TNE30019/TNE80014 – Unix for Telecommunications

Apache – Common Gateway Interface

Dr. Jason But

Swinburne University

Outline

Common Gateway Interface (CGI)

What is CGI?

- Allows for dynamic content
  - Returned content is generated based on user input
  - User can send input via GET or POST to server
  - **CGI program** uses input to generate content that is returned

CGI programs

- Regular program on server
- Can be written in any language (including shell scripts)

CGI – Input

- Input generated by browser
- Apache server will pass any input to CGI program

GET requests

- `GET <url>?<parameters>`
- `GET <url>?name1=value1&name2=value2`

POST requests

- `POST` posts back form with data in fields

Apache will convert all variables into single line of text

- `name1=value1&name2=value2&name3=value3`
- Spaces and other "special" characters are converted to ASCII/hex (for example space will be %20)
- Parameter string passed to CGI program via **stdin**

CGI – Output

- CGI program writes all output to **stdout**
- Output must be correctly formatted for web browser
- First, output HTTP name-value pair specifying MIME type
- Then, output actual content

Sample output for MIME type `text/html`

```
Content-type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html lang="en">
<head>
etc...
```

- Program can be tested by executing it within shell

CGI and Apache

When GET/POST of CGI Program is requested

- Apache forks new process to run CGI Program
- **Program is executed as user Apache is running as**
- Apache creates bidirectional pipe between itself and CGI process
- Data output by CGI program is piped to Apache which forwards it to web browser

But Apache must be configured to do this

CGI – Configuring Apache

Dedicated directories for scripts

```
ScriptAlias /cgi-bin/ /home/www/cgi-bin
```

- Goes inside `<Directory>` block
- All files in this directory assumed to be executable
- Apache will treat them as CGI programs – even if they are not
- Can create many ScriptAlias directives to specify many locations for executable programs

Dual-purpose directories (static content and scripts)

```
Options +ExecCGI
AddHandler cgi-script .cgi .pl .exec
```

- Static and dynamic content from same path
- Programs with specified extensions will be treated as CGI programs

CGI – Common Problems

CGI Program is not executable

Programs must be executable for user Apache is running as

Program written for command line

- Uses environment variables that may not exist
- Assumes default PATH settings
- Define variables including PATH explicitly

Output to stdout is not properly formatted

- MIME type essential
- Data must be formatted to match specified MIME type

Troubleshoot Problems → Error Logs

- Generated by Apache with info what failed
- Generated by CGI Program by writing to **stderr**

CGI – Writing Programs

Generating HTML code is not easy

Most languages used for CGI provide libraries

- Help generate portions of HTML code
- Parse input from **stdin**
- Generate headers
- Generate forms
- Generate HTML constructs
- Need to be downloaded and installed

Most CGI Programs written as scripts

- Easy to debug
- Quick to test – no need to compile
- PHP, Perl, Python
- But can use any scripting language

CGI – Example CGI Program

```
#!/usr/local/bin/bash
echo "Content-type: text/html"
echo ""
echo '<html>'
echo '<head>'
echo '<meta http-equiv="Content-Type"
     content="text/html; charset=UTF-8">'
echo '<title>System Uptime</title>'
echo '</head>'

export PATH="/bin:/usr/bin"

echo '<body>'
echo '<h3>'
hostname
echo '</h3>'
uptime
echo '</body></html>'

exit 0
```


CGI – Example CGI Program

```
<html>
<head>
<title>HTML form that calls itself</title>
</head>
<body>
<?php
if ( ! empty( $_POST['answer'] ) ) {
    print "last answer: ".$_POST['answer'];
}
?>
<form method="post" action="<?php print $_SERVER['PHP_SELF']?>">
<p>
Type your answer here: <input type="text" name="answer" />
</p>
</form>
</body>
</html>
```