Some Mini-Howtos of Interest

Curro Perez-Bernal <francisco.perez@dfaie.uhu.es>

Version $Revision: 1.90 $ ($Date: 2015/03/13 13:20:02 $)

Abstract

A little collection of Mini-Howtos for GNU/Linux systems trying to help solving everyday problems I have found in different aspects, from system administration to basic graphic edition or text processing. The howtos were prepared having Debian systems in mind, but they should be useful for other GNU/Linux systems as well. You can get the latest version of this document from github (Mini Howto GEM (https://github.com/currix/Mini-Howtos-GEM)). Translations of all or part of this document to French (http://www.autoteiledirekt.de/science/certains-mini-howtos-dinteret) (by Kate Bondareva) is also available online.

Copyright Notice

Copyright 2013 by Curro Perez-Bernal <francisco.perez@dfaie.uhu.es>

This document may be used under the terms of the GNU Free Documentation License version 1.3 or higher. (http://www.gnu.org/licenses/fdl-1.3.html)

Contents

Chapter 1

Basic Administration

1.1 Reinstalling a wiped out GRUB

Sometimes, specially if an operating system other than GNU/Linux is installed, the Grub bootloader can be wiped out and the booting process broken. In order to fix this, we will first assume that we have the new version Grub2 installed. Afterwards we explain how to proceed for the older Grub versions.

1.1.1 Grub2 case.

Updated on January 19th, 2012.

The first step is to prepare or find a rescue Live-CD (or bootable Live-USB) so we can start our system. We should know which is the partition of the hard drive that is mounted as root partition in /. We will assume in this example that the HD where the system is installed is /dev/sda and the root partition is sda7. If we do not know this information we can retrieve it using fdisk and mount. Once the Live-CD has booted, we require to open a rescue session in the booting advanced options. We will be prompted for the root partition and we the rescue disk will open a shell with the given root partition mounted.

It is important the if the /var directory is in a partition by itself (e.g. sda8) it is also mounted as follows

```
# mount /dev/sda7 /var
```

We now proceed to run dpkg-reconfigure grub-pc, and Grub2 will be reconfigured and the problem fixed.

In some cases it may be necessary to reinstall Grub in a system running with a live CD. In this case you should first mount the root partition and then install Grub using grub-install

```
sudo grub-install /dev/sda --root-directory=/mnt
sudo reboot
```

In case that the boot menu may not be the same than the previous one, after rebooting you can update it

```
sudo update-grub
```

This last order is the command you need to run when you make changes in the Grub2 configuration. In this version the configuration file is not anymore in the /boot directory but in /etc: /etc/grub.cfg and /etc/grub.d/*.

1.1.2 Old Grub case.

In order to recover the old Grub version we proceed as in the previous case, booting the computer from a Live-CD or Live-USB and opening a terminal. In this terminal we launch Grub

```
$ sudo grub
```

Once in the `Grub` application we define the root filesystem with the command `root` *partition*. Remember that root syntax for disk and partitions is `hd`*disk*`,`*partition*, and it starts counting from zero. If, as in the previous case, we assume in this example that the HD where the system is installed is `/dev/sda` and the root partition is `sda7`. Once the root partition is defined we install grub in the disk whose MBR is going to be used (`sda` in this case) and we quit.

```
root (hd0,6)
setup (hd0)
quit
```

We can now reboot the computer normally. In order to recover the previous menu it could be necessary to, after rebooting, execute

```
sudo update-grub
```

1.1.3 References

1 http://sites.google.com/site/easylinuxtipsproject/grub#TOC-The-new-Grub-2-present-in-Ubunt

2 http://sites.google.com/site/easylinuxtipsproject/oldgrub

1.2 Problem with journalizing

There could be a problem with the journalizing if an `ext3` filesystem refuses to be mounted. In the following we suppose that we are mounting an `ext3` filesystem associated to */dev/sdb2* in mount-point */media/usb_disk/*.

```
$ mount /media/usb_disk/
mount: wrong fs type, bad option, bad superblock on /dev/sdb2,
       missing codepage or other error
       In some cases useful info is found in syslog - try
       dmesg | tail  or so
```

And in the `dmesg` file you will find something like

```
JBD: no valid journal superblock found
EXT3-fs: error loading journal.
```

In order to solve this problem first check if the filesystem can be mounted as `ext2`

```
# mount -t ext2 /dev/sdb2 /mnt/
# ls -a
.  ..  lost+found  misc
```

If, as in the example, it works, unmount it and recreate the journal:

```
# tune2fs -O ^has_journal /dev/sdb2
tune2fs 1.37 (21-Mar-2005)

# tune2fs  -j /dev/sdb2
tune2fs 1.37 (21-Mar-2005)
Creating journal inode: done
This filesystem will be automatically checked every 22 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

And now you can regularly mount the `ext3` filesystem.

1.3 Labels in `ext2` and `ext3` units

The use of labels is a convenient way to handle disk units, specially for *usb* units that can be plugged in and removed. Instead of taking care of the dynamically associated device name the label can unambiguously identify the unit.

The command used to display and change the label name for `ext2` and `ext3` filesystems is `e2label`. The syntax of the command is such that to change or define a label named *newlabel* for unit *device*, the order issued is `e2label` *device newlabel*. For example

```
# e2label /dev/sdb2 usb_disk
```

To display the label name the command is used with the device name as an argument.

```
# e2label /dev/sdb2
usb_disk
```

To define the corresponding `fstab` entry the following line is added to the file `/etc/fstab`

```
LABEL=usb_disk /media/usb_disk  ext3    user,noauto     0       0
```

Note that the last option should be zero for removable media because if this is not the case, the booting process will be affected if the disk is not plugged in. The corresponding mount point should be defined:

```
# mkdir /media/usb_disk
```

Now you can regularly mount the `ext3` filesystem.

```
$ mount /media/usb_disk
```

1.4 Open a `xconsole` as normal user

First the existence and permissions of the file `/dev/xconsole` has to be checked

```
ls -l /dev/xconsole
prw-r-----  1 root adm 0 2006-05-02 12:40 /dev/xconsole
```

The user who is going to open the `xconsole` in his window manager (e.g. *bob*) has to be added to the `adm` group.

```
adduser bob adm
Adding user 'bob' to group 'adm' ...
Done.
```

The xconsole program can be now normally launched

```
xconsole -file /dev/xconsole
```

1.5 Resetting the `root` password

The first and simplest option is to boot in `single-user` mode. To do so, if the bootloader is `GRUB` stop the booting process and edit the line where the kernel is selected adding `1` at the end of the line. The system should boot to a `root` prompt and the password can be changed using the `passwd` command.

Another possibility is to boot using a rescue disk and once that you are at the command prompt mount the system's root directory if it has not been already mounted. For example, let's assume that the system's root partition is `/dev/sda1` and will be mounted in `/mnt/sysrootdir`

```
mkdir /mnt/sysrootdir
mount /dev/sda1 /mnt/sysrootdir
```

Then, after switching to the system's root directory with `chroot`, reset the password.

```
chroot /mnt/sysrootdir
passwd
```

1.5.1 References

1 Tech tip in http://www.linuxjournal.com/issue/180 (http://www.linuxjournal.com/issue/180)

1.6 Configure `exim` to use `SMTP-TLS`

The following instructions explain in a short and direct way the minimal steps necessary to include `SMTP-TLS` support in `exim` to connect with a *smarthost* node relaying mail. The data provided are appropriate for connecting with the server `mailgw.uhu.es` and for a box with `Debian Lenny`.

The necessary steps are the following

1 Install packages `exim4-base`, `exim4-config`, and `exim4-daemon-light`.

```
[root@localhost ~]# apt-get install exim4-base exim4-config \
exim4-daemon-light
```

2 If during the `exim4` installation no question is asked or if the packages are already installed in the computer, reconfigure the mail agent:

```
dpkg-reconfigure exim4-config
```

The following answers apply to configure a box in CLGEM-UHU:

1 `mail sent by smarthost; received via SMTP or fetchmail`
2 `local mail name` *boxname.dfa.uhu.es*
3 `IP-addresses to listen on for incoming SMTP connections:` *127.0.0.1*
4 `Other destinations for which mail is accepted:` *[Blank]*
5 `Machines to relay mail for:` *[Blank]*
6 `IP address or host name of the outgoing smarthost:` *mailgw.uhu.es*
7 `Hide local mail name in outgoing mail?` *Yes*
8 `Visible domain name for local users:` *[Blank]*
9 `Keep number of DNS-queries minimal (Dial-on-Demand)?` *No*
10 `Delivery method for local mail:` *mbox format in /var/mail/*
11 `Split configuration into small files?` *Yes*

3 The next step is the generation of the necessary keys.

```
# cd /etc/exim4/
# openssl req -x509 -newkey rsa:1024 -keyout rsa.key -out rsa.cert \
-days 9999 -nodes
# openssl dhparam -out dh.key 1024
```

4 Enable `TLS` in `exim`'s configuration editing the file
`/etc/exim4/conf.d/main/03_exim4-config_tlsoptions`
and adding the following lines

```
tls_certificate = /etc/exim4/rsa.cert
tls_privatekey = /etc/exim4/rsa.key
tls_dhparam = /etc/exim4/dh.key
```

5 Edit the user authentication info on file `/etc/exim4/passwd.client`[1]. For example, for user `coco.elmo.dfaie` with LDAP password `frdg098r`

```
# password file used when the local exim is authenticating to a remote
# host as a client.
#
# see exim4_passwd_client(5) for more documentation
#
# Example:
### target.mail.server.example:login:password
mailgw.uhu.es:coco.elmo.dfaie:frdg098r
```

6 Restart the `exim`'s daemon.

```
# /etc/init.d/exim4 restart
```

[1] There should be a way to do this from a user's perspective and not editing a general file for the system (TO DO).

1.6.1 References

1 http://www.amk.ca/diary/2003/03/enabling_smtptls_with_exim.html

1.7 Compile and install the LAPACK95 interface driver routine.

Updated on September 16th, 2011.

This howto has been checked in a Debian Squeeze box, with gfortran 4.4.5 compiler and in a Ubuntu Lucid Lynx with compiler gfortran 4.4.3.

In this howto we install from scratch the Fortran 77 libraries BLAS and LAPACK, and then compile the FORTRAN 95 extension to LAPACK. If the system already has working BLAS and LAPACK libraries this first step could be skipped. In any case it is interesting to compile all the libraries with optimized flags for the system in question.

First we should upload the LAPACK and LAPACK 95 libraries, for example from the *Netlib* website (LAPACK tgz from Netlib (http://www.netlib.org/lapack/lapack-3.3.1.tgz), (LAPACK95 tgz from Netlib (http://www.netlib.org/lapack95/lapack95.tgz)). The LAPACK version we will install is 3.3.1.

```
$ wget  -c http://www.netlib.org/lapack/lapack-3.3.1.tgz
--2011-09-16 13:49:37--  http://www.netlib.org/lapack/lapack-3.3.1.tgz
Resolving www.netlib.org... 160.36.58.108
.
.
.
2011-09-16 13:49:48 (439 KB/s) - 'lapack-3.3.1.tgz' saved [4945204/4945204]

$ wget http://www.netlib.org/lapack95/lapack95.tgz
--2011-09-16 13:49:54--  http://www.netlib.org/lapack95/lapack95.tgz
.
.
.
2011-09-16 13:49:59 (368 KB/s) - 'lapack95.tgz' saved [1579613/1579613]
```

Untar the LAPACK library tarball.

```
$ tar xzf lapack-3.3.1.tgz
$ cd lapack-3.3.1/
lapack-3.3.1$
```

Edit the file make.inc to a file conveniently tuned for your system. A working example for gfortran is shown in 'make.inc for LAPACK compilation' on the following page. The library can now be compiled, starting with the compilation of the included BLAS library

```
lapack-3.3.1$ cd BLAS/SRC
lapack-3.3.1/BLAS/SRC$ make
.
.
.
 zhemm.o zherk.o zher2k.o lsame.o xerbla.o xerbla_array.o
ranlib ../../blas_linux_gfortran.a
lapack-3.3.1/BLAS/SRC$
```

The next step is the compilation of the LAPACK library

```
lapack-3.3.1/BLAS/SRC$ cd ../../
lapack-3.3.1$ make
.
.
.
make[1]: Leaving directory '/home/curro/1/lapack-3.3.1/BLAS/TESTING'
( cd BLAS; ./xblat3s < sblat3.in     ; \
            ./xblat3d < dblat3.in     ; \
            ./xblat3c < cblat3.in     ; \
            ./xblat3z < zblat3.in     )
lapack-3.3.1$ ls *.a
blas_linux_gfortran.a  lapack_linux_gfortran.a  tmglib_linux_gfortran.a
```

The next step is to copy the compiled libraries to their final destination

```
lapack-3.3.1$ sudo mkdir /usr/local/lapack-3.3.1
[sudo] password for curro:
lapack-3.3.1$ sudo cp *.a /usr/local/lapack-3.3.1/
```

We now proceed to compile and install the LAPACK 95 library. First we unpack the tarball and edit the make.inc file. A working example for gfortran is shown in 'make.inc for LAPACK95 compilation' on the next page.

