

User Guide

作者：云*飞扬

版本：v1.1

许可：LGPL

2013-12-12

目录

User Guide.....	1
一、概述.....	3
1.1 特点.....	3
1.2 目录结构.....	3
二、Html 网页开发.....	3
2.1 普通网页开发.....	3
2.2 CGI 网页开发.....	4
2.2.1 CGI 语法.....	4
2.2.2 CGI 文件例.....	4
2.3 将网页生成 C 文件.....	4
三 网页的处理.....	5
3.1 系统的裁剪与配置.....	5
3.2 CGI 后台处理.....	5
3.2.1 CGI 后台处理注意.....	5
3.2.2 CGI 后台处理代码.....	6
3.3 Post 处理.....	6
3.3.1 Post 处理注意.....	6
3.3.2 Post 处理代码.....	7
3.4 Upload 处理.....	7
3.4.1 upload 网页代码.....	7
3.4.2 后台处理注意.....	7
3.4.3 处理代码.....	8
四 测试.....	8
4.1 测试代码.....	8
4.2 测试结果.....	9
4.2.1 用户认证：.....	9
4.2.2 主页.....	9
4.2.3 CGI 测试.....	9
4.2.4 POST 测试.....	10

一、概述

1.1 特点

适用 RT-thread 系统和 lwip
使用 LGPL 开源协议
代码可裁剪
支持用户认证
支持 CGI
支持 POST
支持 Upload
网页支持 CSS, JavaScript 等。

1.2 目录结构

当前目录, 存放 HTTPD 核心代码
webfile 目录存放网页文件
interface 目录存放 3 个接口文件
documents 目录存放相关文档

二、Html 网页开发

2.1 普通网页开发

使用 Dreamware 等常用网页开发工具开发网页, 支持 CSS、支持 JavaScript.

2.2 CGI 网页开发

2.2.1 CGI 语法

注释 不处理

i 包含文件,

例“i test1.inc”, test1.inc 被 cgi 包含的文件,直接输出内容

c 命令

交由 cgi 方法处理,格式“c 命令 参数 html 代码”,例 c t 2 <label for="result">%d</label>

此处%d 是经由 cgi 处理(sprintf 函数)的数据, %d—整行,%s—字符串,%c—字符

t 不处理, 直接输出

. 文件结束

2.2.2 CGI 文件例

#cgi demo (#注释 不处理)

t <html xmlns="http://www.w3.org/1999/xhtml"> (t 直接输出当前行)

t <head>

t <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

t <title>cgi demo</title>

t </head>

t <body>

t <p align="center">

t cgi demo

t </p>

t <form name="form1" method="post" action="/cgidemo.cgi">

t <input type="text" name="txt1" id="txt1" >

t +

t <input type="text" name="txt2" id="txt2" >

t =

t <input type="submit" name="calc" id="calc" value="calc">

c t 2 <label for="result">%d</label>(c t 2 ...%d..., (t,2)命令处理%d 参数)

t </form>

t </body>

t </html>

. (. 文件结束符)

2.3 将网页生成 C 文件

将网页放于.\web 目录下, 在 cmd 内执行 makefsfile.exe -i web -o webdata.h

即将 web 目录内的文件全部转换输出到 webdata.h 文件中。

说明:

-i 输入目录

-o 输出文件

三 网页的处理

3.1 系统的裁剪与配置

在 httpd.h 中可以裁剪和配置系统

```
#define RT_LWIP_HTTP_PRIORITY    20
//线程优先级
#define RT_HTTP_USE_CGI
//使用 CGI
#define RT_HTTP_USE_POST
//使用 POST
#define RT_HTTP_USE_UPLOAD
//使用 Upload
#define RT_HTTP_USE_AUTH
//使用用户认证

#ifdef RT_HTTP_USE_AUTH
#define  USER_AUTH_TITLE "Mini Web Server"
//登陆时显示名字
#define  USER_AUTH_NAME   "admin"
//用户名
#define  USER_AUTH_PWD    "admin"
//密码
#endif
```

3.2 CGI 后台处理

3.2.1 CGI 后台处理注意

CGI 后台处理位于 cgi.c 中，命令参数需要与 CGI 文件命令参数对应，
例如：`c t 2 <label for="result">%d</label>`

