

웹 프로그래밍에 관한 이해와 구현

김 종 훈*

< 목 차 >

- I. 서 론
- II. 웹 프로그래밍 언어
- III. C언어로 구현한 CGI 프로그램
- IV. 결 론
- ※ 참고문헌

I. 서 론

웹은 인터넷상에 광범위하게 분산된 다양한 형식의 데이터를 손쉽게 검색할 수 있으며, 모든 응용의 사용자 환경을 손쉽게 포함시킴으로써 새로운 데이터의 통합을 위한 가장 성공적인 기술로 평가받고 있다.

이 웹 상에 문서를 기술하는 방법은 HTML(Hyper Text Markup Language) 이었다. HTML은 누구나 사용할 수 있을 만큼 간단하며, 보편적이며, 특별한 데이터 타입이 사용되지 않고 단순한 텍스트이기 때문에 이식성과 사용이 편리하다. 또한 HTML은 SGML(Standard Generalized Markup Language)의 한 응용으로써, 이미 표준으로 정해진 DTD(Document Type Definition)에 따라 사용자는 그 인스턴트만 작성하면 된다. 즉, 사용할 수 있는 모든 태그(tag)는 전부 HTML의 DTD에 정의되어 있으므로 사용자는 정의된 태그만을 사용하기만 하면 웹 브라우저에서 볼 수 있다.

* 제주교육대학교 컴퓨터교육과 전임강사

그러나, HTML은 이러한 여러 가지 장점에도 불구하고 HTML만으로는 대화적인 웹사이트를 구축하기에는 한계에 와 있어 여러 웹 프로그래밍 기술이 생겨나고 있다.

이에 본 논문에서는 단순히 동적인 느낌만을 주는 웹사이트를 만들기 위한 것이 아닌 클라이언트와 서버간에 상호작용을 할 수 있는 대화적인 웹사이트를 구축할 수 있는 웹 프로그래밍 기술에 대해 다양하게 살펴보고 특히 III장에서는 C언어로 CGI 프로그램을 구현해 본다.

II. 웹 프로그래밍 언어

1. PHP

1.1 PHP의 개요

웹 언어로 일반적으로 사용되는 HTML이나 자바 스크립트와 같은 언어들은 클라이언트의 웹브라우저에서 해석되는 스크립트 언어이다[1]. 이러한 언어만으로는 서버와의 대화가 필요한 동적인 웹사이트 구축은 힘들다. 동적인 웹사이트를 구축하기 위해서는 C와 Perl과 같은 언어를 이용하여 구현한 CGI 프로그램을 이용해야 하는데 이러한 CGI는 프로그래밍하기가 까다로운 단점이 있다. 그러나 PHP는 HTML에 내장되어 동작하는 서버 차원 스크립트 언어로 프로그래밍하기가 수월한 언어다. PHP는 C와 Java, Perl 등으로부터 많은 형식을 따르고 있고, 동적인 웹 문서를 빠르고 쉽게 작성할 수 있도록 해준다[2].

PHP는 Oracle과 Sybase, Informix와 같은 대규모 데이터베이스에서부터 mSQL, MySQL, PostgreSQL과 같은 소규모의 데이터베이스에 이르기까지 다양한 데이터베이스와의 연동이 쉬우며 유닉스, 리눅스 환경과 윈도우 환경에서 동작하는 대부분의 웹서버를 지원한다.

1.2 PHP의 기능

본 절에서는 PHP가 지원하는 기능들에 대해 설명한다[3].

(1) 오류처리

PHP는 <표 1>과 같이 4가지의 오류와 경고가 있다. 오류처리 단계는 <표 1>의 4개의 숫자가 더해져서 결정된다. 기본 오류에 대한 보고는 7로 1+2+4가 되고, 이는 Notice를 제외한 모든 오류를 의미하는 것이다.

<표 1> PHP에서 오류와 경고 형태

1	Normal Function Errors
2	Normal Warnings
4	Parser Errors
8	Notices (무시할 수 있는 경고나 버그의 가능성을 가지고 있는 경우)

(2) HTTP 인증

HTTP 인증이란 클라이언트로부터 ID와 패스워드를 입력받아, 그 ID와 패스워드로 웹 문서에 접근을 허가하거나, 불허하는 기능을 말한다. PHP를 사용한 HTTP 인증은 아파치 모듈로 사용될 때만 동작하며, 아파치 모듈의 오류처리 단계는 PHP 스크립트에서 Header()라는 함수를 사용하면 클라이언트 브라우저에게 사용자 ID와 암호를 입력받는 윈도우를 띄우도록 하는 "인증 요구" 메시지를 보내게 된다. 일단 사용자가 사용자 ID와 패스워드를 입력하면, 사용자 ID, 이름, 패스워드, 인증 형태의 값을 가지는 \$PHP_AUTH_USER, \$PHP_AUTH_PW, \$PHP_AUTH_TYPE의 3개 변수를 가지고 해당 URL이 다시 불리워진다. 현재 인증 형태는 "Basic"만이 지원된다. 또한 이 인증 기능은 MS사의 IIS 서버에서 CGI 버전의 PHP를 사용할 경우에는 IIS의 제약으로 인해 동작하지 않는다.

(3) HTTP 쿠키 지원

PHP는 HTTP 쿠키를 명확하게 지원한다. 쿠키 메카니즘은 트래킹이나 사용자 식별 등을 위해 원격 브라우저에 저장된 데이터를 돌려 받는 과정을 말한다. 쿠키를 설정하기 위해 데이터를 브라우저로 보내는 setcookie() 함수를 사용한다. 쿠키는 HTTP 헤더의 한 부분이므로, SetCookie() 함수는 브라우저로 보내는 어떤 데이터보다도 앞에 사용해야 하며, 동일한 쿠키에 여러 값을 저장하고 싶다면

쿠키 이름에 대괄호를 더하면 된다.

