PLIP Install HOWTO

Table of Contents

Table of Contents

PLIP Install HOWTO

Gilles Lamiral, `lamiral@mail.dotcom.fr`

v1.27, 11 August 2000

---

*This paper describes how to install a **Gnu|Linux** distribution on a computer without Ethernet card, or CD−ROM, but just a **local floppy drive** and a **remote nfs server** attached by a **Null−Modem parallel** cable. The nfs server has a **cdrom drive** mounted and exported.*

You can find the latest release of the PLIP−Install−HOWTO, by Gilles Lamiral, located at: `http://www.linuxdoc.org/HOWTO/PLIP-Install-HOWTO.html`

If you have already read this HOWTO and you wonder what is new is this release, first read the section what's new.

1.Introduction

2.License

3.How did I write this howto ?

4.What do you need ?

5.The documentation

6.Conventions

16.Install the plip interface permanently

17.Compatibility with a printer (lp)

18.Need to compile a new kernel ?

19.What's new

20.To do

21.List of contributors and acknowledgements

1.Introduction

"PLIP is Parallel Line IP, that is, the transportation of IP packages over a parallel port." Donald Becker (becker@super.org).

Writing a PLIP install HOWTO seems useless nowadays since Ethernet cards are cheap: NE2000 cards cost about the same as a Null−Modem cable. This is true for desktop computers, but not for laptop computers, in which the PCMCIA card costs more than 10 times as much the Null−Modem cable. Moreover, there is a parallel port on every computer, but not always a network card.

Of course, this howto can be used to install Linux on every personal computer without loss of generality.

This paper is just what I've done to install a Debian GNU/Linux distribution on a Toshiba Portégé 620CT laptop, from a NFS exported cdrom drive, via a Null−Modem cable.

A Null−Modem cable is also called a Null−Printer cable.

A Null−Modem cable is also called a LapLink (or Turbo Laplink) cable but the word LapLink is trademarked by Traveling Software under the number 75466713 since 1986, so I won't use it anymore.

This HOWTO will be obsolete when every Linux distribution include a PLIP install option. For example, the Debian installation only needs to add two commands to make this HOWTO obsolete (ifconfig + route). I hope one Debian maintainer will consider this point.

I hoped this HOWTO would become obsolete but in fact it seems that it will never be obsolete because some distributions builders do include PLIP installation sometimes but forget to maintain it in new releases (REDHAT for example).

I would be happy to know if someone used the PLIP−Install−HOWTO to install other Linux distributions from other network protocols (ftp, http, nfs, samba, or even NT/Novell servers)

Feedback with typos, bad English, comments, money, job, joy, fears, cries are welcome and recommended (not all with the same eagerness).

If a translation in any language exists, please let me know, I'll be happy.

2.License

I put this documentation under the OpenContent license. This means this document is given without any warranty. You can use it, redistribute it, or modify it, by respecting the following conditions: You have to clearly specify the nature and the content of the modifications, their date and keep the opencontent license in case of redistribution. If you sell it, you sell the medium, not the content. This is a summary, read the license to get more details.

For those who can't read the license online, I give a textual copy:

```
OpenContent License (OPL)
Version 1.0, July 14, 1998.

This document outlines  the  principles underlying the  OpenContent
(OC)  movement and may    be   redistributed provided it    remains
unaltered. For legal purposes,  this document is the license  under
which OpenContent is made available for use.

The  original   version  of this      document may  be   found   at
http://www.opencontent.org/opl.shtml

LICENSE

Terms and Conditions for Copying, Distributing, and Modifying

Items other than  copying, distributing, and modifying  the Content
with  which this license was distributed  (such as using, etc.) are
outside the scope of this license.

1.  You  may copy and distribute exact  replicas of the OpenContent
(OC)  as   you receive it,    in any  medium,   provided  that  you
conspicuously and appropriately publish on each copy an appropriate
copyright notice  and disclaimer of  warranty; keep  intact all the
notices  that refer  to this   License and to   the absence  of any
warranty; and give  any other recipients of the  OC a copy of  this
```

License along with the OC. You may at your option charge a fee for the media and/or handling involved in creating a unique copy of the OC for use off-line, you may at your option offer instructional support for the OC in exchange for a fee, or you may at your option offer warranty in exchange for a fee. You may not charge a fee for the OC itself. You may not charge a fee for the sole service of providing access to and/or use of the OC via a network (e.g. the Internet), whether it be via the world wide web, FTP, or any other method.

2. You may modify your copy or copies of the OpenContent or any portion of it, thus forming works based on the Content, and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified content to carry prominent notices stating that you changed it, the exact nature and content of the changes, and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the OC or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License, unless otherwise permitted under applicable Fair Use law.

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the OC, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the OC, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it. Exceptions are made to this requirement to release modified works free of charge under this license only in compliance with Fair Use law where applicable.

3. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to copy, distribute or modify the OC. These actions are prohibited by law if you do not accept this License. Therefore, by distributing or translating the OC, or by deriving works herefrom, you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or translating the OC.

NO WARRANTY

4. BECAUSE THE OPENCONTENT (OC) IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE OC, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE OC "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK OF USE OF THE OC IS WITH YOU. SHOULD THE OC PROVE FAULTY, INACCURATE, OR OTHERWISE UNACCEPTABLE YOU ASSUME THE COST OF ALL NECESSARY REPAIR OR CORRECTION.

5. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MIRROR AND/OR REDISTRIBUTE THE OC AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR

```
CONSEQUENTIAL DAMAGES ARISING OUT OF   THE USE OR INABILITY TO  USE
THE OC, EVEN IF SUCH HOLDER OR OTHER PARTY  HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.
```

3.How did I write this howto ?

I wrote this paper during the installation process but since I encountered several problems at the same time, my notes were disparate and sometimes I forgot to write the good command lines because I was too fed up to reboot (kernel compilations) and edit this file each time.

When the installation went wrong (it did, I'm a beginner on laptops, plip, linux nfs, portmapper, Debian), I tried several things to fix the problems. When they remained too long, my brain was focused on solving, not writing.

So, when the installation process finally succeeded, I decided to restart the process from scratch by noting everything in order to save **your** time. It was the first time I had installed Linux twice on the same computer.

Then, in order to verify the document, I again restarted from scratch with a Slink Debian distribution (the next after the Hamm one). It was the first time I had installed Linux 3 times on the same computer. I hope it was the last time I have to do that.

Conclusion: If you do what is in this paper, it should work.

4.What do you need ?

- A Personal Computer, laptop or desktop, called the target computer or simply **target**.
- A 3 1/4" inch floppy drive on the **target**.
- Two or three fresh floppies. They don't need to be formatted. You also need one more DOS formatted floppy if you have to play with `FIPS`.
- Another computer with a cdrom drive and nfs services, called source computer or simply **source**. It doesn't have to be a Linux system but just a system which can export a cdrom drive via nfs. In this HOWTO, I assume it is a Linux system.
- A cdrom distribution. I took a Debian Hamm (2.0r3) from an old Infomagic compilation. I think every distribution can be installed this way, but I'm not sure. I've read a French document, written by

Chmouel Boudjnah, saying the RedHat distribution has an easy installation process. Chmouel's document also deals with the Debian distribution, so if you read French, you can use Chmouel's document instead of this one.

- A Null−Modem DB 25 cable. Is it the same as a serial cable? You'll find the answer in the next release.
- Some time: 3 hours.
- Coffee. 1 liter.

5.The documentation

I recommend these good readings in case you can't figure out how to solve a problem. I've read them. You should too if you're curious or conscientious.

