アクセスログを用いたSocial Bookmarkの構築

03H020 小栗 一也

目次

1 はじめに 1

2 システム 2
2.1 システムの目的 2
2.2 システムの概要 2
2.3 システムに使用するもの 3
2.4 システムに求めるもの 3

3 システムの構築 4
3.1 プログラム実行手順 4
3.2 実行結果 4

4 まとめ 9
4.1 現状 9
4.2 今後の課題 9

付録 A プログラム 11
A.1 プログラムのソース 11

1　はじめに

現在、繰り返して Web page を閲覧するにあたって、Bookmark という機能がある。しかし、Bookmark を個人で登録し、個人で管理し、Bookmark を共有をすることはできない。共有を交え、新たな役立つ知識を発見することが出来れば、モチベーションが上り、研究促進に繋がると考える。

そして、それを実現する方法として現存する、Social Bookmark というサービスがある。ネットワークが繋がっている環境であれば、自分のパソコンで開くブラウザでなくても、自分が Bookmark として登録した URI[*1] を閲覧することが出来、登録した URI が他のユーザーはどれだけ登録しているか、また、その登録しているユーザーの Bookmark が見れ共通の嗜好があれば、コミュニケーションがとれる。

しかし、不特定多数の人が登録するため、新しいものや流行のものを見つけ出すのには適するであろうが、同じ研究を進めている小規模なネットワーク内で利用するにあたって、自分が必要とする情報、役立つ知識を見つけ出すことは難しい。

そこで、本研究では同じ部署や研究室等の小規模ネットワーク内で研究促進に繋がり、如何に活用出来るかを目的とした Social Bookmark を構築する。squid [*2] の access.log を参照し、誰がどのような URI を閲覧したのかを ID 別に表示し、その URI が今日どのくらいアクセスされたか、また、誰が Bookmark として登録しているかを表示する。そして、アクセス件数が非常に多い URI があるのにも関わらず登録されていなければ、自動でその URI を Bookmark し、構築する Web page の top にアクセス件数と共に掲載する。

こういった見栄えと、カウンタをつける事により、研究の内容、また、流行のものが浮き彫りになり、研究の促進、同じ部署で研究に携わるにあたってのよいコミュニケーションになるであろうと推測する。

[*1] URI(Uniform Resource Identifiers)... 主に Web ブラウザを用いてページを見る際に必要な URL、つまり情報の資源の場所を示す記述方式と、それとは逆の概念を持つ、ファイルやデータそのものの名前を指し示す URN の両方持ち合わせたもの。

[*2] Squid... Proxy cache server である。システム管理者管轄内の user が、そのシステムの Proxy を通して、Web ブラウザで URI にアクセスしたとき、誰が、いつ、どこを、どのようにアクセスしたかを log として残す。

2　システム

2.1　システムの目的

現在 Web 上にある、既存の Social Bookmark Site というサービスシステムに似た Bookmark システムを構築し、機能を付加し、小規模ネットワーク内で如何に実用的に使えるかを目指す。

つまり、同志が集まるところのネットワーク内で、研究に必要な、かつ、特定された分野での情報の発見しやすさを目的とする。

2.2　システムの概要

オブジェクト指向型言語 Ruby を用いて、プログラムを記述し、Proxy Server である Squid の アクセスログを参照して、その研究室内の個人 user 名、URI を抽出する。その後、抽出された URI の中で同じ URI があれば数え上げ、個人 user 名毎がどの URI に何回アクセスしたのかを CGI プログラムを用いて Web 上に掲載する。また、個人 user 毎の Bookmark page を設け、そこにもアクセスカウンタをつける。イメージを図 1 に示す。

これによって同志がどのようなことについて研究しているかや、どんな嗜好であるかなどを分析できる。 このシステム監視内で、どの user も Bookmark しておらず、かつ、アクセス件数が高いものをトップページに自動で Bookmark する。これによって必要としているであろう情報があるページの URI を保持し、閲覧が出来る。

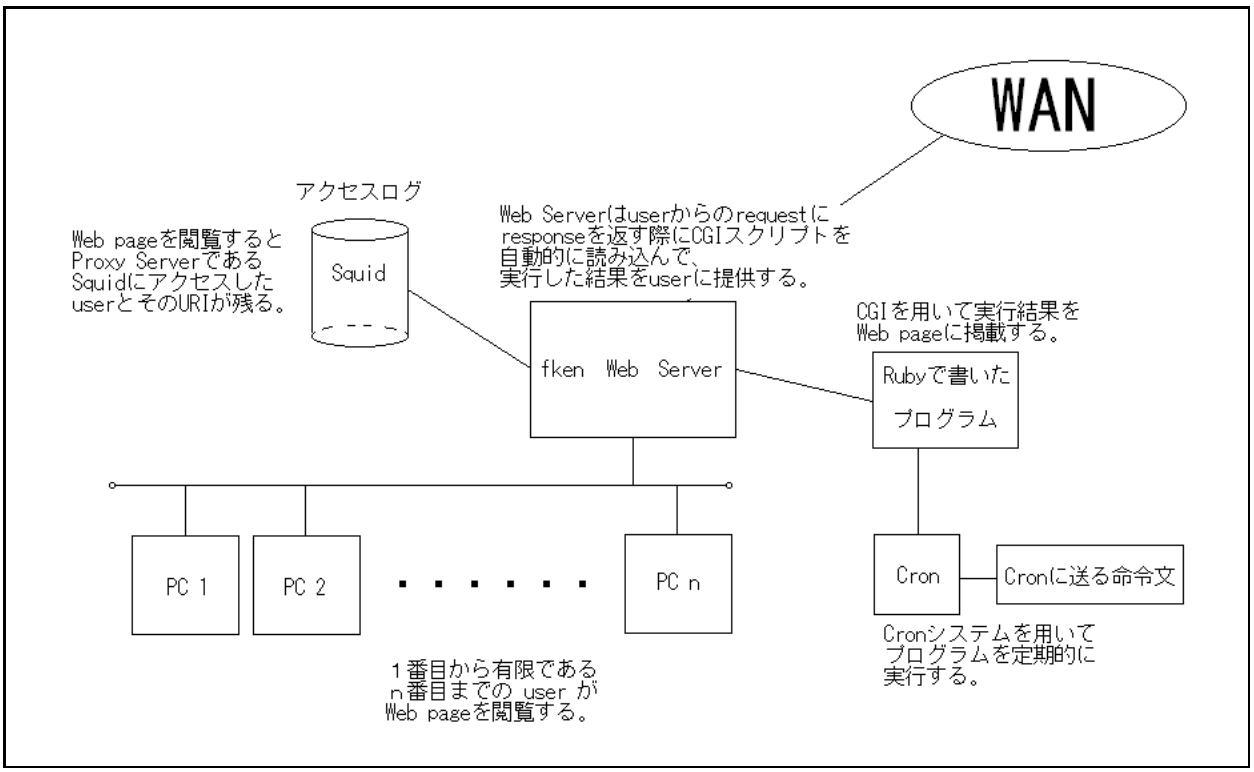

図 1　概略図

2.3 システムに使用するもの

2.3.1 Ruby

まつもとゆきひろ氏が開発したスクリプト言語。手軽なオブジェクト指向プログラミングを実現するための種々の機能を持つオブジェクト指向スクリプト言語である。

本研究で Ruby を使用する理由として以下のことが挙げられる。

- 知らなかった。
- やってみたいと思った。
- 大垣講師、そして赤松氏お薦めだった。

2.3.2 squid のアクセスログ

上記でも述べたように、本研究室では Squid を使用することで、Proxy Server を通して Webpage を閲覧すると、Squid のログファイルにログが記述されるようになっている。

