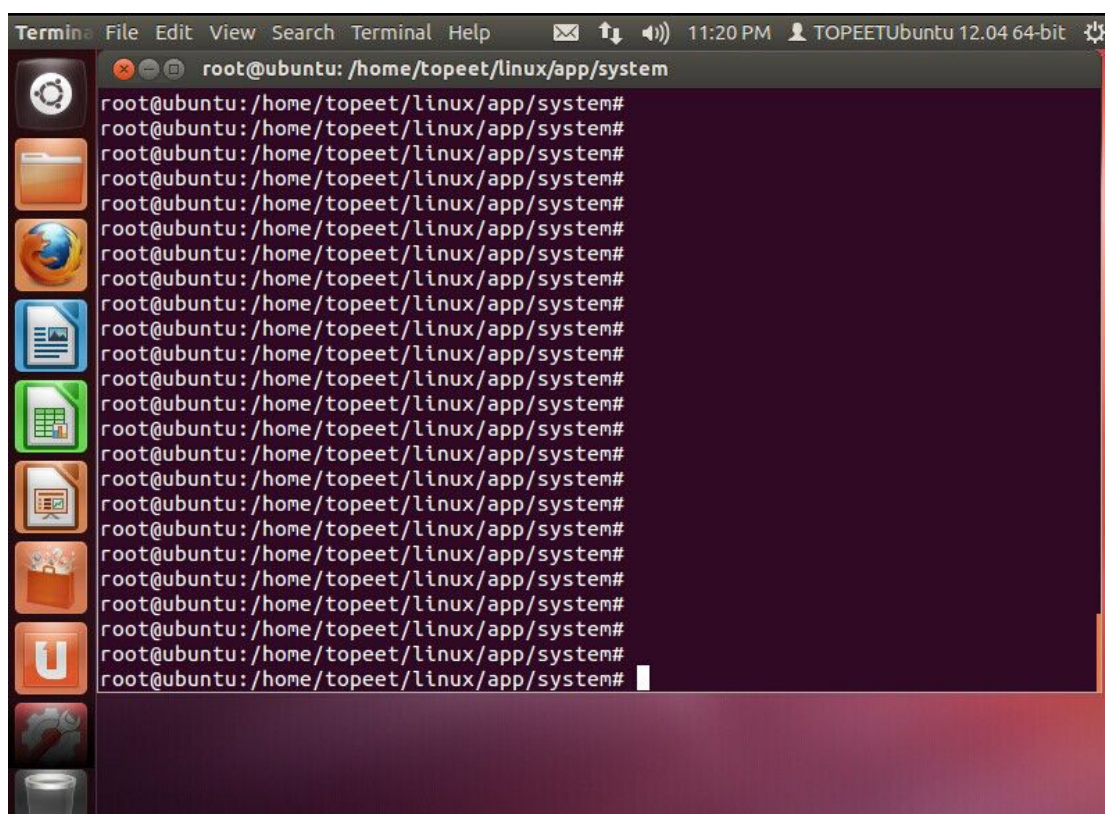


大家好，上一章我们讲了在 iTOP-4412 开发板上搭建 web 服务器，这一章我们在前面一章的基础上讲解一下通过 web 网页实现控制 LED。

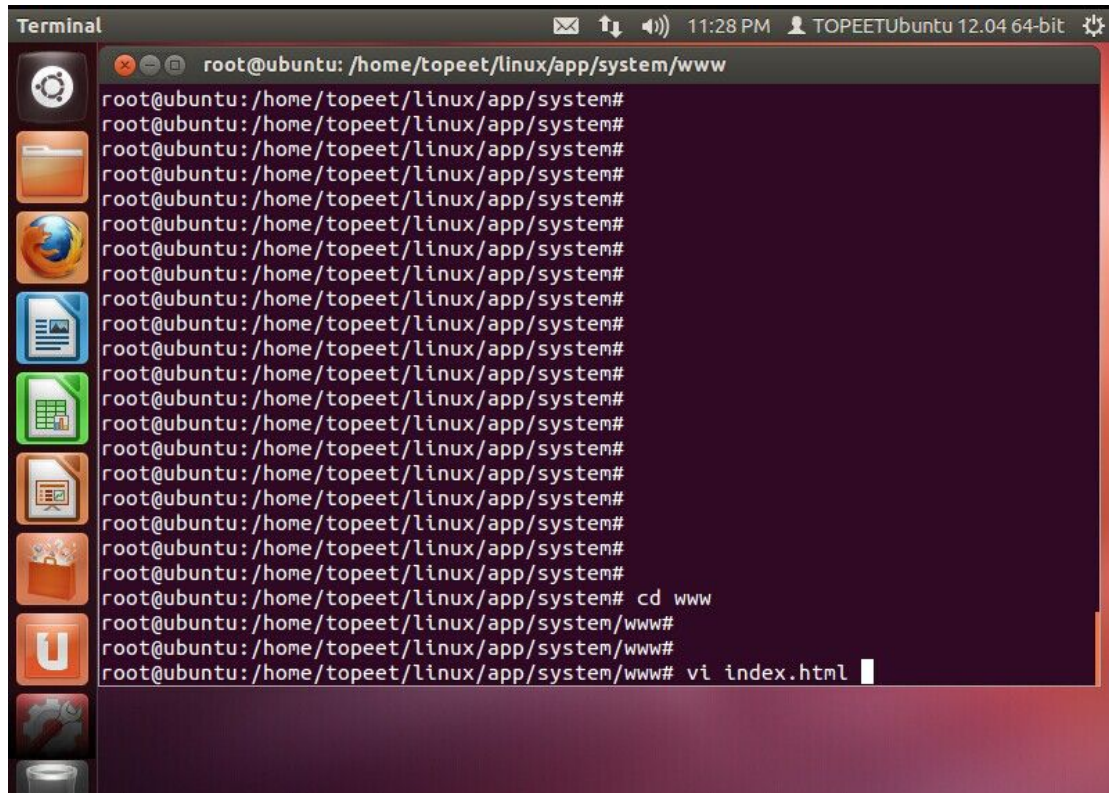
要实现控制 led，我们需要 CGI 编程，CGI (Common Gateway Interface) 是外部应用扩展，应用程序与 www 服务器交互的一个标准接口。按照 CGI 标准编写的外部扩展应用程序可以处理客户端浏览器输入的数据，从而完成客户端与服务器的交互操作。而 CGI 规范就定义了 web 服务器如何向扩展应用程序发送消息，在收到扩展应用程序的信息后又如何进行处理等内容。通过 CGI 可以提供许多静态的 HTML 网页无法实现的功能。比如搜索引擎、基于 web 的数据库访问等等。

首先进入到我们的 NFS 文件系统所在的目录，如下图：



在前面的搭建 web 服务器章节 我们在修改 `boa.conf` 配置文件的时候指定了 web 网页的存放目录 “`DocumentRoot /www`”，现在我们打开之前我们在 `www` 目录

创建的 index.html，如下图所示：



打开 index.html 后，删除掉里面的内容，然后输入下面的内容：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

