CSC 402 – Internet Technology

Recap

- Markup
- Markup Language
- HTML
- DOM
- XML
- HTML vs XML
- XHTML
- HTML vs XHTML

Authoring Tools

- WYSIWYG: **W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et.

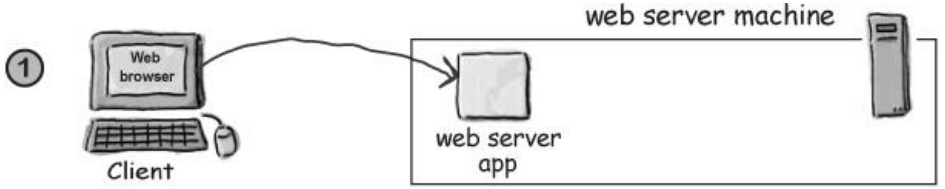
Helper Application

- Why is getting a web application framework so difficult?
  - The web was originally designed for static content.
  - HTML was/is more of an information markup system derived from Tex/Latex.
  - Applications tend to be dynamic and require state full information protocols.
    - HTTP, however, is a stateless protocol.
- Traditional web programming had features like:
  - Encode various state using HTTP cookies and/or other frameworks and languages, e.g. DHTML, JavaScript.
  - Use other languages and their data features to interface with data structures on the back end.
- Solution provided by traditional web programming:
  - Use of other languages on the server-side to provide controller logic (Perl, Java, PHP).
  - Using these languages to “print out” HTML output.
  - Using these languages “embedded” in web content to provide controlling logic to web content.

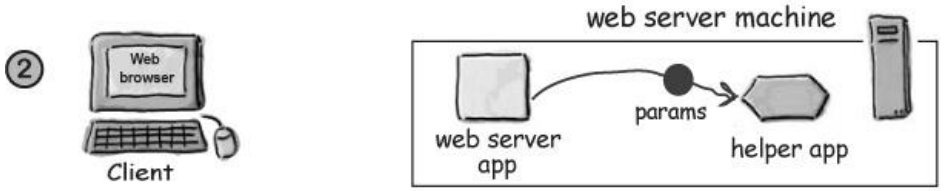
CGI

- Common Gateway Interface
- CGI describes how the HTTP Server communicates with an external program.
- CGI is widely implemented on almost all web-servers distributed for general consumption.
- CGI is termed as “scripts” because they were first written in UNIX shell scripts or Perl scripts.
  - However, CGI can be written in C, Java, C++, etc.
- Main function of CGI:
  - It can gather information sent from a web browser to a web server, and make the information available to an external program.
  - CGI can send the output of a program to a Web browser that request it.
  - Allows interactive web pages to be written. Page created dynamically, based on user request.
- Typically CGI scripts are stored in a directory called cgi-bin.

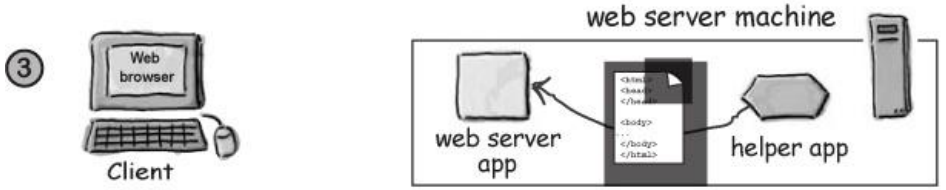
CGI – How It Works



User clicks a link that has a URL to a CGI program instead of a static page.

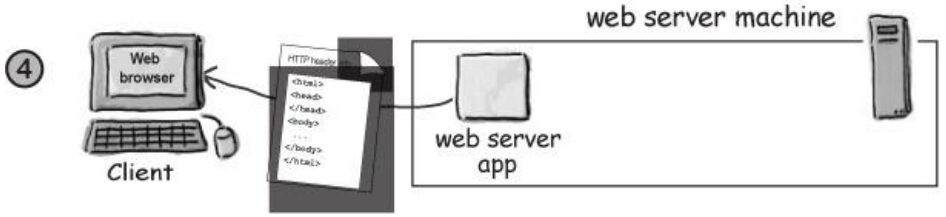


Web server application "sees" that the request is for a helper program, so the web server launches and runs the program. The web server app sends along any parameters from a GET or POST.



The helper app constructs the brand new page (that has the current date inserted) and sends the HTML back to the server.

As far as the web server is concerned, the HTML from the helper app is a static page.