```
$ tar xzf lapack95.tgz
$ cd LAPACK95/
LAPACK95$ cd SRC
LAPACK95/SRC$ make single_double_complex_dcomplex
.
.
.
ranlib ../lapack95.a
rm -fr ../lapack95_modules
mkdir ../lapack95_modules
'cp' *.mod ../lapack95_modules/
rm -f f77_lapack.* f95_lapack.*
rm -f *_lapack_single_double_complex_dcomplex.o
LAPACK95/SRC$
```

The final step is to copy the libfile and modules to a convenient location.

```
LAPACK95$ sudo mkdir /usr/local/lib/lapack95
LAPACK95$ sudo cp -r lapack95.a lapack95_modules /usr/local/lapack95
```

We include a program template invoking this library in 'Example of program using LAPACK95' on the facing page and a makefile that can be used to compile this program in 'makefile for compiling programs with calls to LAPACK95' on page 8.

1.7.1 make.inc for LAPACK compilation

```
# -*- Makefile -*-
####################################################################
#  LAPACK make include file.                                       #
#  LAPACK, Version 3.3.1                                           #
#  April 2011                                                      #
####################################################################
#
# See the INSTALL/ directory for more examples.
#
SHELL = /bin/sh
#
#  The machine (platform) identifier to append to the library names
#
PLAT = _linux_gfortran
#
#  Modify the FORTRAN and OPTS definitions to refer to the
#  compiler and desired compiler options for your machine.  NOOPT
#  refers to the compiler options desired when NO OPTIMIZATION is
#  selected.  Define LOADER and LOADOPTS to refer to the loader
#  and desired load options for your machine.
#
FORTRAN  = gfortran -O2 -m32
OPTS     =
DRVOPTS  = $(OPTS)
NOOPT    = -g -O0
LOADER   = gfortran -g
LOADOPTS =
#
# Timer for the SECOND and DSECND routines
#
# Default : SECOND and DSECND will use a call to the EXTERNAL FUNCTION ETIME
# TIMER    = EXT_ETIME
# For RS6K : SECOND and DSECND will use a call to the EXTERNAL FUNCTION ETIME_
# TIMER    = EXT_ETIME_
# For gfortran compiler: SECOND and DSECND will use a call to the INTERNAL FUNCTION ETIME
TIMER    = INT_ETIME
# If your Fortran compiler does not provide etime (like Nag Fortran Compiler, etc...)
# SECOND and DSECND will use a call to the Fortran standard INTERNAL FUNCTION CPU_TIME
# TIMER    = INT_CPU_TIME
# If neither of this works...you can use the NONE value... In that case, SECOND and DSECND will always return 0
# TIMER     = NONE
#
#  The archiver and the flag(s) to use when building archive (library)
#  If you system has no ranlib, set RANLIB = echo.
#
ARCH     = ar
ARCHFLAGS= cr
RANLIB   = ranlib
#
#  The location of BLAS library for linking the testing programs.
#  The target's machine-specific, optimized BLAS library should be
#  used whenever possible.
#
BLASLIB      = ../../blas$(PLAT).a
#
```

```
#  Location of the extended-precision BLAS (XBLAS) Fortran library
#  used for building and testing extended-precision routines.  The
#  relevant routines will be compiled and XBLAS will be linked only if
#  USEXBLAS is defined.
#
# USEXBLAS    = Yes
XBLASLIB     =
# XBLASLIB    = -lxblas
#
#  Names of generated libraries.
#
LAPACKLIB    = lapack$(PLAT).a
TMGLIB       = tmglib$(PLAT).a
EIGSRCLIB    = eigsrc$(PLAT).a
LINSRCLIB    = linsrc$(PLAT).a
```

1.7.2 `make.inc` for LAPACK95 compilation

```
#
#  -- LAPACK95 interface driver routine (version 2.0) --
#     UNI-C, Denmark; Univ. of Tennessee, USA; NAG Ltd., UK
#     August 5, 2000
#
FC  = gfortran
FC1      = gfortran
# -dcfuns  Enable recognition of non-standard double
#          precision  complex intrinsic functions
# -dusty   Allows the compilation and execution of "legacy"
#          software by downgrading the category  of  common
#          errors  found  in  such software from "Error" to
# -ieee=full enables all IEEE arithmetic facilities
#          including non-stop arithmetic.

OPTS0    = -O2 -m32
MODLIB   = -I./../lapack95_modules
OPTS1    = -c $(OPTS0)
OPTS3    = $(OPTS1) $(MODLIB)
OPTL     = -o
OPTLIB   =

LAPACK_PATH = /usr/local/lapack-3.3.1/

LAPACK95 = ../lapack95.a
LAPACK77 = $(LAPACK_PATH)/lapack_DEBSQUEEZE_DECKARD.a
TMG77    = $(LAPACK_PATH)/tmglib_DEBSQUEEZE_DECKARD.a
BLAS     = $(LAPACK_PATH)/blas_DEBSQUEEZE_DECKARD.a

LIBS     = $(LAPACK95) $(TMG77) $(LAPACK77) $(BLAS)
SUF      = f90

XX = 'rm' -f $@; \
        'rm' -f $@.res; \
 $(FC) $(OPTS0) -o $@ $(MODLIB) $@.$(SUF) $(OPTLIB) $(LIBS); \
        $@ < $@.dat > $@.res; \
        'rm' -f $@

YY = $(FC) $(OPTS0) -o $@ $(MODLIB) $@.$(SUF) $(OPTLIB) $(LIBS)

.SUFFIXES: .f90 .f .o

.$(SUF).o:
 $(FC) $(OPTS3) $<

.f.o:
 $(FC1) $(OPTS3) $<
```

1.7.3 Example of program using LAPACK95

```
PROGRAM LA_SSPSV_EXAMPLE

  !  -- LAPACK95 EXAMPLE DRIVER ROUTINE (VERSION 1.0) --
  !     UNI-C, DENMARK
  !     DECEMBER, 1999
  !
  !  .. "Use Statements"
  USE LA_PRECISION, ONLY: WP => SP
  USE F95_LAPACK, ONLY: LA_SPSV
  !  .. "Implicit Statement" ..
  IMPLICIT NONE
  !  .. "Local Scalars" ..
  INTEGER :: I, N, NN, NRHS
  !  .. "Local Arrays" ..
  INTEGER, ALLOCATABLE :: IPIV(:)
  REAL(WP), ALLOCATABLE :: B(:,:), AP(:)
  !  .. "Executable Statements" ..
  WRITE (*,*) 'SSPSV Example Program Results.'
  N = 5; NRHS = 1
```

```
  WRITE(*,'(5H N = , I4, 9H; NRHS = , I4)') N, NRHS
  NN = N*(N+1)/2
  ALLOCATE ( AP(NN), B(N,NRHS), IPIV(N) )
  !
  OPEN(UNIT=21,FILE='spsv.ma',STATUS='UNKNOWN')
  DO I=1,NN
     READ(21,'(F3.0)') AP(I)
  ENDDO
  CLOSE(21)
  !
  WRITE(*,*)'Matrix AP :'
  DO I=1,NN; WRITE(*,"(15(I3,1X,1X),I3,1X))") INT(AP(I));
  ENDDO
  !
  OPEN(UNIT=21,FILE='spsv.mb',STATUS='UNKNOWN')
  DO I=1,N
     READ(21,'(F3.0)') B(I,1)
  ENDDO
  CLOSE(21)
  !
  WRITE(*,*)'Matrix B :'
  DO I=1,N; WRITE(*,"(10(I3,1X,1X),I3,1X)')") INT(B(I,1));
  ENDDO
  !
  WRITE(*,*)" CALL LA_SPSV( AP, B, 'L', IPIV )"
  !
  CALL LA_SPSV( AP, B, 'L', IPIV )
  !
  WRITE(*,*)'AP on exit: '
  DO I=1,NN; WRITE(*,"(15(E13.5))") AP(I);
  ENDDO
  !
  WRITE(*,*)'Matrix B on exit :'
  DO I=1,N; WRITE(*,"(F9.5)") B(I,1);
  ENDDO
  WRITE(*,*)'IPIV = ', IPIV
  !
END PROGRAM LA_SSPSV_EXAMPLE
```

1.7.4 `makefile` for compiling programs with calls to `LAPACK95`

```
#
#  -- LAPACK95 makefile (version 1.0) --
#
FC  = gfortran
#
MODLIB   = -I/usr/local/lib/lapack95_modules
OPTS1    = -c
OPTS3    = $(OPTS1) $(MODLIB)
OPTL     = -o
OPTLIB   = -lblas -llapack

LAPACK_PATH = /usr/local/lib
LAPACK95_PATH = /usr/local/lib

LAPACK95 = $(LAPACK95_PATH)/lapack95.a

LIBS     = $(LAPACK95)
SUF      = f90

YY = $(FC) -o $@ $(MODLIB) $@.$(SUF) $(OPTLIB) $(LIBS)

.SUFFIXES: .f90 .f .o

.$(SUF).o:
 $(FC) $(OPTS3) $<

ejemplo_la_spsv:
 $(YY)

clean:
 'rm' -f *.o *.mod core
```

1.8 Compile and link statically with `NAG` and `LAPACK`

It is important to be able to compile and link statically programs when libraries are not available in all nodes. This is the case with the `NAG` library which is not compatible with `gfortran`, the only `Fortran` compiler in `Debian Lenny`. The program statically linked in one node (where `Etch` is installed and `g77` is available can the be executed in any other node.

An example of compilation is the following

```
g77 -static -o infsq_box_1D  infsq_box_1D.f ../Potentials/wsaxon_Box_pot.f -L /usr/lib/atlas -L/usr/local2/NAG -lnag  -llapack -lblas
```

Some important points:

1 It is necessary to include both `Lapack` and `Blas` libraries.

2 The `Blas` library should be `blas-3`.

3 The use of standard `Lapack` and `Blas` libraries give an error due to the different sizes of object files. Something like

```
(xerbla.o): In function 'xerbla_': multiple definition of 'xerbla_'
/usr/lib/liblapack.a(xerbla.o): first defined here
/usr/bin/ld: Warning: size of symbol 'xerbla_' changed from 86
in /usr/lib/liblapack.a(xerbla.o) to 38
in /usr/lib/libblas.a(xerbla.o)
collect2: ld returned 1 exit status
```

This is a known bug and can be solved using the libraries provided with the `Atlas` packages and adding the corresponding path to the compilation: `-L /usr/lib/atlas`.

4 The order of the libraries is not irrelevant. In particular I found that `lapack` has to be invoked prior to `blas-3`.

1.9 Copying `CUPS` configuration from one server to another

The simplest way to copy the `CUPS` configuration from one server to another is the following:

1 Stop CUPS on the target system.

```
target# /etc/init.d/cups stop
```

2 Rename or backup the existing configuration directory

```
target# mv /etc/cups /etc/cups.orig
```

3 Copy the /etc/cups directory from the source system to the destination system.

```
target# scp -r source:/etc/cups /etc/cups
```

4 Copy any modified model files from the source system to the destination system. These files should be in `/usr/share/cups/model`.

