CSE 303:
Concepts and Tools for Software Development

Dan Grossman

Spring 2005

Lecture 27— HTML, CGI, Servers, ...

Homework 7

- An HTML file with a form
- A CGI program in C-Shell
- A Java program for "sanitizing" the CGI input
- Your homework-6 solution (an application written in C)

Why?

- A fun way to "put it all together"
- Expose you to new tools and interfaces (yet more practice quickly picking things up)
- Expose you to "software duct tape"

Today

- Demystify the World Wide Web
  - and remind you that 12 years ago you hadn't heard of it (because *it* not *you* were too young).
- See a basic example of client-server computing
- Understand some CGI basics
- Consider some security implications

Note: Other languages (e.g., Python) have much better support for easy and secure CGI programming. But we know C-Shell and it shows the "bare bones" approach.

The static web

Assuming an Internet, the web is jarringly simple:

- A browser *displays* HTML.
- An HTML *link* causes the browser to use HTTP (or another protocol) to *fetch* a page.
  - HTTP is just simple text: A `GET` request names the server and the file.
- A *web server* can be just an "ordinary" program running on an "ordinary" computer.
  - It tells the O/S it wants HTTP requests sent to the computer.
- The server sends back ordinary HTML text.

In the CSE department, your `www` subdirectory is "on the web" at `www.cs.washington.edu/homes/userid`

HTML in 2 minutes

A *markup* language (includes plain text and does not "execute").

You can view what the browser sees.

My homepage is a fine place to start because I write it by hand.

Most people don't write HTML by hand anymore; do you prefer the control of emacs or the GUIness of Word? (Many HTML editors give you both!)

Absolute basics:

- Open tags and close tags
- <, >, and & are special (so escape them)
- The homework uses `<pre>`

HTML is easy for humans and easy for machines (programs) to produce (and read)!

Security

- The web server cannot trust the browser.
  - It doesn't know there is a browser; it could be a "bad guy" sending HTTP requests.
- The browser cannot trust the web server.
  - It could be a site trying to send evil stuff to the desktop.
- Neither can trust the computers in-between on the Internet.
  - May passively spy or actively change requests/responses
- There are access protocols, but we won't go into them.
  - Passwords, SSL, etc.
  - File permissions based on domain names.

But for *static content* (text, pictures, links), only so much can go wrong.

Dynamic content

You may have noticed :) that often the browser sends data (e.g., queries) to the server that affects what HTML comes back.

There are fancy protocols for this; the simplest is CGI (Common Gateway Interface, which means "about the simplest way to do this").

- The web-site designer puts a *program* on the web.
- The browser tells the server to run the program giving it access to one string.
- Whatever the program prints to stdout the server sends to the browser.

There are actually two methods ("get" and "post") – we'll just do "get" (which is simpler but less powerful).

CGI (get) in more detail

- Put a program (written in any language) on the web:
  - End file with `.cgi` so the server treats requests for it as "run the program" not "return the contents"
  - "The string" is in an *environment variable* `QUERY_STRING`.
- Request running the CGI program giving it access to one string.
  - Request: the program as the filename, then `?`, then the string.
  - Many characters can't appear in the input string, so there's an encoding scheme (not important for hw7).
- The program's stdout gets sent to the browser.
  - Server runs the program in an environment where `QUERY_STRING` is set and stdout is piped over the net.
  - The 1st line *must* indicate the format (e.g., `Content-type: text/html`) and the 2nd line *must* be blank.

HTML Forms

We now know two ways to run CGI programs remotely:

- Type the URL with query string directly.
- Make a link that includes the URL with query string.

A more user-friendly way is to use HTML:

- the user fills out a form and clicks submit
- the browser makes an appropriate request for a CGI program

There are few *input types* a form can have (hw7 uses text boxes and radio buttons).

The HTML tags control what query-string gets generated. Basic format:

Using the query-string

`name1=value1&name2=value2&name3=value3`

The CGI program must convert this unwieldy string back into what it wants:

- Homework 7 uses a Java program to do that.
- **Security alert:** You cannot trust the query-string to be anything reasonable. Users can always access your CGI program directly!
  - Might be "evil" things like backquotes, dollar signs, ...

Libraries usually exist in various languages to get useful information about query strings.

Homework 7 does it quite manually.

Summary

The Web changed the developed world, but it's one of the least magical things we've learned about all quarter.

Dynamic content is scary if you're the web server:

- Please be careful with homework 7; strangers can test your program.
- You will need to transfer the files you need to `abstract`.