NSLU2

NSLU2

L.J.M. Dullaart

1. Introduction

Originally made as a cheap NAS, the NSLU2 seems capable of much more. I run a number of NSLU2s, because a) they are cheap b) they consume a lot less power than a normal PC. I will not copy all the installation websites. I will just give you a simple log of what I am doing, and give the URLs where you can find more. The stuff is organized per Linux-version.

Ah, yes, before we forget: Never use the Linksys EraseAll tool. It seems that there are still some fools out there, so here is the warning.

Another warning: this is what I did. You need to understand what you are doing and change things if they are different on your network or systems. So read it through and try to understand before you do anything.

1.1 Some hardware issues

I found that the Slug runs perfectly with a large USB-stick (4G or more). Unless, ofcourse, you will be using it as a torrent or mail server. Then you'll need more diskspace.

I've had some problems earlier on with USB -sticks. The problems seem to be dependend on the brand:

— GoodSony, Kingston

— Bad Danelec

Someone told me that this was the first sign of a failing power supply.

1.2 Others

There are also some experiences with the other (ie: non-debian) installs. I have removed them all because the debian installation is far more flexible and they provide no advantages over Debian.

2. Installation

2.1 Installation of Debian

Installation should be relatively easy using http://www.nslu2-linux.org/wiki/Debian/HomePage. However, I found that the installation process is almost always broken. I have never succeeded installing an NSLU2 using that process. Maybe it's me, but the first time, the installer was broken, the second NSLU2 the installer ran out of memmory, and the third it did not recognise the disks.

What does work is http://www.cyrius.com/debian/nslu2/unpack the manual way of doing it. It has the advantage of being faster as well, although you need another Linux system with a webservers and a DHCP server. In my DHCP-server, I have the folowing lines:

```
option domain-name "home";
option domain-name-servers 192.168.1.101;
option routers 192.168.1.3;
option broadcast-address 192.168.1.255;
option subnet-mask 255.255.255.0;
default-lease-time 600;
max-lease-time 7200;
subnet 192.168.1.0 netmask 255.255.255.0 {
  range 192.168.1.160 192.168.1.170;
  option domain-name-servers 192.168.1.101;
  default-lease-time 600;
  max-lease-time 7200;
}
```

The steps I did are below. Note that, in stead of sda, you might get a different driveletter, and instead of `/mnt`, you might want to mount somewhere else. Change accordingly if you want to reproduce.

First get the required images:

— the etch firmware

— the root-filesystem If the links are dead, please go to the unpack-page mentioned above. On that page you'll also find the check-sums.

Prepare the USB-stick. First partition the stick:

```
   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1         120      963868+  83  Linux
/dev/sda2             121         311     1534207+   5  Extended
/dev/sda5             121         132       96358+  82  Linux swap
/dev/sda6             133         311     1437786   83  Linux
```

Sizes may vary; I found that 2GB for sda1 is the minimum and 1GB for the swap is quite sufficient. Make the filesystems and swapspace:

```
mkfs.ext3 /dev/sda1
mkfs.ext3 /dev/sda6
mkswap /dev/sda5
```

Mount sda1:

```
mount /dev/sda1 /mnt
```

and untar the base image you got from http://people.debian.org/~tbm/nslu2/etch/base.tar.bz2:

```
cd /mnt
tar -xjvf /your/path/base.tar.bz2
```

That takes a while, so this is time for coffee.

To make things easier afterward, I put my ssh-key public key in `/root/.ssh/authorized_keys` and gave it a fixed IP-address. See furtheron how to do that.

Put the ETCH-image in the root-html of your webserver and reflash the NSLU2. Some people like to use specific software or the web-interface. I'm more a commandline lover, so I use redboot. A word of warning: with redboot you are on the bare metal. No safegards. If you mess-up, you can throw away your slug.