<head>

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

```
<title>led 远程控制</title>
```

```
<style type="text/css">
```

body {

```
background-color: #999900;
```

```
text-align: center;
}

.ziti {
font-size: 24px;
}

.juzhong {
text-align: center;
}

.hsz {
text-align: center;
}

.hsz td {
color: #00F;
font-size: 18px;
}

.hsz {
background-color: #FCC;
}

.juzhong table {
text-align: center;
}
```

```
.juzhong table tr {  
}  
  
#h1 {  
    background-color: #0FC;  
}  
  
#h2 {  
    background-color: #FF9;  
}  
  
.h3 {  
    background-color: #0CF;  
}  
  
.ys1 {  
    font-size: 24px;  
}  
  
.STYLE1 {font-size: 36px}  
</style>  
</head>  
  
<body class="juzhong">  
<table width="900" border="0" align="center" cellpadding="0"  
cellspacing="0">
```

```
<tr>

<td> <p class="STYLE1">&nbsp;</p>

<p class="STYLE1">iTOP-4412 WEB SERVER </p> </td>

</tr>

<tr>

<td height="30">&nbsp;</td>

</tr>

<tr>

<td> <form action="/cgi-bin/myled.cgi" method="get"
enctype="application/x-www-form-urlencoded" name="form1"
target="_blank" id="form1">

<table width="300" border="1" align="center" cellpadding="1"
cellspacing="1">

<tr>

<td>Led1</td>

<td> <input name="led1" type="checkbox" id="led1" value="1"
/>

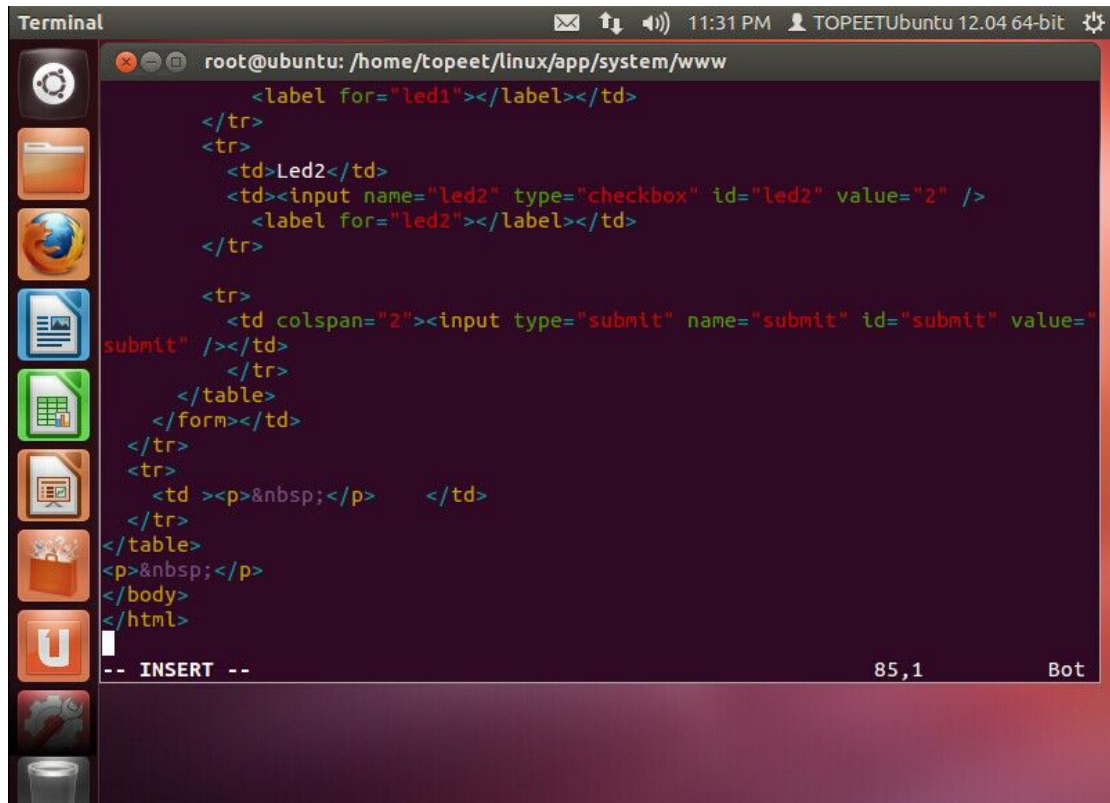
<label for="led1"> </label> </td>

</tr>

<tr>
```

```
<td>Led2</td>
<td> <input name="led2" type="checkbox" id="led2" value="2"
/>
<label for="led2"> </label> </td>
</tr>
<tr>
<td colspan="2"> <input type="submit" name="submit"
id="submit" value="submit" /> </td>
</tr>
</table>
</form> </td>
</tr>
<tr>
<td > <p>&nbsp;</p> </td>
</tr>
</table>
<p>&nbsp;</p>
</body>
</html>
```

如下图所示：

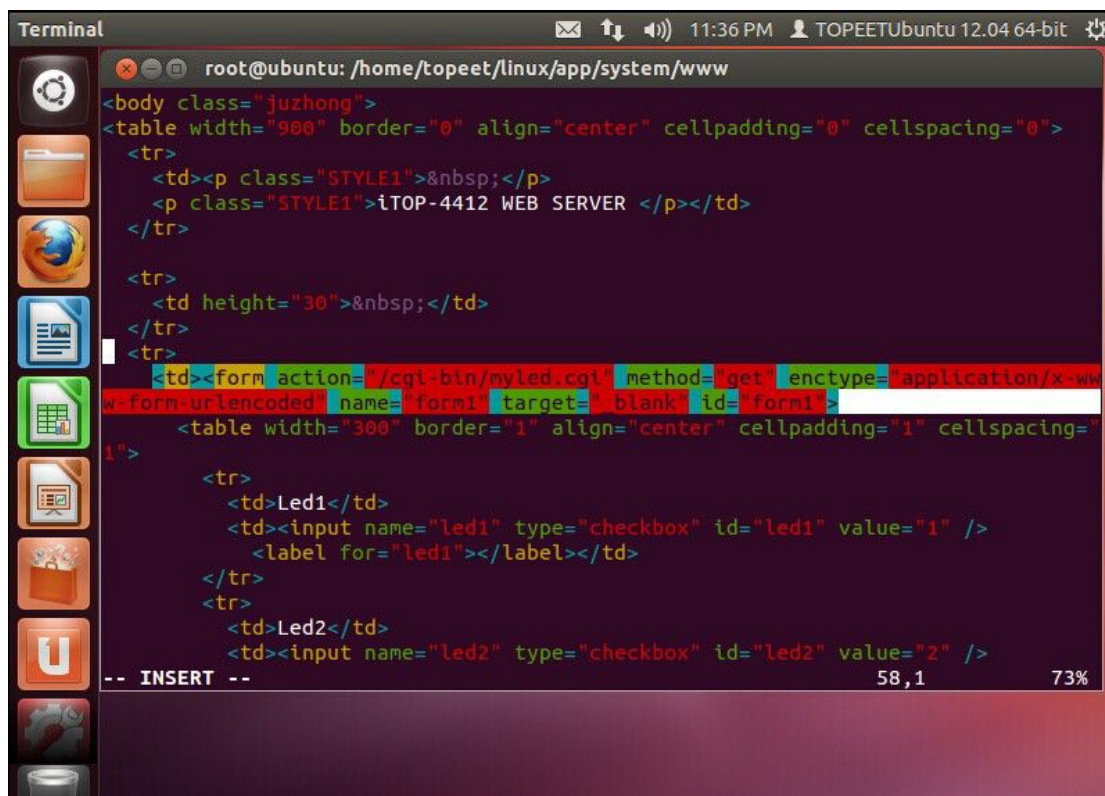
A terminal window titled "Terminal" with a dark background and light-colored text. The prompt is "root@ubuntu: /home/topeet/linux/app/system/www". The code being entered is HTML for a web form. It includes a table with two rows. The first row has a label for "led1". The second row has a label for "led2" and a checkbox with name "led2" and value "2". Below the table is a submit button with name "submit" and value "submit". The code is as follows:

```
<label for="led1"></label></td>
</tr>
<tr>
  <td>Led2</td>
  <td><input name="led2" type="checkbox" id="led2" value="2" />
    <label for="led2"></label></td>
</tr>

<tr>
  <td colspan="2"><input type="submit" name="submit" id="submit" value="
submit" /></td>
</tr>
</table>
</form></td>
</tr>
<tr>
  <td><p>&nbsp;</p>    </td>
</tr>
</table>
<p>&nbsp;</p>
</body>
</html>
```

The terminal also shows "-- INSERT --" at the bottom left, "85,1" at the bottom center, and "Bot" at the bottom right.

