SB100 device driver

sb1000 is a module network device driver for the General Instrument (also known as NextLevel) SURFboard1000 internal cable modem board. This is an ISA card which is used by a number of cable TV companies to provide cable modem access. It's a one-way downstream-only cable modem, meaning that your upstream net link is provided by your regular phone modem.

This driver was written by Franco Venturi <fventuri@mediaone.net>. He deserves a great deal of thanks for this wonderful piece of code!

Needed tools

Support for this device is now a part of the standard Linux kernel. The driver source code file is drivers/net/sb1000.c. In addition to this you will need:

1. The "cmconfig" program. This is a utility which supplements "ifconfig" to configure the cable modem and network interface (usually called "cm0");
2. Several PPP scripts which live in /etc/ppp to make connecting via your cable modem easy.

   These utilities can be obtained from:

   http://www.jacksonville.net/~fventuri/

   in Franco's original source code distribution .tar.gz file. Support for the sb1000 driver can be found at:

   - http://web.archive.org/web/%2E/http://home.adelphia.net/~siglercm/sb1000.html
   - http://web.archive.org/web/%2E/http://linuxpower.cx/~cable/

   along with these utilities.
3. The standard isapnp tools. These are necessary to configure your SB1000 card at boot time (or afterwards by hand) since it's a PnP card.

   If you don't have these installed as a standard part of your Linux distribution, you can find them at:

   http://www.roestock.demon.co.uk/isapnptools/

   or check your Linux distribution binary CD or their web site. For help with isapnp, pnpdump, or /etc/isapnp.conf, go to:

   http://www.roestock.demon.co.uk/isapnptools/isapnpfaq.html

Using the driver

To make the SB1000 card work, follow these steps:

1. Run `make config`, or `make menuconfig`, or `make xconfig`, whichever you prefer, in the top kernel tree directory to set up your kernel configuration. Make sure to say "Y" to "Prompt for development drivers" and to say "M" to the sb1000 driver. Also say "Y" or "M" to all the standard networking questions to get TCP/IP and PPP networking support.
2. **BEFORE** you build the kernel, edit drivers/net/sb1000.c. Make sure to redefine the value of READ_DATA_PORT to match the I/O address used by isapnp to access your PnP cards. This is the value of READPORT in /etc/isapnp.conf or given by the output of pnpdump.
3. Build and install the kernel and modules as usual.
4. Boot your new kernel following the usual procedures.
5. Set up to configure the new SB1000 PnP card by capturing the output of "pnpdump" to a file and editing this file to set the correct I/O ports, IRQ, and DMA settings for all your PnP cards. Make sure none of the settings conflict with one another. Then test this configuration by running the "isapnp" command with your new config file as the input. Check for errors and fix as necessary. (As an aside, I use I/O ports 0x110 and 0x310 and IRQ 11 for my SB1000 card and these work well for me. YMMV.) Then save the finished config file as /etc/isapnp.conf for proper configuration on subsequent reboots.
6. Download the original file sb1000-1.1.2.tar.gz from Franco's site or one of the others referenced above. As root, unpack it into a temporary directory and do a `make cmconfig` and then `install -c cmconfig /usr/local/sbin`. Don't do `make install` because it expects to find all the utilities built and ready for installation, not just cmconfig.
7. As root, copy all the files under the ppp/ subdirectory in Franco's tar file into /etc/ppp, being careful not to overwrite any files that are already in there. Then modify ppp@gi-on to set the correct login name, phone number, and frequency for the cable modem. Also edit pap-secrets to specify your login name and password and any site-specific information you need.
8. Be sure to modify /etc/ppp/firewall to use ipchains instead of the older ipfwadm commands from the 2.0.x kernels. There's a neat utility to convert ipfwadm commands to ipchains commands:

http://users.dhp.com/~whisper/ipfwadm2ipchains/

You may also wish to modify the firewall script to implement a different firewalling scheme.

9. Start the PPP connection via the script /etc/ppp/ppp@gi-on. You must be root to do this. It's better to use a utility like sudo to execute frequently used commands like this with root permissions if possible. If you connect successfully the cable modem interface will come up and you'll see a driver message like this at the console:

```
cm0: sb1000 at (0x110,0x310), csn 1, S/N 0x2a0d16d8, IRQ 11.
sb1000.c:v1.1.2 6/01/98 (fventuri@mediaone.net)
```

The "ifconfig" command should show two new interfaces, ppp0 and cm0.

