Configuration HOWTO

By Guido Gonzato, <guido‘‘at’’ibogeo.df.unibo.it> v1.3.7, 6 September 1999

This HOWTO aims at making the fine–tuning of your newly installed Linux box quicker and easier. Here you will find a set of configurations for the most common applications and services.

Contents

1 Introduction **3**

1.1 Why This HOWTO . . . 3

1.2 What We Will Be Configuring . . . 3

2 General System Setup **3**

2.1 A Few Words About Security . . . 3

2.2 Start the Logbook! . . . 4

2.3 Keyboard . . . 4

2.4 Boot and Rescue Floppy . . . 4

2.5 Kernel Matters . . . 4

2.6 Sendmail Lock . . . 5

2.7 Hard Disk Performance . . . 6

2.8 Parallel Port Zip Drive . . . 6

2.9 Device Drivers . . . 6

2.10 Sound Card . . . 7

2.11 Login Messages . . . 7

2.12 Hostname . . . 7

2.13 Mouse . . . 7

2.14 Mount Points . . . 8

2.15 lilo(8) and LOADLIN.EXE . . . 8

2.15.1 Security Tip . . . 9

2.16 Mail Capabilities . . . 10

2.17 Printer Configuration . . . 10

2.18 SVGATextMode . . . 11

3 Common Administration Tasks **12**

3.1 Network configuration . . . 12

3.2 Sharing the Internet . . . 13

3.3 Restricting Network Access . . . 13

3.4 NFS Exports . . . 14

3.5 Name Server 14

4 Software Configuration **14**

4.1 bash(1) 14
4.2 ls(1) 20
4.3 less(1) 20
4.4 emacs(1) 21
4.5 joe(1) 22
4.6 jed(1) 22
4.7 pine(1) 23
4.8 minicom(1) 23
4.9 efax(1) 23
4.10 Ghostscript 24
4.11 TeX and Friends 24
4.12 Avoid PPProblems! 25
4.12.1 A Quick Start with eznet 25
4.12.2 A Quick Start with wvdial 25
4.13 POP Client 26
4.14 X Window System (XFree86) 26
4.14.1 Setting Up the X Server 26
4.14.2 Keypad 27
4.14.3 Graphical Login with xdm 27
4.14.4 Window Manager 28
4.14.5 Defaults for X11 Apps 28
4.15 Users' Configurations 29
4.16 Making .rpms 29
4.17 Upgrading 29

5 Configuration Software **29**

6 The End **29**

6.1 Copyright 29
6.2 Feedback 30
6.3 Disclaimer 30

1 Introduction

1.1 Why This HOWTO

I have installed Linux on many PCs and noted that current distributions are terrific but, annoyingly, lack some basic configuration. Most applications will work out of the box, but others won't. Moreover, I have noted that the same questions crop up on c.o.l.setup over and over again. To try and remedy this situation, and for my own convenience, I wrote a to–do list that eventually became this HOWTO. Here you will find configuration tips and examples for the most common applications, programs, and services, which should save you a fair amount of time and work.

A few of the examples outlined in this HOWTO depend somehow on the distribution. Currently, I only have access to Red Hat 5.2 machines, kernel 2.0.36; so don't take any of my tips as gospel if you have other distributions. It shouldn't be hard to adapt my tips to other distros, though.

This HOWTO can't, and is not meant to, replace others. Reading docs and HOWTOs always pays, so you're strongly advised to do so if you want to know more. Also, no spoon–feeding here: if you realise you don't understand something, please refer to the relevant HOWTO. Let me remind you that the right place for seeking help with Linux configuration is Usenet, i.e. `<news:comp.os.linux.setup>`. *Please*, don't seek help from me because I'm quite overloaded.

The official place for this document, including some translations, is available at `<http://metalab.unc.edu/mdw/HOWTO>`. Here you will also find all other HOWTOs I refer to. The very latest version is available at `<http://ibogeo.df.unibo.it/guido/Configuration>`.

1.2 What We Will Be Configuring

There can be endless hardware configurations for a PC, but in my experience one is quite common: a PC fitted with a large HD split into three partitions (one for DOS/Windows, one for Linux, one for the swap), sound card, modem, CD–ROM drive, printer, mouse. A parallel port Zip Drive is also very common. This machine is possibly part of a mixed Windows-Linux network, where it acts as server.

This is the hardware I'll assume you want to configure, but it's easy to adapt the following tips to different configurations. It's implicitly assumed that you'll be root when editing/fixing/hacking.

And now, lads, sleeves up.

2 General System Setup

2.1 A Few Words About Security

Even before your system boots, you should decide what level of security you want to implement. To begin with, don't connect your machine to the network until you have decided what to do.

Security is a vast subject that goes beyond the scope of this HOWTO; two good starting points are the Linux Security Administrator's Guide at `<http://www.seifried.org/lasg>` and the Linux Security Guide at `<http://nic.com/~dave/Security>`. You should consider at least the following steps: using shadow passwords (Shadow Password HOWTO), restricting network access to the machine (Section 3.3 (Restricting Network Access)), using the Secure Shell (`<http://www.cs.hut.fi/ssh/>`) or the Secure Remote Password (`<http://srp.stanford.edu/srp/>`). Good luck.

2.2 Start the Logbook!

To keep your installation in shape, it's *essential* that you know exactly what happened to your machine, which packages you installed that day, what you removed or modified, and so on. So, the first thing you'll do before you tamper with your machine is start a "logbook". Therein you'll take note of *every* move you make as root; in my own logbook I also keep a section where I list all modified system files, additional .rpms, and .tar.gz I installed. Optimally, backtracking your moves you should be able to re-obtain a pristine installation.

Make a backup copy of the system files you touch. Better still, use RCS; you'll be able to backtrack all changes. Never work as root without logging your moves!

2.3 Keyboard

If you missed this step during installation or have changed your keyboard, you'll have to:

- find a suitable key table starting from `/usr/lib/kbd/keymaps/`; for example, `it.map` selects the Italian keyboard;
- edit the file `/etc/sysconfig/keyboard` so as it reads: `KEYTABLE="it"`;
- to set up the keyboard repeat rate and delay time, add this line to `/etc/rc.d/rc.sysinit` (Red Hat) or `/etc/rc.d/rc.boot` (Caldera) or `/etc/rc.d/boot` (S.u.S.E.):

```
/sbin/kbdrate -s -r 16 -d 500  # or whatever you like
```

To load the key table, issue `/etc/rc.d/init.d/keytablestart`. Other special keys will be considered in the following sections.

To enable NumLock on by default, add these lines to `/etc/rc.d/rc.sysinit`:

```
for tty in /dev/tty[1-9]*; do
  setleds -D +num < $tty
done
```

2.4 Boot and Rescue Floppy

Make a couple of boot floppies for your newly installed system. Your distribution may include a command for creating such a floppy; if not, these commands will do:

```
#~ dd if=/boot/vmlinuz-2.0.36-0.7 of=/dev/fd0  # use your kernel image
#~ rdev /dev/fd0 /dev/hda2                     # your Linux partition
```

Also, have at least a couple of rescue disks ready. There's a wide choice of rescue disks at `<ftp://metalab.unc.edu/pub/Linux/system/recovery>`; if you don't know which one to choose, I suggest you try out Tomsbtrt.