```
target# scp -r source:/usr/share/cups/model /usr/share/cups/model
```

5 On the target server edit the file `/etc/cups/cupsd.conf` and check if the hostname or IP address of your source system is present. If so, change it to the target server hostname or IP. Check that hostnames are defined properly in the `/etc/hosts` file of the target system.

6 If any custom groups or accounts are used on the old system to manage `CUPS` recreate them on the new system.

7 Restart cups in the target system and test it.

1.9.1 References

1 Tech tip in My Scripts and Tips (http://www.tipsandscripts.net/archives/19)

1.10 Mapping network interfaces to a fixed name.

As modules on the booting process or during the system's activity are not loaded always in the same order, it is difficult to get the network devices named in a constant way. This can be achieved using the package `ifrename`. This software maps the interfaces to a fixed name based on the MAC address (for example). You just need to add the file `/etc/iftab`:

```
# Mapping NICs to fixed names
ether0  driver 8139too mac XX:XX:XX:XX:XX:XX
wifi0  driver ipw2200 mac XX:XX:XX:XX:XX:XX
```

Another way to get the same result especially prone to `Debian` systems is to include an addition of a `udev` rule. In order to do so, create the file `/etc/udev/rules.d/net.local.rules`, and populate it with the following:

```
KERNEL=="eth*", SYSFS{address}=="00:01:80:50:dc:f2", NAME="wired"
KERNEL=="eth*", SYSFS{address}=="00:0d:61:a1:20:15", NAME="wireless"
```

changing the `MAC`'s addresses and `NAME`'s accordingly.

1.11 Using labels to mount disk partitions

Using the possibility of labelling the disk partitions it is useful for avoiding ambiguities and errors mounting them. This is valid for `ext2`, `ext3`, and `ext4` filesystems. To name *usbdisk0* the second partition of a disk which is currently `/dev/sdb` we should do as follows[2]

```
# e2label /dev/sdb2 usbdisk0
```

There are other tools to label partitions for different filesystems.

1 `ext2 ext3 ext4`: `e2label`

2 `FAT16 FAT32`: `mtools`

3 `jfs`: `jfstune`

4 `NTFS`: `ntfsprogs`

Then, once the partition is labelled, we create the mount point, e.g. `mkdir /media/usb_disk_0`, and we can add a line in `/etc/fstab` like the following

```
LABEL=usbdisk0       /media/usb_disk_0 ext3    exec,user,noauto  0       0
```

When we use the command `mount /media/usb_disk_0` the previously labelled partition will be mounted.

1.12 Using `find` and `xargs`

Updated on August 16th, 2013.

The combination of the utilities `find` and `xargs` is a powerful tool when you need to apply a program or do a task with several files.

As an example let's imagine that we have a directory with many `LaTeX` files, and some of them are letters, named as `lett_`*name*`.tex` and we should transform some of them into pdf files. A simple `perl` oneliner that can do the task of transforming a `tex` file into a `pdf` file is the following.

```
perl -e '(my $name= $ARGV[0])=~s/\.tex//; system "latex $name;dvipdf $name"' file.tex
```

[2] It is also possible to rename the partition using a graphical interface such as `gparted`.

How to select the LaTeX files with different and flexible criteria and apply this program to all of them? A possible answer is to combine the powerful find and xargs programs.

Let's suppose that we want to transform to pdf format all the files having names starting as lett_. Then we can execute

```
$ find . -name "lett_*.tex" -print
./lett_diput_Huelva.tex
./lett_Hospital_IE.tex
./lett_CEPSA.tex
./lett_del_JA_Huelva.tex
./lett_audiencia_Huelva.tex
./lett_subdel_gob_Huelva.tex
./lett_INNOV.tex
./lett_ayto_Huelva.tex
./lett_Hospital_JRJ.tex
./lett_ayto_Palos.tex
```

This command finds all the files starting with lett_ and finishing as .tex and print its names. The option -iname makes a case insensitive search. Other interesting options are -amin n and -atime n where n is the number of minutes or hours that have passed since the last time the file was accessed. If the number is negative the effect is the contrary and look for files accessed prior to this time.

The program xargs can be combined with find. This program in its simplest form takes lines of input and apply programs to them. The simplest use is xargs -I {} *run_command* {}. In this form xargs read lines from the standard input and apply to each of them the command *run_command* substituting {} by the input line. Instead of the standard input the option -a *filename* permits xargs to get its input from a file rather that the standard input. We can combine the two command as follows

```
$ find . -iname "lett*.tex" -print | xargs -I {} perl -e '(my $name= $ARGV[0])=~s/\.tex//; system "latex $name;dvipdf $name"' {}
```

In multiprocessor computers we can use the interesting option -P *n*, which makes xargs to run *n* number of commands in parallel.

Another interesting option of find is -newer filename. With this option the program displays the names of the files that are more recent than the file filename. Using this in combination with xargs we can for example, transfer using scp all the files in a directory that are more recent that a given file. If we want to upload to a server called destination.org all files in a directory that are more recent than a file called fulltext.pdf we shoud execute

```
$ find ./* -newer fulltext.pdf -print | xargs -I {} scp {} user@destination.org:FilesPath
```

The following case is a practical example that shows how useful is the combination of pipes in bash, making also use of the xargs command.

Let's assume that we are copying into a directory a backup of the home folders of several users. Then, by mistake, instead of the directory of a user called tigu, we copied the contents of his directory to the backup directory, wreaking some havoc and transforming

```
user1 user2 user3  ...   tigu
```

into

```
user1 user2 user3  ...   tigu and all tigu files and directories
```

A quick solution, avoiding the selection and deletion by hand of each of the misplaced files and directories, consists in the combination with pipes of several commands

```
$ find . -maxdepth 1 -user tigu | grep -v tigu | xargs rm -r
```

The first command output is a list (non-recursive) of all files and directories in the current directory (assuming that the current directory is where the backup takes place) belonging to user tigu.

The first pipe *pipes* the result of this search to the grep command, where the directory named tigu. This is done to prevent the command from erasing the original user home directory.

Finally the output of grep is piped to a combination of xargs and rm that removes the files. For more details in the use of xargs see the references below.

Several commands can be launched by xargs. For example, let's assume that we want to check the differences between files in two different directories, and we want to know the file it is being tested each time. The way to accomplish this, combining an echo statement with the diff statement using xargs is the following

```
$ ls -1 *f90 | xargs -I % sh -c 'echo %; diff % /users/home/laura/prototipo_1.0/fortran/1D/1body/pseudostates/src/%;' | less
```

In this case we are also using a different character as a dummy variable for `xargs` with the option `-I %`.

A last example of several pipes and `xargs` is the following, where we are reconstructing using `rdiff-backup` the accounts of a series of users, excepting user `laura`, from a directory `/home_backup/`*`username`* in a node called `backup_server` to the directory `/home` in the node `server`

```
ssh root@backup_server ls -1 /home_backup | grep -v laura | xargs -I % sh -c "rdiff-backup -v4 --restore-as-of now  root@backup_serv
```

1.12.1 References

1 http://www.linuxjournal.com/article/10643

1.13 Preparing a USB bootable `Linux` installation device (Fast way).

Updated on October 6th, 2014.

We need a USB disk that is at least 256 MB in size with a `FAT16` partition. Normally the sticks come with a preconfigured single `FAT16` partition that is valid. If it is necessary to format the disk you can find instructions in 'Preparing a USB bootable `Debian` installation device.' on the facing page. Take also into account that *all data in the stick will be erased*. Plug in and mount the dist. Let's assume that the USB disk corresponds to the `/dev/sdb` device.

```
$ mount
/dev/sdb1 on /media/disk type vfat (rw,nosuid,nodev,uhelper=hal,uid=1001,shortname=winnt)
```

Download the `Debian` image. Take into account that the image should fit into the stick.

```
$ wget -c http://cdimage.debian.org/debian-cd/6.0.1a/i386/iso-cd/debian-6.0.1a-i386-netinst.iso
--2011-05-03 13:37:50--  http://cdimage.debian.org/debian-cd/6.0.1a/i386/iso-cd/debian-6.0.1a-i386-netinst.iso
Resolving cdimage.debian.org... 130.239.18.173, 130.239.18.163, 2001:6b0:e:2018::163, ...
.
.
.
.
Saving to: 'debian-6.0.1a-i386-netinst.iso'

100%[======================================================================>] 198,064,128 1.54M/s   in 98s

2011-05-03 13:39:35 (1.92 MB/s) - 'debian-6.0.1a-i386-netinst.iso' saved [198064128/198064128]
```

The CD or DVD image you choose should be written directly to the USB stick, overwriting its current contents with the command `dd if=`*`file`* `of=`*`device`* `bs=4M; sync`. If the pendrive device is `/dev/sdb` and the iso image path is `/media/MSDOS/FSTS/ubuntu-14.04.1-desktop-amd64.iso` the command should be

```
# dd if=/media/MSDOS/FSTS/ubuntu-14.04.1-desktop-amd64.iso of=/dev/sdb bs=4M; sync
245+1 records in
245+1 records out
1028653056 bytes (1.0 GB) copied, 22.5565 s, 45.6 MB/s
```

And the stick is ready.

1.13.1 References

1 http://www.debian.org/releases/stable/i386/ch04s03.html.en

1.14 Preparing a USB bootable `Debian` installation device.

The first step is the creation of a `FAT16` partition and filesystem in the USB disk. Suppose that the USB disk corresponds to the `/dev/sdb` device. Then, as root we format and include the filesystem.

```
# fdisk /dev/sdb


Disk /dev/sdb: 1 GB, 1997649920 bytes
62 heads, 62 sectors/track, 1015 cylinders
Units = cylinders of 3844 * 512 = 1968128 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1   *           1        1015     1950799    6  FAT16


# mkdosfs /dev/sdb1
mkdosfs 3.0.1 (23 Nov 2008)
```

We now proceed to install a bootloader, as `syslinux`, `grub`, or `lilo`. We include `syslinux`[3]

```
# syslinux /dev/sdb1
#
```

We now add the installer image, mounting the stick and copying the files `vmlinuz` (kernel binary) and `initrd.gz` (initial ramdisk image) to the stick. Both files can be found under the directory `/debian/dists/lenny/main/installer-i386/current/images/hd-media/` of the distribution.

```
# mount /dev/sdb1 /mnt
# ls /mnt
ldlinux.sys
#
# cd /mnt
/mnt# wget http://http.us.debian.org/debian/dists/lenny/main/installer-i386/current/images/hd-media/vmlinuz
--2010-03-12 16:16:23--  http://http.us.debian.org/debian/dists/lenny/main/installer-i386/current/images/hd-media/vmlinuz
Resolving http.us.debian.org... 149.20.20.135, 204.152.191.39, 35.9.37.225, ...
Connecting to http.us.debian.org|149.20.20.135|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1468976 (1.4M) [text/plain]
Saving to: 'vmlinuz'

100%[======================================>] 1,468,976    624K/s   in 2.3s

2010-03-12 16:16:26 (624 KB/s) - 'vmlinuz' saved [1468976/1468976]

deckard:/mnt# wget http://http.us.debian.org/debian/dists/lenny/main/installer-i386/current/images/hd-media/initrd.gz
--2010-03-12 16:16:37--  http://http.us.debian.org/debian/dists/lenny/main/installer-i386/current/images/hd-media/initrd.gz
Resolving http.us.debian.org... 128.30.2.36, 149.20.20.135, 204.152.191.39, ...
Connecting to http.us.debian.org|128.30.2.36|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5183930 (4.9M) [application/x-gzip]
Saving to: 'initrd.gz'

100%[======================================>] 5,183,930   1.42M/s   in 3.5s

2010-03-12 16:16:40 (1.42 MB/s) - 'initrd.gz' saved [5183930/5183930]
```

Now we should create a `syslinux.cfg` configuration file, which at a bare minimum should contain the following two lines:

```
default vmlinuz
append initrd=initrd.gz
```

We finally copy a Debian ISO image (businesscard, netinst or full CD image; be sure to select one that fits) onto the stick and unmount the USB memory stick.

1.14.1 References

1 http://www.debian.org/releases/stable/i386/ch04s03.html.en

[3] Please, note that the packages `syslinux` and `mtools` have to be installed to be able to run `syslinux`.

1.15 Upgrading `Flashplayer` in `Debian Lenny` using *backports*.

We start adding *Debian Backports*[4] to `/etc/apt/sources.list`. Be warned: once backports is active he resulting system *can't be considered plain "stable" anymore.*

```
# backports Added by Currix 28/04/10
deb http://www.backports.org/debian lenny-backports main contrib non-free
```

You then proceed to run `apt-get update`. There would probably be an error due to the lack of `backports.org` archive's key.

```
# apt-get update

...

Reading package lists... Done
W: GPG error: http://www.backports.org lenny-backports Release: The following signatures couldn't be verified because the public key
W: You may want to run apt-get update to correct these problems
```

In order to correct this we can now install the public key from `backports.org`.

```
# apt-get install debian-backports-keyring
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libsilc-1.1-2 libhesiod0 libzephyr3
Use 'apt-get autoremove' to remove them.
The following NEW packages will be installed:
  debian-backports-keyring
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 3362B of archives.
After this operation, 49.2kB of additional disk space will be used.
WARNING: The following packages cannot be authenticated!
  debian-backports-keyring
Install these packages without verification [y/N]? y
Get:1 http://www.backports.org lenny-backports/main debian-backports-keyring 2009.02.20 [3362B]
Fetched 3362B in 0s (29.2kB/s)
Selecting previously deselected package debian-backports-keyring.
(Reading database ... 219559 files and directories currently installed.)
Unpacking debian-backports-keyring (from .../debian-backports-keyring_2009.02.20_all.deb) ...
Setting up debian-backports-keyring (2009.02.20) ...
OK
```

We can now run `apt-get update` without any warning or error. Backport packages are deactivated by default (i.e. the packages are pinned to 1 by using NotAutomatic: yes in the Release files, just as in experimental). If you want to install a package from backports run:

```
apt-get -t lenny-backports install package
```

or

```
aptitude -t lenny-backports install package
```

In order to update to `Flash Player 10` first you should install the package `flashplugin-nonfree`

```
# apt-get -t lenny-backports install flashplugin-nonfree

   .
   .
   .