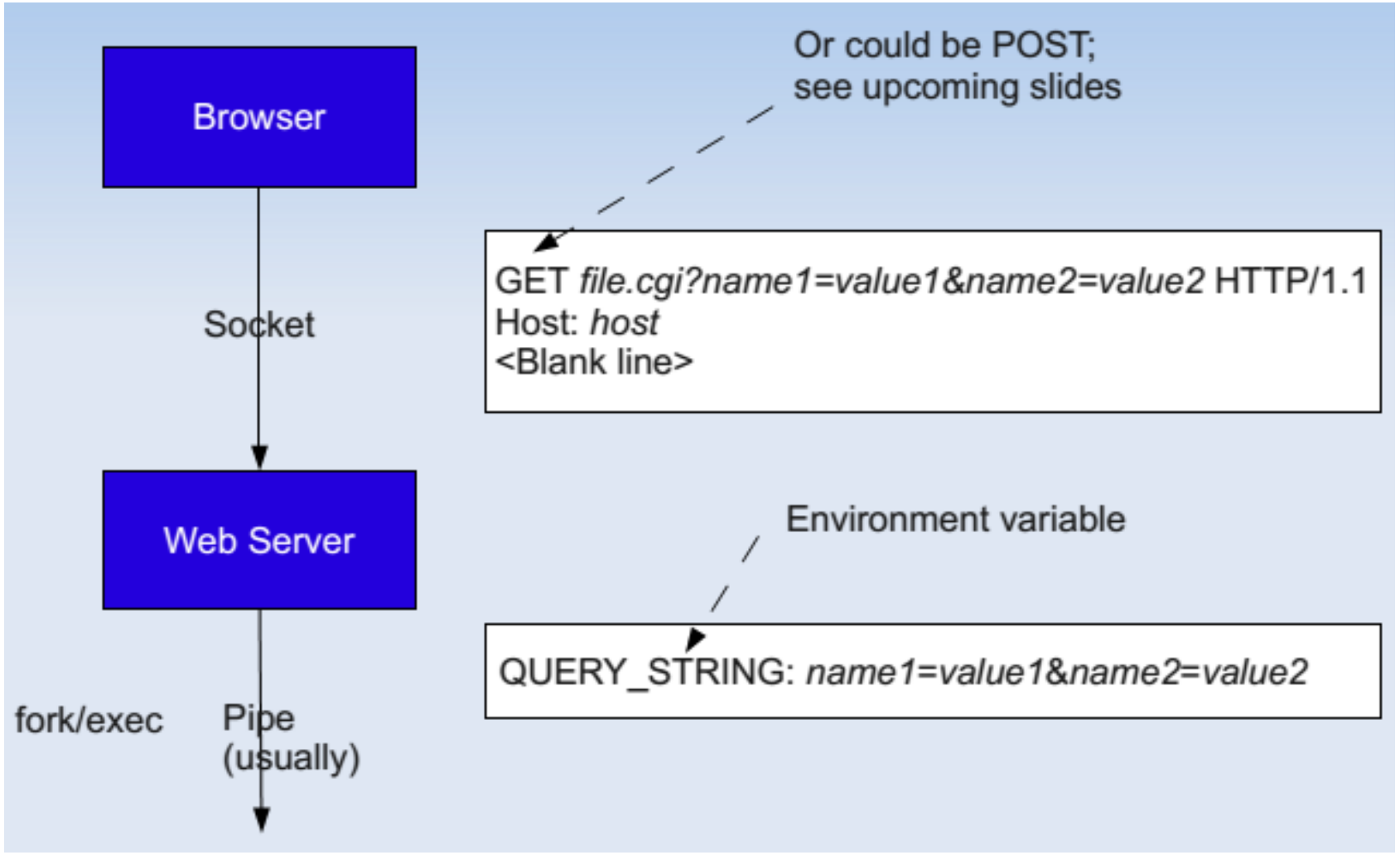


The helper application is shut down, and the client gets back an HTML page that has the current date as part of its now-static content.

