

本稿は [Linux Japan 誌](#) 1999 年 8 月号に掲載された記事に補筆修正したものです。

メール環境を快適に

研究室の机に座るや否や、くると向きを変えて、`mnews -m` で電子メールを読むというのが私の一日の始まりとなっています。講義ノートを `LATEX` で書いているときや、Linux に背を向けて机で仕事をしているときでもメールの着信の音を心待ちにしている自分を発見します。もうメールの無い生活は考えられませんね。皆さんも多かれ少なかれこのメール中毒症におかされているのではないのでしょうか？

いつ来るかわからないメールの着信を文字や画像あるいは音で知らせてくれるツールは `biff` と呼ばれます。ところで、皆さんは、一体どこにメールが着いたら知らせて欲しいのでしょうか？ この問に答えるために、ちょっとメールの移動をおさらいしてみましょう。標準的に `sendmail` を使うと、まずサーバーの `/var/spool/mail/$USER` にメールが届くのが普通です。次に、このメールを取り出す方法には、NFS で `export` して各ホストで直接ファイルを取り出す方法（旧態依然とした方法です）、POP などでサーバーから取ってくるという2つの方法があります。前者は、常時接続してないと実現できません、プロバイダー接続のユーザーは、後者の方法に頼ることになります。その場合には `biff` の出番はあまりないことに気づきます。なぜなら、プロバイダーに接続したときに新着メールをもらうのは、定期便ってことなので `biff` は単に定期便の有無を知らせてくれるだけの役割しか果たさないことになるからです。それでも、いつも新着メールの数が多くて、話題毎にフォルダーを作成し、そこに `procmail` などで振り分けている場合などは、どのフォルダーに新着があったかを表示してくれると嬉しいでしょう。そういう機能を持つ `biff` も需要があります。

さて、最初の問に対する答ですが、一般的あるいは基本的には、(i) まずサーバー上の `/var/spool/mail` への着信を知らせるべきものです。しかし、サーバーから取り寄せてからじっくり読むという場合には、(ii) 各自のホームにある `mailboxes` の状況を知らせて欲しいとなるでしょう。例によって、無理矢理な分類ですが2部構成くらいにしないとどうにも体裁が整わないので、(i) を単一ボックス型 (ii) をマルチボックス対応型と称して、話しを進めます。

ちょっとその前に、昔からある `biff` について簡単に説明しておきます。

biff

端末に突然割り込んできて、次のようにメール着信を告げる `biff` に驚いた経験は、どなたもお持ちのことと思います。

```
aya:~/LJP/7th$
New mail for matuda@aya has arrived:
----
From: Namio MATUDA <matuda>
Subject: test for comsat
Biff!
----
```

`vi` で編集している時なんかですと迷惑この上ないので、`biff n` として、禁止しているユーザーも多いでしょう。ついでに、このような端末への書き込みを行うツールには、他に `write` や `talk` がありますが、こちらは `mesg n` で禁止することになっています。

以上のような理由で、X クライアント（端末とは別窓ですから）が好まれるわけです。ところで、`biff` には X クライアントにはない大きな特徴があります。実は監視用のデーモン `comsat` が動いていて着信時には、即座に知らせてくれる手筈となっています。X クライアントの場合には、クライアント自身が一定間隔でメールボックスの変化を調べに行きます（メール着信を判断するアルゴリズムも多種多様です）。したがって、間隔を1秒などと設定してしまうと、ほとんどは無駄な通信を行う結果となります。ネットワーク資源を無駄にしないという点では `biff` は大変優れた存在である点を強調しておきます。

単一ボックス型

`biff` もローカルに見える `/var/spool/mail` の状況を知らせてくれるので、単一ボックスタイプと言えましょう。X では `xbiff` や `xpbiff` などが専用のクライアントとして有名ですが、他にも `rclock` や `mclock` 等の時計、ちょっと変わったところでは `oneko ver2β`（コタツ猫）も `biff` 機能があります。

xbiff

名前もずばり `xbiff` で、まさに X 版 `biff` です。ビットマップ画像の指定ができることと `shape` に対応して背景を透かすことができます。デフォルトの画像は `flagdown` と `flagup` ですが、`~/Xdefaults` に

```
Xbiff*emptyPixmap: mailempty
Xbiff*emptyPixmapMask: mailemptymsk
Xbiff*fullPixmap: mailfull
Xbiff*fullPixmapMask: mailfullmsk
```

と記述すると、図の右二つの絵柄に変更することができます。./usr/include/X11/bitmaps に付属している該当ビットマップをコピーするのを忘れないください。メールをスプールから取り出すか、あるいはボタンクリックで、未着の状態になります。監視の時間間隔は -update <sec> で、監視するファイルは -file <file> で設定します。また、世の中いろいろ考える人がいるもので、着信時のみ画面に現れ未着時には隠れてしまうオプション -hide が使えるバージョンもあります。