2010-04-28 09:46:20 (450 KB/s) - `./install_flash_player_10_linux.tar.gz' saved [4050435/4050435]

#
```

[4] Backports are recompiled packages from testing (mostly) and unstable (in a few cases only, e.g. security updates), so they will run without new libraries (wherever it is possible) on a stable Debian distribution. I recommend you to pick out single backports which fits your needs, and not to use all backports available here (from http://www.backports.org/dokuwiki/doku.php).

Then update the Flash Player with the command

```
# /usr/sbin/update-flashplugin-nonfree --install
#
```

To check if the plugin has been recognized by iceweasel type

```
about:plugins
```

in the browser address line. If there are previous versions of the plugin installed they can interfere. Uninstall them removing from ~/.mozilla/plugins the corresponding file.

1.15.1 References

1 http://backports.org/dokuwiki/doku.php?id=instructions

2 http://wiki.debian.org/FlashPlayer

3 http://plugindoc.mozdev.org/

1.16 Passive network exploration with p0f.

This utility uses passive techniques to try to guess the what machines exist on a network and properties of these computers. By default the application only listens to packets addressed to the machine executing the application

```
# p0f
p0f - passive os fingerprinting utility, version 2.0.8
(C) M. Zalewski <lcamtuf@dione.cc>, W. Stearns <wstearns@pobox.com>
p0f: listening (SYN) on 'eth0', 262 sigs (14 generic, cksum 0F1F5CA2), rule: 'all'.
```

To look at all the packets, not only the packets addressed to the machine where p0f is running, the interface should be in promiscuous mode, using the -p option. Be aware that this can cause a large capture data rate.

Using the -s option, the application can work with tcpdump data logs. The -w option allows to save network traffic data in tcpdump format.

Other interesting options are -M to activate the masquerade detection algorithm to try to identify hosts behind a NAT, and -l to format the output in a one-line style that is easier to grep.

1.17 Preparing a bootable flashdisk with Billix.

We start downloading billix from its *sourceforge* page. The link can be found in 'References' on the next page. In the present example the downloaded file name is billix-0.27.tar.gz. We need a usb flashdisk of a minimum size of 256 MB and with a FAT or FAT32 filesystem. In the present example is the device /dev/sdb1 mounted in /media/disk.

```
$ mount | grep disk
/dev/sdb1 on /media/disk type vfat (rw,nosuid,nodev,uhelper=hal,shortname=winnt,uid=1001)
```

The downloaded file is untarred in the flashdisk

```
$ cd /media/disk
/media/disk$ tar xzf ~/Downloads/billix-0.27.tar.gz
```

The second step is the installation of the MBR (Master Boot Record) in the flahsdisk. [5] This has to be executed as superuser and the syntax is install-mbr -p1 *device* and it is important to select the correct device, as this can wreak havoc if executed in an incorrect device.

[5] In Debian systems it is necessary to install the mbr package to have access to this tool.

```
# install-mbr -p1 /dev/sdb
```

We can now install the bootsector within the first partition running `syslinux -s device/partition`. Again this is a potentially dangerous operation and should never be performed in a wrong partition. Superuser privileges are not necessary to perform this operation.

```
/media/disk$ syslinux -s /dev/sdb1
```

The bootable flash disk is now ready and can be used as nice help for the system administrator. If there is spare space in the flash disk can be used normally for file storing.

1.17.1 References

1 Billix project homepage (http://sourceforge.net/projects/billix/)

1.18 Recovering a console that is unusable.

Sometimes it is convenient to clean the display of a text console, something that can be done using the shell command `clean`. For more extreme cases, when the console has gone totally berserk, e.g. after displaying binary characters, the initial state can be recovered using the command `reset`. It is possible that you will not be able to see the command while you type, but after its execution the console should return to a saner state.

1.19 Simple configuration of `sudo`

A very quick and dirty configuration of `sudo` in order to allow a user, e.g. *tuxie*, to execute programs with superuser privileges can be done adding one line to the `sudoers` file:

```
# echo "tuxie  ALL=(ALL) ALL" >> /etc/sudoers
```

A more detailed configuration is required in a sensitive environment.

1.20 Check the groups to which a user belongs.

In order to check to what groups a user, e.g. *tuxie*, belongs the command `groups` can be used:

```
$ groups tuxie
tuxie adm dialout cdrom floppy audio video plugdev
```

1.21 Check the available system `locale` options and establish a default one.

In order to check what `locale` options are compiled and available in a `Debian` system the user should execute.

```
$ locale -a
C
en_IE
en_IE@euro
en_IE.iso88591
en_IE.iso885915@euro
en_IE.utf8
en_US
en_US.iso88591
es_ES
es_ES@euro
es_ES.iso88591
es_ES.iso885915@euro
es_ES.utf8
POSIX
spanish
```

The system's default locale in `Debian` systems is found in the `/etc/default/locale` file.

```
$ cat /etc/default/locale
LANG=en_IE.UTF-8
```

The `locale` setting when executing a program can be changed on the fly

```
$  LANG=en_IE.utf8 date
Wed Oct 27 11:54:44 CEST 2010
$  LANG=es_ES.utf8 date
miÃ© oct 27 11:54:52 CEST 2010
$
```

It is recommended to use always a `utf-8` locale as a standard. And to define it only setting the *$LANG* variable, and not a complicate set of *$LC_** variables. To define the default local for the system use the `dpkg-reconfigure locales` command.

1.22 Change the encoding of given files.

To change the encoding of a file is very useful the application `iconv`. This application transforms a file from a given coding system to the default system encoding or to a different encoding. The different available encoding schemes can be shown with the option `-l` or `--list`

```
$ iconv -l
The following list contain all the coded character sets known.  This does
not necessarily mean that all combinations of these names can be used for
the FROM and TO command line parameters.  One coded character set can be
listed with several different names (aliases).

  437, 500, 500V1, 850, 851, 852, 855, 856, 857, 860, 861, 862, 863, 864, 865,
  866, 866NAV, 869, 874, 904, 1026, 1046, 1047, 8859_1, 8859_2, 8859_3, 8859_4,
  8859_5, 8859_6, 8859_7, 8859_8, 8859_9, 10646-1:1993, 10646-1:1993/UCS4,

  ...

  ISO-2022-JP-2, ISO-2022-JP-3, ISO-2022-JP, ISO-2022-KR, ISO-8859-1,
  ISO-8859-2, ISO-8859-3, ISO-8859-4, ISO-8859-5, ISO-8859-6, ISO-8859-7,
  ISO-8859-8, ISO-8859-9, ISO-8859-9E, ISO-8859-10, ISO-8859-11, ISO-8859-13,
  ISO-8859-14, ISO-8859-15, ISO-8859-16, ISO-10646, ISO-10646/UCS2,
  ISO-10646/UCS4, ISO-10646/UTF-8, ISO-10646/UTF8, ISO-CELTIC, ISO-IR-4,

  ...

  UNICODEBIG, UNICODELITTLE, US-ASCII, US, UTF-7, UTF-8, UTF-16, UTF-16BE,
  UTF-16LE, UTF-32, UTF-32BE, UTF-32LE, UTF7, UTF8, UTF16, UTF16BE, UTF16LE,
  UTF32, UTF32BE, UTF32LE, VISCII, WCHAR_T, WIN-SAMI-2, WINBALTRIM,
  WINDOWS-31J, WINDOWS-874, WINDOWS-936, WINDOWS-1250, WINDOWS-1251,
  WINDOWS-1252, WINDOWS-1253, WINDOWS-1254, WINDOWS-1255, WINDOWS-1256,
  WINDOWS-1257, WINDOWS-1258, WINSAMI2, WS2, YU
```

For example, if the default encoding is `UTF-8` and we want to transform an `ISO-8859-1` file named *example* to this encoding we should run

```
$ file non_standard_ISO
non_standard_ISO: ISO-8859 text
$ file non_standard_ISO
non_standard_ISO: ISO-8859 text
curro@deckard:~/temp$ iconv -f ISO-8859-1 non_standard_ISO
Ã± ± ± ± ± ± ±
Ã¡Ã¡Ã¡Ã¡
Ã©Ã©Ã©
ÃÃ
Ã³
Ãº
```

If no output file name is given with the option `-o` the result is displayed in the standard output. Thus, to save the file as `non_standard_UTF` we can execute

```
$ iconv -f ISO-8859-1 non_standard_ISO -o non_standard_UTF
curro@deckard:~/temp$ file non_standard_UTF
non_standard_UTF: UTF-8 Unicode text
```

We can also transform to an encoding that is not the system's default one with the option `-t`

```
$ iconv -f ISO-8859-1 -t UNICODE non_standard_ISO -o non_standard_UNI
$ file non_standard_UNI
non_standard_UNI: Little-endian UTF-16 Unicode character data
```

1.23 Problem with the wireless and `nm`

Sometimes, specially after having tweaked with the network manually, the NetworkManager applet does not appear any more, or when it appears it informs that it does not manage the wireless networks.

In order to fix this, ensure that the file `/etc/network/interfaces` is like the following, commenting any other thing previously added:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback
```

and make sure to change the option `false` to `true` in the file `/etc/NetworkManager/NetworkManager.conf`

```
[ifupdown]
managed=true
```

With this changes, restarting the network should be all you need to solve the problem.

1.24 Updating a `Debian` box from `Lenny` to `Squeeze`

Added on May 19, 2011.

First edit the file `/etc/apt/sources.list` and replace any appearence of `lenny` to `squeeze`. It could be a good idea to backup your original file before, just in case... You just need to update and upgrade the system following this steps

```
sudo apt-get update
sudo apt-get install apt dpkg
sudo apt-get dist-upgrade
```

In the update from `Lenny` to `Squeeze` the following error message appears:

```
Unable to migrate to dependency-based boot system
 â
 â Tests have determined that problems in the boot system exist which prevent migration to dependency-based boot sequencing:
 â
 â insserv: warning: script 'K20scdate' missing LSB tags and overrides, insserv: warning: script 'S25libdevmapper1.02' missing LSB ta
 â warning: script 'scdate' missing LSB tags and overrides, insserv: warning: script 'libdevmapper1.02' missing LSB tags and override
 â but not purged, package modutils removed but not purged
 â
 â If the reported problem is a local modification, it needs to be fixed manually. If it's a bug in the package, it should be reporte
 â the package. See http://wiki.debian.org/LSBInitScripts/DependencyBasedBoot for more information about how to fix the problems prev
 â
 â To reattempt the migration process after the problems have been fixed, run "dpkg-reconfigure sysv-rc". Unable to migrate to depend
 â
 â Tests have determined that problems in the boot system exist which prevent migration to dependency-based boot sequencing:
 â
 â insserv: warning: script 'K20scdate' missing LSB tags and overrides, insserv: warning: script 'S25libdevmapper1.02' missing LSB ta
 â warning: script 'scdate' missing LSB tags and overrides, insserv: warning: script 'libdevmapper1.02' missing LSB tags and override
 â but not purged, package modutils removed but not purged
 â
 â If the reported problem is a local modification, it needs to be fixed manually. If it's a bug in the package, it should be reporte
 â the package. See http://wiki.debian.org/LSBInitScripts/DependencyBasedBoot for more information about how to fix the problems prev
 â
 â To reattempt the migration process after the problems have been fixed, run "dpkg-reconfigure sysv-rc".
```

1.25 Error in file /var/lib/dpkg/status

Added on May 21, 2011.

The arrival of Squeeze has brought the "illegalization" of the underscore as a valid character in the /var/lib/dpkg/status file and this cause some havoc if there are some packages having this character in its name

```
warning, in file '/var/lib/dpkg/status' near line 6627 package 'virtualbox':
 error in Version string '1.3.6_Debian_etch': invalid character in version number
warning, in file '/var/lib/dpkg/status' near line 6628 package 'virtualbox':
 error in Config-Version string '1.3.6_Debian_etch': invalid character in version number
```

The solution is to clean this packages from the list and from the available packages info. In this particular case the package was not installed but still caused the problem.

```
sudo dpkg --clear-avail
sudo aptitude purge virtualbox
```

1.26 How to record terminal sessions using ttyrec

Added on September 2nd, 2011.

The recording of terminal sessions can be of utility, specially for teaching purposes. A nice tool to obtain this recordings is the ttyrec program.

We will briefly show the way to record your terminal sessions. The first step is to install the ttyrec package (in Debian)

```
# apt-get install ttyrec
Reading package lists... Done
Building dependency tree... Done
The following NEW packages will be installed:
  ttyrec
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 30.9kB of archives.
After unpacking 131kB of additional disk space will be used.
Get:1 http://archive.debian.org etch/main ttyrec 1.0.6-1 [30.9kB]
Fetched 30.9kB in 1s (21.0kB/s)
Selecting previously deselected package ttyrec.
(Reading database ... 103998 files and directories currently installed.)
Unpacking ttyrec (from .../ttyrec_1.0.6-1_i386.deb) ...
Setting up ttyrec (1.0.6-1) ...
#
```

Then a second step is to open a terminal window with a 80x25 size where ttyrec will be executed.

