

## 構築手順書 「登録端末への DHCP による自動アドレス割当機構」

### フォームの作成

#### 1 入力フォーム

入力フォームでは、管理者が利用者のユーザデータを入力しやすく、かつ入力誤りが少なくなるように配慮して作成する。「校内 IP アドレス設定フォーム」では、利用者の最低限必要データとしてユーザ名、MAC アドレスが必要となる。その他に、割り当てる IP アドレスを入力する必要がある。それ以外の情報として、登録するユーザが新規または修正なのか、誰が登録作業を行ったかといった情報を残すために、設定状況と入力担当者の欄を設けている。

##### ユーザ名

ユーザ名については、利用者を判別する際に利用されるため、ここでは半角英数字で入力するようコメントを表示する。

##### MAC アドレス

MAC アドレスは、16 進数 12 桁で 2 桁ずつの区切りがあるため、その記述が統一した形で入力できるように半角英数字のみ入力させるようにする。

##### 割当 IP アドレス

ここに入力する IP アドレスは、校内で決めたプライベートアドレスを利用する。別に校内割当 IP 一覧などを作成し、空きアドレスを割り当てるといったことを行う。MAC アドレス同様、半角数字のみの入力となる。

##### 設定状況

これは必ずしも必要ではないが、用途に応じて導入するか検討する。ここでは 2 つを選択するようにしているため、ラジオボタンを採用している。

##### 入力担当者

入力した係りが責任を持つように、また誰が利用者の登録を行ったかわかるように、あらかじめ入力担当の係りの名前をこのページ作成時に入力しておく。ここではプルダウンメニューを使い、入力担当者を選択するようにしている。

##### 登録ボタン

入力が完了すると、「登録」のボタンをクリックすることにより登録作業が終了する。

##### クリアボタン

入力の中止の際に利用する。

IPアドレス登録 - Microsoft Internet Explorer

ファイル(E) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)

戻る 進む 検索 印刷 移動 リンク

アドレス( ) http://linux.sasano/rensyuu/model/form6.html

Google ウェブ検索 ブロック数: 113 オプション

### 校内IPアドレス設定フォーム

ユーザ名:  \*半角英数字で入力して下さい

MAC アドレス:  :  :  :  :  :  \*例 00:01:80:31:EC:E7

割当IPアドレス:  .  .  .  \*例 192.168.0.1

設定状況: ☒ 新規 ☐ 修正

入力担当者:

ページが表示されました インターネット

## 2 入力フォームのソースプログラム

ここでは、CGI・Perl のプログラムを使い、入力フォームに必要事項を入力し、「登録」ボタンをクリックすると form6.cgi の画面が戻ってくるようにしている。完成したフォームの保存場所については、Linux サーバ上の / var / www / html / に保存する。ファイル名については任意でよいが、ここでは form.html としている。( / var / www / html / form.html )

```
<HTML>
<HEAD>
<meta http-equiv="Content-Type" content="text/html; charset=euc-jp">
<TITLE>IPアドレス登録</TITLE>
</HEAD>
<BODY>

<H2>校内IPアドレス設定フォーム</H2>

<FORM METHOD="POST" ACTION="/cgi-bin/form6.cgi">
ユーザ名：
<INPUT TYPE="text" name="name">
*半角英数字で入力して下さい
<P>

MAC アドレス：
<INPUT size="1.5" TYPE="text" name="mac1"> :
<INPUT size="1.5" TYPE="text" name="mac2"> :
<INPUT size="1.5" TYPE="text" name="mac3"> :
<INPUT size="1.5" TYPE="text" name="mac4"> :
<INPUT size="1.5" TYPE="text" name="mac5"> :
<INPUT size="1.5" TYPE="text" name="mac6"> :
*例 00:01:80:31:EC:E7
<P>

割当IPアドレス：
<INPUT size="2" TYPE="text" name="ip1"> .
<INPUT size="2" TYPE="text" name="ip2"> .
<INPUT size="2" TYPE="text" name="ip3"> .
<INPUT size="2" TYPE="text" name="ip4">
*例 192.168.0.1
<P>

設定状況：
<INPUT TYPE="radio" name="jyoukyou" VALUE="new" CHECKED>新規
<INPUT TYPE="radio" name="jyoukyou" VALUE="correction">修正
<P>

入力担当者：
<SELECT NAME="tantou">
<OPTION SELECTED VALUE="sasano">笹野
<OPTION VALUE="yasutake">安武
<OPTION VALUE="yamauchi">山内
</SELECT>
<P>

<INPUT TYPE="submit" VALUE="登録">
<INPUT TYPE="reset" VALUE="クリア">
</FORM>

</BODY>
</HTML>
```

