# 简单Web服务器

实现一个简单的 Web 服务器 myhttpd。能够给浏览器提供服务，供用户借助浏览器访问服务器主机中的文件。

## 超文本标记语言HTML

超文本标记语言（Hyper Text Mark-up Language HTML）是构成网页文档的主要语言。可以说明文字、图形、动画、声音、表格、链接等。在计算机中以.html、.htm作为扩展名，可以被浏览器访问。

### 简介

#### HTML特点

语法非常简洁、比较松散，以相应的英语单词关键字进行组合

html标签不区分大小写

大多数标签是成对出现的, 有开始, 有结束.<html>…</html>

不成对出现的称之为短标签 <br/> <hr/>

#### 标签的属性和值

属性="属性值"

<font color="red">hello, world</font>

属性值建议加引号。(语法上 双, 单引号, 不加都可以)

#### html组成部分

html文件结构包括头(head)和主体(body)两大部分，头部描述的是浏览器，所需信息，而主题则包含所要说明的具体内容。可在Linux下用浏览器打开一个网页，使用Ctrl-u查看该网页的html文件形式。Ctrl-w恢复。

<!doctype html> 声明文档类型

<html>文档的头部和主体内容 </html> 根标签

<head> 文档的头部信息 </head> 头部标记只能有一对

<title>显示在浏览器窗口的标题栏中“网页名称”</title> 位于<head> 标记之内

<body></body> 主体标记位于<html>之内,<head>标记之后

简单的错误网页形式，如：

<!doctype html>

<html>

<head><title>404 Not Found</title></head>

<body bgcolor="#cc99cc">

<h4>404 Not Found</h4>

File not found.

<hr>

<address>

<a href="<http://www.itcast.cn/>">sina</a>

</address>

</body>

</html>

#### 注释:

<!-- 我是一个html注释 -->

### 常用标签

学习html，主要是标签的使用，能熟练的掌握各种常用标签的功能、特性，可以大大提高开发速度和效率。初学的同学可以从以下标签入手学习。

#### 文本和标题标签

* **标题标签:**

<h1></h1> // 最大 只有一个，搜索引擎优化: seo

<h2></h2>

...

<h6></h6> // 最小 1-6依次变小, 自动换行

* **文本标签**

<font></font>

属性:

color: 文字颜色

表示方式:

英文单词: red green blue......

使用16进制的形式表示颜色: #ffffff -- (rgb)

使用rgb(255, 255, 0)

size: 文字大小

范围 1 – 7。7最大、1最小

* **文本格式化标签**

加粗

<strong></strong> 或 <b></b>

工作里尽量使用strong

倾斜

<em></em> 或 <i></i>

工作里尽量使用em

删除线

<del></del> 或 <s></s>

工作里尽量使用del

下划线

<ins></ins> 或 <u></u>

工作里尽量使用ins

* **段落：**

<p> xxx </p>

特点: 上下自动生成空白行

* **块容器:**

<div>This is a div element.</div>

用于没有语义含义的内容的块级容器(或网页的"划分")。

属性: 对齐方式：

align

left

center

right

* **换行 <br/>**
* **水平线<hr/>**

属性:

color: 3种表示方法

size: 1-7

<hr color="red" size="3"/>

#### 列表标签

* **无序列表**

标签

<ul>

<li></li> 列表项

<li></li>

</ul>

属性: type

实心圆圈: disc -- 默认

空心圆圈: circle

小方块: square

* **有序列表**

标签

<ol>

<li></li> 列表项

<li></li>

</ol>

属性:

type -- 序号

1 -- 默认

a

A

i -- 罗马数字(小)

I -- 罗马数字(大)

start

从序号的什么位置开始表示

* **自定义列表**

标签

<dl>

<dt></dt> 小标题

<dd></dd> 解释标题

<dd></dd> 解释标题

</dl>

#### 图片标签

<img src="bird.gif" alt="图片加载失败" title="小鸟飞翔！" width="300" height="200" />

* **属性:**

src: 图片的来源 必写属性

alt: 替换文本 图片不显示的时候显示的文字

title: 提示文本 鼠标放到图片上显示的文字

width: 图片宽度

height: 图片高度

* **注意：**

图片没有定义宽高的时候，图片按照百分之百比例显示

如果只更改图片的宽度或者高度，图片等比例缩放。

#### 超链接标签

<a **href**="http://jd.com" **title**="A dog" **target**="\_blank"> 超链接 </a>

* **属性:**

href: 去往的路径（跳转的页面）必写属性

title: 提示文本, 鼠标放到链接上显示的文字

target取值：

\_self: 默认值 在自身页面打开（关闭自身页面，打开链接页面）

\_blank: 打开新页面 （自身页面不关闭，打开一个新的链接页面）

示例:

<a href="http://www.baidu.com">百度一下</a>

* **锚链接**

事先定义一个锚点标签: <p id="top">

超链接到锚点: <a herf="#top"> 回到顶点 </a>

#### 表格标签

* **<table> </table>**

属性:

border -- 表格线, 宽度1-7

bordercolor -- 表格线颜色

width

height

* **<tr> -- 行**

属性

align -- 对齐方式

center

left

right

* **<td> -- 单元格(列)**

对其属性设置同tr

* **示例:**

<table border=1>

<tr>

<td></td> 第一列

<td></td> 第二列

</tr>

<tr>

<td></td>

<td></td>

</tr>

<tr>

<td></td>

<td></td>

</tr>

</table>

### 自学网址：

<http://www.w3school.com.cn/html/index.asp>

<http://www.runoob.com/>

## HTTP协议基础

HTTP，超文本传输协议（ HyperText Transfer Protocol ）。[互联网](https://baike.baidu.com/item/%E4%BA%92%E8%81%94%E7%BD%91" \t "_blank)应用最为广泛的一种[网络应用层协议](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E5%8D%8F%E8%AE%AE)。它可以减少网络传输，使浏览器更加高效。

通常HTTP消息包括客户机向服务器的请求消息和服务器向客户机的响应消息。

### 请求消息(Request)

浏览器 —> 发给 —> 服务器。主旨内容包含4部分：

* **请求行:** 说明请求类型, 要访问的资源, 以及使用的http版本
* **请求头:** 说明服务器要使用的附加信息
* **空 行:** 必须！, 即使没有请求数据
* **请求数据**: 也叫主体, 可以添加任意的其他数据

以下是浏览器发送给服务器的http**协议头**内容举例, 注意：9行的空行(\r\n)也是协议头的一部分：

**1. GET /hello.c HTTP/1.1**

2. Host: localhost:2222

3. User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:24.0) Gecko/201001 01 Firefox/24.0

4. Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

5. Accept-Language: zh-cn,zh;q=0.8,en-us;q=0.5,en;q=0.3

6. Accept-Encoding: gzip, deflate

7. Connection: keep-alive

8. If-Modified-Since: Fri, 18 Jul 2014 08:36:36 GMT

**9.**

### 响应消息(Response)

服务器 —> 发给 —> 浏览器。主旨内容包含4部分：

* **状态行:** 包括http协议版本号, 状态码, 状态信息
* **消息报头:** 说明客户端要使用的一些附加信息
* **空 行:** 必须!
* **响应正文:** 服务器返回给客户端的文本信息

以下是经服务器按照http协议，写回给浏览器的内容举例，1~9行是**协议头**部分。注意：9行\r\n的空行不可忽略

**1. HTTP/1.1 200 Ok**

2. Server: xhttpd

3. Date: Fri, 18 Jul 2014 14:34:26 GMT

**4. Content-Type: text/plain; charset=iso-8859-1 (必选项)**

5. Content-Length: 32 （ 要么不写 或者 传-1， 要写务必精确 ！ ）

6. Content-Language: zh-CN

7. Last-Modified: Fri, 18 Jul 2014 08:36:36 GMT

8. Connection: close

**9.**

10. #include <stdio.h>

11.

12. int main(void)

13. {

14. printf(“ Welcome to itcast ... \n");

15.

16. return 0;

17. }

### HTTP请求方法

* **GET**

请求指定的页面信息，并返回实体主体。

* **POST**

向指定资源提交数据进行处理请求（例如提交表单或者上传文件）。数据被包含在请求体中。POST请求可能会导致新的资源的建立和/或已有资源的修改。

* **HEAD**

类似于get请求，只不过返回的响应中没有具体的内容，用于获取报头

* **PUT**

从客户端向服务器传送的数据取代指定的文档的内容。

* **DELETE**

请求服务器删除指定的页面。

* **CONNECT**

HTTP/1.1协议中预留给能够将连接改为管道方式的代理服务器。

* **OPTIONS**

允许客户端查看服务器的性能。

* **TRACE**

回显服务器收到的请求，主要用于测试或诊断。

### HTTP常用状态码

状态代码有三位数字组成，第一个数字定义了响应的类别，共分五种类别:

* 1xx：指示信息--表示请求已接收，继续处理
* 2xx：成功--表示请求已被成功接收、理解、接受
* 3xx：重定向--要完成请求必须进行更进一步的操作
* 4xx：客户端错误--请求有语法错误或请求无法实现
* 5xx：服务器端错误--服务器未能实现合法的请求
* **常见状态码：**

