

## boa服务器移植

# 嵌入式Linux中的Web服务器应用

Boa是一个非常小巧的Web服务器，其可执行代码只有60K左右。它是一个单任务的Web服务器，只能依次完成用户的请求，而不会fork出新的进程处理并发连接请求。但boa支持cgi，能够为cgi程序fork出一个进程来执行。Boa的设计目标是速度和安全，在其站点公布的性能测试中，boa的性能要好于apache服务器。

随着网络技术的迅猛发展，在嵌入式设备的管理和交互中，基于Web方式的应用成为目前的主流，用户可以直接通过远程登录的方式对设备进行管理和维护，大大方便了使用性。下面就为大家讲解一下boa服务器在嵌入式Linux系统中的移植过程。

### 一、BOA服务器移植

工具链：gcc version 4.3.2 (crosstool-NG-1.8.1-none)

平台：处理器：s3c2410 内核：linux-2.6.35

#### 1.解压源码

```
tar xvf boa-0.94.13.tar.tar
```

```
cd boa-0.94.13
```

#### 2.进入src/

```
./configure 生成Makefile
```

```
修改Makefile
```

```
修改CC = gcc 为 CC = arm-none-linux-gnueabi-gcc
```

```
修改CPP = gcc -E 为 CPP = arm-none-linux-gnueabi-gcc -E
```

#### 3.make编译

编译一个linux下的c系统，包含词法和语法分析模块，Linux上用bison和flex。

yacc是一个文法分析器的生成器,bison即是yacc的GNU版本.Lex和YACC是用于构造词法分析机和语法解释器的工具，利用Lex和YACC你可以轻松的构造一个语法解释器。

#### 1. 下载Lex和YACC

```
Apt-get install bison flex
```

#### 2. 修改 src/compat.h

找到 #define TIMEZONE\_OFFSET(foo) foo##->tm\_gmtoff

修改成

```
#define TIMEZONE_OFFSET(foo) (foo)->tm_gmtoff
```

#### 3. 修改 src/log.c

注释掉

```
if (dup2(error_log, STDERR_FILENO) == -1) {  
    DIE("unable to dup2 the error log");  
}
```

为:

```
/*if (dup2(error_log, STDERR_FILENO) == -1) {  
    DIE("unable to dup2 the error log");  
}*
```

#### 4. 修改src/boa.c

注释掉

```
if (passwdbuf == NULL) {  
    DIE("getpwuid");  
}  
if (initgroups(passwdbuf->pw_name, passwdbuf->pw_gid) == -1) {  
    DIE("initgroups");  
}
```

为

```
#if 0
```

```
if (passwdbuf == NULL) {  
    DIE("getpwuid");  
}  
if (initgroups(passwdbuf->pw_name, passwdbuf->pw_gid) == -1) {  
    DIE("initgroups");  
}
```

```
#endif
```

注释掉

```
if (setuid(0) != -1) {
```

```

        DIE("icky Linux kernel bug!");
    }
    为
    #if 0
        if (setuid(0) != -1) {
            DIE("icky Linux kernel bug!");
        }
    #endif

```

执行make  
 然后给boa瘦身  
 Arm-none-linux-gnueabi-strip boa

## 二、Boa服务器配置

### 1、创建目录

mkdir /source/rootfs/etc/boa

### 2、将boa源码目录下的boa.conf拷贝到/source/rootfs/etc/boa目录下

cp boa.conf /source/rootfs/etc/boa

### 3、修改配置文件boa.conf

vim /source/rootfs/etc/boa

#### (1)Group的修改

修改 Group nogroup

为 Group 0

#### (2)user的修改

修改 User nobody

为 User 0

#### (3)ScriptAlias的修改

修改ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/

为 ScriptAlias /cgi-bin/ /www/cgi-bin/

#### (5)DocumentRoot的修改

修改DocumentRoot /var/www

为 DocumentRoot /www

#### (6)ServerName的设置

修改#ServerName [www.your.org.here](http://www.your.org.here)

为 ServerName [www.your.org.here](http://www.your.org.here)