When the slug boots it will briefly allow you to telnet into 192.168.0.1 (netmask 255.255.255.0) and interact with the monitor. This monitor is called Redboot. It also means that you must be able to communicate with 192.168.0.1. I had to set an ifconfig alias:

```
ifconfig eth0:0 192.168.0.55
```

To telnet into Redboot, execute:

```
while ! ping -W 1 -c 1 192.168.0.1 2>&1 >/dev/null; do true; done && telnet
192.168.0.1 9000
```

and start the slug. As soon as Redboot answers, hit control-C. Sometimes, you have less than a second to do it, so be ready. At the prompt, first verify that you can indeed see your webserver, load the image and flash it:

```
ping -n 1 -h 192.168.0.55
load -r -b 0x01000000 -h 192.168.0.55 -m http /sda1-2.6.18.dfsg.1-23
fis write -f 0x50060000 -b 0x01060000 -l 0x7a0000
```

Connect the USB-stick and type reset into redboot.

2.2 Initial configuration

After a while, the log of your DHCP-server will give you the IP-address of your slug. Ssh into it; user root, password root and immediately change the password. Even if you are just at home. Even if no-one can get to your network.

Regenerate the ssh-keys:

```
rm /etc/ssh/ssh_host*
ssh-keygen -t dsa -f /etc/ssh/ssh_host_dsa_key -N ""
ssh-keygen -t rsa -f /etc/ssh/ssh_host_rsa_key -N ""
```

Add your users (using adduser), and add the appropriate lines in `/etc/sudoers` using `visudo`. Set the hostname in `/etc/hostname` and edit `/etc/network/interfaces`. Mine contains:

```
iface eth0 inet static
        address 192.168.1.102
        netmask 255.255.255.0
        gateway 192.168.1.3
        dns-search home
        dns-nameservers  192.168.1.101
```

I also changed mu hostname by vi-ing `/etc/hostname`. Make sure you can access the Internet from your slug (ping something). If name resolution isn't working: edit `/etc/resolv.conf`.

Install `sudo`

```
apt-get install sudo
```

Update your slug:

```
apt-get update
apt-get dist-upgrade
```

That also takes a while. If you had enough coffee, try some tea instead. Midway, you need to promise that you will reboot. So at the end, you reboot. Voila, your slug is ready.

3. DNS/DHCP server

3.1 Install the DNS-server

```
apt-get install bind
```

I do the configuration with the script below, which I placed in `/etc/bind`. It requires a simple description-file in the form of

```
fontaine     192.168.1.101
aesopos      192.168.1.102
phaedrus     192.168.1.103
```

But you can always do it by hand.

3.2 DHCP server

Getting a DHCP-server is simple as well.

```
apt-get install dhcpd
```

and edit /etc/dhcpd.conf. An example of what I use is somwhere above and you can get many other examples.

