

#### 知识点1【CGI的概述】（了解）

##### 1、cgi的概念

##### 2、CGI编程注意点

#### 知识点2【GET的异步方式】（了解）

#### 知识点2【POST的异步方式】（了解）

#### 知识点3【GET的同步方式】（了解）

#### 知识点3【POST的同步方式】（了解）

## 知识点1【CGI的概述】（了解）

### 1、cgi的概念

CGI 是通用网关接口(Common Gateway Interface);是 HTTP 服务器与其它程序进行“交谈”的工具 通过 CGI 接口就能在服务器端调用其他程序（**cgi程序**）

### 2、CGI编程注意点

1、CGI 可以用任何一种语言编写，只要这种语言具有标准**输入**、标准**输出**、和获取环境变量

(1) CGI 程序通过标准输入(stdin)、标准输出(stdout)实现与 web 服务器间信息的传递

(2) 环境变量为 Web 服务器和 CGI 接口之间约定的,用来向 CGI 程序传

2、CGI 传送给 Web 服务器的信息可以用各种格式,通常是以 HTML 文本或者 XML 文本的形式

(1) 传输 **HTML 文本** 第一行输出的内容必须是“**content-type:text/html**”

(2) 传输 **XML 文本** 第一行输出的内容必须是“**content-type:text/xml**”

(3) 还有其他的一些格式：JIF(image/gif)、JPEG(image/jpeg)、AVI(video/avi)

3、两个重要的 CGI 环境变量

(1) **QUERY\_STRING**：在浏览器端以 **GET** 的方法输入的数据，数据的内容就是 url 问号后的数据

(2) **CONTENT\_LENGTH**：在浏览器端以 **POST** 方法输入的数据的字节数，数据的内容通过标准输入获取

## 知识点2 【GET的异步方式】（了解）

index.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <script type="text/javascript" src="11_js.js"></script>
7     <title>Document</title>
8 </head>
9
10 <body>
11     Data1:<input type="text" id="data1">
12     <br>
13     Data2:<input type="text" id="data2">
14     <br>
15     结果:<input type="text" id="res">
16     <br>
17     <input type="button" value="相加" onclick="my_calc(1)">
18     <input type="button" value="相减" onclick="my_calc(0)">
19 </body>
20
21 </html>
```

11\_js.js

```
1 function getXMLHttpRequest() {
2     var xmlhttp = null;
3     if (window.XMLHttpRequest) //自动检测当前浏览器的版本, 如果是 IE5.0 以上的高版本的浏览器
4     {
5         // code for IE7+, Firefox, Chrome, Opera, Safari
6         xmlhttp = new XMLHttpRequest(); //创建请求对象
7     }
8     else //如果浏览器是低版本的
9     {
10        // code for IE6, IE5
11        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP"); //创建请求对象
12    }
13    return xmlhttp; //返回请求对象
14 }
```

```
14 function my_calc(arg) {
15
16     //获取data1 data2的数据
17     var data1 = document.getElementById("data1").value;
18     var data2 = document.getElementById("data2").value;
19
20     //判断data1 data2必须是数值
21     if (isNaN(data1) || isNaN(data2)) {
22         alert("请输入数值");
23         document.getElementById("data1").value = "";
24         document.getElementById("data2").value = "";
25     }
26     else {
27
28         //创建对象
29         var xmlhttp;
30         xmlhttp = getXMLHttpRequest();
31
32         //设置回调
33         xmlhttp.onreadystatechange = function () {
34             //4交换完成 200是文件找到 才会获取服务器的应答
35             if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
36                 //获取服务器的应答
37                 var ret = xmlhttp.responseText;
38                 //局部更新网页内容
39                 document.getElementById("res").value = ret;
40                 alert(ret);
41             }
42         }
43
44         //构建url
45         var url = "/cgi-bin/calc.cgi?";
46         if (arg == 1) {
47             // "/cgi-bin/calc.cgi?10+20"
48             url = url + data1 + "+" + data2;
49         }
50         else if (arg == 0) {
51             // "/cgi-bin/calc.cgi?10-20"
52             url = url + data1 + "-" + data2;
53         }
54     }
55 }
```

