Installation procedures for the *iX*-Terminal

Harald Milz, `hm@ix.de`

This document contains installation information for the *iX*-Terminal described in *iX* 8/95, pp. 36-42. It is maintained by *Harald Milz* (`http://www.ix.de/ix/editors/hm.html`) (*hm@ix.de* (`mailto:hm@ix.de`)). Last change: Oct 9, 1995

Contents

1 Copyright

2 What is the *iX*-Terminal?

The *iX*-Terminal is a diskless Linux machine, equipped with a SMC8013-compatible network card, a graphics adapter and at least one serial port (or a bus mouse). The basic setup is described in *iX* 8/95, pp. 36-42. This article, however, is written in German, and will not be reproduced here at full extent.

The basic function of *Gero Kuhlmann's* `(mailto:gero@gkminix.han.de)` package `netboot-nfs` is described in the file `netboot/README`, contained in *netboot-nfs.tar.gz* `(ftp://ftp.ix.de/pub/ix/Linux/iX-Terminal/netboot-nfs.tar.gz)` which you can find at the same place you got this INSTALL from. In short terms, a special boot ROM on the NIC gets its IP address from a `bootpd` server. Then the ROM code fetches the Linux kernel via TFTP. Gero Kuhlmann modified the kernel so that it is able to mount its root (i.e. complete) file system via NFS.

File access, though, is kept at a minimum after the initial boot of the *iX*-Terminal. Performance should thus not be a problem and can be further enhanced by using an X11 font server instead of the supplied fonts.

8 MBytes of RAM are sufficient for a system with 1024 x 768 pixel resolution and 256 colors to avoid swapping (see section "Swapping"). If you want to display 1280 x 1024 pixels with 24 bit color resolution, you should run the terminal with 16 MByte or more.

3 Chosing the right PC mainboard

(This section is yet uncomplete, just a short-notice statement concerning BIOSes and the used boot ROM.) Due to a bug in the original boot ROM code by *Jamie Honan* `(mailto:jhonan@kralizec.zeta.org.au)`, the ROM code gets executed immediately after the BIOS calls the initialization routines. Some odd BIOSes don't like this with the effect of slowing down the CPU by about 90 percent. I encountered this behaviour when using an ASUS SP3 board (the old one) and an ALI M1439/M1445 board. The ASUS PVI-486SP3/33 works just fine with an Am486DX-4/100 in my sample setup I'm using daily.

This bug will be fixed in a future version of the boot ROMs.

4 Installation

Please: Before actually doing anything, do yourself (and me :-) a favour and read this instruction completely.

The distribution mainly consists of the directory tree *TEMPLATE* (ftp://ftp.ix.de/pub/ix/Linux/iX-Terminal/TEMPLATE.etc-dev.tar.gz). You should provide for approximately 23 MByte harddisk space for the initial installation and about 250 KBytes for each *iX*-Terminal. At an appropriate place (say, `/tftpboot/iX-Terminal`), unpack the tar tape with the command

```
$ tar xvf <tape device>
```

This will create the `TEMPLATE` tree, the installation script *newhost* (ftp://ftp.ix.de/pub/ix/Linux/iX-Terminal/newhost) plus some sources for the supplied Linux kernel *linux-1.2.12-netboot.tar.gz* (ftp://ftp.ix.de/pub/ix/Linux/iX-Terminal/linux-1.2.12-netboot.tar.gz) and *gzip-1.2.4* (ftp://ftp.ix.de/pub/ix/Linux/iX-Terminal/gzip-1.2.4.tar.Z) (which you can immediately delete if you have `gzip` already installed; if you haven't, see section "Installing gzip").

After unpacking the stuff, export the respective directory so that root access will be allowed from the new *iX*-Terminal(s). Otherwise, root processes (like `init` or the X server) will not be able to write to files. Don't forget to set up `bootpd` and `tftp` on the NFS server (consult your local man pages).

Harddisk usage can be reduced by using a font server (which is strongly encouraged). Simply delete the `TEMPLATE/usr/X11R6/lib/X11/fonts/` directory tree (saves 5.5 MBytes). You can further save space if you delete all X servers that are not needed from `TEMPLATE/usr/X11R6/bin` (saves another 10 MBytes). You have to edit `TEMPLATE/etc/XF86Config` to use a font server (Section Files, see example).