3.3 make_config.perl

```
#!/usr/bin/perl
$i_am=‘hostname‘; chomp $i_am;
$i_am_long=‘hostname -f‘;chomp $i_am_long;
$i_am_domain=‘hostname -d‘;chomp $i_am_domain;
$i_am_ip=‘hostname -i‘;chomp $i_am_ip;
# Read the hosts-file for this named
$infile=@ARGV[0];
open (INFILE,"<$infile") || die "cannot open $infile ; did you supply a source file as
argument? ";
@hosts=<INFILE>;
close INFILE;
$qty=0;
$zns=0;
for (@hosts){
        s/\#.*//;s/[   ]*$//;
        if (/^$/){
                print ’ ’;
        }
        else {
                ($h,$ip,$dhcp)=split;
                print "\n$h,$ip =";
                ($a,$b,$c,$d)=split(’\.’,$ip);
                @ip1[$qty]=$ip;
                $zn=sprintf("%03d%03d%03d",$a,$b,$c);
                print "$zn -";
                $fl=0;
                for ($i=0;$i<$zns;$i++){ if (@zones[$i] eq $zn) {$fl=1;}}
                if ($fl==0){
                        @zones[$zns]=$zn;
                        @nw[$zns]="$c.$b.$a";
                        print "@nw[$zns]";
                        $zns++;
                }
                $h=~s/\..*//;
                @hst[$qty]=$h;
                @hw[$qty]=$dhcp;
                $qty++;
        }
}
open (NDCONF, ">named.conf.local") || die "cannot write named.conf.local";
print NDCONF "zone \"$i_am_domain.\" in { type master ; file \"/etc/bind/db.$i_am_do-
main\"; };\n";
for($i=0;$i<$zns;$i++){
        print   NDCONF   "zone   \"@nw[$i].in-addr.arpa\"   in   {   type   master;   file
\"/etc/bind/db.@zones[$i]\"; };\n";
}
close NDCONF;
```

```
open (DB,">/etc/bind/db.$i_am_domain");
print DB "@       IN      SOA     $i_am_long. $i_am_long. (\n";
print DB "        1       ; serial\n";
print DB "        360000  ; refresh\n";
print DB "        3600    ; retry\n";
print DB "        960000  ; expire\n";
print DB "        36000   ; ttl\n";
print DB "        )\n";
print DB "        IN      NS      $i_am_long.\n";
for($i=0;$i<$qty;$i++){
       print DB "@hst[$i]    IN      A   @ip1[$i]\n";
}
close DB;
for ($i=0;$i<$zns;$i++){
       open (DB,">/etc/bind/db.@zones[$i]");
       print DB '$TTL    604800';
       print DB "\n@       IN      SOA     $i_am_ip  $i_am_long. (\n";
       print DB "     2       ; serial\n";
       print DB "     360000  ; refresh\n";
       print DB "     3600    ; retry\n";
       print DB "     960000  ; expire\n";
       print DB "     36000   ; ttl\n";
       print DB "     )\n";
       print DB "     IN      NS      $i_am_long.\n";
       for($j=0;$j<$qty;$j++){
              ($a,$b,$c,$d)=split('\.',@ip1[$j]);
              $zn=sprintf("%03d%03d%03d",$a,$b,$c);
              if (@zones[$i] eq $zn) {
                     printf    DB    ("%-4d     IN        PTR      @hst[$j].$i_am_do-
main.\n",$d);;
              }
       }
       close DB;
}
print "\n";
system "pkill -HUP named";
open (DHCP,">/etc/dhcpd.conf") || die "Cannot open dhcpd.conf";
print DHCP "option domain-name \"home\";\n";
print DHCP "option domain-name-servers 192.168.1.101;\n";
print DHCP "option routers 192.168.1.3;\n";
print DHCP "option broadcast-address 192.168.1.255;\n";
print DHCP "\n";
print DHCP "option subnet-mask 255.255.255.0;\n";
print DHCP "default-lease-time 600;\n";
print DHCP "max-lease-time 7200;\n";
print DHCP "\n";
print DHCP "subnet 192.168.1.0 netmask 255.255.255.0 {\n";
```

```
print DHCP "  range 192.168.1.160 192.168.1.170;\n";
print DHCP "  option domain-name-servers 192.168.1.101;\n";
print DHCP "  default-lease-time 600;\n";
print DHCP "  max-lease-time 7200;\n";
print DHCP "}\n";
# Hosts which require special configuration options can be listed in
# host statements.   If no address is specified, the address will be
# allocated dynamically (if possible), but the host-specific information
# will still come from the host declaration.
print "\n\n";
for($i=0;$i<$qty;$i++){
        print "host @hst[$i] @hw[$i]\n";
        if (@hw[$i]=~/:/){
                @ds=split (',',@hw[$i]);
                for (@ds){
                        print DHCP "host @hst[$i] {\n";
                        print DHCP "hardware ethernet $_;\n";
                        print DHCP "option domain-name-servers 192.168.1.101;\n";
                        print DHCP "fixed-address @hst[$i].home;\n";
                        print DHCP "}\n\n";
                        print "host @hst[$i] $_;\n";
                }
        }
}

system "/etc/init.d/dhcp restart";
```

4. Installation of Apache

Installing and configuring a webserver is a bit more complicated. The reason is, that Apache2 is becoming more sophisticated, and therefore less suitable for simple webpages. But Apache still is the standard. So here we go.

