

CGI / Perl

Assistente de ensino: Marcelo da Silva Reis¹

Professor: Fabio Kon¹

¹Instituto de Matemática e Estatística, Universidade de São Paulo

MAC0211 - Laboratório de Programação I

17 de junho de 2009



Conteúdo

CGI em Perl

O que é CGI?

Produzindo scripts CGI

Exemplos de aplicações

Exercício

Conteúdo

CGI em Perl

O que é CGI?

Produzindo scripts CGI

Exemplos de aplicações

Exercício

O que é CGI?

- ▶ **C**ommon **G**ateway **I**nterface
- ▶ protocolo para chamar aplicações remotas através de um servidor *web*
- ▶ Tá, mas é qual a utilidade disso?

O que é CGI?

- ▶ **Common Gateway Interface**
- ▶ protocolo para chamar aplicações remotas através de um servidor *web*
- ▶ Tá, mas é qual a utilidade disso?
 - ▶ permite disponibilizar recursos dinâmicos na *web*;
 - ▶ ao contrário de recursos estáticos (ex: páginas HTML), os dinâmicos contêm informações que podem mudar a cada solicitação

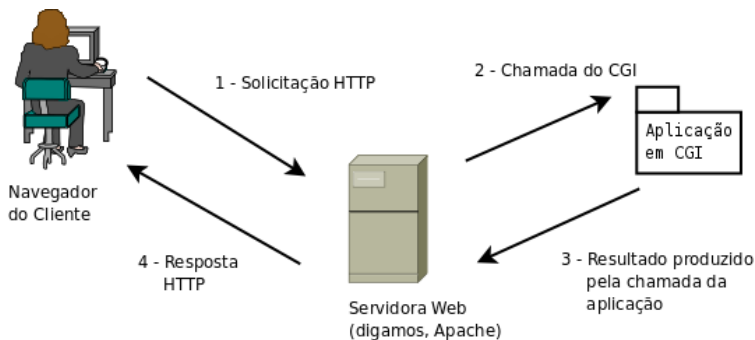


O que é CGI?

- ▶ **Common Gateway Interface**
- ▶ protocolo para chamar aplicações remotas através de um servidor *web*
- ▶ Tá, mas é qual a utilidade disso?
 - ▶ permite disponibilizar recursos dinâmicos na *web*;
 - ▶ ao contrário de recursos estáticos (ex: páginas HTML), os dinâmicos contêm informações que podem mudar a cada solicitação
- ▶ Perl é, de longe, a linguagem mais utilizada em scripts CGI



Uma chamada de um script CGI



Roteiro de execução de CGI / Perl

1. Seu navegador solicita a uma servidora *web* a execução de um script CGI, com ou sem parâmetros. Exemplo:
`http://www.usp.br/cgi-bin/foo.pl?p1=bar&p2=baz`
2. A servidora *web* recebe a solicitação, executa o programa Perl com os parâmetros (se houverem)
3. O programa Perl é processado com os parâmetros recebidos e devolve ao Apache um resultado (normalmente uma página HTML)
4. A servidora *web* devolve ao computador que fez a solicitação o resultado produzido pelo programa Perl



Servidora HTTP Apache

- ▶ servidora *web* de código aberto e livre
- ▶ uma dos componentes do *solution stack* LAMP ²

LAMP: **L**inux, **A**pache, **M**ySQL, **P**erl

²<http://www.think-lamp.com/>

Mensagens HTTP

- ▶ Métodos :
 - ▶ GET: sem conteúdo no corpo na mensagem HTTP
(o Apache passa os parâmetros para o CGI através da URL)
 - ▶ POST: com conteúdo no corpo da mensagem, em formato de string
(o Apache passa os parâmetros utilizando o STDIN)
- ▶ Cabeçalho:
 - ▶ Content-type: text/html, application/pdf, etc.
 - ▶ Host: nome do host solicitante



Exemplos

Solicitação:

```
GET /cgi-bin/hello.pl HTTP/1.1  
Host: localhost
```

Resposta:

```
HTTP/1.1 200 OK  
Content-Type: text/html  
Content-Length: 70  
<HTML>  
<HEAD><TITLE>Hello, world!</TITLE>  
<BODY><P>Hello, World!</BODY>  
</HTML>
```



“Hello, world!” em CGI / Perl

```
#!/usr/bin/perl -w

use strict;

print "Content-type: text/html\n\n";

print<<END_OF_PAGE;
<HTML>
<HEAD><TITLE>Hello, world!</TITLE>
<BODY><P>Hello, World!</BODY>
</HTML>
END_OF_PAGE
```



O ambiente de um programa CGI

- ▶ Permissões de scripts: normalmente eles têm as mesmas permissões que o servidor *web*
 - ▶ Por razões de segurança, esses scripts devem ser capazes de ler e de escrever apenas em áreas de seu domínio
- ▶ o E/S é feito principalmente através dos ponteiros STDIN, STDOUT e STDERR
- ▶ informações adicionais são coletadas através de variáveis de ambiente (em %ENV)

Ponteiros de arquivos em CGI / Perl

- ▶ STDIN: o programa lê os parâmetros do método POST através dele (nunca o leia quando o método for GET, pois pode “matar” a execução)
- ▶ STDOUT: o programa escreve a resposta ao Apache neste ponteiro
- ▶ STDERR: quando ocorre um erro, o Apache pode anexar a mensagem em STDERR à resposta a ser enviada ao cliente



Algumas variáveis de ambiente

CONTENT_LENGTH	o comprimento dos dados (em bytes) enviados em STDIN
QUERY_STRING	as informações na URL que estão após o "?"
REQUEST_METHOD	o método utilizado na solicitação (POST, GET, etc)

Exemplo de uso:

```
if ($ENV{REQUEST_METHOD} eq "GET"){  
    $mensagem = $ENV{QUERY_STRING};  
}
```



CGI.pm

- ▶ módulo Perl para programação de scripts CGI
- ▶ fornece uma boa API para manipular os dados de entrada e de saída (HTML ou XHTML)
- ▶ duas opções de interface: procedural e orientada à objetos
- ▶ página do CGI.pm:
`http://stein.cshl.org/WWW/software/CGI/`

Exemplos de aplicações

Agora vamos ver alguns exemplos de aplicações.

Exercício

- ▶ instale a servidora Apache em seu computador;
 - ▶ dê uma olhada se o daemon é executado, (`ps -ef | grep apache2`)
 - ▶ leia o arquivo de configuração (geralmente algo como `/etc/apache2/apache2.conf`)
 - ▶ verifique os caminhos dos diretórios `www` e `cgi-bin`, assim como se chamadas ao `localhost` em seu computador funcionam corretamente

Exercício

- ▶ crie um arquivo TXT, com cada linha contendo um nome de aluno e 2 números (notas)
- ▶ faça um HTML com uma caixa de texto (para escrever um nome ou parte dele) e um botão de submissão
- ▶ por fim, escreva um script CGI em Perl que receba a solicitação do HTML do ítem acima e imprima uma tabela HTML que contenha o nome, notas e média de todos os alunos cujo nome case com a expressão fornecida.



Referências

1. Perl.org. <http://www.perl.org/>.
Acesso em 17 de junho de 2010.
2. Comprehensive Perl Archive Network.
<http://www.cpan.org/>.
Acesso em 17 de junho de 2010.
3. Livros da O'Reilly:
 - ▶ *Learning Perl*.
 - ▶ *Programming Perl*.
 - ▶ *CGI Programming with Perl, Second Edition*