Configuration HOWTO

Table of Contents

Table of Contents

Configuration HOWTO

By Guido Gonzato, `ggonza at tin.it`

Version 1.99.6. 31 August 2000.

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

1.Introduction

2.General System Setup

3.Common Administration Tasks

4.Software Configuration

5.Configuration Software + Docs

6.The End

1.Introduction

1.1 Why This HOWTO

Current distributions are approaching perfection, but some fine−tuning is still needed. Many new users are intimidated by the apparent complexity of a Linux system, and as a result I note 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.

I realise that this HOWTO is quite RedHat−centric. Currently, I only have access to Red Hat and Mandrake machines, kernels ranging from 2.0.36 to 2.2.15; so don't take any of my tips as gospel if you have other distributions. Previous versions of this HOWTO provided some information for SuSE, Debian and Caldera; but as I no longer have access to those machines, I can't keep the details up−to−date. No information is better that inaccurate information, so it's up to you to to adapt my tips to your distribution.

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, which also hosts all other HOWTOs I refer to and some translations, is http://www.linuxdoc.org.

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.securityportal.com/lasg and the Linux Security Guide at http://nic.com/~dave/SecurityAdminGuide/index.html. You should consider at least the following steps: using shadow passwords (Shadow Password HOWTO), restricting network access to the machine (Section Restricting Network Access), using the Secure Shell (http://www.openssh.org) 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 fresh 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:

- look for a suitable key table starting from /usr/lib/kbd/keymaps/i386; for example, querty/it–latin1.kmap.gz supports the Italian keyboard;
- edit the file /etc/sysconfig/keyboard so as it reads: `KEYTABLE="it-latin1"`;
- to set up the keyboard repeat rate and delay time, add this line to /etc/rc.d/rc.sysinit:

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

To load the key table, issue

```
/etc/rc.d/init.d/keytable start
```

Other special keys will be dealt with 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
```

Normally, the Linux console doesn't distinguish between, say, Arrow and Shift−arrow, but some applications (namely, the editor Jed) do. Normally, these key bindings are only available in xterm. The following key map, which you can choose to load at boot time, is very handy:

```
# Load this key map with: loadkey shift.map
# Shift + Up
shift keycode 103 = F100
string F100 = "\033[a"
# Shift + Left
shift keycode 106 = F101
string F101 = "\033[c"
# Shift + Right
shift keycode 105 = F102
string F102 = "\033[d"
# Shift + Down
shift keycode 108 = F103
string F103 = "\033[b"
# Ctrl + Ins
control keycode 110 = F104
string F104="\033[2^"
# Shift + Ins
shift keycode 110 = F105
string F105="\033[2$"
# Shift + PgUp
shift keycode 104 = F106
string F106 = "\033[5$"
# Shift + PgDn
shift keycode 109 = F107
string F107 = "\033[6$"
# Shift + Home
shift keycode 102 = F108
string F108 = "\033[1$"
# Shift + End
shift keycode 107 = F109
string F109 = "\033[4$"
# Shift + Del
shift keycode 111 = F110
string F110 = "\033[3$"
# Ctrl + Del
control keycode 111 = F111
```

```
string F111 = "\033[3^"
```

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 (say, `mkbootdisk` or something like that); 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 root 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, whose home page is http://www.toms.net/rb. It's very complete, but some utilities seem missing at first; for example, `ftp` is not there but you get `nc` (netcat) instead. Please read its documentation.

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;
- notebook users: if you plan to use a PCMCIA modem/fax, remember to compile serial support *in the kernel*. Don't compile it as a module, or your PCMCIA modem won't work;
- remember to compile everything you need; i.e., don't forget the pcmcia modules, or the ALSA sound drivers;
- 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. If you upgrade the kernel and use ``make oldconfig'', your old config file will be taken and any not included features will be prompted whether they should be included, resulting in a new, upgraded config file.

2.6 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
```

2.7 Parallel Port Zip Drive

Stock kernels include the driver for both old (ppa) and new (imm) Zip drives. If you recompile the kernel, 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 have better use kernel modules.

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  # imm for recent models
```

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.

2.8 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 wish, do `chmod 666` to these devices to make them fully accessible by every user.

2.9 Sound Card

My desktop PC is fitted with 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.

Besides the standard kernel sound drivers, the Alsa drivers (http://www.alsa−project.org) are an excellent choice. Strangely, though, the sound channels are muted by default. You'll want to use `aumix` and this

/etc/aumixrc to set the volume to 100%:

```
vol:100:100:P
synth:100:100:P
pcm:100:100:P
line:100:100:P
mic:100:100:R
cd:100:100:P
```

2.10 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
```

2.11 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.your_domain`);
- edit /etc/HOSTNAME appropriately;
- append the new hostname in the line in /etc/hosts:

```
127.0.0.1       localhost  new_host_name.your_domain
```

2.12 Mouse

`gpm` mouse services are useful to perform cut and paste in tty mode, and to use the mouse in some applications. Make sure 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''.