4.1 Sample `newhost` session

After unpacking everything in a suitable place, you should call `newhost` for every *iX*-Terminal you intend to set up. **ATTENTION**: Some Unixware installations don't have `mkdirhier`. This shell script belongs to X11R5 and X11R6 and *should* be there. It is capable of creating a directory and its parent directories if needed, e.g. `mkdirhier /usr/local/foo/bar`. If you don't have `mkdirhier`, you might use GNU `mkdir` and replace the `mkdirhier` calls in `newhost` by `mkdir -p`.

`newhost` asks a lot of questions and sets up the root directory of the new device as completely as possible. You have to manually edit some files after running `newhost` (e.g. `XF86Config`) because I don't see a way to automagically set them up. But see the sample session first (a colon means a prompt) :

```
$ ./newhost

This script generates a file tree for a new "iX-Terminal". It will ask you
following items:
- hostname, domain name and several IP addresses in internet dotted notation
  (xxx.xxx.xxx.xxx)
- type of graphics adapter
```

```
- type of mouse
- keyboard type
so you might want to keep your documentation handy...

hostname, e.g. nathan: charlie
domain name, e.g. ix.de: ix.de
IP address of the new terminal, e.g. 192.168.10.10: 192.54.43.145
local net address, e.g. 192.168.10.32: 192.54.43.0
local netmask, e.g. 255.255.255.224: 255.255.255.0
local broadcast address, e.g. 192.168.10.255: 192.54.43.255
your default gateway, e.g. 192.160.10.1: 192.54.43.1
your nameserver, e.g. 192.168.10.34: 192.54.43.34
name of the syslog logging host: nathan
name of the XDMCP host: opus40

Now we must determine your graphics adapter.

Please enter    if you have a

8514            8514-compatible adapter
AGX             AGX-based board, e.g. a Hercules Graphite HG-210
Mach8           ATI Mach8
Mach32          ATI Mach32
Mach64          ATI Mach64
P9000           Weitek P9000-based board
S3              S3-based board
SVGA            any SVGA board, e.g. ET4000
W32             ET4000 W32 board

type of graphics adapter: Mach32

It's time to determine your mouse device. You have the choice of the following:

logibm          Logitech bus mouse
psaux           PS/2 bus mouse
inportbm        Microsoft bus mouse
atibm           ATI XL bus mouse
ttyS0           serial mouse connected to the first serial port
ttyS1           serial mouse connected to the second serial port

type of mouse: ttyS0

We are going to set up the right keyboard mapping. Following
Keyboard types are supported:

azerty.map                es.map                    pl.map
be-latin1.map             fi-latin1.map             ru.map
cf.map                    fi.map                    ru1.map
```

```
de-latin1-nodeadkeys.map  fr-latin1.map            ru2.map
de-latin1.map             fr.map                   sf-latin1.map
de.map                    gr-lat1.map              sf.map
defkeymap.map             gr-pc.map                sg-latin1-lk450.map
defkeymap_V1.0.map        gr.map                   sg-latin1.map
dk-latin1.map             hebrew.map               sg.map
dk.map                    it-ibm.map               slovene.map
dvorak.map                it.map                   uk.map
emacs.map                 no-latin1.map            us.map
emacs2.map                no.map
if in doubt, see the docs in TEMPLATE/usr/lib/kbd/keytables
for a German keyboard, type gr-lat1.map is recommended.
your keyboard type: gr-lat1.map

These are your present settings:
hostname                        charlie
domain                          ix.de
IP address                      192.54.43.145
your graphics adapter           Mach32
your mouse type                 ttyS0
your local network address      192.54.43.0
your netmask                    255.255.255.0
your broadcast address          192.54.43.255
your default gateway            192.54.43.1
your nameserver                 192.54.43.34
the syslog logging host         nathan
the XDMCP host                  opus40
your keyboard type              gr-lat1.map

Everything correct? (Y/N) y
setting up directory tree for charlie
copying all in /etc
hardlinking all the rest
creating some symlinks
generating NETCONF HOSTNAME syslog.conf resolv.conf hosts

Now you should cd to charlie/etc and edit
XF86Config              (mandatory)
rc.d/rc.local           (if you like)
inittab                 (if you like)

When booting your new iX-Terminal for the first time, the kernel will ask
for a suitable video mode. Press RETURN and choose one of the modes
shown. To permanently set this video mode upon boot, you can create a new
bootImage with the "rdev" and "mknbi" commands.

Don't forget to set up the file /etc/bootptab on your BOOTP host. You
should consult the documentation of the respective host, i.e.  "man
```

```
bootpd". If BOOTP host and TFTP host are identical, a minimum bootptab
entry might look like:

charlie:ht=ethernet:ip=192.54.43.145:ha=0000c0123456: \
       bf=/usr/local/iX-Terminal/charlie/bootImage

Have fun!
```