### 3 受け取りフォーム

入力フォームに必要事項を入力し、「登録」ボタンをクリックすると「IP アドレス受け取り状況」が表示されるようにする。ここでは、ユーザ名を sasano、MAC アドレスを 00:0d:60:1e:0c、割当 IP アドレスを 192.168.0.3、設定状況を「新規」、入力担当者を「笹野」とし、登録ボタンをクリックした状態である。入力フォーム（校内 IP アドレス設定フォーム）で入力されたデータが、受け取りフォーム（IP アドレス受け取り状況）に表示される。

The image contains two screenshots of a Microsoft Internet Explorer browser window. The left screenshot shows the '校内IPアドレス設定フォーム' (In-school IP Address Setting Form) at the URL 'http://linux.sasano/rensyuu/model/form6.html'. The form contains the following fields and values: 'ユーザ名' (User Name) is 'sasano'; 'MAC アドレス' (MAC Address) is '00:0d:60:93:1e:0c'; '割当IPアドレス' (Assigned IP Address) is '192.168.0.3'; '設定状況' (Setting Status) has '新規' (New) selected; '入力担当者' (Input Person) is '笹野' (Sasano). There are '登録' (Register) and 'クリア' (Clear) buttons. The right screenshot shows the 'IPアドレス受け取り状況' (IP Address Reception Status) at the URL 'http://linux.sasano/cgi-bin/form6.cgi'. It displays the message '書き出しが完了しました。' (Output completed.) followed by the submitted data: 'ユーザ名 sasano', 'MACアドレス 00:0d:60:93:1e:0c', '割当IPアドレス 192.168.0.3', '設定状況 : new', and '入力担当者 : sasano'.

**校内IPアドレス設定フォーム**

ユーザ名 : sasano \*半角英数字で入力して下さい

MAC アドレス : 00 : 0d : 60 : 93 : 1e : 0c \*例 00:01:80:31:EC:E7

割当IPアドレス : 192 . 168 . 0 . 3 \*例 192.168.0.1

設定状況 : ☒新規 ☐修正

入力担当者 : 笹野

ページが表示されました

**IPアドレス受け取り状況**

書き出しが完了しました。

ユーザ名 sasano

MACアドレス 00:0d:60:93:1e:0c

割当IPアドレス 192.168.0.3

設定状況 : new

入力担当者 : sasano

ページが表示されました

### 4 受け取りフォームのソースプログラム

入力フォームで「登録」のボタンをクリックすることにより、受け取りフォームにデータを返し表示させるというアクションだけではなく、ここではそれと同時に dhcpd.conf とバックアップファイルにも書き出すようにしている。dhcpd.conf とバックアップファイル作成については後で述べる。

