NYIT

**Fall 2021**

Homework No: 11

Title: Web Servers

**Name:** Patade, Yash Arun

**Class ID#:**  11

**School ID#:** 1284979

**Course:** Java Networking

**Course ID:** CSCI 725

**Date:** 11/28/2021

Assignment Contents

[1) MIME. 3](#_Toc90770905)

[2) Format of MIME Messages. 3](#_Toc90770906)

[3) HTTP Protocol Status Messages. 3](#_Toc90770907)

[4) TTP/1.1 Improvements. 4](#_Toc90770908)

[5) HTTP/1.1 Connection Management. 4](#_Toc90770909)

[6) HTTP/1.1 Virtual Host. 5](#_Toc90770910)

[7) What HTML tag is used to access directly HTTP? 5](#_Toc90770911)

[8) CGI Script Output. 6](#_Toc90770912)

[9) What is “CGI”? 6](#_Toc90770913)

[10) Web Script. 6](#_Toc90770914)

[11) CGI Programs & Locations. 7](#_Toc90770915)

[12) CGI Programs & Security. 7](#_Toc90770916)

[13) What is script output below? 8](#_Toc90770917)

[14) CGI - Common Gateway Interface. 8](#_Toc90770918)

[15) CGI. 8](#_Toc90770919)

[16) CGI Four Basic Steps. 9](#_Toc90770920)

[17) CGI Enabled Sever – Data Input. 9](#_Toc90770921)

[18) Is our HTTP client a simple Web browser program? 10](#_Toc90770922)

[References. 11](#_Toc90770923)

# 1) MIME.

• MIME (Multi-Purpose Internet Mail Extensions) protocol.

• An extension of the original Internet e-mail protocol that helps HTTP exchange different types of data messages:

– Text (ASCII and other codes)

– Audio,

– Video,

– Images,

– Application programs,

– and other kinds.

# 2) Format of MIME Messages.

• A MIME message consists of a set of headers and one or more body parts.

• If there are multiple parts, one of the introductory headers consists of a string that will be used ark boundaries between parts.

# 3) HTTP Protocol Status Messages.

• HTTP protocol specifies a number of HTTP response codes.

Table

Description automatically generated

# 4) TTP/1.1 Improvements.

• Persistent Connections - reuses TCP/IP connection for several consecutive sessions.

• Request pipelining - reduces latency between requests and responses and delivers better perceived performance.

• Cache control. One of the biggest missing features in HTTP/1.0 is the absence of an explicit cache control mechanism. HTTP/ 1.1 introduces a variety of directives that can be used to control caching on proxies and in clients.

• Virtual Hosting - enables server identification by name via header directive.

# 5) HTTP/1.1 Connection Management.

• HTTP is a simple request-response protocol. In basic HTTP/1.0, each request is made over a new TCP/IP connection. After the data transfer is complete, the connection is torn down, and a new one is established to get another resource from the same server.

• The keep-alive (persistent connection) feature allows the same connection to remain open for multiple requests from the same client.

Graphical user interface, application

Description automatically generated

• In HTTP/1.1, the persistent connection feature became the default.

• The client and server must now explicitly specify if they do not want persistent connections by sending the header.

# 6) HTTP/1.1 Virtual Host.

• The Host: request header was added to the HTTP protocol as an extension to HTTP/1.0. This header carries the hostname used to yield the IP address.

A picture containing text

Description automatically generated

# 7) What HTML tag is used to access directly HTTP?

Answer)

• Anchor tag <A>

• Example: To enable user of the original rendered/displayed HTML page pageOld.html to just click and activate HTTP session with the HTTP server and download new resource pageNew.html on the Web server www in the DNS domain compX.xom it is enough to use in pageOld.html HTML code text:

A picture containing text

Description automatically generated

# 8) CGI Script Output.

• Just as CGI scripts take ‘input’ from the user via a web browser, it returns output back to the user via the web browser as well.

• Anything printed to STDOUT is sent back to the web browser that contacted the script.

• This means you don’t have to learn any new output functions. print() will now throw data to the web browser.

# 9) What is “CGI”?

• Common Gateway Interface (Not Computer Graphics Imaging!)