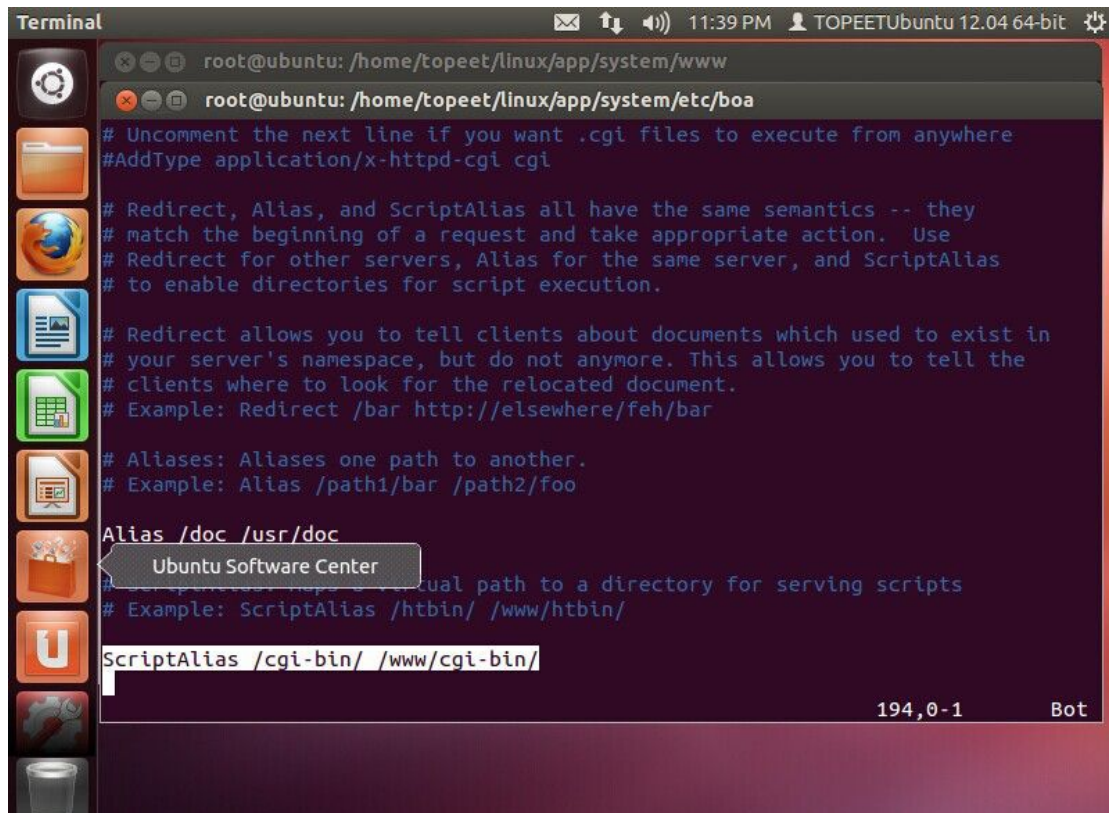
然后保存并退出。上面输入的是 HTML 格式的代码，主要是用到了通过表单向服务器提交信息，在表单里面指定了服务器端处理接收到信息的 CGI 程序是 myled，这是在 form 表单的属性里设置的，代码是 “form action="/cgi-bin/myled.cgi" method="get”，使用的传递数据的方式是 get 方法，如下图所示：

A terminal window titled 'Terminal' with a top bar showing 'root@ubuntu: /home/topeet/linux/app/system/www' and system icons. The terminal displays HTML code for an index.html file. The code includes a table with a center-aligned paragraph 'iTOP-4412 WEB SERVER', a form with a text input field, and two checkboxes labeled 'Led1' and 'Led2'. The form's action is '/cgi-bin/myled.cgi'. The terminal also shows a status bar at the bottom with '-- INSERT --', '58,1', and '73%'.

```
Terminal
root@ubuntu: /home/topeet/linux/app/system/www

<body class="juzhong">
<table width="900" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td><p class="STYLE1">&nbsp;</p>
<p class="STYLE1">iTOP-4412 WEB SERVER </p></td>
</tr>
<tr>
<td height="30">&nbsp;</td>
</tr>
<tr>
<td><form action="/cgi-bin/myled.cgi" method="get" enctype="application/x-www-form-urlencoded" name="form1" target="blank" id="form1">
<table width="300" border="1" align="center" cellpadding="1" cellspacing="1">
1">
<tr>
<td>Led1</td>
<td><input name="led1" type="checkbox" id="led1" value="1" />
<label for="led1"></label></td>
</tr>
<tr>
<td>Led2</td>
<td><input name="led2" type="checkbox" id="led2" value="2" />
-- INSERT --
58,1 73%
```

修改完了 index.html，我们需要些 CGI 程序。在 boa 的配置文件 boa.conf 里面我们指定了 CGI 程序的存储目录是 "ScriptAlias /cgi-bin/ /www/cgi-bin/"，如下图所示：



```
Terminal
root@ubuntu: /home/topeet/linux/app/system/www
root@ubuntu: /home/topeet/linux/app/system/etc/boa
# Uncomment the next line if you want .cgi files to execute from anywhere
#AddType application/x-httpd-cgi cgi

# Redirect, Alias, and ScriptAlias all have the same semantics -- they
# match the beginning of a request and take appropriate action. Use
# Redirect for other servers, Alias for the same server, and ScriptAlias
# to enable directories for script execution.

# Redirect allows you to tell clients about documents which used to exist in
# your server's namespace, but do not anymore. This allows you to tell the
# clients where to look for the relocated document.
# Example: Redirect /bar http://elsewhere/feh/bar

# Aliases: Aliases one path to another.
# Example: Alias /path1/bar /path2/foo

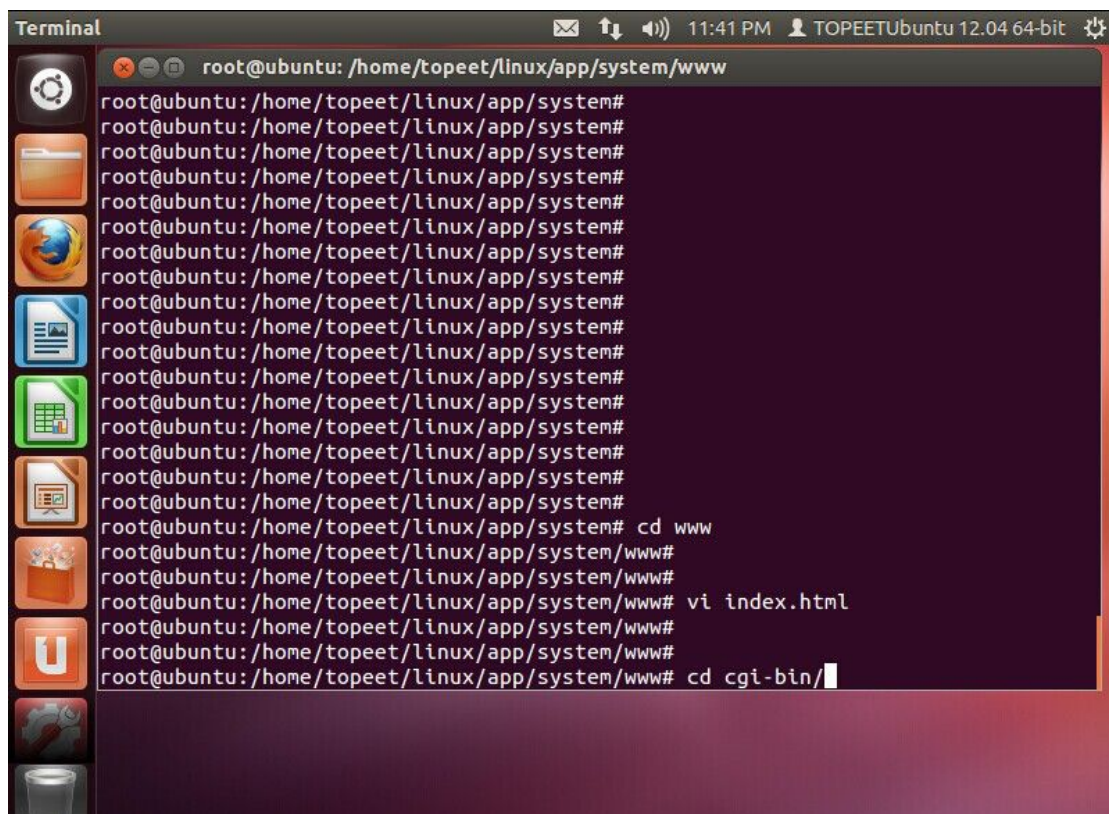
Alias /doc /usr/doc

# ScriptAlias maps a virtual path to a directory for serving scripts
# Example: ScriptAlias /htbin/ /www/htbin/

ScriptAlias /cgi-bin/ /www/cgi-bin/

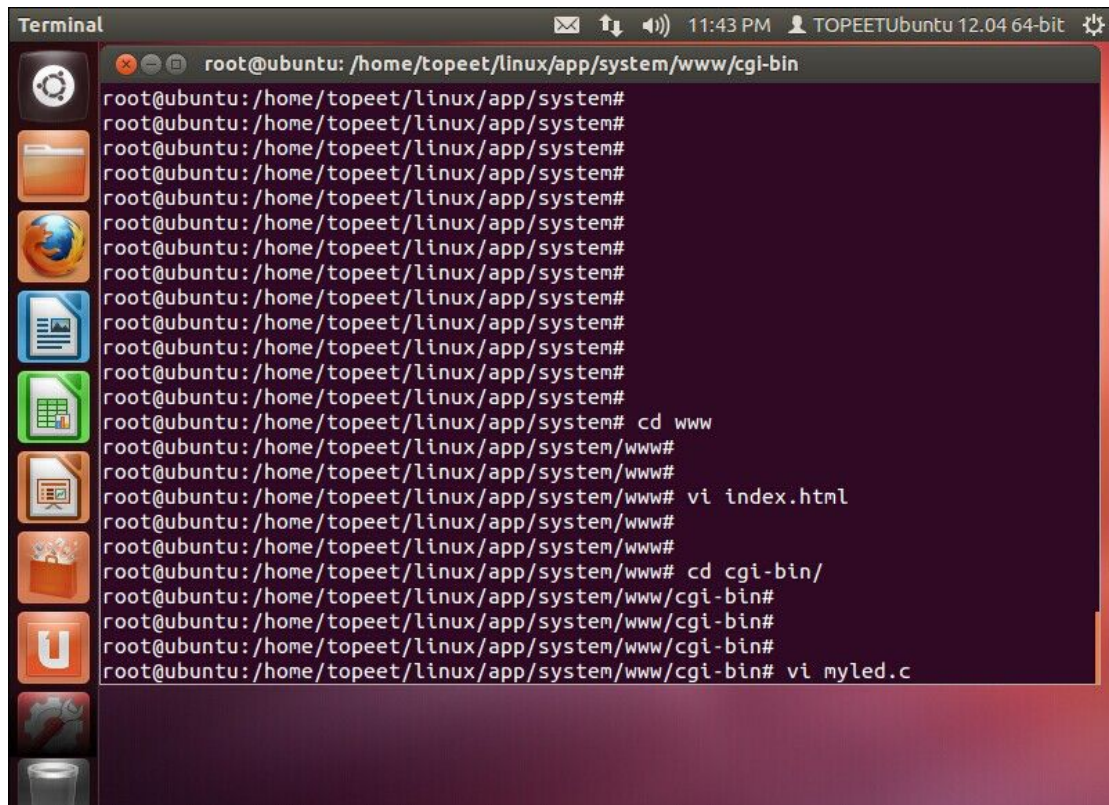
194,0-1 Bot
```