完成した受け取りフォームの保存場所については、Linux サーバ上の /var/www/cgi-bin/ に保存する。ファイル名については任意でよいが、ここでは form6.cgi としている。( /var/www/cgi-bin/form6.cgi )

```
#!/usr/bin/perl

read(STDIN,$formin,$ENV{'CONTENT_LENGTH'});

@indata = split(/&/,$formin);
foreach $tmp(@indata)
{
    ($name,$value)=split(/=/,$tmp);
    $name{$name}=$value;
}

$macaddr="$name{'mac1'}:$name{'mac2'}:$name{'mac3'}:$name{'mac4'}:$name{'mac5'}:$name{'mac6'}";
$ipaddr="$name{'ip1'}.$name{'ip2'}.$name{'ip3'}.$name{'ip4'}";

#backup data
open(OUT,">>/etc/dhcpd-data/bkdata.txt");
print OUT "host ";
print OUT "$name{'name'}";
print OUT " {\n";
print OUT "     hardware ethernet $macaddr";
print OUT ";\n";

print OUT "     fixed-address $ipaddr";
print OUT ";\n";
print OUT "#tantou : $name{'tantou'}\n";
print OUT "#jyoukyou : $name{'jyoukyou'}\n";
print OUT "}\n";
close(OUT);

#dhcpd.conf data
open(OUT,">>/etc/dhcpd-data/test.txt");
print OUT "host ";
print OUT "$name{'name'}";
print OUT " {\n";
print OUT "     hardware ethernet $macaddr";
print OUT ";\n";

print OUT "     fixed-address $ipaddr";
print OUT ";\n";
print OUT "#tantou : $name{'tantou'}\n";
print OUT "#jyoukyou : $name{'jyoukyou'}\n";
print OUT "}\n";
close(OUT);

print "Content-type: text/html\n\n";
print "<META http-equiv=\"Content-Type\" text/html; charset=EUC-JP\">\n";

print "<HTML>\n";
print "<HEAD><TITLE>IPアドレス受け取り</TITLE></HEAD>\n";
print "<BODY>\n";
print "<H2>IPアドレス受け取り状況</H2><HR>\n";

print '<H3>書き出しが完了しました。</H3><BR>';
print "ユーザ名 $name{'name'} <BR><BR>";
print "MACアドレス $name{'mac1'}:$name{'mac2'}:$name{'mac3'}:$name{'mac4'}:$name{'mac5'}:$name{'mac6'} <BR><BR>";
print "割当IPアドレス $name{'ip1'}.$name{'ip2'}.$name{'ip3'}.$name{'ip4'} <BR><BR>";
print "設定状況 : $name{'jyoukyou'}<BR><BR>\n";
print "入力担当者 : $name{'tantou'}<BR><BR>\n";

print "</BODY>\n";
print "</HTML>\n";
exit;
```

## 5 受け取りフォームの解説

\$macaddr ・ \$ipaddr

入力フォームから入力された MAC アドレスと IP アドレスを記録するための書式を定義している。  
ここで定義された書式を#backup data と#dhcpd.conf data で活用する。

```
$macaddr="$name{'mac1'}:$name{'mac2'}:$name{'mac3'}:$name{'mac4'}:$name{'mac5'}:$name{'mac6'}";  
$ipaddr="$name{'ip1'}.$name{'ip2'}.$name{'ip3'}.$name{'ip4'}";
```

#backup data

open 文で、/etc/dhcpd-data/という場所に格納されている bkdata.txt ファイルを開き、print 文の内容を書き込んでから閉じるという内容である。print 文で記述している内容は、dhcpd.conf に利用者を登録するための書式となる。

```
#backup data  
open(OUT, ">>/etc/dhcpd-data/bkdata.txt");  
print OUT "host ";  
print OUT "$name{'name'}";  
print OUT " {\n";  
print OUT "     hardware ethernet $macaddr";  
print OUT ";\n";  
  
print OUT "     fixed-address $ipaddr";  
print OUT ";\n";  
print OUT "#tantou : $name{tantou}\n";  
print OUT "#jyoukyou : $name{jyoukyou}\n";  
print OUT "}\n";  
close(OUT);
```

#dhcpd.conf data

上記の#backup data とほとんど同じであり、保存先と保存するファイル名が異なる程度である。open 文で、/etc/dhcpd-data/という場所に格納されている test.txt ファイルを開き、print 文の内容を書き込んでから閉じるという内容である。print 文で記述している内容は、dhcpd.conf に利用者を登録するための書式となり、このファイルが dhcpd.conf ファイルに直接コピーされる。

```
#dhcpd.conf data  
open(OUT, ">>/etc/dhcpd-data/test.txt");  
print OUT "host ";  
print OUT "$name{'name'}";  
print OUT " {\n";  
print OUT "     hardware ethernet $macaddr";  
print OUT ";\n";  
  
print OUT "     fixed-address $ipaddr";  
print OUT ";\n";  
print OUT "#tantou : $name{tantou}\n";  
print OUT "#jyoukyou : $name{jyoukyou}\n";  
print OUT "}\n";  
close(OUT);
```

```
ddns-update-style interim;  
subnet 192.168.0.0 netmask 255.255.255.0 {  
    range 192.168.0.10 192.168.0.17;  
    option subnet-mask 255.255.255.0;  
    option routers 192.168.0.1;  
    option broadcast-address 192.168.0.255;  
    option domain-name-servers , , ;  
}  
host work {  
    hardware ethernet 00:0d:60:93:1e:0c; # work  
    fixed-address 192.168.0.2;  
}  
host sasano {  
    hardware ethernet 00:0d:60:93:1e:0c;  
    fixed-address 192.168.0.3;  
    #tantou : sasano  
    #jyoukyou : new  
}
```