2.5 Kernel Matters

IMHO, the first thing to do next is build a kernel that best suits your system. It's very simple to do but, in any case, refer to the `README` file in `/usr/src/linux/` or the Kernel HOWTO. Hints:

- consider carefully your needs. Choosing a kernel configuration, applying the patches, and compiling it once and for all is more productive than reconfiguring and recompiling each month; this is especially true if your Linux box is a server. Don't forget to include support for all the hardware you might likely add in the future (e.g. SCSI, Zip, network cards, etc); using modules is usually the best choice;
- if your PC is based on a Cyrix CPU that is not properly recognised as such by the kernel, apply appropriate patches to improve its performance. Information on <http://www.linuxhq.com/patch/20-p0591.html>;
- notebook users will want to improve their LCD screen legibility. I once suggested that a kernel patch should be applied, but `SVGATextMode` (see Section 2.18 (SVGATextMode)) is probably the best solution. Anyway, the Big Cursor kernel patch is on <http://www.linuxhq.com/patch/20-p0239.html>;
- again for notebook users, if you plan to use a PCMCIA modem/fax *don't* compile serial support as a module; compile it in the kernel, otherwise your PCMCIA modem won't work;
- if you use PCMCIA cards, remember to compile those modules as well; also, don't forget to edit `/etc/pcmcia/network.opts` if you use a PCMCIA network card;
- if you get those pesky messages from `modprobe` indicating that some modules can't be located, that means that your `/etc/conf.modules` needs mending. For instance, if you don't use the ipx and appletalk modules, add these lines:

```
alias net-pf-4 off
alias net-pf-5 off
```

- to save time the next time you reconfigure and recompile the kernel, it's a good idea to save your configuration on a file and keep it in a safe place. Beware: if you upgrade your kernel this config file is bound to be useless, because it doesn't contain the features of the new kernel.

2.6 Sendmail Lock

On some systems, `sendmail` hangs the machine for a couple of minutes at boot time. There are two cases: 1) the machine is not directly connected to the Internet, 2) the machine is connected, and has a permanent IP address.

Quick fix for the first case: make sure your `/etc/hosts` contains a line that reads

```
127.0.0.1   localhost
```

For the second case: the lock is caused by `/etc/hosts` containing a line like

```
127.0.0.1   localhost   your_host_name
```

which you will split this way:

```
127.0.0.1    localhost
w.x.y.z      your_host_name
```

See also Section 2.12 (Hostname).

2.7 Hard Disk Performance

Your (E)IDE hard disk's performance can be greatly enhanced by *carefully* using `hdparm`(8). If your Linux distribution doesn't include it, you'll find on <ftp://metalab.unc.edu/pub/Linux/system/hardware>; look for a file called `hdparm-X.Y.tar.gz`.

Since many details depend on your hard disk and HD controller, I can't give you a general recipe. You risk to toast your filesystem, so *read the man page carefully* before using some of the options. At its simplest, you could add the following line to `/etc/rc.d/rc.sysinit`:

```
/sbin/hdparm -c1 /dev/hda  # first IDE drive assumed
```

which enables (E)IDE 32-bit I/O support. As for the '`-m`' option, this is what `hdparm` author Mark Lord emailed me:

> (...) if your system uses components from the past couple of years [< 1997], it will be fine. Older than that, there *may* be a problem (unlikely). The really buggy chips were the CMD0646 and RZ1000 chips, used *extensively* on 486 and (early) 586 motherboards about 2-3 years ago.

For recent machines, these settings should work fine:

```
/sbin/hdparm -c1 -A1 -m16 -d1 /dev/hda
```

2.8 Parallel Port Zip Drive

To use the parallel port version of the Zip drive you can use the default driver that comes with recent (2.x.x) kernels. During kernel configuration, make sure that SCSI support and SCSI disk support are enabled. Remember, there can be conflicts between the printer and the Zip drive on the same parallel port, so you will want to use kernel modules. There's an alternative `ppa` driver at <http://www.torque.net/~campbell>.

Zip disks are sold preformatted on partition `/dev/sda4`. To enable the Zip, append this to `/etc/rc.d/rc.sysinit`:

```
# Enable the Zip drive
/sbin/modprobe ppa
```

Zip disks can be mounted via `/etc/fstab` as shown below, or via Mtools adding this line to your `/etc/mtools.conf`:

```
drive z: file="/dev/sda4" exclusive
```

besides, the command `mzip` allows you to eject, query the status, write and password protect Zip disks; `man mzip` for details. The Mtools home page is at <http://linux.wauug.org/pub/knaff/mtools>.

2.9 Device Drivers

Devices in `/dev` (or better, links to the actual device drivers) may be missing. Check what devices your mouse, modem, and CD–ROM drive correspond to, then do what follows:

```
~# cd /dev
/dev# ln -s ttyS0 mouse; ln -s ttyS1 modem; ln -s hdb cdrom; ln -s sda4 zip
```

In most notebooks the mouse device is `/dev/psaux`: take this into account when configuring X11.

If you want, do `chmod 666` to these devices to make them fully accessible by every user.

2.10 Sound Card

All I own is an old Sound Blaster 16; even if you've got something different, you may take what follows as guidelines.

I compiled the sound card support as a module (`sb.o`). Then I put this in `/etc/conf.modules`:

```
options sb io=0x220 irq=5 dma=1 dma16=5 mpu_io=0x330
alias sound sb
```

To enable the sound, make sure that `modprobe sound` is invoked in `/etc/rc.d/rc.sysinit`. Alternatively, get the tool `sndconfig` from the RedHat site.

2.11 Login Messages

If you wish to customise the login messages, check whether your `/etc/rc.d/rc.local` overwrites `/etc/issue` and `/etc/motd`. (RedHat does.) If so, get on with your editor.

If you'd like a colourised login message, you may adapt your `rc.local` inserting lines like these:

```
# put a real escape character instead of ^[. To do this:
# emacs: ^Q ESC   vi: ^V ESC   joe:  ‘ 0 2 7   jed: ‘ ESC
ESC="^["  # a real escape character
BLUE="$ESC[44;37m"
NORMAL="$ESC[40;37m"
CLEAR="$ESC[H$ESC[J"

> /etc/issue
echo "$CLEAR" >> /etc/issue
echo "$BLUE   Welcome to MyServer (192.168.1.1)   " >> /etc/issue
echo "$NORMAL " >> /etc/issue
echo "" >> /etc/issue
```

2.12 Hostname

Issuing the command `hostname new_host_name` may not be enough. To avoid the dreaded `sendmail` lock, follow these steps (only valid for a stand–alone machine):

- edit `/etc/sysconfig/network` and change the hostname therein (e.g. `new_host_name.localdomain`);
- edit `/etc/HOSTNAME` appropriately;
- append the new hostname in the line in `/etc/hosts`:

  ```
  127.0.0.1       localhost  new_host_name.locadomain
  ```

2.13 Mouse

`gpm` mouse services are useful to perform cut and paste in tty mode, and to use the mouse in some applications. For Red Hat, check that you have a file called `/etc/sysconfig/mouse` and that it reads:

```
MOUSETYPE="Microsoft"
XEMU3=yes
```

Moreover, you must have a file `/etc/rc.d/init.d/gpm`, where you'll put additional command line parameters. Mine reads:

```
...
       daemon gpm -t $MOUSETYPE -d 2 -a 5 -B 132 # two-button mouse
...
```

Obviously, make sure this configuration is right for your mouse type. In most notebooks, `MOUSETYPE` is "`PS/2`".