Installing is easy; `apt-get install apache2`

4.1 Configure a simple website

What I basically did is put a file 000-default in `/etc/apache2/sites-enabled` with the following content:

```
NameVirtualHost *
<VirtualHost *>
        ServerAdmin webmaster@localhost
        DocumentRoot /home/www/html/
        <Directory />
                 Order Deny,Allow
                 Deny from all
        </Directory>
        <Directory /home/www/>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
                Order allow,deny
                allow from all
                # This directive allows us to have apache2's default start page
                # in /apache2-default/, but still have / go to the right place
                RedirectMatch ^/$ /ljm/
        </Directory>
        ScriptAlias /cgi/ /home/www/cgi/
        <Directory "/home/www/cgi">
                AllowOverride None
                Options ExecCGI -MultiViews +SymLinksIfOwnerMatch
                Order allow,deny
                Allow from all
        </Directory>
        ErrorLog /var/log/apache2/error.log
        # Possible values include: debug, info, notice, warn, error, crit,
        # alert, emerg.
        LogLevel warn
        CustomLog /var/log/apache2/access.log combined
</VirtualHost>
```

I also created the site:

```
mkdir /home/www
mkdir /home/www/html
mkdir /home/www/html/ljm
mkdir /home/www/cgi
chmod -R a+rx /home/www /home/www/html /home/www/cgi
```

and in `/home/www/html/ljm` I put a `index.html` with some happy message.

And that seemed to work.

4.2 Other webserver-stuff

If your webserver needs to display only static pages, this is enough. I wanted more.

First is webmail. I have my own mail server at home, so I want to be able to access my mail via the Internet. So I use squirrelmail. It requires php5, which is installed with apt-get.

Next is the posibility to display my photos. The simplest is Simpleviewer There are other flashes on that site that look nice as well. I'll try them another time.

5. An NFS server

Although I like to try a different stuff (see further), I need a reliable nfs server for archiving purposes. Debian has proved to be a reliable installation, so I decided to go back to debian for my NFS. First install the nfs server:

```
apt-get install nfs-kernel-server nfs-common portmap
```

All my disks are on a USB-hub. For a reliable boot, I have to unplug the hub and plug it in when the system is up. That means that the disks cannot be in /etc/fstab. I decided to make a little script that automatically mounts all that is connected and exports them as well (edit for your own use if you want it). For those interested:

```
aesopos:/# df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/sda1              2064144    351024   1608216  18% /
tmpfs                    14992         0     14992   0% /lib/init/rw
udev                     10240        92     10148   1% /dev
tmpfs                    14992         0     14992   0% /dev/shm
/dev/sda6              4610360      9144   4367016   1% /home
/dev/sdb1            480719056  95838860 360460996  22% /media/MAXTOR_B
/dev/sdc1            961432072 117935664 794658408  13% /media/WD_ELEMENTS_A
/dev/sdd1            961432072 110090048 802504024  13% /media/WD_ELEMENTS_C
/dev/sde1              2071384     68700   1897460   4% /media/Elements_D_sys
/dev/sde6            958318668    204568 909434344   1% /media/WD_ELEMENTES_D
/dev/sdf1            961432072    204568 912389504   1% /media/WD_ELEMENTES_E
```

That's 4.5 Terra. For mounting the stuff on clients, I use this script.

Why would anyone put so much diskspace on a relative slow access, you might ask. Well, as backup. I backup my important data every day with `cp -rup --backup=numbered` to those nfs-mounted disks. It is faster and less cumbersome than using tapes.

One drawback is that the nslu2 has not enough memmory to do an fsck on this kind of disks. So, when you reboot, you'll need to do the fsck on another computer before plugging them in. Of course, if you had a recent reboot and a clean shutdown, you do not need a complete fsck.

6. Installing a mailserver

One of my nslu2s is going to be a mailserver. That is not a simple task. If you look at all the mail programs and their configuration, it seems that anyone creating them has a severe psygological disorder.

