ログインシェルとして使用されている。

参考文献の [5] から抜粋

4　システム

4.1　システムの目的

システムの目的は、既存のメーリングリストから効率よく有効な情報を手に入れ、授業で使用しているメーリングリストを対象に、もっと便利に (面白く) 活用できるようにすることである。

また、過去のデータを使って、メーリングリストで同じ質問に答える二度手間 (無駄な行為) をなくす。そして、質問メールを過去のデータと照らし合わせ、同じ内容があれば過去のスレッドのリストを返すようにする。

ユーザは、わからないことがあれば FAQ のリストから捜し出して解決することが期待できる。

4.2　システムの概要

授業で使用するメーリングリストでは、質問メールが何通も届く。すべてのメールをチェックするのは大変である。また、同じ内容の質問に何度も答えることは手間のかかる行為である。そこで、その為にメーリングリストのデータベースから質問メールとそれ以外を分け、質問とそれに連なるスレッドをリスト化し FAQ を生成する。学生の質問メールを受けて、もし過去に同じ内容のものがあれば、過去に同じ質問があったことを教える文と該当する情報 (スレッドリスト) を過去のデータから取り出してメール本文を生成し、発信する。しかし、過去のデータから該当するものがない場合は解答文を作成して発信する。

4.3　プログラムの流れ

ひとつのデータベースに入れられたメーリングリストのデータをプログラムにかける。

プログラムは

1. メールのヘッダーとボディを分離
2. ボディから加点フレーズの行を抽出
3. ボディから減点フレーズの行を抽出
4. 加点フレーズの行数をカウントし、変数 neet に入れる
5. 減点フレーズの行数をカウントし、変数 poo に入れる
6. 加点カウント数から減点カウント数を引き、残った数を変数 tani に入れる。
7. 残った数が 4 以上のメールだけをピックアップして question の箱に入れる
8. 該当しないものは、other の箱に入れる。

こうして、プログラムによって分離されたものを最終的に一覧表示して、疑問解決の際に役に立てられることを期待する。

5　プログラムの検証

5.1　検証方法

　どれぐらいの正しい割合で分別されるか実験結果をここでは報告する。検証用として、大垣先生からサンプルデータを借りた。次にあげるのは、実際に行ったプログラムの性能がどれだけのものかを検証する方法である。

方法 1

- 加点法のフレーズを 1 つだけ残し、どれだけ取り出せるかを検証

方法 2

- 加点フレーズを含む行をカウントしていき、どれだけ取り出せるかを検証

方法 3

- 減点フレーズを含む行をカウントしていき、どれだけ取り出せるかを検証

方法 5

- 方法 2 のカウント数から方法 3 のカウント数を引き、3 カウント以上あるメールだけを取り出して、どれだけあるかを検証

5.2　検証結果

　次にあげるのは、上記の方法で検証した結果である。なお、メール数のよこに付随している括弧の番号は、取り出した内の質問メールが少ない場合、プログラム改善の為、どのメールかを把握するために該当するメールを判別する為のものである。

1. 方法 1 の検証結果
   加点フレーズには大きく二つの種類に分けられる。それは、疑問文と質問文である。
   疑問文は、質問する内容のことである。質問文は、疑問を問うことである。
   つまり、「C 言語がわからないので教えてください」という文なら、「わからない」が疑問文で、「教えてください」が質問文となる。
   ここでは、二種類のフレーズでもよく使われているものを加点フレーズとして、どれだけの確率で取り出されているか検証している。
   結果は
   (a) 取り出せたメール数
   (b) 取り出した内の質問メールの数
   (c) 取り出したメール数に対する取り出した質問メール数の割合
   (d) 取得率
   以上の順番にそれぞれ表示している。
   なお、サンプルメールは 498 通、そのうち質問メールは 77 通ある。