(4) 파일 업로드 지원

PHP는 RFC-1867을 지원하는 브라우저로부터 파일을 업로드 받을 수 있는 기능이 있다. 이 기능을 사용하면 텍스트뿐 아니라 이진 파일도 업로드가 가능하다.

업로드가 성공되면 지정된 파일에는 다음과 같은 변수들이 정의되어진다.

- \$userfile : 업로드된 파일 내용이 저장되어 있는 서버의 임시 파일명
- \$userfile_name : 업로드한 시스템에서 사용하는 파일의 원래 이름
- \$userfile_size : 업로드된 파일의 크기(바이트 단위)
- \$userfile_type : 만약 브라우저가 업로드된 파일의 MIME 형식을 안다면, 그 MIME 형식

또한 한 번에 여러 개의 파일을 동시에 전송하는 것도 가능하다. 이때 PHP는 이 파일들에 대한 정보를 배열로 전달한다. 따라서 이런 경우에는 여러 개를 선택하는 select나 checkbox처럼 HTML 폼의 아이템에 동일한 배열명을 적어주어야 한다.

(5) 연결 제어

내부적으로 PHP는 연결 상태를 <표 2>에서처럼 3가지 중 하나로 설정한다.

<표 2> PHP의 연결 상태

0	Normal
1	Aborted
2	Timeout

PHP 스크립트가 일반적으로 실행되고 있을 때는 Normal 상태가 된다. 만약 원격 클라이언트가 연결을 끊게 되면 Aborted 상태가 되며, PHP가 설정한 시간 제한에 걸리게 되면 Timeout 상태가 된다.

(6) 영구적인 데이터베이스 접속

영구적인 접속이란 스크립트가 종료되어도 SQL 링크를 닫지 않는 접속을 의미한다. 영구적인 접속이 요청되면, PHP는 기존의 접속이 있는가를 검사한다. 만약 있다면 기존의 것을 사용하고, 없다면 새로운 링크를 만든다. 영구적인 접속은 비영구적인 접속에 비해 어떠한 기능적인 향상도 주지는 못하지만, 효율성에서는 이점이 있다. 영구적인 접속은 SQL 서버와의 링크를 만드는 것에 따른 오버헤드가 클 경우에 유용하다. 이 오버헤드 요인은 데이터베이스의 종류, 데이터베이스 서버의 위치 등에 따라 크게 달라진다. 접속에 따른 오버헤드가 클 경우에 영구적인 접속이 유용하다. 이는 자식 프로세스에서 SQL 서버에 접속을 요청할 때마다 접속을 만드는 대신, 이 프로세스가 종료될 때까지 살아있는 한 개의 접속만을 사용한다. 이것은 모든 영구적인 접속을 사용한 프로세스는 그에 해당하는 한 대의 영구적인 접속을 가지고 있다는 것을 의미한다. 영구적인 접속은 일반적인 접속에 대해 1:1로 대응되도록 설계되었다. 이것은 스크립트의 기능은 그대로 두고, 언제라도 비영구적인 연결을 영구적인 연결로 배치할 수 있다는 것을 의미하는데 이로써 효율의 향상을 가져올 수 있다.

1.3 PHP 문법

본 절에서는 PHP3의 기본적인 문법을 살펴본다[4].

(1) Variable

- 변수는 \$로 시작한다.
- 변수는 별도로 선언하는 부분이 없으며 (type을 미리 지정하지 않으며) 변수 type은 변수가 사용된 context에 따라 결정된다.
- 변수의 type으로는 기본적으로 integer, double, string type이 있으며 이외에 array와 object type이 있다. 변수 type은 `gettype()`, `is_long()`, `is_double()`, `is_string()`, `is_array()`, `is_object()` 등의 함수로 알아낼 수 있다.
- type casting은 C에서와 같은 형식으로 이루어진다. (int) (integer) (real) (double) (float) (string) (array) (object) 등의 casting operator가 있다.

- string conversion : 문자열이 숫자로 변환될 때에 문자열이 '.', 'e', 'E'를 포함하고 있으면 double로 그렇지 않으면 integer로 변환이 된다.
- variable variable : 변수의 값이 변수의 이름이 되는 것을 말한다. 예를 들어 \$a = "hello" 일 때 \$\$a = "world"라고 정의하면 \$hello = "world"로 정의되는 것이다.
- scope : user-defined function에서는 local function scope가 적용된다. 즉 function 안에서 정의된 변수는 function안에서만 의미가 있다. 주의할 점은 외부에서 정의된 변수도 user-defined function안에서는 의미가 없다는 것이다. 외부에서 정의된 변수를 사용하려면 함수 안에서 global \$externalVar: 형식으로 정의를 하고 사용하거나 \$GLOBALS ["externalVar"] 형식으로 직접 변수를 사용하여야 한다.
- static variable : local function scope를 가지지만 scope를 벗어나더라도 값이 유지되는 변수를 말한다.
- Array
 - array는 \$array[] 형식으로 사용된다.
 - scalar array : 첨자가 숫자로 주어지는 배열이다. 예를 들면 \$array[0] = 1;
 - associative array : 첨자가 숫자가 아니라 문자열로 주어지는 배열이다. 예를 들면 \$array["first"] = 1;
 - 그냥 \$array[]에 값을 지정하면 array에 값이 하나 추가된다. \$array[] = 1; \$array[] = 2; 는 \$array[0] = 1; \$array[1] = 2;와 같은 의미를 지닌다.
 - array() 함수를 통하여 array를 만들 수도 있고, count() 함수로 element의 개수를 얻을 수도 있다. next(), prev() 함수나 each() 함수를 통하여 element들을 참조할 수도 있다.
- External Variables
 - HTML form (GET/POST) : form으로 전달된 값은 form에서 name field로 지정한 이름의 변수로 지정이 된다. form에서 type = "image"인

경우에는 마우스로 클릭 한 위치가 name_x, name_y 형태로 추가로 전달된다.

