

CGI

Daniel Morató
Area de Ingeniería Telemática
Departamento de Automática y Computación
Universidad Pública de Navarra
daniel.morato@unavarra.es
Laboratorio de Interfaces de Redes
<http://www.tlm.unavarra.es/asignaturas/lir>

¿Qué significa “CGI”?

- CGI = “Common Gateway Interface”
- Define una forma de comunicarse con un servidor Web:
 - La forma de obtener del servidor Web información sobre las peticiones del cliente Web (ej: contenido de formularios)
 - La forma de darle al servidor Web un nuevo documento que enviar al cliente (ej: los bytes de una imagen o el código HTML de una nueva página Web)
- Los comúnmente llamados “CGIs” son simples programas preparados para ejecutarse en un sistema operativo en concreto.
- ¿En qué lenguaje hay que escribirlos? En casi cualquiera: C, C++, Pascal, Fortran, Perl, script de una shell, Python, Tcl/Tk, AppleScript, Visual Basic...
- ¿Qué se debe poder hacer para comunicarse con el servidor Web?:
 - Para recibir información del servidor Web debemos: poder leer de la entrada estándar y leer el contenido de las variables de entorno
 - Para mandar información al servidor Web debemos: poder escribir por la salida estándar

¿Qué lenguaje emplear?

- Tenemos dos tipos de lenguajes para elegir:

- Lenguajes compilados

- C, C++, Pascal...
- Hay un código fuente que es compilado en un fichero ejecutable en el lenguaje de la máquina en que vaya a ejecutarse
- Se puede distribuir solo el ejecutable con lo que se mantiene control sobre el código
- Pero hay que compilar el código para la combinación procesador+S.O. en concreto en que se vaya a ejecutar
- Generalmente son programas más pequeños y rápidos

- Lenguajes interpretados

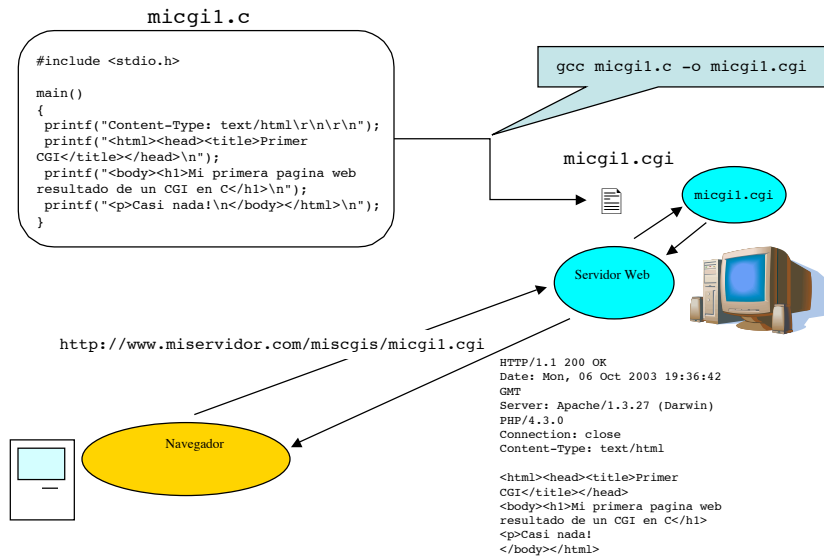
- Perl, Tcl/Tk, Python, bash...
- El código del programa es interpretado por otro programa. No se compila sino que un programa compilado (el intérprete) ejecuta las instrucciones que indica el "script"
- Siempre que sobre esa plataforma exista el intérprete se puede ejecutar el script sin cambios
- Generalmente son más lentos porque requieren ejecutar primero el programa intérprete y luego éste es más lento ejecutando las instrucciones que si fuera código compilado
- Suelen ser lenguajes más sencillos de programar y depurar

21 Oct

CGI

2/12

Ejemplo simple



21 Oct

CGI

3/12

Ejemplo simple

```
micgil.c
#include <stdio.h>
main()
{
    printf("Mi primera pagina web resultado de un CGI en C\n");
    printf("Casi nada!\n");
}
```

Primer CGI

http://localhost/~daniel/miscgis/cgi1.cgi

Google

Apple Amazon eBay Yahoo! News ResearchIndex VersionTracker CDS Interfaces

Mi primera pagina web resultado de un CGI en C

Casi nada!

http://www.m...miscgis/micgil.cgi

Navegador

```
HTTP/1.1 200 OK
Date: Mon, 06 Oct 2003 19:36:42 GMT
Server: Apache/1.3.27 (Darwin) PHP/4.3.0
Connection: close
Content-Type: text/html

<html><head><title>Primer CGI</title></head>
<body><h1>Mi primera pagina web resultado de un CGI en C</h1>
<p>Casi nada!</p>
</body></html>
```