For Caldera, all you have to do is append this line to `/etc/rc.d/rc.boot`:

```
/usr/bin/gpm
```

On S.u.S.E., `gpm` options go in `/etc/rc.config`; on Debian, you'll edit `/etc/gpm.conf`.

If you like to use menus in console with Ctrl-button, then configure `gpm-root`. Edit the default menu in `/etc/gpm-root.conf`, then launch `gpm-root` from `/etc/rc.d/rc.local`.

2.14 Mount Points

It's handy to have mount points for the floppy, other devices and NFS-exported directories. For example, you can do the following:

```
~# cd /mnt
/mnt# mkdir floppy; mkdir cdrom; mkdir win; mkdir zip; mkdir server
```

This creates mount points for a DOS/Win floppy, an ext2 floppy, the CD–ROM, the Windows partition, the parallel port Zip drive, and a NFS directory.

Now edit the file `/etc/fstab` and add the following entries:

```
/dev/fd0        /mnt/floppy     auto            user,noauto 0 1
/dev/cdrom      /mnt/cdrom      iso9660         ro,user,noauto 0 1
/dev/zip        /mnt/zip        vfat            user,noauto,exec 0 1
/dev/hda1       /mnt/win        vfat            user,noauto 0 1
server:/export  /mnt/server     nfs             defaults
```

Obviously, you must use the correct device in the first field. Recent kernels support fat32 partitions natively; for older ones, there's a kernel patch at `<http://bmrc.berkeley.edu/people/chaffee/fat32.html>`. `man mount` for further information.

Note the "auto" filesystem type in the first line; it allows you to mount both ext2 and vfat (DOS/Windows) floppies. You may find `mtools` more convenient.

2.15 lilo(8) and LOADLIN.EXE

Many users run both Linux and DOS/Windows on their PC, and want to choose at boot time which os to use; this should be done at install time, but in case, do what follows. Let's suppose that `/dev/hda1` contains DOS/Windows and that `/dev/hda2` contains Linux.

```
~# fdisk
Using /dev/hda as default device!

Command (m for help):a
Partition number (1-4): 2

Command (m for help):w
~#
```

This makes the Linux partition bootable. Then write this basic `/etc/lilo.conf` file:

```
boot = /dev/hda2
compact                 # may conflict with "linear"
delay = 100             # 10 seconds
linear                  # gets rid of the "1024 cylinder" problem
# message = /boot/bootmesg.txt  # write your own, if you will
root = current
image = /boot/vmlinuz  # boot linux by default as this entry comes first
  label = linux
  read-only
#  append="mem=128M"   # to see more memory than 64M
other = /dev/hda1
  table = /dev/hda
  label = dos
```

Now issue `/sbin/lilo` and you're done. Being `lilo` a crucial part of your installation, you're strongly advised to read its documentation anyway.

To boot Linux from DOS/Windows without resetting, put `LOADLIN.EXE` in a directory (in the DOS partition!) included in the DOS path; then copy your kernel to, say, `C:\TEMP\VMLINUZ`. The following simple `.BAT` file will boot Linux:

```
rem   linux.bat
smartdrv /C
loadlin c:\temp\vmlinuz root=/dev/hda2 ro
```

If you use Windows 9x, set the properties of this `.BAT` so as it starts in MS–DOS mode.

2.15.1 Security Tip

Making a backup copy of your MBR before installing Linux is a safe move. Use `restorrb` (included in the `FIPS` package) before installation, or a Linux rescue floppy to issue this command:

```
rescue:~# dd if=/dev/hda of=MBR bs=512 count=1
```

then make at least two copies of the file `MBR` on floppies. Should disaster strike, you'll be able to restore your old MBR by issuing:

```
rescue:~# dd if=/mnt/MBR of=/dev/hda bs=446 count=1
```

assuming that a floppy containing `MBR` is mounted under `/mnt`. Alternatively, use a DOS rescue floppy to issue `FDISK /MBR`.

2.16 Mail Capabilities

You will want to be able to read mail messages written in HTML or containing exotic file formats. Make sure you have two files: `/etc/mime.types` and `/etc/mailcap`. The first one lists file types and related extensions, like:

```
application/postscript          ps eps
image/jpeg                      jpe jpeg jpg
text/html                       html
```

while the second one tells the mail client how to display that file type.

You may receive mail from people who use Microsoft Outlook, whose messages are in multi–part MIME format. These two lines, put in `etc/mailcap`, should let you read those messages:

```
text/plain; less %s; needsterminal
text/html; lynx -force_html %s; needsterminal
```

2.17 Printer Configuration

All distributions I know have a configuration tool for setting up the printer (`printtool`, `yast`, or `magicfilter`); if you don't have it, this is a basic manual configuration.

Let's suppose you have a non–PostScript (non "Windows-only" too!) printer you want to use to print raw text (e.g., C source files) and PostScript files via Ghostscript, which is assumed to be already installed.

Setting up the printer involves a few steps:

- find out which one the parallel print device is: try

  ```
  ~# echo "hello, world" > /dev/lp0
  ~# echo "hello, world" > /dev/lp1
  ```

  and take note which one works.

- make two spool directories:

  ```
  ~# cd /var/spool/lpd
  /var/spool/lpd/# mkdir raw; mkdir postscript
  ```

- if your printer exibits the "staircase effect" (most inkjets do), you'll need a filter. Try to print two lines with

  ```
  ~# echo "first line" > /dev/lp1 ; echo "second line" > /dev/lp1
  ```

  if the output is like this:

  ```
  first line
            second line
  ```

  then save this script as `/var/spool/lpd/raw/filter`:

  ```
  #!/bin/sh
  # This filter eliminates the "staircase effect"
  awk '{print $0, "\r"}'
  ```

  and make it executable with `chmod 755 /var/spool/lpd/raw/filter`.

- make a filter for PostScript emulation. Write the following filter as `/var/spool/lpd/postscript/filter`:

```
#!/bin/sh

DEVICE=djet500
RESOLUTION=300x300
PAPERSIZE=a4
SENDEOF=

nenscript -TUS -ZB -p- |
if [ "$DEVICE" = "PostScript" ]; then
        cat -
else
        gs -q -sDEVICE=$DEVICE \
                -r$RESOLUTION \
                -sPAPERSIZE=$PAPERSIZE \
                -dNOPAUSE \
                -dSAFER \
                -sOutputFile=- -
fi

if [ "$SENDEOF" != "" ]; then
        printf "\004"
fi
```

  (in this example an HP DeskJet printer is assumed. Fix it to suit your printer).

- finally, add the following entries in `/etc/printcap`:

```
# /etc/printcap
lp|ps|PS|PostScript|djps:\
        :sd=/var/spool/lpd/postscript:\
        :mx#0:\
        :lp=/dev/lp1:\
        :if=/var/spool/lpd/postscript/filter:\
        :sh:
raw:\
        :sd=/var/spool/lpd/raw:\
        :mx#0:\
        :lp=/dev/lp1:\
        :if=/var/spool/lpd/raw/filter:\
        :sh:
```

For more complex or exotic printing configurations, the Printing-HOWTO awaits you.

If you use `printtool`, be aware that the GSDEVICE chosen by Printtool will work, but not necessarily at its best for your printer. You may consider fiddling a bit with the file `postscript.cfg`; for instance, I changed GSDEVICE from `cdj500` to `djet500` and now my prints come out much quicker.