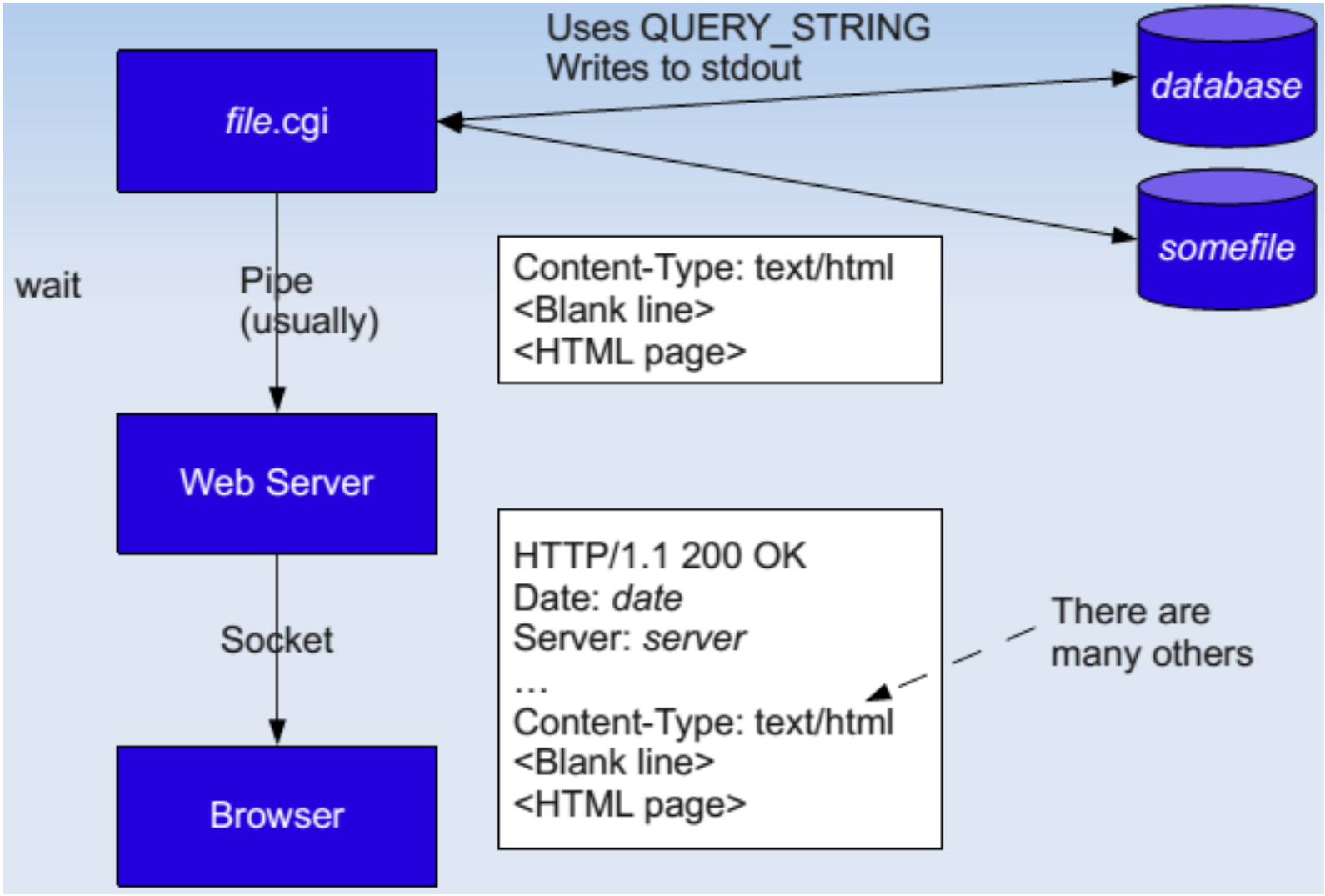
CGI – How It Works

- User clicks a link that has URL to a dynamic page instead of a static page.
- The URL decides which CGI program to execute.
- Web Servers run the CGI program in separate OS shell. The shell includes OS environment and the process to execute code of the CGI program.
- The CGI response is sent back to the Web Server, which wraps the response in an HTTP response and send it back to the web browser.
- HTTP/URL – revisited
  - Of type: protocol://host:port/file.cgi?name1=value1&name2=value2&...
  - Dynamic content: http://www.deerwalk.edu.np/file.cgi?name1=value1&name2=value2
  - Here, say file.cgi?first=deer&last=walk