现在我们进入到 NFS 文件系统的 www/cgi-bin 目录，如下图所示：



```
Terminal
root@ubuntu: /home/topeet/linux/app/system/www
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system# cd www
root@ubuntu: /home/topeet/linux/app/system/www#
root@ubuntu: /home/topeet/linux/app/system/www# vi index.html
root@ubuntu: /home/topeet/linux/app/system/www#
root@ubuntu: /home/topeet/linux/app/system/www#
root@ubuntu: /home/topeet/linux/app/system/www# cd cgi-bin/
```

然后再 cgi-bin 目录我们创建 myled.c , 如下图所示 :

A terminal window titled 'Terminal' with a dark background and light text. The window shows a series of commands and their outputs. The user starts at the root directory of the system, navigates to the 'www' directory, then to the 'cgi-bin' directory, and finally creates a new file named 'myled.c' using the 'vi' editor. The terminal output shows the prompt changing as the user navigates through the directory structure.

```
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system# cd www
root@ubuntu: /home/topeet/linux/app/system/www#
root@ubuntu: /home/topeet/linux/app/system/www#
root@ubuntu: /home/topeet/linux/app/system/www# vi index.html
root@ubuntu: /home/topeet/linux/app/system/www#
root@ubuntu: /home/topeet/linux/app/system/www# cd cgi-bin/
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin#
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin#
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin#
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin# vi myled.c
```

然后在 myled.c 里输入下面的代码 :

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

```
#include <stdlib.h>
```

```
int main()
```

```
{
```

```
    char *data;
```

```
    int leds[2] = {0, 0};
```

```
long m, n;
```

```
int exit=0,i,fd;
```

```
printf("Content-Type:text/html;charset=gb2312\n\n");
```

```
printf("<html>\n");
```

```
printf("<body>\n");
```

```
printf("<title>iTOP-4412</title> ");
```

```
printf("<h3>iTOP-4412</h3> ");
```

```
data = getenv("QUERY_STRING");
```

```
printf("<p>receive data:%s</p>",data);
```

```
while(*data != '\0')
```

```
{
```

```
    if(*data=='=')
```

```
        switch(*(data+1))
```

```
        {
```

```
            case '1':leds[0]=1;break;
```

```
            case '2':leds[1]=1;break;
```

```
default:exit=1;break;
```

```
}
```

```
if(exit == 1)
```

```
break;
```

```
data++;
```

```
}
```

```
fd=open("/dev/leds",0);
```

```
for(i=0;i<2;i++)
```

```
{
```

```
if(leds[i]==1)
```

```
printf("<p>%d\\t</p>",i+1);
```

```
ioctl(fd,leds[i],i);
```

```
}
```

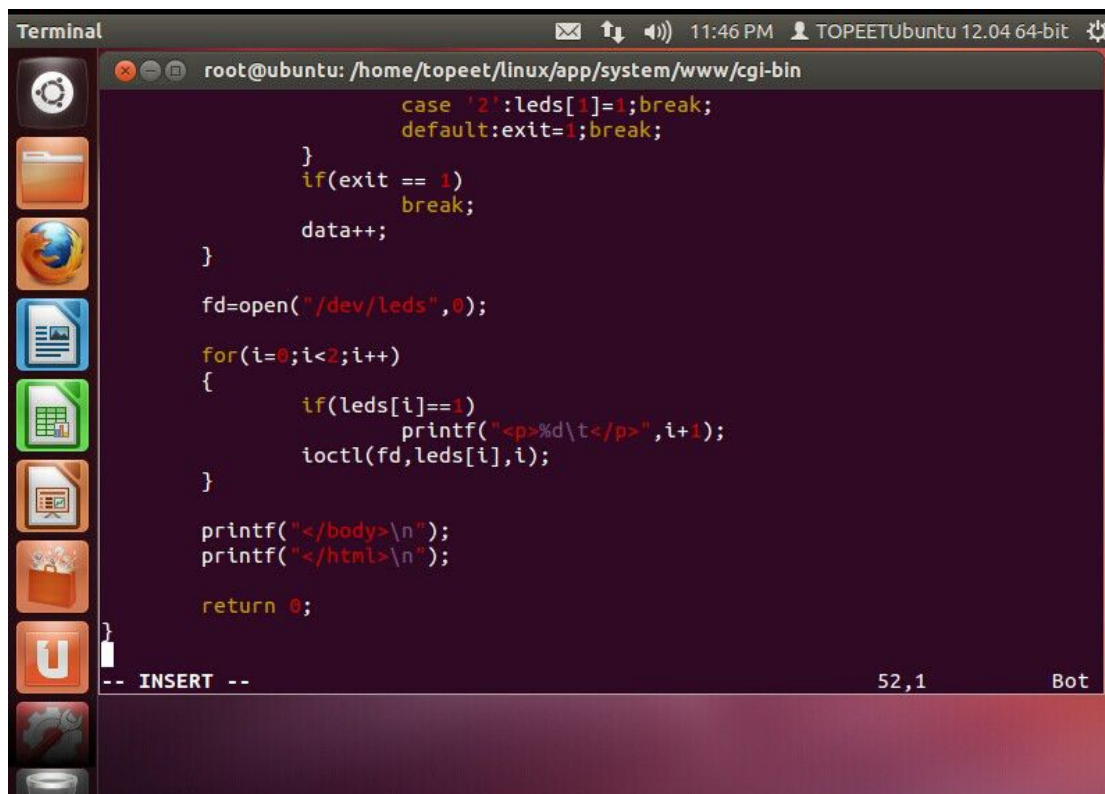
```
printf("</body>\\n");
```

```
printf("</html>\\n");
```

```
return 0;
```

```
}
```