2.18 SVGATextMode

This utility, available on `<ftp://tsx-11.mit.edu/pub/linux/sources/sbin>`, is useful for changing the console screen resolution, font, and cursor shape. Users whose language include accented characters will be

able to use them in console applications, while notebook users may change the cursor shape to make it more visible.

Edit `/etc/TextConfig` or `/etc/TextMode`, starting with the default VGA definition. Europeans should be happy with this "LoadFont" section:

```
Option "LoadFont"
FontProg "/usr/bin/setfont"
FontPath "/usr/lib/kbd/consolefonts"
FontSelect "lat1u-16.psf"   8x16 9x16 8x15 9x15
FontSelect "lat1u-14.psf"   8x14 9x14 8x13 9x13
FontSelect "lat1u-12.psf"   8x12 9x12 8x11 9x11
FontSelect "lat1u-08.psf"   8x8  9x8  8x7  9x7
```

Once you're done, try your configuration with a command like `SVGATextMode "80x34x9"`, and if everything appears to be working fine, remove the warnings from `/etc/TextMode` and include this line in `etc/rc.d/rc.sysinit`:

```
# SVGATextMode
/usr/sbin/SVGATextMode "80x34x9"
```

Please note that the block cursor only works with some modes; on my notebook, "80x32x9".

3 Common Administration Tasks

So many things to do, so little time! Here is where the fun begins. This section is rather network-centric, though many other tasks await you.

Networking is a vast subject which cannot be fully covered here. The reference is the NET-3 HOWTO, and most distributions provide documentation on setting up network services. Only a few points will be recalled here.

A quick to-do list for the services you may want to install: cron and timed tasks like calendar or reminder, Http, Samba, telnet/ssh access, anonymous ftp, POP/IMAP server, NFS services...

3.1 Network configuration

Although the actual method of starting network services of your distribution may be much more complex, the following script should be enough to get you started:

```
#!/bin/sh

# net-up.sh: set up network access

DEVICE=eth0
IPADDR=192.168.1.100
NETMASK=255.255.255.0
NETWORK=192.168.1.0
GATEWAY=192.168.1.1

ifconfig $DEVICE $IPADDR netmask $NETMASK up
route add -net $NETWORK netmask $NETMASK $DEVICE
route add default gw $GATEWAY
```

This script is handy for enabling network access when you use a rescue disk. Obviously, this lets you only ping, ftp and telnet to the outside.

3.2 Sharing the Internet

One of the most useful tasks for a Linux server. Currently, most stock kernels come with IP firewalling, masquerading and forwarding enabled by default; if in doubt, consult the IP-Masquerade mini-HOWTO to learn how to enable them. Then install ipfwadm (kernels 2.0.x; <http://www.xos.nl/linux/ipfwadm/>) or ipchains (kernels 2.2.x; <http://www.adelaide.net.au/~rustcorp/ipfwchains/ipfwchains.html>). Remember to enable kernel modules for the services you need, e.g. for ftp you'll add this line to `/etc/rc.d/rc.sysconfig`:

```
/sbin/modprobe ip_masq_ftp
```

Other modules are usually found in `/lib/modules/KERNEL-VERSION/ipv4`.

Enabling IP masquerading for other machines in your local network is very simple. First, check the network initialisation scripts (`/etc/sysconfig/network` should be the right place) to see if they contain a line that reads `FORWARD_IPV4=true`. It's used to set `/proc/sys/net/ipv4/ip\_forward` to 1 when the network subsystem comes up.

Add these lines to `/etc/rc.d/rc.sysinit`:

```
# default: packets cannot go reach the outside
/sbin/ipfwadm -F -p deny
# allow all machines on the local network to reach the Internet
/sbin/ipfwadm -F -a m -S 192.168.1.0/24 -D 0.0.0.0/0
# alternatively, allow only these two machines
# /sbin/ipfwadm -F -a m -S 192.168.1.100/24 -D 0.0.0.0/0
# /sbin/ipfwadm -F -a m -S 192.168.1.101/24 -D 0.0.0.0/0
```

If you use a kernel of the 2.2.x series, use `ipfwadm-wrapper` instead of `ipfwadm` to get started quickly.

Now you'll want something to let client machines dial the ISP; I use Mserver (<http://cpwright.villagenet.com/mserver/>). Edit `etc/mserver.conf`; the only entries that you should modify are "checkhost", "shadow", and "cname". Then define your connection(s). Obviously, install one of the available clients on the client machines.

3.3 Restricting Network Access

Let's suppose you connect to the Internet via PPP. Once you're connected, your machine may become vulnerable to attacks. Insert this in `/etc/hosts.allow`:

```
# only allow access to localhost
ALL: 127.
```

and this in `/etc/hosts.deny`:

```
# deny access to everyone
ALL: ALL
```

If you belong to a network with direct Internet access, you had better disable finger, telnet, and possibly other services for security reasons; use `ssh` instead of telnet. The file to edit is `/etc/inet.conf`. Alternatively, you can restrict network access putting this in `/etc/hosts.allow`:

```
in.telnetd: 192.168.1., .another.trusted.network
in.ftpd: 192.168.1., .another.trusted.network
```

and this in `/etc/hosts.deny`:

```
in.telnetd: ALL
in.ftpd: ALL
```

3.4 NFS Exports

It is common to export the home directories on the server; a problem arises if a user's UID and GID are not consistent across different machines. If user guido has UID/GID equal to 500 on `server` and UID/GID equal to 512 on `client`, a convenient configuration is this:

```
# /etc/exports
/tmp            my.client.machine(rw)
/home/guido     my.client.machine(rw,map_static=/etc/nfs/client.map)
```

In `/etc/nfs/client.map` you'll put this:

```
# /etc/nfs/client.map
# NFS mapping for client
#       remote          local
uid     512             500
gid     512             500
```

3.5 Name Server

Not written yet.

4 Software Configuration

These are the the configuration files we are going to custimise: `/etc/profile` `/etc/bashrc` `.bashrc` `.bashrc` `.bash_profile` `.bash_logout` `.inputrc` `.less` `.lessrc` `.xinitrc` `.fvwmrc` `.fvwm2rc95` `.Xmodmap` `.Xmodmap.num` `.Xdefaults` `.jedrc` `.abbrevs.sl` `.joerc` `.emacs` . Don't add users until you have completed your system configuration; you'll put the dot files in `/etc/skel`.

4.1 bash(1)

Arguably, the most important piece of software after the kernel. To tailor the behaviour of `bash`, these are the main files to edit:

- `/etc/bashrc` contains system wide aliases and functions;
- `/etc/profile` contains system wide environment stuff and startup programs;

- `$HOME/.bashrc` contains user aliases and functions;
- `$HOME/.bash_profile` contains user environment stuff and startup programs;
- `$HOME/.inputrc` contains key bindings and other bits.

