
インターフェイスの街角 (44) — なぞなぞドア

増井俊之

前回は、5 択のなぞなぞに正しく解答すると Web ページが参照可能になる “なぞなぞ認証” について説明しました。今回は、同じ手法をオフィスなどの入室管理に適用した “なぞなぞドア” を紹介します。

なぞなぞドア

オフィスの入退室管理などに、磁気カードや非接触カードを使っている会社は多いのではないのでしょうか。私が勤めている会社でも、磁気式の社員カードを使って入退室の管理をしています。しかし、カードをつねに携帯していなければならないのが面倒に感じられる場合もあります。社員カードは通常は首からぶら下げる決まりになっていますが、うっとうしくなって机の上に置いたのを忘れて帰宅してしまったり、カードを持たずにちょっと席を外したりすると、入室できずに途方に暮れるといった目に遭うことになります。

もちろん、ふだんは注意しているので閉め出されるようなことはめったにありません。とはいえ、部屋を出入りするたびにカードが必要なのも煩わしいですし、カードを手放さないようにいつも気を遣っていると疲れてしまいます。

パスワードとして、テンキーなどから数字の入力を求めるシステムもあります。しかし、厳しく管理するには長い数字列が必要なので、憶えるのが難しく、そもそも番号を他人に知られたらお手上げです。

“なぞなぞドア” は、入室しようとする人になぞなぞを提示し、すべての問題に正しく答えられた場合だけドアを開けるシステムです。前回のなぞなぞ認証と同様、関係者には自明でも、外部の人間にはさっぱり分からないような問

写真 1 なぞなぞドア



近くから見たところ



題をたくさん用意しておけば、カードなどを使わずに簡単に認証できるというわけです。

なぞなぞドアの実例

写真 1 が、なぞなぞドアの全景です。ガラス扉の向こう側に iMac が置いてあり、iMac に接続されたテンキーがドアの外側に貼り付けてあります。

図 1 案内情報

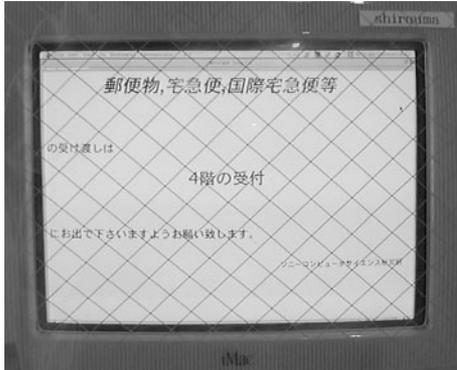
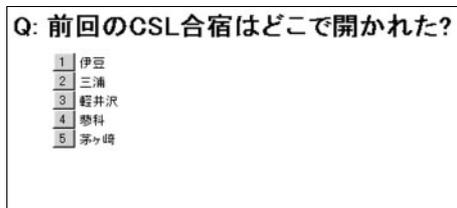


図 2 なぞなぞ問題の例



iMac のディスプレイには、ふだんは図 1 のような案内情報が表示されています。ここで、テンキーの ``*`` キーを押すと、図 2 のようななぞなぞが表示されます。

テンキーを使って答えると次の問題が出題され、すべてに正解するとドアが開きます。

なぞなぞドアの実装

なぞなぞドアを実装するには、なぞなぞの出題と解答を処理する装置と、ドアを開閉する装置が必要です。今回は、なぞなぞの出題と解答入力には、パーソナル・コンピュータの Web ブラウザとテンキーを使いました。ドアの開閉については、ネットワークに接続されたワンボード・マイコンでセキュリティ・システムの電気錠を操作する方法を利用しています。

なぞなぞの表示

なぞなぞの提示や入力された解答の処理をおこなうには、GUI ツールキットでプログラムをすることもできます。しかし、この程度の処理であれば、Netscape などの一般的な Web ブラウザで CGI と JavaScript を使うほうが簡単です。

Web ブラウザを利用すれば、メッセージや表などを

HTML で記述したり変更できます。さらに、別のソースから情報を取り込んで変換するのも容易なので、なぞなぞ認証をおこなっていない状態のとき、手軽に有用な情報を表示しておくことができます。

通常、ブラウザはマウスで操作しますが、このシステムではあらゆる操作にテンキーを利用します。JavaScript を使えば、画面の切替えや問題への解答がキーボードから簡単におこなえます。

