データベース構築論2

18章「メールの一覧表示と検索」

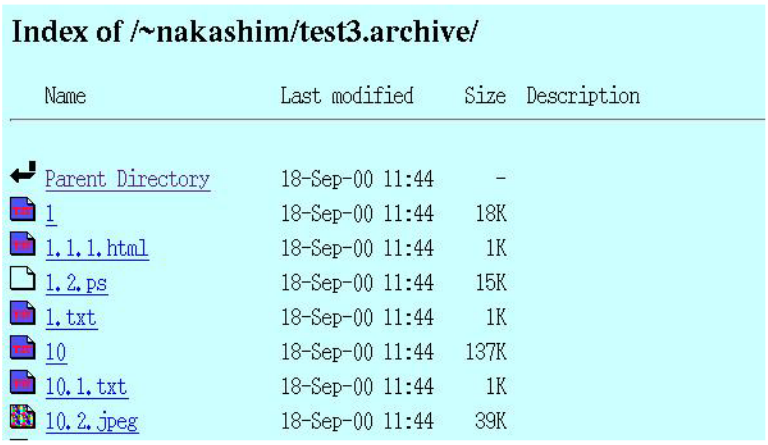
中島康彦

§18. 1 単純な一覧表示

前回の続き. このままでも一覧は表示できる

- WEBブラウザによるメール一覧表示

  http://i.econ.kyoto-u.ac.jp/~xxx/ml00-xxx.archive



Index of /~nakashim/test3.archive/

Name	Last modified	Size	Description
Parent Directory	18-Sep-00 11:44	-	
1	18-Sep-00 11:44	18K	
1.1.1.html	18-Sep-00 11:44	1K	
1.2.ps	18-Sep-00 11:44	15K	
1.txt	18-Sep-00 11:44	1K	
10	18-Sep-00 11:44	137K	
10.1.txt	18-Sep-00 11:44	1K	
10.2.jpeg	18-Sep-00 11:44	39K	

- ただし, ファイルは名前順.
- 発信日/発信者/題名はファイルを開くまで不明.
- リアルタイムに更新されない.

§18. 2 プログラムによる一覧表示

CGIを使った一覧表示

- 必要なファイルの作成

```
% cd ~/Html/ml00-xxx.archive/
% cp /usr/local/etc/httpd/htdocs/test3.archive/index.cgi .
% cp /usr/local/etc/httpd/htdocs/test3.archive/mailindex .
% cp /usr/local/etc/httpd/htdocs/test3.archive/mailshow.cgi .
% ls
index.cgi mailindex mailshow.cgi
```

§18. 2 プログラムによる一覧表示(続き)

CGIができたら再表示してみる

- WEBブラウザによるメール一覧表示(URLは同じ)

Thu, 20 Apr 2000 [test3.25] FAX添付	
Thu, 20 Apr 2000 [test3.24] ビデオキャプチャ	24.2.jpeg
Thu, 20 Apr 2000 [test3.23] もう一度	23.2.html
Thu, 20 Apr 2000 [test3.22] さらに複雑な送信	22.2.html
Thu, 20 Apr 2000 [test3.21] HTML送信	21.2.jpeg
Thu, 20 Apr 2000 [test3.20] HTML添付	20.2.octet-stream
Thu, 20 Apr 2000 [test3.19] スキャナ (JPG)	19.2.jpeg
Thu, 20 Apr 2000 [test3.18] スキャナー入力	18.2.bmp
Thu, 20 Apr 2000 [test3.17] 動画の場合	17.2.avi
Thu, 20 Apr 2000 [test3.16] mhmail直接送信	
Thu, 20 Apr 2000 None	
Thu, 20 Apr 2000 [test3.14] をを順番が入れ替わっている！	
Thu, 20 Apr 2000 [test3.13] UNIXからの添付ファイルテスト	13.2.txt
Thu, 20 Apr 2000 [test3.12] 長い長い長い長い長い長い長い	

- ファイルは日付の新しい順.
- 添付ファイルは横に表示.
- 発信日/題名が表示される. (発信者の表示も可)
- リアルタイムに更新される.

§18. 2 プログラムによる一覧表示(続き)

- ファイルを開くと，必要な情報のみが表示される.

長いタイトルに対する耐性テスト

中島康彦 e-mail: nakashim@econ.kyoto-u.ac.jp
tel: 075-753-3424 fax: 075-753-3492

- ヘッダの表示が抑止されている.
- 添付ファイルは別ファイル.