I tried different howto's but none of them seems to work. The list of failed howto;'s are:

http://www.nslu2-linux.org/wiki/OpenSlug/MailServer http://lika.be/wp/2005/08/setting-up-the-nslu2-as-mail-server http://www.nslu2-linux.org/wiki/HowTo/SetUpAnEmailServer
http://www.nslu2-linux.org/wiki/HowTo/SetUpAnEmailServer2
http://www.nslu2-linux.org/wiki/HowTo/QMailOnTheNSLU2
http://lena.franken.de/linux/debian_and_vserver/sendmail.html http://www.aboutdebian.com/internet.htm
All seem to work up to a point, but most leave your slug is such a state that you need to reinstall.

They provide, however, some insight in what kind of configuration is needed First I installed the software:

```
apt-get install sendmail sendmail-bin uw-imapd uw-imapd-ssl mailx
```

This suggests to install `uw-mailutils sendmail-doc sasl2-bin libsasl2-modules libsasl2-modules-plain libsasl2-digestmd5-plain libsasl2-digestmd5-des cyrus-common logcheck mutt` and `imap-client`. It also asks whether you want to continue without maildir support. We'll continue without (anwer Yes). Check that following entries exist in `/etc/services file`.

```
    imaps           993/tcp
    pop3s           995/tcp
```

Add the following entries in /etc/inetd.conf :

```
    pop3s     stream  tcp     nowait  root    /opt/sbin/ipop3d      ipop3d
    imaps     stream  tcp     nowait  root    /opt/sbin/imapd       imapd
```

Run `newaliases`.

Now, connecting to phaedrus works. I used Thunderbird and connected through ssl/tls. It complains about an unsigned certificate. That is ok for me, because I am only on a small home-network.

AFter a while, your certificate will expire. Mail clients will comlpain and for your non-technical faniliy members, panic will break out. Simply regenerate the certificate with:

```
openssl req -new -x509 -nodes -days 365 \
-out /etc/ssl/certs/imapd.pem \
-keyout /etc/ssl/certs/imapd.pem
```

When it asks for a

```
Common Name (eg, YOUR name) []:
```

answer with the fully qualified hostname of your IMAP-server, phaedrus.home in my case.

But there is only a single empty inbox in the account. So let's send some mail:

```
phaedrus:/usr/local/bin# telnet 127.0.0.1 smtp
helo there
mail from: napoleon@elba.fr
rcpt to: ljm@phaedrus.home
data
Alons enfants de la patri-i-e..
.
quit
```

Trying to get mail now from Phaedrus hangs. Thunderbird seems to take ages to open the inbox. From the log, there is a complaint about a lost lock.

That means some additional configuration will be required. First the sendmail.mc:

```
include(/usr/share/sendmail/cf/m4/cf.m4)
VERSIONID(‘sendmail.mc - ljm 200906031210’)
OSTYPE(linux)dnl
define(‘SMART_HOST’,‘smtp.xs4all.nl’)dnl
FEATURE(‘masquerade_envelope’)dnl
FEATURE(‘always_add_domain’)dnl
FEATURE(‘local_procmail’, ‘/usr/bin/procmail’)dnl
FEATURE(‘genericstable’, ‘hash -o /etc/mail/genericstable.db’)dnl
GENERICS_DOMAIN(‘localhost pheadrus.home’)dnl
MAILER(local)dnl
MAILER(smtp)dnl
LOCAL_CONFIG
Cw localhost pheadrus.home
VERSIONID should be optional, but it is good to include it anyway
OSTYPE includes a set of defaults for the Linux OS.
FEATURE calls a set of predefined macros
MAILER gives the possible mail deliveries
dnl is just a list option
```

And pass it through m4:

```
m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf
```

I need the masquerade_envelope and genericstable, because I need to rewrite my originator address. Well, I might, if I decide that all outgoing mail needs to go through my mailserver, perhaps, one day,...

Restart sendmail:

```
/etc/init.d/sendmail restart
```

Make sure that the spool directories are accessible for all mail recepients. If you’re not too concerned about security, make the spool directory permissions 1777. And resend the mail from Napoleon. And then:

```
ljm@phaedrus:~$ mail
Mail version 8.1.2 01/15/2001.  Type ? for help.
"/var/mail/ljm": 3 messages 3 new
>N  1 napoleon@elba.fr   Tue Jun  2 20:34   12/435
p
Message 1:
From napoleon@elba.fr  Tue Jun  2 20:34:54 2009
Date: Tue, 2 Jun 2009 20:32:27 +0200
From: napoleon@elba.fr
To: undisclosed-recipients:;
Alons enfants de la patri-i-e..
```

Next is the IMAP. For some reason, it now works without additional configuration. Make sure that your mail client uses SSL!