(a) 疑問文

i.「でしょうか」

- 235 通
- 60 通
- 26 %
- 78 %

i.「?」

- 225 通
- 42 通
- 19 %
- 55 %

i.「ませんか」

- 472 通
- 60 通
- 13 %
- 78 %

i.「のですが」

- 213 通
- 51 通
- 24 %
- 66 %

i.「のですけれど」

- 1 通
- 1 通 (105030) 付録に添付
- 100 %
- 1 %

i.「どうやって」

- 11 通
- 1 通
- 9 %
- 1 %

i.「どのような」

- 14 通
- 4 通
- 29 %
- 5 %

i.「できましたが」

- 5 通
- 1 通 (105127) 付録に添付
- 20 %
- 1 %

(a) 質問文

i.「方法」

- 98 通
- 25 通
- 26 %
- 32 %

i.「ご教授」

- 48 通
- 20 通
- 42 %
- 26 %

i.「質問」

- 41 通
- 11 通
- 27 %
- 14 %

i.「わからず」

- 11 通
- 1 通 (105046) 付録に添付
- 9 %
- 1 %

i.「解決」

- 45 通
- 6 通
- 13 %
- 8 %

i.「どなたか」

- 19 通
- 11 通
- 58 %
- 14 %

i.「お願い」

- 93 通
- 45 通
- 48 %
- 58 %

i.「ありましたら」

- 15 通
- 9 通
- 60 %
- 12 %

i.「ご存知」
- 16 通
- 10 通
- 63 %
- 13 %

i.「ご存じ」
- 6 通
- 3 通
- 50 %
- 4 %

i.「皆様の」
- 6 通
- 1 通 (105208) 付録に添付
- 17 %
- 1 %

i.「みなさま」
- 3 通
- 2 通
- 67 %
- 3 %

i.「いらっしゃいましたら」
- 22 通
- 11 通
- 50 %
- 14 %

i.「教え」
- 53 通
- 19 通
- 36 %
- 25 %

i.「ありますか」
- 15 通
- 5 通
- 33 %
- 6 %

以上のように加点フレーズひとつずつで検索すると、質問メールの取得率は最大で 78 %、最小で 1 %になる。また、質問以外のメールが入って来る確率も多いので、対象を絞りこむために複数条件を付ける必要がある。

1. 方法 2 の検証結果

ここでは、方法 1 で挙げた加点フレーズを全部使えばどうなるかを検証したものである。その際、引用文に含まれる加点フレーズは除外するようコマンドを付け加えて実行している。

(a) 方法 2 － 1

- 取り出したメール：336 通
- 質問メール：77 通
- 確率：23 %
- 取得率：100 %

ほとんどのメールが取り出されている。そのため、質問メールと取り出されたメールの割合が低い。しかし、質問メールが全て含まれるため、取得率は高い。

(a) 方法 2 － 2

方法 2 － 1 で出た結果をさらに絞るため、1 フレ- ズあたり 1 ポイントとカウントし、3 ポイント以上のものだけを取り出す。

- 取り出したメール：152 通
- 質問メ- ル：70 通
- 確率：46 %
- 取得率：90 %

方法 2 － 2 は方法 2 － 1 よりも正答率は落ちてしまうが、取り出す質問以外のメ- ルを取り出す確率は下がる。それは、メール文中に加点する該当フレーズが少なかった為にはじかれたからである。しかも、文章が短か過ぎて加点フレーズを増やすことも不可能なメールもあるためメール数をしぼりつつ質問メールを全て捕獲するのは難しい。

(a) 方法 2 － 3

方法 2 － 1、2 － 2 では引用文の判定を先頭に「 > 」があるときに限定していたが、例外もあるため行中に「 > 」が含まれるものを除くように変更して実行する。

- 取り出したメ- ル：145 通
- 質問メ- ル：71 通
- 確率：49 %
- 取得率：92 %

正答率は方法 2 － 1 よりは落ちるが、方法 2 － 2 よりは若干上がっている。また、取り出すメ- ルの数は方法 2 － 1、2 － 2 よりも絞りこむことができるようになった。

1. 方法 3 の検証結果

(a) 方法 3 － 1

リタ－ ンメ－ ルと判断したメ－ ルを取り出す。

- 取り出したメ－ ル：343 通
- 質問メール：0 通

リターンメールには、質問メールの返答が多い。また、質問しているメールも含まれているが、質問の返答からさらに質問するという形であった。そこで、リターンメールは質問メールと判断しない。つまり、スレッドの先頭部分にある質問メールのみを質問メールと判断し、リスト化した時に、返信文等同じスレッドに含まれるものをひとまとめに表示する。

(a) 方法 3 － 2

3 つの減点フレ－ ズをカウントしていき、1 つでも該当する行あれば取り出す。

- 取り出したメール：181 通
- 質問メール：8 通 (105134、105217、105208、105266、105328、105346、105348、105430)