• A means of running an executable remote program using the Web

– Typically used to generate a dynamic webpage

• Almost any language can be used to develop CGI programs

• Unix shell language is the easiest to use and Perl is the most productive (text string processing oriented)

# 10) Web Script.

• Script: Interpreted Source Language Program that is executed by the interpreter instruction-by-instruction.

• Web Client script: Program embedded in HTML or XML document that adds (programmatic 3GL code) dynamism to a displayed page.

• Web Server script: External source language program that server runs via interpreter in response to client’s request. Output produced by server scripts is returned to the Web client.

– Equivalent to enabling client to provide I/O and run a program on the remote server host, (Remote program execution.

# 11) CGI Programs & Locations.

• To use a CGI program simply place it in a configured directory in web directory structure and have URL point to it.

– CGI program may be written in any language that is allowed to be executed on the system.

– CGI programs written in C/C++ will execute faster than scripts, and CGI programs that are very large and complex may be written in Java.

– CGI programs that are small and simple may be written as scripts in one of the scripting languages, e.g., UNIX-shell, Perl or TCL.

• When using a scripting language such as PERL or TCL:

– The script program itself must reside in the configured CGI directory.

– Interpreter program must reside in one of the directories of the system PATH.

# 12) CGI Programs & Security.

• CGI program must know (expect, have pre-programmed) parameter names beforehand.

• There is a tight-dependence/tight-coupling between the client form program and server side CGI script program.

• Great security feature: Arbitrary queries are not allowed. – Only the pre-programmed form-defined requests are allowed.

# 13) What is script output below?

Text

Description automatically generated with medium confidence

Answer)

String of text produced by the echo command, that could be a web page.

# 14) CGI - Common Gateway Interface.

• CGI is a standard/common mechanism/technology for:

– Associating URLs with programs that can be run by a web server.

• CGI is a protocol (set of rules) specifying:

– How the request is used to initiate execution of the external program behind the Web server,

– How the request parameters are passed from the HTTP server to the external program, and

– How the external program sends the response to the HTTP server.

# 15) CGI.

• Standard for interfacing Web application server program with the HTTP server.

• CGI script can be written in any language that can:

– read STDIN “file”,

– write to STDOUT “file”, and

– read environment variables (echo $VARX)

# 16) CGI Four Basic Steps.

1. Read the user’s form input. Data can be sent by GET or POST

–GET: stores data in QUERY\_STRING environment variable

–POST: client sends Content-Length heading HTTP variable, blank line and then data.

• data read from STDIN and is more general-purpose

2. Output HTTP headers.

– CGI program mimics HTTP server

– Usually CGI program generates an incomplete set of HTTP headers and lets server fill in the rest

– CGI program must generate Content-Type

–\* May also generate Date, Content-Length, Server, etc.

3. Send a blank line.

4. Generate a document.

– Frequently HTML.

# 17) CGI Enabled Sever – Data Input.

• CGI enabled server has built in CGI software module that can perform the following tasks:

– It can read (parse) HTTP data and Client generated URL text string and extract text elements equivalent to the input parameter values of the named CGI program.

– CGI web server module can present HTTP client sent data as:

• Shell or environment variables to the CGI program, or

• Keyboard typed input, i.e., stdin file content.

– CGI module starts execution of the CGI program

• CGI enabled server has built in CGI software module that can perform the following output tasks:

– It can accept CGI program standard output and pass it to the HTTP server for Response message formatting and delivery

– Anything written by the CGI program to stdout device file is forwarded by the web server to the client.

# 18) Is our HTTP client a simple Web browser program?

Answer)

• No

– It is too simple (trivial)

– It can download only one page per test-run.

– It cannot replace current page with the new pointed-at page.

# References.

[1] [Index of /02\_725/11\_Web\_Server (tfbor.com)](http://tfbor.com/02_725/11_Web_Server/)

[2] [Microsoft PowerPoint - 725\_10\_Lecture\_01\_GetWebPage\_.pptx (tfbor.com)](http://tfbor.com/02_725/11_Web_Server/725_11_Lecture_01_GetWebPage__.pdf)

[3] [Microsoft PowerPoint - \_725\_10\_Lecture\_03\_WebAppScrpt\_.pptx (tfbor.com)](http://tfbor.com/02_725/11_Web_Server/725_11_Lecture_03_WebAppScrpt_.pdf)