Please, do not forget to consider the LDP mirrors, listed at:
http://www.linuxdoc.org/mirrors.html

Most HOWTOs are translated in many languages. Just go to the bottom of a mirror page and follow the **`translations/`** link.

If you want to use Linux on a laptop, read the latest Laptop−HOWTO, by Werner Heuser, located at:
http://www.snafu.de/~wehe/index_li.html.

If you haven't installed any distribution yet, carefully read the Installation−HOWTO, by Eric S. Raymond, located at:
http://www.linuxdoc.org/HOWTO/Installation−HOWTO.html. And make more coffee :−)

If you need information about your cdrom drive, read the CDROM−HOWTO, by Jeff Tranter, located at:
http://www.linuxdoc.org/HOWTO/CDROM−HOWTO.html.

If you haven't installed an nfs server yet, read the NFS−HOWTO, by Nicolai Langfeldt, located at:
http://www.linuxdoc.org/HOWTO/NFS−HOWTO.html.

If you need to compile a new kernel on the **source** box, read the Kernel−HOWTO, by Brian Ward, located at:
http://www.linuxdoc.org/HOWTO/Kernel−HOWTO.html.

If you are new to PLIP, read the PLIP MINI−HOWTO, by Andrea Controzzi, located at:
http://www.linuxdoc.org/HOWTO/mini/PLIP.html.

If you plan to make your Null−Modem cable yourself, a good reading is the file PLIP.txt, by Donald Becker, located at:

`/usr/src/linux/Documentation/networking/PLIP.txt`
on your **source** linux computer, Luke.

6.Conventions

The file contents and line commands and install screen−shots are always in `typewriter` font, like this:

```
#!/bin/bash
#############################################
#### This is the great file /bin/Windows ####
#############################################

while [ "1" ]; do
        echo "I do my best because I'm the best"
        echo "Very soon, next Y2Kill (the 01/01/0000)"
        echo "A new marvelous 64 bit release !"
        echo "Please wait a little more"
        sleep 18446744073709551615 # 2^64-1
done
```

or this:

```
$ killall Windows
Terminated
```

The file content lines should never begin with white space. You'll have to remove them, if any. Sorry, I'm fed up with `C−a M−AltGr−\` (remember, I'm a French azerty writer). Tab−emacs reflex is untameable (coders who use Emacs always press the tabular key like a twitch).

Command input lines begin with a dollar `$` (the prompt), you don't have to type the dollar, just type the rest of the line; other lines are the command output, you don't have to type them either.

Because all the configuration commands are important, you'll need to use a system administrator shell, like root, on the **source** and the **target** computers.

```
$ su
Password: blabla
#
```

graham says:

Why this "#" caracter ? Because this is for root's prompt. So, in the next paragraph, say that you will use the '$' for the remainder of this. See my comment below.

The prompt will be shown as "`$`" in the remainder of this documentation "`$`", even if it should be "`#`". This is because "`#`" often means comment, so it is ambiguous. I don't like ambiguity in computer science.

7.Building your own cable

Extract from /usr/src/linux/Documentation/networking/PLIP.txt, written by Donald Becker (becker@super.org):

```
Parallel Transfer Mode 0 Cable
==============================

The cable for the first transfer mode is a standard
printer "null" cable which transfers data four bits at a time using
data bit outputs of the first port (machine T) connected to the
status bit inputs of the second port (machine R).  There are five
status inputs, and they are used as four data inputs and a clock (data
strobe) input, arranged so that the data input bits appear as contiguous
bits with standard status register implementation.

A cable that implements this protocol is available commercially as a
"Null Printer" or "Turbo Laplink" cable.  It can be constructed with
two DB-25 male connectors symmetrically connected as follows:

    STROBE output       1*
    D0->ERROR   2 - 15          15 - 2
    D1->SLCT    3 - 13          13 - 3
    D2->PAPOUT  4 - 12          12 - 4
    D3->ACK     5 - 10          10 - 5
    D4->BUSY    6 - 11          11 - 6
    D5,D6,D7 are   7*, 8*, 9*
    AUTOFD output 14*
    INIT   output 16*
    SLCTIN      17 - 17
    extra grounds are 18*,19*,20*,21*,22*,23*,24*
    GROUND      25 - 25
* Do not connect these pins on either end

If the cable you are using has a metallic shield it should be
connected to the metallic DB-25 shell at one end only.
```

I checked my cable twice. On both sides, pins are connected like this:

```
 1 -  1 Yes
 2 - 15
 3 - 13
```

```
 4 - 12
 5 - 10
 6 - 11
 7 not connected
 8 not connected
 9 not connected
10 -  5
11 -  6
12 -  4
13 -  3
14 - 14 Yes
15 -  2
16 - 16 Yes
17 - 17
18 not connected
19 not connected
20 not connected
21 not connected
22 not connected
23 not connected
25 - 25 not connected to metallic shield
```

Unlike the kernel document says, my cable has pins 1, 14, 16 connected. It doesn't matter, apparently, since plip works fine for me.

8.Network parameters

We will make a plip point−to−point network with those IP configuration:

- **target**: 192.168.0.1 netmask 255.255.255.255
- **source**: 192.168.0.2 netmask 255.255.255.255

You should not already have those names in your name space:

```
$ ping source
ping: unknown host source

$ ping target
ping: unknown host target
```

You should not already have those IP addresses in your network space:

```
$ ping  192.168.0.1
PING 192.168.0.1 (192.168.0.1): 56 data bytes
ping: sendto: Network is unreachable
ping: wrote 192.168.0.1 64 chars, ret=-1

--- 192.168.0.1 ping statistics ---
1 packets transmitted, 0 packets received, 100% packet loss
```

```
$ ping 192.168.0.2
PING 192.168.0.2 (192.168.0.2): 56 data bytes
ping: sendto: Network is unreachable
ping: wrote 192.168.0.2 64 chars, ret=-1

--- 192.168.0.2 ping statistics ---
1 packets transmitted, 0 packets received, 100% packet loss
```

If those commands don't give you errors, change the names or the addresses.

You can choose other addresses, names or netmask (netmask must be the same on both sides). In the rest of this document, I'll suppose you choose these addresses and these names.

Add a line in **`/etc/exports`**

```
#### file /etc/exports ####
...
/cdrom          source(ro) target(ro)
#### EOF ####
```

Because the portmaper tries to resolve IP addresses, add the IP addresses and names in **`/etc/hosts`**

```
#### file /etc/hosts ####
...
192.168.0.1     target
192.168.0.2     source
#### EOF ####
```

Verify you have the item **`files`** for the **`hosts`** search list in the file **`/etc/nsswitch.conf`**

```
#### file /etc/nsswitch.conf ####
...
hosts:      files nis dns
...
#### EOF ####
```

9.PLIP on the Source side

This section describes how to set up the `plip` interface in the **source** server. If you run into trouble, I suggest that you read the PLIP MINI–HOWTO.

Check that your `lp` device is not set. You should **not** have this entry:

```
$ cat /proc/devices
Character devices:
...
6 lp
...
```

If you do have it, kill the `lpd` daemon and remove the `lp` module:

```
$ /etc/rc.d/init.d/lpd.init stop
Shutting down lpd: lpd

$ rmmod lp
```

If you can't remove the `lp` module then you have to recompile the **kernel** with `lp` service as a module.

Now, the "`6 lp`" line has disappeared from the `/proc/devices` file, which is a reflection of the kernel capabilities.