Finally, you should edit `XF86Config` and set up a suitable `bootptab` entry. Your new *iX*-Terminal should boot nicely if you already set up your network interface card (see next subsection). **Hint**: if you have a number of *iX*-Terminals with the same graphics cards and the same monitors attached to each of these gadgets, you may consider setting up a single `XF86Config` and hard linking it to the respective locations. This saves a reasonable amount of time.

4.2 Installing the boot ROM on the network interface card

If you purchased the tape from Emedia Co., you should have received a boot EPROM. This ROM is compatible with newer SMC8013 cards (those without jumpers) and SMC Elite16 cards. The Elite16 Ultra is not compatible! Older SMC8013 cards with jumpers and EEPROM may be usable too but this has not been tested. Maybe you have to set up the NIC with a DOS-tool or similar. Very old WD8013 cards which only have jumpers are not usable for this task.

If you received the software package *via FTP* (`ftp://ftp.ix.de/pub/ix/Linux/iX-Terminal`), you have some more work to do. Unpack the archive `netboot-nfs.tar.gz` and grab the file `netboot/bootrom/rom.bin`. Now you need an EPROM burner and an erased 32K EPROM (27C256 or similar). Store the ROM image in the EPROM and you're done.

When inserting the EPROM in the empty socket on the NIC, make sure the notch of the EPROM package is in line with the corresponding mark on the printed circuit board or the socket. Try to avoid touching the EPROM pins with your fingers if possible.

To boot from a floppy disk (if you have a diskette drive installed), you can copy the file `netboot/bootrom/fimage.bin` onto a disk:

```
$ dd if=fimage.bin of=<diskette device>
```

Insert the floppy into the drive and switch on the *iX*-Terminal to boot.

4.3 Making a new bootkernel

Normally, the supplied kernel should meet all your requirements. If you need to make a new kernel, unpack `linux-1.2.12-netboot.tar.gz` at a suitable place (on a Linux machine, that is :-) and `cd linux`. There is a `README` which describes how to basically compile a new kernel. If you are unsure, ask someone who knows how to do it. The commands

```
$ make config
```

```
$ make dep
$ make zImage
```

leave the compressed kernel image in **arch/i386/boot/zImage**. There are many questions to answer when running **make config**, but as a Linux newbie, you don't want to build a new kernel anyway. The file **TEMPLATE/.config** shows you the configuration settings of the supplied kernel.

Now copy **zImage** to the root directory of your *iX*-Terminal and follow the instructions in **netboot/README**. Basically, you create a new **bootImage** with:

```
$ mknbi -d rom -i rom -k zImage -o bootImage
```

You can perform these steps on the **root** prompt of your *iX*-Terminal.

If you want to get rid of the console mode prompt at each boot, you can write down the number of your favourite mode and run

```
$ rdev -v zImage <your favourite mode>
```

before calling **mknbi**.

4.4 Unpacking XFree86-3.1.1

The XFree86-3.1.1 distribution is stored in a second tar archive on the tape (25 MBytes). To access it, you have to fast forward your tape first, using the non-rewinding device:

```
$ mt -f <non-rewinding tape device> fsf 1
$ tar xvf <non-rewinding tape device>
```

Finally, rewind your tape. XFree86-3.1.1 will end up in **XFree86-3.1.1/** in the current directory.

4.5 Installing gzip

If you haven't already installed **gzip** on your boot host, here's how to do it: unpack **gzip-1.2.4.tar.Z** in a suitable place:

```
$ uncompress gzip-1.2.4.tar.Z
$ tar xvf gzip-1.2.4.tar
$ cd gzip-1.2.4
```

In this directory, read the files **README** and **INSTALL** to get an idea how to compile and install **gzip**. Be assured, it's fairly easy.

5 What if I have got the stuff via FTP?

In this case you have to grab the respective selection of files yourself from a Linux machine in the neighborhood (see `ls-laRF.gz`). But beware: `init` must be a statically linked program at least for the first boot process because the symbolic links for the shared libraries aren't created yet. Afterwards, you can replace `init` by a dynamically linked version.