2.3.3 CGI(Common Gateway Interface) プログラム

Web ブラウザからの要求に応じて Web Server がプログラムを起動するための仕組みである。

不特定多数の user からのアクセスに対話的に、動的に Webpage を表示できるようになる。

2.3.4 cron システム

指定した時刻に指定したプログラムを実行させるシステム。

2.4 システムに求めるもの

他の Social Bookmark サービスにはないものを新機能として付加する。

- アクセス件数表示。複数の user がアクセスした URI を数え上げ、link されている横にその日にアクセスされた件数を表示させる。
- 個人登録 URI のページには、昨日何件その URI が同じ研究室内でアクセスされたかを表示する。
- アクセスログから URI だけを月曜日から土曜日にかけて抽出する。
- 日曜日に Cron を使用して抽出された URI を数え上げ、アクセス頻度の高いものを表示する。また、今日それが何回アクセスされているか表示する。

3　システムの構築

3.1　プログラム実行手順

1. squid の アクセスログ から個人名を抜き出すプログラムと、その個人別に URI を抽出し、アクセス件数表示させる。
2. Cron システムを使用し、上記のプログラムを実行させ、実行結果をテキストファイルに保存する。
3. 実行結果を出力したテキストファイルを参照し、CGI を用いて Web 上に掲載する。
4. さらに、登録した URI と Cron システムによって昨日一日分ののアクセスログとを比較し、昨日何件アクセスされたかを CGI によって Web 上に掲載する。

3.2　実行結果

Kterm 上で Ruby スクリプトを用いて個人の名前を抽出した結果を図 2 に示す。

同じく Kterm 上で Ruby スクリプトを用い、個人別のログを抽出し、出力した結果を図 3 に表す。例では今日の実行したその瞬間までにアクセスした私自身のアクセスログを抽出している。

そして、端末上で動かしていたプログラムを Cron システムによって自動実行し、その実行結果を CGI にて個人名別に link しているものを図 4 に示す。

図 5 には、図 4 に記されていた link にアクセスしたとき表示される出力画面を掲載する。

登録した URI を昨日のアクセスログと比較し、何件アクセスされていたかを表示するものを図 6 に示す。また個人の必要に応じて登録した URI を消すためにチェックボタンを配置し、チェックをいれてから delete キーを押すと消えるようにした。

図 7 には管理人への連絡手段をメイルにて行えるようにしたものを表示する。また、図 8 に出力結果を掲載する。

図 9 に、Auto Bookmark 表示結果を示す。

```
[oguri@lindow04 sotuken]$ namae.rb
root
andrew
oguri
yada
tadaaki
kashio
uetsuji
ohnishi
imanaka
hajime
mikane
nakamae
[oguri@lindow04 sotuken]$
```

図 2　個人名抽出結果

```
[oguri@lindow04 sotuken]$ oguri.rb
88回: http://www.fken/rwho4.html
8回: http://mixi.jp/
8回: http://www.google.co.jp/
5回: http://www2.ruby-lang.org/ja/20020101.html
5回: http://www.fken/%7Eoguri/
5回: http://www2.ruby-lang.org/ja/20020102.html
3回: http://www.fken/~oguri/log/test4/
2回: http://www0.ise.osaka-sandai.ac.jp/%7Eoguri/log/test/mailform.html
1回: http://www.nslabs.jp/
1回: http://www.fken/~oguri/log/test3/
1回: http://www.fken/%7Eohnishi/
1回: http://www.freebsd.org/doc/ja_JP.eucJP/books/handbook/disks.html
1回: http://www.ruby-lang.org/ja/20020101.html
1回: http://www.freebsd.org/doc/ja_JP.eucJP/books/handbook/openssh.html
[oguri@lindow04 sotuken]$
```