Prior to the execution it is useful to create an initialization file to start ttyrec with the appropriate settings. This is not mandatory, but can be interesting. A sample file is the following:

```
# input file for starting a bash session using ttyrec
# by Currix TM.
#
# vt100 setting
export TERM=vt100
#
# change prompt
export PS1="$ "
#
# Remove trailing CTRL-C from comments
bind 'set echo-control-characters off'
```

If the name of this file is .inputrc_ttyrec the program ttyrec should be invoked as

```
ttyrec -e 'bash --rcfile .inputrc_ttyrec'
```

In the references for this chapter ('References' on the following page) a link to a recorded session is found. There it is shown how for example a vi session can be included or how comments are made by typing CTRL-c at the end of each comment line.

The default output of ttyrec is a file named ttyrecord. To save the output in a different file the filename should be included at the end of the ttyrec program invocation. The output file can be played back using the program ttyplay, included in the ttyrec package. A sample file can be downloaded and played

```
wget http://www.uhu.es/gem/clinux/descargas/ttyrecord
ttyplay ttyrecord
```

The speed of the playback can be increased with the + key or slowed with -.

An interesting site to find terminal recordings is Playterm, whose URL can be found in the references section.

1.26.1 References

1 ttyrec homepage (http://0xcc.net/ttyrec/index.html.en)

2 ttyrecord file sample (http://www.uhu.es/gem/clinux/descargas/ttyrecord)

3 Playterm homepage (http://www.playterm.org/)

1.27 Using Bash to count the number of files in a directory

Added on September 6th, 2011.

The number of files in a directory, or in general, the number of files displayed by the command ls can be counted using a pipe and the command wc

```
$ ls -1 1*gif
115810.strip.print.gif
115811.strip.print.gif
115813.strip.print.gif
116637.strip.print.gif
116639.strip.print.gif
116642.strip.print.gif
117321.strip.print.gif
117324.strip.print.gif
118015.strip.print.gif
$ ls -1 1*gif | wc -l
9
```

Some alternatives with Perl oneliners can be found in 'Interesting perl oneliners' on page 42

1.28 Getting detailed information about your hardware

Added on November 30th, 2011.

The command lshw lists a (very) detailed information about your hardware. The Debian package has the same name. This program should be executed as superuser.

```
deckard
    description: All In One
    product: iMac8,1
    vendor: Apple Inc.
    version: 1.0
.
.
. Very long output here...
.
.
  *-network DISABLED
       description: Ethernet interface
       physical id: 1
       logical name: vboxnet0
       serial: 0a:00:27:00:00:00
       capabilities: ethernet physical
       configuration: broadcast=yes multicast=yes
```

To get an abridged output with the essential info the program can be run with the option -short.

1.29 Adding your servers load to the screen hardstatus line

Added on December 1st, 2011.

It is possible to configure screen to display the cpu load or other information of interest in the hardstatus line. In this case we use Perl and ssh. We prepare a short Perl script that takes as arguments an argument related to the output format and a server ip. The considered output formats are green (cpu load < 1.0), yellow (3.0 > cpu load >= 1.0), and red (cpu load >= 3.0). The Perl code is called perl_load

```
#!/usr/bin/perl
#
use strict;
use warnings;
use 5.010;
#
# by Currix TM
#
my $code =  "$ARGV[0]";
my $hostname = "$ARGV[1]";
#
chomp(my $output = `ssh $hostname cat /proc/loadavg`);
my @load = ($1,$2,$3) if ($output =~ /^(\d+\.\d+)\s+(\d+\.\d+)\s+(\d+\.\d+)\s+\d.*$/);
#
given($code) {
  when ("red") {
    if ($load[0] >= 3.0) {
      $output = $hostname.": ".$load[0];
    } else {
      $output = '';
    }
  }
  when ("orange") {
    if ($load[0] >= 1.0 && $load[0] < 3.0) {
      $output = $hostname.": ".$load[0];
    } else {
      $output = '';
    }
  }
  when ("green") {
    if ($load[0] < 1.0) {
      $output = $hostname.": ".$load[0];
    } else {
      $output = '';
    }
  }
  default {
    $output = "wrong option";
  }
}
print "$output";
```

The script output is the server name and the cpu load average for the past minute if the load is in the interval defined by the format output.

The inclusion of this script in the `.screenrc` is accomplished using `backticks`. For more information on this point and color formatting check the `screen` man page. An example `.screenrc` is given below

```
shell -$SHELL
#
backtick 101 31 31 /home/username/bin/perl_load red server_name_or_IP
backtick 102 61 61 /home/username/bin/perl_load orange server_name_or_IP
backtick 103 91 91 /home/username/bin/perl_load green server_name_or_IP
#
hardstatus alwayslastline
hardstatus string '%{= kG}[ %{G}%H %{g}][%= %{=kw}%?%-Lw%?%{r}(%{W}%n*%f%t%?(%u)%?%{r})%{w}%?%+Lw%?%?%= %{g}][%{B}%Y-%m-%d %{W}%c  |

# Default screens
screen -t mutt          0       mutt
screen -t server 1
screen -t box_1         2       ssh username@rbox_1
screen -t box_2  3       ssh username@box_2
```

1.30 Making a backup of a `Gmail` account.

Added on December 7th, 2011.

We describe a way of backing up a `Gmail` mail account making use of `IMAP` and the `getmail` program. We assume that the copies want to be made in different `mbox` files for different periods of time. For example, a backup copy on a yearly basis.

The first step consist in enabling `IMAP` access in the settings section of the `Gmail` account to be backed up and the creation of a set of labels in the `Gmail` account grouping the messages that will be backed up. This can be accomplished using a search with `before` and `after`. For example, we can search for mail that matches the search `before:2007/10/01 AND after:2007/09/01`, which will include conversations for September 2007, and group the result of the search under the label `mail_sept_2007`.

The second step is the installation of the `getmail4` package, a mail agent program far simpler to configure than `fetchmail`.

```
$ sudo aptitude install getmail4
```

The next step is the more complicated one and implies the configuration of the `getmail` program. In order to do so a `.getmail` directory should be created, with restricted permissions, and a config file `getmail.gmail` edited

```
$ mkdir .getmail
$ chmod og-xr .getmail/
```

This is a sample `getmail.gmail`

```
$ cat getmail/getmail.gmail
[retriever]
type = SimpleIMAPSSLRetriever
server = imap.gmail.com
username = XXXXX@gmail.com
password = XXXXXXXXXXXX
mailboxes = ("mail_sept_2007",)

[destination]
type = Mboxrd
path = DESTINATION_PATH/mail-sept_2007.mbox

[options]
# print messages about each action (verbose = 2)
# Other options:
# 0 prints only warnings and errors
# 1 prints messages about retrieving and deleting messages only
verbose = 2
message_log = ~/.getmail/gmail.log
```

This file should also have read permissions only for the user owning it. Messages in this example are saved with `Mbox` format. For other alternatives check references. The `Mbox` file has to be created before backing up.

```
$ touch DESTINATION_PATH/mail-sept_2007.mbox
```

The last step is to proceed to download the mail

```
$ getmail -r getmail.gmail
getmail version 4.20.0
Copyright (C) 1998-2009 Charles Cazabon.  Licensed under the GNU GPL version 2.
SimpleIMAPSSLRetriever:currix@gmail.com@imap.gmail.com:993:
.
. Lots of output here...
.
  1200 messages (596565682 bytes) retrieved, 0 skipped
Summary:
Retrieved 1200 messages (596565682 bytes) from SimpleIMAPSSLRetriever:XXXX@gmail.com@imap.gmail.com:993
```

The resulting `Mbox` files can be easily accessed with `Mutt` or other mail reader program.

1.30.1 References

1 Matt Cutts Blog (http://www.mattcutts.com/blog/backup-gmail-in-linux-with-getmail)

2 Peng.u.i.n Blog (http://blog.pengdeng.com/2008/03/backup-gmail-via-imap-using-getmail_16.html)

1.31 Redirecting `STDERR` to `SDTOUT`

Added on Febuary 3rd, 2012.

Sometimes you need to redirect the standard error output, `STDERR`, merging it with the standard output, `STDOUT`. This is needed, for example, when you have a long error output from a program compilation and you want to pipe it to `less` to peruse the compiler info. This can be done using the standard `bash` redirection tools as `2>&1`. If we want to check the error output of a compilation this can be done as follows

```
$ make all 2>&1 | less
```

Instead, if instead you want to merge `STDOUT` into `STDERR` this is accomplished as `1>&2`.

1.32 Listing and extracting files from a deb package file.

Added on August 11th, 2012.

The deb extension marks Debian software package format. This standard is also used in other distributions that are based on Debian (e.g. Ubuntu or Knoppix).

The dpkg program is the low level package manager for Debian and with this command you can list and extract the files containd in a package, as shown below.

We use as an example the ovpc package. Once we download the file, to display the contents of a debian package the required option is dpkg -c

```
$ dpkg -c ovpc_1.06.94-3_i386.deb
dr-xr-xr-x root/root         0 2010-02-25 10:54 ./
dr-xr-xr-x root/root         0 2010-02-25 10:54 ./ovpc/
dr-xr-xr-x root/root         0 2010-02-25 10:54 ./ovpc/pkg/
dr-xr-xr-x root/root         0 2010-02-25 10:54 ./ovpc/pkg/lib/
dr-xr-xr-x root/root         0 2010-02-25 10:48 ./ovpc/pkg/lib/header/
-r-xr-xr-x root/root       130 2009-10-29 17:06 ./ovpc/pkg/lib/header/libov.so
.
.
.

-r-xr-xr-x root/root       131 2009-10-29 17:06 ./ovpc/pkg/etc/conf
dr-xr-xr-x root/root         0 2010-02-25 10:54 ./ovpc/pkg/etc/conf/log.conf
```

You can use dpkg -x to extract the files from a deb package as shown below.

```
$ dpkg -x  ovpc_1.06.94-3_i386.deb /tmp/ov
$ ls /tmp/ov
ovpc
```

But there is also an alternative way of proceeding. As deb files are ar archives, containing three files: debian-binary, control.tar.gz, and data.tar.gz. We can use the ar and tar commands to extract and view the files from the deb package.

First, extract the content of the deb archive file using ar.

```
$ ar -vx ovpc_1.06.94-3_i386.deb
x - debian-binary
x - control.tar.gz
x - data.tar.gz
$
```

Next, extract the content of data.tar.gz file as follows.

```
$ tar -xvzf data.tar.gz
./
./ovpc/
./ovpc/pkg/
./ovpc/pkg/lib/
./ovpc/pkg/lib/header/
./ovpc/pkg/lib/header/libov.so
.
.
./ovpc/pkg/etc/conf
./ovpc/pkg/etc/conf/log.con
```

1.32.1 References

1 TheGeekStuff (http://www.thegeekstuff.com/2010/04/view-and-extract-packages/)

1.33 Using script to keep terminal output records.

Added on November 18th, 2012.

The script command saves in a file the output of the terminal where it has been executed. The basis command syntax is

```
$ script [filename]
```

If no filename is given then the default filename is `typescript`. Be aware that script works best without making use of programs that manipulate the screen such that `vi` or `less`.

To exit `script` you can type `CTRL-D` or `exit`.

You can use the `-a` option to append the output to a file without overwriting it. An example of use of script is the following:

```
$ script test_script
Script started, file is test_script
$ who
bara    tty8         2012-11-17 13:57 (:0)
bara    pts/1        2012-11-17 13:58 (:0.0)
$ exit
exit
Script done, file is test_script
$ ls -l test_script
-rw-r--r-- 1 bara rsrchrs 780 Nov 18 12:02 test_script
$ cat test_script
Script started on Sun 18 Nov 2012 12:02:17 PM CET
$ who
bara    tty8         2012-11-17 13:57 (:0)
bara    pts/1        2012-11-17 13:58 (:0.0)
$ exit
exit

Script done on Sun 18 Nov 2012 12:02:33 PM CET
$
```

Another interesting option is to share a session output between two users. This is very handy when explaining to someone a particular feature in the console. In order to do so the script output is sent to a fifo file and the flush option `-t` is added while the second user should have `ssh` access to the computer where script is run.