200 OK 客户端请求成功

400 Bad Request 客户端请求有语法错误，不能被服务器所理解

401 Unauthorized 请求未经授权，

403 Forbidden 服务器收到请求，但是拒绝提供服务

404 Not Found 请求资源不存在，eg：输入了错误的URL

500 Internal Server Error 服务器发生不可预期的错误

503 Server Unavailable 服务器当前不能处理客户端的请求。

### 常见网络文件类型：

普通文件: text/plain; charset=iso-8859-1

\*.html : text/html; charset=iso-8859-1

\*.jpg : image/jpeg

\*.gif : image/gif

\*.png: image/png

\*.wav: audio/wav

\*.avi : video/x-msvideo

\*.mov : video/quicktime

\*.mp3 : audio/mpeg

charset=iso-8859-1 西欧的编码，说明网站采用英文编码

charset=gb2312 说明网站采用的编码是简体中文

charset=utf-8 代表世界通用的语言编码；可以用到中文、韩文、日文等世界上所有语言编码

charset=euc-kr 说明网站采用的编码是韩文；

charset=big5 说明网站采用的编码是繁体中文；

### 自学网址：

http://www.runoob.com/http/http-tutorial.html

## 正则表达式：

### sscanf 函数

函数描述:

读取格式化的字符串中的数据。

函数原型:

**int sscanf(const char \*buffer，const char \*format, [ argument ] ... )；**

1. 取到 指定字符为止的字符串。如在下例中，取遇到空格为止字符串。

sscanf("123456 abcdedf", "%[^ ]", buf);

printf("%s\n", buf);

结果为：123456

1. 取 仅包含指定字符集的字符串。如在下例中，取仅包含1到9和小写字母的字符串。

sscanf("123456abcdedfBCDEF", "%[1-9a-z]", buf);

printf("%s\n", buf);

结果为：123456abcdedf

1. 取到 指定字符集为止的字符串。如在下例中，取遇到大写字母为止的字符串。

sscanf("123456abcdedfBCDEF", "%[^A-Z]", buf);

printf("%s\n", buf);

结果为：123456abcdedf

### 自学网址：

<http://deerchao.net/tutorials/regex/regex.htm>

<http://www.jb51.net/tools/regexsc.htm>

## 开发注意事项

### 浏览器请求ico

准备一个favicon.ico 文件放置到 服务器提供访问的资源目录中。

浏览器在请求图片的同时，会请求一个ico图标，用于浏览器<title>标签文字部分前端的小图标显示。

这个ico的文件名固定——favicon.ico。因此，自行准备一个ico文件，放置于服务器提供给浏览器访问的目标目录即可。

### 容错处理

**返回值一定要检查！！！**

尤其是 read、write、recv、send、recvfrom、sendto 函数，返回的errno为 **EINTR**和**EAGAIN**时，不代表是一个错误，但会严重影响程序运行结果！通常使用continue来处理这种情况即可。

### 快捷遍历目录scandir()

服务器端，可以使用文件操作时“递归遍历目录的”源码，实现遍历目录内文件名，回显给浏览器。另外标准C库中，提供了scandir函数，可以便捷的实现该功能。函数原型如下：

**int scandir(const char \*dirp, struct dirent \*\*\*namelist,**

**int (\*filter)(const struct dirent \*),**

**int (\*compar)(const struct dirent \*\*, const struct dirent \*\*));**

scandir(待操作的目录，&子目录项列表数组，过滤器(通常NULL)，alphasort）；

**dirp:** 待访问的目录名称

**struct dirent类型：**（参考readdir()函数）

struct dirent {

ino\_t **d\_ino;** /\* inode number \*/

off\_t d\_off; /\* not an offset; see NOTES \*/

unsigned short d\_reclen; /\* length of this record \*/

unsigned char d\_type;

char **d\_name[256]**; /\* filename \*/

};

**compar:**参数取如下函数即可:

**int alphasort(const void \*a, const void \*b);** 默认是由的排序算法。

**调用：** struct dirent\*\* namelist;

int num = scandir(dirname, &namelist, NULL, alphasort);

for (i = 0; i < num; ++i)

char\* name = namelist[i]->d\_name;

### 含有汉字的文件

访问带有汉字的文件时，将这个URL复制到新的 浏览器地址栏中，可以看到它所对应的在浏览器中使用的字符编码。

如：汉字 “大学”对应的编码为：%E5%A4%A7%E5%AD%A6

在命令行使用 unicode 命令（如不存在，使用apt-get命令安装）可查看汉字对应的Unicode编码。如： unicode 大学 --> UTF-8: e5 a4 a7 , UTF-8: e5 ad a6

因此，在访问带有汉字的文件时，应该在**服务器回发**数据给浏览器时进行**“编码”**操作，在**浏览器请求**资源目录的汉字文件时进行**“解码”**操作。

## 借助telnet调试

可使用 telnet 命令，借助IP和port，模拟浏览器行为，在终端中对访问的服务器进行调试，方便查看服务器回发给浏览器的http协议数据。使用方式如：

命令行键入：telnet 127.0.0.1 9999 回车，手动写入http请求协议头，如：

GET /hello.c http/1.1 回车

此时，终端中可查看到服务器回发给浏览器的http应答协议及数据内容。可根据该信息进行调试。