如下图所示：



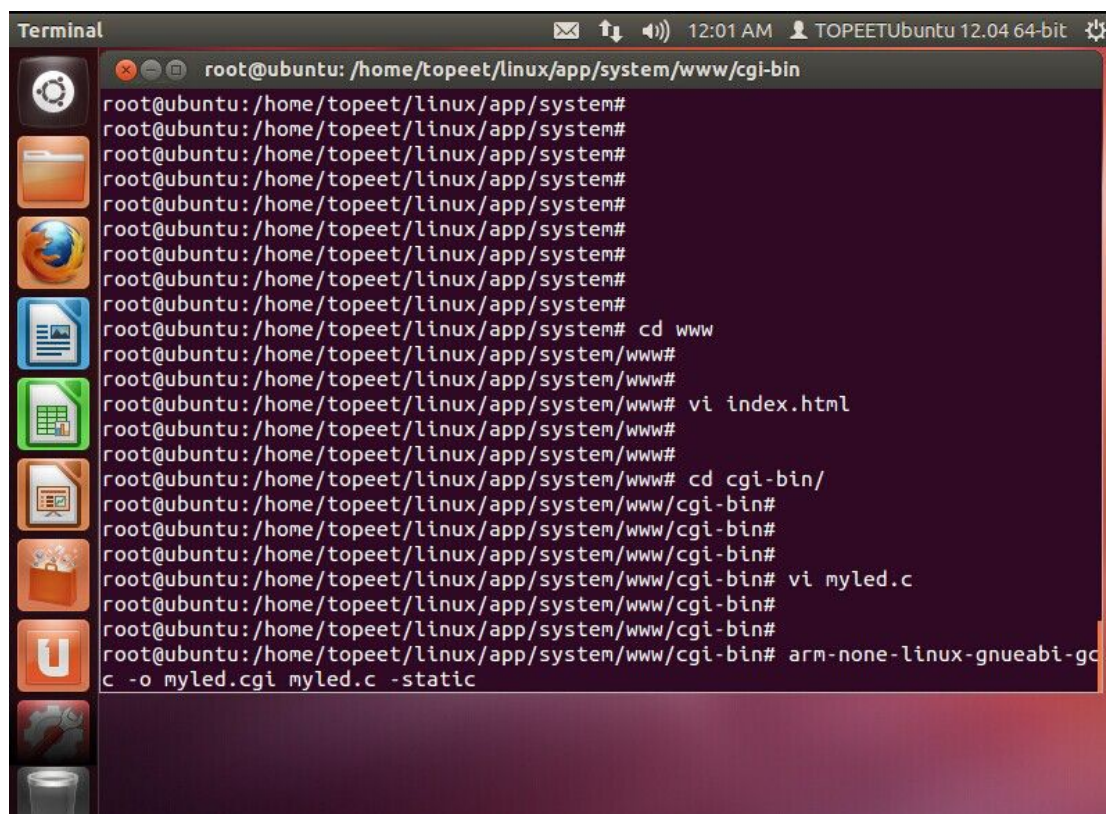
```
Terminal
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin

        case '2': leds[1]=1; break;
        default: exit=1; break;
    }
    if(exit == 1)
        break;
    data++;
}
fd=open("/dev/leds", 0);
for(i=0; i<2; i++)
{
    if(leds[i]==1)
        printf("<p>%d</p>", i+1);
    ioctl(fd, leds[i], i);
}
printf("</body>\n");
printf("</html>\n");
return 0;
}
-- INSERT --
52,1
Bot
```

最后保存并退出。

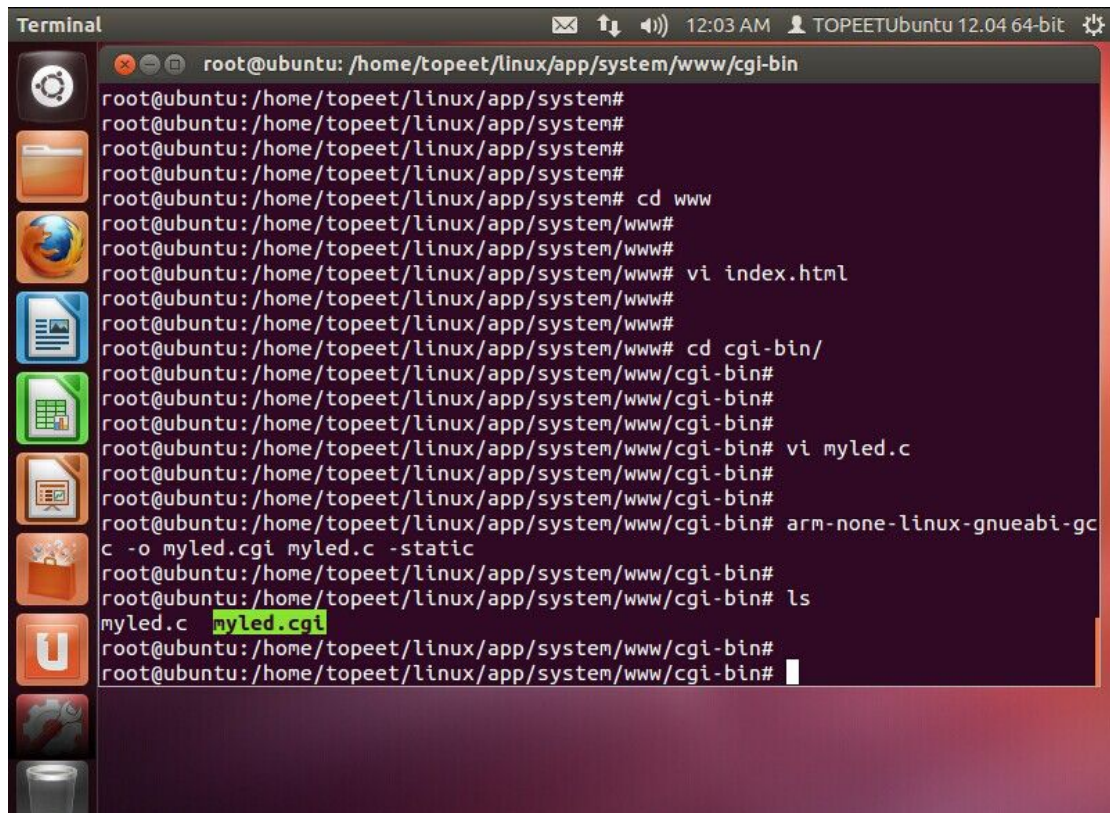
上面的代码是 C 语言，主要是完成了获取 web 提交过来的数据，然后对获得的数据解析，最后会调用 led 的 ioctl 函数来点亮或关闭 led。里面的 printf 打印语句，最终会生成一个 web 页面。接下来编译 myled.c 生成 myled.cgi (在 index.html 的表单里面设置了处理文请求的 cgi 程序是 myled.cgi)，使用命令

“arm-none-linux-gnueabi-gcc -o myled.cgi myled.c -static” 编译 myled.c，如下图所示：

A terminal window titled 'Terminal' with a dark background and light text. The window shows a series of commands and their outputs in a root shell at the path /home/topeet/linux/app/system/www/cgi-bin. The commands include navigating to the www directory, editing index.html, navigating to the cgi-bin directory, editing myled.c, and finally compiling myled.c into myled.cgi using the arm-none-linux-gnueabi-gcc compiler with the -static flag. The terminal also shows a sidebar with various application icons on the left and system status information at the top right.