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.

2.13 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; mkdir floppy cdrom win zip server
```

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

Note the `auto' filesystem type in the first line; it allows you to mount both ext2 and vfat (DOS/Windows) floppies, but you need a recent version of `mount`. You may find `mtools` more convenient.

2.14 Automount Points

If you don't like the mounting/unmounting thing, consider using `autofs`(5). You tell the autofs daemon what to automount and where starting with a file, /etc/auto.master. Its structure is simple:

```
/misc   /etc/auto.misc
/mnt    /etc/auto.mnt
```

In this example you tell autofs to automount media in /misc and /mnt, while the mountpoints are specified in /etc/auto.misc and /etc/auto.mnt. An example of /etc/auto.misc:

```
# an NFS export
server          -ro                     my.buddy.net:/pub/export
# removable media
cdrom           -fstype=iso9660,ro      :/dev/hdb
floppy          -fstype=auto            :/dev/fd0
```

Start the automounter. From now on, whenever you try to access the inexistent mount point /misc/cdrom, il will be created and the CD−ROM will be mounted.

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 = win
```

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.

Security Tip

Making a backup copy of your MBR before installing Linux is a safe move. Prepare a Windows rescue floppy, and make sure it includes `FDISK.EXE`. To restore the MBR, all you have to do is

```
A:\> fdisk /mbr
```

art 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
-*-
```

––>

2.16 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. Adapt it to 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.

2.17 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, "80x30x9".

3.Common Administration Tasks

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...

3.1 Network Configuration

If your network card wasn't recognised at install time, don't worry: in most cases it's either NE2000 or 3c59x compatible. Issue the command `modprobe ne` or `modprobe 3c59x` and see if the relevant module is loaded, then add this line in /etc/conf.modules:

```
alias eth0 ne  # or 3c59x
```

Now you're ready to use `netcfg` or similar tool to set up the network configuration. The relevant files are /etc/HOSTNAME, etc/hosts, /etc/resolv.conf, /etc/sysconfig/network, and /etc/sysconfig/network−scripts/ifcfg−eth0; services should be started with scripts in /etc/rc.d/init.d.

This is a sample etc/hosts:

```
127.0.0.1               localhost
192.168.1.1             paleo.eocene.net        paleo
192.168.1.2             nautilus.eocene.net     nautilus
```

This is /etc/resolv.conf:

```
search df.unibo.it,eocene.net
nameserver 195.210.91.100
```

This is /etc/sysconfig/network (Red Hat−dependent):

```
NETWORKING=false
FORWARD_IPV4=true
HOSTNAME=nautilus.eocene.net
DOMAINNAME=eocene.net
```

And finally, /etc/sysconfig/network−scripts/ifcfg−eth0. This one, too, is Red Hat−dependent; it must be executable.

```
DEVICE=eth0
IPADDR=192.168.1.2
NETMASK=255.255.255.0
NETWORK=192.168.1.0
BROADCAST=192.168.1.255
ONBOOT=no
```

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; it won't start any daemon.

3.2 Network for Notebooks

When you plug the network PC card in, the script /etc/pcmcia/network will be executed. All it needs is a properly set up /etc/sysconfig/network−scripts/ifcfg−eth0.

Setting up the network can become a bit trickier, though. In fact, you must provide the right settings for each network you connect to, as well as settings for the notebook when it's not connected.

I rolled up a rough but functional solution. I use my notebook as a stand−alone machine, connecting to the net via PPP; at home, IP address 192.168.1.2; and at university, IP 137.204.x.y. So, I created a set of configuration files for each network; all these are kept in /etc/mobnet. A script is then used to select the working environment. For instance, this is /etc/mobnet/home.cfg:

```
# /etc/mobnet/home.conf

HOSTNAME=nautilus.eocene.net    # complete hostname
DOMAINNAME=eocene.net           # your domain
IPADDR=192.168.1.2
NETMASK=255.255.255.0
NETWORK=192.168.1.0
BROADCAST=192.168.1.255
GATEWAY=192.168.1.1
FORWARD_IPV4=true
NAMESERVER=195.210.91.100       # required
SEARCH=df.unibo.it,eocene.net   # optional
SERVICES="inet httpd smb sshd"
```

This is `mnet`, the script I use to choose the network profile:

```
#!/bin/sh
# mnet: script to set up the "mobile network" configuration.
# Last modified: 15 July 2000

# start or stop services
activate_services()
{
  for service in $(echo $SERVICES) ; do
    [ -x /etc/rc.d/init.d/$service ] && /etc/rc.d/init.d/$service $1
  done
}

# usage
if [ $# = 0 ] ; then
  echo "Usage: mnet <config name>"
  echo "Example: mnet office"
  exit 1
fi

# check if the configuration exists
if [ ! -e /etc/mobnet/$1.conf ]; then
  echo "This configuration doesn't exist."
  exit 1
fi

# read the configuration
. /etc/mobnet/$1.conf

# set up the host name
echo $HOSTNAME > /etc/HOSTNAME
/bin/hostname $HOSTNAME

# set up the name server(s)
cat <<EOF > /etc/resolv.conf
# /etc/resolv.conf
search $SEARCH
nameserver $NAMESERVER
EOF

# stop previous services, if any
if [ -f /etc/mobnet/services.prev ]; then
  NEWSERVICES=$SERVICES
  . /etc/mobnet/services.prev
  activate_services stop
  SERVICES=$NEWSERVICES
fi

if [ $1 != "none" ]; then
# set up the network parameters
  cat <<EOF > /etc/sysconfig/network
  NETWORKING=yes
  FORWARD_IPV4=true
  HOSTNAME=$HOSTNAME
  DOMAINNAME=$DOMAINNAME
  GATEWAY=$GATEWAY
  GATEWAYDEV=eth0
EOF
```

```
  cat <<EOF > /etc/sysconfig/network-scripts/ifcfg-eth0
  DEVICE=eth0
  IPADDR=$IPADDR
  NETMASK=$NETMASK
  NETWORK=$NETWORK
  BROADCAST=$BROADCAST
  ONBOOT=no
EOF
  /bin/chmod +x /etc/sysconfig/network-scripts/ifcfg-eth0

  # copy the other config files
  /bin/cp -f /etc/mobnet/hosts.$1       /etc/hosts
  /bin/cp -f /etc/mobnet/smb.conf.$1    /etc/smb.conf

  echo -n "Insert the network PC card and press <enter> when done: "
  read

  # OK, now start services
  activate_services start
  echo "SERVICES=\"$SERVICES\"" > /etc/mobnet/services.prev

else # it's not "none"

  cat <<EOF > /etc/sysconfig/network
  NETWORKING=false
  FORWARD_IPV4=false
  HOSTNAME=$HOSTNAME
  DOMAINNAME=$DOMAINNAME
EOF
  /bin/rm -f /etc/sysconfig/network-scripts/ifcfg-eth0*
  /sbin/ifconfig eth0 down
  echo "SERVICES=$SERVICES" > /etc/mobnet/services.prev
  echo "Now you may remove the PC card."
  exit 0

fi

# end of mnet.
```

As I said, it is rough and even not complete: other files may depend on the network, like /etc/fstab, /etc/exports, and /etc/printcap. Think about network printers and NFS shares. Feel free to adapt this bare−bone solution to your needs.

3.3 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 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. More information at http://ipmasq.cjb.net.

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 a suitable client on the client machines.

3.4 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
```

3.5 NFS Exports

It is common to export 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 = 500 on `server` and UID/GID = 512 on `client`, a convenient configuration is this:

```
# /etc/exports
/tmp            my.client.machine(rw)
/home/guido     my.client.machine(rw,all_squash,anonuid=512,anongid=512)
```

3.6 Samba

Almost trivial, but there's always a little bit to do. If you want to connect Windows 98/NT clients, did you remember to read the docs and, in case, enable clear text passwords? The distribution includes .reg files for Win9x/NT/2000; if your clients can't connect to the Linux server, load them on every client.

Samba comes with a fairly complete sample /etc/smb.conf, but strangely it lacks a section showing how to (un)mount removable media. The clauses `preexec` and `postexec` do the trick:

```
[cdrom]
```

```
    comment = CD-ROM
    path = /mnt/cdrom
    public = yes
    read only = yes
  ; you might need to use "root preexec/postexec"
    preexec = mount /mnt/cdrom
    postexec = umount /mnt/cdrom
```

Also: you know what Swat is, don't you? Enable it adding this line in your /etc/inetd.conf:

```
swat      stream  tcp     nowait.400      root /usr/sbin/swat swat
```

and this in `/etc/services`:

```
swat            901/tcp
```

Restart `inetd` with SIGHUP, and point your browser to `http://localhost:901`.