dhcpd.conf ファイル

## 受け取りフォーム

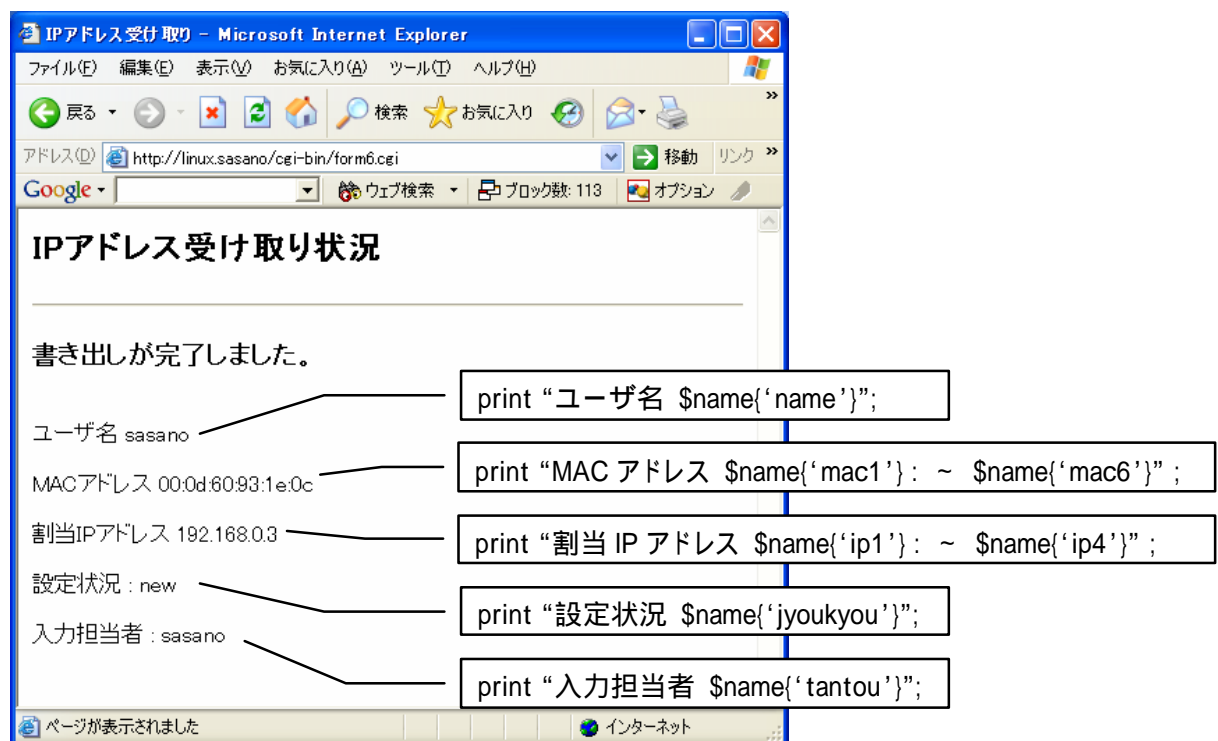
以下のプログラムが受け取りフォームのソースとなる。

```
print "Content-type: text/html\n\n";
print "<META http-equiv=\"Content-Type\" text/html; charset=EUC-JP\">\n";

print "<HTML>\n";
print "<HEAD><TITLE>IPアドレス受け取り</TITLE></HEAD>\n";
print "<BODY>\n";
print "<H2>IPアドレス受け取り状況</H2><HR>\n";

print '<H3>書き出しが完了しました。</H3><BR>';
print "ユーザ名 $name{'name'} <BR><BR>";
print "MACアドレス $name{'mac1'}:$name{'mac2'}:$name{'mac3'}:$name{'mac4'}:$name{'mac5'}:$name{'mac6'} <BR><BR>";
print "割当IPアドレス $name{'ip1'}.$name{'ip2'}.$name{'ip3'}.$name{'ip4'} <BR><BR>";
print "設定状況 : $name{'jyoukyou'}<BR><BR>\n";
print "入力担当者 : $name{'tantou'}<BR><BR>\n";

print "</BODY>\n";
print "</HTML>\n";
exit;
```