You are not obliged to eliminate the lp device : the scheme may work with lp. Without guaranty (it works for me). Check it yourself.

Check that your parallel port is handled:

```
$ ls /proc/parport/
0/

$ cat /proc/parport/0/hardware
base:   0x378
irq:    7
dma:    none
modes:  SPP,ECP,ECPEPP,ECPPS2
```

If you don't have any directory under `/proc/parport/` then you have to load the `parport` and `parport_pc modules:`

```
$ insmod parport
$ insmod parport_pc
```

You should see this new entry in **`/var/log/messages`**:

```
Oct  9 20:50:47 louloutte kernel:
parport0: PC-style at 0x378 [SPP,ECP,ECPEPP,ECPPS2]

Oct  9 20:50:47 louloutte kernel:
parport0: detected irq 7;
use procfs to enable interrupt-driven operation.
```

I repeat the message "detected `irq 7`, use procfs to enable interrupt–driven operation", so:

```
$ echo 7 >  /proc/parport/0/irq
```

Check that `plip` module is loaded:

```
$ lsmod |grep plip
```

If `plip` module is not loaded, then load it:

```
$ insmod plip
```

You should see something like this in **`/var/log/messages`**

```
==> /var/log/messages <==
Oct  8 16:34:12 louloutte kernel:
NET3 PLIP version 2.3-parport gniibe@mri.co.jp

Oct  8 16:34:12 louloutte kernel:
plip0: Parallel port at 0x378, using IRQ 7
```

If you can't load the `plip` module then you have to recompile the **kernel** with `plip` service as a module.

The syslog message says the module is loaded on the `plip0` interface. Configure the `plip0` interface:

```
$ ifconfig plip0 source pointopoint target netmask 255.255.255.255 up
```

Check that everything is okay.

```
$ ifconfig plip0
plip0     Link encap:10Mbps Ethernet  HWaddr FC:FC:C0:A8:00:02
     inet addr:192.168.0.2  P-t-P:192.168.0.1  Mask:255.255.255.255
     UP POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
```

```
          RX packets:0 errors:0 dropped:0 overruns:0
          TX packets:0 errors:0 dropped:0 overruns:0
          Interrupt:7 Base address:0x378
```

Now you can `ping` locally the **source** server:

```
$ ping source
PING source (192.168.0.2): 56 data bytes
64 bytes from 192.168.0.2: icmp_seq=0 ttl=255 time=0.3 ms
64 bytes from 192.168.0.2: icmp_seq=1 ttl=255 time=0.2 ms

--- source ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.3 ms
```

Verify that the route to **target** exists:

```
$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
target          *               255.255.255.255 UH    0      0        0 plip0
```

If the route doesn't exist, add it:

```
$ route add -host  192.168.0.1  dev plip0
```

When the **target***is* configured you will be able to do a ping test:

```
$ ping target
PING 192.168.0.1 (192.168.0.1): 56 data bytes
64 bytes from 192.168.0.1: icmp_seq=0 ttl=64 time=4.5 ms
64 bytes from 192.168.0.1: icmp_seq=1 ttl=64 time=4.3 ms

--- 192.168.0.1 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 4.3/4.4/4.5 ms
```

But if you try it now you should have:

```
$ ping target
PING target (192.168.0.1): 56 data bytes

--- target ping statistics ---
5 packets transmitted, 0 packets received, 100% packet loss
```

Now, the server network is ready to work. Congratulations.

10.The cdrom nfs server side

This section describes how to mount and export via NFS a cdrom drive.

Mount the cdrom. If you encounter a problem with your cdrom drive, read the ***CDROM−HOWTO***. I assume that the cdrom device is `/dev/hdd` but it could be `/dev/sr0` or `/dev/hdb` etc. The mount point I choose is `/cdrom` but you can choose the one you want:

```
$ mount −t iso9660 /dev/hdd  /cdrom
```

You have to set up nfs services. If something goes wrong, read the ***NFS−HOWTO***

Verify your kernel supports nfs:

```
$ cat /proc/filesystems
        ext2
nodev   proc
nodev   nfs
nodev   ncpfs
nodev   devpts
        iso9660
```

Verify your portmapper can handle mountd and nfs client requests:

```
$ rpcinfo −p
program vers proto   port
100000    2   tcp    111  portmapper
100000    2   udp    111  portmapper
100005    1   udp    655  mountd
100005    1   tcp    657  mountd
100003    2   udp   2049  nfs
100003    2   tcp   2049  nfs
```

Lines with `nfs` or `portmapper` have to be there. NFS on tcp ? progress have been made !

Then, rerun `portmap`, `mountd`, `nfs`:

```
$ /etc/rc.d/init.d/portmap.init stop
```

```
Stopping INET services: portmap

$ /etc/rc.d/init.d/portmap.init start
Starting portmapper: portmap

$ /etc/rc.d/init.d/nfs stop
Shutting down NFS services: rpc.mountd rpc.nfsd

$ /etc/rc.d/init.d/nfs start
Starting NFS services: rpc.mountd rpc.nfsd
```

`/etc/rc.d/init.d/` is `/sbin/init.d/` on SuSE Linux systems. I had a dream last nigth: LSB was respected. (LDB: Linux Standard Base)

Then, try to mount it "locally" (via nfs) on the **source** box and clean the test:

```
$ mkdir /tmp/nfstest
$ mount -t nfs source:/cdrom /tmp/nfstest
$ ls  /tmp/nfstest/
README             debian/             locatedb.3          tools/
TRANS.TBL          ftp.netscape.com/   ls_lR.3             upgrade/
boot/              install/            realaudio/

$ echo great stuff !
$ umount /tmp/nfstest
$ rmdir /tmp/nfstest
```

Bad luck ? Read the ***NFS−HOWTO*** and **`/var/log/messages`**.

Check the nfs server with those commands:

```
$ rpcinfo -p
```

Run `portmap` with the `-v` flag:

```
 $ portmap -v
```

kill `portmap`, `mountd`, `nfsd` and rerun them in this order: `portmap`, `mountd`, `nfsd`. Take a rest, get more coffee. Read again the ***NFS−HOWTO*** and restart.

Now, the exported cdrom drive nfs server is ready to work. Congratulations.

11.FIPS, PARTED, splitting of hard disk partitions

This section is for anybody who has only one primary FAT partition (DOS, Windows 3.xx, NT) and wants to keep it without loosing data. Make some backups because if you burn it, I won't be responsible − you will.

Journey to PARTED

Andrew Clausen (clausen@gnu.org), GNU Parted maintainer, wrote "You might want to mention GNU Parted in the PLIP−Install Howto." That is done. Since I didn't try it I can only give you the links to parted by http and parted ftp. Help yourself. Parted seems very good.

Andrew said: "GNU Parted is a *LOT* more powerful than FIPS (think Partition Magic). For example, it can convert FAT16 <=> FAT32, change the size of the FATs, doesn't require a defragger, etc. And it supports ext2, linux−swap..."

Back to FIPS.

Download FIPS. Don't forget the mirrors ftp.**xx**.debian.org where **xx** is your country abbreviation (fr, fi, us, uk, etc.).

At the time of this writing the archive file is called **fips20.zip**.

The **fips** program is already in the debian cdrom distribution. I found it (yes, after downloading the **fips20.zip** file) in `/cdrom/debian/tools/fips15.zip` (It supposes your cdrom is mounted on **/cdrom**)

I guess you are on a Unix world but you're not compelled to do so. Go in a good working place on the **source** computer:

```
$ mkdir /tmp/fips−2.0/
$ cd /tmp/fips−2.0/
$ unzip −l /archive/fips/fips20.zip
...
$ unzip  /archive/fips/fips20.zip
...
$ ls
$ dos2unix fips.doc fips.faq  readme.1st
```

1. Read the file readme.1st
2. Read the file fips.doc
3. Read the file fips.faq

Boot your **target** computer. Be in pure DOS (quit Windows).