Examples of these files are shown below. First, the most important: `/etc/profile`. It's used to configure a lot of features in your Linux box, as you will see in the following sections. Please look out for reverse quotes!

```
# /etc/profile

# System wide environment and startup programs
# Functions and aliases go in /etc/bashrc

# This file sets up the following features and programs:
#
#   o path
#   o prompts
#   o a few environment variables
#   o colour ls
#   o less
#   o rxvt
#
# Users can override these settings and/or add others in their
# $HOME/.bash_profile

# set a decent path
PATH="$PATH:/usr/X11R6/bin:$HOME/bin:."

# notify the user: login or non-login shell. If login, the prompt is
# coloured in blue; otherwise in magenta. Root's prompt is red.
# See the Colour-ls mini HOWTO for an explanation of the escape codes.
USER=`whoami`
if [ $LOGNAME = $USER ] ; then
  COLOUR=44  # blue
else
  COLOUR=45  # magenta
fi

if [ $USER = 'root' ] ; then
  COLOUR=41  # red
  PATH="$PATH:/usr/local/bin"
fi

ESC="\033"
PROMPT='\h'    # hostname
STYLE=';1m'    # bold
# PROMPT='\u'  # username
# STYLE='m'    # plain
PS1="\[$ESC[$COLOUR;37$STYLE\]$PROMPT:\[$ESC[37;40$STYLE\]\w\\$ "
PS2="> "
```

```
# no core dumps, please
ulimit -c 0

# set umask
if [ `id -gn` = `id -un` -a `id -u` -gt 14 ]; then
  umask 002
else
  umask 022
fi

# a few variables
USER=`id -un`
LOGNAME=$USER
MAIL="/var/spool/mail/$USER"  # sendmail, postfix, smail
# MAIL="$HOME/Mailbox"        # qmail
NNTPSERVER=news.myisp.it      # put your own here
VISUAL=jed
EDITOR=jed
HOSTNAME=`/bin/hostname`
HISTSIZE=1000
HISTFILESIZE=1000
export PATH PS1 PS2 USER LOGNAME MAIL NNTPSERVER
export VISUAL EDITOR HOSTNAME HISTSIZE HISTFILESIZE

# enable colour ls
eval `dircolors /etc/DIR_COLORS -b`
export LS_OPTIONS='-s -F -T 0 --color=yes'

# customize less
LESS='-M-Q'
LESSEDIT="%E ?lt+%lt. %f"
LESSOPEN="| lesspipe.sh %s"
LESSCHARDEF=8bcccbcc13b.4b95.33b. # show colours in ls -l | less
# LESSCHARSET=latin1
PAGER=less
export LESS LESSEDIT LESSOPEN VISUAL LESSCHARDEF

# fix the backspace key in rxvt/xterm
CTRL_H="\010"
NULL_STRING=" $CTRL_H" # space + backspace
if [ "$NULL_STRING" != "" ] ; then
  stty erase ^?
else
  stty erase ^H
fi

# set xterm title: full path
case $TERM in
  xterm*)
```

```
    PROMPT_COMMAND='echo -ne "\033]0;${USER}@${HOSTNAME}: ${PWD}\007"'
    ;;
esac

for i in /etc/profile.d/*.sh ; do
  if [ -x $i ]; then
    . $i # beware - variables and aliases might get overridden!
  fi
done

# call fortune, if available
if [ -x /usr/games/fortune ] ; then
  echo ; /usr/games/fortune ; echo
fi
```

This is a sample /etc/bashrc:

```
# /etc/bashrc

# System wide functions and aliases
# Environment stuff goes in /etc/profile
# Insert PS1 definitions here if you experience problems.

export CDPATH="$CDPATH:~"

# common aliases
alias cp='cp -i'
alias l=less
alias ls="ls $LS_OPTIONS"
alias mv='mv -i'
alias rm='rm -i'
alias rmbk='/bin/rm -f .*~ *~ *aux *bak *log *tmp 2> /dev/null'
alias u='cd ..'
alias which="type -path"
alias x=startx

# A few useful functions
c ()    # cd to the new directory and list its contents
{
  cd $1 ; ls
}

inst()  # Install a .tar.gz archive in current directory
{
  if [ $# != 0 ]; then tar zxvf $1; fi
}

cz()    # List the contents of a .zip archive
{
  if [ $# != 0 ]; then unzip -l $*; fi
}
```

```
ctgz()  # List the contents of a .tar.gz archive
{
  for file in $* ; do
    tar ztf ${file}
  done
}

tgz()   # Create a .tgz archive a la zip.
{
  if [ $# != 0 ]; then
    name=$1.tar; shift; tar -rvf ${name} $* ; gzip -9 ${name}
  fi
}

crpm()  # list information on an .rpm file
{
  if [ $# != 0 ]; then rpm -qil $1 | less; fi
}
```

This is a sample `.bashrc`:

```
# $HOME/.bashrc
# Source global definitions

if [ -f /etc/bashrc ]; then
  . /etc/bashrc
fi

# this is needed to notify the user that they are in non-login shell
if [ "$GET_PS1" = "" ] ; then
  COLOUR=45; ESC="\033"; STYLE=';1m';  # STYLE='m'
  USER=`whoami`
  export PS1="\[$ESC[$COLOUR;37$STYLE\]$USER:\[$ESC[37;40$STYLE\]\w\\$ "
fi

# personal aliases
alias backup='tar -Mcvf /dev/fd0'
alias dial='eznet up myisp'
alias f='cd ~/fortran'
alias hangup='eznet down'
alias lyx='lyx -width 580 -height 450'
alias restore='tar -M -xpvf /dev/fd0'

# personal functions
xj()    # Launch xjed and a file in background
{
  xjed $1 &
}
```

This is a sample `.bash_profile`:

```
# $HOME/.bash_profile

# User specific environment and startup programs
# This file contains user-defined settings that override
# those in /etc/profile

# Get user aliases and functions
if [ -f ~/.bashrc ]; then
  GET_PS1="NO"  # don't change the prompt colour
  . ~/.bashrc
fi

# set a few 'default' directories
export CDPATH="$CDPATH:$HOME:$HOME/text:$HOME/text/geology"
```

This is a sample `.inputrc`:

```
# $HOME/.inputrc

# key bindings
"\e[1~": beginning-of-line
"\e[3~": delete-char
"\e[4~": end-of-line
# (F1 .. F5) are "\e[[A" ... "\e[[E"
"\e[[A": "info \C-m"

set bell-style visible          # please don't beep
set meta-flag On                # allow 8-bit input (i.e, accented letters)
set convert-meta Off            # don't strip 8-bit characters
set output-meta On              # display 8-bit characters correctly
set horizontal-scroll-mode On   # scroll long command lines
set show-all-if-ambiguous On    # after TAB is pressed
```

To make the backspace and delete keys work correctly in `xterm` and other X11 applications, the following is also needed:

- put this in your `.xinitrc`:

  ```
  usermodmap=$HOME/.Xmodmap
  xmodmap $usermodmap
  ```

- then your `.Xmodmap` will contain:

  ```
  keycode 22 = BackSpace
  keycode 107 = Delete
  ```

  this fixes the console. To fix `xterm`:

- put this in your `.Xdefaults`:

  ```
  xterm*VT100.Translations: #override <Key>BackSpace: string(0x7F)\n\
          <Key>Delete:        string(0x1b) string("[3~")\n\
  ```

```
        <Key>Home:          string(0x1b) string("[1~")\n\
        <Key>End:           string(0x1b) string("[4~")\n\
        Ctrl<Key>Prior:     string(0x1b) string("[40~")\n\
        Ctrl<Key>Next:      string(0x1b) string("[41~")

nxterm*VT100.Translations: #override <Key>BackSpace: string(0x7F)\n\
        <Key>Delete:        string(0x1b) string("[3~")\n\
        <Key>Home:          string(0x1b) string("[1~")\n\
        <Key>End:           string(0x1b) string("[4~")\n\
        Ctrl<Key>Prior:     string(0x1b) string("[40~")\n\
        Ctrl<Key>Next:      string(0x1b) string("[41~")
```

`rxvt` is a wee bit more complicated, as some compile–time options influence its behaviour. See the above `/etc/profile`.