図 1 xbiff:デフォルトの絵柄(左2つ)とリソース変更後の絵柄(右2つ)

xpbiff

着信を瞬間芸(アニメーションで)通知してくれる愉快的なツールで、PJE では標準配付ですね。作者は首藤一彦(shutoh@lm.emi.yamaha.co.jp)氏です。xbiff がない機能の一つに、メールのヘッダーの一部の表示があります(図右)。もっとも、数が多いと却ってうるさいかもしれません。監視時間間隔の設定オプションは -polling_time <sec> です。改良版では -sound <sound.au> オプションにより、au ファイルを再生するものもあります。未着時には隠れているようにするオプションはないようです。



図 2 瞬間芸付き biff: xpbiff

xbiff

メールのヘッダーの一部をリスト(文字)にして表示して知らせます。デフォルトでは、リストを MH の scan を使って作成します。すなわち

```
scan -file /var/spool/mail/$USER
```

を実行した結果を表示することになります。リソース scanCommand を指定してリスト作成の方法を変更できます。例えば /bin/mail を使って

```
scan echo 'x' | mail | grep '^.[UN]'
```

などとするのも一つの方法だそうです。フォーム、味わい深いですね。作者の Ed Santiago (esm@auspex.com)氏は『届いたメールをいつも見たいとは思わない、xbiffはどんなメールか教えてくれないので xlbiff を作成した』という趣旨を README で述べています。

```
1 06/03 To:matuda test for xlbiff <<Hello!
2 06/03 To:matuda test2 <<Hi!
```

図 3 xlbiff:普段は隠れているが、着信時にメールヘッダーの一部を表示

xpbiff

xbiff も xpbiff も表示する画像は2値ビットマップでちょっと寂しいです。当然、XPM を表示するものがある筈です。実際、マルチボックス対応型のもの比較的最近に作成されたので、カラー対応です。単一ボックス型をわざわざ探すこともないかもしれませんが、そこはそれ収集家の業で、xpbiff をみつけました。作者は Fredrik Vhrstrk (d92-foh@nada.kth.se)氏で、1995年12月9日の日付けがみえます。標準の画像が付属していないので、youbin に付属しているものの背景を透明にして使ってみました。

```
xpbiff youbinEmpty.xpm youbinFull.xpm 15
```

のように起動します。最後の 15 が監視時間間隔指定です。



図 4 xpbiff:カラー画像(XPM)を表示

マルチボックス対応型

メールを一日に数百通ももらう人は、全てを読むわけには行かないので、読むかどうかの選択ができるようにいろいろと工夫していることでしょう。私の場合には、

inc で全てのメールを取り込み mnews の一覧で判断しています。最初から、話題毎にフォルダー（ディレクトリ）やボックス（ファイル）を作成しておき、procmail など新着メールを自動的に振りわけている人もいます。その場合には、それぞれのフォルダーやボックスを監視して知らせてくれる biff が必要となります。

xbuffy

mbox, MH, maildir そして nntp のフォルダーに対応していて便利ですが、表示はそっけないですね。設定ファイルを使うことができるので、~/xbuffy とでもして、

```
xbuff -boxfile ~/.xbuffy
```

と起動しましょう。設定ファイルには監視するメールボックスと表示用の名前、監視時間間隔や着信時に起動する外部コマンドなどを、個別に記述します。例えば

```
box /var/spool/mail/matuda
title spool
polltime 15

box /home/matuda/Mail/inbox
command kterm -km euc -e mnews -m &
```

とすれば、メールスプールの監視と、MH の inbox フォルダーの監視ができて、タイトルと未読メールの数が表示されます。また、着信時にはヘッダーの一部を表示する機能もあります（図右）。作者は Bill Pemberton (wfp5p@virginia.edu) 氏です。Brandon Long (blong@fiction.net) 氏が改作した版をネットワークで入手しました。



図 5 xbuffy: 様々なメールボックスやフォルダーに対応した万能 biff

glbiff

OpenGL 互換の Mesa ライブラリを用いた、3G アニメーションが楽しい biff です。メール着信時には赤い旗が上がります、左ボタンでクリックすると郵便受けが回転し、蓋が開きます。中にはメールの数分の手紙が並んでいたりして、面白い奴ですね。デフォルトは大きいの

で -geometry 56x56 で Wharf ボタンに押し込めます。設定ファイル ~/.glbiffrc には監視するメールボックス（フォルダーには対応してません）の名前や、監視時間間隔等を記述します。時刻に応じて空の色を変える設定もできるらしいです。glbiff ではメールボックスファイルの大きさとアクセス日時・更新日時の前後関係を元に新着を判断していますが、/usr/sbin/mail.local による配送では更新日時とアクセス日時が等しくなるらしく、着信のフラグが立ちません。この不都合を回避するには、mail.cc の 103 行目にある箇所を以下のように変更してみてください。

```
unreadmail = (st.st_mtime > st.st_atime ... を
unreadmail = (st.st_mtime >= st.st_atime ... に
```

