UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

O início da interatividade

QXD0279 - Desenvolvimento de Software para Web 2

Prof. Bruno Góis Mateus (brunomateus@ufc.br)

Agenda

- Introdução
- CGI
- Formulários
- JavaScript

Introdução

Introdução

- Em 1994 o navegador Mosaic já dava suporte a formulários para envio de dados para o servidor
- Neles, a entrada do usuário é obtida em elementos de formulário
  - O envio do formulário remete essa entrada para o servidor
  - O servidor então processa a entrada e gera uma nova página HTML
    - Normalmente com novos elementos de formulário para exibição no cliente

Introdução

- Diante do potencial apresentado pela web, o time do National Center for Supercomputing Applications (NCSA) desenvolveu naquela época uma tecnologia que permitiu o uso de páginas dinâmicas
  - Common Gateway Interface (CGI)
- Em 1994, páginas dinâmicas poderiam ser servidas via método POST
- Combinando formulários com CGI, naquele momento um novo mundo surgia
  - Um mundo minimamente dinâmico

Um mundo dinâmico

Um mundo dinâmico

Common Gateway Interface Programming (CGI)

- Anos antes da invenção do JavaScript, a especificação do que viríamos a conhecer como CGI começou a sair do papel
- Esta tecnologia foi pioneira na adição de interatividade em páginas web
- Foi inventada em 1993 no NCSA, dona de um dos mais notáveis servidores web da época, o NCSA httpd
  - No futuro o seu código-fonte serviu base para o Apache HTTP Server

A ideia era possibilitar que um programa pudesse ser iniciado via página web e que seu resultados pudesse ser enviado ao cliente