図 3　個人別 URI 抽出結果

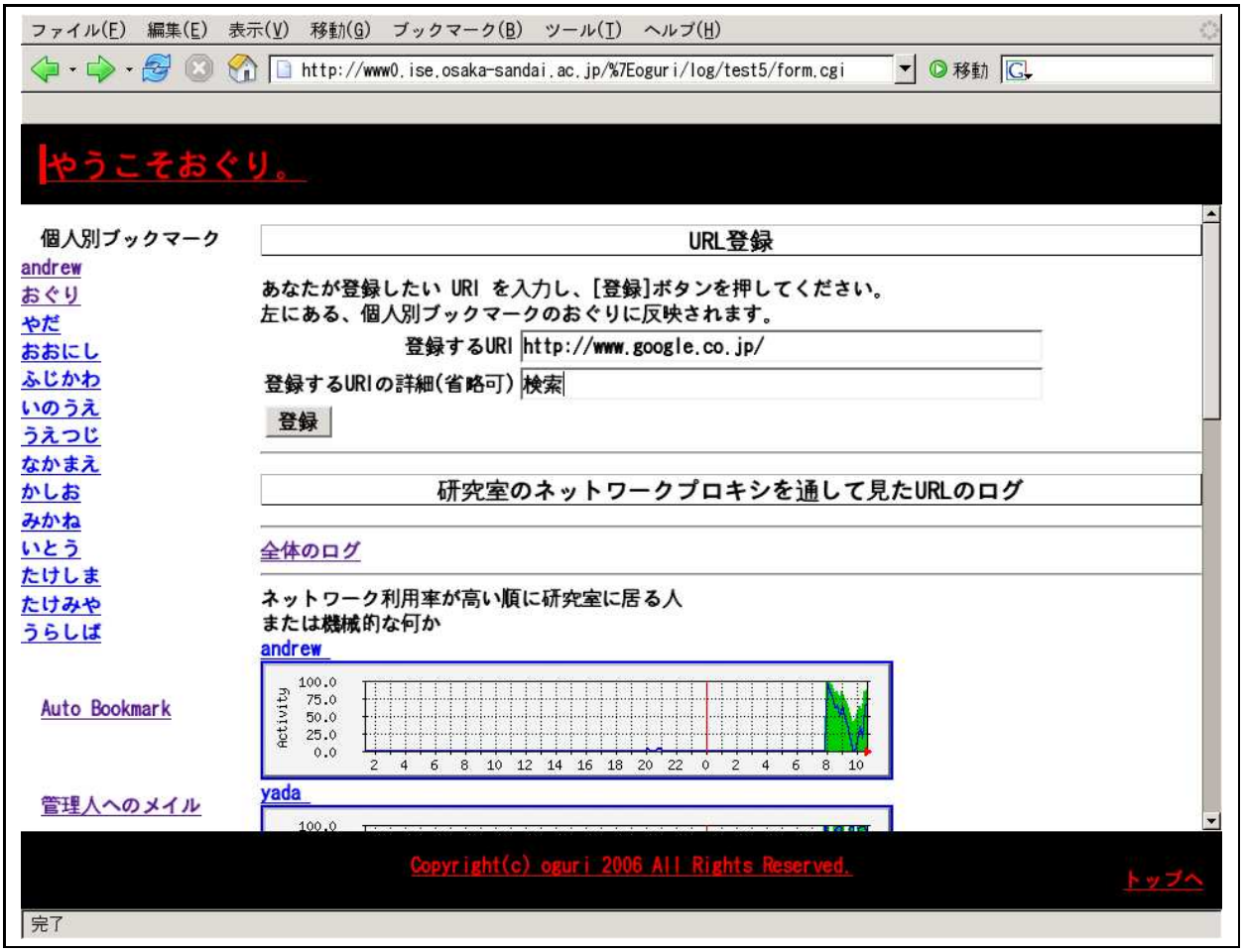

図 4　Web 上に動的に掲載されるもの

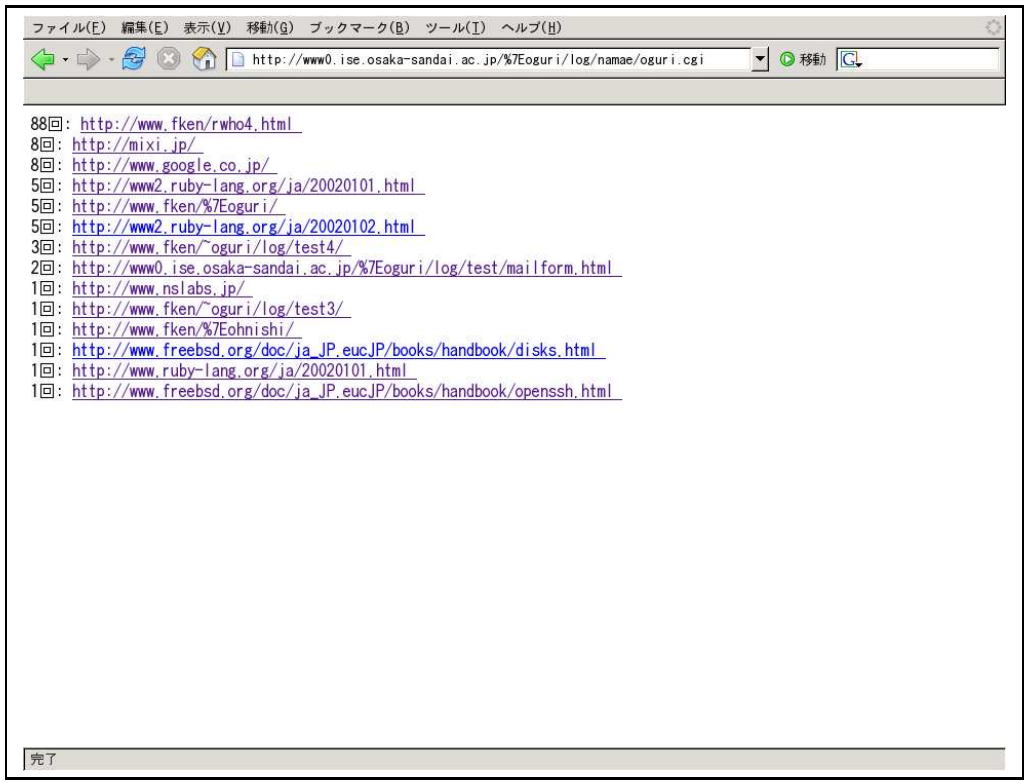

図 5　全体の閲覧 URI

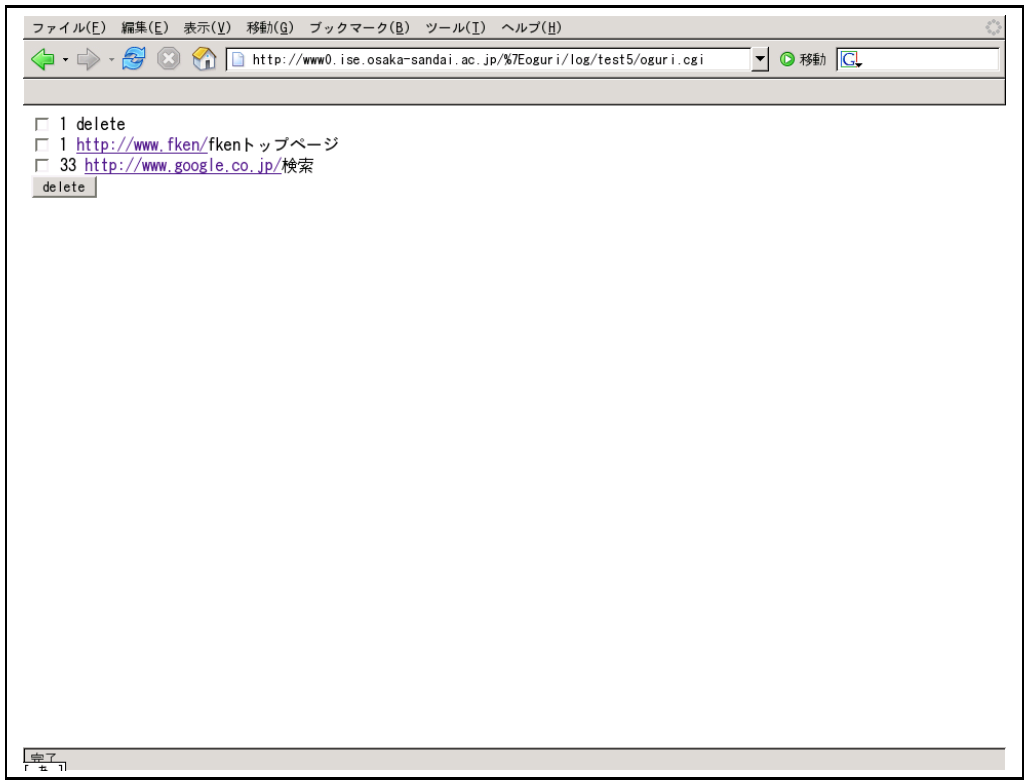

図 6　個人登録 URI 画面

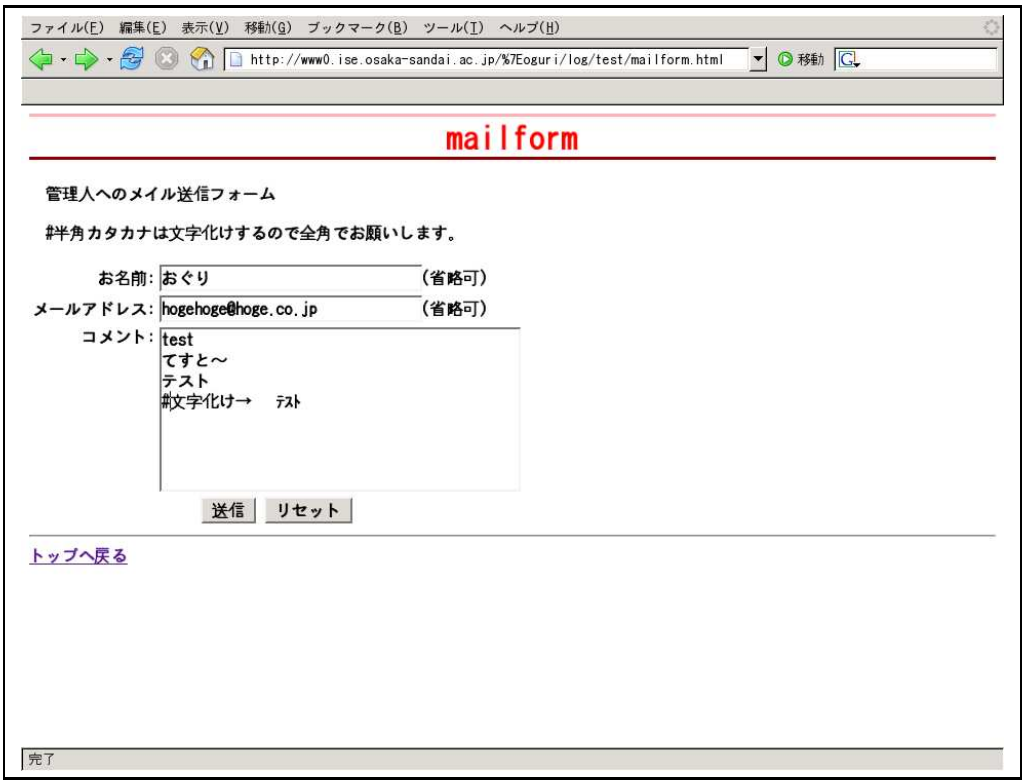

図７　メイル送信フォーム

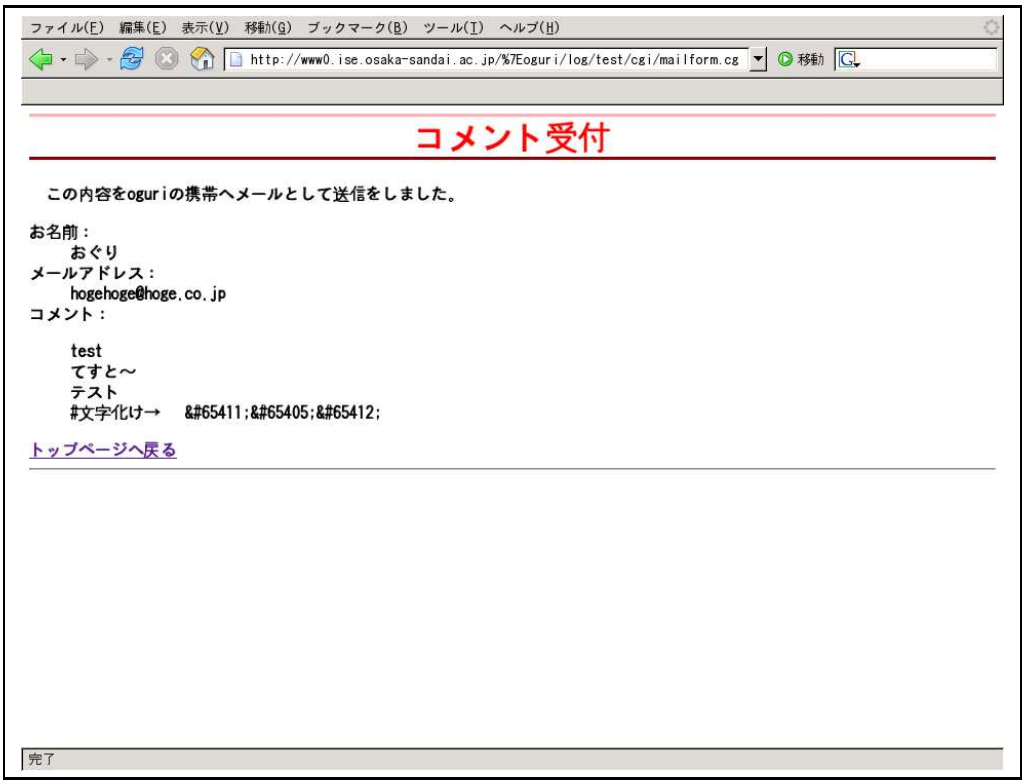

図８　メイル出力結果確認

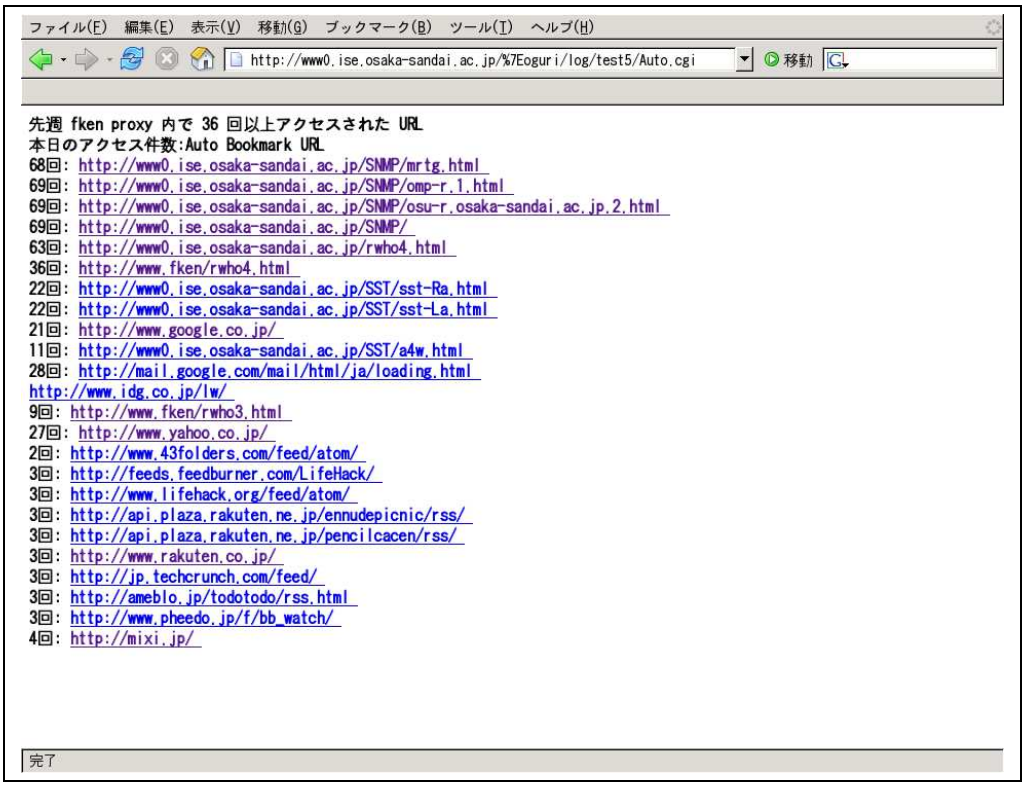

図 9　Auto Bookmark 画面

4 まとめ

4.1 現状

1. Web ブラウザ等を通して、URI を参照し、Web page をみた各 user 名をアクセスログから抽出するプログラムと、その user が参照した URI を抽出し、何回参照したのかを表示するプログラムを書いた。
2. Cron システムを使用し、上記のプログラムを 15 分毎に実行し、テキストファイルに保存するように設定した。なお、15 分毎にテキストファイルは上書きされる。
3. 上記で保存しておいた user 名を、CGI プログラムを用い、Web 上に掲載し、link で表示した。
4. 上記の link からその user 名が本日閲覧した URI を link 表示で掲載されるように CGI プログラムを構築した。
5. 登録する URI をログイン後表示されるテキストボックスに入力すると、左にある名前のページに書き込まれるようにプログラムを構築した。
6. Auto Bookmark 機能として、先週アクセスされた URI の件数が 36 回を越えると Web 上にその URI が掲載される。また、その URI が、今日何回アクセスされたかも表示させた。

4.2 今後の課題

- 個人で登録している URI の閲覧件数表示に重ねて、ウェブを何分みていたかという閲覧時間を squid から読み取って抽出するプログラムを追加する。

謝辞

本研究を進めていく上で、御指導及び御協力を頂いた大垣 斉講師・藤井 信夫教授・fken.a4w メイルリストのメンバーの方々、そして情報システム工学科の卒研生の方々に深く感謝の意を表します。

参考文献

[1] オブジェクト指向言語 Ruby, [http://www2.ruby-lang.org/ja/]
[2] IT 用語辞典, [http://e-words.jp/]
[3] たのしい Ruby -Ruby で始める気軽なプログラミング-, ISBN[4-7973-1408-7]
[4] Ruby プログラミング基礎講座,ISBN[4-7741-2645-4]
[5] Ruby Magic -Ruby で極める正規表現,ISBN[4-274-06488-3]
[6] Ruby レシピブック,ISBN[4-7973-2429-5]
[7] Ruby de CGI -Ruby で作るインタラクティブ Web サイト-, ISBN[4-274-06472-7]
[8] Ruby による CGI プログラミング,ISBN[4-88647-226-5]

付録 A　プログラム

A.1　プログラムのソース