More info in `bash`(1) and `readline`(3) man pages.

Don't expect every application to work correctly! If you run `joe` in `xterm`, for instance, some keys won't work; the same holds for some versions of `rxvt`.

4.2 ls(1)

`ls` can display directory listings using colours to highlight different file types. To enable this feature, you just need a couple of lines in `/etc/profile` as seen above. However, this won't work with some versions of `rxvt`; use some flavour of `xterm` instead. It looks like some old `rxvt`s have a bug that prevents them from inheriting the environment correctly in some circumstances.

Caldera's `ls` doesn't have colours, but there's an equivalent `color-ls`. Add this in `/etc/bashrc`:

```
alias ls="color-ls $LS_OPTIONS"
```

4.3 less(1)

With this excellent pager you can browse not only plain text files, but also gzip compressed, tar and zip archives, man pages, and what have you. Its configuration involves a few steps:

- to use it with the movement keys, have this plain ASCII file `.lesskey` in your home directory:

  ```
  ^[[A   back-line
  ^[[B   forw-line
  ^[[C   right-scroll
  ^[[D   left-scroll
  ^[OA   back-line
  ^[OB   forw-line
  ^[OC   right-scroll
  ^[OD   left-scroll
  ^[[6~  forw-scroll
  ^[[5~  back-scroll
  ^[[1~  goto-line
  ^[[4~  goto-end
  ^[[7~  goto-line
  ^[[8~  goto-end
  ```

  then run the command `lesskey`. (These are escape sequences for vt100-like terminals.) This creates a binary file `.less` containing the key bindings.

- write the following file as `/usr/local/bin/lesspipe.sh`:

```
#!/bin/sh
# This is a preprocessor for 'less'.  It is used when this environment
# variable is set:   LESSOPEN="|lesspipe.sh %s"

lesspipe() {
  case "$1" in
  *.tar) tar tf $1 2>/dev/null ;; # View contents of .tar and .tgz files
  *.tgz|*.tar.gz|*.tar.Z|*.tar.z) tar ztf $1 2>/dev/null ;;
  *.Z|*.z|*.gz) gzip -dc $1  2>/dev/null ;; # View compressed files correctly
  *.zip) unzip -l $1 2>/dev/null ;; # View archives
  *.arj) unarj -l $1 2>/dev/null ;;
  *.rpm) rpm -qpil $1 2>/dev/null ;;
  *.cpio) cpio --list -F $1 2>/dev/null ;;
  *.1|*.2|*.3|*.4|*.5|*.6|*.7|*.8|*.9|*.n|*.l|*.man) FILE=`file -L $1`
    FILE=`echo $FILE | cut -d ' ' -f 2`
    if [ "$FILE" = "troff" ]; then
      groff -s -p -t -e -Tascii -mandoc $1
    fi ;;
  *) file $1 | grep text > /dev/null ;
    if [ $? = 1 ] ; then # it's not some kind of text
      strings $1
    fi ;;
  esac
}

lesspipe $1
```

then make it executable with `chmod 755 lesspipe.sh`.

- put the necessary variables in `/etc/profile` as seen above.

4.4 emacs(1)

I rarely use `emacs`, so I have only a couple of tips for you. Some `emacs` distributions don't come preconfigured for colours and syntax highlighting. Put this in your `.emacs`:

```
(global-font-lock-mode t)
(setq font-lock-maximum-decoration t)
```

This only works in X11. Moreover, to enable accented characters you'll add this line:

```
(standard-display-european 1)
```

I'll leave it to you to peruse all of `emacs`' documentation to find out how to tailor it to your needs—potentially, it can take months of hacking. The Dotfile generator (Section 5 (Configuration Software)) is a good helping hand.

4.5 joe(1)

Some versions of `joe` don't work with colours in console, and some special keys don't work either. A quick and dirty (and inelegant) solution to the former problem is this:

```
~$ export TERM=vt100
~$ joe myfile
   (edit your file)
~$ export TERM=linux
```

To make the special keys work, all you have to do is edit `.joerc`, `.jstarrc` or your favourite emulation; you can start from the system-wide config files in `/usr/lib/joe`. Look for the fourth section (key bindings). This enables Home and End:

```
bol ^[ [ 1 ~    Go to beginning of line
eol ^[ [ 4 ~    Go to end of line
```

Find out the desired ESC sequences typing `cat` followed by the special keys.

4.6 jed(1)

This is my favourite editor: it does what I need, it's lighter and easier to configure than `emacs`, and emulates other editors quite well. Many users at my university use `jed` to emulate `EDT`, VMS' system editor.

`jed`'s configuration files are `.jedrc` and `/usr/lib/jed/lib/*`; the former can be adapted from `jed.rc` in the latter directory.

- to make `jed` use the special keys correctly, write the file `/usr/lib/jed/lib/defaults.sl` whose only line reads:

  ```
  () = evalfile("linux");
  ```

- if `xjed` apparently doesn't recognise the DEL key, add these lines to your `.jedrc`:

  ```
  #ifdef XWINDOWS
    x_set_keysym (0xFFFF, 0, "\e[3~");
    setkey (``delete_char_cmd'', "\e[3~");
  #endif
  ```

- edit `/usr/lib/jed/lib/linux.sl` to specify `Info_Directory = "/usr/info";` and `/bin/mail` after `/UCB_Mailer = "/bin/mail";;`

- to make `jed` emulate `EDT` (or other editors) all you have to do is edit a couple of lines in `.jedrc`. If you want the numeric keypad '+' to delete words instead of a single character, add this in `.jedrc`:

  ```
  unsetkey("\eOl");
  unsetkey("\eOP\eOl");
  setkey("edt_wdel", "\eOl");
  setkey("edt_uwdel", "\eOP\eOl");
  ```

  after the line that reads `() = evalfile("edt")` (or similar);

- to make `xjed` use the numeric keypad for `EDT` emulation, insert the following in `.Xmodmap`:

```
keycode 77  = KP_F1
keycode 112 = KP_F2
keycode 63  = KP_F3
keycode 82  = KP_F4
keycode 86  = KP_Separator
```

- colour customization for `xjed` is done adding lines like these in `.Xdefaults`:

```
xjed*Geometry: 80x32+150+50
xjed*font: 10x20
xjed*background: midnight blue
# and so on...
```

- the "abbreviation" feature is an invaluable timesaver. Write a file like the following as `$HOME/.abbrevs.sl` (you can change this name by inserting `variable Abbrev_File = "/usr/lib/jed/abbrev.sl";` in `.jedrc`):

```
create_abbrev_table ("Global", "0-9A-Za-z");
define_abbrev ("Global", "GG", "Guido Gonzato");
create_abbrev_table ("TeX", "\\A-Za-z0-9");
define_abbrev ("TeX", "\\beq", "\\begin{equation}");
define_abbrev ("TeX", "\\eeq", "\\end{equation}");
% and so on...
```

  and type `ESC x abbrev_mode` to enable it. To have the abbreviation enabled by default, add entries like these to your `.jedrc`:

```
define text_mode_hook ()
{
  set_abbrev_mode (1);
}
%
define fortran_hook ()
{
  set_abbrev_mode (1);
  use_abbrev_table ("Fortran");
}
% and so on...
```

4.7 pine(1)

Edit the global configuration in `/usr/lib/pine.conf`, taking care at least of the following fields: `user-domain`, `smtp-server`, and `nntp-server`. Note that `inbox-path` depends on your MTA: if you use `sendmail` or `postfix`, that'll be `var/spool/mail/$USER`; with Qmail, `/home/\$USER/Mailbox` (but root will use `/var/qmail/alias/Mailbox`.