- HTTP Cookie : browser로 cookie를 전달하려면 SetCookie(name, value, timeout) 함수를 이용한다. client로부터 전달되는 cookie는 PHP 변수로 변환이 된다.
- Environment Variable : 환경변수 또한 PHP 변수처럼 사용할 수 있다. 환경변수는 getenv()라는 함수를 이용해 얻을 수 있으며 putenv() 함수로 환경변수를 지정할 수도 있다.

(2) Class

- class keyword를 사용하여 class를 정의하며 instance 생성은 new operator를 이용한다.
- class의 상속은 extends keyword를 사용한다.
- constructor 정의는 class와 같은 이름을 갖는 상수를 정의함으로써 이루어진다. constructor는 default parameter를 가질 수 있다.

예)

```
class testClass {
    var $value;
    function testClass($defValue = "test") { $value = $defValue; }
    function doSet($setValue) { $value = $setValue; }
    function doPrint() { echo $value; }
}

class testLineClass extends testClass {
    function doPrintLine() { echo("$value\n"); }
}

$test = new testClassLine;
$test->doSet("hello");
$test->doPrint();
$test->doPrintLine();
```

(3) Expression

- expression이란 값으로 환산되는 것을 말한다.
- assignment는 expression이므로 \$a = \$b = 5; 같은 문장을 쓸 수 있다.
- pre and post increment / decrement : \$a++, ++\$a, \$a--, --\$a
- comparison operator는 boolean 값을 갖는 expression이다.
- operator와 assignment의 결합 : \$a += 3; \$b = ++\$a; \$c = double(--\$b); \$d = \$c += 10;
- boolean : 숫자에서 0은 false, 0이 아닌 값은 true이다. 문자열에서 ""와 "0"은 false, 나머지는 true이다. array에서는 element가 하나도 없으면 false, 하나라도 있으면 true이다.

(4) Statement

〈표 3〉에 나타나 있다.

〈표 3〉 PHP의 statement

if 문	<pre>if (expression) { do anything 1; } elseif (expression) { do anything 2; } else do anything 3;</pre> <p>또는</p>	while 문	<pre>while (expression) do anything;</pre> <p>또는</p> <pre>while (expression) : do anything; endwhile;</pre>
	<pre>if (expression) : do anything 1; elseif (expression) : do anything 2; else : do anything 3; endif;</pre>	switch 문	<pre>switch (expression) { case ... : do anything; break; default : do anything; }</pre>
do... while 문	<pre>do { do anything; } while (expression);</pre>	for 문	<pre>for (expr1; expr2; expr3) do anything;</pre>

(5) Function

- 함수는 function 이라는 keyword를 가지고 정의가 되며 별도로 return type은 지정하지 않는다.
- function의 기본형은 다음과 같다.

```
function funcName($arg1, $arg2, $arg3, ..., $argn) {
do anything;
return $retval;
}
```
- return value는 list와 object를 포함하여 어떤 type이든 될 수 있다. 예를 들어 array를 return하려면 return array(0, 1, 2);
- argument는 default로 pass by value이다. pass by reference로 하려면 argument definition에서 변수명 앞에 &를 붙여주면 된다 (function funcName(&\$arg1)). function이 pass by value로 정의가 되었다더라도 함수를 부를 때 argument에 &를 붙여서 넘기면 pass by reference가 된다 (call : doFunc(&\$var)).
- default parameter : C++에서 사용하는 방식으로 default parameter를 정의할 수 있다 : function funcName(\$var = 1) { }

(6) Operator

- operator들은 C언어에서의 operator와 비슷하며 다음과 같은 operator들이 있다.
 - arithmetic operator : +, -, *, /, %
 - string operator : . (concatenation)
 - assignment operator : =
 - bitwise operator : &, |, ~
 - logical operator : and (&&), or (||), xor, !
 - comparison operator : ==, !=, <, >, <=, >=

(7) 기 타

- require : #include와 똑같은 의미로 사용된다.
- include : include 문장을 만날 때마다 지정한 파일을 포함한다. require는 #include처럼 무조건 파일을 포함시키지만 include는 loop나 if 문 등에서 사용할 수 있으며 필요한 경우에만 파일을 포함하도록 할 수 있다.

2. CGI

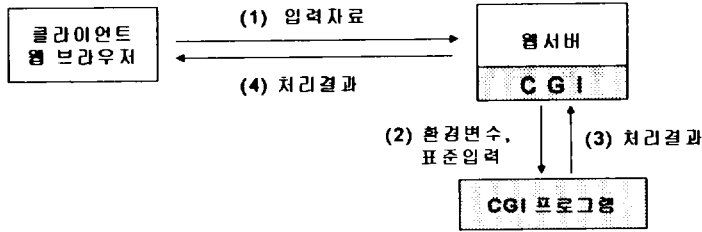
2.1 CGI의 개요

CGI는 Common Gateway Interface의 약자다. 웹브라우저에서 HTML로 여러 가지 정보를 처리하지만, 그 기능만으로 모든 정보처리를 다 할 수 없다. 이것을 보충하기 위한 외부 프로그램과 웹서버(HTTP Server) 간의 연결 역할을 하기 위한 규약이 CGI이다[5]. 또는 넓은 의미로 CGI를 수행하는 외부 프로그램을 포함하여 말하기도 한다. 예를 들어, 홈페이지에 방문객들의 comment를 받을 수 있는 방명록을 만들려고 할 때, 웹에서 구현하는 HTML만으로는 해결할 수 없다. 그래서 외부 프로그램이 필요한데, 이 때 외부 프로그램과 웹간에 서로 주고받을 수 있는 규약을 CGI라고 하고, 그 때 사용하는 프로그램을 gateway프로그램이라고 하는데 이것을 흔히 CGI 프로그램(혹은 스크립트)이라고 한다. 이 CGI 프로그램은 통상적으로 C/C++나 Perl 혹은 UNIX Shell, Tcl/Tk 등을 사용하여 구현한다. 본 논문에서는 III장에서 C 언어를 사용하여 구현한 CGI 프로그램을 소개한다.

