

1. xml

○ 特点:

- 有一个文件头

`<?xml version="1.0" encoding="ut-8"?>`

- 由一系列的标签组成: 逻辑上是树结构

- 根标签只有一个

- 一般成对出现`<hello></hello>`

- 不成对: `<hello/>`

- 标签区分大小写

- 标签可以设置属性

``

- 注释:

`<!-- XXXX -->`

○ 中国

- XX

- XXX

- XX

○ ./a.out

- ldd a.out

2. Json

○ json数组

- [字符串, 布尔, 整形, json数组, json对象]

○ json对象

```
{  
    key:value,  
    key1:value  
}
```

键值: 不能重复, 必须是字符串

value: 字符串, 布尔, 整形, json数组, json对象

- cJSON

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

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

1. HTML简介

- HTML, Hyper Text Markup Language, 超文本标记语言。
- 在计算机中以.html、.htm作为扩展名。
- 可以被浏览器访问,就是经常见到的网页。

2. 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
 -
 <hr/>

3. 标签中的属性和属性值

- 属性="属性值"
 - hello, world
 - 属性值建议加引号,(双,单引号,不加都可以)

4. html组成部分

- <!doctype html> 声明文档类型
- <html> 文档的头部好和主体内容 </html> 根标记
- <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
- 例:

```
<html>
  <head>
    <title>这是一个标题</title>
  </head>

  <body>
    <font color="red" size="5">hello, world</font>
```

```
</body>  
</html>
```

5. 注释:

- <!-- 我是一个html注释 -->

文字和标题标签

1. 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

2. 文本标签

- ``
 - 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

3. 文本格式化标签

- 文本加粗标签
 - ``
 - ``
 - 工作里尽量使用strong
- 文本倾斜标签
 - ``
 - `<i></i>`
 - 工作里尽量使用em

- 删除线标签
 - ``
 - `<s></s>`
 - 工作里尽量使用del

- 下划线标签（插入文本）
 - `<ins></ins>`
 - `<u></u>`
 - 工作里尽量ins

4. 段落：

- `<p>xxx</p>`
 - 特点：
 - 上下自动生成空白行

5. 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

6. 换行

- `
`

7. 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

1. 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

2. 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

3. 自定义列表

- 标签

<d1>

<dt></dt> 小标题

<dd></dd> 解释标题

<dd></dd> 解释标题

</d1>

图片标签

图片标签

- ``

- 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

1. 超链接标签

- `超链接`

- 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

- _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:

- `百度一下`

2. 锚链接

- 先定义一个锚点: `<p id="sd">`

- 超链接到锚点: `回到顶点`

表格标签

1. 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=>`

```
<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>
```

http协议

http协议 - 应用层

- 请求消息(Request) - 浏览器给服务器发

○ 四部分: 请求行, 请求头, 空行, 请求数据

- 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: 空行是必须要有的, 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

○ 例:

GET /3.txt HTTP/1.1

请求行

/: 资源目录的根目录

三部分内容由空格间隔

Host: localhost:2222

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:24.0) Gecko/20100101 Firefox/24.0

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

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

Accept-Encoding: gzip, deflate

Connection: keep-alive

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

请求头: 由键值对构成的

www.baidu.com

xx.tar.gz
x.tar.bz2

请求数据

换行: \r\n

- 响应消息(Response) - 服务器给浏览器发

○ 四部分: 状态行, 消息报头, 空行, 响应正文

- 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: 空行是必须要有的
- 响应正文: 服务器返回给客户端的文本信息

○ 例:

HTTP/1.1 200 Ok

状态行: 协议版本号 状态码 状态信息

Server: micro_httpd

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

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

告诉浏览器发送的数据是什么类型

Content-Length: 32

发送的数据的长度

Content-Language: zh-CN

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

Connection: close

消息报头

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    printf("hello world!\n");
```