## 6 各動作の流れ

ブラウザで入力フォームを表示し、利用者の必要事項を入力する。「登録」ボタンをクリックすると受け取りフォームに入力データが入ったものが表示されると同時に、dhcpd.conf の元になるデータ（この場合のファイル名：test.txt）とバックアップデータ（bkdata.txt）がテキストファイルとして保存される。この2つのファイルは、既存のファイルに追加されたデータを上書きして保存するようになる。受け取りフォームのソースプログラムの「#backup data」と「#dhcpd.conf data」の部分がそれぞれの操作

の内容である。

校内IPアドレス設定フォーム

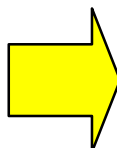
ユーザ名: sasano

VLANアドレス: 30

固定IPアドレス: 192.168.0.1

設定状況: 新規

入力担当者: sasano

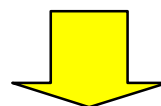


```
ddns-update-style interim;
subnet 192.168.0.0 netmask 255.255.255.0 {
    range 192.168.0.10 192.168.0.17;
    option subnet mask 255.255.255.0;
    option routers 192.168.0.1;
    option broadcast-address 192.168.0.255;
    option domain-name-servers ;
}
host work {
    hardware ethernet 00:0d:80:93:1c:0e; # work
    fixed-address 192.168.0.2;
}
host sasano {
    hardware ethernet 00:0d:60:93:1c:0c;
    fixed-address 192.168.0.3;
    #lanoui : sasano
    rjyoukyou : new
}
```

**bkdata.txt ( backup data )**

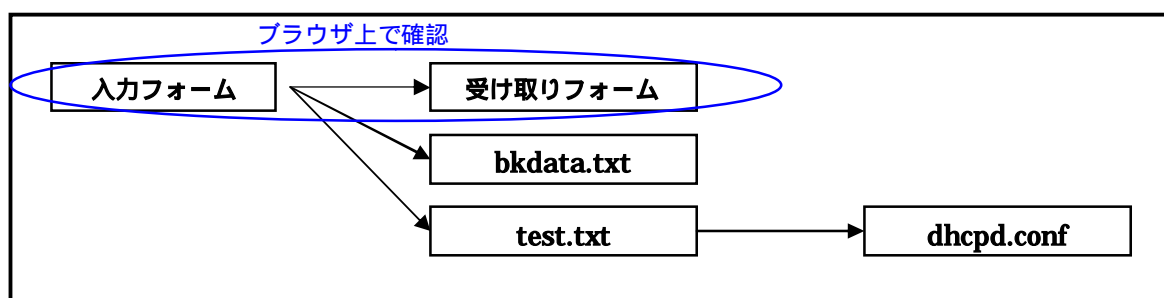
```
ddns-update-style interim;
subnet 192.168.0.0 netmask 255.255.255.0 {
    range 192.168.0.10 192.168.0.17;
    option subnet mask 255.255.255.0;
    option routers 192.168.0.1;
    option broadcast-address 192.168.0.255;
    option domain-name-servers ;
}
host work {
    hardware ethernet 00:0d:80:93:1c:0e; # work
    fixed-address 192.168.0.2;
}
host sasano {
    hardware ethernet 00:0d:60:93:1c:0c;
    fixed-address 192.168.0.3;
    #lanoui : sasano
    rjyoukyou : new
}
```