Next is fetchmail.

```
apt-get install fetchmail
```

To be honest, I want to run this mail server a little while in parallel to my current server. So I made an extra mailbox at my provider for testing purposes. Let’s say the mailbox is called testmail.

In the home-directory of root I made a file .fetchmailrc with the following contents:

```
poll pop.xs4all.nl with proto POP3
        user "testmail", with password "ZeEr GeHeIM", is ljm here warnings 3600
```

Because there are passwords in the file, chmod 600 .fetchmailrc. Fetchmail won't run otherwise. And start fetchmail:

```
phaedrus:~# fetchmail -v -v -v
fetchmail: WARNING: Running as root is discouraged.
fetchmail: 6.3.6 querying pop.xs4all.nl (protocol POP3) at Wed Jun  3 19:26:50 2009:
poll started
Trying to connect to 194.109.6.55/110...connected.
fetchmail: POP3< +OK xs-pop3d (1.75 04-Dec-2008) at mailpop20.xs4all.nl starting
fetchmail: POP3> CAPA
fetchmail: POP3< +OK Kappa 10-4
fetchmail: POP3< TOP
fetchmail: POP3< USER
fetchmail: POP3< UIDL
fetchmail: POP3< LAST
fetchmail: POP3< RESP-CODES
fetchmail: POP3< .
fetchmail: pop.xs4all.nl: opportunistic upgrade to TLS failed, trying to continue.
fetchmail: POP3> USER testmail
fetchmail: POP3< +OK Password required for testmail.
fetchmail: POP3> PASS *
fetchmail: POP3< +OK testmail has 0 messages (0 octets)
fetchmail: selecting or re-polling default folder
fetchmail: POP3> STAT
fetchmail: POP3< +OK 0 0
fetchmail: No mail for testmail at pop.xs4all.nl
fetchmail: POP3> QUIT
fetchmail: POP3< +OK Updating mailbox - exit
fetchmail: 6.3.6 querying pop.xs4all.nl (protocol POP3) at Wed Jun  3 19:26:51 2009:
poll completed
fetchmail: not swapping UID lists, no UIDs seen this query
fetchmail: Query status=1 (NOMAIL)
fetchmail: Deleting fetchids file.
fetchmail: normal termination, status 1
fetchmail: Deleting fetchids file.
phaedrus:~#
```

That went well, but there was no mail! So next send some mail and we're done.

7. Alternatives

7.1 Unslung

One of my slugs will be used as an archive with big disks, preferrably more than 2. So I decided to try Unslung. So I downloaded the latest Unslung from http://www.slug-firmware.net/u-dls.php and put in the root of my webserver. Reboot & redboot:

```
while ! ping -W 1 -c 1 192.168.0.1 2>&1 >/dev/null; do true; done && telnet
192.168.0.1 9000
```

and flash:

```
ping -n 1 -h 192.168.0.55
load -r -b 0x01000000 -h 192.168.0.55 -m http /Unslung-6.10-beta.bin
fis write -f 0x50060000 -b 0x01060000 -l 0x7a0000
reset
```

DO NOT plug in any disks For some reason the slug turns up as 192.168.1.77, but gateway and DNS are correct. That's OK; we'll change that. At 192.168.1.77, you'll get a modified web-interface, where you can login as admin/admin. First set the network parameters. In that way, it won't interfere with my other slug-activities.

Now, apart from the nice Tux, nothing much changed compared to the original Linksys interface. So we'll first enable telnet. That is a non-obvious action. You need to simulate a web-client. See

http://www.nslu2-linux.org/wiki/FAQ/EnableTelnetThroughTheWebInterface.

```
telnet 192.168.1.103 80
POST /Management/telnet.cgi HTTP/1.1
Host: 192.168.1.103
Content-Type: application/x-www-form-urlencoded
Content-Length: 20
Connection: Keep-Alive
Authorization: Basic YWRtaW46YWRtaW4=
action=Enable+Telnet
```

Don't forget the blank lines before and after the action-line. As an answer, you'll get a web-page. If not, add an extra blank line.