1. 方法 4 の検証結果

ここでは加点と減点の両方を行い、計算する。

(a) 方法 4 － 1

3 カウント以上あるメールだけを取り出す。

- 取り出したメールは 116 通
- 質問メールは 67 通
- 確率は 58 %

(a) 方法 4 － 2

3 カウント以上あるメールから、さらにリタ－ ンメ－ ルを除外したものを取り出す。

- 取り出したメールは 71 通
- 質問メ－ ルは 67 通
- 確率は 94 %

以上の結果からかなり正確に取り出すことができると考えられる。

5.3 グラフ

方法 2 と方法 4 を検証した結果をグラフにして表示する。

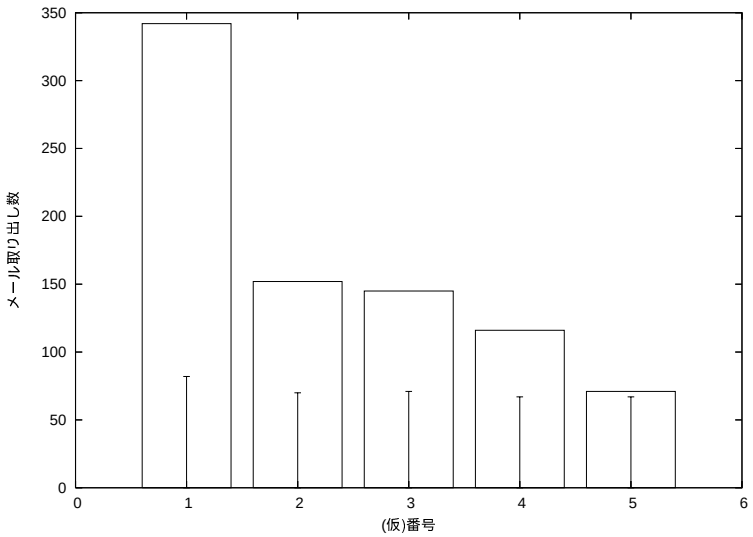

図 1　方法 2 と 4 の検証グラフ

左から 1 から 6 と棒グラフが並んでいる。順番に 1 は方法 2-1、2 は方法 2-2、3 は方法 2-3、4 は方法 4-1、5 は方法 4-2 の検証結果である。それぞれ、棒グラフが取り出されたメール数、線グラフが取り出された内の質問メール数となっている。

5.4 問題点

プログラム作成中に問題となった点を以下にあげる。

1 . 実際に人が見ても質問かそうでないのか判断できない曖昧な文章がある
2 . 質問を質問で返しているメールがある
3. Re:を除外するコマンドに引っかからないリターンメールがある

この 3 つの問題を改善する方法として、加点フレーズや減点フレーズを増やし、加点フレーズで + カウント、減点フレーズで- カウントさせ、合計したカウント数が 3 カウントあれば取り出すという方法をとっている。

5.5 今後の課題

現状では、質問メールとそれ以外を別々のディレクトリに分けていくところまでできている。しかし、このままでは肝心の FAQ システムとしては使用することができない。今後の課題は、質問メールだけを取り出せる確率をあげるようプログラムを改良し、検出したメールをリスト化し、FAQ システムとして確立させることである。

6　考察とまとめ

6.1　考察

検証の結果わかることは、

- 加点フレーズだけでは質問メールを見分けることができない
- サブジェクトから Re:がつくものを除外しても質問以外のものが入ってしまう (ソースの文字コードが違う為だと考えられる)
- 減点フレーズを増やし過ぎると質問メールまで除外してしまう

以上のことから、質問メールとそれ以外のメールに分別するには条件の量を加減することが必要だと判断できる。そして、あらゆる質問の仕方を考慮し、加点フレーズ、減点フレーズを定めなければならない。

また、質問の時にはテンプレートなど定められた形式や書きはじめフレーズを用意すれば、質問メールの取りこぼしや条件付けが楽になり、管理者側も使いやすい。

6.2　まとめ

この研究テーマに決定したきっかけは、「授業で使っているメーリングリストに便利な機能を実装したい」であった。そして、メールで質問したら自動で返答が送られ、過去に同じ内容の質問があればリストが届くようなシステムを目指し、研究テーマになった。本研究では、目指したシステムとは少し異なり、システム一部、重要な分別プログラムの作成だけを行った。