作者は、Maciej Kalisiak (mac@dgp.utoronto.ca) 氏です。

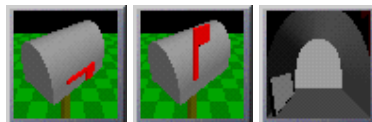


図 6 3DCG が楽しい glbiff

xpnbiff

首藤氏の名作、瞬間芸付き xpnbiff を元に、河田幸博 (kawada@den.fujifilm.co.jp) 氏が NNTP news 対応にした版です。MH のフォルダーに対応しているので紹介することにしました。

```
xpnbiff -mh
```

と起動すると、~/Mail 以下の全てのフォルダーを監視します。ただし、MH の設定ファイル ~/.mh_profile に

```
Unseen-Sequence: unseen
```

を設定します。これは xpnbiff が未読の情報を、各フォルダーにある .mh_sequence から読み取っているためです。xpbiff と画像が微妙に違ってきます。

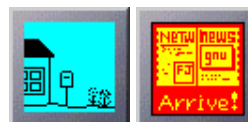


図 7 3DCG が楽しい glbiff

asmail

私が今使っている AfterStep 御用達の biff です。付属の Xpm 画像が綺麗なのと、着信時に一連の Xpm 画像を順に表示するアニメーションが気に入ったので使っています。設定は ~/.asmailrc に記します。最近の作なので、機能が豊富なのですが、メール数および未読メール数の表示は案外有用です（図右端）。



図 8 asmail：着信時にアニメーションを表示し続けます。付属の“e の字の回転”アニメーションもなかなかのものです。

メールのチェック

Biff は簡単にいうと、新着メールの有無を調べてそれを音や文字や画像で知らせてくれるツールです。後半部分がユーザーの目にふれる部分で、いろいろと凝ってしまう部分ですが、ここでは、前半、すなわちメールの有無をどう判断しているかを調べてみましょう。ソースを見て、もっとも一般的なものは、stat() 関数でメールボックスファイルの状態を取得し、大きさと更新日時やアクセス日時を調べる方法です。List 1 に示す、stat() 関数を使った簡単なチェックプログラム mcheck.c を試してみてください。

List1 mcheck.c

```
#include <stdio.h>
#include <sys/stat.h>
#include <stdlib.h>
#include <unistd.h>

main(int argc, char **argv)
{
    char *filename;
    char *username;
    struct stat buf;
    off_t size;

    /* 環境変数 USER の取得 */
    username=getenv("USER");

    /* メールボックスのファイル名構成 */
    sprintf(filename, "/var/spool/mail/%s", username);

    /* メールボックスファイルの状態取得 */
```

```
    stat(filename, &buf);

    if (size = buf.st_size > 0){
        printf("メールがあるようです.\n");
    }
    if (size > 0 && buf.st_mtime >= buf.st_atime){
        printf("未読メールがあるようです.\n");
    }
}
```

一回だけのチェックですが、メールボックスファイル /var/spool/mail/\$USER の状態を調べて、メールの有無と未読の有無を判断します。実行例を示します。< > で括られた部分では mail コマンドを使って、指示内容を実現してください。

```
< 自分宛にメールを送る。 >
\$ mcheck
メールがあるようです。
未読メールがあるようです。

< mail コマンドでメールを読み、x で抜ける。 >
\$ mcheck
メールがあるようです。

< mail コマンドでメールを読み、q で抜ける。 >
\$ mcheck
<-- 何も表示されない
```

このチェックルーチンはかなり安易ですが、一応、メールの有無やある程度の未読の判断ができます。あとは、音や画像で知らせる部分を工夫すれば自前の Biff の出来上がりです。xscreensaver のところで紹介したルート画面へのアニメーションを起動するようにすると面白いでしょうね。

心残り

qmail に対応した xqbiff や、comsat デーモンに替わってネットワーク越しにメール着信を通知する youbin プロトコルなども紹介すべきとは思ったのですが、筆者の勉強不足で書ききれませんでした。メールに関連するツールはバラエティにとんでおり、近い将来もう一度まとめて紹介したいと思っています。Netscape オンラインのユーザーにも古き良き UNIX のツールの奥深さを伝えることができますでしょうか？

次回予告

月刊化したので、遅筆（単なる怠け者？）の筆者は、~~切~~切がますます恐ろしくなりそうです。『通知』ということで続けて、ユーザー同士でメッセージ交換を行うツールの紹介をします。