For example if user `bara` in computer `prague` wants to show her output to user `alice` in computer `wland` the necessary steps are the following:

On `bara`'s side

```
bara@prague$ mkfifo foo; script -f foo
```

The console now will be temporarily frozen until `alice` starts getting the output from the fifo file `foo`.

```
alice@wland$ ssh alice@prague
alice@prague's password:
WWelcome to Ubuntu 12.04.1 LTS (GNU/Linux 3.2.0-33-generic i686)

 * Documentation:  https://help.ubuntu.com/

0 packages can be updated.
0 updates are security updates.

Last login: Sun Nov 18 10:50:37 2012 from platea.local
alice@prague$ cat /home/bara/foo
Script started on Sun 18 Nov 2012 11:21:29 AM CET
bara@prague$
```

And from now on user `bara` can proceed to work on the terminal sharing the output with `alice`. Beware that the permissions in the fifo file and `bara`'s home directory should allow `alice` to access the fifo.

1.34 Checking actual kernel compiling options.

Added on January 8th, 2013.

Sometimes it is necessary to check if the kernel of your computer has been compiled including a particular option. For example, let's check if the kernel has been compiled with the `CONFIG_EFI_PARTITION` option turned on (see 'Formatting HD partitions larger than 2 TB' on the facing page).

```
$ cd /boot/
$ ls
System.map-2.6.32-5-amd64  config-2.6.32-5-amd64  grub/  initrd.img-2.6.32-5-amd64  vmlinuz-2.6.32-5-amd64
$ grep _EFI_ config-2.6.32-5-amd64
CONFIG_EFI_VARS=m
CONFIG_EFI_PARTITION=y
```

1.35 Formatting HD partitions larger than 2 TB

Added on January 8th, 2013.

The first step is to check whether your kernel has been compiled with the `CONFIG_EFI_PARTITION` option turned on (see 'Checking actual kernel compiling options.' on the preceding page). If this is so, let's assume that the device assigned to the HD that will be partitioned is `/dev/sdb`, that it is a 2 TB HD and will be mounted in `/large_disk`. The `fdisk` program and its variants cannot be used because the new partition will be too large (larger than 1.5 TB). Be warned that `fdisk` and all its variants will not issue warnings but will create partitions not larger than 1.5 TB.

Be also warned that all the contents of the disk will be lost. *Make the necessary backups in advance.*

Let's assume that you've installed the `parted` program.

```
# parted /dev/sdb
GNU Parted 2.3
Using /dev/sdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

You can print the actual partition scheme

```
(parted) print
Model: DELL PERC H700 (scsi)
Disk /dev/sdb: 1933GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number  Start   End     Size    Type     File system  Flags
 1      1049kB  1933GB  1933GB  primary  ext3

(parted)
```

The next step is to remove the existing partition

```
(parted) rm 1
(parted) print
Model: DELL PERC H700 (scsi)
Disk /dev/sdb: 1933GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number  Start  End  Size  Type  File system  Flags

(parted)
```

Next, the disk is labeled as a gpt disk

```
(parted) mklabel gpt
Warning: The existing disk label on /dev/sdb will be destroyed and all data on this disk will be lost. Do you want to continue?
Yes/No? Yes
(parted) print
Model: DELL PERC H700 (scsi)
Disk /dev/sdb: 1933GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start  End  Size  File system  Name  Flags

(parted)
```

Once the disk is relabeled, the new partition can be created and you can exit `parted`

```
(parted) mkpart
Partition name?  []? primary
File system type?  [ext2]?
Start? 0
End? -0
Warning: The resulting partition is not properly aligned for best performance.
Ignore/Cancel? I
(parted) print
Model: DELL PERC H700 (scsi)
Disk /dev/sdb: 1933GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start   End     Size    File system  Name     Flags
 1      17.4kB  1933GB  1933GB               primary

(parted) quit
Information: You may need to update /etc/fstab.
```

Create the filesystem in the usual way

```
# mkfs.ext4 /dev/sdb1
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
117964800 inodes, 471859191 blocks
23592959 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
14400 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000, 7962624, 11239424, 20480000, 23887872, 71663616, 78675968,
        102400000, 214990848
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 37 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

Finally check the partition `UUID` and edit accordingly the `fstab` file.

```
# ls -l /dev/disk/by-uuid/
total 0
lrwxrwxrwx 1 root root 10 Sep 21 11:43 346dc157-f150-4a1e-8b83-bf4a50dc3cad -> ../../sda5
lrwxrwxrwx 1 root root 10 Sep 26 13:00 43615088-9470-419c-9958-199b0bcbbbd6 -> ../../sdb1
lrwxrwxrwx 1 root root 10 Sep 21 11:43 5fa63da1-1ff9-4fbf-93c2-4fcdef686892 -> ../../sda7
lrwxrwxrwx 1 root root 10 Sep 21 11:43 93272bd4-51b2-4e8a-940d-f85d5fa1978d -> ../../sda6
#
```

In our case the previous partition `UUID` in `fstab` needs to be replaced by the new value `43615088-9470-419c-9958-199b0bcbbbd6` and the filesystem type should be changed also changed if necessary. The corresponding `fstab` line should read as follows

```
# cat /etc/fstab | grep large
# /large_disk was on /dev/sdb1 during installation
UUID=43615088-9470-419c-9958-199b0bcbbbd6 /large_disk     ext4    defaults        0       2
```

The final result is

```
# df -h | grep large
/dev/sdb1             1.8T  196M  1.7T   1% /large_disk
```

If necessary, reboot and check that the partition is correctly mounted.

1.35.1 References

1 Cyberciti (http://www.cyberciti.biz/tips/fdisk-unable-to-create-partition-greater-2tb.html)

1.36 Adding a partition to `fstab` by `UUID`

Added on January 31st, 2015.

Assume that you have reformatted your hard disk and created a new partition, maybe liberating some space from other operating system, and you want to add the new partition to your `GNU/Linux` system. In our case we suppose that the new partition is `/dev/sda9` with an `ext4` filesystem and we want to mount it in a `/data` directory. We then want to give access to a system's user, for instace, user `curro` to this partition.

The first step is check the partition `UUID`,

```
$ ls -l /dev/disk/by-uuid/
total 0
lrwxrwxrwx 1 root root 10 Feb  1 20:17 2AF23ED0F23E9FCD -> ../../sda1
lrwxrwxrwx 1 root root 10 Jan 31 20:43 2e6850cc-0bb9-4f36-b0a8-c167aa257dd6 -> ../../sda7
lrwxrwxrwx 1 root root 10 Jan 31 20:43 55d1fe60-aed5-43c2-9c57-34cdea5accda -> ../../sda6
lrwxrwxrwx 1 root root 10 Jan 31 20:43 578091f7-b057-481a-82b3-6db5a3b86bec -> ../../sda5
lrwxrwxrwx 1 root root 10 Feb  1 20:17 5c1dccb0-e78a-48cc-a63d-95148a8f6cb5 -> ../../sda9
lrwxrwxrwx 1 root root 10 Feb  1 20:16 5F1F-1069 -> ../../sdc1
lrwxrwxrwx 1 root root 10 Feb  1 20:17 6E223D37223D059D -> ../../sda3
lrwxrwxrwx 1 root root 10 Feb  1 20:16 9dbd2628-0353-4798-85e4-738b25d0cdd5 -> ../../sdc2
lrwxrwxrwx 1 root root 10 Feb  1 20:17 c30b003f-11f6-430d-8c19-892d68b2c0cf -> ../../sda8
lrwxrwxrwx 1 root root 10 Feb  1 20:17 EC7A41587A412126 -> ../../sda2
```

We then, as root, create the mount point, `/data` and add the following line to the `/etc/fstab`

```
UUID=5c1dccb0-e78a-48cc-a63d-95148a8f6cb5       /data           ext4    defaults,user_xattr        0       2
```

Once the `fstab` file is saved, we can now mount the new partition

```
# mount /data
# mount | grep sda9
/dev/sda9 on /data type ext4 (rw,relatime,data=ordered)
```

The last step is to give user *curro* access to this directory. This is done creating a directory and assigning the directory to this user

```
# mkdir downloads_curro
root@kimoshi:/data# chown curro:curro downloads_curro/
```

Then user curro can move the data in his folder `Downloads` to this new directory and link the old folder to the new one.

```
$ mv Downloads/* /data/downloads_curro
$ rmdir Downloads
$ ln -fs /data/downloads_curro ./Downloads
```

1.37 Error ejecting a CD

Added on January 31st, 2015.

Sometimes when you eject a CD from the terminal the system displays the following error

```
$ eject
eject: unable to eject, last error: Inappropriate ioctl for device
```

The same error is obtained when we try to eject the CD as superuser

```
# eject
eject: unable to eject, last error: Inappropriate ioctl for device
```

The verbose output is

```
# eject -v /dev/sr0
eject: device name is '/dev/sr0'
eject: expanded name is '/dev/sr0'
eject: '/dev/sr0' is not mounted
eject: '/dev/sr0' is not a mount point
eject: '/dev/sr0' is not a multipartition device
eject: trying to eject '/dev/sr0' using CD-ROM eject command
eject: CD-ROM eject command failed
eject: trying to eject '/dev/sr0' using SCSI commands
eject: SCSI eject failed
eject: trying to eject '/dev/sr0' using floppy eject command
eject: floppy eject command failed
eject: trying to eject '/dev/sr0' using tape offline command
eject: tape offline command failed
eject: unable to eject, last error: Inappropriate ioctl for device
```

The solution is, as root, use the `-i off` option and then eject the cd

```
deckard:~# eject -i off /dev/cdrom
CD-Drive may be ejected with device button
deckard:~# eject /dev/cdrom
```

An the CD is cleanly ejected. Happy end.

1.38 Creating a `Makefile` to use `cpp` with `gfortran`

Added on November 09th, 2015.

There is a way of using the C language preprocessor, `cpp`, with gfortran. In this way we can communicate via `Makefile` with our program. Let's assume that we have a `Fortran 90` program to compute two normally distributed set of random numbers that use the Box-Muller transform. The program is called `box_muller.f90` and we also compute the mean, median and standard variance of the resulting distributions using a subroutine called `stats.f90`[6]. The `FORTRAN 90` codes are provided in 'Fortran Codes' on the facing page.

The `Makefile` used for compilation is

```
# Fortran Compiler
FC=gfortran
#
# C Preprocessor
CPP=cpp
#
TYPE_DEF=-DPREC=DP
ifdef USE_SINGLE
TYPE_DEF = -DPREC=SP
endif
#
# Optimization Flags
OPTFLAGS = -O3 -funroll-loops -march=native
OPTFLAGS_P = "\"-O3 -funroll-loops -march=native\""
#
all: box_muller
#
.FORCE:
#
box_muller: box_muller.f90 stats.f90 .FORCE
 $(CPP)  -std=c89 $(TYPE_DEF) -DFLAGS=$(OPTFLAGS_P) $< > /tmp/$<
 $(CPP)  -std=c89 $(TYPE_DEF)  stats.f90 > /tmp/stats.f90
 $(FC) $(OPTFLAGS) -o $@ /tmp/stats.f90 /tmp/$<
 rm /tmp/stats.f90 /tmp/$<
#
clean:
 rm -f *.o box_muller
```

Note how `cpp` is called twice, once for each file and the `PREC` string are replaced either by `DP` or `SP`, the last case if the option `USE_SINGLE` is set. In the main program the string `FLAGS` is replaced by the contents of the variable `OPTFLAGS_P` to print the compilation flags used. Thus to run the program in single precision the user proceed as follows

```
$ make all USE_SINGLE=1
cpp  -std=c89 -DPREC=SP  -DFLAGS="\"-O3 -funroll-loops -march=native\"" box_muller.f90 > /tmp/box_muller.f90
cpp  -std=c89 -DPREC=SP   stats.f90 > /tmp/stats.f90
gfortran -O3 -funroll-loops -march=native -o box_muller /tmp/stats.f90 /tmp/box_muller.f90
rm /tmp/stats.f90 /tmp/box_muller.f90
$ ./box_muller
10000
 Time Box Muller subroutine :   4.00000019E-03
  MEAN =   -7.31508713E-03
  STANDARD DEVIATION =   0.995381057
  MEDIAN IS =    1.01202750E-04
 Time STATS subroutine 1:  0.100005999
  MEAN =    2.96602229E-04
  STANDARD DEVIATION =   0.999957144
  MEDIAN IS =    1.29099786E-02
 Time STATS subroutine 2:  0.100006007
 KIND = SINGLE ;  Optimization Flags: -O3 -funroll-loops -march=native
```

[6] Please, note that the subroutine is not optimized and could take a long time for a large number of data due to an inefficient sort algorithm.