今回のシステムは、授業とはまた違う方面で使用しているメーリングリストのデータをサンプルデータとして使用した。そのため、実際に授業で使う場合とは異なるパターンの文章が多々あると見受けられた。そのため、授業用のメーリングリストとして実装するためには、さらなるパターンの検証とフレーズの追加・削除が必要だと考える。

謝辞

本研究を進めていく上で、御指導および御協力を頂いた大垣斉講師ならびに fken.a4w のメーリングリストのメンバの皆様に深く感謝の意をここに述べる。

参考文献

[1] シェルスクリプト (Bash), [http://cyberam.dip.jp/linux_command/shellscript/shellscript_main.html#5]
[2] ウィキペディア フリー百科事典, [http://ja.wikipedia.org/wiki/]
[3] 入門 bash,[ISBN4-9009-00788]
[4] 月刊 Linux World,[雑誌 09309-01]
[5] シェルスクリプト ポケットリファレンス bash 編, [ISBN4-7741-2481-8 C3055]
[6] UNIX プログラミング環境,[ISBN4-87148-351-7]
[7] IT 用語辞典 e-Words,[http://e-words.jp/]

付録 A　プログラム

A.1　プログラムソース

```
#! /bin/sh

# 変数設定
TEMP=/tmp/Req-$$-1
HEAD=/tmp/Req-$$-1-Head
BODY=/tmp/Req-$$-1-Body
MAIL=/tmp/Req-$$-1-Mail
TMP2=/tmp/Req-$$-1-tmp
NEET=/tmp/Req-$$-1-neet
POO=/tmp/Req-$$-1-poo
NETA=/tmp/Req-$$-1-neta
SUB=/tmp/Req-$$-1-sub
RE=/tmp/Req-$$-1-re

# 後片付
trap "rm -f /tmp/Req-$$*; exit 0" 0 1 2 3 15

# メールだけを取り出すため、数字のファイル名だけを指定
for i in [0-9]*
do
# 環境設定
LOOP="X-Loop: Req v 0.1"

# TEMP にファイルをいれる
# i にはファイル名が入っている
cat $i > ${TEMP}

# ヘッダーとボディを分離させる
sed -n -e '1,/^$/p' < $TEMP > $HEAD  #ヘッダーだけ取り出し
sed -e '1,/^$/d'    < $TEMP > $BODY  #ボディだけ取り出し

#サブジェクトを取り出す
grep "^Subject: " ${HEAD} | sed -e 's/^Subject: //g'  > $SUB

#サブジェクトに"Re:"と付いているメール以外を取り出す
grep -v "Re:" $SUB > $RE

# 加点するフレーズを取り出す
```

```
lgrep "でしょうか" $BODY  > $TMP2
lgrep "？" $BODY >> $TMP2
lgrep "ご教授" $BODY  >> $TMP2
lgrep "質問" $BODY >> $TMP2
lgrep "わからず" $BODY >> $TMP2
lgrep "解決できません" $BODY >> $TMP2
lgrep "ませんか" $BODY >> $TMP2
lgrep "どなたか" $BODY >> $TMP2
lgrep "お願い" $BODY >> $TMP2
lgrep "のですが" $BODY >> $TMP2
lgrep "のですけれど" $BODY >> $TMP2
lgrep "ありましたら" $BODY >> $TMP2
lgrep "どうやって" $BODY >> $TMP2
lgrep "ご存知" $BODY >> $TMP2
lgrep "ご存じ" $BODY >> $TMP2
lgrep "どのような" $BODY >> $TMP2
lgrep "皆様の" $BODY >> $TMP2
lgrep "みなさま" $BODY >> $TMP2
lgrep "いらっしゃいましたら" $BODY >> $TMP2
lgrep "教え" $BODY >> $TMP2
lgrep "ありますか" $BODY >> $TMP2
lgrep "できましたが" $BODY >> $TMP2
lgrep "方法" $BODY >> $TMP2

# 引用文に含まれる加点フレーズは除外
grep -v "^>" $TMP2 > $NETA

# 減点フレーズ取り出す
lgrep "ということ" $BODY > $MAIL
lgrep "と思います" $BODY >> $MAIL
lgrep "ですよね" $BODY >> $MAIL

# 加点結果取り出し
cat  $NETA | wc -l | awk '{print$1}'  > $NEET

# 減点結果取り出し
cat  $MAIL | wc -l | awk '{print$1}'  > $POO

# 加点結果 ー 減点結果 残りはいくつ?
neet=`cat $NEET`
neet1=`cat $POO`
neet2=`expr $neet - $neet1`
```

```
# 残りポイント表示
# 3 ポイント以上抽出されたものをピックアップ
# 質問メールとそれ以外のものをフォルダ分けする

if [ `echo $neet2` -ge 3 ];
then
# 該当するファイル名を表示
echo $i

# RE に 1 つ以上行が入っていればピックアップ
#if  [ -s $RE ];
then

# 質問メールを question フォルダに振り分ける
cp $i question/

else

# 質問メール以外を other フォルダに振り分ける
cp $i other/

fi
fi
done
```