§18. 3 index.cgi

```
#!/bin/sh ★UNIXのシェル(sh)を起動するおまじない

cat <<! ★次の!までをそのまま表示
Content-Type: text/html

<html>
<meta http-equiv="Refresh" content="600"> ★600秒毎に再実行するおまじない
<head><title>ファイル一覧</title></head>
<body BGCOLOR="c0ffc0" TEXT="000000" LINK="0000ff" VLINK="800080" ALINK="ff00
00">
<pre>
! ★ここまで

for FILE in `ls -t [0-9]*` ★数字で始まる名前のファイルを日付の新しい順に選択
do
    ./mailindex $FILE ★★★ファイル毎にmailindexを実行
done

cat <<! ★次の!までをそのまま表示
</pre>
<hr> ★線を引く
</body>
</html>
! ★ここまで
```

§18. 4 mailindex

```
#!/usr/local/bin/perl ★Perl(Practical Extraction and Report Language)を起動するおまじない

$Filename = $ARGV[0]; ★index.cgi中の記述 "./mailindex $FILE" の$FILEを取り出す

if ($Filename !~ /^\d+$/) {
    exit 0;
}

open(FILE0, "|/usr/local/bin/nkf -m -e>/tmp/mailindex0-$$");
while (<>) {
    print FILE0;
}
close FILE0;
```

★以上で，$FILEのMIMEデコードおよびEUC漢字コードへの変換が完了．変換結果は，/tmp/mailindex0-$$に入っている．$$は一意に付けられる番号．

§18. 4 mailindex(続き)

```
open(FILE0, "</tmp/mailindex0-$$");
while (<FILE0>) { ★変換結果を1行ずつ読む
    chop;
    if (/^Date:\s*/) { ★空行以前に，行頭がDate:であれば変数$Dateに代入
        ($Date = $') =~ s/</(/;
    }
    elsif (/^From:\s*/) { ★空行以前に，行頭がFrom:であれば変数$Fromに代入
        ($From = $') =~ s/</(/;
    }
    elsif (/^Return-Path:\s*/) { ★空行以前に，行頭がReturn-Path:であれば〃
        ($ReturnPath = $') =~ s/</(/;
    }
    elsif (/^Subject:\s*/) { ★空行以前に，行頭がSubject:であれば〃
        ($Subject = $') =~ s/</(/;
    }
    elsif (/^\s*$/) { ★空行を検出したらヘッダ解析を終了
        $Date       = "Unknown"       unless ($Date);
        $From       = "Not Specified" unless ($From);
        $ReturnPath = "Not Specified" unless ($ReturnPath);
        $Subject    = "None"          unless ($Subject);
        last; ★whileループを抜ける
    }
} ★以上でファイルの読み込みを終了
```

§18. 4 mailindex(続き)

```
$firstfile = 1;

foreach $addfile (<$Filename.*>) {
    if ($firstfile == 1) {
        printf "<a href=%s target=frame3>%-16.16s %-40.40s </a>", $addfile,
$Date, $Subject;
        $firstfile = 0;
    }
    else {
        printf " <a href=%s target=frame3>%s</a>", $addfile, $addfile;
    }
} ★最初の[0-9]*.xxxがmailshow.cgiを実行するようなHTMLを生成

if ($firstfile == 1) {
    printf "<a href=\"mailshow.cgi?%s\" target=frame3>%-16.16s %-40.40s
</a>", $Filename, $Date, $Subject;
} ★1つしかファイルがない場合，[0-9]*がmailshow.cgiを実行するようなHTMLを生成

printf "\n";

close FILE0;
unlink("/tmp/mailindex0-$$");
exit 0; ★作業用ファイルを削除して終了
```

§18. 5 mailshow.cgi

```
#!/usr/local/bin/perl ★Perl(Practical Extraction and Report Language)を起動す
るおまじない

$Filename = $ARGV[0]; ★mailindex中の記述 "./mailshow.cgi?%s" の%sを取り出す

print "Content-Type: text/plain\n\n";

open(FILE0, "|/usr/local/bin/nkf -m -e>/tmp/mailindex0-$$");
while (<>) {
    print FILE0;
}
close FILE0;
```

★以上で，%sのMIMEデコードおよびEUC漢字コードへの変換が完了．変換結果は，
/tmp/mailindex0-$$に入っている．$$は一意に付けられる番号．