21 Oct CGI 4/12

Análisis del ejemplo

- Lo que el programa (CGI) envía a la salida estándar (stdout) llega al navegador a través del servidor Web
- Lo que envía el CGI tiene una primera parte que no muestra el navegador:

```
HTTP/1.1 200 OK
Date: Mon, 06 Oct 2003 19:36:42 GMT
Server: Apache/1.3.27 (Darwin) PHP/4.3.0
Connection: close
Content-Type: text/html
```

- Es la cabecera HTTP (el protocolo empleado en la conexión TCP). Va antes del documento enviado y separado por una línea en blanco.
- Esta cabecera tiene muchas opciones. Content-Type es lo mínimo de la cabecera HTTP que debe construir el CGI, el resto lo construirá el servidor Web (si el script construye toda la cabecera HTTP se llama script NPH=Non-Parsed Header)
- El Content-Type especifica el tipo de documento que se está enviando, en este caso una página HTML. El formato es: "Content-Type: Tipo/Subtipo", donde Tipo/Subtipo hacen referencia a un tipo MIME (Multipurpose Internet Mail Extensions, RFC 1521).
Algunos tipos son: image/gif, video/mpeg, application/pdf, image/jpeg, etc.
- Las líneas en la cabecera HTTP y la línea en blanco deben estar terminadas por un retorno del carro (\r en C) y un fin de línea (\n en C) aunque con muchos servidores Web funciona el emplear solo el fin de línea.
- Tras la cabecera HTTP va el contenido del documento, del tipo que se ha especificado. En el ejemplo es un texto (un documento en HTML) pero podría ser por ejemplo una imagen en cuyo caso ahora se escribirían los datos binarios que forman la imagen.

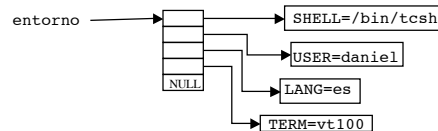
- El CGI tiene un nombre terminado en .cgi lo cual no es imprescindible. Al servidor Web se le configura qué ficheros, cuando se soliciten, son ejecutables CGI que debe lanzar en otro proceso. En este caso el servidor se ha configurado para reconocer estos ficheros por la extensión .cgi

Datos de entrada para un CGI

(Variables de entorno)

- El CGI puede acceder a datos generales sobre el servidor Web, el navegador y la petición. En versiones UNIX se hace mediante variables de entorno:

- Las llamadas "variables de entorno" son un conjunto de cadenas accesibles por el programa
- En las Shells se puede dar valor y consultar dichas variables. Se heredan al crear nuevos procesos con `fork()` (lógicamente) y se pueden especificar para un nuevo proceso al ejecutar un programa con `execve()`
- En C estas cadenas están en un array de cadenas:



- En C se puede conseguir este array de dos formas:
Está en el tercer argumento de la función `main()`: `main(int numero_args, char *args_en_linea[], char *entorno[]);`
Hay una variable global con ese valor con nombre `environ` que se puede declarar como: `extern char **environ;`
- La forma de modificarlo dependerá del lenguaje en que se escriba el CGI: En C existen unas funciones muy útiles (`getenv()`, `setenv()`...)
- Algunas variables de entorno que crea el servidor Web:

<code>REMOTE_ADDR</code>	Dirección IP del cliente (el navegador)
<code>HTTP_ACCEPT</code>	Lista de tipos MIME que acepta el navegador
<code>HTTP_USER_AGENT</code>	Descripción del navegador (nombre, versión, sistema operativo...)
<code>SERVER_PORT</code>	Puerto por el que aceptó la conexión el servidor Web
<code>SERVER_SOFTWARE</code>	Nombre y versión del servidor web

21 Oct

CGI

6/12

Datos de entrada para un CGI

(Información de formularios)

- El CGI puede acceder a información introducida por el usuario en un formulario. La información puede venir de dos formas diferentes:

- Método GET:

Se envía como parte del URL. Ejemplo: `http://myserver.org/cgi-bin/procesa.cgi?nombre=John+Smith&edad=54`
El servidor Web se lo entrega al CGI dentro de la variable `QUERY_STRING`

Problemas:

- El tamaño máximo suele estar limitado
- Se ve el contenido del formulario en el URL

- Método POST:

Se envía en la cabecera HTTP

El CGI tiene acceso a él a través de la entrada estándar (se lo envía el servidor Web)

En la variable `CONTENT_LENGTH` se especifica la longitud en bytes de lo que se puede leer por `stdin`

- Se especifica que se esté empleando un método u otro en la variable de entorno `REQUEST_METHOD` que valdrá `GET` o `POST` y proviene de que en el formulario se especifique el valor del atributo `method` del tag `FORM` como uno u otro
- El navegador codifica el contenido del formulario antes de enviarlo. Se llama "URL encoding" (RFC 1738):
 - Los diferentes campos del formulario se separan con un ampersand (&)
 - Se coloca nombre y valor de cada campo donde el nombre es el valor del atributo `name` y el valor depende del tipo de elemento
 - Los espacios se cambian por el signo +
 - Los caracteres "extraños" (generalmente que no están en el US-ASCII o que sea un carácter reservado) aparecen como un signo de porcentaje seguido de un código hexadecimal