```

54
55
56     //open创建请求 GET 异步true
57     xmlhttp.open("GET", url, true);
58
59     //send发送请求 GET方式send不需要参数 因为参数在url的? 后面
60     xmlhttp.send();
61 }
62 }

```

## 11\_calc.c

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  int main(int argc, char const *argv[])
4  {
5      //1、规定printf输出格式
6      printf("content-type:text/html\n\n");
7
8      #if 1
9          //2、获取服务器的数据（?号后面的数据）（GET方式）
10         //msg格式: 10+20 10-20
11         char *msg = getenv("QUERY_STRING");
12         int data1 = 0, data2 = 0;
13         char ch = 0;
14         sscanf(msg, "%d%c%d", &data1, &ch, &data2);
15
16         if (ch == '+')
17         {
18             //将结算结果交给服务器
19             printf("%d\n", data1 + data2);
20         }
21         else if (ch == '-')
22         {
23             //将结算结果交给服务器
24             printf("%d\n", data1 - data2);
25         }
26     #endif
27     return 0;
28 }

```

## 知识点2 【POST的异步方式】（了解）

## index.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <script type="text/javascript" src="12_js.js"></script>
7     <title>Document</title>
8 </head>
9
10 <body>
11     Data1:<input type="text" id="data1">
12     <br>
13     Data2:<input type="text" id="data2">
14     <br>
15     结果:<input type="text" id="res">
16     <br>
17     <input type="button" value="相加" onclick="my_calc(1)">
18     <input type="button" value="相减" onclick="my_calc(0)">
19 </body>
20
21 </html>
```

## 12\_js.js

```
1 function getXMLHttpRequest() {
2     var xmlhttp = null;
3     if (window.XMLHttpRequest) //自动检测当前浏览器的版本, 如果是 IE5.0 以上的
        高版本的浏览器
4     { // code for IE7+, Firefox, Chrome, Opera, Safari
5         xmlhttp = new XMLHttpRequest(); //创建请求对象
6     }
7     else //如果浏览器是低版本的
8     { // code for IE6, IE5
9         xmlhttp = new ActiveXObject("Microsoft.XMLHTTP"); //创建请求对象
10    }
11    return xmlhttp; //返回请求对象
12 }
13
14 function my_calc(arg) {
15
16     //获取data1 data2的数据
```

```
17     var data1 = document.getElementById("data1").value;
18     var data2 = document.getElementById("data2").value;
19
20     //判断data1 data2必须是数值
21     if (isNaN(data1) || isNaN(data2)) {
22         alert("请输入数值");
23         document.getElementById("data1").value = "";
24         document.getElementById("data2").value = "";
25     }
26     else {
27
28         //创建对象
29         var xmlhttp;
30         xmlhttp = getXMLHttpRequest();
31
32         //设置回调
33         xmlhttp.onreadystatechange = function () {
34             //4交换完成 200是文件找到 才会获取服务器的应答
35             if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
36                 //获取服务器的应答
37                 var ret = xmlhttp.responseText;
38                 //局部更新网页内容
39                 document.getElementById("res").value = ret;
40                 alert(ret);
41             }
42         }
43
44         //构建url POST方式cgi后不能有?
45         var url = "/cgi-bin/calc.cgi";
46         //定义传递给cgi数据
47         var data = "";
48         if (arg == 1) {
49             // data ="10+20"
50             data = data1 + "+" + data2;
51         }
52         else if (arg == 0) {
53             // data="10-20"
54             data = data1 + "-" + data2;
55         }
56     }
```

```

57         //open创建请求 POST 异步true
58         xmlhttp.open("POST", url, true);
59
60         //send发送请求 POST方式send的参数就是data
61         xmlhttp.send(data);
62     }
63 }

```

## 12\_calc.c

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  int main(int argc, char const *argv[])
4  {
5      //1、规定printf输出格式
6      printf("content-type:text/html\n\n");
7
8      #if 1
9          //2、POST方式 获取服务器的数据长度
10         char *len = getenv("CONTENT_LENGTH");
11         //3、通过len使用fgets获取内容
12         char msg[128] = "";
13         fgets(msg, atoi(len) + 1, stdin);
14
15         //msg格式: 10+20    10-20
16         int data1 = 0, data2 = 0;
17         char ch = 0;
18         sscanf(msg, "%d%c%d", &data1, &ch, &data2);
19
20         if (ch == '+')
21         {
22             //将结算结果交给服务器
23             printf("%d\n", data1 + data2);
24         }
25         else if (ch == '-')
26         {
27             //将结算结果交给服务器
28             printf("%d\n", data1 - data2);
29         }
30     #endif
31     return 0;
32 }

```

Data1:	<input type="text" value="30"/>
Data2:	<input type="text" value="10"/>
结果:	<input type="text" value="20"/>
<input type="button" value="相加"/>	<input type="button" value="相减"/>