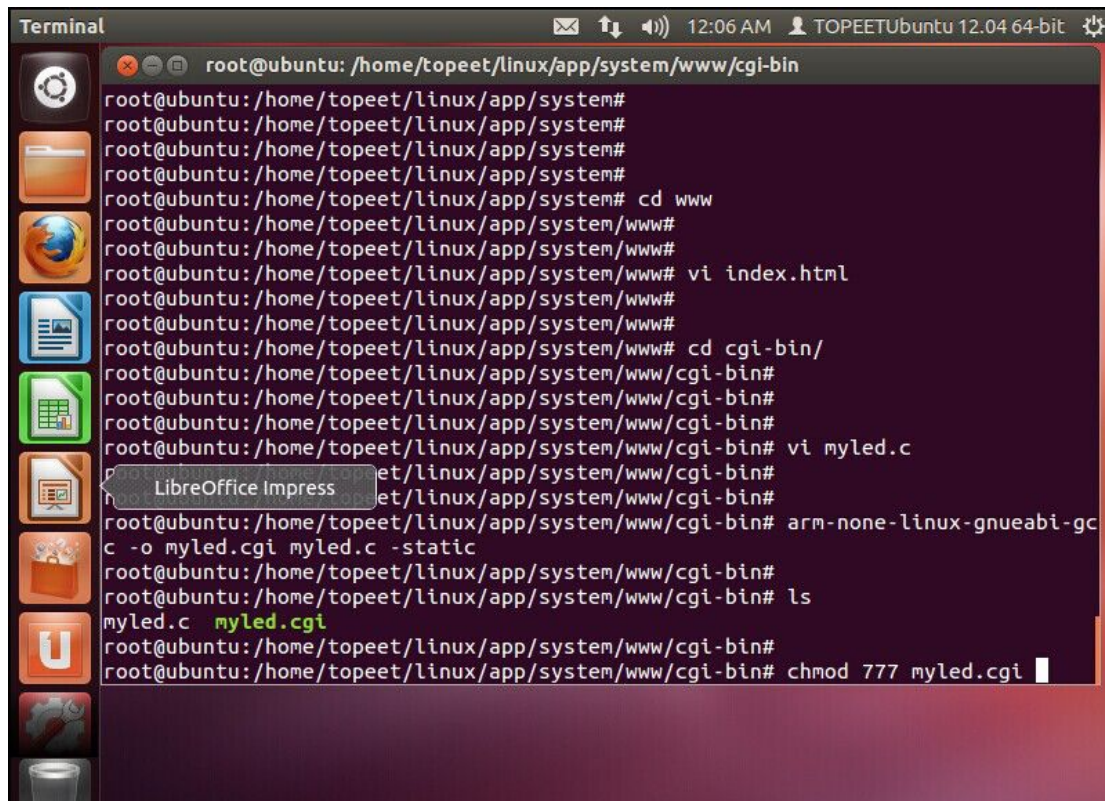
```
Terminal 12:01 AM TOPEETUbuntu 12.04 64-bit
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system# cd www
root@ubuntu: /home/topeet/linux/app/system/www#
root@ubuntu: /home/topeet/linux/app/system/www# vi index.html
root@ubuntu: /home/topeet/linux/app/system/www#
root@ubuntu: /home/topeet/linux/app/system/www#
root@ubuntu: /home/topeet/linux/app/system/www# cd cgi-bin/
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin#
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin#
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin# vi myled.c
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin#
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin#
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin# arm-none-linux-gnueabi-gcc
c -o myled.cgi myled.c -static
```

编译完成后，我们可以看到在 www/cgi-bin 目录下生成了 myled.cgi，如下图所示：

A terminal window titled 'Terminal' with a top bar showing system status (12:03 AM, TOPEETUbuntu 12.04 64-bit). The terminal shows a user at the root@ubuntu prompt navigating through the directory /home/topeet/linux/app/system/www/cgi-bin. The user creates a file index.html, enters the cgi-bin directory, creates myled.c, and compiles it into myled.cgi using the command 'arm-none-linux-gnueabi-gcc -o myled.cgi myled.c -static'. The 'myled.cgi' file is highlighted in green in the terminal output.

```
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system# cd www
root@ubuntu: /home/topeet/linux/app/system/www#
root@ubuntu: /home/topeet/linux/app/system/www# vi index.html
root@ubuntu: /home/topeet/linux/app/system/www#
root@ubuntu: /home/topeet/linux/app/system/www# cd cgi-bin/
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin#
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin#
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin# vi myled.c
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin#
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin#
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin# arm-none-linux-gnueabi-gcc -o myled.cgi myled.c -static
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin# ls
myled.c  myled.cgi
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin#
```

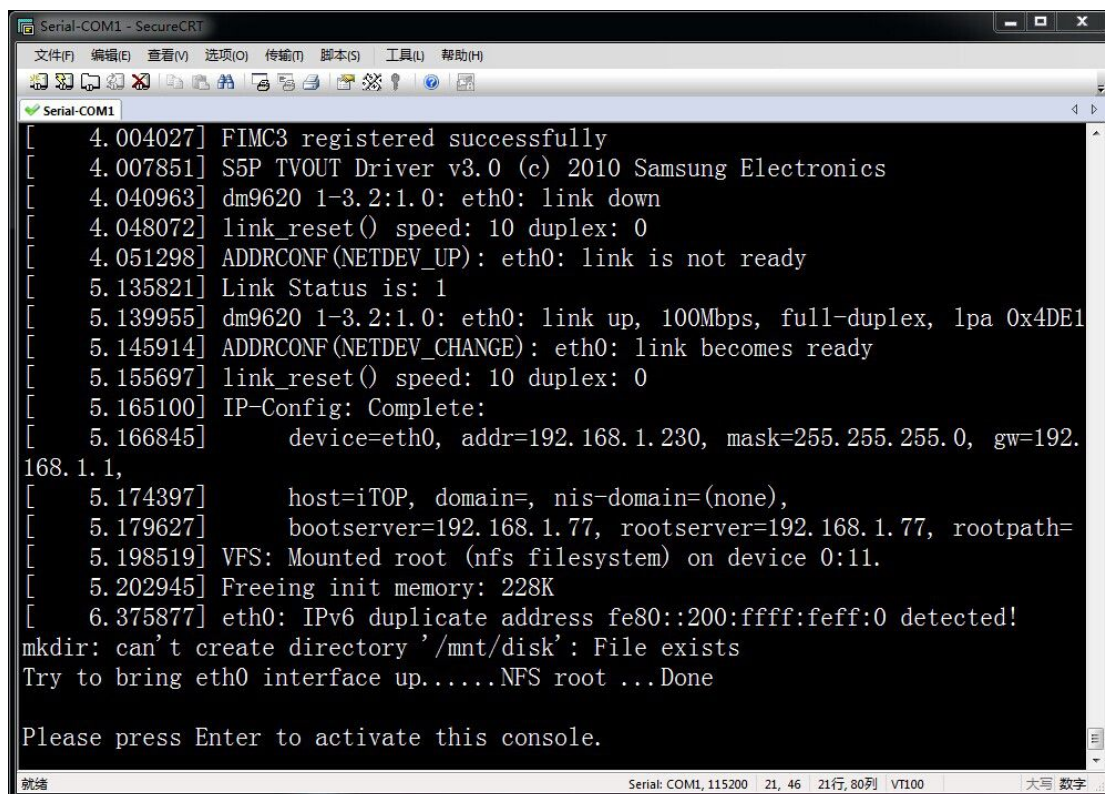
注意：生成的 myled.cgi 需要有可执行权限，所以我们使用 chmod 命令修改下它的权限，如下图所示：



A terminal window titled "Terminal" showing a series of commands and their outputs. The user is root@ubuntu and is navigating through the directory structure /home/topeet/linux/app/system/www/cgi-bin. The commands include navigating to the directory, creating an index.html file, creating a myled.c file, compiling it with arm-none-linux-gnueabi-gcc, and setting permissions with chmod 777 myled.cgi.

```
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system#
root@ubuntu: /home/topeet/linux/app/system# cd www
root@ubuntu: /home/topeet/linux/app/system/www#
root@ubuntu: /home/topeet/linux/app/system/www# vi index.html
root@ubuntu: /home/topeet/linux/app/system/www#
root@ubuntu: /home/topeet/linux/app/system/www# cd cgi-bin/
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin#
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin#
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin# vi myled.c
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin#
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin# arm-none-linux-gnueabi-gcc -o myled.cgi myled.c -static
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin#
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin# ls
myled.c  myled.cgi
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin#
root@ubuntu: /home/topeet/linux/app/system/www/cgi-bin# chmod 777 myled.cgi
```