§18. 5 mailshow.cgi(続き)

```
open(FILE0, "</tmp/mailindex0-$$");
$parse_header = 1; ★空行を検出するまで1, 検出したら0
while (<FILE0>) { ★変換結果を1行ずつ読む
    chop;
    if ($parse_header == 0) { ★本文はそのまま表示
        print;
        print "\n";
    }
    elsif (/^Date:\s*/) { ★空行以前に, 行頭がDate:であれば変数$Dateに代入
        ($Date = $') =~ s/</(/;
    }
    elsif (/^From:\s*/) { ★空行以前に, 行頭がFrom:であれば変数$Fromに代入
        ($From = $') =~ s/</(/;
    }
    elsif (/^Return-Path:\s*/) { ★空行以前に, 行頭がReturn-Path:であれば"
        ($ReturnPath = $') =~ s/</(/;
    }
    elsif (/^Subject:\s*/) { ★空行以前に, 行頭がSubject:であれば"
        ($Subject = $') =~ s/</(/;
    }
```

§18. 5 mailshow.cgi(続き)

```
    elsif (/^\s*$/) { ★最初の空行の場合, ヘッダ部分を整形して表示
        $Date       = "Unknown"       unless ($Date);
        $From       = "Not Specified" unless ($From);
        $ReturnPath = "Not Specified" unless ($ReturnPath);
        $Subject    = "None"          unless ($Subject);
        ★各行が存在しない時の表示
        printf "Date:       %s\n", $Date;
        printf "From:       %s\n", $From;
        printf "ReturnPath: %s\n", $ReturnPath;
        printf "Subject:    %s\n", $Subject;
        print "================================================\n";
        $parse_header = 0;
        ★ヘッダ部分の表示
    }
} ★以上でファイルの読み込みを終了

close FILE0;
unlink("/tmp/mailindex0-$$");
exit 0; ★作業用ファイルを削除して終了
```

§18. 6 メール検索機能

FORMとCGIを使った検索

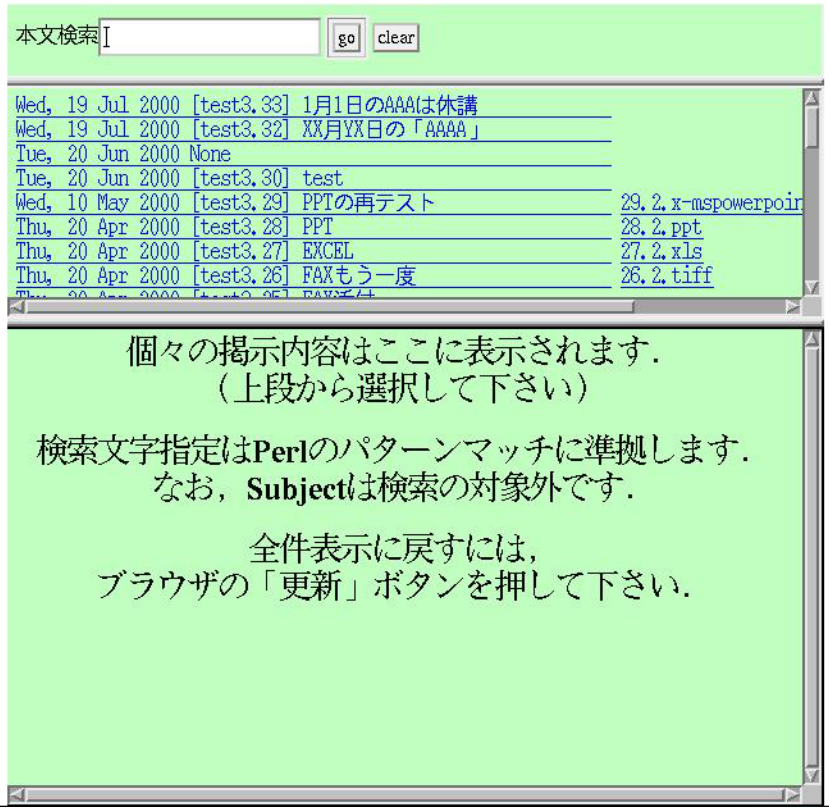
▶ 必要なファイルの作成

```
% cd ~/Html/ml00-xxx.archive/
% cp /usr/local/etc/httpd/htdocs/test3.archive/howto.html .
% cp /usr/local/etc/httpd/htdocs/test3.archive/index.shtml .
% cp /usr/local/etc/httpd/htdocs/test3.archive/jcode.pl .
% cp /usr/local/etc/httpd/htdocs/test3.archive/mailsearch .
% cp /usr/local/etc/httpd/htdocs/test3.archive/search.cgi .
% cp /usr/local/etc/httpd/htdocs/test3.archive/search.shtml .
% ls
howto.html   index.shtml   jcode.pl
mailsearch   search.cgi    search.shtml
```