否则会出现错误“gethostbyname::No such file or directory”

#### (7)AccessLog修改

修改AccessLog /var/log/boa/access\_log

为#AccessLog /var/log/boa/access\_log

#### (8)以下配置和boa.conf的配置有关，都是在ARM根文件系统中创建

以下步骤在开发板上进行：

创建HTML文档的主目录/www

mkdir /www

创建CGI脚本所在目录 /www/cgi-bin

mkdir /www/cgi-bin

当不能使用cgi时

(重点)将#AddType application/x-httpd-cgi cgi改为AddType application/x-httpd-cgi cgi

## boa器测试

将boa拷贝到开发板根文件系统的/etc/boa下

#cp src/boa /source/rootfs/etc/boa

将ubuntu下/etc/mime.types拷贝到开发板根文件系统的/etc下

#cp /etc/mime.types /source/rootfs/etc

将你的主页index.html拷贝到www目录下

运行boa,然后在主机浏览器输入开发板网址

```
[root@farsight boa]# ./boa
```

```
[30/10/2011:19:10:36 +0000] [root@farsight boa]# boa: server version Boa/0.94.13
```

```
[30/10/2011:19:10:36 +0000] boa: server built 10 30 2011 at 19:10:36
```

```
[30/10/2011:19:10:36 +0000] boa: starting server pid=968, port 80
```

## 附1)：boa配置文件参数说明

boa的配置文件是/etc/boa/boa.conf。

Port: boa服务器监听的端口，默认的端口是80。如果端口小于1024，则必须是root用户启动服务器。

Listen: 绑定的ip地址。不使用这个参数时，将绑定所有的地址。

User: 连接到服务器的客户端的身份，可以是用户名或UID。

Group: 连接到服务器的客户端的组，可以是组名或GID。

ServerAdmin: 服务器出故障时要通知的邮箱地址。

ErrorLog: 指定错误日志文件。如果路径没有以“/”开始，则相对于ServerRoot路径。没有配置时默认的文件是/dev/stderr。若不想记录日志，指定文件为/dev/null。

AccessLog: 设置存取日志文件，与ErrorLog类似。

UseLocaltime: 设置使用本地时间，使用UTC时注释这个参数。这个参数没有值。

VerboseCGILogs: 在错误日志文件中记录CGI启动和停止时间，若不记录，注释这个参数。这个参数没有值。

ServerName: 指定服务器的名称，当客户端使用gethostname + gethostbyname时返回给客户端。

VirtualHost: 虚拟主机开关。使用此参数，则会在DocumentRoot设定的目录添加一个ip地址作为新的DocumentRoot来处理客户端的请求。如DocumentRoot设置为/var/www，则http://localhost/ 则转换成/var/www/127.0.0.1/，若注释此参数，则为/var/www/。

DocumentRoot: HTML文件的根目录（也就是网站的目录）。

UserDir: 指定用户目录。

DirectoryIndex: 指定预生成目录信息的文件，注释此变量将使用DirectoryMaker变量。这个变量也就是设置默认主页的文件名。

DirectoryMaker: 指定用于生成目录的程序，注释此变量将不允许列目录。

DirectoryCache: 当DirectoryIndex文件不存在，而DirectoryMaker又被注释掉时，将列出这个参数指定目录给客户端。

KeepAliveMax: 每个连接允许的请求数量。如果将此值设为"0"，将不限制请求的数目。

KeepAliveTimeOut: 在关闭持久连接前等待下一个请求的秒数。（秒）。

MimeTypes: 设置包含mimetypes信息的文件，一般是/etc/mime.types。

DefaultType: 默认的mimetype类型，一般是text/html。

CGIPath: 相当于给CGI程序使用的\$PATH变量。

SinglePostLimit: 一次POST允许最大的字节数，默认是1MB。

AddType: 增加MimeType没有指定的类型，例: AddType type extension [extension ...]。要使用cgi，必须添加cgi类型: AddType application/x-httpd-cgi cgi

Redirect: 重定向文件

Aliases: 指定路径的别名。

ScriptAlias: 指定脚本路径的虚拟路径。