PLIP Install HOWTO

Table of Contents

PLIP Install HOWTO....1
Gilles Lamiral, lamiral@mail.dotcom.fr....1
1.Introduction....1
2.License....1
3.How did I wrote this howto ?....1
4.What do you need ?....1
5.The documentation....1
6.Conventions....1
7.Network parameters....2
8.PLIP on the Source side....2
9.The cdrom nfs server side....2
10.FIPS, splitting of harddisk partitions....2
11.Debian installation....2
12.Install the plip interface permanently....2
13.Need to compile a new kernel ?....2
14.What's new....2
15.Todo....2
16.List of contributors and acknowledgements....3
1.Introduction....3
2.License....3
3.How did I wrote this howto ?....5
4.What do you need ?....6
5.The documentation....6
6.Conventions....7
7.Network parameters....8
8.PLIP on the Source side....9
9.The cdrom nfs server side....13
10.FIPS, splitting of harddisk partitions....14
11.Debian installation....16
11.1 Preparing the two floppies....17
11.2 The real installation process....17
11.3 A install break: PLIP on the target side....22
11.4 Return to the normal install process....24
11.5 Installing from a DOS partition 28
12.Install the plip interface permanently....29
12.1 On the source side....29
12.2 On the target side....31
13.Need to compile a new kernel ?....31
14.What's new....32
15.Todo....33
16.List of contributors and acknowledgements....33

PLIP Install HOWTO

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

v1.15, 26 November 1999

---

*This paper describes how to install a **Gnu|Linux** distribution on a computer without Ethernet card, nor cdrom, 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 last release of the PLIP−Install−HOWTO, by Gilles Lamiral, located at:
`http://www.linuxdoc.org/HOWTO/PLIP-Install-HOWTO.html`

1.Introduction

2.License

3.How did I wrote this howto ?

4.What do you need ?

5.The documentation

6.Conventions

7.Network parameters

8.PLIP on the Source side

9.The cdrom nfs server side

10.FIPS, splitting of harddisk partitions

11.Debian installation

- 11.1 Preparing the two floppies
- 11.2 The real installation process
- 11.3 A install break: PLIP on the target side
- 11.4 Return to the normal install process
- 11.5 Installing from a DOS partition

12.Install the plip interface permanently

- 12.1 On the source side
- 12.2 On the target side

13.Need to compile a new kernel ?

14.What's new

15.Todo

16.List of contributors and acknowledgements

1.Introduction

Writing a PLIP install HOWTO seems useless nowadays since Ethernet cards are cheap: NE2000 cards cost near the same as a Null−Modem cable. This is true for desktop computers, but not for laptop computers, which the PCMCIA card costs more than 10 times as the Null−Modem cable. Moreover, there is a parallel port on every computer, not 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 LapLink cable but this word 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 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 for typos, bad English, comments, money, job writings, joy, fears, cries are welcome and recommended (not with the same eagerness, choose yours).

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, 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 wrote this howto ?

I wrote this paper during the installation process but since I encountered several problems in the same time, my notes were disparate and sometimes I forgot to write the good command lines because I was 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, portmaper, 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 came out, I decided to restart the process from scratch by noting everything in order to save **your** time. It was the first time I 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 installed Linux 3 times on the same computer. I hope it was the last time I 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 formated floppy if you have to play with `FIPS`.
- A other 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 a easy installation process. The Chmouel's document also deals with the Debian distribution, so if you read French, you can use the Chmouel's document instead of this one.
- A Null−Modem DB 25 cable.
- Some time: 3 hours.
- Coffee.

5.The documentation

I recommend those good readings when you can't figure out 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

Mostly every HOWTO are translated in many languages. Just go to a mirror bottom page and follow the **`translations/`** link.

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

If you haven't install any distribution yet, read carefully 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 install a 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 with plip, read the PLIP MINI−HOWTO, by Andrea Controzzi, located at:
http://www.linuxdoc.org/HOWTO/mini/PLIP.html.

If you plan to build your Null−Modem cable on your own, 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 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 untamable.

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 neither.

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
#
```

The prompt will remain "`$`" in 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.Network parameters

We will make a plip pointopoint 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 a error, 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 those addresses and those names.

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

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

Because the portmaper do try 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 ####
```

8.PLIP on the Source side

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

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

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

If you 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 is disappeared from the `/proc/devices` file, which is a reflection of the kernel possibilities.

You are not obliged to eliminate the lp device because it can work with it but it isn't sure (it works for me). Check it yourself.

Check if your parallel port is handle:

```
$ 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 if `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 if everything is ok.

```
$ 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***will* be configurated you can 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.

9.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 suppose 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 portmaper 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
```

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. Read again the ***NFS-HOWTO*** and restart.

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

10.FIPS, splitting of harddisk partitions

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

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 download 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.

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 configurated from Program−Manager−>Control−Panel−>Extent (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 ?

Go to sleep.

Run `SCANDISK`:

```
C:\> SCANDISK
```

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

Quit `SCANDISK`

Run `DEFRAG` on C:

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

All clusters are at the beginning of the harddisk.

Quit `DEFRAG`

Make a bootable floppy disk:

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

Copy **`AUTOEXEC.BAT`** and **`CONFIG.SYS`** on 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:
```

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.

11.Debian installation

I'm not 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 Lin lover (and French too :–) it was time to fall in love.

Debian is well, clearly and internationally documented. Thanks to all those guys 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 details.

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 with 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 by this solution, go directly to the section Installing from a DOS partition. I keep the old one installation process because this new one needs a DOS partition that becomes unnecessary when you definitively want to leave the dark side.

11.1 Preparing the two floppies

On the **source** side, mount the cdrom and go in 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
```

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

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

Lock the writing on those floppies.

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

11.2 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
/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*.

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 mega bytes of ram. If you have 2 giga bytes 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 is 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 your 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 to initialize as a Linux "ext2"
file-system.
```

Select /dev/hda3 (normals 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.

11.3 A install 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
...

Cntrl-c
```

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 ok, return to the normal installation process:

On the target: Press `Alt F1`

11.4 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 formated

```
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.

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

Enter n and press <ENTER>

```
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 now 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 mount simply 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.

11.5 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. Log in 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.

12.Install the plip interface permanently

12.1 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. Be aware to place the K??plip files after the ones that shutdown services depending on plip. Be aware to place the S??plip files before the ones that start services depending on plip, nfs, nis, ftp, http etc.

Update the `/etc/conf.modules` file:

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

Choose the good irq number (7 is mine, not yours).

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 accasion to reboot a Unix system, to check the modifications. Do it:

```
$ init 6
```

12.2 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
```

13.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 compile 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
    < > or <M> SLIP (serial line) 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
```

14.What's new

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".
Made the different chapters dealing with the detailed Debian installation be a single chapter with the actual chapters as just subsections.
Made a new section "Installing from a DOS partition". A simpler and faster method.

15.Todo

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

16.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."

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 made me search a solution that is, in fact, simpler and faster.