Don't forget to check where your console `getty` expects `utmp`. It is possible that the necessary directory is not listed in `ls-laRF`. You have to create it manually.

6 Linux Device Special Files

You can hardly create the device special files needed by the small Linux system on the NFS host. If you compare the definitions in `/usr/include/sys/stat.h` (or a file `#include`'d by this file) on several systems, you will find different major and minor device numbers for some systems. POSIX compatible systems should all have the same numbering scheme, but special files which are "proprietary" for the individual Unix flavor will not fit into any scheme.

Consequently, I added a small patch in the Linux kernel source file `init/main.c` which creates `/dev/tty`, `/dev/tty0` and `/dev/tty1` just before spawning `init`. All other device special files are created by the script `/sbin/makedevs`, called by `init` through an `inittab` entry. `/sbin/makedevs` is only called if `/dev/tty2` doesn't exist, i.e. at the very first boot and then never again. It takes about a minute, so please be patient. At least it tells you what it's doing.

7 Console Login

If you boot the *iX*-terminal, you will see a login prompt on the console before the X server takes over. Yes, there is a getty running on `/dev/tty0` so that you can login as `root` (set up a new password immediately!) or as `x11` which is a non-password non-privileged login. As `x11`, you can e.g. `telnet` or `ftp` somewhere else. The `root` login is suitable to locally administer something. There are some preformatted man pages for most of the available commands, too.

To access the console login prompt after the X server has taken over, press `CTRL-ALT-F1`. To get back, `CTRL-ALT-F2` (or `ALT-F2`) is fine. You can do this as often as you like.

If you cannot swap any longer, the X server is probably buggy; please report this a.s.a.p., with a detailed description which graphics card and which X server you use, the way the screen looked like in the end and a copy of the file `var/run/xsession-errors` under the root directory of the respective *iX*-Terminal.

8 Swapping

If you have enough RAM, there is no need for swapping via NFS which is dead slow. However if you don't want (or cannot) add more RAM or if you temporarily need some more memory, some swap space may be valuable and preferred over dumb error messages. Following commands give you 4 MBytes (i.e. 4096 KBytes) of swap space:

On the NFS server (boot host), cd to <hostname>/tmp and

```
$ dd if=/dev/zero of=swapfile bs=1k count=4096
```

On the `iX`-Terminal, login as `root`, cd to `/tmp` and (as described in the `mkswap` man page):

```
$ mkswap swapfile 4096
$ sync
$ swapon swapfile
```

You should add

```
/sbin/swapon /tmp/swapfile
```

to your `/etc/rc.d/rc.S` in order to have the swap space available after every reboot.

If you feel you have to get rid of the swap file (which uses considerable harddisk space on the boot host!) simply call

```
$ swapoff swapfile
$ rm -f swapfile
```

on the *iX*-Terminal `root` prompt and delete the `swapon` call in `/etc/rc.d/rc.S`.

WARNING: On slow machines, or if either the NFS server or the *iX*-Terminal is very busy, you may encounter strange crashes. In this case, forget about swapping over NFS. The UDP protocol offers no safe packet delivery. To make network swapping safe and reliable, a TCP-based protocol is necessary. Unfortunately, such a protocol hasn't even been negotiated yet.

9 XFree86 Configuration

Please consult the XFree86 README files to get an idea how to set up XFree86. The file `X311doc.tgz` available on all XFree86 FTP servers (and on the distribution tape) contains thorough information. See the file `VideoModes.doc` contained in this archive. The package on the distribution tape is a mirror of the directory *ftp://sunsite.unc.edu/pub/Linux/X11/XFree86-3.1.1* (`ftp://sunsite.unc.edu/pub/Linux/X11/XFree86-3.1.1`) as of end-June 1995. (Remark: XFree86-3.1.2 was announced officially on July 31st, 1995. Due to timing considerations, it didn't make it into this package. Plus, I haven't tested it. Users who desperately want to run XFree86-3.1.2 can find it on *ftp://sunsite.unc.edu/pub/Linux/X11/XFree86-3.1.2* (`ftp://sunsite.unc.edu/pub/Linux/X11/XFree86-3.1.2`) and associated mirrors.)

10 Questions?

If you have any questions or suggestions, direct them to:


```
Harald Milz
Editorial Board iX Multiuser Multitasking Magazine
Helstorfer Str. 7
D-30625 Hannover
phone: +49 511 5352-377
fax:   +49 511 5352-361
email: hm@ix.de
```