付録 B　サンプルメール

B.1　方法 1 疑問文よりメール本文抜粋

105030

　iMac ボンダイブルーに gentoo linux をインストールしているところです。
日本語を使用するために以下の url などを参考にしながら作業を進めました。

http://www.deer-n-horse.jp/linux/gentoo/install_14rc2.html
http://bd.tank.jp/gentoo/install8.html
http://homepage1.nifty.com/ssaychi/debian/x643.html

　いまのところ window maker 上で kterm を動かし、日本語表示ができるところ
まで来ています。window maker のメニューが文字化けしていたのですけれど、
フォントをインストールしてから X を立ち上げ直すときちんと日本語表示
されるようになりました。
　ただ、いまのところメニューから呼び出されるユーティリティその他の
画面上で、ボタンメニューが文字化けしており、これを解決できないで
います。
　上の三つの url のうち、最後のものに

（WindowMaker で使うフォント指定の例）
-misc-fixed-medium-r-normal-*-14-*-*-*-*-*-jisx0208.1983-0

（フォント指定を追加するファイルとそのフィールド）
~/GNUstep/Defaults/WindowMaker
WindowTitleFont
MenuTitleFont
MenuTextFont
IconTitleFont
ClipTitleFont
DisplayFont

~/GNUstep/Defaults/WMGLOBAL
SystemFont
BoldSystemFont

とありますけれど、これらの中でどれがボタンメニューに対応した
ものなのでしょうか。ひととおり変更を加えてみたのですけれど、

まだ正常に表示される組み合わせに行き着きません。あるいは
これら以外の設定を変更する必要があるのでしょうか。

105127

HP DL360 について 2.4.31 カーネルをコンパイルしていれました。
問題なくブートできましたが
以下のメッセージが/var/log/messages にでてきてしまいました。
以下を解消する場合、どのようなパラメータの設定になるか
ご教授ください。
よろしくお願いいたします。

B.2　方法 1 質問文よりメール本文抜粋

105208

120GB の同型ハードディスクが 2 台インストールされて
いるマシンで不可解な現象です。
プライマリ IDE にマスター、スレイブで接続していて、
hda、hdb で認識されています。
ハードディスクは工場出荷状態で、
パーティションを切ろうと fdisk でジオメトリを調べたところ、

```
hda: 255 heads, 63 sectors, **** cylinders
hdb: 16 heads, 63 sectors, **** cylinders
```

というように同型なのに異なるジオメトリで認識されてしまいます。
（シリンダー値は忘れましたが全体をヘッダ、
セクタで割った値でした。）
LBA では 255 heads, 63 sectors で表されるのが
一般的ということで hda が正しいのではないかと考えています。

参考までに BIOS 上では auto で以下のように認識しました。
HDD 内に記録されている値だと思われます。

```
master: 16 heads, 255 sectors, **** cylinders
slave: 16 heads, 255 sectors, **** cylinders
```

かなり調べているのですが、原因を追究できず。
皆様のお力をお貸し下さい。

105046

ネット上で、
http://www.kobitosan.net/postfix/ML/arc.2/msg00662.html
を見つけました。
icmp なんてすべて drop しても構わないと思ってました。。。
メールサーバが動いてるサーバで
drop してはいけない icmp の type は何番でしょうか？
ネット上検索してもわからず。。。

よろしくお願いします。