Read again the file **`fips.doc`** from the section "5. Before you start".

Be aware of hidden files.

Graham comment: Hidden files will not be moved by DeFrag (as far as I know), so FIPS will reclaim less space for use by Linux. Use the ATTRIB command to remove the "hidden" attribute. Some things may require that files are hidden, so it will be necessary to hide the same files again later. I think that it would also be a good idea to run scandisk at this moment. Perhaps I do not trust DOS.

Be aware to eliminate the "virtual memory" file (swap for Win*) during the fips process. In Windows 3.11 (quite up to date, no?) this swap file is configured from Program−Manager−>Control−Panel−>Enhanced (a i386 chip icon).

Click, Click−>Click on the chip and re−click on a button called "Virtual Memory".

Adjust the size to none, Click<−Click<−Click<−Click back plus Alt−F4 to close all your windows. Don't you think the Gates are too closed too?

Graham comment: I am not sure that this is a good idea. What happens if "386spart.par" is not hidden? I think that DEFRAG will move it, and Windows will complain next time it starts, then rebuild the file. Since we are clearly keeping DOS/Windows, the user will require a swap file again later. By not deleting it, the disc space is reserved for that time.

I don't what to say. I've just parroted FIPS manual because I succeeded with it.

So drink coffee, and pray.

Run `SCANDISK`:

```
C:\> SCANDISK
```

Correct every cluster. You should have no dead cluster now.

Quit `SCANDISK`

Run `DEFRAG` on C:

```
C:\> DEFRAG C:
```

All used clusters are at the beginning of the hard disk.

Quit `DEFRAG`

Make a bootable floppy disk:

```
C:\> FORMAT A: /S
```

Copy **`AUTOEXEC.BAT`** and **`CONFIG.SYS`** to the new floppy if you want to keep your local features (keyboard keys etc.). Remove the line running `SMARTDRIVE.EXE` in `AUTOEXEC.BAT`. Keep a minimal `AUTOEXEC.BAT`

On the **source** box: Copy the fips files **restorrb.exe**, **fips.exe** and **errors.txt** to this floppy disk.

```
$ mcopy errors.txt fips.exe restorrb.exe a:
```

Graham comment: Doesn't this assume that mtools are installed? If the floppy disc is mounted as type msdos, the standard "cp" command should do the job, and mtools would not be needed. I have never used mtools, but I have played with DOS files from Linux.

Yes, this assume that mtools are installed. mcopy is faster to type than mkdir, mount, cp, umount, rmdir.

Boot the **target** with the floppy. Run fips while reading the section "Using FIPS" in the file **fips.doc**

Do not forget to answer **yes** when fips ask for a rootboot saving.

When everything is ok, save your hard disk space:

```
$ cd
$ rm -rf /tmp/fips-2.0/
```

Now you have a nice free space for a new operating system.

12.Debian installation

I'm a Debian proselyte since it is my first Debian installation. Debian lovers are the same kind of people as Macintosh lovers or Linux lovers in the operating systems space. Nothing else counts. Since I'm already a Mac and Linux lover (and French too :−) it was time to fall in love.

Debian is well, clearly and internationally documented. Thanks to all those people who bred this really open distribution.

I could leave you on your own during the installation process. But since we have to interact with a shell during it, the entire process will be described in detail.

I know the description is Debian specific. I prefer to give you a complete example than nothing except a ``run a shell at the right moment and type bla bla...''. I do like concrete examples.

Simon Forget <sforget@camelot.ca> told me he could not use this howto with his Toshiba Libretto 50CT because the kernel could not recognize the pcmcia floppy drive during the installation process. I don't know why but there is a solution using plip, if you already have a dos partition on the **target** computer.

This solution is simpler and faster because no floppy drive nor plip network are necessary during the installation process. If you are interested in this solution, go directly to the section Installing from a DOS partition. I'm keeping the old installation routine because this new one needs a DOS partition that becomes superfluous when you definitely want to leave the dark side.

12.1 Preparing the two floppies

On the **source** side, mount the cdrom and go to the install directory

```
$ mount -t iso9660 /dev/hdd  /cdrom
$ cd /cdrom/debian/dists/stable/main/disks-i386/current/
```

Read the file install.html with a browser or install.txt with a cat, less or more (a dog?).

Now, record the install (alias rescue) floppy. Write a "`resc1440.bin`" label on it:

```
$ dd if=resc1440.bin of=/dev/fd0H1440
```

Write the drivers floppy. Write a "`drv1440.bin`" label on it:

```
$ dd if=drv1440.bin of=/dev/fd0H1440
```

Then make sure that the floppies are write–protected.

Now you're ready to start the real installation process.

12.2 The real installation process

Insert the resc1440.bin floppy on **target** drive. Reboot your **target** box.

```
Welcome to Debian GNU/Linux 2.x!
...
```

Read the text. The prompt is:

```
boot:
```

Press `<ENTER>`

```
Loading root.bin..........
loading linux...
....
```

A new screen:

```
Next: Select Color or Monochrome display
```

Choose yours with the arrow up/down keys and press `<ENTER>`

```
Next: Continue with the installation
```

press `<ENTER>`

```
Software in the Public Interest
        presents
 *** Debian GNU/Linux 2.1 ***
...
```

Read the text Press `<ENTER>` (You see `<Continue>` on the screen)

```
Next: Configure the Keyboard
```

press `<ENTER>` Select your country and press `<ENTER>`

```
Next : Partition a Hard Disk
```

Press `<ENTER>`

```
Select Disk Drive
usually /dev/hda
```

If you have several disks choose the good one (where you want to install Linux) and press `<ENTER>`

You are now using the program `cfdisk`. Using `cfdisk` is safe until you decide to `[Write]` the partition on disk. Leaving `cfdisk` with `[Quit]` is safe.

We are going to make just two new primary partitions, a Swap one and a big Linux one. If you want to do more partitions, leave some free space or extended partitions, you can. But read the documentation about Partitioning in the *Installation−HOWTO*, especially if you have more than 1024 cylinders and an old LILO software.

If you previously ran the `FIPS` program, you see the `hda1` partition (`DOSFAT16`). Switch to the next free partition with the up/down arrow keys.

Select `[New]` with the right/left arrow keys and press `<ENTER>`

Select `[Primary]` and press `<ENTER>`

Enter the size of your swap partition. Twice the RAM is usual if you have less than 128 megabytes of ram. If you have 2 gigabytes of RAM, it's because you don't want to swap. In that case, no swap partition is needed.

Select `[Beginning]` and press `<ENTER>`

Select `[Type]` and press `<ENTER>`. Type `82` (Linux Swap) and press `<ENTER>`.

Switch to the next free partition with the up/down arrow keys. Select `[New]` and press `<ENTER>`

Select `[Primary]` and press `<ENTER>`

Enter the size in MB (you can leave the default) and press `<ENTER>` Its type should be already Linux. If not, change to Linux (83) with `[Type]`

The partition table is defined now. Verify everything looks good. If you're not sure, read the documentation with `[Help]`. If doubts are still there, select `[Quit]` and leave the installation process. Go for a walk and restart from the beginning of this section.

I assume you are confident now.