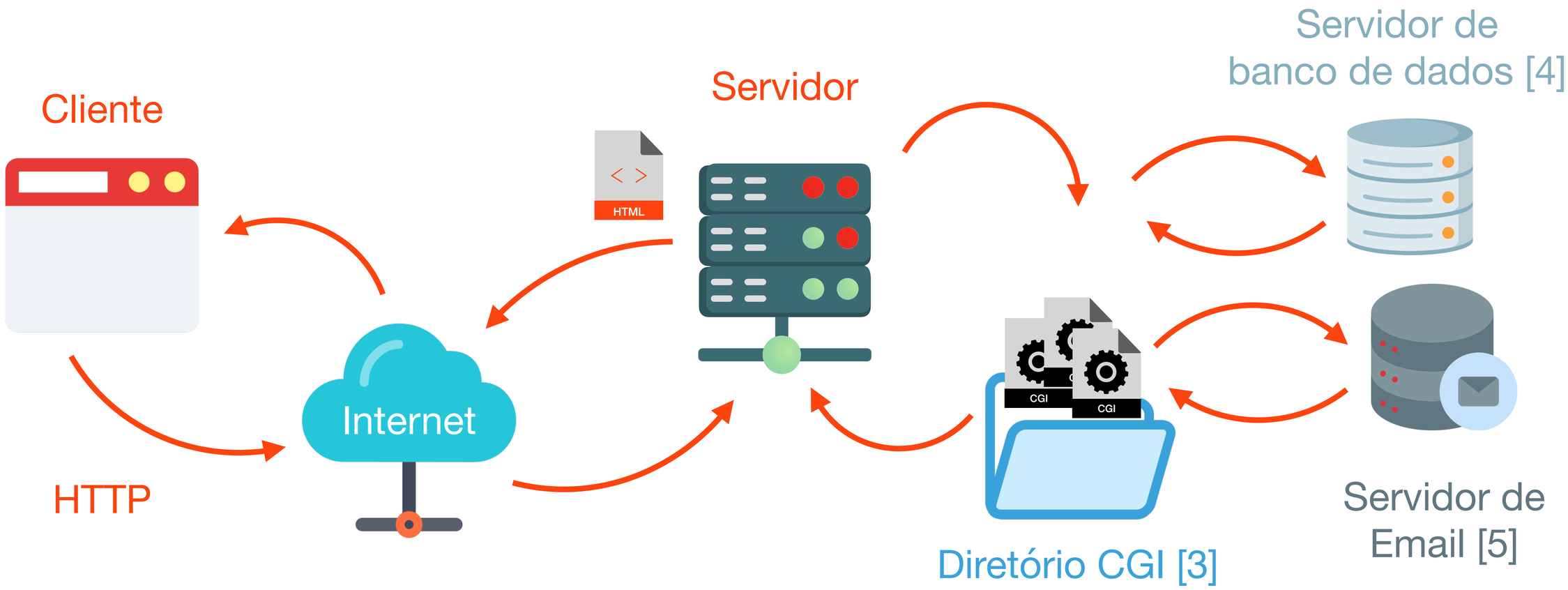
Um mundo dinâmico

Common Gateway Interface Programming (CGI)

- Geralmente, o servidor HTTP tinha uma pasta, que é designada como uma coleção de arquivos,
  - Estes arquivos são aqueles que podem ser enviados para os clientes
- CGI estende esse sistema, permitindo designar um diretório dentro da coleção de documento contendo scripts executáveis
  - Também conhecido como diretório CGI
  - Por exemplo, /usr/local/apache/htdocs/cgi-bin

Um mundo dinâmico

Common Gateway Interface Programming (CGI)



Um mundo dinâmico

Um Processo, Uma Requisição, Um Problema

1. **Requisição Chega:** O servidor (ex: Apache) recebe o pedido (/cgi-bin/script.pl)
2. **Criação do Processo (O Custo!)**: Para cada requisição, o servidor faz um fork ou spawn para inicializar um novo processo do zero para rodar o script
3. **Transferência de Dados:**
    1. Entrada: Dados da requisição (POST) vão para o STDIN do novo processo
    2. Contexto: Metadados HTTP (Headers) vão para as Variáveis de Ambiente do processo
4. **Resposta:** O script gera o HTML e o envia para o STDOUT (Saída Padrão)
5. **Destruição:** O processo CGI é imediatamente encerrado e destruído

Um mundo dinâmico

CGI e Formulários HTML

- Os scripts CGI eram usados em conjunto com formulários HTML
  - Remontam aos primórdios da Web e são anteriores à própria JavaScript
  - São o mecanismo por trás da primeira geração de aplicativos Web
  - Neles, a entrada do usuário é obtida em elementos de formulário
    - O envio do formulário remete essa entrada para o servidor
    - O servidor processa a entrada e gera uma nova página HTML (normalmente com novos elementos de formulário) para exibição pelo cliente

Um mundo dinâmico

Common Gateway Interface Programming (CGI)

- O script CGI podiam ser escritos em qualquer linguagem de programação
  - Inicialmente eram escritos em C
  - Script em Perl foram muitos populares

O CGI permitiu que as páginas da web fossem personalizadas para o usuário ou atualizadas com informações em tempo real (ex: contadores de acesso, formulários de contato, leituras de banco de dados).

Um mundo dinâmico

Exemplo de uso de CGI

```
<form action = "/cgi-bin/hello_get.cgi" method = "GET">
   First Name: <input type = "text" name = "first_name">  <br>

   Last Name: <input type = "text" name = "last_name">
   <input type = "submit" value = "Submit">
</form>
```

Um mundo dinâmico

Exemplo de uso de CGI

```
#!/usr/bin/perl

local ($buffer, @pairs, $pair, $name, $value, %FORM);
# Read in text
$ENV{'REQUEST_METHOD'} =~ tr/a-z/A-Z/;

if ($ENV{'REQUEST_METHOD'} eq "POST") {
   read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
} else {
   $buffer = $ENV{'QUERY_STRING'};
}

# Split information into name/value pairs
@pairs = split(/&/, $buffer);

foreach $pair (@pairs) {
   ($name, $value) = split(/=/, $pair);
   $value =~ tr/+/ /;
   $value =~ s/%(..)/pack("C", hex($1))/eg;
   $FORM{$name} = $value;
}
```

```
$first_name = $FORM{first_name};
$last_name  = $FORM{last_name};

print "Content-type:text/html\r\n\r\n";
print "<html>";
print "<head>";
print "<title>Hello - Second CGI Program</title>";
print "</head>";
print "<body>";
print "<h2>Hello $first_name $last_name - Second
CGI Program</h2>";
print "</body>";
print "</html>";

1;
```