タイムアウト処理

今回のなぞなぞドアでは、3 枚のページを用意して、オフィスの内線電話番号や入口情報などを順番に表示するようにしています。

HTML ソースのどこかに以下のような JavaScript コードを記述しておく、時間の経過とともに別のページにジャンプさせることができます。以下の例では、4 秒後に page2.html にジャンプします(誌面の都合上、⇒ で折り返しています。以下同様)

```
window.setTimeout('window.location.replace⇒  
("page2.html")',4000);
```

各ページでこのような指定をしておけば、スライドショーのように順番にページを表示していくことができます。

キー入力の処理

JavaScript では、各種のユーザー操作に対して関数を割り当て、操作にダイナミックに反応する Web ページが作れます。たとえばキー入力で CGI を起動させたい場合は、以下のようなプログラムを使えばいいでしょう。

```
document.onkeypress = keypress;  
function keypress(event)  
{  
  var i = keycode(event);  
  if(i == 0x2a){  
    window.location.href="door.cgi";  
  }  
}
```

この例では、``*`` キー (0x2a) を押すと、CGI プログラム door.cgi が起動されます。

keycode() は、入力されたキーの ASCII コードを返す関数ですが、Internet Explorer と Netscape では若干仕様が異なるため、イベント取得に関する場合分けをおこなっています。

```
function keycode(event)
{
  if(navigator.appName.indexOf("Microsoft") =>
    != -1)
    return window.event.keyCode;
  if(navigator.appName.indexOf("Netscape") =>
    != -1)
    return event.keyCode;
}
```

このようにすこし手間はかかるものの、JavaScript ではキー操作やタイムアウトで Web ページを移動させることができるので、Web ブラウザをプレゼンテーションなどに利用するときに重宝します。

なぞなぞの出題と解答処理

なぞなぞを出題する CGI プログラムは、前号で使用したものとほとんど同じです。ただし、すべての問題に正解した場合、Cookie を用いた認証の代わりに開錠のための処理をおこないます。このほかに、しばらく放っておくともとの案内画面に戻るようなタイムアウト処理も必要でしょう。なぞなぞの出題と解答を処理する door.cgi を末尾のリスト 1 に示します。

ドアの開閉

なぞなぞドアでは、なぞなぞにすべて正解した場合に CGI で錠を開けるための仕組みが必要です。

このところ、シリンダ錠の“ピッキング”被害が多発しているせいか、電気的に錠を開閉する“電気錠”が多くの会社から発売され、家庭用の製品もけっこう増えているようです。これを計算機から制御すればよいわけですが、さすがに計算機やネットワークに直結できる製品はほとんどありません。これを CGI で制御するには、ちょっとした工夫が必要になります。

スイッチの ON/OFF くらい、標準的なパーソナル・コンピュータで簡単に制御できてよさそうに思います。しかし、通信/印刷機器以外の外部装置を制御するための汎用入出力ポートは、どういうわけか普通のパーソナル・コンピュータには付いていません。特殊な装置を制御したいときは、シリアルポートやパラレルポートに細工したり、シリアル制御のワンチップ・マイコンを接続したりと、かなり面倒な作業が必要になるのが普通でした。たんにスイッチをいくつかつなぐためだけに、ワンチップ・マイコンを接続したりプログラミングしたりするのは、なんと

写真 2 PICNIC



く無駄に感じられたものです。ところが、最近になって、Ethernet に直結し、CGI による簡単な制御が可能なマイコンボードが発売されたため、このような処理も容易におこなえるようになりました。

PIC と PICNIC

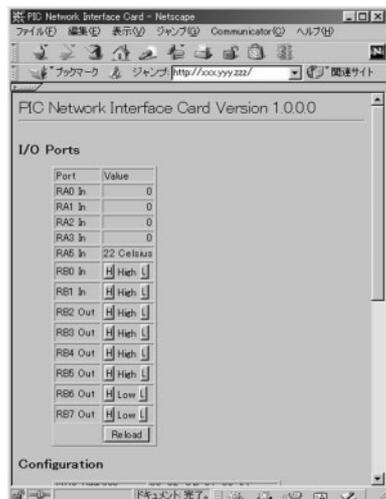
小さな機器に計算機を組み込んだり、あるいは非標準の入出力装置を使いたい場合、Microchip[1] の「PIC」という CPU がひろく使われています。PIC は 8bit の安価なマイクロ・プロセッサですが、昔の 8080 や Z80 などよりも高速で、入出力ポートや A/D コンバータ、シリアル通信などの機能を内蔵しているモデルもあり、組み込みシステムを簡便に作りたいときの定番 CPU になっています¹。PIC のキットや開発ソフトは、秋葉原の秋月電子通商 [2] や大阪・日本橋の共立電子産業 [3] で販売されています。