A.1.1　ログオン画面のソース

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html lang="ja">
<head>
<link rel="stylesheet" type="text/css" href="/%7Eoguri/style/oguri.css">
  <title>cgi Hack 中</title>
</head>
<body>
<h1>Social Bookmark へのログオン</h1>
<form method="post" action="./form.cgi">
<table border="0" class="panel">
  <tr><th align="right">ユーザー ID:
    <td><input type="text" size="10" name="user">
  <tr><th align="right">パスワード:
    <td><input type="password" size="10" name="pwd">
  <tr><td><input type="submit" value="ログオン">
</table>
</form>

<hr>
</body>
</html>
```

A.1.2　CGI スクリプト

```
#!/usr/bin/ruby

require "cgi"

$SAFE = 1
cgi = CGI.new

Dir.chdir("/var/tmp/oguri/wwwdata")
pwdfile = "pwdfile"
#pwd チェック
def check_pwd(pwdfile, user, pwd)
```

```
  return false if !user || !pwd
  File.open(pwdfile, "r") {|fp|
    fp.each_line {|line|
      x = line.gsub(/[\r\n]/, '').split(':')
      return true if user == x[0] && pwd == x[1]
    }
  }
  return false
end


user = cgi["user"].first
pwd = cgi["pwd"].first
uri = cgi["URI"].first
kome = cgi["kome"].first

if !check_pwd(pwdfile, user, pwd)

  print <<EOF
Content-Type: text/html; charset=EUC-JP

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html lang="ja">
<head>

<link rel="stylesheet" type="text/css" href="/%7Eoguri/style/oguri.css">
    <title>cgi Hack 中</title>
</head>

<body>

<h2>ユーザー ID またはパスワードが違います。</h2>
<p>Log On フォームへは<a href="http://www0.ise.osaka-sandai.ac.jp/%7Eoguri/log/test4/home.html">