Select `[Write]`

```
Are you sure you want to write the partition table to disk?
```

Type "`yes`" and press `<ENTER>`

Select `[Quit]` and press `<ENTER>`

```
Next: Initialize and Activate a Swap Partition
```

Press `<ENTER>`

```
Please select the partition to initialize as a swap device
```

Select `/dev/hda2` (normally already selected). Press `<ENTER>`

```
Scan for Bad Blocks?
```

select `<YES>` and press `<ENTER>`

```
Are you Sure?
```

select `<YES>` and press `<ENTER>`

```
Initializing swap partition
...
```

```
Next: Initialize a Linux Partition
```

Press <ENTER>

```
Select Partition. Please select the partition to initialize as a
Linux "ext2" file-system.
```

Select /dev/hda3 (normally already selected) and press <ENTER>.

```
Scan for Bad Blocks?
```

Select <YES> and press <ENTER>

```
Are you Sure?
```

Select <YES> and press <ENTER>

A new page full of numbers. You can take a rest because it takes some time (especially with big hard disk).

```
Next: Mount a Previously-Initialized Partition
```

Press <ENTER>

```
Please select the partition to mount
```

Select /dev/hda3 and press <ENTER>

```
Mount the /dev/hda3 device as the Root FileSystem?
```

Select <Yes< and press <ENTER>

```
Next: Install Operating System Kernel and Modules
```

Press <ENTER>

```
Please select the medium you will use to install the system
```

Select /dev/fd0 and press <ENTER>

```
Please place the Rescue Floppy in the first floppy drive
```

The floppy is already there.
Select <Continue> and press <ENTER>

```
Installing the Rescue Floppy ...
Please place the Drivers Floppy in the first floppy drive
```

Eject the Rescue Floppy and insert the Drivers Floppy, the one I you labelled `drv1440.bin` (you did it, didn't you?).
Select `<Continue>` and press `<ENTER>`

```
Installing the Drivers  Floppy ...
```

A new screen:

```
Next: Configure Device Driver Modules
```

Press `<ENTER>`

```
Select Category
```

Read the text.

```
Please select the category of modules
```

Select `net` and press `<ENTER>`
Select `plip`.

```
Module plip
```

Select "Install the module in the kernel" and press `<ENTER>`
No parameters are needed.
Select `<Ok>` and press `<ENTER>`

```
Installation succeeded
Please press ENTER when you are ready to continue.
```

Press `<ENTER>`

Select `Exit` ("Finished with these modules") and press `<ENTER>`
Then you see the same kind of screen again.
Select `Exit` ("Finished with these modules") and press `<ENTER>`

```
Next: Configure the Network
```

Choose a name, you can use a different name than `debian` or `target`.
Press `<ENTER>`

```
Is your system connect to a network?
```

Select <No> and press <ENTER>

STOP STOP STOP. You see now:

```
Next: Install the base system
```

Now we need a shell.

Press `Alt F2` and <ENTER> You are in a root shell.

12.3 An installation break: PLIP on the target side

You are in a root shell.

Verify the plip module is loaded:

```
$ lsmod
Module    Pages    Used by
plip          3          0
```

Find the exact name of the plip interface:

```
$ dmesg
...
NET3 PLIP version 2.2 gniibe@mri.co.jp
plip1: Parallel port at 0x378, using assigned IRQ 7
```

You can find the same information with

```
$ cat /proc/kmsg
...
<4>NET3 PLIP version 2.2 gniibe@mri.co.jp
<4>plip1: Parallel port at 0x378, using assigned IRQ 7
...

Ctrl-c (or ^C if you prefer)
```

Configure the plip interface:

```
$ ifconfig plip1 192.168.0.1  pointopoint  192.168.0.2 netmask 255.255.255.255 up
```

Verify everything is ok:

```
$ ifconfig plip1
plip0     Link encap:Ethernet  HWaddr FC:FC:C0:A8:00:01
     inet addr:192.168.0.1  P-t-P:192.168.0.2  Mask:255.255.255.255
     UP POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
     RX packets:0 errors:0 dropped:0 overruns:0 frame:0
     TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
     Collisions:0
     Interrupt:7 Base address:0x378
```

Verify the route to **source** exists:

```
$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0        0 lo
```

If not, like in the previous example, add the route to 192.168.0.2, the **source**:

```
$ route add -host 192.168.0.2 dev plip1
```

Now the route is installed:

```
$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.0.2     0.0.0.0         255.255.255.255 UH    0      0        0 plip1
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0        0 lo
```

Now you can perform a successful `ping` from the **source** server (`ping` is not available on the Debian install process):

```
$ ping target
PING target (192.168.0.1): 56 data bytes
64 bytes from 192.168.0.1: icmp_seq=0 ttl=64 time=14.0 ms
64 bytes from 192.168.0.1: icmp_seq=1 ttl=64 time=4.3 ms

--- target ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 4.3/9.1/14.0 ms
```

If it doesn't work, check the Null–Modem cable connection, rerun `dmesg`, `ifconfig`, `route`. Verify everything. Take a rest. Restart.

When the ping is okay, return to the normal installation process:

On the target: press `Alt F1`

12.4 Return to the normal install process

You are back to the normal installation process. You should still see:

```
Next: Install the base system
```

Press <ENTER>

```
Please select the medium you will use to install the system
```

Select `nfs` and press <ENTER>

```
Please choose the NFS server and the mount path ...
```

Type: `192.168.0.2:/cdrom`
Press <ENTER>

```
Please choose the path inside the mounted NFS filesystem
```

Type /debian (normally already there) and press <ENTER>

If no error message complains about the nfs mount then bravo. You can verify this mount by entering the shell again:

Press `Alt F2`

```
$ mount
...
192.168.0.2:/cdrom on /instmnt type nfs (rw, addr=192.168.0.2)

$ ls /instmnt/debian
README                  README.non-US          doc/
README.CD-manufacture   README.pgp@            hamm/
README.mirrors.html     TRANS.TBL              tools/
README.mirrors.txt      dists/
```

Press `Alt F1`

You're back again to the normal installation process.

Now the plip nfs cdrom connection is done. Let's go on and finish our job.

```
Please select the directory containing a file base2_0.tgz
```

Select `list` and press <ENTER>

```
Please Wait
The installation program is building a list of ...
```

A new screen:

```
Select Archive Path
Please select the directory that you will use to install the Base
System from.
```

Only one long item, already selected. Press `<ENTER>`

```
The Base System is being extracted from
/instmnt/debian/dists/.....
```

Take a second rest, you deserve it

```
Next: Configure the Base System
```

Press `<ENTER>`

```
Select Timezone
```

Select your timezone and your directory and press `<ENTER>`

```
Timezone Configuration
```

Read and press `<ENTER>`

```
Timezone Configuration.
```

An other page. Read, select `<YES>` and press `<ENTER>`

```
Next: Make Linux Bootable Directly From Hard Disk
```

Press `<ENTER>`

```
Create Master Boot Record?
```

Read
Select `<Yes>` and press `<ENTER>`

```
Make Linux the Default Boot Partition?
```

Read.
Select `<No>` and press `<ENTER>`

```
Next: Make a boot Floppy
```

Press <ENTER>

```
Change Disk
Please place a blank floppy disk in the first floppy drive.
```

Do it and press <ENTER>
The floppy is being formatted

```
Creating a filesystem on the floppy...
Copying the operating system kernel...
```

A new screen:

```
Next: Reboot the System
```

Press <ENTER>

```
Reboot the system ?
```

Remove the floppy and press <ENTER>

The system reboot. Are you still with the dark side? I guess yes, so insert the boot floppy you've just made and reboot again with `Ctrl-Alt-Del`

See the boot messages.
Read the text.

```
New password:
```

Enter a root password.

```
Re-enter new password
```

Do it. And learn it.

```
Shall I create a normal user account now? [Y/n]
```

Enter `n` and press <ENTER>. Unless you want to do it (it is safe and good, in fact).

```
Shall I install shadow passwords? [Y/n]
```

Enter `y` and press <ENTER>

```
Do you want to use a PPP connection to install
```

Enter `n` and press <ENTER>

```
Now you may choose one of several selections ...
Do you want to perform this step?
```

Enter `n` and press <ENTER>

```
I'm going to start the 'dselect' program...
```

Press <ENTER>
Select `[Q]uit` and press <ENTER>

```
You may now login as 'root' at the login: prompt...
...
debian login:
```

Enter `root`.

```
Password:
```

Enter the root password.

```
...
debian:~#
```

The system is installed and working. CONGRATULATIONS!