4.8 minicom(1)

Users can't use `minicom` unless a global configuration has been made by root. Remember to make it.

4.9 efax(1)

This package is probably the most convenient for simple sending/receiving of faxes. You'll have to tailor the script `/usr/bin/fax`; easy job, but a couple of quirks caused me quite a headache:

- to find out whether your modem is class 1, 2, or 2.0, use minicom or similar program to issue the command `at+fclass=?`. The reply may be like `0,1,2`; 1 and 2 are the classes supported by your modem;
- DIALPREFIX: chances are that simply putting '`T`' or '`P`' won't work in some countries (in Italy, at least). Put '`ATDT`' or '`ATDP`' instead;
- INIT and RESET: these strings contain the initialisers '`-i`' and '`-k`', needed by `efax`. If you want to add an AT command, add it to the appropriate string leaving out '`AT`' and preceding the rest with either '`-i`' or '`-k`'. Example: to add the '`ATX3`' command to INIT, you'll append '`-iX3`'.

That done, there are a few permissions to fix to enable non-root users to send and receive faxes. The directories `/var/lock` and `/var/spool/fax` must be writable. To do so, create the group `faxusers`, add users to it, then type:

```
~# chown root.faxusers /var/lock
~# mkdir /var/spool/fax
~# chown root.faxusers /var/spool/fax; chmod g+w /var/spool/fax
```

4.10 Ghostscript

This essential tool suffers from a small snag. Owing to to the well-known export regulations in the USA, the utility `pdf2ps` doesn't work on encrypted .pdf files. Never mind: turn your browser to `<http://www.ozemail.com.au/~geoffk/pdfencrypt>`, download the file `pdf_sec.ps` and put it in place of the file with the same name that comes with the distribution of Ghostscript.

4.11 TeX and Friends

I'll assume you have the teTeX distribution.

- download additional LaTeX packages from your nearest CTAN mirror site, e.g. `<ftp://ftp.dante.de/pub/tex>`. Add the files under `/usr/share/texmf/tex/latex`, then run the command `texhash` so that teTeX recognises the new package;
- to configure the hyphenation pattern for your language, edit the file `/usr/share/texmf/tex/generic/config/language.dat`, then do:

  ```
  ~# texconfig init ; texconfig hyphen
  ```

- to tailor `dvips`, the file to edit is `/usr/share/texmf/dvips/config/config.ps`. Be aware that the fields regarding the default resolution also affect `xdvi`'s behaviour; if you experience annoying attempts to create fonts each time you run it, put the line

  ```
  XDvi*mfmode:
  ```

  in `.Xdefault`. This should help.

- to include PostScript figures that reside in subdirectories, you can expand TeX's search path to include subdirectories. Put this command in your `.bash\_profile`:

  ```
  export TEXINPUTS="$HOME/figures::./figures"
  ```

  which makes TeX search in `$HOME/figures` before the default directories, and `./figures` after the default directories.

4.12 Avoid PPProblems!

I'll take it for granted that your kernel has PPP + TCP/IP support compiled in, that loopback is enabled, and that you already have the `pppd` package correctly installed and, if you will, suid root. Obviously, your ISP must support PPP.

There are now two ways to get PPP to work: a) manual configuration, and b) a configuration program that automagically sees to it. Whichever option you choose, have the following information on hand:

- your ISP's telephone number;
- your ISP's name, mail and news server address;
- your ISP's domain;
- your username and password.

Manual configuration is a drudgery. It's about editing files and writing scripts; not too much work, but it's easy to make mistakes and newcomers are often intimidated. The PPP HOWTO is there for you. Alternatively, there are tools that ask for the information above and do all the work.

Gnome and KDE include, respectively, `gnome-ppp` and `kppp` which are easy enough to set up. Alternatively, I suggest that you have a look at a couple of tty–based tools, `wvdial` and `eznet`. You feed them your ISP's phone number, your username, your password, and you're in business. Their home pages are at `<http://www.worldvisions.ca/wvdial>` and `<http://www.hwaci.com/sw/eznet>`. Both are great, but I prefer the latter.

4.12.1 A Quick Start with eznet

First of all, create an `/etc/resolv.conf` like this:

```
nameserver w.x.y.z
```

where you'll insert the address of your ISP's nameserver. To create an account with `eznet`, issue the following command:

```
#~ eznet add service=YOUR_ISP user=NAME password=PASSWORD phone=PHONE
```

which creates the file `/var/eznet/eznet.conf`, owned by root.root with permissions 600; chmod it to 666 if you want it to be world readable. Now dial your ISP with `eznet up YOUR_ISP`. If the modem keeps waiting for the dial tone and won't connect, then try this command:

```
#~ eznet change YOUR_ISP init0=atx3
```

To hang up, the command is `eznet down`. That's all!

4.12.2 A Quick Start with wvdial

`wvdial`'s setup is even shorter. Type `wvdialconf /etc/wvdial.conf`, then edit the resulting file to include your username, password, and phone number. Try it out with `wvdial`, and keep your fingers crossed. To hang up, stop it with Ctrl-C.

4.13 POP Client

To retrieve your mail from a POP3 server, you use a POP client like `fetchpop` or `fetchmail`; the latter is more advanced, but requires that you run `sendmail`. This is a bit of an overkill on low-spec machines. They're available on <ftp://metalab.unc.edu/pub/Linux/system/mail/pop>.

To configure these clients:

- `fetchpop`: the first time you run it, you'll be prompted for some information. Answer the questions and you're set. `fetchpop` must be used with the `-r` switch if your ISP's POP3 server doesn't implement the command LAST properly.
- `fetchmail`: adapt this sample `.fetchmailrc`:

  ```
  # $HOME/.fetchmailrc
  poll mbox.myisp.com with protocol pop3;
    user john there with password _Loo%ny is john here
  ```

  One user reported that adding "smtphost localhost" to the second line improved performance dramatically.

  You must set the permissions to this file with the command `chmod 600 .fetchmailrc`, otherwise `fetchmail` will rightly refuse to start. This example is very basic; there are endless possibilities of configuration. Check out at <http://www.ccil.org/~esr/fetchmail>.

4.14 X Window System (XFree86)

4.14.1 Setting Up the X Server

Come on, it's not difficult as it used to be... All major distributions include a tool for setting up X11 (e.g. `XConfigurator`, `sax`, `XF86Setup`, or at least `xf86config`). X configuration is virtually automatic these days, but a few video cards may refuse to work. Each time I experience problems setting up X on a system, I resort to a simple method that has always worked:

- make sure that the plain VGA server is installed;
- go to <ftp://ftp.XFree86.org/pub/XFree86/current/binaries>, cd to the proper Linux subdirectory, and download the archives `X_version_bin.tgz`, `X_version_set.tgz`, and all the servers. Amongst other programs, the first archive contains the most up-to-date `SuperProbe`;
- unpack `X_version_bin.tgz` to a temporary directory, cd to it, and run `./SuperProbe`. If your video card is recognised, chances are that you'll be able to set it up. Otherwise, hard luck;
- install the servers and `X_version_set.tgz` from `/usr/X11R6/`, then run `XF86Setup`.