2.2 CGI 동작원리

CGI는 웹 브라우저와 웹 서버, 그리고 응용 프로그램간의 인터페이스로 웹 문서 교환방식의 하나다. 다시 말하면 클라이언트가 웹 브라우저를 통해 입력한 정보가 웹 서버를 거쳐 CGI 프로그램으로 전달되고, 가공되어 다시 클라이언트의 웹 브라우저로 전달되기까지의 인터페이스를 의미한다.

<그림 1>은 이러한 동작을 나타낸 것으로 이를 설명하면 다음과 같다.



〈그림 1〉 CGI를 이용한 데이터의 흐름

- (1) 클라이언트(웹브라우저)에서 서버로 TCP/IP 연결을 통해 데이터를 요청
- (2) 서버가 CGI프로그램에게 실행을 하도록 데이터를 전달
- (3) CGI프로그램에서 서버로 처리결과가 전달
- (4) 적절한 MIME헤더를 갖고 서버가 클라이언트로 응답

2.3 입력값 전달 방법

사용자가 브라우저를 통해 서버에 있는 CGI 프로그램으로 값을 넘겨주는 방법에는 GET, POST 방법이 있는데 구성은 일반적으로 다음과 같다.

〈FORM METHOD=GET/POST ACTION="/cgi-bin/test.cgi"〉

여기서 METHOD 생략하면 GET이 선택된다.

(1) GET 메소드

FORM 태그에서 METHOD=GET으로 하거나 생략하면 사용자의 입력 값들이 환경변수(Environment Variable)에 저장되어 넘겨진다. 즉 각 입력 값들이 기본URL에 붙는 인수로서 첨가되어 CGI프로그램으로 값을 넘겨주게 되는 것이다. 물론 이 GET 메소드는 FORM 태그를 사용하지 않고 바로 URL에 인수를 첨가하여 사용할 수도 있다. 예를들어, `http://www.abc.com/cgi-bin/abc.cgi?First+Name=foo&Last+Name=bar` 와 같은 형식으로 사용될 수 있다.

이 GET 메소드를 이용하면 그 입력 값들이 환경변수의 하나인 QUERY_STRING에 들어가서 전달되는데, CGI스크립트는 그 QUERY_STRING에 들어 있는 값을 읽게 되는데, 이 때 그 값들은 입력된 그대로 넘어가는 것이 아니라 서

버에 의해 변환(인코딩)되어 넘어가는데 CGI스크립트에서는 그 값들을 해독(디코딩)해야 한다.

이 GET 메소드는 보통 입력 값들이 많지 않는 경우 혹은 그냥 URL에 붙는 파라미터로 넘겨서 CGI스크립트로 전달할 때에 사용한다.

(2) POST 메소드

FORM 태그에서 METHOD=POST로 하면, GET 메소드가 환경변수중의 하나인 QUERY_STRING을 통해 전달되는 것과 달리 stdin(표준입력)을 통해서 전달된다. GET 메소드가 인수를 통해서 전달되므로 명령행의 길이에 의한 제한을 받는 반면에, POST 메소드는 stdin을 이용하므로 데이터 양의 제한이 없고, 또한 POST 메소드에서도 환경변수들은 stdin과 함께 전달된다. 그리고 POST 메소드도 마찬가지로 입력 값들이 변환되어 넘어옴으로 CGI에서 그 값들을 해독해야 한다.

2.4 환경변수

CGI프로그래밍을 하기 위해서는 이 환경변수를 잘 이해해야 합니다. 클라이언트가 요청하여 gateway프로그램이 수행되면 일련의 환경변수가 넘어온다. 이 환경변수는 크게 두 종류로 나눌 수 있는데, 한 종류는 클라이언트의 요청과 무관한 것으로, 서버 metainformation이라고 일컫는 서버의 정보에 관한 것이고, 다른 한 종류는 클라이언트의 요청에 따라 값이 달라지는 것들이다.

〈표 4〉와 〈표 5〉는 이러한 환경변수를 나타낸 것이다.

〈표 4〉 웹 서버 정보를 나타내는 환경변수

환경변수 이름	설 명
SERVER_SOFTWARE	웹서버의 종류와 버전을 나타낸다.
SERVER_NAME	서버의 호스트 이름과 DNS alias 혹은 IP address
GATEWAY_INTERFACE	웹서버와 외부 프로그램 사이의 인터페이스를 규정하는 표준으로 보통 CGI의 버전을 나타낸다.
SERVER_PORT	웹서버가 사용하고 있는 포트번호를 나타내는 것으로 일반적으로 80번을 사용한다.
SERVER_ADMIN	웹서버 관리자 e-mail 주소가 저장된다. 이 변수의 지정은 웹서버의 httpd.conf 내의 ServerAdmin 지시자로 지정된다.
SERVER_PROTOCOL	웹서버가 지원하는 HTTP 프로토콜 버전을 나타낸다.

