Boa :是一种非常小巧的Web服务器，其可执行代码只有大约60KB左右。作为一种单任务Web服务器，Boa只能依次完成用户的请求，而不会fork出新的进程来处理并发连接请求。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。

CGI :在物理上是一段程序，运行在服务器上，提供同客户端HTML页面的接口。即客户端与服务器的接口。

比如留言本的工作流程：

先由用户在客户端输入一些信息，如名字之类的东西。

接着用户按一下“留言”（到目前为止工作都在客户端），浏览器把这些信息传送到服务器的CGI目录下特定的cgi程序中，于是cgi程序在服务器上按照预定的方法进行处理。

在本例中就是把用户提交的信息存入指定的文件中。然后cgi程序给客户端发送一个信息，表示请求的任务已经结束。此时用户在浏览器里将看到“留言结束”的字样。整个过程结束。

1.find / -name 'boa\*'      //查找文件

2.vi /etc/boa/boa.conf   //boa的配置文件

Port 80

User root

Group root

ErrorLog /dev/console

AccessLog /dev/null

ServerName friendly-arm   //服务器名称

DocumentRoot /www          //根目录

DirectoryIndex index.html   //默认首页

KeepAliveMax 1000

KeepAliveTimeout 10

MimeTypes /etc/mime.types   //默认类型

DefaultType text/plain

CGIPath /bin   //CGI目录

AddType application/x-httpd-cgi cgi  //对CGI文件的支持

3.删掉相关文件

rm -rf www

rm -rf /etc/boa

rm -rf /user/sbin/boa

4.开始移植

下载boa源码 www.boa.org

tar xzf boa-0.94.13.tar.gz

cd boa-0.94.13/src

make clean

vi Makefile

     修改CC和GCC

make

补充1：

UNIX和Linux中比较重要的三个设备文件是：/dev/console，/dev/tty和/dev/null。

0 :  /dev/console

这个设备代表的是系统控制台，错误信息和诊断信息通常会被发送到这个设备。

每个UNIX系统都会有一个指定 的终端或显示屏用来接收控制台信息。

1 : /dev/tty

如果一个进程有控制终端的话，那么 /dev/tty就是这个控制终端的别名。

像cron这样的进程是没有控制终端的，因此它也就无法打开/dev/tty。

     cron :计划任务，是任务在约定的时间执行已经计划好的工作，这是表面的意思。在Linux中，我们经常用到 cron 服务器来完成这项工作。cron服务器可以根据配置文件约定的时间来执行特定的作务。比如我们可以在配置文件中约定每天早上4点，对httpd 服务器重新启动，这就是一个计划任务；

tty 这个词源于Teletypes，最早是源于电传打印机。

如果你登录了一个shell，那么/dev/tty就是你当前使用的终端。你也 可以用tty命令查看/dev/tty设备实际链接到的设备是哪个。

/dev/tty其实就是类似于“符号链接”一样的东西。像我的 tty输出就是:

[root@wupengchong dev]# tty

/dev/pts/0

2 : /dev/null

这是个空设备，也称为“位桶bit bucket”。所有写向这个设备的输出都将被丢弃，而如果你读/dev/null，则会立即得到一个文件尾标志而返回。

在cp命令 里，可以用/dev/null来作为拷贝空文件的源文件。

在shell中，通常将不需要的东西重定向到/dev/null中。

这 里要提一下/dev/zero，它和null是有区别的。通俗的讲，/dev/null是一个饭桶，你可以无穷无尽的往里塞垃圾，它不会埋怨，而写进去的东西也就像掉进了黑洞。而/dev /zero是一个输入设备，它给你无穷无尽的提供0（就是null），它可以用于向文件或设备写入无穷无尽的0.

/dev/zero和 /dev/null的用法比较：

$dd if=/dev/zero of=mydoc.txt bs=1k count=1

$find . -name “\*.c” 2>/dev/nul   重定向到/dev/null

补充2：

  gcc -e 只编译不链接

     gcc -E 只激活预处理,这个不生成文件,你需要把它重定向到一个输出文件里面.

     例子用法:

     gcc -E hello.c > pianoapan.txt

     gcc -E hello.c | more