§18. 6 メール検索機能(続き)

各自のホームページを表示してみる

http://i.econ.kyoto-u.ac.jp/~xxx/ml00-xxx.archive



§18. 7 index.shtml

今までのindex.cgiよりも優先順位が高い

```
<html>
<head><title>ファイル一覧</title></head>

<frameset rows=10%,30%,60%>
  <frame name="frame1" src="search.shtml" scrolling="no">
  <frame name="frame2" src="index.cgi"    scrolling="yes">
  <frame name="frame3" src="howto.html"   scrolling="yes">
</frameset>

</html>
```

§18. 8 howto.html

使用方法の表示

```
<html>
<head><title>内容表示</title></head>
<body BGCOLOR="c0ffc0" TEXT="000000" LINK="0000ff" VLINK="800080" ALI
NK="ff0000">

<center>
<h1>
個々の掲示内容はここに表示されます．<br>
(上段から選択して下さい)<br>
</h1>
<h1>
検索文字指定はPerlのパターンマッチに準拠します．<br>
</h1>
<h1>
全件表示に戻すには，<br>
ブラウザの「更新」ボタンを押して下さい．<br>
</h1>
</center>

</html>
```

§18. 9 search.shtml

検索FORM

- 入力文字列がsearch.cgiに渡される.

```
        <html>
        <head><title>ファイル検索</title></head>
        <body BGCOLOR="c0ffc0" TEXT="000000" LINK="0000ff" VLINK="800080" ALI
NK="ff0000">

        <form method="get" action="search.cgi?." target=frame2>
                本文検索<input name="string" size=20 maxlength=20>
                <input type="submit" value="go">
                <input type="reset"  value="clear">
        </form>

        </html>
```

§18. 10 検索のための正規表現

正規表現(Regular Expression)

- メタ文字

^	行の最初に一致
$	行の終わりに一致
.	任意の1文字に一致
*	0回以上
+	1回以上
?	0回または1回
[]	文字クラス
()	文字グループ
|	選択
\	次のメタ文字を通常文字と解釈

- 上記以外の通常文字

  その文字自身に一致

§18. 10 検索のための正規表現(続き)

経済	"経済"に一致
^経済	行頭の"経済"
経済$	行末の"経済"
^経済$	一行が"経済"のみ
経済.1	"経済01", "経済A1", "経済J1"
経済.*1	"経済01", "経済001", "経済0001"
経済[2-9]1	"経済21", "経済31"
経済[2-9A-Z]1	"経済21", "経済31", "経済A1"
経済[^2-9A-Z]1	"経済01", "経済11", "経済a1"
(経済|情報)	"経済"または"情報"

§18. 11 データの授受

search.cgiに渡されるデータ

- 検索 "econ"の場合
  string=econ

- 検索 "^econ$"の場合
  string=%5Econ%24

- 検索 "経済"の場合
  string=%B7%D0%BA%D1

- 検索 "^経済$"の場合
  string=%5E%B7%D0%BA%D1%24

§18. 12 search.cgi

```
#!/bin/sh ★UNIXのシェル(sh)を起動するおまじない

cat <<! ★次の!までをそのまま表示
Content-Type: text/html

<html>
<head><title>検索結果一覧</title></head>
<body BGCOLOR="c0ffc0" TEXT="000000" LINK="0000ff" VLINK="800080" ALINK="ff00
00">
<pre>
! ★ここまで

echo "検索結果" ★タイトルの表示
for FILE in `ls -t [0-9]*` ★数字で始まる名前のファイルを日付の新しい順に選択
do
    ./mailsearch $QUERY_STRING $FILE ★★★ファイル毎にmailsearchを実行
done

cat <<! ★次の!までをそのまま表示
</pre>
<hr> ★線を引く
</body>
</html>
! ★ここまで
```