〈표 5〉 클라이언트의 요청에 따른 환경변수

환경변수 이름	설 명
SERVER_PROTOCOL	클라이언트 요청이 사용되는 프로토콜
SERVER_PORT	클라이언트 요청을 보내는 포트번호
REQUEST_METHOD	HTML 폼이 사용되는 METHOD 보통 GET나 POST
PATH_INFO	URL 상에서 문서의 상대 디렉토리를 지정할 때 사용함
PATH_TRANSLATED	PATH_INFO에 나타난 가상경로를 실제의 물리적인 경로로 바꾼 값
SCRIPT_NAME	현재 실행이 요청된 스크립트명
QUERY_STRING	GET 방식에서 URL의 뒤에 나오는 정보를 저장하거나 폼입력 정보를 저장한다.
REMOTE_ADDR	클라이언트의 IP 주소
REMOTE_HOST	클라이언트의 호스트 이름
REMOTE_USER	서버가 사용자 인증을 지원하고, 스크립트가 그 확인을 요청한다면, 이것이 확인된 사용자의 이름이 된다.
REMOTE_IDENT	서버가 RFC931 사용자 확인을 지원한다면 서버로부터 발행된 사용자 이름이 이 환경변수에 저장된다. 이 변수의 용도는 로그인에만 한정된다.
CONTENT_LENGTH	POST 방식을 사용하여 클라이언트로 넘겨주는 모든 데이터는 표준 입력을 사용한다. 이때 표준입력으로 전달되어 넘어오는 데이터의 끝을 나타내는 문자가 없다. 그래서 사용자가 입력한 실 데이터의 길이를 나타내는 CONTENT_LENGTH를 사용하여 표준 입력으로 실 데이터를 갖고 온다.
CONTENT_TYPE	POST 방식을 이용하여 검색문에 정보가 들어 있는 경우에 그 정보의 타입을 나타낸다.
DOCUMENT_ROOT	웹 페이지 문서가 저장되어 있는 루트디렉토리를 나타낸다.
PATH	유닉스 상에서 특정 파일 경로를 지정할 때 사용하는 환경변수로 CGI 프로그램에서 다른 프로그램을 호출할 때 이 환경 변수를 통해 여러 파일에 대한 경로를 알 수 있다.

3. XML

3.1 XML의 개요

HTML은 고정된 태그 집합들을 이용하여 페이지들의 레이아웃 정보만을 지원하

고 문서의 구조적 정보를 포함하지 않으며, 잘못된 태그들은 레이아웃에서 배제시키는 정도의 검증만을 지원하는 문제점을 지니고 있다[6]. 따라서, HTML은 현재와 같은 네트워크 상에서 복잡한 정보나 효과적인 검색, 재사용, 검증과 같은 능력에는 취약점을 드러내고 있다. W3C는 이러한 HTML의 취약점을 보완하기 위하여 새로운 마크업 언어인 XML(eXtensible Markup Language)을 1996년 W3C WG에서 공개하였으며, 1998년 2월에는 W3C의 권고안으로서 XML1.0을 발표하였다[7].

XML은 웹 상에서 구조화된 문서를 전송 가능토록 한 마크업 언어로써 HTML과 SGML의 장점을 수용하고, 단점을 극복한 표준언어이다. 특히 XML은 기능면으로는 구조화된 문서의 작성을 지원하므로 보다 복잡한 문서의 작성을 용이하게 하며, 인터넷 기반의 마크업 언어이기 때문에 SGML보다 인터넷 언어로서 사용이 용이하다[8].

특히 최근에 W3C는 사람이 이해하기 쉽고, 시스템에서도 다루기 편리하다는 점에서 XML을 인터넷 문서작성의 차세대 표준언어로 정했고, MS사의 차세대 네트워크인 'Microsoft.NET'의 핵심기술도 XML인 만큼 떠오르는 웹 언어라 아니할 수 없다.

3.2 XML 프로세서

(1) XML 브라우저와 프로세서

XML 프로세서는 XML 문서를 읽어서 문서 내용과 문서 구조에 대한 접근을 제공하는 소프트웨어를 의미하고 XML 브라우저는 XML로 작성된 문서나 데이터를 해석하고 보여주기 위해서 XML 프로세서를 내장하고 있다. XML 프로세서가 HTML 프로세서와 구별되는 차이점은 똑같은 XML 문법으로 작성된 문서라 하더라도 사용자가 작성한 DTD에 따라 서로 다르게 해석되어져야 하는 요소(element)를 처리할 수 있는 방법을 제공한다는 것이다.

XML 프로세서를 이용하는 방법에는 크게 두 가지의 방법이 존재한다. 하나는 XML 브라우저를 이용하여, 문서를 파싱, 해석하며, XSL이나 CSS 같은 스타일 시트 정보를 이용하여 적절히 표시/인쇄하는 프로그래밍이 필요 없는 방법이다. 이 경우의 예로는 기존의 웹 출판 시스템과 같이 문서로서의 보급 및 출판 등의 작업

이 해당된다. 둘째는 XML 프로세서가 제공하는 API를 이용하여 XML 문서에 대한 접근, 갱신, 처리 작업을 할 수 있도록 XML 응용을 프로그래밍하는 방법이다. 프로세서는 문서와 DTD를 읽어 문서의 정확성을 검증하고, 문서 내의 요소와 콘텐츠 트리, 카탈로그 등을 생성해 내며, 이를 이용하여 XML 응용은 해당 요소에 대한 특정 동작을 기술함으로써 XML 문서의 처리를 수행토록 한다.

(2) XML 프로세서의 종류

XML 문서를 파싱하고 접근하는 방법에는 세 가지 방법이 있다. 첫째, 정규식을 이용하여 패턴 대조에 따른 접근이고 두 번째는 XML 프로세서가 제공하는 자체 API를 이용한 문서의 접근 및 변경이고 세 번째는 표준화된 API를 이용한 문서의 접근 및 변경이다.

