

GoAhead WebServer，它是一个源码免费、功能强大、可以运行在多个平台的嵌入式 WebServer。

GoAhead WebServer 的主要特性有：

1. 支持 ASP
2. 嵌入式的 javascript
3. 标准的 CGI 执行
4. 内存中的 CGI 处理 GoForms
5. 扩展的 API
6. 快速响应，每秒可处理超过 50 个请求
7. 完全和标准兼容
8. 如果不包含 SSI，仅要求 60K 的内存；包含 SSI，要求 500K 内存
9. web 页面可以存在 ROM 或文件系统中
10. 支持多种操作系统，包括 eCos、LINUX、LynxOS、QNX、VxWorks、WinCE、pSOS 等

目前 GoAhead WebServer 最新版本为 2.1.8，可在 <http://webserver.goahead.com> 上免费获取。下载可能需要填写一些信息，下载完成文件为 webs218.tar.gz。用解压工具打开即可看到其全部内容。解压后 ws031202 文件夹根目录下的文件为 GoAhead 的源代码文件，子文件夹对应各个支持系统，web 文件夹下面是一个示例 web 页面，同时也提供了 GoAhead 的帮助信息，当 webServer 建立后，可将 web 文件夹拷贝至您需要的 web 目录，然后直接访问即可，缺省情况下 GoAhead 支持 web 文件夹所有的内容。

目前 WebServer 中所支持的操作系统，都会有对应的 BSP 文件夹与之对应。在其中存在有支持该系统所必须的启动代码及系统相关代码。

转入正题，webServer 在移植过程相对简单，仅需简单的配置和操作就可以完成。

下面讲述一下通过 downloadable 工程来架设 GoAhead WebServer 的过程，当然也可以用类似的方法把它集成到 bootable 工程中。

第一步、新建一个 downloadable 工程，将下列下载的源文件加入到工程中：

```
balloc.c
base64.c
default.c
ejlex.c
ejparse.c
emfdb.c
form.c
h.c
handler.c
md5c.c
mime.c
misc.c
mocana_ssl.c
page.c
ringq.c
rom.c
security.c
sock.c
```

```

sockGen.c
sym.c
uemf.c
um.c
umui.c
url.c
value.c
webrom.c
webs.c
asp.c
websuemf.c
websda.c
cgi.c
/vxworks/main.c

```

在上述文件的列表中，几乎包含了 webServer 的全部文件，少了两个文件 websSSL.c 和 webcomp.c，（也就是说你可以将其外层所有的.c 文件包含后，再包含/vxworks/main.c，去掉上述两个文件即可。）websSSL.c 文件用在 ssl 上，如果需要使用 ssl 的话，还需要下载 openssl 系列的文件，此处省略。webcomp.c 文件用在网页压缩上，功能是将网页内容全部转换成 C 源代码，一起编译进 vxworks。考虑到嵌入式系统网页内容一般比较小，此处不考虑页面压缩。

第二步、完成需要修改该项目的配置，打开该项目的编译属性，将文件所在路径增加进 Include 路径中，同时在编译属性中增加：-DWEBS -DUEMF -DOS="VXWORKS" -DVXWORKS -DUSER_MANAGEMENT_SUPPORT，这些选项中定义了 vxworks 系统，用户管理等属性，如需要增加存取控制，可再增加-DDIGEST_ACCESS_SUPPORT。此部分的修改可在 MakeFile 文件中实现，作者修改完成的 MakeFile 文件中如下：

```

CFLAGS = -g -mcpu = pentium
-march = pentium -nostdlib -fno-builtin -fno-defer-pop -I. -I$(WIND_BASE)/target/h/
-I$(PRJ_DIR)/ws031202 -IE:/Mpe5/mpe5/head -le:/mpe5/mpe5/Templet/head
-le:/Mpe5/mpe5/Templet/head/include -IE:/Mpe5/mpe5/Templet -IE:/Mpe5/mpe5/interface
-DCPU=PENTIUM -DTOOL_FAMILY=gnu -DTOOL=gnu -DWEBS -DUEMF -DOS="VXWORKS"
-DVXWORKS -DUSER_MANAGEMENT_SUPPORT。

```

在两种方式中建议修改编译属性，如果非常熟悉 makefile 则可修改 Makefile 文件。

注：makefile 可以把宏传递给源文件，作用就相当于在预编译的每个文件中定义了这样的宏。

第三步、这一步要根据实际系统要求来进行修改 main.c。

将下面列出的定义按照您系统的实际情况进行修改，

```

#define ROOT_DIR T("/ata0a")
static char_t *rootWeb = T("/web"); /* Root web directory */
static char_t *password = T(""); /* Security password */
static int port = 80; /* Server port */

```

修改后的系统 web 文件放在/ata0a/web 中，没有安全密码，使用缺省 80 端口进行通讯。

另外 websSetDefaultPage(T("home.asp")); 中定义了缺省页面，即用户直接输入 IP 地址后进入时所看到的页面，缺省为 home.asp，更换成您所需要的文件即可。利用搜索工具查找一下 home.asp 可能不止一处，将碰到的都修改完成。

最后一步、到现在为止，该 downloadable 项目可以正确编译了，编译完成后下载后，

必需用命令启动它，最简单的方式是 `main(0,0)`，更可靠的方式为添加任务。

对于 `bootable` 项目，可在 `usrAppinit()` 中加入 `web server` 入口 “`main(0,0)`”

`web server` 只用了系统的 `socket` 协议栈和定时器，自己没有建任务，所以为了性能监视方便，建议专门建一个任务给他。

这样 `web server` 的内存使用、任务状态就可以看到了。任务堆栈的大小在刚开始时可给出一个较大的值如 `200k`，然后不断观察其最大值后再调整，`web` 页面的不同其占用就会发生变化，一般情况下（没有 `ssl`）`50K` 足够了。

对于有些 `OS` 没有完整的 `TCP stack`，对于 `GoAhead WebServer` 的移植就有点麻烦了，还得按照标准的协议栈补全。

如果大家碰到什么问题，欢迎和我讨论，fish_penny@163.com，fish.dubo@gmail.com