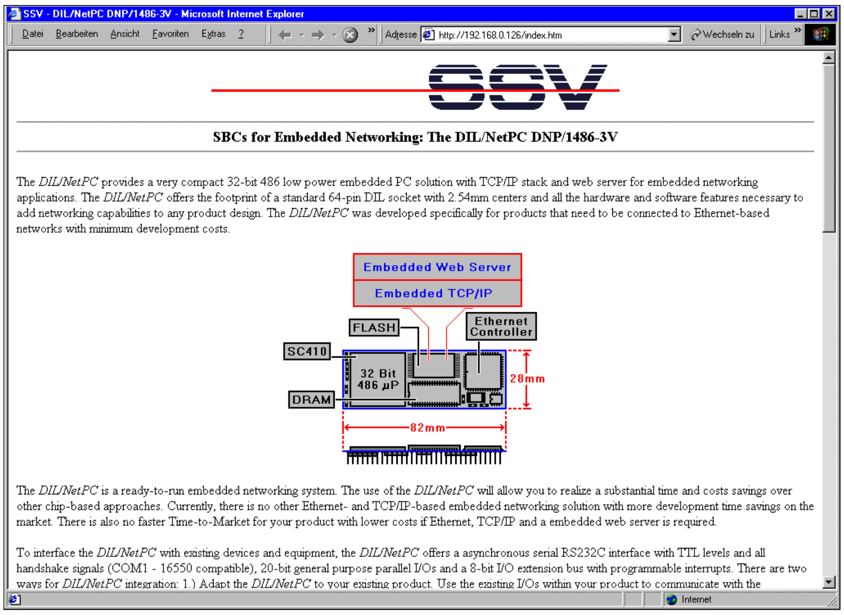
The *thttpd* web-server for Linux

A very important server for an embedded system under Linux is the web-server (HTTP-server). Such a server is based on the HTTP protocol (hypertext transfer protocol) and allows the access via web browser. Through that, e.g. an on-line maintenance or a remote configuration for an embedded system can be implemented. A graphical user interface (GUI) is often implemented with the help of a web-server.


SSV - DIL/NetPC DNP/1486-3V - Microsoft Internet Explorer
Datei Bearbeiten Ansicht Favoriten Extras ?
Adresse http://192.168.0.126/index.htm
Wechseln zu
Links
Embedded Web Server
Embedded TCP/IP
FLASH
Ethernet Controller
SC410
32 Bit 486 µP
DRAM
28mm
82mm
Internet

SBCs for Embedded Networking: The DIL/NetPC DNP/1486-3V

The *DIL/NetPC* provides a very compact 32-bit 486 low power embedded PC solution with TCP/IP stack and web server for embedded networking applications. The *DIL/NetPC* offers the footprint of a standard 64-pin DIL socket with 2.54mm centers and all the hardware and software features necessary to add networking capabilities to any product design. The *DIL/NetPC* was developed specifically for products that need to be connected to Ethernet-based networks with minimum development costs.

The *DIL/NetPC* is a ready-to-run embedded networking system. The use of the *DIL/NetPC* will allow you to realize a substantial time and costs savings over other chip-based approaches. Currently, there is no other Ethernet- and TCP/IP-based embedded networking solution with more development time savings on the market. There is also no faster Time-to-Market for your product with lower costs if Ethernet, TCP/IP and a embedded web server is required.