**test.txt ( dhcpd.conf data )**



```
ddns-update-style interim;
subnet 192.168.0.0 netmask 255.255.255.0 {
    range 192.168.0.10 192.168.0.17;
    option subnet mask 255.255.255.0;
    option routers 192.168.0.1;
    option broadcast-address 192.168.0.255;
    option domain-name-servers ;
}
host work {
    hardware ethernet 00:0d:80:93:1c:0e; # work
    fixed-address 192.168.0.2;
}
host sasano {
    hardware ethernet 00:0d:60:93:1c:0c;
    fixed-address 192.168.0.3;
    #lanoui : sasano
    rjyoukyou : new
}
```

**dhcpd.conf ファイル**



入力フォームで入力されたデータが、受け取りフォーム・bkdata.txt・test.txt の3つファイルに渡され、受け取りフォームはブラウザに表示され、bkdata.txt と test.txt ファイルは /etc/dhcpd-data/ に保存される。また、test.txt は、dhcpd.conf (/etc/dhcpd.conf) に書き換えられる。

このような流れにより、新たに登録された利用者のデータが dhcpd.conf とバックアップファイルに追加される。



## dhcpcd.conf とバックアップファイルの作成

### 1 dhcpcd.conf の reload

入力フォームから入力されたデータが bkdata.txt と test.txt に書き込まれることについては、上記で説明したが、test.txt として作成されたファイルを dhcpcd.conf にどのようにして反映させているかという、reload (再読み込み) させている。dhcpcd.conf のファイルは、DHCP サーバの中核的なプログラムのため、フォームで操作を行った内容をそのまま書き込めるとトラブルの原因になりかねない。また、Linux では dhcpcd.conf ファイルの書き換えを行うと DHCP サーバを再起動させなければならない。そこで、一旦別のテキストファイルに書き込んだファイルを dhcpcd.conf に再読み込みさせるといったイメージとなる。そのプログラム例としては、以下のようになる。

```
#!/bin/sh
cd /etc/dhcpcd-data
if [ -f test.txt ]; then
    mv /etc/dhcpcd.conf /etc/dhcpcd.conf-bak
    cp test.txt /etc/dhcpcd.conf
    kill -TERM `cat /var/run/dhcpcd.pid`
    /usr/sbin/dhcpcd eth1
fi
```

ここでは、/etc/dhcpcd-data/という記憶場所に、dhcpcd-conf-reload というファイル名で上記のファイルを保存する。こうすることにより、入力フォームで入力されたデータが、dhcpcd.conf ファイルに反映され、上書き保存される。

### 2 バックアップファイルの作成

バックアップファイルの作成については、上記でも述べたように受け取りフォームのプログラムに記述されている ( #backup data ) のところの open 文で、/etc/dhcpcd-data/という場所に格納されている bkdata.txt ファイルを開き、print 文の内容を書き込んでから閉じる。当然上書きされるということになる。このファイルがあれば dhcpcd.conf ファイルが壊れたりしても、dhcpcd.conf ファイルの内容と同じものをほぼ同時にバックアップファイルとして作成しているため、バックアップファイル ( bkdata.txt ) のファイル名を変更することにより dhcpcd.conf ファイルを再生することができる。

### 3 バックアップのタイミング

上記のように、利用者を登録する都度バックアップファイルを作成しているが、更に Linux の cron を使ったタイマー的なバックアップを行う。ここでは、毎日 5 分おきにバックアップの更新を行うようにしている。