My job stops here. Read the install documentation of your distribution and go on with the Unix system administration job.

When you reboot your system, the plip connection won't be in good shape. But now you know what to do.

I suggest you some work:

- Configure the plip interface for the normal boot process (in the file `/etc/init.d/network`).
- Configure `/etc/fstat` on the **target** to simply mount the remote **source** cdrom via nfs.

```
source:/cdrom              /cdrom   nfs     noauto,intr 1 2
```

- Learn Lilo. Configure it for your DOS and Linux systems and install it on a floppy.
- When you master Lilo on the floppy, install it on your hard drive.
- Install and configure the X Window System.
- Have fun.
- Mail me a feedback.

13.Installing Slackware, RedHat, Caldera

This section is written by Florent SAUNIER.
sabetflo@freesurf.fr

I have performed a Slakware installation using PLIP very succesfully. I have also been able to install REDHAT and caldera Open linux using the very same method described below.

As previously specified by Gilles, a slakware installation requires everything specified in the chapter "What do you need" with time increased to 10 hours if you desire to install xfree(86) and some developpement tools.

Further on, you need to read first Gilles installation as I set up mine accordingly to his preset.

Finally, I have performed Slakware and Caldera installation on Twinhead 550C laptop, with 8Mb Ram, and 2Gbytes Hard drive.

13.1 Slakware installation

- 1− Make 2 disks (boot and root) and prepare a plip disk
- 2− Install PLIP module
- 3− Create a NFS drive to use PLIP with
- 4− Launch the Slakware (or Caldera) Installation
- 5− Drink all remainder of coffee and go to bed waiting completion of installation

Making boot,root and plip disks

To install slakware verion using ATAPI CDROM from your source you will need standard kernel image bareapm.i or bare.i. If you want to make an install from SCSI support take scsinet.i image.

Other image exists, such as net.i which sould include PLIP function but on my i486 laptop, Twinhead550C, the imagedoes not work. I have experienced that the standard image allways work on any laptop I have used.

1a− creating the boot and root disks.

From ms−dos create the boot disk by typing:

```
rawrite bareapm.i
```

then create the root disks:

```
rawrite color.gz.a  (if you are using color screen)
```

1b− Creating the PLIP disk

On a dos format disk copy the following file :

- 1− plip.o (object program for the plip tool)
- 2− parport_pc
- 3− parport

Those programs are compulsory prior a PLIP install, you will find them into the slakware installation CDROM under the directory /MODULES. Same apply to the Caldera installtion. MAke find file if you cannot find them.

1c− launching the installation.

Insert your boot disk in your Laptop and start computer. Follow the installtion procedure and wait till the system ask for the root disks. Insert it and wait for the prompt login: "Slakware login".

To start the Slakware installation you must log as root. Type then root at the prompt. Be aware that at that time of installation your keyboard is still qwerty.

2a− Install PLIP module

Install your source computer according to Gille indication. From chapter 7 to 9.

Install PLIP on the target. To install PLIP module you need to launch program from your PLIP disk. Remember you did it no !!!

Access to your floppy disk by:

```
& mount /dev/fd0h1440 /floppy
& cd /floppy
& insmod parport      'it will install your parallel port
& insmod parport_pc   'it will set it up
& insmod plip         'it will install plip module
```

If during the PLIP installation you encounter an error. It means that your paralell port has not been correctly installed. you may type again the instruction above, it does work sometimes !!, or go to the parport directory and write appopriate setting.

```
& cd /proc/parport/0
& vi IRQ            'Verify that 7 is written in it if you are using
                    'that interruption'
```

Look at chapter 8,"PLIP on the source side" to get the proper setting.

You should not bother too much as at 95% on the 5 laptop I have installed everyting went smoothly.

3a− Now create your NFS drive.

From that point I have choosed to use the creation of a NFS drive on the target computer. Then I will make an installation using the "From directory" option rather than from an nfs support.

Do not forget you are still in US keyboard. Of course you have set up your source accordingly to chapter 7, "Network parameters". You then did ping your source, but yet you cannot ping the target.

So the first step is to configure your PLIP connection: On the target type:

```
& ifconfig plip0 192.168.0.1 pointopoint 192.168.0.2 netmask 255.255.255.255 up
```

Test your Plip configuration by pinging the target from the source

```
& ping target
```

then create your nfs drive:

```
& cd /
& mkdir nfsdrive
mount −t nfs 192.168.0.2:/cdrom /nfstest
```

You have just configure a drive linked to your source directory /cdrom were you suppose to have slakware CDROM connected to, from your target directory call nfstest

Now everything is set up and you are ready to launch the Slakware installation program, by typing "setup".

note:if you do not have enough memory, below 8mb setup will not being launched then, you will have to make a swap space to allow the program to work. first Create you disk partition with your swap by typing **& fdisk** note your swap disk name eg /hda3 quit fdisk with 'w' Once its done type:

```
& mkswap −c /dev/hda3
& swapon /dev/ha3
```

Your swap is now set up. You may then launch the "setup" program.

4a− Installing Slakware

the setup programm shows various menu:

- HELP − Help file
- KEYMAP − Keyboard select (Azerty at least !!!)
- ADDSWAP − Create and select swap
- TARGET − Select target
- SOURCE − select the source
- SELECT − Select pakages to install
- CONFIGURE − Set up your linux system and make LILO
- EXIT − bye bye setup

4b− Select your keyboard, for French take −Fr−latin1.map

4c− Select your swap partition if not already configured

4d– Select the target, default is '/'

4e– Select the source. Here is the tricky stuff choose Installing from a drive type the drive so : /nfsdrive/slakware *'/slakware' because your drive must point to your slakware CDROM directory*

4f– Select packages to install

4g– Wait, Wait, Wait and Wait........

13.2 Caldera Installation

I told you that Caldera work also with that install. In fact as Caldera launch automatically is Setup program, once you arrived at the menu were you suppose to choose your source disks type ALT F2 and go to another screen then log as root and make your nfs drive as described above in point 2 to 4. Then go back to the installtion screen with ALT F1, and type the famous nfsdrive as your source directory. Caldera will than install without problem.

13.3 Troubleshooting

If you are experiencing any trouble with that installation you may report it tome and I'll try to help as best as I can.

14.Installing RedHat 6.x

This section is written by Dr. Tilmann Bubeck, bubeck@think–at–work.de

I tried hard to install a laptop through PLIP with a RedHat 6.2 but the installer from RedHat is buggy and could not be used for PLIP. I verified that it was already broken in RedHat 6.1, but works in RedHat 5.2 as desired. Patches have been sent to RedHat and will hopefully be applied to the upcoming RedHat 7.0.