## 知识点3 【GET的同步方式】（了解）

index.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <script type="text/javascript" src="13_js.js"></script>
7     <title>Document</title>
8 </head>
9
10 <body>
11     Data1:<input type="text" id="data1">
12     <br>
13     Data2:<input type="text" id="data2">
14     <br>
15     结果: <input type="text" id="res">
16     <br>
17     <input type="button" value="相加" onclick="my_calc(1)">
18     <input type="button" value="相减" onclick="my_calc(0)">
19 </body>
20
21 </html>
```

13\_js.js

```
1 function getXMLHttpRequest() {
2     var xmlhttp = null;
3     if (window.XMLHttpRequest) //自动检测当前浏览器的版本, 如果是 IE5.0 以上的高版本的浏览器
4     { // code for IE7+, Firefox, Chrome, Opera, Safari
5         xmlhttp = new XMLHttpRequest(); //创建请求对象
```



```
6     }
7     else////如果浏览器是底版本的
8     {
9         xmlhttp = new XMLHttpRequest("Microsoft.XMLHTTP");//创建请求对象
10    }
11    return xmlhttp;//返回请求对象
12 }
13
14 function my_calc(arg) {
15
16    //获取data1 data2的数据
17    var data1 = document.getElementById("data1").value;
18    var data2 = document.getElementById("data2").value;
19
20    //判断data1 data2必须是数值
21    if (isNaN(data1) || isNaN(data2)) {
22        alert("请输入数值");
23        document.getElementById("data1").value = "";
24        document.getElementById("data2").value = "";
25    }
26    else {
27
28        //创建对象
29        var xmlhttp;
30        xmlhttp = getXMLHttpRequest();
31
32        //构建url
33        var url = "/cgi-bin/calc.cgi?";
34        if (arg == 1) {
35            // "/cgi-bin/calc.cgi?10+20"
36            url = url + data1 + "+" + data2;
37        }
38        else if (arg == 0) {
39            // "/cgi-bin/calc.cgi?10-20"
40            url = url + data1 + "-" + data2;
41        }
42
43
44        //open创建请求 GET 同步false
45        xmlhttp.open("GET", url, false);
46    }
```

```

47      //send发送请求 GET方式send不需要参数 因为参数在url的? 后面
48      xmlhttp.send();
49
50      //4交换完成 200是文件找到 才会获取服务器的应答
51      if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
52          //获取服务器的应答
53          var ret = xmlhttp.responseText;
54          //局部更新网页内容
55          document.getElementById("res").value = ret;
56          alert(ret);
57      }
58  }
59 }

```

### 13\_calc.c

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  int main(int argc, char const *argv[])
4  {
5      //1、规定printf输出格式
6      printf("content-type:text/html\n\n");
7
8      #if 1
9          //2、获取服务器的数据（?号后面的数据）（GET方式）
10         //msg格式: 10+20  10-20
11         char *msg = getenv("QUERY_STRING");
12         int data1 = 0, data2 = 0;
13         char ch = 0;
14         sscanf(msg, "%d%c%d", &data1, &ch, &data2);
15
16         if (ch == '+')
17         {
18             //将结算结果交给服务器
19             printf("%d\n", data1 + data2);
20         }
21         else if (ch == '-')
22         {
23             //将结算结果交给服务器
24             printf("%d\n", data1 - data2);
25         }
26     #endif

```

```
27     return 0;
28 }
```

## 知识点3 【POST的同步方式】（了解）

index.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <script type="text/javascript" src="14_js.js"></script>
7      <title>Document</title>
8  </head>
9
10 <body>
11     Data1:<input type="text" id="data1">
12     <br>
13     Data2:<input type="text" id="data2">
14     <br>
15     结果:<input type="text" id="res">
16     <br>
17     <input type="button" value="相加" onclick="my_calc(1)">
18     <input type="button" value="相减" onclick="my_calc(0)">
19 </body>
20
21 </html>
```

14\_js.js

```
1  function getXMLHttpRequest() {
2      var xmlhttp = null;
3      if (window.XMLHttpRequest) //自动检测当前浏览器的版本, 如果是 IE5.0 以上的高版本的浏览器
4          { // code for IE7+, Firefox, Chrome, Opera, Safari
5              xmlhttp = new XMLHttpRequest(); //创建请求对象
6          }
7      else //如果浏览器是低版本的
8          { // code for IE6, IE5
9              xmlhttp = new ActiveXObject("Microsoft.XMLHTTP"); //创建请求对象
10         }
11      return xmlhttp; //返回请求对象
12 }
```