こちら</a>
</body>
</html>
EOF

else
```

```
Dir.chdir("/var/tmp/oguri")

count_file = "count.txt"
count = nil
open(count_file, "r+") do |fp|
  fp.flock(File::LOCK_SH)
  count = fp.read
  count = count.to_i
  count += 1
  count = count.to_s
  fp.flock(File::LOCK_UN)
  fp.flock(File::LOCK_EX)
  fp.rewind
  fp.truncate(0)
  fp.print count
  fp.flock(File::LOCK_UN)
end

Dir.chdir("/var/tmp/oguri/wwwdata")
pwdfile = "pwdfile"
#名前をとる関数
def get_name(pwdfile, user)
    File.open(pwdfile, "r") {|fp|
    fp.each_line {|line|
      x = line.gsub(/[\r\n]/, '').split(':')
      return x[2] if user == x[0]
    }
  }
  return nil
end

print <<EOF
Content-Type: text/html; charset=EUC-JP

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html lang="ja">
<head>
<link rel="stylesheet" type="text/css" href="/%7Eoguri/style/oguri.css">
    <title>home</title>
</head>
<body>
<div id="head">
```

```
<h4>やうこそ#{get_name(pwdfile, user)}。</h4>

</div>

<div id="menu">

<p>個人別ブックマーク<br>

<a href="./andrew.cgi">andrew</a><br>
<a href="./oguri.cgi">おぐり</a><br>
<a href="./yada.cgi">やだ</a><br>
<a href="./ohnishi.cgi">おおにし</a><br>
<a href="./fujikawa.cgi">ふじかわ</a><br>
<a href="./tadaaki.cgi">いのうえ</a><br>
<a href="./uetsuji.cgi">うえつじ</a><br>
<a href="./nakamae.cgi">なかまえ</a><br>
<a href="./kashio.cgi">かしお</a><br>
<a href="./mikane.cgi">みかね</a><br>
<a href="./hajime.cgi">いとう</a><br>
<a href="./takesima.cgi">たけしま</a><br>
<a href="./takemiya.cgi">たけみや</a><br>
<a href="./urashiba.cgi">うらしば</a><br><br>