To interface the *DIL/NetPC* with existing devices and equipment, the *DIL/NetPC* offers a asynchronous serial RS232C interface with TTL levels and all handshake signals (COM1 - 16550 compatible), 20-bit general purpose parallel I/Os and a 8-bit I/O extension bus with programmable interrupts. There are two ways for *DIL/NetPC* integration: 1.) Adapt the *DIL/NetPC* to your existing product. Use the existing I/Os within your product to communicate with the

Figure 1: Access on a web-server

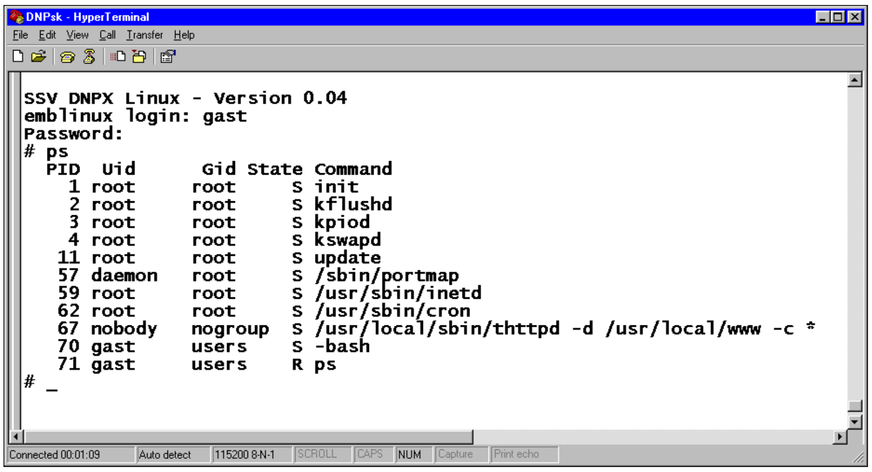
For many users, a web server in connection with Linux as operating system is identical with Apache. This legendary server owes his name the term "A Patchy Server", because it was put together from individual code- and patch files. Today Apache is – in according to an investigation of the Netcraft Survey – the most frequently used web server worldwide.

Apache is made for the presentation of web pages from companies or organizations in the Internet. You can see that on the capability and resource requirements of Apache. It is very unsuitable to use Apache on typical embedded systems considering the required resources. The capacity of Apache is also much higher as the actual requirements of an embedded system.

Additionally, on the most embedded systems you will find the thttpd (Tiny httpd-server) from the ACME Laboratories (ACME Labs – www.acme.com). This server is in the source code available under the GPL-conditions. But there are also even more interesting HTTP server for embedded systems (for e.g.

show on www.boa.org or also www.fhttpd.org). The thttpd from ACME Labs is to this point of time the most distributed of them. The reason for this is the excellent performance of the thttpd server.

Typically was a web-server under Linux started as a daemon. That means: the server runs into the background and waits for the try to connect from the client.



```
SSV DNPX Linux - Version 0.04
emblinux login: gast
Password:
# ps
  PID  Uid      Gid State Command
    1 root     root     S init
    2 root     root     S kflushd
    3 root     root     S kpiod
    4 root     root     S kswapd
   11 root     root     S update
   57 daemon   root     S /sbin/portmap
   59 root     root     S /usr/sbin/inetd
   62 root     root     S /usr/sbin/cron
   67 nobody   nogroup  S /usr/local/sbin/thttpd -d /usr/local/www -c *
   70 gast     users    S -bash
   71 gast     users    R ps
# _
```

Figure 2: Overview of the processes onto a DIL/NetPC

A HTTP-server make use of the TCP-protocol from a TCP/IP protocol stack and use the TCP-port 80 as default. Under this port number the server expects the call request by a web browser.

For the DIL/NetPC is a preconfigured root file system RIMAGE.GZ with the embedded web-server thttpd available. The server is stored in the directory `/usr/local/sbin`. Listing 1 shows the details of this directory. In the listing it is clearly to see that the thttpd-server requires only approx. 64-kByte memory space on the RAM-Disk. Within the packed root file system RIMAGE.GZ there is even a smaller memory space needed. For an immediate test of the thttpd you find some HTML- and GIF files in the directory `/usr/local/www.` The listing 2 shows the content of this subdirectory.

The figure 2 shows the command line that starts the thttpd during the Linux boot process. To become contact to the thttpd of a DIL/NetPC you only have to start an arbitrary web browser on a further computer. This computer must be in the same network like the DIL/NetPC.

