

Notes on running TR under Linux

Kevin Schmidt, W9CF
6510 S. Roosevelt St.
Tempe, Arizona 85283

1 Dosemu and Freedos

You need to have Dosemu with Freedos installed. These are available from <http://dosemu.org>.

1.1 Installing

Download the stable release and install it. I use the binary version of Freedos (currently beta 0.95) and compile the source for dosemu (currently dosemu-1.2.2.tgz). Follow the directions in the file INSTALL.

1.2 Configuration

After installation edit the global configuration file (/etc/dosemu/dosemu.conf by default). I added the following lines:

```
$_rdtsc = (on)
$_com3 = "/dev/ptyq1"
$_com4 = "/dev/ptyq2"
$_ports = " device /dev/lp0 fast range 0x378 0x37f"
$_ports = $_ports, " device /dev/null fast range 0x3f8 0x3ff"
$_ports = $_ports, " device /dev/null fast range 0x2f8 0x2ff"
```

The first line uses the pentium timer. I don't know if this is necessary, but I never turned it off. The second two lines tell dosemu to use the linux pseudotys /dev/ptyq1 and /dev/ptyq2 as the com3 and com4 serial ports rather than hardware serial ports. This will allow us to connect these fake

serial ports to linux applications like telnet or secure shell for packet or networking. Note that no other application should be using the pseudotys specified. The proper way to do this is to use system calls to open /dev/ptmx to get a unique pseudotty. Here we just pick some pseudotys and hope that no other application will use them. The last three lines gives Dosemu permission to use the port range needed for direct hardware serial and parallel port access so TRlog can key through the serial or parallel port. Notice that it associates the parallel port with the linux device /dev/lp0, so if something else in linux is using /dev/lp0, these ports will not be available.

My shack computer has two serial and one parallel port. I can tell TRlog to use serial 1 or 2 or parallel 1 for keying, etc. Packet and networking between computers running TR is done using the fake serial ports 3 or 4.

1.3 Running as root

For TRlog to have permission to have direct access to the parallel or serial ports, unfortunately, Dosemu must be run as root. The safest way to do this is to use visudo to edit the sudo configuration file. My shack computer is called boltzmann, so to allow user wlaw to run dosemu on boltzmann as root without typing a password, I add the lines

```
wlaw boltzmann=(root) NOPASSWD: /usr/local/bin/dosemu.bin
wlaw boltzmann=(root) NOPASSWD: /usr/local/bin/xdosemu
```

be sure to change wlaw, boltzmann, and the location of dosemu.bin and xdosemu to the appropriate values for your set up.

I start dosemu under X windows with the command

```
sudo xdosemu
```

2 Packet with telnet

With the configuration I use, com3 under dos is connected to pseudotty q1. With the shack computer connected to the internet (I have a dsl line at home with a wireless network, so the shack computer is connected whenever it is on.) After starting TRlog in the xdosemu window, with TRlog's logcfg.dat having the lines:

```
PACKET PORT = SERIAL 3  
EIGHT BIT PACKET PORT = TRUE
```

I open a shell and type:

```
stty icrnl erase '^h' < /dev/ttyq1  
telnet n7us.net < /dev/ttyq1 > /dev/ttyq1
```

The first line tells the pseudotty to use control-h as the backspace key and to add carriage-returns to the newlines. The second line connects the pseudotty to N7US's telnet cluster in Mesa, Arizona. You should use a cluster close to you. At this point I can use ctrl-b in TRlog to log in to the packet cluster and receive spots.

3 Connecting TRlog using linux networking

With the configuration I use, com4 under dos is connected to pseudotty q2. Turning on the networking in TRlog with

```
MULTI PORT = SERIAL 4
```

I then open a shell on each computer and type

```
ssh receivecomputer "cat < /dev/ttyq2" > /dev/ttyq2
```

where receivecomputer is the name of the computer I want to send my data to around the networking loop. Do the same on the other computers to complete the loop. I have assumed that they all are using pseudotty q2. If not adjust the devices accordingly when performing the ssh command. Notice that this uses whatever hardware you have set up for networking under linux. It will work with ethernet, wireless, parallel or serial port ip, ppp, ieee1394, or whatever else you have. My set up uses ethernet and wireless.

4 Voice Keyers

sbdvp is available at <http://fermi.la.asu.edu/w9cf> can integrate a sound card with TRlog. However, it only partially works under Dosemu. Dosemu emulates to some extent a soundblaster. Run sbdvp as under DOS. There are two main problems. First, sbdvp will *not* record under Dosemu. You must

record your files under linux before loading them with sbdvp. Second, under Dosemu, the last half second or so of audio is not played. Therefore you will need to make sure that a little dead air is left at the end of the messages.

Most linux distribution come with the sox package which contains frontends `rec` and `play`. For example, to record `cqf2.dvp` in an sbdvp compatible format, use

```
rec -r12048 -c1 -fu -b -t raw cqf2.dvp
```

You can play the file under linux as

```
play -r12048 -c1 -fu -b -t raw cqf2.dvp
```

to see how it sounds. The flags on these commands say to use a rate of 12048 Hz, one channel, i.e. not stereo, use unsigned, data is in bytes, write raw data to file `cqf2.dvp`.

External voice keyers should work as under DOS if the corresponding ports are set up correctly.