```
13
14 function my_calc(arg) {
15
16     //获取data1 data2的数据
17     var data1 = document.getElementById("data1").value;
18     var data2 = document.getElementById("data2").value;
19
20     //判断data1 data2必须是数值
21     if (isNaN(data1) || isNaN(data2)) {
22         alert("请输入数值");
23         document.getElementById("data1").value = "";
24         document.getElementById("data2").value = "";
25     }
26     else {
27
28         //创建对象
29         var xmlhttp;
30         xmlhttp = getXMLHttpRequest();
31
32         //构建url POST方式cgi后不能有?
33         var url = "/cgi-bin/calc.cgi";
34         //定义传递给cgi数据
35         var data = "";
36         if (arg == 1) {
37             // data ="10+20"
38             data = data1 + "+" + data2;
39         }
40         else if (arg == 0) {
41             // data="10-20"
42             data = data1 + "-" + data2;
43         }
44
45         //open创建请求 POST 同步false
46         xmlhttp.open("POST", url, false);
47
48         //send发送请求 POST方式send的参数就是data
49         xmlhttp.send(data);
50
51         //4交换完成 200是文件找到 才会获取服务器的应答
52         if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
53             //获取服务器的应答
```

```

54         var ret = xmlhttp.responseText;
55         //局部更新网页内容
56         document.getElementById("res").value = ret;
57         alert(ret);
58     }
59 }
60 }

```

## 14\_calc.c

```

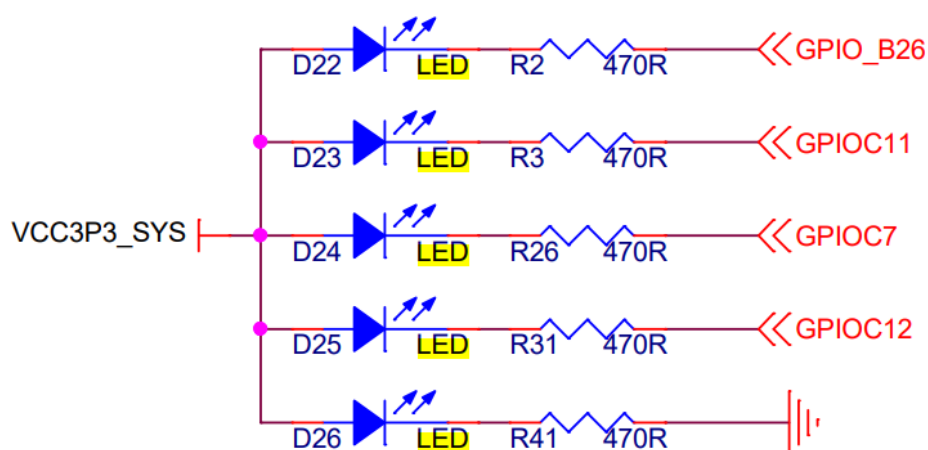
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main(int argc, char const *argv[])
4  {
5      //1、规定printf输出格式
6      printf("content-type:text/html\n\n");
7
8      #if 1
9          //2、POST方式 获取服务器的数据长度
10         char *len = getenv("CONTENT_LENGTH");
11         //3、通过len使用fgets获取内容
12         char msg[128] = "";
13         fgets(msg, atoi(len) + 1, stdin);
14
15         //msg格式: 10+20  10-20
16         int data1 = 0, data2 = 0;
17         char ch = 0;
18         sscanf(msg, "%d%c%d", &data1, &ch, &data2);
19
20         if (ch == '+')
21         {
22             //将结算结果交给服务器
23             printf("%d\n", data1 + data2);
24         }
25         else if (ch == '-')
26         {
27             //将结算结果交给服务器
28             printf("%d\n", data1 - data2);
29         }
30     #endif
31     return 0;
32 }

```

Data1:	30
Data2:	10
结果:	20

## 知识点4 【LED】 (了解)

### 1、led的原理图：



**LED**

原理图分析：0亮 1灭

A53一个GPIO组 (A,B,C, D) ,每一组32管脚, A组 (1~32) , B组 (33~64) , 所以 GPIO\_B26 ==58编号

## 2、GPIO控制

### 1、使能58号IO口

```
1 echo 58 > /sys/class/gpio/export
```

### 2、设置IO口的工作方向

```
1 echo out > /sys/class/gpio/gpio58/direction
```

### 3、设置IO的高低电平

```
1 echo 1 > /sys/class/gpio/gpio58/value  设置为高电平 灭
2 echo 0 > /sys/class/gpio/gpio58/value  设置为低电平 亮
```