So if you want to install a RedHat 6.x over PLIP you **must** follow these instructions.

The problem is the "bootnet" disk. This disk should be used for PLIP installation but has 2 serious errors:

- it does not contain the necessary driver "parport_pc"
- it does not allow you to specify important module parameters to "parport_pc" (especially "iobase" and "irq").

14.1 Using a fixed "bootnet" disk

Probably the easiest way to solve the above problem is to use a fixed "bootnet" disk. You can get a fixed "bootnet" disk for RedHat 6.2 from http://www−ti.informatik.uni−tuebingen.de/~bubeck/bootnet−plip−62.img

Use the following command to copy the downloaded image to an empty disk:

`dd if=bootnet−plip−62.img of=/dev/fd0`.

Then insert the disk and turn your computer on. When you come to the point where you must choose your network device driver please do:

- 1− Choose "Parallel Port Module for PC" and select "specify module parameters".
- 2− Enter iobase and interrupt for your parallel port. Typical values are iobase=0x378 or iobase=0x3bc. Interrupt most often is "irq=7".
- 3− Validate, that the kernel loads the module and finds the parallel port. Use ALT+F4 to switch to kernel messages and look for a line saying: "parport: ... irq: 7".
- 4− The installation will then try to access the network which is still not available. So the installer asks again for the installation method. Choose your method again.
- 5− Now choose the "plip" module without any parameters. Again verify the loading of the module in the kernel messages.
- 6− Network should be up and running, installation should proceed. If not, then check the modules parameters to "parport_pc".

14.2 Building your own "bootnet" disk

If the above bootnet disk is not suitable for your needs, I will present a small description on how to transform a standard bootnet disk for using "plip". This description is quite short and is addressed to the advanced user.

- 1− make a copy of the standard bootnet disk by using "dd".
- 2− insert this copy of standard bootnet disk and mount it with: `mount /dev/fd0 /mnt`
- 3− gunzip initial ramdisk with `cp /mnt/initrd.img /tmp/initrd.img.gz; gunzip /tmp/initrd.img.gz`
- 4− Mount initial ramdisk with `mkdir /mnt2; mount −o loop /tmp/initrd.img /mnt2`
- 5− gunzip and extract modules directory with: `cd /tmp; zcat /mnt2/modules/modules.cgz | cpio −ivd`
- 6− You should now have a directory called with your kernel version number (e.g. "2.2.14−5.0BOOT") containing a lot of modules. Check, if the modules "parport_pc.o" is included in this directory (it is unfortunately NOT included in RedHat 6.0−6.2). If you can not find it, then please build a custom kernel with the EXTRAVERSION number like your bootnet disk (in this case: "5.0BOOT") and "parport" support as module. Copy the newly created parport_pc.o module to your new modules directory: `cp .../parport_pc.o /tmp/2.2.14−5.0BOOT`.
- 7− delete a normally unused module, so that you have additional space on the disk: `rm /tmp/2.2.14−5.0BOOT/sktr.o` (driver for token ring network adapter).
- 8− rebuild modules archive with: `find /tmp/2.2.14−5.0BOOT | cpio −ov −H crc |`

`gzip > /mnt2/modules/modules.cgz`

- 9− insert a new modules dependency line into modules.dep saying that module "parport_pc" needs module "parport": `echo "parport_pc: parport" >> /mnt2/modules/modules.dep`
- 10− insert entry for parport_pc module into module−info so that the user could select the parport_pc module during installation. Append the following text to the end of /mnt2/modules/module−info:

```
parport_pc
        plip
        "Parallel Port Module for PC"
        io "Base I/O address"
        irq "IRQ level"
```

- 11− unmount and gzip initial ramdisk with: `umount /mnt2; gzip −9 /tmp/initrd.img`
- 12− copy initial ramdisk to bootnet disk and unmount: `cp /tmp/initrd.img.gz /mnt/initrd.img; umount /mnt`

By following these instructions, you get a bootnet disk containing all necessary modules and therefore allowing for a PLIP installation. Please follow the instructions above on how to do the installation with the new disk.

15.Installing from a DOS partition

You can use this section if you already have a DOS partition on your **target** computer. I think it is a faster installation method.

Read the section 5.3.1 "Installing from a DOS partition" from install.txt. This section is also available on the Debian CD in the html file ch−install−methods.html.

I just going to help you to make the first point : "1. Get the following files from your nearest Debian" repository"

The transfer is easy with tomsrtbt.

Export the **source** cdrom drive via nfs.

Read the file tomsrtbt.FAQ

Create the floppy under DOS or Linux, you have the choice.

```
Linux installation:
a) extract the .tar.gz archive
b) Be root
c) Be in the tomsrtbt-<version> directory
d) Have a blank floppy with no bad sectors
e) Do './install.s'
```

Boot the **target** with the floppy. Login as root.

```
$ insmod plip
$ ifconfig plip1 192.168.0.1 pointopoint 192.168.0.2 \
netmask 255.255.255.255 up
$ route add -host 192.168.0.2 dev plip1
$ mount -t nfs 192.168.0.2:/cdrom /cdrom
$ mkdir /c
$ mount -t msdos /dev/hda1 /c
$ mkdir /c/debian
$ cd /cdrom/dists/stable/main/disks-i386/current/
$ cp resc1440.bin drv1440.bin base2_1.tgz root.bin linux \
install.bat loadlin.exe /c/debian
```

The \ means the command line continues on the next line. The `cp` command takes some time. When finished, remove the floppy and then :

```
$ reboot
```

Boot under DOS.

```
C:\> cd debian
C:\DEBIAN> install
```

Here you go to install Debian from a DOS partition. You do not need plip nor floppy during the installation process. Everything deals with the hard drive. Read the Debian install documentation, it is a very good and clear one.

16.Install the plip interface permanently

16.1 On the source side

I use an old Linux RedHat 4.1 distribution. The location of the files can be different on other GNU/Linux distributions but the philosophy is the same (The Unix System V convention).

Create the file `/etc/rc.d/init.d/plip` with this content:

```
#!/bin/sh

###############################
# file /etc/rc.d/init.d/plip #
```

```
##############################

# See how we were called.
case "$1" in
  start)
        # Start daemons.
        /bin/echo "Starting plip interface: "
        /bin/echo "Doing /sbin/ifconfig plip0 source pointopoint target netmask 255.255.255.255 up"
        /sbin/ifconfig plip0 source pointopoint target netmask 255.255.255.255 up
        /bin/echo  "Doing /bin/ping -q -c 4 target"
        /bin/ping -q -c 4 target
        /bin/echo "Starting plip interface: done"
        ;;
  stop)
        # Stop daemons.
        /bin/echo  "Shutting down plip interface:"
        /bin/echo  "Doing /sbin/ifconfig plip0 source pointopoint target netmask 255.255.255.255 dow
        /sbin/ifconfig plip0 source pointopoint target netmask 255.255.255.255 down
        /bin/echo  "Doing /sbin/modprobe  -r plip "
        /sbin/modprobe  -r plip
        /bin/echo "Shutting down plip interface: done"
        ;;
  *)
        echo "Usage: $0 {start|stop}"
        exit 1
esac

exit 0

# === End of File ===
```

Only the ifconfig lines are strictly necessary. Perhaps you will need to add some `modprobe` commands if you don't use `kerneld` nor the `kmod` feature of new kernels 2.2.x

Create the symbolic links in the `rc*.d` directories:

```
      $ cd /etc/rc.d/rc0.d/
      $ ln -s ../init.d/plip K97plip

      $ cd /etc/rc.d/rc1.d/
      $ ln -s ../init.d/plip K92plip

      $ cd /etc/rc.d/rc3.d/
      $ ln -s ../init.d/plip S11plip


      $ cd /etc/rc.d/rc5.d/
      $ ln -s ../init.d/plip S11plip
```

You can choose other numbers. Make sure that the two−digit number after 'K' is greater than the number of every other file that stops a service depending on plip.