Necessary as URL (Uniform Resource Locater) for the web browser are the IP address of the DIL/NetPC and the file name index.htm. An example of the URL-input is:

```
http://192.168.0.126/index.htm
```

After that, the browser build up a connection to the DIL/NetPC and shows the content of the file `index.htm` in the directory `/usr/local/www`. Please note, that the default IP address of a DIL/NetPC with embedded Linux is set on 192.168.0.126. If this IP address was changed you have to use the changed IP address in the URL.

```
List of /usr/local/sbin
drwxr-xr-x  2 root root    160 Jan 1  05:48 ./
drwxr-xr-x  4 root root    128 Nov 2  2000 ../
-rwxr-xr-x  1 root root  64540 Feb 20 2001 thttpd
```

Listing 1: The web server is located in `/usr/local/sbin`

The web server thttpd from the sample configuration in RIMAGE.GZ will be started automatically in run level 2 during the Linux boot process. The file `s50httpd` in the directory `/etc/rc2.d` is responsible for that.

```
List of /usr/local/www
drwxr-xr-x 3 root root   288 Jan  1 06:07 ./
drwxr-xr-x 4 root root   128 Nov  2 2000 ../
drwxrwxrwx 2 root root    64 Oct 23 2000 cgi-bin/
-r--r--r-- 1 root root  4765 Apr  7 2000 dilnpc24.gif
-r--r--r-- 1 root root  3546 Apr  7 2000 dilnpcbx.gif
-r--r--r-- 1 root root   769 Apr  7 2000 ssvlogo.gif
-r--r--r-- 1 root root 10028 Apr  7 2000 pinout.htm
-r--r--r-- 1 root root  7785 Apr  7 2000 index.htm
```

Listing 2: HTML files and GIF pictures in `/usr/local/www`

The run level is a digit for the designation of the current system status during the start-up of a Linux operating system. In every specific system status specific commands which are stored in the root file system directory `/etc/rc`**`n`**`.d` (**n** is in this case the digit for the respective run level) will be carried out. The file RIMAGE.GZ includes the directories /etc/rc0.d to /etc/rc6.d for a detailed system start control.

The *thttpd*-Parameter

The thttpd web server knows numerous parameters, which can be transmitted within a command line during the initial stage of a program. The example configuration in RIMAGE.GZ use the following command line with the two parameters -d and -c for the start:

```
thttpd -d /usr/local/www -c "*"
```

The parameter -d transmits the directory path with the HTML pages as files inside, onto the web server. Through that, the URL refers:

`http://192.168.0.126/`**`index.htm`**

on the file `/usr/local/www/`**`index.htm`**.

The directory, which is specified about the parameter `–d,` act as *default*-directory for web pages.

The parameter `-c` controls the possibilities to start programs via CGI (Common Gateway Interface). The chosen attitude `-c "*"` makes happen that only specific programs can be started by CGI. These programs must be built in according to special rules. During the try to start an arbitrary Linux-Executable (executable Linux program - normally a binary file in the 32 Bit-ELF-Format) via CGI send the thttpd an error report onto the web browser. With thttpd it is possible to activate almost every arbitrary Linux program via CGI. In this case the following command line should be used to start:

```
thttpd -d /usr/local/www -c "**"
```

Now, the parameter `–c` with the argument "**" will be used. This starts almost every arbitrary C program via CGI. The program should be located in the directory `/usr/local/www/cgi-bin`. The URL would be then:

```
http://192.168.0.126/cgi-bin/programm name
```

All outputs that were written to STDOUT from a C program via `printf (...)` will be send to the browser by thttpd. The browser shows then the texts.