<a href="./Auto.cgi"><p>Auto Bookmark</a><br>

<a href="../test/mailform.html">


<div style="position:absolute;bottom:10px;">
管理人へのメイル
</div>
</a>
</div>

<div id="main">

<h2>URL 登録</h2>
あなたが登録したい URI を入力し、[登録] ボタンを押してください。<br>
左にある、個人別ブックマークの#{get_name(pwdfile, user)}に反映されます。<br>

<form method="post" action="./form.cgi">
<table border="0" class="panel">
  <tr><th align="right">登録する URI
```

```
    <td><input type="text" size="50" name="URI">
  <tr><th align="right">登録する URI の詳細（省略可）
    <td><input type="text" size="50" name="kome">
    <input type="hidden" name="user" value="#{user}">
    <input type="hidden" name="pwd" value="#{pwd}">
  <tr><td><input type="submit" value="登録">
</table>


<hr>

<h2>研究室のネットワークプロキシを通して見た URL のログ</h2>

<hr>

<a href="/%7Eoguri/log/namae/zentai.cgi">全体のログ</a><br>

<hr>

ネットワーク利用率が高い順に研究室に居る人<br>
または機械的な何か<br>

EOF

Dir.chdir("/home/2006/oguri/public_html/log/namae")

  open("namae.txt","r") do |namae|
    while line = namae.gets
      print <<EOF
<a href="/%7Eoguri/log/namae/#{line}.cgi">#{line}</a><br>

<td><a href="/SNMP/rwho.#{line}.html"><img src="/SNMP/rwho.#{line}-day.png"></a></td><br>

EOF

    end
  end
print <<EOF

<hr>

<a href="./home.html">ログアウト</a>
```

```
<p align="right">Total Visitor  -  #{count}</p>

</div>

<div id="foot">

<h5><p align="center">Copyright(c) oguri 2006 All Rights Reserved.</p></h5>


<a href="http://www0.ise.osaka-sandai.ac.jp/%7Eoguri/">
<span style="position:absolute;right:0px;bottom:0px;">
<h5>トップへ</h5>

</span>
</a>


</div>

</body></html>
EOF

end

Dir.chdir("/var/tmp/oguri/wwwdata/user")
if user == "oguri"
  if %r|^(http://[^?#;]*)|x =~ uri
    File.open("oguri.txt", "a") do |fh|
      fh.print $1 ,"\s", kome,"\n"
    end
  end
end

if user == "tadaaki"
  if %r|^(http://[^?#;]*)|x =~ uri
    File.open("tadaaki.txt", "a") do |fh|
      fh.print $1 ,"\s", kome,"\n"
    end
  end
end


if user == "andrew"
```

```
  if %r|^(http://[^?#;]*)|x =~ uri
    File.open("andrew.txt", "a") do |fh|
      fh.print $1 ,"\s", kome,"\n"
    end
  end
end
if user == "mikane"
  if %r|^(http://[^?#;]*)|x =~ uri
    File.open("mikane.txt", "a") do |fh|
      fh.print $1 ,"\s", kome,"\n"
    end
  end
end
if user == "hajime"
  if %r|^(http://[^?#;]*)|x =~ uri
    File.open("hajime.txt", "a") do |fh|
      fh.print $1 ,"\s", kome,"\n"
    end
  end
end
if user == "nakamae"
  if %r|^(http://[^?#;]*)|x =~ uri
    File.open("nakamae.txt", "a") do |fh|
      fh.print $1 ,"\s", kome,"\n"
    end
  end
end
if user == "kashio"
  if %r|^(http://[^?#;]*)|x =~ uri
    File.open("kashio.txt", "a") do |fh|
      fh.print $1 ,"\s", kome,"\n"
    end
  end
end
if user == "fujikawa"
  if %r|^(http://[^?#;]*)|x =~ uri
    File.open("fujikawa.txt", "a") do |fh|
      fh.print $1 ,"\s", kome,"\n"
    end
  end
end
if user == "ohnishi"
  if %r|^(http://[^?#;]*)|x =~ uri
```

```
    File.open("ohnishi.txt", "a") do |fh|
      fh.print $1 ,"\s", kome,"\n"
    end
  end
end
if user == "urashiba"
  if %r|^(http://[^?#;]*)|x =~ uri
    File.open("urashiba.txt", "a") do |fh|
      fh.print $1 ,"\s", kome,"\n"
    end
  end
end
if user == "uetsuji"
  if %r|^(http://[^?#;]*)|x =~ uri
    File.open("uetsuji.txt", "a") do |fh|
      fh.print $1 ,"\s", kome,"\n"
    end
  end
end
if user == "takesima"
  if %r|^(http://[^?#;]*)|x =~ uri
    File.open("takesima.txt", "a") do |fh|
      fh.print $1 ,"\s", kome,"\n"
    end
  end
end
if user == "takemiya"
  if %r|^(http://[^?#;]*)|x =~ uri
    File.open("takemiya.txt", "a") do |fh|
      fh.print $1 ,"\s", kome,"\n"
    end
  end
end
if user == "yada"
  if %r|^(http://[^?#;]*)|x =~ uri
    File.open("yada.txt", "a") do |fh|
      fh.print $1 ,"\s", kome,"\n"
    end
  end
end
```

A.1.3 Cron

% Crontab のソース

%15 分に一度プログラムを実行して、保存する。

```
0,15,30,45 * * * * /home/2006/oguri/sotuken/bin/pro2/namae.rb >
/home/2006/oguri/public_html/log/namae/namae.txt

0,15,30,45 * * * * /home/2006/oguri/sotuken/bin/pro2/root.rb >
/home/2006/oguri/public_html/log/namae/root.txt

0,15,30,45 * * * * /home/2006/oguri/sotuken/bin/pro2/andrew.rb >
/home/2006/oguri/public_html/log/namae/andrew.txt

0,15,30,45 * * * * /home/2006/oguri/sotuken/bin/pro2/uetsuji.rb >
/home/2006/oguri/public_html/log/namae/uetsuji.txt

0,15,30,45 * * * * /home/2006/oguri/sotuken/bin/pro2/oguri.rb >
/home/2006/oguri/public_html/log/namae/oguri.txt

0,15,30,45 * * * * /home/2006/oguri/sotuken/bin/pro2/mikane.rb >
/home/2006/oguri/public_html/log/namae/mikane.txt

0,15,30,45 * * * * /home/2006/oguri/sotuken/bin/pro2/tadaaki.rb >
/home/2006/oguri/public_html/log/namae/tadaaki.txt

0,15,30,45 * * * * /home/2006/oguri/sotuken/bin/pro2/yada.rb >
/home/2006/oguri/public_html/log/namae/yada.txt

0,15,30,45 * * * * /home/2006/oguri/sotuken/bin/pro2/urashiba.rb >
/home/2006/oguri/public_html/log/namae/urashiba.txt