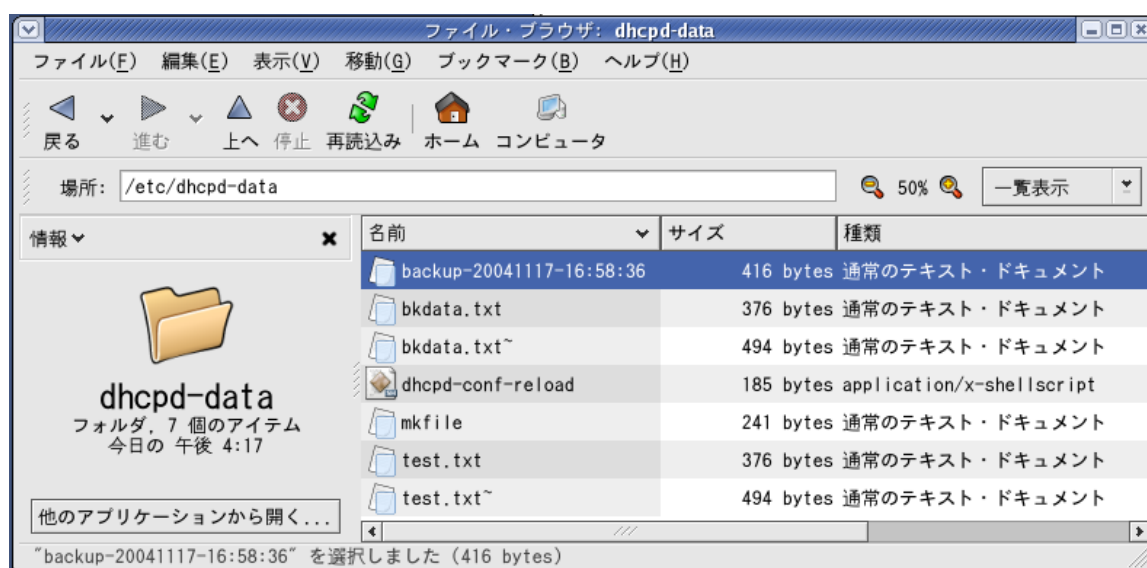
```
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (/tmp/crontab.5866 installed on Wed Nov 17 16:45:49 2004)
# (Cron version -- $Id: crontab.c,v 2.13 1994/01/17 03:20:37 vixie Exp $)
0,5,10,15,20,25,30,35,40,46,50,55 * * * * /etc/dhcpcd-data/dhcpcd-conf-reload
```

プログラムといっても、上記プログラム例の 4 行目だけである。左から 5 分おきに時間指定がなされている。この見方については、0,5,10,15, ~ 50,55 の部分が、分 ( minutes ) のみの指定で、その後の \* が時 ( hour )、その次の \* が日 ( day )、月 ( month )、年 ( year ) となっている。このように細かく指定することができる。



このファイルについては、/var/spool/cron/という場所にある root に 4 行目だけを書き足し保存すればよい。また、バックアップファイルを作成する際に、日付と時間入りのファイル名で保存するプログラムの例を示す。このプログラムについても、/etc/dhcpd-data/という場所に mkfile というファイル名で保存している。

```
#!/perl
($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) = localtime(time);
$mon = $mon + 1;
$year = $year + 1900;
$sti = "$year$mon$mday-$hour:$min:$sec";
open(FILE, "> backup-$sti");
print FILE "current date-time is $sti\n";
close(FILE);
```



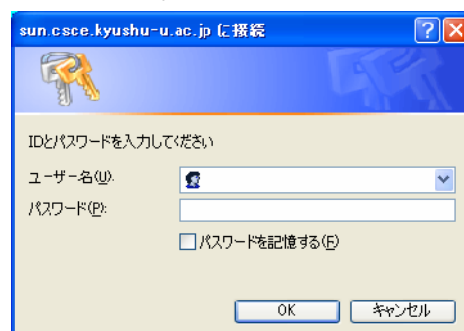
## ユーザ認証用フォームの作成

### 1 パスワードファイルの作成

今回は BASIC 認証を利用したため、その作成について簡単に述べておく。

BASIC 認証でアクセス制限を行うには、「.htaccess ファイル」と「パスワードファイル」の 2 つが必要である。まず、「パスワードファイル」を作成する。ファイル名は特に決まはないが、一般的に「.htpasswd」というファイル名が用いられる。

このファイルには、許可するユーザ名とパスワードを保存する。このとき、パスワードを暗号化する必要があり、パスワードを暗号化する方法について 2 つの方法を紹介しておきたい。