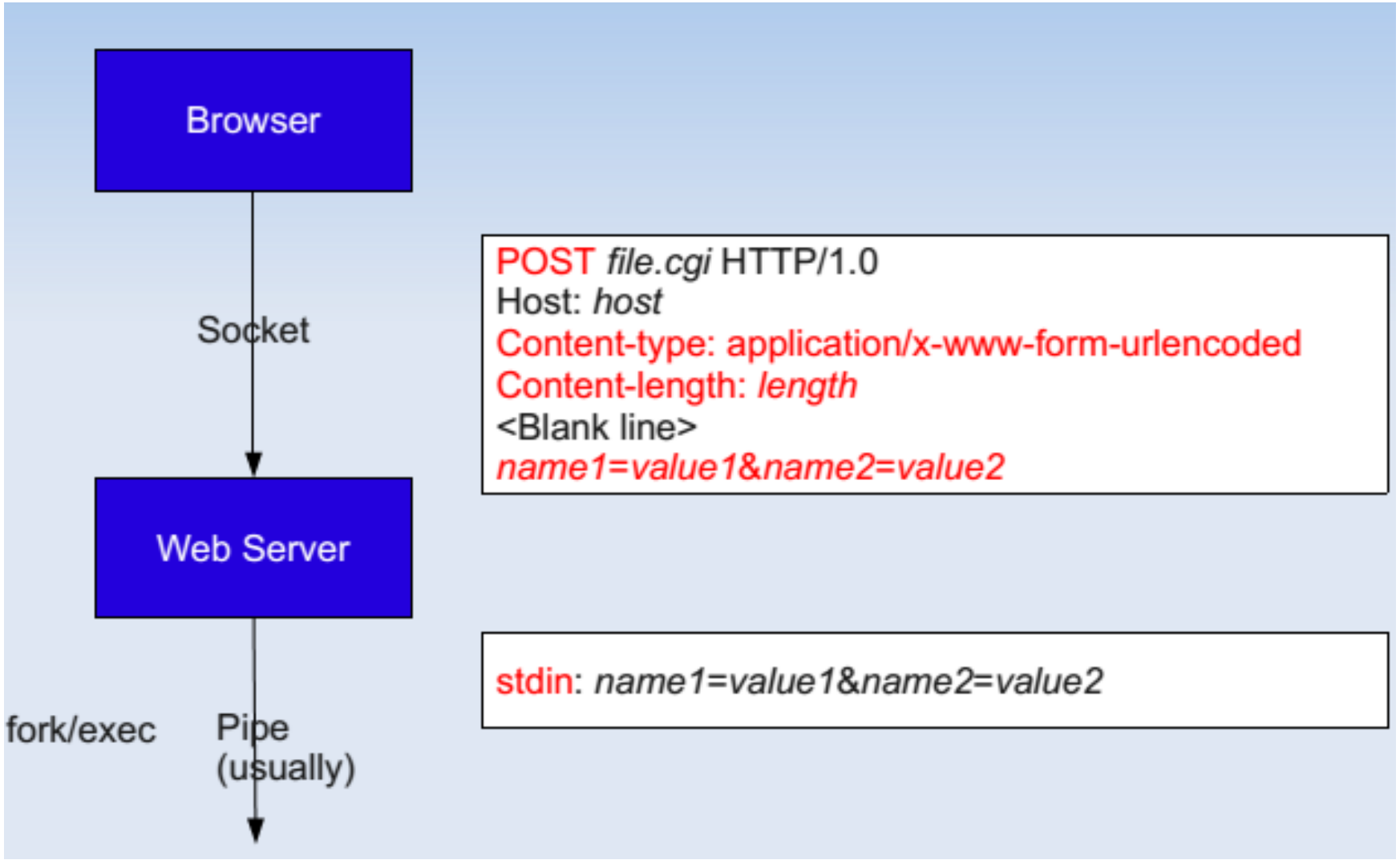
CGI – Get Method



CGI – Get Method



CGI – Post Method



CGI

- GET - The client appends data to the URL it passes to the server separated by amperstand.
  - Disadvantages:
    - Some browsers are limited ~ 1,000 chars
    - Visible in the address bar.
    - A user can manipulate the URL -- the GET portion.
    - Useful only for smaller forms.
- POST - The client sends data to the server by way of the HTTP message data field, thereby overcoming size limitations inherent to the GET method.
  - After a user clicks the Submit button on a form, the client browser URL Encodes user input in the same manner it does for GET.
  - The POST method uses the message body to send additional information from the user, rather than encoding it as part of the URL.
  - The data is sent in a data block to the server as part of the POST operation.

Servlet

- When a request comes in for a servlet, the server hands the request to the Web Container.
- Web Container is responsible for making the servlet visible or creating a new thread to handle the request.
- Its the job of Web Container to get the request and response to the servlet.
- The container creates multiple threads to process multiple requests to a single servlet.
- Servlets don't have a main() method.

CGI vs Servlets

- Disadvantages of CGI:
  - High response time because CGI programs execute in their own OS shell.
  - CGI is not scalable.
  - CGIs are mostly written in scripting languages like Perl so the extensibility is very poor in case of CGIs.
  - CGI programs are not always secure or object-oriented.
  - It is Platform dependent (usually run in UNIX environment).
- Advantages of Servlets:
  - Less response time because each request runs in a separate thread.
  - Servlets are scalable.
  - Servlets are just Java classes so it is easy maintain and extend their functionality. All the object oriented concepts can be directly applied to Servlets as well.
  - Servlets are robust and object oriented.
  - Servlets are platform independent.
  - Security
  - Servlets are running inside the sand box of JVM, so it is hard to damage the server side modules by malfunctioning the Servlets. Since CGIs are native applications, using CGIs a hacker can damage the server side components easily compared to Servlets.
  - Servlets provides easiest error handling mechanism jointly with web container.

Other Helper Application

- PERL
- JAVA
- JAVA SCRIPTS
- PHP
- ASP.NET
- Etc.