---

§18. 12 mailsearch

```
#!/usr/local/bin/perl ★Perlを起動するおまじない

require 'jcode.pl';
sub init_form {
    local($query, @assocarray, $assoc, $property, $value, $charcode, $method)
;
    $charcode = $_[0];
    $method = $ENV{'REQUEST_METHOD'};
    $method =~ tr/A-Z/a-z/;
    if ($method eq 'post') {
        read(STDIN, $query, $ENV{'CONTENT_LENGTH'});
    }
    else {
        $query = $ENV{'QUERY_STRING'};
    }
    @assocarray = split(/&/, $query);
    foreach $assoc (@assocarray) {
        ($property, $value) = split(/=/, $assoc);
        $value =~ tr/+/ /;
        $value =~ s/%([A-Fa-f0-9][A-Fa-f0-9])/pack("C", hex($1))/eg;
        &jcode'convert(*value, $charcode);
        $form{$property} = $value;
    }
} ★以上は，&init_form('euc');を使うためのライブラリとサブルーチン
```

§18. 12 mailsearch(続き)

```
$QUERY_STRING = $ARGV[0]; ★search.cgi中の記述 "./mailsearch $QUERY_STRING $FILE" の$QUERY_STRINGを取り出す
$Filename = $ARGV[1]; ★同様に$FILEを取り出す
shift; ★<>から$QUERY_STRINGを除いて$FILEのみにする

open(FILE0, "|/usr/local/bin/nkf -m -e>/tmp/mailindex0-$$");
while (<>) {
    print FILE0;
}
close FILE0;
```

★以上で，$FILEのMIMEデコードおよびEUC漢字コードへの変換が完了．変換結果は，/tmp/mailindex0-$$に入っている．$$は一意に付けられる番号．

§18. 12 mailsearch(続き)

```
&init_form('euc');
$String = $form{'string'};
★検索文字列をEUC漢字コードへ変換し，結果を$Stringに格納する．
$found = 0; ★検索文字列を検出するまで0，検出したら1

open(FILE0, "</tmp/mailindex0-$$");
$parse_header = 1; ★空行を検出するまで1，検出したら0
while (<FILE0>) { ★変換結果を1行ずつ読む
    chop;
    if (/$String/i) { ★各行に検索文字列($String)があるか(大小文字区別なし)
        $found = 1;
        last; ★あれば，$foundに1を代入して終了
    }
    if ($parse_header == 0) {
    }
    elsif (/^Date:\s*/) {
        ($Date = $') =~ s/</(/;
    }
    elsif (/^From:\s*/) {
        ($From = $') =~ s/</(/;
    }
    elsif (/^Return-Path:\s*/) {
        ($ReturnPath = $') =~ s/</(/;
    }
```

§18. 12 mailsearch(続き)

```
    elsif (/^Subject:\s*/) {
        ($Subject = $') =~ s/<//;
    }
    elsif (/^\s*$/) {
        $Date       = "Unknown"       unless ($Date);
        $From       = "Not Specified" unless ($From);
        $ReturnPath = "Not Specified" unless ($ReturnPath);
        $Subject    = "None"          unless ($Subject);
        $parse_header = 0;
    }
}
```
★以上でファイルの読み込みを終了

§18. 12 mailsearch(続き)

```
if ($found == 1) {
    $firstfile = 1;
    foreach $addfile (<$Filename.*>) {
        if ($firstfile == 1) {
            printf "<a href=%s target=frame3>%-16.16s %-40.40s </a>",
$addfile, $Date, $Subject;
            $firstfile = 0;
        }
        else {
            printf " <a href=%s target=frame3>%s</a>", $addfile, $addfile;
        }
    }
    if ($firstfile == 1) {
        printf "<a href=\"mailshow.cgi?%s\" target=frame3>%-16.16s %-40.40s
</a>", $Filename, $Date, $Subject;
    }
    printf "\n";
}
```
★ファイルが検索文字列($String)を含む場合，クリックするとmailshow.cgiを実行するようなHTMLを生成(mailindexと同じ)

```
close FILE0;
unlink("/tmp/mailindex0-$$");
exit 0;
```
★作業用ファイルを削除して終了

§18. 13 検索ホームページとのリンク

http://i.econ.kyoto-u.ac.jp/test3.archive/

- メーリングリストの検索

```
<form action="ml00-xxx.archive/search.cgi?.">
<input name="string">
```

- Gooの検索

```
<form action="http://www.goo.ne.jp/default.asp?.">
<input name="MT">
```

- Yahooの検索

```
<form action="http://search.yahoo.co.jp/bin/search?.">
<input name="p">
```

今日はここまで