0,15,30,45 * * * * /home/2006/oguri/sotuken/bin/pro2/hajime.rb >
/home/2006/oguri/public_html/log/namae/hajime.txt

0,15,30,45 * * * * /home/2006/oguri/sotuken/bin/pro2/nakamae.rb >
/home/2006/oguri/public_html/log/namae/nakamae.txt

0,15,30,45 * * * * /home/2006/oguri/sotuken/bin/pro2/ohnishi.rb >
/home/2006/oguri/public_html/log/namae/ohnishi.txt

0,15,30,45 * * * * /home/2006/oguri/sotuken/bin/pro2/fujikawa.rb >
/home/2006/oguri/public_html/log/namae/fujikawa.txt
```

```
0,15,30,45 * * * * /home/2006/oguri/sotuken/bin/pro2/kashio.rb >
/home/2006/oguri/public_html/log/namae/kashio.txt

0,15,30,45 * * * * /home/2006/oguri/sotuken/bin/pro2/takemiya.rb >
/home/2006/oguri/public_html/log/namae/takemiya.txt

0,15,30,45 * * * * /home/2006/oguri/sotuken/bin/pro2/takesima.rb >
/home/2006/oguri/public_html/log/namae/takesima.txt

0,15,30,45 * * * * /home/2006/oguri/sotuken/bin/pro2/zentai.rb >
/home/2006/oguri/public_html/log/namae/zentai.txt

% アクセスログが更新されるまえのものを保存。
14 21 * * 1,2,3,4,5,6 /home/2006/oguri/sotuken/bin/pro2/zentai2.rb >
/home/2006/oguri/public_html/log/test5/zentai.txt

%Auto Bookmark
0 12 * * 0 rm /home/2006/oguri/public_html/log/test5/Auto.txt
0 21 * * 1,2,3,4,5,6 /home/2006/oguri/public_html/log/test5/log.rb
10 12 * * 0 /home/2006/oguri/public_html/log/test5/log2.rb
20 12 * * 0 /home/2006/oguri/public_html/log/test5/log3.rb
30 12 * * 0 rm /home/2006/oguri/public_html/log/test5/accesslog.txt
30 12 * * 0 rm /home/2006/oguri/public_html/log/test5/accesslog2.txt
```

A.1.4 oguri's CGI

```
% oguri の登録 URI 解析 CGI のソース
#!/usr/bin/ruby

print "Content-Type: text/html; charset=EUC-JP\n\n"

require "cgi"
cgi = CGI.new
$SAFE = 1

print "<html><body>\n"

Dir.chdir("/var/tmp/oguri/wwwdata/user")
i = 0
open("oguri.txt","r") do |fp|
  while line = fp.gets
    i += 1
    for x in 1..#{i} do
        i += 1
      name = line.split(/\s+/,2)

      if "on" == cgi["check#{x}"].first
        name[0].sub!(/#{name[0]}/,'')
        name[1].sub!(/#{name[1]}/,'delete')
      end
    end
    Dir.chdir("/home/2006/oguri/public_html/log/test5")
    open("zentai.txt","r") do |fh|
      while line2 = fh.gets
        name2 = line2.split(/\s+/,2)
        if %r|#{name[0]}\s| =~ name2[1]
          hoge = name2[0]
        end
      end

#      p name2[0]
    print <<EOF
<form method="post" action="./oguri.cgi">
<input type="checkbox" name="check#{i}" id="c#{i}">
        #{hoge}
<a href="#{name[0]}">#{name[0]}</a>#{name[1]}<br>
```

```
EOF
    end

  end
  print <<EOF
<input type="submit" value="delete">
</body></html>
EOF
end
```

A.1.5 oguri's CGI

% 個人アクセス URI

```
#!/usr/bin/ruby
print "Content-Type: text/html; charset=EUC-JP\n\n"

name = []

require "cgi"


print "<html><body>"

Dir.chdir("/home/2006/oguri/public_html/log/namae")

open("oguri.txt","r") do |root|
  while line = root.gets
    name = line.split(/\s+/,2)
    print <<EOF
    #{name[0]}
    <a href="#{name[1]}">#{name[1]}</a><br>
EOF
  end
  print"</body></html>"
end
```

A.1.6 namae ソース

```
% 名前抽出ソース

#!/usr/bin/ruby

name = []


count = Hash.new(0)

Dir.chdir("/misc/log/squid")

open("access.log", "r") do |log|
  while line = log.gets
    name = line.split(/\s+/,10)

    if name[7] == "-"
      name[7] = "root"
    end

    name[7].each{|namae| #スペースで区切られた 7 番目のところを配列に格納
      count[namae] +=1
    }

  end
end

count.sort{|a,b|
  b[1] <=> a[1]
}.each{|key,value|
  print "#{key}\n"

}
```

A.1.7 mailform source

```
#メイル送信フォームの HTML ソース
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html lang="ja">
<head>
```

```
<link rel="stylesheet" type="text/css" href="/%7Eoguri/style/oguri.css">
<link rel="index" href="./">
<LINK REV="MADE" HREF="mailto:oguri@h-ps011.ise.osaka-sandai.ac.jp">
<meta http-equiv="Content-Style-Type" content="text/css">
<meta http-equiv="Content-Type" content="text/html
;charset=ISO-2022-JP">

<title>cgi hack 中</title>
</head>
<body>

<h1>mailform</h1>

<form method="post" action="./cgi/mailform.cgi">
<p>管理人へのメイル送信フォーム
<p>#半角カタカナは文字化けするので全角でお願いします。
<table class="mailform" style="margin-top:1em">
<tr>
 <th align="right">お名前:
 <td nowrap>
  <input type="text" name="name" size="30">(省略可)

<tr>
 <th align="right">メールアドレス:
 <td nowrap>
  <input type="text" name="mail" size="30">(省略可)

<tr>
 <th align="right" valign="top">コメント:
 <td><textarea name="body" rows="7" cols="40"></textarea>
<tr>
 <td colspan="2" align="center">
  <input type="submit" value="送信">
  <input type="reset">

</table>
</form>



<hr>
<a href ="http://www0.ise.osaka-sandai.ac.jp/%7Eoguri/" > トップへ戻る</a>
</body>
```

```
</html>
```

A.1.8 mailform CGI source

#メイルフォームの CGI のソース

```
#!/usr/bin/ruby
# -*- mode:ruby; encoding:euc-jp -*-

require "cgi"
require "nkf"
require "mailform-conf.rb"