21 Oct

CGI

7/12

Datos de entrada para el CGI

(URL encoding: Ejemplo)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html><head><title>Controles</title></head><body>
<form action="http://localhost/~daniel/miscgis/cgi2.cgi"
method="get">
Texto:<input type="text" name="cadena" value="Texto por
defecto"><br>
Fecha de nacimiento:<input type="text" name="fecha"><br>
<input type="checkbox" checked name="check1">Checkbox<br>
<input type="checkbox" checked name="check2">Otro checkbox<br>
<input type="radio" name="miradio" value="1" checked>Radio
button<br>
<input type="radio" name="miradio" value="2">Otro radio button
<input type="submit" value="Envia todo"><br>
<input type="reset" value="Borra todo">
</form></body></html>
```

Texto:

Fecha de nacimiento:

Checkbox

Otro checkbox

Radio button

Otro radio button

QUERY_STRING=cadena=ocho+es+%3E+tres&fecha=&check1=on&miradio=1

Redirección

- La cabecera HTTP de respuesta del CGI en vez de incluir un "Content-Type" puede llevar un "Location"
- Le indica al servidor que el CGI devuelve una referencia a un documento en vez del documento
- Se indica el camino al documento:
 - Si se indica como un camino relativo al disco local el servidor enviará ese fichero al cliente sin que éste note diferencia con que lo hubiera pedido directamente
 - Si se indica un URI a otro servidor le llegará al cliente una indicación de que debe dirigirse a ese otro documento en ese otro servidor

```
printf("Location: /~daniel/index.html\n\n");
```

```
printf("Location:
http://www.tlm.unavarra.es/~daniel/docencia/lpr/
lpr03_04\n\n");
```

```
HTTP/1.1 200 OK
Date: Tue, 07 Oct 2003 19:40:38 GMT
Server: Apache/1.3.27 (Darwin) PHP/4.3.0
Last-Modified: Tue, 17 Jun 2003 15:06:29 GMT
ETag: "2a-1fad-3ee2e75"
Accept-Ranges: bytes
Content-Length: 8109
Connection: close
Content-Type: text/html
X-Pad: avoid browser bug

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN"
"http://www.w3.org/html/99/xhtml-head-ht-meta http-
equiv="content-type"
content="text/html; charset=iso-8
```

```
HTTP/1.1 301 Moved Permanently
Date: Tue, 07 Oct 2003 19:37:28 GMT
Server: Apache/1.3.22 (Unix) (Red-Hat/Linux) PHP/3.0.15
mod_perl/1.21
Location: http://www.tlm.unavarra.es/asignaturas/lpr/
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD><TITLE>301 Moved Permanently</TITLE></HEAD><BODY>
<H1>Moved Permanently</H1>
The document has moved <A
HREF="http://www.tlm.unavarra.es/asignaturas/lpr/">here</A>.<P><HR>
<ADDRESS>Apache/1.3.22 Server at www.tlm.unavarra.es Port
80</ADDRESS>
</BODY></HTML>
```

Server-Side Includes (SSI)

- Podemos indicarle al servidor Web una parte de un documento HTML que debe ser substituido por el resultado de algún comando o CGI.
- El servidor web busca en la página web esos tags especiales, ejecuta lo que indican y coloca el resultado en el documento que envía al navegador.
- En la página que recibe el navegador no queda rastro de ese CGI incluido en la página.
- Apache emplea la etiqueta de comentario de HTML para marcar estos SSIs, de forma que si están desactivados no confundan al cliente. Ejemplo:

Documento .shtml

```
<html>
<head><title>SSI 1</title></head>
<body>
<h1>SSI 1</h1>
<p>
Bla bla bla
<!--#exec cmd="uncgi.cgi" -->
</body>
</html>
```

uncgi.cgi

```
#!/bin/sh
echo "<hr><p>Soy el resultado de un SSI"
echo -n "<p>Y hoy es "
date
echo "<br><hr>"
```

HTML recibido por el navegador

```
<html>
<head><title>SSI 1</title></head>
<body>
<h1>SSI 1</h1>
<p>
Bla bla bla
<hr><p>Soy el resultado de un SSI
<p>Y hoy es Thu Oct 9 13:48:00 CEST 2003
<br><hr>
</body>
</html>
```

21 Oct

CGI

10/12

Resumen

- Los CGIs son programa que ejecuta el servidor cuando se le solicita un URI que hace referencia a un CGI
- Pueden obtener información del usuario extrayéndola de los formularios que rellena, a través de variables globales o la entrada estándar
- El resultado del programa por la salida estándar llega al navegador a través del servidor web
- El CGI puede sacar texto, html, imágenes, etc. Debe indicar el tipo (MIME) de lo que saca
- Podemos hacer que el resultado de un CGI se incluya en el contenido de un documento que solicita el navegador (Server Side Includes)

21 Oct

CGI

11/12

Próximo día

Introducción a PHP

21 Oct

CGI

12/12