PIC は外部機器の制御に十分な機能をもっていますが、プログラミングはやはり面倒です。しかし、トライステート [4] から「PICNIC」というボード(写真 2)が発売され、状況が変わってきました。

PICNIC は、A/D コンバータやシリアル通信回路を内蔵した PIC16F877 チップとその周辺回路に加え、Ethernet を制御する NIC (Network Interface Controller) が実装されており、TCP/IP に対応しています。PICNIC のボードには、Ethernet ポートはもちろん、シリアルポート、アナログ入力ポート、デジタル入出力ポート、温度センサーがあり、これらをネットワーク経由で制御できます。細かい制御をおこなうには UDP を用いたプログラミングが必要ですが、簡単な HTTP サーバーや CGI が動くので、スイッチを単純に ON/OFF したり、温度

¹ 「子供の科学」の 7 月号で、なんと「PIC にハマる!」という特集が組まれていたのは驚きました。子どもがそんなものにハマっているのでしょうか?

図 3 Web ブラウザから PICNIC を覗く



などのアナログ値を調べたりする程度であれば、Web ブラウザから簡単に制御できます。

各種のセンサーやスイッチを PICNIC の A/D コンバータや入出力ポートにつなぎ、家庭内の LAN に接続するだけで、Web ブラウザなどからいろいろな制御がおこなえるようになります。

図 3 は、Web ブラウザから PICNIC にアクセスしたときの画面です。[H] または [L] のボタンを押すと PIC の出力ポートが変化し、PICNIC ボードの LED が点滅します。このように、PICNIC のサーバーの CGI を呼び出すだけで PIC の出力を制御できるわけです。

PICNIC からの電気錠制御

電気錠の本体に細工するのは、かなりの知識と経験がないと危険です。しかし、たいていの電気錠は電気的なスイッチ操作によって手動で開閉できる仕組みになっています。ですから、こちらを改造すれば、比較的安全に電気錠の開閉操作がおこなえます。PICNIC の出力で LED を点滅させる代わりにリレーを開閉し、電気錠の制御スイッチに直結すれば CGI による電気錠の開閉が可能になります。同様に、錠が開閉に関する情報を PICNIC で取得すれば、どこからでも錠などの状態をチェックすることができます。

これらの処理は、すべて PICNIC に標準で備わっている HTTP サーバで制御できます。したがって、PICNIC 自体のプログラミングをするまでもなく、CGI 要求

を Ethernet 経由で PICNIC ボードに送るだけで錠を開閉したり家の状態を調べたりすることができます。将来のセキュリティ・システムでは PICNIC のような装置が不可欠になるかもしれません。

運用実績

なぞなぞドアは、私が勤めている会社で試験的に運用しています。といっても、ふだんは案内板として意味のある情報を掲示しているだけです。来客があっても、なぞなぞで扉が開く仕組みになっていると言わないかぎり、たんに iMac を使った案内板があるな、としか思わないようです。

そこで、“じつは、なぞなぞを解くと扉が開くのだ”と説明すると、たいていの人はおもしろがって問題に挑戦しようとしています。当然のことながら、オフィスの事情に通じていないと解けないような問題にしてありますし、しかも、すぐにタイムアウトして消えてしまうので、ほとんどの場合は 3 問目くらいであきらめてあきらめてしまうようです。チャレンジしてもらうことを目指しているわけではないのでタイムアウトは短くしているのですが、時間がなくて解けないとよく文句を言われています。まあ、とりあえず笑いをとることは成功しているようですが……。

前回のなぞなぞ認証と同じく、自分にしか解けない問題を作るのは比較的簡単ですが、同僚には解けても外部の人には難解ななぞなぞを作るのはかなり難しいように思います。内部の人間しか知らない規定や社内旅行の際のエピソードを使うのが、一番安全かもしれません(下手をすると、このシステムの重要な目的の 1 つである“楽しい”認証という要素が欠けてしまうので、それなりに注意しなければなりません……)。

