Practical 21

Aim: Introduction to PERL and CGI programming.

What is CGI?

- CGI is an acronym that stands for ***Common Gateway Interface*** is a standard for interfacing external applications with information servers, such as HTTP or Web servers.
- This interface provides a means for browsers and the server where document resides to communicate and pass information back and forth.
- Primarily, this is done through the <FORM> tag, but there can be other ways to use CGI effectively, like through Server Side Includes (SSI).
- CGI, permits interactivity between a client and a host operating system through the World Wide Web via the Hyper Text Transfer Protocol (HTTP).
- CGI program can be written in C or C++, Perl, ASP, PHP, Python, TCL, shells, and many others languages and scripts.

Examples of uses for CGI

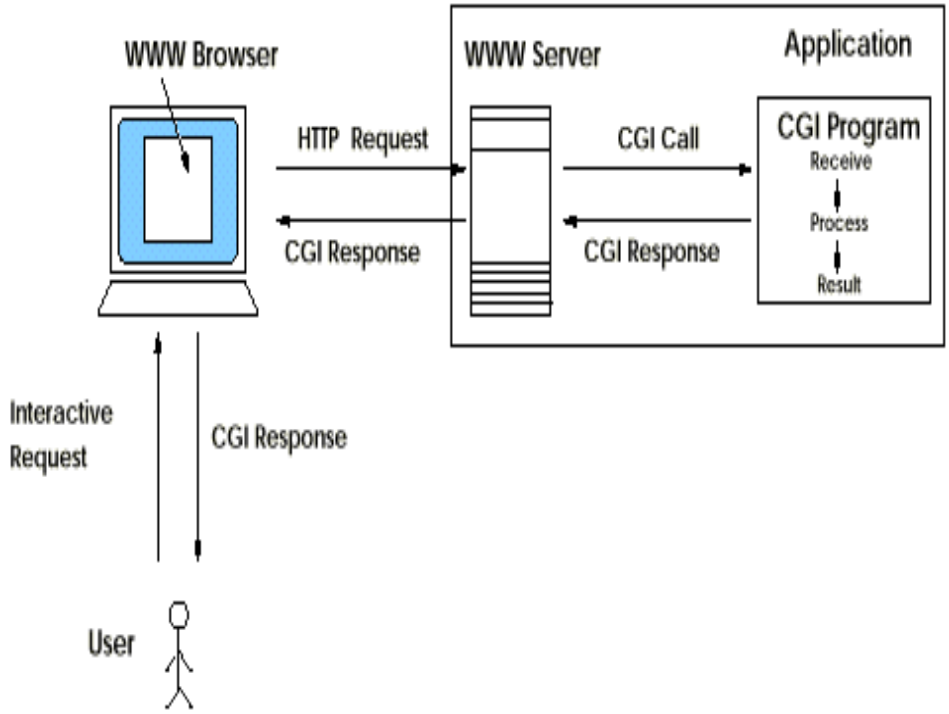
- Forms
  - forms on web sites allow the user to enter information which is processed by CGI and mailed to an administrator or logged
- On-the-Fly Pages
  - web pages can be created dynamically (as needed) with up-to-date information.
- Database Interaction
  - an application of on-the-fly page creation. Web pages can be created using information read from a database, or a web site form can allow a user to update database entries
- Logging / Counters
  - a log file can record traffic data updated with information on each visitor. A counter can be included on the web page to advertise traffic.
- Animation
  - "server-push" programs can be used to feed the client successive images in an animated sequence.
- Catalogs, Search engines

Requirements

- Web server (Apache, IIS, Microsoft Personal Web server etc.)
- Compiler (C/C++) or Interpreter (Perl), PHP, ASP
- Web browser (Mozila, IE etc.)

Writing CGI programs involves

- Obtaining input from a user or from a data file.
- Storing that input in program variables.
- Manipulating those variables to achieve some desired purpose, and
- Sending the results to a file or video display.



CGI Programming in PERL

- PERL stands for *Practical Extraction Report Language*.
- PERL
  - Is an interpreted language.
  - Runs on multiple platforms: Windows, Unix, Mac, …, etc.
  - Is a scripting language.
  - Is a type less language.
  - Has some aspects similar to C language.
  - Could be as procedural as you want it to be.
  - Could be as object-oriented as you want it to be (PERL 5).

Introduction to PERL

- PERL can be downloaded from:
- http://www.perl.com/pub/a/language/info/software.html
- To run PERL programs:
  - On UNIX, type the command from UNIX shell:
- *perl perl_prog.pl*
  - On Windows, type the command from DOS prompt:
- *perl perl_prog.pl*
- PERL is a case-sensitive language just like C or Java.
- "#"sign is used for comments in PERL

PERL Data Types

- PERL has three built-in data types: scalars, arrays of scalars, and associative arrays of scalars, known as "hashes".
- Scalar Variables
    - A scalar may contain one single value in any of three different flavors: a number, a string, or a reference.
    - Scalar values are always named with '$‘ at the beginning, even when referring to a scalar that is part of an array or a hash.
    - ***Examples***:
- **$day #A simple scalar value "day"**
- **$day[28] #the 29th element of @day**
- **$day{'Feb'} #The 'Feb' value from %day**

- Associative Arrays “Hashes”
    - Associative arrays are created with a set of key/value pairs. A key is a text string of your choice that will help you remember the value later.
    - A hash name begins with % sign.

      Examples:
        - %hashName = ('key1', 'value1', 'key2', 'value2');
        - %ourFriends = ('best', 'Don', 'good', 'Robert', 'worst', 'Joe');
    - To access an element, use $+hash name+{+key+}.

      Examples:
        - $hashName{‘key1’} #This will return value1
        - $ourFriends{'good'} #This will return #‘Robert’