```
    return 0;
```

```
}
```

- HTTP1.1的五种请求方法

○ 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，这个状态代码必须和WWW-Authenticate报头域一起使用
- 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，一段时间后可能恢复正常

http使用get和post请求数据

2017年6月11日 15:15

使用get方法请求数据:

GET /3.txt HTTP/1.1

Host: localhost:2222

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:24.0)

Gecko/201001 01 Firefox/24.0

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

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

Accept-Encoding: gzip, deflate

Connection: keep-alive

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

空行

请求数据(可以为空)

使用post方法请求数据:

POST HTTP/1.1

Host: localhost:2222

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:24.0)

Gecko/201001 01 Firefox/24.0

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

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

Accept-Encoding: gzip, deflate

Connection: keep-alive

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

空行

user=詹姆斯&pwd=James&sex=男

浏览器地址栏:

192.168.1.115/hello.c

浏览器封装一个http请求协议

get /hello.c http/1.1

key:value

key:value

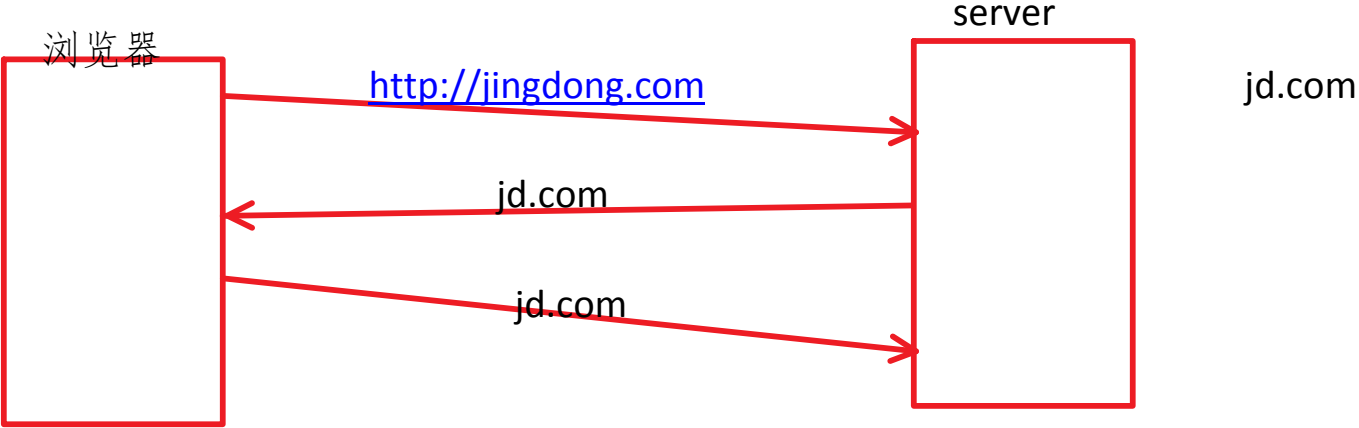
key:value

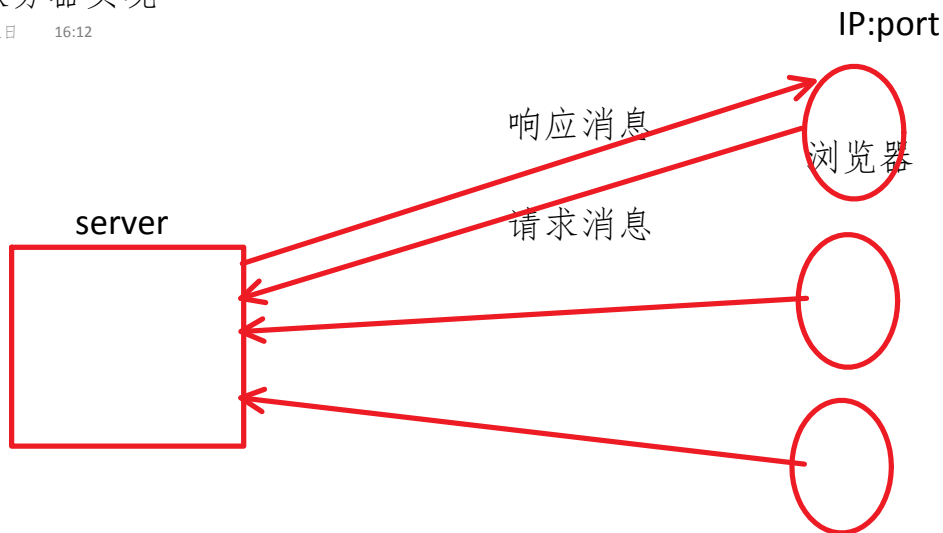
key:value

\r\n

重定向

2017年6月11日 15:44





http

- 传输协议: tcp

服务器端的实现:

```
void http_respond_head(int cfd, char* type)
{
    char buf[1024];
    // 状态行
    sprintf(buf, "http/1.1 200 OK\r\n");
    write(cfd, buf, strlen(buf));
    // 消息报头
    sprintf(buf, "Content-Type: %s\r\n", type);
    write(cfd, buf, strlen(buf));

    // 空行
    write(cfd, "\r\n", 2);
}

void main()
{
    // 修改进程的工作目录
    chdir(path);
    // 创建监听的套接字
    int lfd = socket(AF_INET, SOCK_STREAM, 0);
    // 绑定
    struct sockaddr_in serv;
    serv.family = AF_INET;
    serv.port = htons(8989);
    bind(lfd, &serv, len);
    // 监听
    listen();

    int cfd = accept();
```

```

// 读数据
read(cfd, buf, sizeof(buf));
// 先将buf中的请求行拿出来
// GET /hello.c http/1.1
char method[12], path[1024], protocol[12];
// 得到文件名
char* file = path+1;
// 打开文件
int fdd = open(file, O_RDONLY);
int len = 0;
http_respond_head(cfd, "text/plain");
// 循环读数据
while( (len=read(fdd, buf, sizeof(buf))) > 0)
{
    // 数据发送给浏览器
    write(fdd, buf, len);
}
}

```

== http 中的文件类型

普通文件: text/plain; charset=utf-8
*.html : text/html; charset=utf-8
*.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, 说明网站采用的编码是英文;
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上
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;

QFile

2Day

1. 编写函数解析http请求
 - GET /hello.html HTTP/1.1\r\n
 - 将上述字符串分为三部分解析出来
2. 编写函数根据文件后缀，返回对应的文件类型
3. sscanf - 读取格式化的字符串中的数据
 - 使用正则表达式拆分
 - [^]的用法
4. 通过浏览器请求目录数据
 - 读指定目录内容
 - opendir
 - readdir
 - closedir
 - scandir - 扫描dir目录下(不包括子目录)内容
5. http重定向
 - 消息报头：
 - Location: path
 - 状态码: 302
6. http中数据特殊字符编码解码问题
 - 编码
 - 解码

正则表达式

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

数据转码

url在数据传输过程中不支持中文，需要转码。

- 汉字
 - 特殊字符
 - 查看manpage
 - man ascii
 - 要处理可见字符
 - 从space开始 - 32
 - 前0-31个不可见
 - 不需要转换的特殊字符
 - .
 - _
 - *
 - /
 - ~
 - 0-9
 - a-z
 - A-Z
 - 需要转换的字符使用其16进制的值前加%表示
- 可以在shell下通过unicode工具查看
- 安装unicode
- sudo apt-get install unicode