Make sure that the two−digit number after 'S' is less than the number of every other file that start a service depending on plip: nfs, nis, ftp, http etc.

Update the `/etc/conf.modules` file, choosing the correct IRQ number (7 is mine, yours may be different):

```
# /etc/conf.modules
...
alias parport_lowlevel parport_pc
post-install parport_pc echo 7 >  /proc/parport/0/irq
...
```

Test the plip shell:

```
$ /etc/rc.d/init.d/plip
Usage: /etc/rc.d/init.d/plip {start|stop}

$ /etc/rc.d/init.d/plip stop
Shutting down plip interface:
Doing /sbin/ifconfig plip0 source pointopoint target netmask 255.255.255.255 down
Doing /sbin/modprobe  -r plip
Shutting down plip interface: done

$ /etc/rc.d/init.d/plip start
Starting plip interface:
Doing /sbin/ifconfig plip0 source pointopoint target netmask 255.255.255.255 up
Doing /bin/ping -q -c 4 target
PING target (192.168.0.1): 56 data bytes

--- target ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 4.4/8.3/14.0 ms
Starting plip interface: done
```

Updating the start scripts is a good occasion to reboot a Unix system, to check the modifications. Do it:

```
$ init 6 # or "shutdown -t now" or "reboot"
```

16.2 On the target side

Update the file `/etc/init.d/network`:

```
#! /bin/sh
#######################
# /etc/init.d/network #
#######################

ifconfig lo 127.0.0.1
route add -net 127.0.0.0
```

```
ifconfig plip1 192.168.0.1 pointopoint 192.168.0.2 netmask 255.255.255.255 up
route add -host 192.168.0.2 dev plip1
```

That's all because the parport features are directly in the kernel.

Updating the start scripts is a good occasion to reboot a Unix system, to check the modifications. Do it:

```
$ init 6
```

17.Compatibility with a printer (lp)

I use the same parallel port for printing on a printer and networking with my laptop without big problems. It takes less than one second to switch from printing to networking. I just add a cheap manual switch between the parallel port, the printer and the laptop. The cable between the switch and the desktop computer is a fully connected one. The switch is just the one used to shared printers when no network is available. It works.

Sometimes, the kernel is not very happy because it can't finish its network job. In that case, I shutdown the plip interface and restart it when needed.

I bought a new parallel card last year but it is still in its box. I'll write an "install a second parallel port" when I'll succeed in opening the box and add that beautiful card in the last available connector I have left.

18.Need to compile a new kernel ?

Several times during the **source** server configuration we mentioned that a kernel compilation on the **source** computer could be needed.

I list the things you need to change or set. If you have never compiled a kernel, read the *Kernel−HOWTO*.

```
Loadable module support  --->
    [*] Enable loadable module support

General setup  --->
    [*] Networking support
    <M> Parallel port support
    <M>    PC-style hardware

Networking options  --->
    [*] TCP/IP networking

Network device support  --->
```

```
    [*] Network device support
    <*> Dummy net driver support
    <M> PLIP (parallel port) support
    <*> or <M>  PPP (point-to-point) support

Character devices  --->
    <M> Parallel printer support


Filesystems  --->
    [*] /proc filesystem support
    <*> or <M> ISO 9660 CDROM filesystem support
    Network File Systems  --->
        <*> or <M> NFS filesystem support
```

19.What's new

This tag is automatic so it IS the reference. (Sometimes I forget to update the version number and the date)

$Id: PLIP−Install−HOWTO.sgml,v 1.27 2000/08/11 06:30:06 gilles Exp $

v1.27, 11 August 2000

The SLIP line was not needed, SLIP is serial line oriented. The PPP line was partly wrong, we need PPP support by the kernel.
Finished to correct my bad English checked by Graham. I know it still remains many mistakes and new ones come every time I write a sentence...
Added a "compatibility with a printer" section.
Added a Null−Printer == Null−Modem comment.
Added a "build your own cable" section.
Corrected a bug in the Slackware section. Here (inverted IP adresses). Thanks to Brad Rigby.
never be obsolete paragraph in Introduction section.
known translated version demand in the Introduction section.
Mentioned the what's new section at the top of the document.
Added a vc−header in the What's new section
Added the WHAT_IS_NEW label

v1.22, 16 June 2000

Added a mention to PARTED http://www.gnu.org/software/parted as told by Andrew Clausen (clausen@gnu.org).
Added section Installing Slackware, RedHat, or Caldera, by Florent SAUNIER.
Corrected some English faults, thanks to Graham and Tomas.

v1.15, 26 November 1999

The section "What's new". It is this section.
The section "Install the plip interface permanently".
The section "List of contributors and acknowledgements".
Joined the different chapters dealing with the detailed Debian installation to form a single chapter with the actual chapters as just subsections.
Made a new section "Installing from a DOS partition". A simpler and faster method.

20.To do

- Check a kernel compilation from scratch (no last `.config` file) with only the options I gave. Add the missing ones if any.
- Make the Mandrake, SuSE (Graham? Are you there?), Turbo Linux, and Stampede detailed installation chapters. Contributions are welcome. Gilles Lamiral won't do that job unless given new computers and CDs.

  Debian (done), Red Hat (redone by Tilmann Bubeck), Slackware (done), Caldera OpenLinux (done).
- Make "The CD−ROM NFS server side" chapter become simply "the server side" chapter with subsections like "cdrom medium", "zip medium", "hard disk medium", and "ftp server" "nfs server", "samba server" "http server". Again, contributions are welcome.

21.List of contributors and acknowledgements

I thank all the people in this list of contributors or projects. If you think I forgot someone, do not hesitate to complain.

Werner Heuser <wehe@snafu.de>

He said: "yesterday I had time enough to read your document. It's great and in the next issue of my HOWTO (probably 2. December) I will make a link to it."
He did.

LDP <linux−howto@metalab.unc.edu>

The Linux Documentation Project accepted the PLIP−Install−HOWTO without any hesitation.

Simon Forget <sforget@camelot.ca>

Simon had a problem with his pcmcia floppy drive. So he led me search for a solution that is, in fact, simpler and faster (the DOS one).

Andrew Clausen <clausen@gnu.org>

Andrew suggested me to talk about parted http://www.gnu.org/software/parted

Florent SAUNIER <FlorentSAUNIER@sabetflo.freesurf.fr>

Florent wrote the section called "Installing Slackware/Caldera".

Graham Bosworth <graham@bozikins.connectfree.co.uk>

Graham corrected my bad english all over the document.

Tomas Pospisek <tpo@spin.ch>

Tomas also corrected my bad english.

Brad Rigby <brigby@rocketmail.com>

Brad corrected a bug in the Slackware section (inverted IP adresses).

Donald Becker <becker@super.org>

Donald wrote the PLIP kernel documentation /usr/src/linux/Documentation/networking/PLIP.txt

Tilmann Bubeck <bubeck@think−at−work.de>

Tilmann wrote the section called "Installing RedHat 6.x".