DO NOT plug in any disks That enables telnet. You can login as admin/admin, but the slug logs you out immediately. Look at http://www.nslu2-linux.org/wiki/FAQ/CantTelnetToMyNewlyUnslungNSLU2. So, because we have not plugged-in any disks, we can login with root/uNSLUng. That does not lock us out.

Now, Unslung has a very unclear way of handling authentication. There are files all over the place. You'll find that, once you plugged in a disk, you suddenly cannot log-in anymore. If you plug in an extra disk, it often does not get recognized. Attaching via a HUB is also out of the question.

Behaviour of Unslung is unpredictable and it does not do more than the original NSLU2. So, although I got it working, I will remove it. It is just marginally better than the original Linksys firmware.

So that is the last you'll see of my unslung-adventure.

7.2 SlugOS

Another minimal slug replacement is slugOS. Download it from http://www.slug-firmware.net/, and put it in the root of your webserver. Telnet into redboot:

```
while ! ping -W 1 -c 1 192.168.0.1 2>&1 >/dev/null; do true; done && telnet
192.168.0.1 9000
```

and flash the stuff.

```
ping -n 1 -h 192.168.0.55
load -r -b 0x01000000 -h 192.168.0.55 -m http /slugosbe-4.8-beta-nslu2.bin
fis write -f 0x50060000 -b 0x01060000 -l 0x7a0000
reset
```

And the root-password is available from

http://www.nslu2-linux.org/wiki/FAQ/OpenSlugDefaultPassword.

After login, you'll need to initialize a bit; First run turnup init and answer the dialog:

```
root@pheadrus:~$ turnup init
Please enter a new password for 'root'.
The password must be non-empty for ssh login to succeed!
Changing password for root
Enter the new password (minimum of 5, maximum of 8 characters)
Please use a combination of upper and lower case letters and numbers.
Enter new password:
Re-enter new password:
Password changed.
Host name [pheadrus]:
Domain name [home]:
Boot protocol (dhcp|static) [static]:
IP address [192.168.1.103]:
IP netmask [255.255.255.0]:
IP gateway [192.168.1.3]:
First DNS server [192.168.1.2]: 192.168.1.101
Second DNS server []: 192.168.1.2
Third DNS server []:
turnup init: you must reboot for the changes to take effect
 You may want to run 'turnup preserve' to save these settings,
 after making any additional configuration changes which you
 require.
```

The USB-stick was neatly recognized. Next, we want to put the root filesystem on the usb-stick:

```
turnup disk -i /dev/sda1 -t ext3
/sbin/turnup: umounting any existing mount of /dev/mtdblock4
turnup: copying root file system
15213 blocks
done
turnup: initialising dev file system
turnup: ensuring /var/volatile mountpoint exists
turnup: ensuring tmpfs will not be mounted on /var
turnup: /etc/syslog.conf: changed to file buffering
 Old (buffer) version in /etc/syslog.conf.sav
 Log messages will be in /var/log/messages
root@pheadrus:~$  turnup preserve
```

And reboot.

After the reboot, I found everything in order. Even the big WD-elements disk was there:

```
root@pheadrus:~$ df
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/sda1              3617116     75968   3357408   2% /
/dev/mtdblock4            6528      4952      1576  76% /initrd
/dev/sda1              3617116     75968   3357408   2% /dev/.static/dev
tmpfs                    2048        36      2012   2% /dev
/dev/sda2              122839      4146    117425   3% /media/sda2
/dev/sdb1           961432072 117935664 794658408  13% /media/sdb1
tmpfs                   15188         0     15188   0% /var/volatile
tmpfs                   15188         0     15188   0% /dev/shm
```