1.38.1 Fortran Codes

Main program: `box_muller.f90`

```
PROGRAM Box_Muller_Prog
  !
  IMPLICIT NONE
  !
  ! Single and double precision kind parameters
  INTEGER, PARAMETER :: SP = KIND(1.0)
  INTEGER, PARAMETER :: DP = KIND(1.0D0)
  !
  INTEGER :: I, IERR
  REAL(KIND = PREC), DIMENSION(:), ALLOCATABLE :: X, Y
  REAL(KIND = PREC) :: M, SD, MEDIAN
  REAL(KIND = SP) :: time_end, time_start
  !
  !
  CHARACTER(LEN = 6) :: PRECISION
  !
  ! interface block
  INTERFACE
     SUBROUTINE STATS(VECTOR,N,MEAN,STD_DEV,MEDIAN)
       IMPLICIT NONE
       ! Single and double precision kind parameters
       INTEGER, PARAMETER :: SP = KIND(1.0)
       INTEGER, PARAMETER :: DP = KIND(1.0D0)
       INTEGER , INTENT(IN)                    ::  N
       REAL(KIND = PREC)      , INTENT(IN) , DIMENSION(:)   :: VECTOR
       REAL(KIND = PREC)      , INTENT(OUT)                 :: MEAN
       REAL(KIND = PREC)      , INTENT(OUT)                 :: STD_DEV
       REAL(KIND = PREC)      , INTENT(OUT)                 :: MEDIAN
     END SUBROUTINE STATS
  END INTERFACE
  !
  ! Set Precision for output
  IF (PREC == DP) THEN
     PRECISION = "DOUBLE"
  ELSE
     PRECISION = "SINGLE"
  ENDIF
  !
  ! Input number of data
  READ*, I
  !
  ALLOCATE(X(1:I), STAT = IERR)
  IF (IERR /= 0) THEN
     PRINT*, "X allocation request denied."
     STOP
  ENDIF
  !
  ALLOCATE(Y(1:I), STAT = IERR)
  IF (IERR /= 0) THEN
     PRINT*, "Y allocation request denied."
     STOP
  ENDIF
  !
  ! Set time start
  CALL CPU_TIME(time_start)
  !
  CALL BOX_MULLER(I)
  !
  ! Set time end
  CALL CPU_TIME(time_end)
  !
  !PRINT*, X
  !PRINT*, Y
  !
  PRINT*, "Time Box Muller subroutine :", time_end - time_start
  time_start = time_end
  !
  CALL STATS(X,I,M,SD,MEDIAN)
  !
  PRINT *,' MEAN = ',M
  PRINT *,' STANDARD DEVIATION = ',SD
  PRINT *,' MEDIAN IS = ',MEDIAN
  !
  ! Set time end
  CALL CPU_TIME(time_end)
  !
  PRINT*, "Time STATS subroutine 1:", time_end - time_start
  time_start = time_end
  !
  IF (ALLOCATED(X)) DEALLOCATE(X, STAT = IERR)
  IF (IERR /= 0) THEN
     PRINT*, "X NON DEALLOCATED!"
     STOP
  ENDIF
  !
  CALL STATS(Y,I,M,SD,MEDIAN)
```

```
    !
    !
    PRINT *,' MEAN = ',M
    PRINT *,' STANDARD DEVIATION = ',SD
    PRINT *,' MEDIAN IS = ',MEDIAN
    !
    ! Set time end
    CALL CPU_TIME(time_end)
    !
    PRINT*, "Time STATS subroutine 2:", time_end - time_start
    time_start = time_end
    !
    IF (ALLOCATED(Y)) DEALLOCATE(Y, STAT = IERR)
    IF (IERR /= 0) THEN
       PRINT*, "Y NON DEALLOCATED!"
       STOP
    ENDIF
    !
    PRINT*, "KIND = ", PRECISION, " ;  Optimization Flags: ", FLAGS
    !
  CONTAINS
    !
    SUBROUTINE BOX_MULLER(dim)
      !
      ! Uses the Box-Muller method to create two normally distributed vectors
      !
      INTEGER, INTENT(IN) :: dim
      !
      REAL(KIND = PREC), PARAMETER :: PI = ACOS(-1.0)
      REAL(KIND = PREC), DIMENSION(dim) :: RANDOM_u, RANDOM_v ! Automatic arrays
      !
      CALL RANDOM_NUMBER(RANDOM_u)
      CALL RANDOM_NUMBER(RANDOM_v)
      !
      X = SQRT(-2.0_ PREC*LOG(RANDOM_u))
      Y = X*SIN(2*PI*RANDOM_v)
      X = X*COS(2*PI*RANDOM_v)
      !
    END SUBROUTINE BOX_MULLER
    !
  END PROGRAM Box_Muller_Prog
```

Subroutine: `stats.f90`

```
  SUBROUTINE STATS(VECTOR,N,MEAN,STD_DEV,MEDIAN)
    IMPLICIT NONE
    ! Single and double precision kind parameters
    INTEGER, PARAMETER :: SP = KIND(1.0)
    INTEGER, PARAMETER :: DP = KIND(1.0D0)
    ! Argument definition
    INTEGER , INTENT(IN)                      ::  N
    REAL(KIND = PREC)      , INTENT(IN) , DIMENSION(:)    ::  VECTOR
    REAL(KIND = PREC)      , INTENT(OUT)                  ::  MEAN
    REAL(KIND = PREC)      , INTENT(OUT)                  ::  STD_DEV
    REAL(KIND = PREC)      , INTENT(OUT)                  ::  MEDIAN
    ! Local variables
    REAL(KIND = PREC) :: VARIANCE = 0.0
    REAL(KIND = PREC)      :: SUMXI = 0.0, SUMXI2 = 0.0
    REAL(KIND = PREC)      , DIMENSION(1:N)              ::  Y
    !
    SUMXI=SUM(VECTOR)
    SUMXI2=SUM(VECTOR*VECTOR)
    MEAN=SUMXI/N
    VARIANCE=(SUMXI2-SUMXI*SUMXI/N)/(N-1)
    STD_DEV = SQRT(VARIANCE)
    Y=VECTOR
    ! Sort values
    CALL SELECTION
    IF (MOD(N,2) == 0) THEN
       MEDIAN=(Y(N/2)+Y((N/2)+1))/2
    ELSE
       MEDIAN=Y((N/2)+1)
    ENDIF
  CONTAINS
    SUBROUTINE SELECTION
      IMPLICIT NONE
      INTEGER :: I,J,K
      REAL :: MINIMUM
      DO I=1,N-1
         K=I
         MINIMUM=Y(I)
         DO J=I+1,N
            IF (Y(J) < MINIMUM) THEN
               K=J
               MINIMUM=Y(K)
            END IF
         END DO
         Y(K)=Y(I)
         Y(I)=MINIMUM
      END DO
```

```
    END SUBROUTINE SELECTION
  END SUBROUTINE STATS
```

1.38.2 References

1 StackOverflow (https://stackoverflow.com/questions/31649691/stringify-macro-with-gnu-gfortran/31753386#31753386)

Chapter 2

Network and Navigators

2.1 Running multiple `firefox` instances

In order to execute `firefox` (`mozilla` or `iceweasel`) from a machine that is different to the local box while running `firefox` locally the program should be launched as:

```
firefox -no-remote
```

2.2 Electronic signature with `iceweasel`

In order to sign documents of the *Junta de AndalucÃa* or *Universidad de Huelva*with `firefox` (`mozilla` or `iceweasel`) the following steps should be accomplished. This has been tested with the electronic administration website of the UHU (`www.uhu.es/ae`)

First a certificate from the `FNMT` accrediting the identity of the person who is going to sign the document should be among the certificates available to the navigator. If it is not so, it should be imported (`Preferences -> Advanced -> Encryption -> View Certificates -> Import`).

Once this is done the package[1] `sun-java6-plugin` should be installed and the navigator should be restarted after its installation.

The files jss33.jar (`https://ws022.juntadeandalucia.es/firmadigital/servicio/paginas/jss33.jar`) and libjss3.so (`https://ws022.juntadeandalucia.es/firmadigital/servicio/paginas/libjss3.so`) have to be downloaded. A page is opened with the instructions when the navigator detects its absence. They have to be copied with the appropriate permissions to the right plugin directory

```
# chmod 644 jss33.jar
# cp jss33.jar /usr/lib/jvm/java-6-sun/jre/lib/ext/
# chmod 755 libjss3.so
# cp libjss3.so /usr/lib/jvm/java-6-sun/jre/lib/i386/client/
```

Once this is done the electronic administration page should allow us to fill the form and (hopefully) to sign it.

2.3 Some tips to access the web using the console

The original console web browser is `lynx`. In order to avoid questions inquiring whether you accept or not cookies it can be invoked using the option

```
$ lynx -accept_all_cookies
```

[1] Written for Debian Lenny.

Another interesting option is `-dump`. When using this option the program writes the lynx output and writes it to the standard output. More modern and flexible is the `links` browser, that can be used in text mode but also can be compiled to be accessed from a graphical display (option `-g`). The `-dump` option is also available. A strictly text replacement for `lynx` (and more modern) is `Elinks` with several interesting options.

In order to download an off line copy of some web pages, so you can browse them later offline you can use `curl`, that can upload or download data from to or from a server. You can specify multiple URL's by ranges as in the following two examples

```
$ curl http://site.(s1,s2,s3).org
$ curl http://site.s[1-3].org
```

The copy of an entire site can be done using `wget`

```
$ wget -k -r -p http://www.interesting_site.org
```

The option `-r` recurses through the site links starting from `http://www.interesting_site.org/index.html`. The `-k` option make the links relative allowing the correct navigation through the downloaded pages. The `-p` option downloads all extra content on the page. This order makes a true mirror of a site in your computer.

Finally, the program `wput` uploads contents to internet using `FTP` as an interface and with a syntax like the `wget` one.

2.3.1 References

1 Upfront, Linux Journal, issue 197, September 2010 (http://www.linuxjournal.com)

2.4 Make files in a directory downloadable from a web server

In order that files can be listed downloaded when accessing a directory on a webserver the following line should be added to the `.htaccess` file

```
Options +Indexes
```

Be warned that the contents of all subdirectories of the directory will also be listable and downloadable...

Chapter 3

The `ssh` application

3.1 Removing a `known_hosts` entry

In order to remove from the `known_hosts` file the entry corresponding the the computer *hostname*, execute[1]

```
ssh-keygen -R hostname
```

3.2 Generating a key pair and exporting to a remote host

If we want to access a remote node with `ssh` without providing our password, we can do so using automatic login. In order to do so, we must first check if we have a keypair already generated.

```
$ ls .ssh/
known_hosts
```

If it is not generated, like in this example, we generate it using the command

```
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/users/home/username/.ssh/id_rsa): [press enter]
Enter passphrase (empty for no passphrase): [press enter]
Enter same passphrase again: [press enter]
Your identification has been saved in /users/home/username/.ssh/id_rsa.
Your public key has been saved in /users/home/username/.ssh/id_rsa.pub.
The key fingerprint is:
8b:93:61:e7:2d:4a:50:30:a3:23:7d:fc:c5:21:af:d7
```

The next step is to copy the generated public key to the remote host we want to be able to login automatically, *hostname*.

```
$ ssh-copy-id -i ~/.ssh/id_rsa.pub hostname
36
The authenticity of host 'hostname (XXX.163.XXX.XXX)' can't be established.
RSA key fingerprint is 37:2b:77:61:50:0f:2a:d2:7f:da:c9:a9:10:29:37:t6.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'hostname,XXX.163.XXX.XXX' (RSA) to the list of known hosts.
Password:
Now try logging into the machine, with "ssh 'hostname'", and check in:

  .ssh/authorized_keys

to make sure we haven't added extra keys that you weren't expecting.
```

And that's all. Now you can login via `ssh` without being prompted for a password.

[1] The `IP` value can also be used instead of the computer's hostname.

3.3 Launching with ssh programs that require a terminal

There are some programs that cannot be launched directly with ssh because they require an associated terminal. For example, mutt or screen,

```
$ ssh user@server screen
Must be connected to a terminal.
```

To solve this problem there is an option in ssh to force pseudo-tty allocation. For example, if we want to re-attach to a previous screen session in the node *server* we can do

```
$ ssh -t user@server screen -dr
```

3.3.1 References

1 Hack and / - Lightning Hacks, Linux Journal, issue 195, July 2010 (http://www.linuxjournal.com)

3.4 Tunneling with ssh

The ssh program has the powerful feature of making the user able to stablish encrypted tunnels between nodes. This is a major advantage of this extremely useful tool. There are several possibilities. Let's assume that we are user *bob* in a node called *home_box*, that has a private IP and it's behind a firewall, and we can access a second node, called *work_box*, where we are user *william*. We can connect from *home_box* to *work_box* but not the other way around. Thus, we want to make a encrypted tunnel that enables the coconnection from *work_box* to *home_box*. This is known as reverse ssh tunneling.

