More CGI Programming

Multiple Submits
Cookies
Emailing
File Uploading

Deciding Where to Go

- What if you want to have more than one functionality on your form? In other words, have more than one button the user can push.
- We saw last week in the dump() example that the name and value of the submit button are passed as parameters.
- This is useful.

Multiple Submits

- Just as you can have many different text fields or checkboxes, you can have different submit buttons
- Make sure you give each submit a different name.
- Only the submit button that is pressed will be passed as a parameter.
- Check to see if this parameter exists.

```
<input type=submit name=Submit1 value="Go Here!">
<input type=submit name=Submit2 value="Go There!">
```

```
if (param('Submit1'){ … }
elsif (param('Submit2'){ … }
else{ … }
```

File Uploading

- Another input method we did not talk about last week is a file-upload field.
- To use file-uploading feature, must use a special kind of form.
  - Add `ENCTYPE="multipart/form-data"` to `<form>`
  - Or, use start_multipart_form() instead of start_form()
- `<input type='file' name='uploaded'>`
- `filefield(-name=>'uploaded')`
- Creates a field in which user can enter name of file to send to server. Also creates 'Browse' button to search local machine.
- User enters name or path of a file to upload.
- When form submitted, CGI script can then get this file

Getting the File

- To get the name of the file user wants to upload, use param() function.
- $file = param('uploaded');
- If you use $file as a string, it will be the name of the file. If you use $file as a filehandle, it will be a link to the actual file.

```
print "Contents of file $file are:<br>\n";
foreach $line <$file>{
  print "$line<br>";
}
```

That's Great for Text Files…

- But users can upload any kind of file.
- Need to find out what kind of file it was.
- uploadInfo() function. Returns reference to a hash containing info about the file.
- `$file = param('uploaded');`
- `$info = uploadInfo($file);`
- `$type = $info->{'Content-Type'};`
- $type may contain "text/html", "text/plain", "image/jpeg", etc etc…

If File is not Text

- Need function to read from binary files.
- read($filename, $buffer, $size)
  - $filename➔filehandle to read
  - $buffer➔scalar in which to store data
  - $size➔max number of bytes to read
  - returns number of bytes read

```
$file = param('uploaded');
open UPLOAD, ">binary.jpg";
while ($num=read($file,$buf,1024))
  { print UPLOAD $buf; }
close UPLOAD;
```

Emailing from your CGI Script

- In actuality, you can use this process to email from any Perl program.
  - I just feel like teaching it now.
- Note that this will be a Unix-specific (in fact, RPI CS dept – specific) lecture. There are ways to accomplish the same thing on Windows, but we're not going into it.

sendmail

- barebones emailing program. No friendly user interface whatsoever.
- standard with most Unix distributions.
- on RPI CS system, located in /usr/lib/
- We need to run it with the –t flag. This tells the program to search the message for the To:, Cc:, Bcc:, etc…
- For more information, `man sendmail`

Pipes

- You can open a 'pipe' to another program or process in almost the same way you open a file.
- A pipe is a connection between your program and another executable program. You can feed it input as though you were writing to the file
- Instead of <, >, or >>, use the | character in front of file name.
- `open (PROG, "|myprogram.exe") or die "Cannot open program";`
- For more information, CSCI-4210 & CSCI-4220

Put Them Together

```
open (MAIL, "|/usr/lib/sendmail –t") ||
  die "Cannot begin mail program";
print MAIL "From: lallip\@cs.rpi.edu\n";
print MAIL "To: president\@rpi.edu\n";
print MAIL "Subject: I want a raise!\n";
print MAIL "You know, Dr. J, I'm not quite
  sure this is really worth it. …\n";
close MAIL;
```

Cookies

- Love them or hate them, they exist. And you'll learn how to use them.
  - learning to use them responsibly is your own task.
- A cookie is a (usually very small) piece of text that a server sends to a web browser for later retrieval.
- Can be used to 'track' a user's preferences, or other information user has told the server.

To Set a Cookie

- Create the cookie
- cookie() function. Takes many (mostly optional) parameters:
  - -name=> Name of the cookie
  - -value=> Value of the cookie – can be a scalar, array reference, or hash reference
  - -expires=> Expiration date/time of the cookie
  - -path=> Path to which cookie will be returned
  - -domain=> Domain to which cookie will be returned
  - -secure=> 1 if cookie returned to SSL only

Cookie Expiration

- Expires: absolute or relative time for cookie to expire
  - +30s → in 30 seconds
  - +10m → in 10 minutes
  - +1h → in one hour
  - -d → yesterday (ASAP)
  - now → immediately
  - +3M → in 3 Months
  - +10y → in 10 Years
  - Wed, 05-Dec-2001 18:00:00 GMT → On Wednesday, 12/5/2001 at 6pm GMT.

Cookie Path

- 'region' of server to check before sending back the cookie.
- If I set a cookie with path = /perl/f01/
- Then only CGI scripts in /perl/f01 (and its subdirectories) will receive the cookie.
- By default, path is equal to the path of the current CGI script.
- To send cookie to all CGI scripts on server, specify path = /

Cookie Domain

- domain (or partial domain) to send cookie back to.
- must contain at least 2 periods (so can't send cookie to all .com domains)
- if I set cookie domain = .rpi.edu, cookie will be sent to scripts on www.rpi.edu, www.cs.rpi.edu, cgi.cs.rpi.edu, etc
- if set to .cs.rpi.edu, cookie only sent to www.cs.rpi.edu, cgi.cs.rpi.edu, cgi2.cs.rpi.edu, etc
- if set to www.cs.rpi.edu, cookie sent only to pages on www.cs.rpi.edu
- Note that both domain and path must match cookie parameters to be set.

Cookie Created, Now Set it.

```
$cookie = cookie( … );
print header(-cookie=>$cookie);
```

- To set more than one cookie, use array reference

```
$cookie1 = cookie (…);
$cookie2 = cookie (…);
print header(-cookie=>[$cookie1,
  $cookie2]);
```

Read the Cookies

- Once again, use the cookie() function.
- This time, don't use –value parameter. Just give the name
- `$mycookie = cookie('lallip');`
- $mycookie now has value of cookie with name lallip.