The command "cmconfig cm0" will give you information about the cable modem interface.

10. Try pinging a site via `ping -c 5 www.yahoo.com`, for example. You should see packets received.
11. If you can't get site names (like www.yahoo.com) to resolve into IP addresses (like 204.71.200.67), be sure your /etc/resolv.conf file has no syntax errors and has the right nameserver IP addresses in it. If this doesn't help, try something like `ping -c 5 204.71.200.67` to see if the networking is running but the DNS resolution is where the problem lies.
12. If you still have problems, go to the support web sites mentioned above and read the information and documentation there.

Common problems

1. Packets go out on the ppp0 interface but don't come back on the cm0 interface. It looks like I'm connected but I can't even ping any numerical IP addresses. (This happens predominantly on Debian systems due to a default boot-time configuration script.)

Solution

As root `echo 0 > /proc/sys/net/ipv4/conf/cm0/rp_filter` so it can share the same IP address as the ppp0 interface. Note that this command should probably be added to the /etc/ppp/cablemodem script *right*between* the "/sbin/ifconfig" and "/sbin/cmconfig" commands. You may need to do this to /proc/sys/net/ipv4/conf/ppp0/rp_filter as well. If you do this to /proc/sys/net/ipv4/conf/default/rp_filter on each reboot (in rc.local or some such) then any interfaces can share the same IP addresses.

2. I get "unresolved symbol" error messages on executing `insmod sb1000.o`.

Solution

You probably have a non-matching kernel source tree and /usr/include/linux and /usr/include/asm header files. Make sure you install the correct versions of the header files in these two directories. Then rebuild and reinstall the kernel.

3. When isapnp runs it reports an error, and my SB1000 card isn't working.

Solution

There's a problem with later versions of isapnp using the "(CHECK)" option in the lines that allocate the two I/O addresses for the SB1000 card. This first popped up on RH 6.0. Delete "(CHECK)" for the SB1000 I/O addresses. Make sure they don't conflict with any other pieces of hardware first! Then rerun isapnp and go from there.

4. I can't execute the /etc/ppp/ppp@gi-on file.

Solution

As root do `chmod ug+x /etc/ppp/ppp@gi-on`.

5. The firewall script isn't working (with 2.2.x and higher kernels).

Solution

Use the ipfwadm2ipchains script referenced above to convert the /etc/ppp/firewall script from the deprecated ipfwadm commands to ipchains.

6. I'm getting *tons* of firewall deny messages in the /var/kern.log, /var/messages, and/or /var/syslog files, and they're filling up my /var partition!!!

Solution

First, tell your ISP that you're receiving DoS (Denial of Service) and/or portscanning (UDP connection attempts) attacks! Look over the deny messages to figure out what the attack is and where it's coming from. Next, edit /etc/ppp/cablemodem and make sure the ",nobroadcast" option is turned on to the "cmconfig" command (uncomment that line). If you're not receiving these denied packets on your broadcast interface (IP address xxx.yyy.zzz.255 typically), then someone is attacking your machine in particular. Be careful out there....

7. Everything seems to work fine but my computer locks up after a while (and typically during a lengthy download through the cable modem)!

Solution

You may need to add a short delay in the driver to 'slow down' the SURFboard because your PC might not be able to keep up with the transfer rate of the SB1000. To do this, it's probably best to download Franco's sb1000-1.1.2.tar.gz archive and build and install sb1000.o manually. You'll want to edit the 'Makefile' and look for the 'SB1000_DELAY'

define. Uncomment those 'CFLAGS' lines (and comment out the default ones) and try setting the delay to something like 60 microseconds with: '-DSB1000_DELAY=60'. Then do `make` and as root `make install` and try it out. If it still doesn't work or you like playing with the driver, you may try other numbers. Remember though that the higher the delay, the slower the driver (which slows down the rest of the PC too when it is actively used). Thanks to Ed Daiga for this tip!

Credits

This README came from Franco Venturi's original README file which is still supplied with his driver .tar.gz archive. I and all other sb1000 users owe Franco a tremendous "Thank you!" Additional thanks goes to Carl Patten and Ralph Bonnell who are now managing the Linux SB1000 web site, and to the SB1000 users who reported and helped debug the common problems listed above.

Clemmitt Sigler csigler@vt.edu