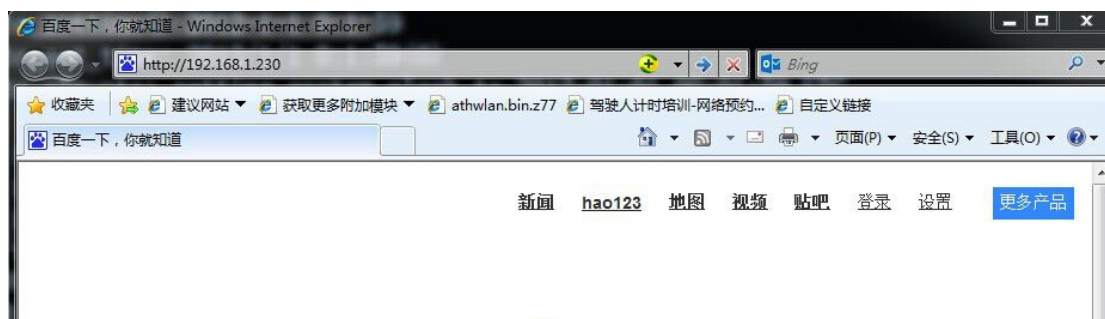
现在控制 led 的 web 程序就已经完成了，现在启动开发板，如下图所示：



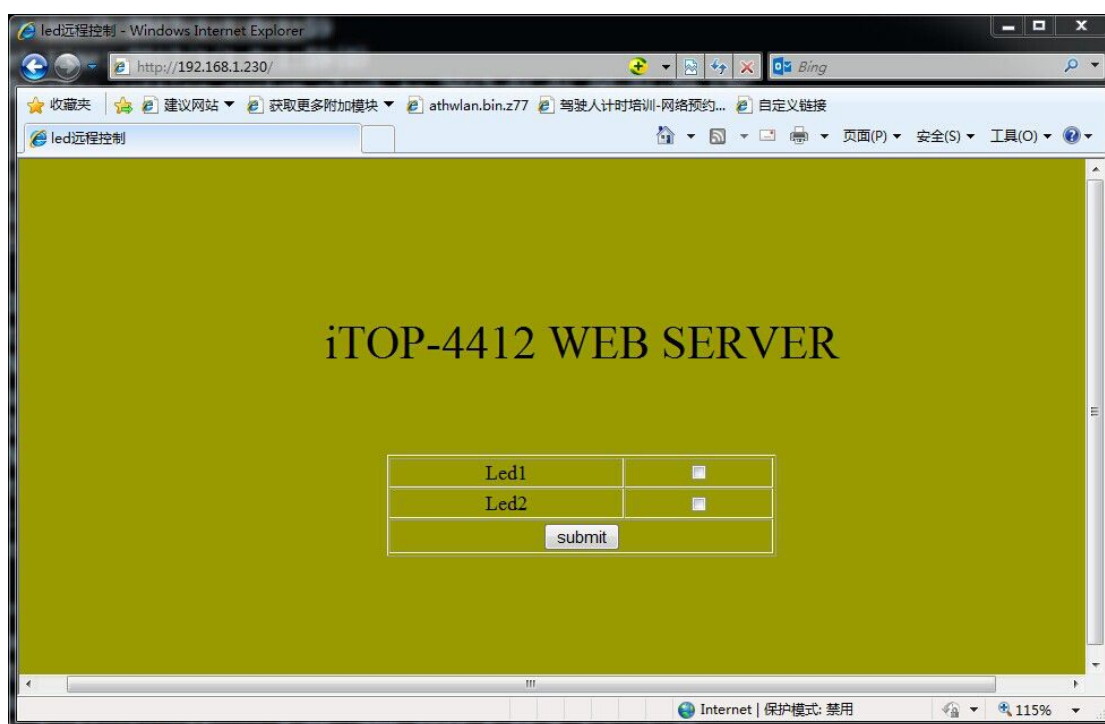
A Serial-COM1 window showing system boot logs. The logs include messages about FIMC3 registration, S5P TVOUT Driver, network link status changes, IP configuration, and NFS mounting. The window title is "Serial-COM1 - SecureCRT".

```
[ 4.004027] FIMC3 registered successfully
[ 4.007851] S5P TVOUT Driver v3.0 (c) 2010 Samsung Electronics
[ 4.040963] dm9620 1-3.2:1.0: eth0: link down
[ 4.048072] link_reset() speed: 10 duplex: 0
[ 4.051298] ADDRCONF(NETDEV_UP): eth0: link is not ready
[ 5.135821] Link Status is: 1
[ 5.139955] dm9620 1-3.2:1.0: eth0: link up, 100Mbps, full-duplex, lpa 0x4DE1
[ 5.145914] ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
[ 5.155697] link_reset() speed: 10 duplex: 0
[ 5.165100] IP-Config: Complete:
[ 5.166845] device=eth0, addr=192.168.1.230, mask=255.255.255.0, gw=192.168.1.1,
[ 5.174397] host=iTOP, domain=, nis-domain=(none),
[ 5.179627] bootserver=192.168.1.77, rootserver=192.168.1.77, rootpath=
[ 5.198519] VFS: Mounted root (nfs filesystem) on device 0:11.
[ 5.202945] Freeing init memory: 228K
[ 6.375877] eth0: IPv6 duplicate address fe80::200:ffff:feff:0 detected!
mkdir: can't create directory '/mnt/disk': File exists
Try to bring eth0 interface up.....NFS root ...Done
Please press Enter to activate this console.
```

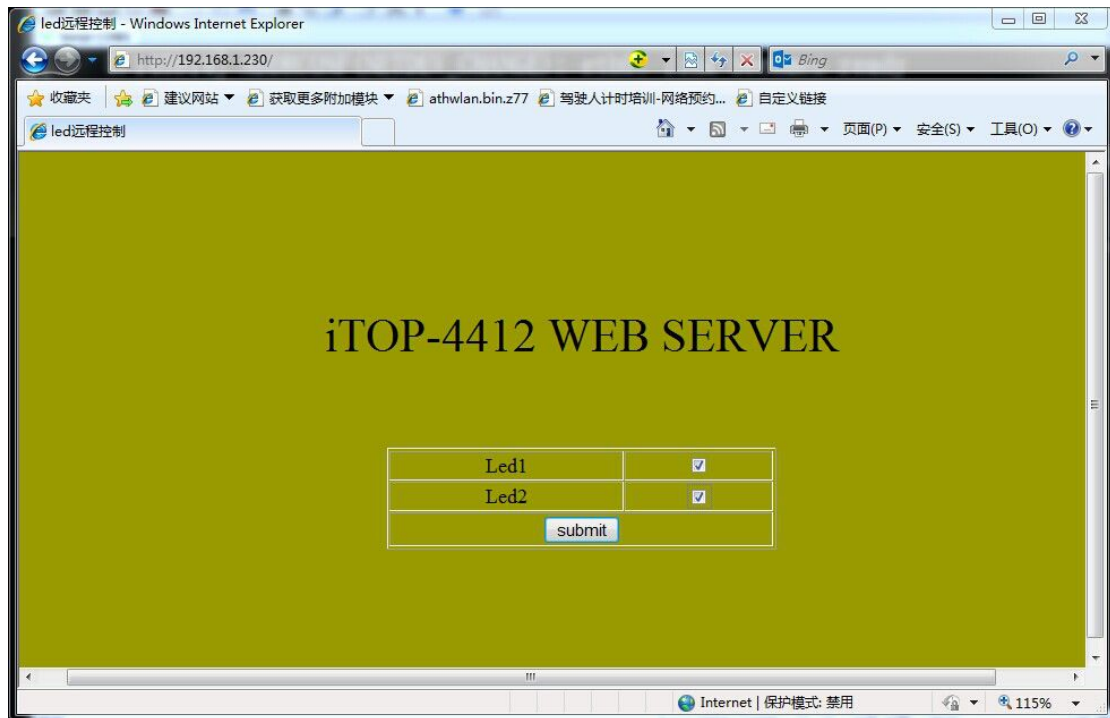
通过上图我们可以看到 iTOP-4412 开发板已经起来了，然后打开 PC 的浏览器，输入开发板的 ip 地址，如下图所示：