정규식을 그대로 표현하여 수행하는 파싱은 단순한 XML 문서의 경우, 파싱을 쉽게 할 수 있고, 파싱 속도가 빠른 이점이 있으나, 이는 단일 DTD에 대한 응용 프로그램이며, 다른 DTD에 대해서는 재 사용할 수 없다는 단점과 정규식을 지원하지 않는 프로그래밍 언어로는 이를 처리하기 어렵다는 점 등 많은 문제점이 존재한다. 따라서 XML 프로세서는 여러 DTD에 대한 XML 문서의 파싱을 위해 보다 특별하고 일반화된 방법을 제공할 수 있도록 하며, 프로세서에 대한 접근은 API를 이용토록 하고 있다. API는 프로세서에 따라 자체 API를 제공하는 방법과 표준화된 API를 제공하는 방법이 있으며, 표준화된 대표적인 API로 SAX(9)나 DOM(10, 11)이 존재한다.

또한, XML 프로세서는 적정 형식에 대한 검증과 유효성에 대한 검증을 제공하는가에 따라 valid 프로세서와 well-formed 프로세서로 구분되어 진다. 즉, valid 프로세서의 경우, 태그의 쌍이 제대로 이루어져 있는가 등에 대한 검증뿐만 아니라 해당 태그 또는 요소들이 지정된 DTD에 정의되어 있는가에 대한 검증도 수행한다.

Ⅲ. C언어로 구현한 CGI 프로그램

본 장에서는 C언어로 간단한 방명록 CGI 프로그램을 구현한다(12).

이는 <그림 2>의 HTML과 <그림 3>과 <그림 4>의 CGI 프로그램으로 이루어지는데 본 논문에서는 소스에 대한 설명은 생략한다. 특히 CGI 프로그램에서는 cgic 라이브러리[13]를 이용하는데, 다른 라이브러리로 qDecoder[14]도 있다.

```
<HTML>
<HEAD> <TITLE>베스트북 방명록 - 글 남기기</TITLE> </HEAD>
<BODY>
  <H2>베스트북 방명록 - 글 남기기</H2>
  <FORM ACTION=write.cgi METHOD=post>
  <TABLE WIDTH=600>
    <TR>
      <TD> <SMALL><B>이름</B></SMALL> </TD>
      <TD> <INPUT NAME=name SIZE=30> </TD>
    </TR>
    <TR>
      <TD> <SMALL>E-메일</SMALL> </TD>
      <TD> <INPUT NAME=email SIZE=50> </TD>
    </TR>
    <TR>
      <TD> <SMALL>홈페이지</SMALL> </TD>
      <TD> <INPUT NAME=homepage SIZE=50 VALUE="http://"> </TD>
    </TR>
    <TR>
      <TD> <SMALL>성별</SMALL> </TD>
      <TD> <SMALL>
        <INPUT TYPE=radio NAME=sex VALUE=woman CHECKED>여성(우)
        <INPUT TYPE=radio NAME=sex VALUE=man>남성(♂) </SMALL>
      </TD>
    </TR>
    <TR>
      <TD> <SMALL>홈페이지에<BR>대한 느낌</SMALL> </TD>
      <TD>
```



```

<TABLE CELLSPACING=0 CELLPADDING=0>
<TR ALIGN=center>
  <TD><INPUT TYPE=radio NAME=feel VALUE=1></TD>
  <TD><INPUT TYPE=radio NAME=feel VALUE=2></TD>
  <TD><INPUT TYPE=radio NAME=feel VALUE=3></TD>
  <TD><INPUT TYPE=radio NAME=feel VALUE=4 CHECKED></TD>
  <TD><INPUT TYPE=radio NAME=feel VALUE=5></TD>
</TR>
<TR ALIGN=center>
  <TD WIDTH=60><SMALL>허잡하다</SMALL></TD>
  <TD WIDTH=60><SMALL>별로</SMALL></TD>
  <TD WIDTH=60><SMALL>보통</SMALL></TD>
  <TD WIDTH=60><SMALL>괜찮다</SMALL></TD>
  <TD WIDTH=60><SMALL>최고!</SMALL></TD>
</TR>
</TABLE>
</TD>
</TR>
<TR> <TD COLSPAN=2> &nbsp; </TD> </TR>
<TR> <TD COLSPAN=2> <B>남기실 말씀</B> </TD> </TR>
<TR> <TD COLSPAN=2>
  <TEXTAREA NAME=content COLS=70 ROWS=10>여기에 하시고 싶은 말씀을 써 주세요
  </TEXTAREA>
</TD> </TR>
</TABLE>
<BR>
<INPUT TYPE=submit VALUE="   글 남기기   ">
<INPUT TYPE=reset VALUE="지우고 다시 쓰기">
</FORM>
</BODY>
</HTML>