なぞなぞ問題の自動生成やインターフェイスの改良など、まだまだ課題はたくさん残っていますが、今後も運用を続けながら、なぞなぞ認証となぞなぞドアをさらに発展させていきたいと思っています。

(ますい・としゆき ソニー CSL)

[参考 URL]

- [1] Microchip (<http://www.microchip.com/>)
- [2] 秋月電子通商 (<http://www.akizuki.ne.jp/>)
- [3] 共立電子産業 (<http://www.kyohritsu.com/>)
- [4] トライステート (<http://www.tristate.ne.jp/>)

リスト 1 なぞなぞ CGI プログラム (door.cgi)

```
#!/usr/bin/perl
#
# $Date: 2001/06/12 18:38:03 $
# $Revision: 1.1 $
#
require 'cgiarg.pl';
require 'questions.pl';

$time = time;
&getcgiarg;

if($nq > 0){
    @qs = split(/,/, $qs);
    @as = split(/,/, $as);
    push(@as, $pa-1);
}

# $nask 解答後なら判定
$nask = 5 if !$nask;
&quiz if $nq < $nask;
for($i=0;$i<$nask;$i++){
    &fail if $as[$i] != $questions[$qs[$i]]->[1];
}
&success;

sub quiz { # 出題
    # まだ出題していない問題を選ぶ
    for (@qs){ $qs[$_] = 1; }
    srand(time);
    for(;;){
        $qno = int(rand($#questions+1));
        last if ! $qs[$qno];
    }
    push(@qs, $qno);
    $qs = join(' ', @qs);
    $as = join(' ', @as);
    $q = $questions[$qno]->[0];
    $a = $questions[$qno]->[2];
    $nq++;

    print <<EOF;
Content-type: text/html

<html>
<head><title> なぞなぞドア </title>
<SCRIPT LANGUAGE="JavaScript">
<!--
document.onkeypress = keypress;
function keypress(event)
{
    var i = keycode(event);
    if(i >= 0x31 && i <= 0x35){ # 1~5キーの処理
        document.forms[0].elements[i-0x31].click();
    }
}
function keycode(event)
{
    if(navigator.appName.indexOf("Microsoft") != -1)
        return window.event.keyCode;
    if(navigator.appName.indexOf("Netscape") != -1)
        return event.which;
}
}
```

• door.cgi から呼び出されるなぞなぞ問題 (questions.pl)

```
@questions = (
    ['樽酒で有名な飲み屋は?', 0,
     ['あさり', '権兵衛', 'でり坊', 'わに屋', '笑笑']],
    ['前回のCSL合宿はどこで開かれた?', 1,
     ['伊豆', '三浦', '軽井沢', '蓼科', '茅ヶ崎']],
    ['CSLのメールサーバのIPアドレスの4バイト目は?', 2,
     ['4', '34', '57', '85', '91']],
    ,
    .....
);
```

```

window.setTimeout('window.location.replace("page1.html")',8000);
/-->
</SCRIPT>
</head>
<body bgcolor=white>
<h1>Q: $q</h1>
<form method="post" action="door.cgi?time=$time">
<blockquote>
<input type=submit name=pa value=" 1 "> $a->[0]<br>
<input type=submit name=pa value=" 2 "> $a->[1]<br>
<input type=submit name=pa value=" 3 "> $a->[2]<br>
<input type=submit name=pa value=" 4 "> $a->[3]<br>
<input type=submit name=pa value=" 5 "> $a->[4]<br>
</blockquote>
<input type="hidden" name="nq" value="$nq">
<input type="hidden" name="qs" value="$qs">
<input type="hidden" name="as" value="$as">
</form>
</body>
</html>
EOF
    exit;
}

sub success { # 成功通知
    print <<EOF;
Content-type: text/html

<html>
<head><title>認証成功</title></head>
<body bgcolor=white>
<script language="JavaScript">
window.setTimeout('window.location.replace("page1.html")',4000);
</script>
ドアを開けます!!!!
</body>
</html>
EOF
    system "opendoor.cgi 3f > /dev/null"; # ドア開けCGI起動
    exit;
}

sub fail { # 失敗通知
    print <<EOF;
Content-type: text/html

<html>
<head>
<title>認証失敗!</title></head>

<body bgcolor=white>
<script language="JavaScript">
window.setTimeout('window.location.replace("page1.html")',4000);
</script>

認証に失敗しました!!
</body>
</html>
EOF
    exit;
}

```