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:
# path, prompts, a few environment variables, colour ls, less,
# rxvt, Backspace key behaviour, xterm title.
#
# Users can override these settings and/or add others in their
# $HOME/.bash_profile

# first: root or normal user? Set PATH and umask accordingly. Note that the
# PATH is normally set by login(1), but what if you access the machine
# via ssh?

if [ $(id -gn) = $(id -un) -a $(id -u) -gt 14 ]; then
  umask 002  # normal user
  PATH="/usr/local/bin:/bin:/usr/bin:."
else
  umask 022  # root
  PATH="/sbin:/bin:/usr/sbin:/usr/bin"
fi

# Now extend the PATH.
PATH="$PATH:/usr/X11R6/bin:$HOME/bin:." # !!! Beware of ./ !!!

# notify the user: login or non-login shell. If login, the prompt is
# blue; otherwise, 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" # my choice
fi

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

# Ulimits: no core dumps, max file size 200 Mb.
ulimit -c 0 -f 200000

# 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 PAGER

# you might need this to fix the backspace key in rxvt/xterm
stty erase ^H  # alternative: ^?

# set xterm title: full path
case $TERM in
  xterm*|rxvt)
    PROMPT_COMMAND='echo -ne "\033]0;${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 I18n

(This section doesn't apply to native English speakers.)

A.k.a. ``internationalisation''. Gasp. This long word means ``to adapt Linux to your local conventions: language, format of date, currency etc.''.

Although Red Hat has its own method for setting up i18n (/etc/sysconfig/i18n), you may want to enable your language only in some cases. I, for one, enabled i18n in `kdm` (via `kdmconfig`) and `xfce`, but want to read English messages when I work in console or xterm.

Consider these lines:

```
LANG=it # choose your language: fr, de, es, ...
LANGUAGE=it
```

```
LC_ALL=it
export LANG LANGUAGE LC_ALL
```

If you insert them in your `.xinitrc` or `.xsession` just before the line that starts the window manager, you'll get internationalised messages − including those in xterms started from within the window manager. But if you'd rather get English messages, set the language to ``en'' and put the same lines in `.bash_profile`.

4.3 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 old versions of `rxvt`; use some flavour of `xterm` instead. It looks like some old `rxvts` have a bug that prevents them from inheriting the environment correctly in some circumstances.

4.4 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/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
  *.bz2) bzip2 -dc $1  2>/dev/null ;;
  *.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 variables that affect `less` in /etc/profile as seen above.

4.5 Editor

Only the most popular will be covered here.

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 Configuration Software) is a good helping hand.

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.

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.

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

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

```
  setkey (``delete_char_cmd'', "\e[3~");
#endif
```

- 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 enable the abbreviation 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...
```

4.6 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.

4.7 minicom(1)

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

4.8 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 or (mandrake) /etc/fax.config; 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 # if it doesn't exist yet
~# chown root.faxusers /var/spool/fax; chmod g+w /var/spool/fax
```

As a normal user, you'll issue `newgrp faxusers` before sendig a fax.

4.9 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 with encrypted .pdf files. Never mind: point your browser to http://www.ozemail.com.au/~geoffk/pdfencrypt, download the file `pdf_sec.ps` and replace the file with the same name that comes with the Ghostscript distribution.

4.10 TeX and Friends

The ``root'' of a TeX system is the directory $TEXMF, which is /usr/share/texmf in teTeX; other distributions may differ (search for ``texmf'' on your system). You normally add stuff or fiddle with files therein.

Expanding $TEXINPUTS

To include PostScript figures or TeX files that reside in subdirectories, it is convenient to expand TeX's search path to include subdirectories. Put this command in your .bash_profile:

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

which makes TeX search in `$HOME/mylib`*before* the default directories, and the directory `./figures`*afterwards*.

Hyphen Patterns

To configure the hyphenation pattern for your language, edit the file $TEXMF/tex/generic/config/language.dat, then do:

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

Even if you don't write in English, don't remove the entry ``english''; TeX pukes without it.

Dvips

To tailor `dvips`, the file to edit is $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 your `.Xdefault`. This should help.

Adding LaTeX Packages

Additional LaTeX packages are available from your nearest CTAN (Comprehensive TeX Archive Network) mirror site, e.g. ftp://ftp.dante.de/pub/tex. Unpack the package under $TEXMF/tex/latex.