```

〈그림 2〉 write.html

```
#include "cgic.h"
#include <time.h>
#define BUF_SIZE 1024
#define GUESTBOOK_FILE "content.dat"
int strcheck(char *str) {
    if (str == NULL) return 0;
    if (strlen(str) == 0) return 0;
    return 1;
}
int cgiErrorMsg(char *msg) {
    cgiHeaderContentType("text/html");
    printf("<HTML>\n\n");
    printf("<HEAD> <TITLE> Error Message </TITLE> </HEAD>\n\n");
    printf("<BODY> <B>%s</B> </BODY>\n</HTML>", msg);
    exit(1);
}
int cgiMain(void) {
    char name[BUF_SIZE], email[BUF_SIZE], homepage[BUF_SIZE];
    char sex[BUF_SIZE], content[BUF_SIZE];
    int feel;
    FILE *fp;
    char buf[BUF_SIZE];
    time_t t;
    int i;
    cgiFormString("name", name, BUF_SIZE);
    cgiFormString("email", email, BUF_SIZE);
    cgiFormString("homepage", homepage, BUF_SIZE);
    cgiFormString("sex", sex, BUF_SIZE);
    cgiFormInteger("feel", &feel, 1);
    cgiFormString("content", content, BUF_SIZE);
    if (!strcheck(name)) cgiErrorMsg("이름을 입력 안하셨네요.");
    if (!strcheck(content)) cgiErrorMsg("내용을 써 주셔야죠.");
```

```

fp = fopen(GUESTBOOK_FILE, "a");
if (!fp)
    cgiErrorMsg("방명록에 추가할 수 없습니다.");
fprintf(fp, "<TR> <TD COLSPAN=2 BGCOLOR=#888888> <FONT SIZE=-1
COLOR=white>\n");
time(&t);
fprintf(fp, "<B>%s 에 '%s'님께서 남기신 글</B></FONT>", ctime(&t), name);
fprintf(fp, "</TD> </TR>\n");
/* 올린이 이름과 성별, E-메일 */
fprintf(fp, "<TR>\n");
fprintf(fp, "<TD BGCOLOR=#EEEEEE> <SMALL> <B>올린이</B> </SMALL> </TD>\n");
fprintf(fp, "<TD> <SMALL> <B>%s</B>(", name);
if (strcmp(sex, "man")) fprintf(fp, "♀) </B> ");
else fprintf(fp, "♂) </B> ");
if (strcheck(email)) fprintf(fp, "(<A HREF=\"mailto:%s\">%s</A>)", email, email);
fprintf(fp, "</SMALL> </TD>\n");
fprintf(fp, "</TR>\n");
fprintf(fp, "<TR>\n");
fprintf(fp, "<TD BGCOLOR=#EEEEEE> <SMALL> 홈페이지 </SMALL> </TD>\n");
if (strcheck(homepage)) {
    fprintf(fp, "<TD><SMALL><A HREF=\"%s\">%s</A></SMALL></TD>\n", homepage,
homepage);
}
else fprintf(fp, "<TD> <SMALL> 홈페이지가 없으시다는군요. </SMALL> </TD>\n");
fprintf(fp, "</TR>\n");
fprintf(fp, "<TR>\n");
fprintf(fp, "<TD BGCOLOR=#EEEEEE> <SMALL> 접속한 곳 </SMALL> </TD>\n");
fprintf(fp, "<TD> <SMALL> %s </SMALL> </TD>\n", getenv("REMOTE_ADDR"));
fprintf(fp, "</TR>\n");
fprintf(fp, "<TR>\n");
fprintf(fp, "<TD BGCOLOR=#EEEEEE> <SMALL> 홈페이지 느낌 </SMALL> </TD>\n");
fprintf(fp, "<TD> ");

```

```

switch (feel) {
  case 1 : fprintf(fp, "<B> 허잡하다 </B> ");
           fprintf(fp, "<SMALL> 별로 보통 괜찮다 최고! </SMALL> </TD>\n");
           break;
  case 2 : fprintf(fp, "<SMALL> 허잡하다 </SMALL> <B> 별로 </B> ");
           fprintf(fp, "<SMALL> 보통 괜찮다 최고! </SMALL> </TD>\n");
           break;
  case 3 : fprintf(fp, "<SMALL> 허잡하다 별로 </SMALL> <B> 보통 </B> ");
           fprintf(fp, "<SMALL> 괜찮다 최고! </SMALL> </TD>\n");
           break;
  case 4 : fprintf(fp, "<SMALL> 허잡하다 별로 보통 </SMALL> <B> 괜찮다 </B> ");
           fprintf(fp, "<SMALL> 최고! </SMALL> </TD>\n");
           break;
  case 5 : fprintf(fp, "<SMALL> 허잡하다 별로 보통 괜찮다 </SMALL> ");
           fprintf(fp, "<B> 최고! </SMALL> </TD>\n");
}
fprintf(fp, " </TR>\n");
fprintf(fp, " <TR>\n");
fprintf(fp, "   <TD COLSPAN=2 BGCOLOR=#DDDDDD> <SMALL> <BR>\n");
fprintf(fp, "   &nbsp; ");
for (i = 0; i < strlen(content); i++) {
  if (content[i] != '\n') fprintf(fp, "%c", content[i]);
  else fprintf(fp, "<BR>\n   &nbsp; ");
}
fprintf(fp, " <BR> <BR>\n");
fprintf(fp, "   </TD>\n");
fprintf(fp, " </TR>\n");
fprintf(fp, " <TR> <TD COLSPAN=2> &nbsp; </TD> </TR>\n");
fclose(fp);
printf("Location: guestbook.cgi%c%c", 10, 10);
}