In order to prove the CGI-functionality of the thttpd for the first time, you can use the *hello*-demo from directory `/sample/linux/hello` on the DIL/NetPC-Starterkit-CD-ROM.

```
int main (void)
{
   printf ("\nHello Embedded Linux User !!!\n\n");
   return (0);
}
```

Listing 3: C source code of the *hello*- demo

Please transmit the executable binary code (Linux-Executable) via FTP in the subdirectory `/usr/local/www/cgi-bin` on the RAM-Disk of a DIL/NetPC. Please note absolutely, that the Linux-Executable is equipped with the suitable rights after the FTP-transfer again (`chmod +x hello`). The program must be executable via command line. Otherwise it can not be started by CGI.

```
List of /usr/local/www/cgi-bin
drwxrwxrwx 2 root root     128 Jan  1 01:02 ./
drwxr-xr-x 3 root root     256 Nov  2  2000 ../
-rwxr-xr-x 1 gast users  33070 Jan  1 01:01 hello
```

Listing 4: Listing of the directory which contains the *hello*-file

Now the thttpd must be stopped. During the boot procedure it were started through the sample configuration in RIMAGE.GZ with the parameter `-c` and the argument `"*"`. Through that, only specific programs would be executable by CGI.

In order to stop a program under Linux you have to know the Process-ID (PID). For example you receive this ID via the Linux-command `ps`:

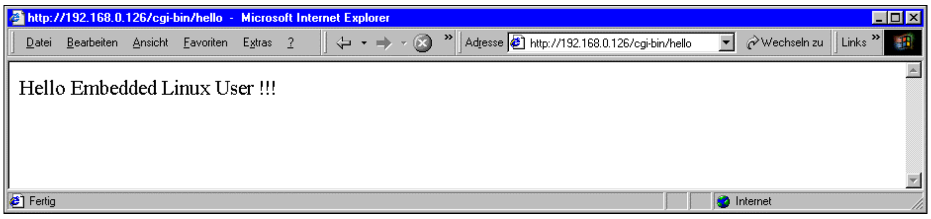
```
PID  Uid     Gid State Command
  1 root   root     S init
  2 root   root     S kflushd
  3 root   root     S kpiod
  4 root   root     S kswapd
 11 root   root     S update
 57 daemon root     S /sbin/portmap
 59 root   root     S /usr/sbin/inetd
 62 root   root     S /usr/sbin/cron
 67 nobody nogroup  S thttpd -d /usr/local/www -c *
 68 gast   users    S -bash
114 gast   users    R ps
```

Listing 5: Determining a PID (Process-ID)

In listing 4 you can see that the thttpd runs with PID=67. In this case you can stop the program and delete from the memory by using the Linux-command `kill 67`. Please consider that you can execute a `kill`-command only as super user (means with root rights). After that, you can start the thttpd again via the command line:

```
thttpd -d /usr/local/www -c "**"
```

The hello-example from the DIL/NetPC-Starterkit-CD-ROM is now executable as CGI-program.


http://192.168.0.126/cgi-bin/hello - Microsoft Internet Explorer
Datei Bearbeiten Ansicht Favoriten Extras ?
Adresse http://192.168.0.126/cgi-bin/hello
Wechseln zu
Links
Fertig
Internet

Hello Embedded Linux User !!!

Figure 3: Start of a CGI-program

The Figure 4 shows the access on the hello-example in the subdirectory `/usr/local/www/cgi-bin` of a DIL/NetPC with a web browser. As URL acts:

```
httpd://192.168.0.126/cgi-bin/hello
```

The output of the `printf (...)-` function – which can be recognized in listing 3 – is send from the thttpd onto the browser. In other words: if a program runs as a CGI-application, all outputs according STDOUT will be transmitted from the web-server to the web-browser.

Figures

Listings

Contact

SSV Embedded Systems
Heisterbergallee 72
D-30453 Hannover
Tel. +49-(0)511-40000-0
Fax. +49-(0)511-40000-40
E-mail: sales@ist1.de
Internet: www.ssv-embedded.de

Document History (Emblinx12e.doc)

Revision	Date		Name
1.00	29.08.2001	First Version.	KDW