HANDOUT
Enable CGI Server
HL
I. Background

Web servers retrieve a file from the storage, .e.g., Flash in the case of embedded systems, and send it down the wire to the requesting browser. For example, you type in http://192.168.0.128/index.html , here is how BOA, and the rest of the Web servers, works in general:

1. The browser needs three pieces of information:
   a. The protocol, "http" in this case
   b. The server IP address, 192.168.0.128
   c. The file name "index.htm."
2. The browser use this IP address to connect to BOA.
   a. If the IP address is not know, use “ifconfig” to find out. Or you can use “ifconfig eth0 192.168.0.128” to set its IP address to 192.168.0.128 or whatever IP address after the system was boot.
   b. In the future lab, we will have to register for a temporary domain name. So let’s say, in the case, you have domain registration, www.whatever.com, then the Browser communicates with a name server to translate the server name into an IP Address to connect to BOA (your development board).
3. The browser then formed a connection to the server at that IP address on port 80, other port numbers are given in Figure 1.

Some some common port numbers:

- *echo 7*
- *daytime 13*
- *qotd 17 (Quote of the Day)*
- *ftp 21*
- *telnet 23*
- *smtp 25 (Simple Mail Transfer, meaning e-mail)*
- *time 37*
- *nameserver 53*
- *nicname 43 (Who Is)*
- *gopher 70*
- *finger 79*
- *WWW 80*

Figure1. Common Port Numbers.

4. Following the HTTP protocol, the browser sent a GET request to the server located at /bin/boa asking for the file "http:/192.168.0.128/index.html" (cookies may be sent from browser to server with the GET request). Since you have mapped the document root directory of boa at /etc/ by “$/bin/boa –c /etc &”, the file “index.html” at /etc/ can be retrieved.
5. The server then sent the HTML text for the Web page to the browser. (Cookies may also be sent from server to browser in the header for the page.) A simple html text is given in Figure 2.

```
<html>
  <body>
     <h1>Hello there!</h1>
  </body>
</html>
```

Figure 2. A simple html text.

6. The browser read the HTML tags and formatted the page to display it onto your screen.

This is how all Web servers handle *static* files, the web page “index.html”. In order to provide dynamic information, e.g., the information, which is gathered and processed based on interaction with the outside world from the server, CGI (Common Gateway Interface) has to be employed.

II. CGI Brief

On most Web servers CGI mechanism has been standardized. In the normal directory tree that the server is the root, “cgi-bin” is a sub-directory. Any file requested from the “cgi-bin” directory should not simply be read and sent, but instead should be executed. Example: http://192.168.0.128/cgi-bin/mycgi…”

The output of the executed program is sent to the browser. For example, a C program, “mycgi” is executed and it produces output which is html text, as shown in Figure 3.

```
#include <stdio.h>

int main()
{
  printf("Content-type: text/html\n\n");
  printf("<html>\n");
  printf("<body>\n");
  printf("<h1>Hello there!</h1>\n");
  printf("</body>\n");
  printf("</html>\n");
  return 0;
}
```

Figure 3. A simple CGI program, whose output upon execution is the html text as shown in Figure 2.

So base on this concept, you may want to expand the simple C program by adding an additional module to deal with inquiry from the browser to display the counter (the number of hits, or visits since the server is up.)

III. Enable CGI Server

CGI server is built into BOA by default for the uClinux distribution. All you have to do is 2 steps: e.g., first, modify “boa.conf” to make sure “cgi-bin”, is mapped to BOA document root directory; Secondly, customize BOA Makefile to make sure cgi program is properly compiled, built, and placed at the proper directory of the kernel image.

Step 1. Note uClinux-dist\user\boa\src\boa.conf is where the configuration file for BOA is located. The first thing is to map the virtual location of /cgi-bin/ to the actual location /usr/lib/cgi-bin/. Since in uClinux distribution, we have use /etc/ as the document root of BOA server, so you can use it as the physical location for CGI programs. Edit "boa.conf" replacing

```
    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
```

by

```
    ScriptAlias /cgi-bin/ /etc/
```

Step 2. Modify BOA Makefile to build the CGI program to the kernel image. This will take 2 steps. First, compile, link and build the CGI program into BOA. In order to accomplish this, BOA's Makefile at /user/boa/src/Makefile has to be modified as shown in Figure4 from No.1 to No. 3. Secondly, add the built program and other files to the file systems in Figure 4 No.4.

The following changes should be made in the Boa's Makefile:

1. Add one line as:

   ```
   TESTEXEC = mycgi
   ```

   The line defines "mycgi" as the CGI executable to be built.

2. Add one line as:

   ```
   TESTOBJS = mycgi.o
   ```

   The line defines that "mycgi.o" as the temporary object file of "testcgi" during compilation.

3. Add multiple lines as:

   ```
   $(TESTEXEC): $(TESTOBJS)
           $(CC) $(LDFLAGS) -o $@
                           $(TESTOBJS) $(SSL_LIBS) $(EXTRALIBS) $(LDLIBS)
   ```

   The line defines the way to build "mycgi" including depending object ("mycgi.o"), the compiler ($(CC)), and the referenced libraries.

4. Add multiple lines under "romfs" entry as:

   ```
   romfs:
               $(ROMFSINST) /bin/$(EXEC)
               $(ROMFSINST) /etc/$(CONFIG)
               $(ROMFSINST) /etc/$(MIME)
               $(ROMFSINST) /etc/$(INDEX)
               $(ROMFSINST) /etc/$(TESTEXEC)
               $(ROMFSINST) /etc/mynotebook.html
   ```

Figure 4. Modify /user/boa/Makefile (see WK. Li report)

(END)