7.2.1 NFS server

Ok; let's install the NFS-server. First update the package-list and install the NFS-server:

ipkg update ipkg install nfs-utils

Now this installs the NFS-server. However, it does not create an nfs-exports-file. For me, creating an nfs-export is simple; I want all the media that I attach exported. So I added the following lines to /etc/init.d/nfsserver

```
for dir in `mount  | grep media | cut -d ' ' -f 3` ; do
        echo "$dir  *.home(sync,no_root_squash)"
done > /etc/exports
```

So next we'll stop/start nfs:

```
root@pheadrus:/$ cd /etc/init.d
root@pheadrus:/etc/init.d$ ./nfsserver stop
stopping statd: done
stopping mountd: done
stopping nfsd: done
removing nfsd kernel module: done
root@pheadrus:/etc/init.d$ ./portmap stop
Stopping portmap daemon: portmap.
root@pheadrus:/etc/init.d$ ./portmap start
Starting portmap daemon: portmap.
root@pheadrus:/etc/init.d$ ./nfsserver start
creating NFS state directory: done
starting 8 nfsd kernel threads: done
starting mountd: done
starting statd: done
root@pheadrus:/etc/init.d$
```

On the client-side, you may want to do some scripting as well. I use the following script to mount everything from phaedrus:

```
!/bin/bash
### Am I root?
if [ `id -u` -gt 0 ] ; then
        echo "You must be root"
        exit 1
fi
### Mount all exports from Phaedrus in a separate directory
/usr/sbin/showmount -e phaedrus |  grep -v xport|
while read exp perm ; do
        dir=${exp##*/}
        if [ ! -d /room/n20/$dir ] ; then
             mkdir /room/n20/$dir
        fi
        if mount | grep /room/n20/$dir  /dev/null ; then
             echo $dir already mounted.
        else
             mount phaedrus.home:$exp /room/n20/$dir
        fi
done
```

This needs some polishing ofcourse, but it's basic idea should be clear. In the end, it provides no extra advantages over a debian distribution. So I went back to debian.

8. Sane remote scanner

Connect a scanner to your nslu2 and install Sane:

```
apt-get install sane
apt-get install libsane libsane-extras sane-utils
```

Put the sane-daemon in inetd.conf

```
sane-port  stream  tcp  nowait  root  /usr/sbin/saned saned
```

Yeah, I know. It should run as saned. But I'm on a home-network and not on a corporate network and I don't feel like setting permissions on the USB-devices. I'm lazy.

Allow computers to have access. Put

```
192.168.1.10
192.168.1.11
192.168.1.101
127.0.0.1
```

in /etc/sane.d/saned.conf. These are the IP-addresses of the hosts that can access the scanners. Put

```
localhost
```

in /etc/sane.d/net.conf and test your installation with scanimage -L:

```
fontaine:~# scanimage -L
device 'coolscan2:usb:libusb:001:004' is a Nikon    LS-40 ED          film scanner
device   'net:localhost:coolscan2:usb:libusb:001:004'   is   a   Nikon      LS-40   ED
film scanner
```

You see that the scanner is visible direct and through the network. So that's that.

On your client, put the scanner server name in /etc/sane.d/net.conf. On the client, you might get something like:

```
device 'v4l:/dev/video0' is a Noname PC Camera virtual device
device   'net:fontaine.home:coolscan2:usb:libusb:001:004'   is   a   Nikon      LS-40   ED
film scanner
device  'hpaio:/usb/Deskjet_F4200_series?serial=CN8A83C26W05BR'  is  a  Hewlett-Packard
Deskjet_F4200_series all-in-one
```

9. problems

9.1 Old stuff; repository not found

There comes a time that your nslu2 should have been upgraded but for some reason, it wasn't. And then suddenly apt-get stops working. The solution is to upgrade, but for that you need a repository. So, put the line

```
deb http://archive.debian.org/debian/ etch main
```

in `/etc/apt/sources.list`. Now it works again.

9.2 A failing power supply

The powersupply of the NSLU2 is weak. It must be replaced with something new after a year. On the website of the Debian NSLU2, many recomendations are done. I have more than one SLUG, so I decided to go for a better solution.

Oh, and despite what others say, a failing powersupply can break your slug. So replace it early.

CONTENTS