$SAFE = 1
ENV['PATH'] = "/bin"

def j2e(s)
  s ? NKF.nkf('-e -J -m0', s) : nil
end

def e2j(s)
  s ? NKF.nkf("-j -E -m0", s) : nil
end

class CGI
  def get(name)
    !self[name].empty? ? self[name].first.strip : ""
  end
end

class String
  def shift_e
    return nil if self == ""
    a = split(//e, 2)
    replace(a[1])
    a[0]
  end

  def unshift(s)
    self[0, 0] = s
  end
end
```

```
def header_out(title)
  print <<EOF
Content-Type: text/html; charset=EUC-JP

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html lang="ja">
<head>
<link rel="stylesheet" type="text/css" href="/%7Eoguri/style/oguri.css">
<title>#{title}</title>
</head>
<body>
<h1>#{title}</h1>
EOF
end

def get_data(cgi)
  name = j2e(cgi.get("name"))
  mail = j2e(cgi.get("mail"))
  body = j2e(cgi.get("body"))
  return [name, mail, body]
end

def print_retry()
  header_out("コメント欄が空白です")
  print <<EOF
<p>コメント欄が空欄です。
<p><a href="./../mailform.html">メイル送信フォームに戻る</a>  
  <a href="http://www0.ise.osaka-sandai.ac.jp/%7Eoguri/">トップページに戻る</a>
<hr>
</body>
</html>
EOF
end

def print_comment(data)
  header_out("コメント受付")
  print <<EOF
<p>この内容を oguri の携帯へメールとして送信をしました。
<dl>
  <dt>お名前：
  <dd>#{CGI.escapeHTML(data[0])}
  <dt>メールアドレス：
```

```
  <dd>#{CGI.escapeHTML(data[1])}
  <dt>コメント：
  <dd><pre>#{CGI.escapeHTML(data[2])}</pre>
</dl>

<a href="http://www0.ise.osaka-sandai.ac.jp/%7Eoguri/">トップページへ戻る</a>
<hr>
</body>
</html>
EOF
end

# メールヘッダーを生成する
# @param name ヘッダーフィールド名
# @param value ヘッダーフィールド値 ( 日本語 EUC )
def mail_header(name, value)
  return name + ": " + value + "\n" if value !~ /[^\x1-\x7f]/
  result = name + ":"
  str = value.dup
  while str.length > 0
    line = ""
    while (c = str.shift_e) != nil &&
          e2j(line + c).to_a.pack("m").chomp.length < 75 - 18
      line += c
    end
    str.unshift(c) if c
    result += " =?ISO-2022-JP?B?" +
              e2j(line).to_a.pack("m").chomp + "?=\n"
  end
  return result
end

# コメントをメールメッセージに整形する
def format_comment(data)
  header = <<EOF
From: #{FROM_ADDR}
To: #{TO_ADDR}
MIME-Version: 1.0
Content-Type: text/plain; charset=ISO-2022-JP
EOF
  header += mail_header("Subject", SUBJECT)
  if data[1] != "" && data[1] !~ /[^\x21-\x7e]/
    header += "Reply-To: #{data[1]}\n"
```

```
  end
  mailbody = <<EOF
Web サイトへのコメントがありました。

          日時：#{CGI.rfc1123_date(Time.now)}
        お名前：#{data[0]}
メールアドレス：#{data[1]}

コメント：
----
EOF
  data[2].split(/\r?\n/).each {|line|
    mailbody << "#{line}\n"
  }
  mailbody << "----\n"
  return [header, mailbody]
end

# メールを送信する
def send_mail(msg)
  pipe = IO.popen(INJECTOR, "w")
  msg[0].split(/\r?\n/).each {|line|
    pipe.print line, "\n"
  }
  pipe.print "\n"
  msg[1].split(/\r?\n/).each {|line|
    pipe.print NKF.nkf("-j -E -m0 -f70", line).chomp, "\n"
  }
  pipe.close
end

def error_out()
  print "Content-Type:text/html\n\n"
  print "<p>#{CGI.escapeHTML($!.inspect)}<br>\n"
  $@.each {|x| print CGI.escapeHTML(x), "<br>\n"}
end

if __FILE__ == $0
  begin
    cgi = CGI.new
    data = get_data(cgi)
    if data[2] == ""
      print_retry()
```

```
    else
      send_mail(format_comment(data))
      print_comment(data)
    end
  rescue StandardError
    error_out()
  rescue ScriptError
    error_out()
  end
end
#mailconf ソース
FROM_ADDR = "oguri@h-ps011.ise.osaka-sandai.ac.jp"
#TO_ADDR = "chon-chon_u-kaz_1004@dk.pdx.ne.jp"

SUBJECT = "Web サイトへのコメント"
INJECTOR = "/usr/sbin/sendmail -i -t"
```

A.1.9 Auto Bookmark のための Ruby スクリプト

```
% 月曜日から土曜日までアクセスログを抽出する分
#!/usr/bin/ruby

Dir.chdir("/misc/log/squid")
open("access.log","r") do |fp|
  while line = fp.gets
    line2 = line.split(/\s+/,10)
    Dir.chdir("/home/2006/oguri/public_html/log/test5")
    open("accesslog.txt","a+") do |fh|
      fh.print line2[6]
      fh.print "\n"
    end
  end
end
% 抽出したものの中で同じ物があればカウントし、まとめる分
#!/usr/bin/ruby


count = Hash.new(0)

open("accesslog.txt", "r") do |log|
  while line = log.gets
    if %r|(http://[^?#;]*/)\n|x =~ line
```

```
      $1.each{|url|
        count[url] +=1
      }
    end
    if %r|(https?://[^?#;]*\.s?html)\n|xi =~ line
      $1.each{|url|
        count[url] +=1
      }
    end
  end
end


count.sort{|a,b|
  b[1] <=> a[1]
}.each{|key,value|
  open("accesslog2.txt","a") do |fp|
    fp.print value
    fp.print "\s"
    fp.print key
    fp.print "\n"
  end
}
```

% まとめたものから 36 回以上の URI を抜くスクリプト

```
#!/usr/bin/ruby

open("accesslog2.txt","r") do |fp|
  while line = fp.gets
    line2 = line.split(/\s/,2)
    if line2[0].to_i >= 36
      open("Auto.txt","a+") do |fh|
        fh.print line2[1]
      end
    end
  end
end
```

A.1.10 Auto Bookmark's CGI

```
#!/usr/bin/ruby
```

```
require "cgi"

cgi = CGI.new

print <<EOF
Content-Type: text/html; charset=EUC-JP

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html lang="ja">
<head>

<link rel="stylesheet" type="text/css" href="/%7Eoguri/style/oguri.css">
    <title>cgi Hack 中</title>
</head>

<body>
先週 fken proxy 内で 36 回以上アクセスされた URL<br>
本日のアクセス件数:Auto Bookmark URL<br>
EOF

open("Auto.txt","r") do |fp|
  while line = fp.gets
    Dir.chdir("/home/2006/oguri/public_html/log/namae")
    open("zentai.txt","r") do |fh|
      while line2 = fh.gets
        line3 = line2.split(/\s+/,2)
        if line == line3[1]
          print line3[0]
        end
      end
    end

    print <<EOF
    <a href="#{line}">#{line}</a><br>
EOF


  end
end

print <<EOF
</body>
</html>
```

```
EOF
```