那么我们处理的是 t 命令，2 参数据，处理数据类型为%d—整数
整处理都要借助 sprintf 方法，将不同类型变量变换到相应的位置。

3.2.2 CGI 后台处理代码

```
/*  
#cgi example  
i file.inc  
c d 1 <TD><INPUT TYPE=TEXT NAME=pw SIZE=10 MAXLENGTH=10 VALUE="%s"></TD></TR>  
.  
*/  
int add_cgi_1=0,add_cgi_2=0,add_cgi;  
void processCgi(int sock,char cmd,char param,char *data,int len){  
    char buff[256],tmpbuff[256];  
    rt_memset(buff,0,256);  
    rt_memset(tmpbuff,0,256);  
    rt_memcpy(tmpbuff,data,len);  
    switch(cmd){  
    case 't'://对应上面 t 命令  
    {  
        switch(param)  
        {  
            case '2': //对应上面 2 参数  
            add_cgi=add_cgi_1+add_cgi_2;  
            sprintf(buff,tmpbuff,add_cgi); // add_cgi 整数对应上面%d  
            break;  
        }  
    }  
    break;  
    }  
    send(sock,buff,strlen(buff),0);  
}
```

3.3 Post 处理

3.3.1 Post 处理注意

Post 处理位于 post.c 中， 处理控件时， 控件 id/name 要与之对应，
如： <input type="text" name="text" id="text">
那么就是处理"text"字符串

3.3.2 Post 处理代码

```
extern  int add_cgi_1,add_cgi_2;
void processPost(int sock,char *name,char * value){
    if(strncmp(name,"ck",2)==0){
        if(strncmp(value,"on",2)==0)
        {
            send(sock,"<p>check box on</p>",19,0);
        }
    }
    else if(strncmp(name,"text",4)==0){ //处理 text 控件
        send(sock,"<p>",3,0);
        send(sock,value,rt_strlen(value),0);
        send(sock,"</p>",4,0);
    }
    else if(strncmp(name,"txt1",4)==0){
        add_cgi_1=atoi(value);
    }
    else if(strncmp(name,"txt2",4)==0){
        add_cgi_2=atoi(value);
    }
}
```

3.4 Upload 处理

3.4.1 upload 网页代码

```
<form action=/ method=POST enctype="multipart/form-data" name=fileupload>
<p align="center">
<input name="File" type="file" size="50" maxlength="1000">
</p>
<p align="center">
<input type=button value="Send" onClick="this.form.submit()">
</p>
</form>
```

3.4.2 后台处理注意

处理代码位于 upload.c 中，

```
void processUpload(int sock,char *name,char * value,unsigned long offset,unsigned long length,unsigned long size)
```

sock—套接字， name--文件名， offset--偏移量,length--当前数据流长度， size--剩余数据流长度。

offset=0:文件开始

size=0:文件结束

3.4.3 处理代码

```
void processUpload(int sock,char *name,char * value,unsigned long offset,unsigned long length,unsigned long size)
```

```
{
    if(offset==0)
    {
        send(sock,"<p>Filename:",12,0);
        send(sock,name,rt_strlen(name),0);
        send(sock,"</p> <p>Data</p> <p>",20,0);
    }
    send(sock,value,length,0);
    if(size==0)
    {
        send(sock,"</p>",4,0);
    }
}
```

3.5 Web 启动

启动函数 void init_httpd(void);它会开起一个线程。

四 测试

4.1 测试代码

测试代码在 web_lm3s8962.zip 中，将该文件减压到 rt-thread\bsp 中编译

4.2 测试结果

4.2.1 用户认证:

默认用户名:admin

默认密码: admin



4.2.2 主页



4.2.3 CGI 测试

服务器将两个文本框内数字加起来显示在后面



4.2.4 POST 测试

将提交的按键的值打印出来



4.2.5 上传测试:

将上传的文件名，与文件内容打印出来



Web Upload



[mini web server v1.1 demo](#)

[cgi demo](#) [post demo](#) [upload demo](#)

Filename:upload.txt

Data

test upload ok or not