If no .sty file exist, run the command `latex newstyle.ins` or `latex newstyle.dtx` to create it, then run the command `texhash` so that teTeX recognises the new package.

4.11 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, set uid 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.

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!

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.

4.12 POP Client

To retrieve your mail from a POP3 server, you need a POP client. Most such clients require that you run an MTA like `sendmail`, `qmail` or `postfix`; a bit of an overkill on low−spec machines. However, there are clients that work without an MTA. The first kind is well represented by `fetchmail`; the second by `fetchpop` or `frenchie`. Sites: ftp://metalab.unc.edu/pub/Linux/system/mail/pop, http://www.lowcountry.com/~jscottb/tcltk.shtml.

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.
- `frenchie`: as above, edit /.frenchie/frenchierc;
- `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.

4.13 Basic Mail Filtering

You will want to protect yourself from spam or huge mail messages. There are two cases: 1) permanent connection to the net, 2) a POP link. In the first case, you can write a `.procmailrc` file, while in the second there are tools for checking the mail prior to fetching it.

A very simple `.procmailrc` that defines a few rules:

```
# $HOME/.procmailrc

MAILDIR=$HOME/mail # make sure it exists

# Store messages directed to the "foo" mailing list to $HOME/mail/foo
:0
* ^To:.*foo
foo

# Discard messages that are not explicitly sent to me or to one of the
# mailling lists I subscribed to.
```

```
:0
* !^TO(guido|jed|lugvr|ldp|nobody)
/dev/null

# ditto, for messages larger than 50k.
:0
* > 50000
/dev/null
```

`man procmailex` for further examples.

POP users will want to use `poppy`, a useful Perl script for checking the mail before fetching it. Get it from ftp://metalab.unc.edu/system/mail/pop.

4.14 X Window System (XFree86)

Setting Up the X Server

Come on, it's no longer as 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 cause a headache.

First of all, check out at the XFree86 site (http://www.xfree86.org) whether your video card is supported. If so, then try this procedure:

- install the plain VGA server;
- 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 you chose wrong specs for your monitor! 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!

If your card isn't supported, you can either: 1) wait for the next version of XFree86; 2) buy a commercial X server; 3) buy a supported video card. *Quartum non datur.*

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`.

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 # also kdm or gdm
```

where 5 is the runlevel corresponding to X11. 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.

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**, **Gnome**, **xfce** and others: 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. Email me if I'm wrong.

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`.

The `.xinitrc` above lets you choose the window manager: try

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

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.

Adding Fonts

Recent versions of XFree86 (say, > 3.3.4) use an X Font Server that supports PostScript Type 1 and True Type fonts natively, so you can use the wealth of fonts available on the net. There's a simple procedure to follow.

Suppose that you download a Type 1 font collection, e.g. Freefont (ftp://ftp.gimp.org/pub/gimp/fonts/freefonts−0.10.tar.gz). To make it visible to the font server, unpack the archive from /usr/X11R6/lib/X11/fonts/. Then edit /etc/X11/fs/config, add an entry for the new directory, and restart the font server.

If you're rolling your own font collection, you'll need to supply the files `fonts.dir` and `fonts.scale`; the tool to make them is `type1inst`, available from http://http://goblet.anu.edu.au/~m9305357/type1inst.html.

As for the True Type fonts, group them in a directory of your choice and create `fonts.dir` using `ttmkfdir > fonts.dir`, included in the Freetype archive; http://www.freetype.org. Then proceed as above. For example, if you want to use the Windows fonts you have in, say, /mnt/win/windows/fonts, go to that directory, run `ttmkfdir`, edit /etc/X11/fs/config and restart the font server.

It all started from the original True Type X font server:
http://http://www.dcs.ed.ac.uk/home/jec/programs/xfsft/.

4.15 Users' Configurations

When you're done editing the dot files, copy them to /etc/skel as seen in Section 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 recent `gcc` versions, it may be advisable to put this in your `/etc/rpmrc`:

```
optflags: i386 -O2 -mpentiumpro
```

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 + Docs

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.

Documents on Linux configuration are popping up everywhere. One of the most complete is TrinityOS, http://www.ecst.csuchico.edu/~dranch/LINUX/index−linux.html. Bug the author to have him turn the document in nicer formats.

A jolly good page is http://dotfiles.com. Just what it says − a collection of configuration files.

6.The End

6.1 Copyright

If you have questions, please refer to the Linux Documentation Project home page, http://www.linuxdoc.org

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

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.

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.

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

Enjoy,

Guido `=8-)`