This has always worked for me, but your mileage may vary. Please note that most times X11 won't start because the specs you choose for your monitor were too high! Start with conservative settings, i.e. 800x600 and 256 colours, then pump it up. *Warning*: these operations are dangerous and your monitor might be damaged!

4.14.2 Keypad

We have seen above how to make a few special keys work. The sample file `.Xmodmap` works well if you want to use Xjed, but it makes the keypad unusable. You'll then need another config file, which we'll call `.Xmodmap.num`:

```
! Definitions can be found in <X11/keysymdef.h>

keycode 77  = Num_Lock
keycode 112 = KP_Divide
keycode 63  = KP_Multiply
keycode 82  = KP_Subtract
keycode 86  = KP_Add
keycode 79  = KP_7
keycode 80  = KP_8
keycode 81  = KP_9
keycode 83  = KP_4
keycode 84  = KP_5
keycode 85  = KP_6
keycode 87  = KP_1
keycode 88  = KP_2
keycode 89  = KP_3
keycode 90  = KP_0
keycode 91  = KP_Decimal
```

Make sure that your `/etc/X11/XF86Config` does not contain these three lines:

```
ServerNumLock
Xleds
XkbDisable
```

and in case, comment them out. To re-enable the keypad, you'll issue the command `xmodmap .Xmodmap.num`.

4.14.3 Graphical Login with xdm

To be greeted by a graphical login, edit the file `/etc/inittab`, which should include a line like this:

```
x:5:respawn:/usr/bin/X11/xdm -nodaemon
```

where 5 is the runlevel corresponding to X11 (S.u.S.E. uses 4). Modify the line that defines the default runlevel (usually 2 or 3), changing it as above:

```
id:5:initdefault:
```

The number of colours is specified in `/etc/X11/xdm/Xserver`:

```
:0 local /usr/X11R6/bin/X :0 -bpp 16 vt07  # first X server, 65k colours
:1 local /usr/X11R6/bin/X :1 -bpp 32 vt08  # second X server, true colour
```

If you already have `.xinitrc`, copy it to `.xsession` and make the latter executable with `chmod +x .xsession`. Now issue the command `telinit 5` and you're in business.

4.14.4 Window Manager

Once X works, there are endless possibilities of configuration; it depends on the window manager you use, there are tens to choose from. Mostly, it's all down to editing one or more ASCII files in your home directory; in other cases you don't have to edit a thing, and use an applet or even a menu.

Some examples:

- **the fvwm family**: copy `/etc/X11/fvwm/system.fvwmrc` (or similar) to your home using the appropriate name, browse it and start experimenting. You may waste a lot of time before you get the precise look and feel you like;
- **WindowMaker**: it has several config files that live under `$HOME/GNUstep`, and a cool configuration applet;
- **KDE** and **Gnome**: nothing to edit manually here: everything can be done via the menu.

In short: if you don't mind editing config file, choose something like `icewm`, `fvwm*`, `blackbox` etc; if you do mind, the choice is currently restricted to KDE, Gnome, WindowMaker, and XFCE.

It's important to have a good `.xinitrc`. An example:

```
#!/bin/sh
# $HOME/.xinitrc

usermodmap=$HOME/.Xmodmap
xmodmap $usermodmap

xset s noblank  # turn off the screen saver
xset s 300 2    # screen saver start after 5 min
xset m 10 5     # set mouse acceleration

rxvt -cr green -ls -bg black -fg white -fn 7x14 \
  -geometry 80x30+57+0 &

if [ "$1" = "" ] ; then  # default
  WINMGR=wmaker
else
  WINMGR=$1
fi

$WINMGR
```

Although it doesn't appear to be strictly required, make it executable with `chmod +x .xinitrc`.

This `.xinitrc` lets you choose the window manager: try

```
$ startx startkde # or other w.m.
```

(it doesn't work with some S.u.S.E. versions, though).

4.14.5 Defaults for X11 Apps

Find out where the `app-defaults` directory is (it should be `/usr/X11R6/lib/X11/app-defaults`). Several apps keep a configuration file there.

4.15 Users' Configurations

When you're done editing the dot files, copy them to `/etc/skel` as seen in Section 4 (Software Configuration).

4.16 Making .rpms

`rpm` is such a wonderful method of keeping packages under control that I'm reluctant to install .tar.gz archives but in very few special cases (e.g., security). Whenever you install a tarball, consider turning it into an .rpm archive, then reinstall it; consult the RPM HOWTO. Also, if you use modern `gcc` versions like `egcs` or `pgcc`, it may be advisable to put this in your `/etc/rpmrc`:

```
optflags: i386 -O2 -mpentium
```

4.17 Upgrading

If you upgrade your machine, do your backup as usual and remember to save a few additional files. Some could be `/etc/X11/XF86Config`, `/usr/bin/fax`, all the stuff in `/usr/local`, the kernel configuration, the whole `/etc`, and all the mail in `/var/spool/mail`.

Then it's time to upgrade (in rare cases, downgrade!) applications that your distribution ship with, and to add additional packages. Keep a list of these ones.

5 Configuration Software

There are several programs that make Linux easy to setup and configure. Some are becoming sort of standard: Red Hat, Caldera and other distributions ship with apps like `setup`, `printtool`, `netcfg`, `usertool`, etc, while S.u.S.E. ships with a comprehensive configuration program called Yast. Other useful programs are:

- **The Dotfile Generator**: fine X app with modules to configure packages like `emacs`, `bash`, `procmail` and more. Its page is at `<http://www.imada.ou.dk/~blackie/dotfile>`;
- **Linuxconf**: the ultimate configuration tool. It can do everything, both in console and under X. Go to `<http://www.solucorp.qc.ca/linuxconf>` at once.

6 The End

6.1 Copyright

Unless otherwise stated, Linux HOWTO documents are copyrighted by their respective authors. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.

If you have questions, please contact Tim Bynum, the Linux HOWTO coordinator, at <linux-howto@metalab.unc.edu> via email.

6.2 Feedback

Perhaps even more than other HOWTOs, this one needs and welcomes your suggestions, criticisms, and contributions. Not only is feedback welcome: it's necessary. If you think something is missing or wrong, please email me. If you have a distribution other than Red Hat/Mandrake and your config files are different or placed in other directories, please tell me and I'll include your tips. My aim is making life with Linux as easy as possible.

Linux has a huge number of packages, so it's impossible to include directions for all of them. Please keep your requests/suggestions pertinent to the "most reasonable" programs—I'll leave it to your common sense.

6.3 Disclaimer

"Configuration HOWTO" was written by Guido Gonzato, <REMOVE_MEguido@ibogeo.df.unibo.it>. (Remove "REMOVE_ME".) Many thanks to all other HOWTO authors and man pages writers/maintainers, whose work I've shamelessly pilfered; and to all people who provided me with feedback.

This document is provided "as is". I put great effort into writing it as accurately as I could, but you use the information contained in it at your own risk. In no event shall I be liable for any damages resulting from the use of this work.

I hope you'll find this work useful, though. Whenever I install a new Linux box, I actually do...

Enjoy,

Guido =8-)