```

〈그림 3〉 write.cgi의 소스 프로그램

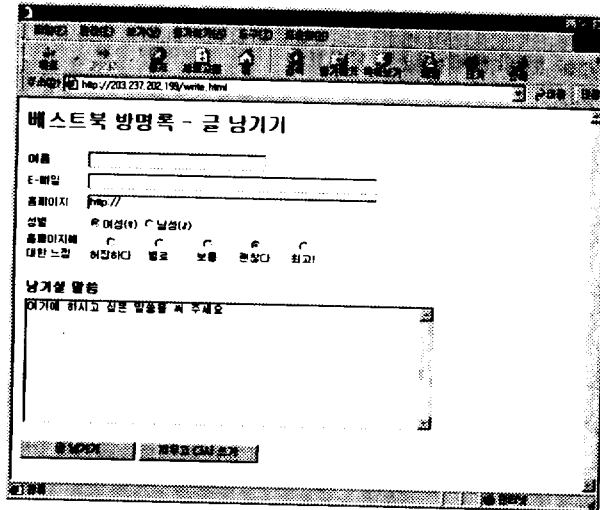
```

#include <stdio.h>
#define GUESTBOOK_FILE "content.dat"
int main(void) {
    char line[255];
    FILE *fp;
    printf("Content-type: text/html%c%c", 10, 10);
    printf("<HTML>\n\n");
    printf("<HEAD> <TITLE>베스트북 방명록</TITLE> </HEAD>\n\n");
    printf("<BODY LINK=black VLINK=black>\n\n");
    printf(" <TABLE WIDTH=600> <TR>\n");
    printf("     <TD> <FONT SIZE=5><B>베스트북 방명록</B></FONT></TD>\n");
    printf("     <TD ALIGN=right>\n");
    printf("         <SMALL><B><A HREF=\"write.html\">글 남기기</A></B></SMALL>\n");
    printf("     </TD>\n");
    printf(" </TR> </TABLE>\n");
    printf(" <BR>\n");
    printf(" <TABLE WIDTH=600>\n\n");
    fp = fopen(GUESTBOOK_FILE, "r");
    if (!fp) {
        printf(" <TR> <TD BGCOLOR=#DDDDDD>\n");
        printf("     <B> 방명록에 올라온 글이 하나도 없습니다.</B>\n");
        printf(" </TD> </TR>\n");
    }
    else while (fgets(line, 255, fp)) {
        printf("%s", line);
    }
    printf(" </TABLE>\n\n");
    printf("</BODY>\n\n");
    printf("</HTML>");
}

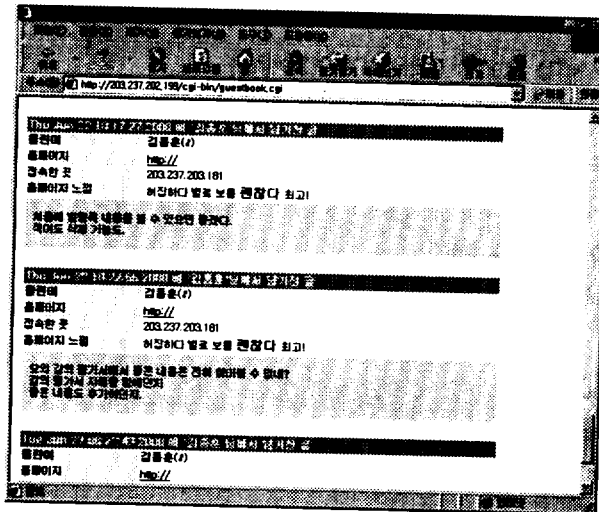
```

〈그림 4〉 guestbook.cgi의 소스 프로그램

〈그림 5〉는 방명록의 첫 화면이고, 방명록에 내용을 작성한 후 글 남기기를 선택했을 때의 화면이 〈그림 6〉이다. 이의 구현은 리눅스 운영체제의 486 개인용 컴퓨터에 아파치 서버를 설치한 환경에서 gcc를 이용하여 이루어졌다.



〈그림 5〉 방명록의 첫 화면



〈그림 6〉 방명록에 글 올렸을 때의 화면

IV. 결 론

HTML만으로는 대화적인 웹사이트를 구축하기에는 한계에 와 있어 여러 웹 프로그래밍 기술이 생겨나고 있다. 이에 본 논문에서는 클라이언트와 서버간에 상호작용을 할 수 있는 대화적인 웹사이트를 구축할 수 있는 웹 프로그래밍 기술 중 PHP, CGI, XML에 대해 살펴보고, C언어로 CGI 프로그램을 구현해 보았다.

PHP는 서버에 내장된 모듈로서 여러 명의 클라이언트가 접속을 해도 하나의 프로세서가 처리하기 때문에 메모리 측면에서 효율적인 장점을 지니고 있고, CGI는 서버와 외부 프로그램과 연결시켜주는 인터페이스로 클라이언트에서 서버에 자료를 넘겨주거나 연산을 필요하게 될 경우에는 서버가 CGI를 통해 외부 프로그램과 대화를 통해 원하는 연산결과를 받기 때문에 데이터를 생성할 수 있는 장점이 있다. 인터넷 문서작성의 차세대 표준언어로 정해진 XML은 웹에서의 문서 작성을 위한 HTML의 기능을 보완하는 단순한 마크업 언어로서 머물지 않고 기존의 응용들이 다루었던 데이터 및 문서의 많은 영역들을 대체할 것이며, 그에 따른 새로운 응용들을 창출해 낼 것이다.

◆ 참고 문헌 ◆

- [1] S. Schumann, Professional PHP Programming, WROX Press, 1999.
- [2] 김창수, 정희경, Web Script 언어 기술동향, 한국정보과학회지 제18권 제4호, pp. 21-31, 2000년 4월.
- [3] <http://www.php.net>.
- [4] <http://php.flyduck.com>.
- [5] <http://www.cgkim.co.kr>.
- [6] 이강찬, 이규철, XML과 그 미래, EDI/EC Magazine, 가을호(15), 1998.
- [7] W3 Consortium, W3C Recommendation: Extensible Markup Language 1.0, <http://www.w3.org/TR/1998/REC-xml-19980210>.
- [8] 이경하, 이강찬, 이규철, XML 프로그래밍, 한국정보과학회지 제18권 제4호, pp. 4-12, 2000년 4월.
- [9] D. Megginson, Simple API for XML(SAX), <http://www.megginson.com/S-AX/index.html>.
- [10] W3 Consortium, Document Object Model(DOM), <http://www.w3.org/DOM>.
- [11] 이강찬, DOM 동향 기술 분석서, <http://dmlab.ce.cnu.ac.kr/~dolphin/xml/atoz/dom.html>.
- [12] 방민석, 카이스트 SPARCS와 함께하는 CGI, 베스트북, 1999.
- [13] <http://www.boutell.com/cgic>.
- [14] <http://freeware.cgiserver.net>.