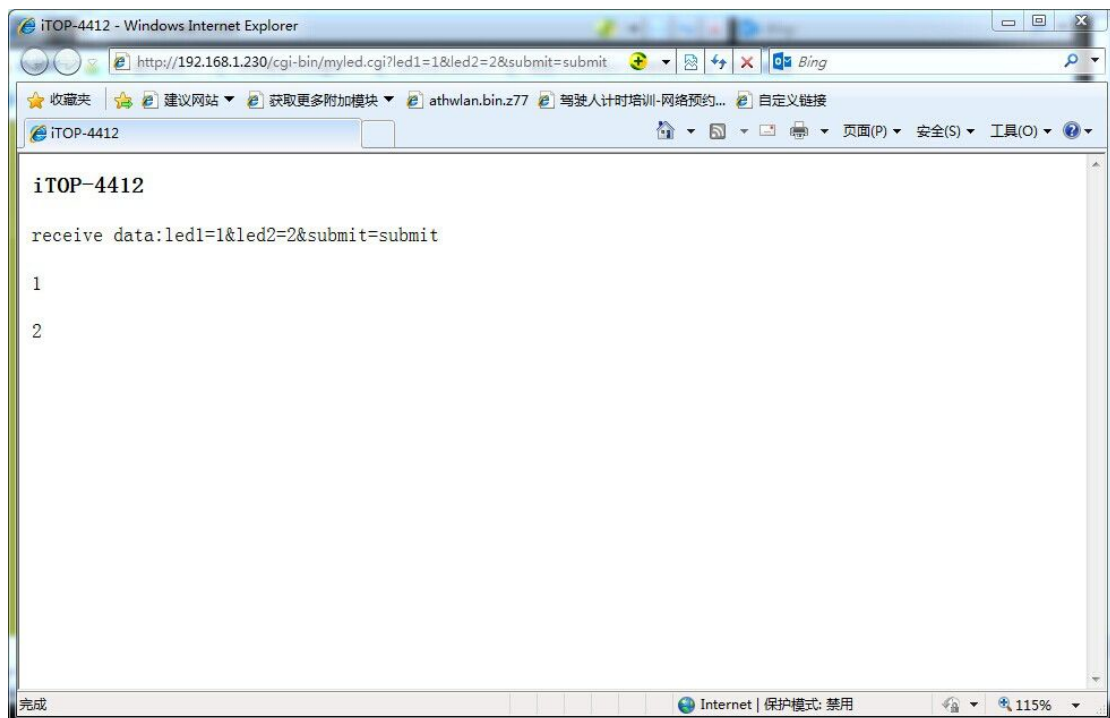
输入完成后，按回车，即可打开开发板的 index.html 页面，如下图所示：



我们可以选中 led1 和 led2 右边的复选框，如下图所示：

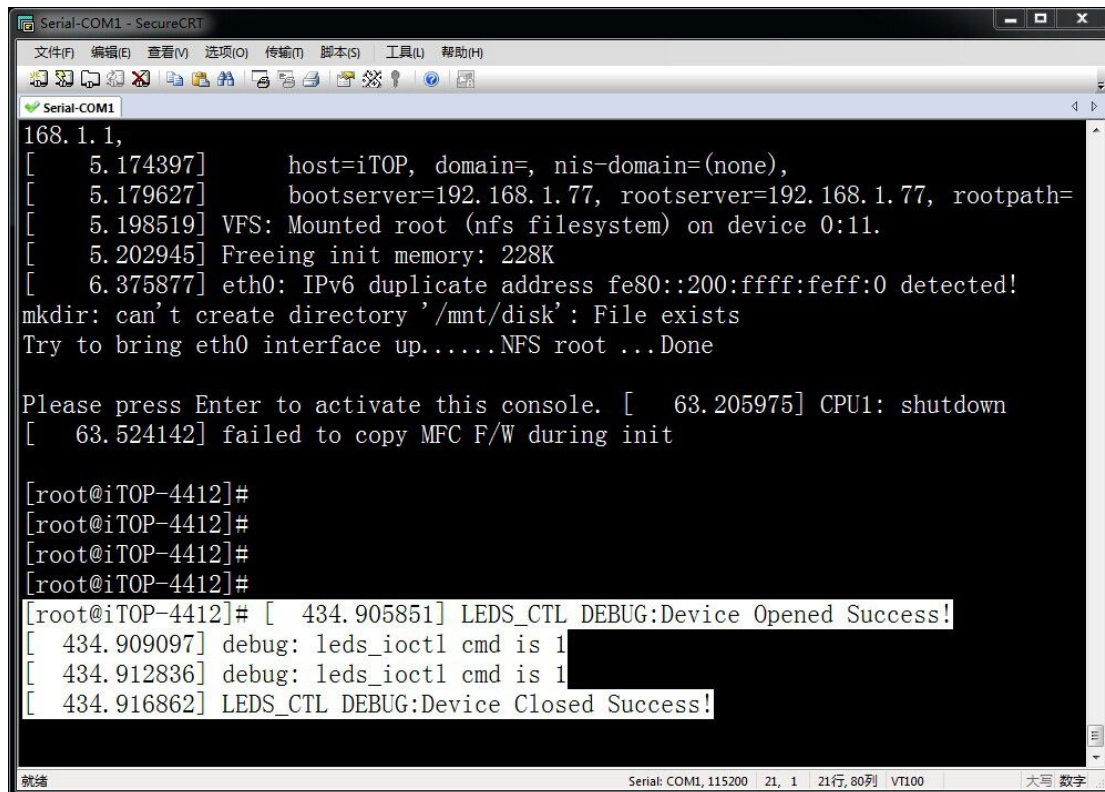


然后点击页面上的“submit”按钮，会打开一个新的 web 页面，如下图：



上图中的页面里面的信息就是我们的 myled.cgi 程序里面的 printf 打印出来的，其中“receive data:led1=1&led2=2&submit=submit”，就是客户端通过 get 方式传递给

web 服务器的数据，下面两行的 1 和 2 是 myled.cgi 程序从 “receive data:led1=1&led2=2&submit=submit” 里面解析出来的值。同时我们还可以在串口上看到 iTOP-4412 开发板 led 驱动打印出来的信息，如下图所示：



```
Serial-COM1 - SecureCRT
文件(F) 编辑(E) 查看(V) 选项(O) 传输(T) 脚本(S) 工具(I) 帮助(H)
Serial-COM1
168.1.1,
[ 5.174397] host=iTOP, domain=, nis-domain=(none),
[ 5.179627] bootserver=192.168.1.77, rootserver=192.168.1.77, rootpath=
[ 5.198519] VFS: Mounted root (nfs filesystem) on device 0:11.
[ 5.202945] Freeing init memory: 228K
[ 6.375877] eth0: IPv6 duplicate address fe80::200:ffff:feff:0 detected!
mkdir: can't create directory '/mnt/disk': File exists
Try to bring eth0 interface up.....NFS root ...Done

Please press Enter to activate this console. [ 63.205975] CPU1: shutdown
[ 63.524142] failed to copy MFC F/W during init

[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]#
[root@iTOP-4412]# [ 434.905851] LEDS_CTL DEBUG:Device Opened Success!
[ 434.909097] debug: leds_ioctl cmd is 1
[ 434.912836] debug: leds_ioctl cmd is 1
[ 434.916862] LEDS_CTL DEBUG:Device Closed Success!

就绪 Serial: COM1, 115200 21, 1 21行, 80列 VT100 大写 数字
```

至此通过 web 页面控制 led 的实验已经完成了，大家可以参照这个例程来自己实现一个控制蜂鸣器的 web 程序。