Understanding Client/Server Architecture

In this lesson you will learn about the Client/Server Architecture that is used as a basis for this course. The architecture of the Internet is based on the Client/Server model. We need to know how this architecture works in order to create web pages in this environment. You will be familiar with most of this material but be sure to think about how this environment will affect the logic of how you will develop processing for the users of your system.

Overview
To begin, think about a client and server as a system for communication. The client makes a request and the server sends back a response. This type of communication can be simple, such as a client entering a URL of a web site to receive back the information of a web page from the server at the URL’s address. Also the communication can be quite complex and may involve several back and forth request-response cycles. A good example of this would be an online shopping site.

The following diagram shows the main overview of the client/server architecture. Let’s break this down a little more.

![Diagram of Client/Server Architecture]

The Client
You might think of the client as a user (person) who is using their computer to connect through the Internet to receive information back. And you would be correct. Also you would be correct if you were to say the client is the computer that is has a communications device that is connected to the Internet. In fact you can also think of the client as several components working together including a person, a computer but also the software, such as a browser.

The user (person) is important from the stand point of design. We want web pages to be designed to allow for ease of use, easy to learn, accuracy, speed, and other considerations that are important to how a user makes requests.

In our course the software is a major component we will need to focus on since we are interested in writing web pages that will be displayed in the browser. The browser requires XHTML instructions to control how the web page displays the content for the user. Also CSS statements are added to present the content. You should have learned by now that XHTML is best used to control the content and CSS controls the design of the web page. We need to remember that when the client browser makes a request the response will come back in the form of a web page consisting of XHTML and CSS.
instructions. Other client side scripting instructions might also be returned in the response. JavaScript is a client-side scripting language.

The third component of a client is the computer. Typically you would think of the computer being a desktop or laptop computer. However client computers could also include several other devices including netbooks, touch pad devices, and smart phones. These devices all use different screen sizes and resolutions that affect how much can be displayed at one time on the browser.

The computer also controls the method of communication. Some different examples of communication include using high speed broadband, networks with Ethernet cables or even wireless.

**The Server**

While the client involved multiple components you might wonder if the server also is similar and you would be right. Some of the components for a server are the computer, software and additional components.

The computer/s on the server side will normally be of higher speed and larger storage capacity than the client computer. This is because the server will need to respond to multiple requests from several clients at one time. If the server isn’t large enough to handle several client requests at one time then the server
will bog down and perhaps lock up. There are times when some malicious user will be targeting for this vulnerability in a server. This is called a denial of service attack.

Software for a client is installed similarly to other software. It is important that this software is active at all times, waiting for a request from a client. The server software we will use for this class is Apache. Another popular server software is Microsoft IIS.

The client makes a request by specifying the server name or IP address in the URL. The URL also contains the folder location and the web page name on the server. The server software will control locating the resources needed to respond to the client's request. This could include various tasks such as generating web pages or handling database processing. To do such tasks, programs must be written. Programs on a server are different than traditional computer programs that are compiled and ran. Server programs are embedded inside of web pages and look like a combination of XHTML statements and server-side language based scripts. Instead of large complex programs, these programs are short but still have some complexity depending on the task. The language used in this class is PHP. Other server-side scripting languages include ASP.NET, Pearl and JSP. Other languages and techniques are also used to create software for the server, such as AJAX and Java servelets.

To handle the request from a client the server may need additional components. Some components could include hardware such as printers or cameras. The most common additional component is a database management system (DBMS). Common DBMSs are Oracle, Microsoft SQL Server, and Microsoft Access. We will use MySQL in this class for the DBMS.
Apache, MySql and PHP are all open source packages and very popular working together in a server configuration. There are several groups who package these components together for ease of use and ease of installation. We will be using the xAMP packages. The examples in the course material use the Windows version of xAMP called WAMP.

© Ray Ryon and Indian Hills Community College