### 2 パスワードの暗号化

パスワードを暗号化する場合、サーバのコンソールにコマンドを入力して作成する方法と、ウェブ上にある暗号化ツールを使うという方法がある。

#### サーバ上での暗号化

サーバのコンソールに telnet や SSH で接続し、htpasswd というコマンドを利用してパスワードフ

ファイルを作成することができる。次の例では、現在のフォルダに.htpasswd というファイルを新しく作成し、sasano というユーザを登録する。パスワードを聞かれるため、同じパスワードを2回入力する(画面には表示されない)。

「-c」はオプションで、パスワードファイルを新しく作成するという意味である。すでにあるファイルにユーザを追加したい場合は不要。作成された.htpasswd は、root 権限で作成すると/root/に保存され、ユーザモードで作成すると、/home/sasano/に保存される。

```
%> htpasswd -c .htpasswd sasano
New password: test
Re-type password: test
Adding password for user sasano
```

#### 暗号化ツールを使う

ウェブ上には、多くの暗号化を行うページが存在し、誰でも自由に利用できるものとして提供されているページがある。そのページにアクセスし、ID とパスワードを入力するとパスワードが暗号化される。 <BASIC 認証用パスワード暗号化ツール <http://orange-factory.com/tool/crypt.cgi> >

そこで暗号化されたパスワードを以下のようにテキストエディタ(メモ帳等)に貼り付け、パスワードファイルとして保存すればよい。(ファイル名: .htpasswd) 保存する場所は特に指定されていないため、ここでは/var/www/html/rensyuu/model/に保存した。

```
sasano:E13BRLEjwKRLU
```

### 3 「.htaccess ファイル」の作成

アクセス制限したいフォルダに「.htaccess」という名前のファイルを作る。これも、内容は普通のテキストファイルである。パスワードファイルと違い、必ずこの名前にする必要がある。

認証用のフォームについては、校内の IP アドレス設定のためのフォームを利用者が閲覧できないようにするための工夫として作成する。これは、制限付きでウェブページを公開する際に用いられる。

作成方法としては、テキストエディタ(メモ帳等)で以下のプログラムコードを入力し、「.htaccess」(拡張子が htaccess ということで、ドットが必要である。)といったファイル名で保存する。保存する場所は、入力フォームを保存しているディレクトリに置いて制限をかけるということになる。

```
AuthType Basic
AuthUserFile /var/www/html/rensyuu/model/.htpasswd
AuthGroupFile /dev/null
AuthName "IDとパスワードを入力してください。"
<limit GET POST>
require valid-user
</limit>
```

- ・ 1 行目「AuthType Basic」:

認証方式を設定する。 Basic 認証を利用する時には「Basic」を指定する。これは、ID とパスワードによるアクセス制御 (Basic 認証) を表す。

- ・ 2 行目「AuthUserFile」:

準備したパスワードファイルを、フルパスで指定する。 Web ページのルートからのパスではなく、

サーバ上パスとなるため、注意が必要である。

- ・ 3 行目「**AuthGroupFile**」:

グループファイル名を指定する。グループを使わないときには「**/dev/null**」を指定する。グループファイルを使うと、あるフォルダは全員閲覧できて、別のフォルダは特定のグループの人だけ閲覧可能、というようなことを実現できる。

- ・ 4 行目「**AuthName**」:

ユーザ名・パスワードを入力するダイアログボックスに表示されるメッセージ。全角文字も指定できるが、文字化けの可能性がある。スペースを含むメッセージを設定するときには、メッセージ全体をダブルクォートで括る。

- ・ 5 行目「**require valid-user**」:

認証させるユーザを指定する。「**valid-user**」と指定すると、「**AuthUserFile**」で指定したファイル内の全ユーザが認証される。「**user ユーザ名**」と指定すると、そのユーザだけが認証される。「**group グループ名**」と指定すると、「**AuthGroupFile**」内に書かれた該当グループのユーザだけが認証される。

その他

以上のファイルを作成し、Linux サーバ上に保存しておけば、DHCP 機能を用いた利用者の固定 IP アドレス割り当てを実現させることができる。

システム開発の構築手順として、以上のように記述したが、これは実行ファイル( exe )とは異なるため、様々な記録媒体に保存して配布することができない。そのため、以上のような操作をコマンドラインで記述していかなければならない。ぜひ、各学校の状況に応じて改良し、活用していただきたいと思う。