Um mundo dinâmico

```perl
#!/usr/bin/perl

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

my $count_file = "acessos.txt";
my $current_count = 0;
open(my $fh, '+<', $count_file) or die "Nao foi possivel abrir $count_file: $!";

# Trava o arquivo para evitar que 2 processos o modifique simultaneamente
flock($fh, 2);
$current_count = <$fh>;
chomp($current_count); # Remove quebras de linha

$current_count = $current_count + 1;
seek($fh, 0, 0); # Volta ao início do arquivo (rewind)
```

Um mundo dinâmico

```
# Trunca o arquivo (limpa o conteúdo)
truncate($fh, 0);
# Escreve o novo valor
print $fh $current_count;
# Libera o bloqueio do arquivo
flock($fh, 8);
close $fh;


print "<html><head><title>Contador CGI</title></head><body>\n";
print "<h1>Visitantes únicos: $current_count</h1>\n";
print "<p>Este valor foi lido e atualizado diretamente no disco pelo script
Perl.</p>\n";
print "</body></html>\n";

exit 0; # Finaliza o script (e o processo)
```

Um mundo dinâmico

O Ponto Fraco do CGI (Escalabilidade)

- **Alto Custo de CPU:**
  - A inicialização e destruição de um novo processo é lenta e intensa para o sistema operacional
- **Falta de Persistência:**
  - Não havia como manter conexões com bancos de dados, cache ou sessões de usuário ativas na memória entre as requisições. Tudo tinha que ser refeito.
- **Consequência**:
  - Sites com alto tráfego rapidamente se tornavam lentos ou indisponíveis devido à sobrecarga do servidor web

Um mundo dinâmico

Exemplo de uso de CGI

- Apesar de não ser um componente dinâmico no navegador, como o JavaScript, possibilitou a execução de programas interativos na web dos tempos de 1993-1994
- No entanto, ainda em 1994, PHP, inicialmente chamada de *Personal Home Page Tools*, se destacou e acabou tomando o espaço do CGI

Processando dados de um formulário

Processando dados de um formulário

Prática

- Usando NodeJs crie um formulário de cadastro e verifique se as opções enviadas pelo usuário são válidas
  - Caso sejam válidas retorne uma página com a mensagem de sucesso
  - Caso contrário, retorne a página do formulário indicando o local do erro
- Implemente o envio dos dados do formulário via GET e POST

Processando dados de um formulário

Obtendo parâmetros de uma requisição GET

- Precisamos utilizar acessar o objeto a URL dentro do objeto que encapsula a requisição HTTP

```
const server = http.createServer(req: IncomingMessage , res: ServerResponse) => {

  const baseURL = `http://${req.headers.host}/`
  const parsedUrl = new URL(req.url, baseURL)

  res.statusCode = 200
  res.setHeader('Content-Type', 'text/html; charset=utf-8')

const name = parsedUrl.searchParams.get('name")
res.end(`<html><head></head><body> Contéudo </body></html>`)
});
```

Parseando a URL

Nome do parâmetro

Processando dados de um formulário

Obtendo parâmetros de uma requisição POST

- Precisamos utilizar acessar o objeto a URL dentro do objeto que encapsula a requisição HTTP

```
const server = http.createServer(async(req: IncomingMessage, res: ServerResponse)) => {

  const baseURL = `http://${req.headers.host}/`
  const parsedUrl = new URL(req.url, baseURL)

  const chunks = [];
  for await (const chunk of req) {
    chunks.push(chunk);
  }
const data = Buffer.concat(chunks);
res.end(data.toString())
});
```

Navegando na stream de dados

Referências

- Web Design History Timeline
- Common Gateway Interface
- Build Your Own Web Server From Scratch In Node.JS
- How To Create a Web Server in Node.js with the HTTP Module
- 1993: CGI Scripts and Early Server-Side Web Programming
- PERL and CGI Tutorial

Por hoje é só