In order to create this tunnel we should run from *home_box*.

```
william@home_box:~$ ssh -R 9999:localhost:22 william@work_box
```

A session in *work_box* is opened, and while this session is active, the tunnel works. Then, if we log into *work_box*, we can connect to *home_box* making use of the tunnel.

```
william@work_box:~$ ssh -p 9999 bob@localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
RSA key fingerprint is b0:b6:f3:78:e2:8d:8f:8b:3f:ab:b4:d4:da:c5:a6:e1.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (RSA) to the list of known hosts.
bob@localhost's password:
bob@home_box:~$
```

A problem in this case is that once the initial connection is closed the tunnel collapses. A possible way to alleviate this problem there is to run an application in the connection to *work_box*, e.g. top trying to keep the connection alive.

This is not the best option. There are better ones, for example, launching a dedicated connection with options -f (detach ssh process from tty), -N (do not execute any command over ssh), and -o TCPKeepAlive=yes to keep the connection alive.

```
william@home_box:~$ ssh -f -N -o TCPKeepAlive=yes -R 9998:localhost:22 \
                    william@work_box
william@home_box:~$
```

We can then connect using the new tunnel.

```
william@work_box:~$ ssh -p 9998 bob@localhost
bob@localhost's password:
bob@home_box:~$
```

3.4.1 References

1 howtoforge :: reverse-ssh-tunneling (http://www.howtoforge.com/reverse-ssh-tunneling)

2 SSH Tunneling Made Easy (http://www.revsys.com/writings/quicktips/ssh-tunnel.html)

3 SSH Tunneling (http://www.linux-mag.com/id/1705)

4 Breaking Firewalls with OpenSSH and PuTTY (http://souptonuts.sourceforge.net/sshtips.htm)

3.5 Using ssh to make a tarball directly in a remote folder

Created on March 13th, 2015.

You could be interested in preparing a tarball and, for lack of space or to save intermediate steps, transfer on the fly the file to a remote box. For example, let's assume that you want to transfer a directory called data_EXP from a computer called laptop to another computer called backup_server (I know, not very imaginative. . . .). If our working directory is data_EXP parent directory we can then run

```
$ tar czf - data_EXP | ssh username@backup_server "cat > data_EXP_dir.tgz"
```

Chapter 4

Emacs text editor

4.1 Emacs macros

To define several emacs macros simultaneously and save them, the macros should be named. To give a macro a name define the macro and then execute M-x (name-last-kbd-macro) and give the macro a name[1].

The defined macro can be executed with the key combination C-x e unless a new macro is defined. When the last macro defined is not the one that should be executed, it should be recalled by its name as M-x *macro_name*.

If you consider the macro worth to be saved for use in future sessions it could be added to the emacs startup file. In order to do so, open the .emacs file or the file where you keep emacs macros and execute M-x (insert-kbd-macro) giving the name of the macro to be saved.

4.1.1 References

1 http://www.emacs.uniyar.ac.ru/doc/em24h/emacs112.htm

4.2 Changing the default system's text editor to Emacs

In order to change the default system's text editor to emacs or other alternative as vi in a Debian system the root user can update the system's default

```
$ sudo update-alternatives --config editor
[sudo] password for curro:

There are 4 alternatives which provide 'editor'.

  Selection    Alternative
-----------------------------------------------
          1    /bin/ed
*+        2    /bin/nano
          3    /usr/bin/vim.tiny
          4    /usr/bin/emacs22

Press enter to keep the default[*], or type selection number: 4
Using '/usr/bin/emacs22' to provide 'editor'.
$
```

Some programs use the environment variables $EDITOR or $VISUAL to decide which editor to use. For the sake of consistency on Debian systems, these variables should be set to /usr/bin/editor.

4.3 Removing empty lines in Emacs

Created on May 21st, 2015.

In order to remove all blank lines in a text file (or a selected text file region) you can use the command M-x flush-lines that removes all lines that matches a regexp condition as follows

[1] It is interesting to prefix the macros with your initials or your username to prevent conflicts.

```
M-x flush-lines RET ^$ RET
```

In case you want to remove lines containing only spaces the command is

```
M-x flush-lines RET ^\s-*$ RET
```

4.3.1 References

1 Mastering Emacs (https://www.masteringemacs.org/article/removing-blank-lines-buffer)

Chapter 5

Perl Language

This chapter has been translated into Spanish (http://www.webhostinghub.com/support/es/misc/instalacion-de-perl) language by Maria Ramos from Webhostinghub.com/support/edu (http://www.webhostinghub.com/support/edu).

5.1 Installing a CPAN module

There are two possible ways to install a CPAN module. We give both alternatives.

5.1.1 First alternative for installing a CPAN module

Download the CPAN module (we use the module Devel-SmallProf-2.02 as an example) and untar it

```
# tar xzvf Devel-SmallProf-2.02.tar.gz
# cd Devel-SmallProf-2.02
Devel-SmallProf-2.02# ls
Changes  MANIFEST  META.yml  Makefile.PL  README  TODO  lib  t
```

Compile and install the module

```
Devel-SmallProf-2.02# perl Makefile.PL
Checking if your kit is complete...
Looks good
Writing Makefile for Devel::SmallProf

Devel-SmallProf-2.02# make
cp lib/Devel/SmallProf.pm blib/lib/Devel/SmallProf.pm
Manifying blib/man3/Devel::SmallProf.3pm

Devel-SmallProf-2.02# make test
PERL_DL_NONLAZY=1 /usr/bin/perl "-MExtUtils::Command::MM" "-e" "test_harness(0, 'blib/lib', 'blib/arch')" t/*.t
t/part1....ok
t/part2....ok
t/part3....ok
t/part4....ok
t/pods.....skipped
        all skipped: Only the author needs to check that POD docs are right
All tests successful, 1 test skipped.
Files=5, Tests=14,  1 wallclock secs ( 0.08 cusr +  0.01 csys =  0.09 CPU)

Devel-SmallProf-2.02# make install
Installing /usr/local/share/perl/5.8.8/Devel/SmallProf.pm
Installing /usr/local/man/man3/Devel::SmallProf.3pm
Writing /usr/local/lib/perl/5.8.8/auto/Devel/SmallProf/.packlist
Appending installation info to /usr/local/lib/perl/5.8.8/perllocal.pod
```

5.1.2 Second alternative for installing a CPAN module

In this case we use an interactive shell that we launch as

```
# perl -MCPAN -e shell
```

The first time the shell is launched the system has to be configured and upgraded. The system prompts for several programs (unzip, etc.). Install them if they are not already installed in the computer. Select a repository (in my case `ftp://ftp.rediris.es/mirror/CPAN`) and most questions can safely answered using the default choice.

The first thing to do after configuration is to upgrade your `CPAN`:

```
cpan> install Bundle::CPAN


CPAN: Storable loaded ok
Fetching with LWP:
  ftp://ftp.rediris.es/mirror/CPAN/authors/01mailrc.txt.gz
Going to read /root/.cpan/sources/authors/01mailrc.txt.gz
CPAN: Compress::Zlib loaded ok
Fetching with LWP:
  ftp://ftp.rediris.es/mirror/CPAN/modules/02packages.details.txt.gz
.
.
.
```

Then you reload it:

```
cpan> reload cpan
```

And install the required module, `Roots` in this example:

```
cpan> install Math::Function::Roots

.
.
.

Appending installation info to /usr/lib/perl/5.8/perllocal.pod
  /usr/bin/make install  -- OK
```

5.2 Interesting perl oneliners

Updated on January 31st, 2015.

Updated on May 21st, 2015.

1 Execute a program, in this case `epstopdf`, using as an input all files sharing an common extension , in this case `eps`, in the current directory.

```
perl -e 'system "epstopdf $_" for (glob "*.eps");'
```

2 Erasing "phantom" files with 0 bytes size.

```
perl -e 'foreach (glob "*") \
{unless (-s $_) {"Deleting $_\n";unlink "$_";}}'
```

3 Checking `Postscript` files referred in a `LaTeX` output message.

```
latex filename.tex | perl -e \
'while (<>){foreach (split) {/<(.*?\.eps)>/ and push(@eps, $1)}};\
foreach (sort @eps) {print;print "\n"}'
```

4 Print apostrophe character.

```
perl -le 'print "'\'' is an apostrophe..."'
```

5 Changing a text file from UTF-8 encoding to ASCII. Note that it does not work for Spanish accented characters.

```
perl -ne 'for (unpack "U*", $_) \
{ printf $_ > 128 ? "x" :  "%c", $_ }' fileUTF.txt > fileASCII.txt
```

6 Adding an `E` character to the output of `Fortran` programs with three digits in the exponent that lacks this character.

```
perl -pi'.bak' -e 's/(\d)-(\d\d\d)/$1E-$2/g' fort.output
```

7 Delete empty lines in a file.

```
perl -ni -e 'chomp($_);print "$_\n" if ($_)'test.dat
```

8 Makes a substitution in a file, in the selected example substitute the word *figures* by *Figures* in all files with extension `tex` in the current directory.

```
perl -pi'.bak' -e 's/figures\//Figures\//g' *.tex
```

9 Search and display occurrences in a log file (*/var/log/loginlog.0* in this example) of succesful logins of user *curro*, showing the number of times the user has logged from each machine.

```
perl -e 'while (<>) {
if (m|\d+:\d+:\d+\s+(.*?)\s+.*ccepted.*curro\s+from\s+(.*?)\s+.*|)
 {$vh{"$1 from $2"}++;} }
 foreach (keys %vh)
  {print "$vh{$_} login(s) to $_\n";}'/var/log/loginlog.0
```

10 Count the number of files in a directory.

In the first example we use globbing to count the total number of files including hidden files or the number of files subject to some restriction.

```
$ perl -e 'my @files = glob "* .*"; print 1+$#files."\n"'
129
$ perl -e 'my @files = glob "1*gif"; print 1+$#files."\n"'
9
```

The same can be accomplished using directory handles and `grep`

```
$ perl -e 'opendir DH, ".";my @files = readdir DH; print 1+$#files."\n"'
129
$ perl -e 'opendir DH, ".";my @files = grep /^1.*\.gif$/, (readdir DH); print 1+$#files."\n"'
9
```

11 Get the last line of a file.

We include a oneliner that gets the last line of a series of files

```
perl -e 'foreach (@ARGV) {my $line = `tail -n 1 $_`; print $line}' output_notes_1* output_notes_2* output_notes_3* ...
```

In the original application the line was prepended with a number appearing in the filename as follows

```
perl -e 'foreach (@ARGV) {/.*(\d\d).*/;my $line = $1.`tail -n 1 $_`; print $line}' output_notes_1* output_notes_2* output_not
```

12 Change `Mac` carriage return to `UNIX` new line

Let's assume we have a bunch of `csv` files with `Mac` carriage return that our system interprets as a very long unique line. Instead of using `emacs` we can easily fix this with

```
perl -pi -e 's/\r/\n/g' *.csv
```

13 Extract the figure names from a `LaTeX` compile output and prepare a tarball with the figure files.

We assume that we compile a file called `rdiary_2014.tex` and all figs are in a directory called `Figs` and are `png` files. We use two pipes, the first one connect the output of `pdflatex` with a `perl` oneliner that reads the standad input and extract the file names. A second pipe sends to `tar` the file names. Notice the `-T -` options.

```
pdflatex rdiary_2014.tex | perl -e 'while (<>) {print "$1\n" if /<.*(F.*png).*>/g}' | tar czf figs.tgz -T -
```

5.3 Environment Codification and Character Ordering

The following short script permit to test a terminal codification.

```
#!/usr/bin/perl
use warnings;
use strict;
use Encode;
my @charsets = qw(utf-8 latin1 iso-8859-15 utf-16);
# some non-ASCII codepoints:
my $test = 'Ue: ' . chr(220) .'; Euro: '. chr(8364) . "\n";
#
for (@charsets){print "$_: " . encode($_, $test);}
```

Once the script is run, different lines appear, and the terminal charset is the one of the line correctly displayed. For example, if we execute the script in a terminal using the `UTF-8` coding system the output is something similar to

```
$ encodings.plx
utf-8: Ue: Ã; Euro: â¬
latin1: Ue: ï¿½; Euro: ?
iso-8859-15: Ue: ï¿½; Euro: ï¿½
utf-16: ï¿½ï¿½Ue: ï¿½; Euro:  ï¿½
```

Note that both special characters, Ã and the euro symbol, only appear in correcto form in the `utf-8` charset line.

The function `chr` in `Perl` takes a number as an argument and returns the character represented by that NUMBER in the selected character set. The function `encode` allows the codification of the character is different character sets. The four most common character sets are the ones included in the former example.

Another problem with character sets arise when ordering alphabetically a set of words of characters when the characters in the set are not the standard 127 `ASCII` character. For example, suppose that we are trying to order alphabetically the following set of names

```
Ãlvarez
MÃnguez
PÃ©rez
Perales
Pilar
Mola
Borrero
DÃaz
Diz
Delgado
Cuesta
Castro
CÃ¡Ã±amo
```

A standard program, comparing with the `cmp` function is as follows

```
#!/usr/bin/perl
use strict;
use warnings;
#
my @names;
#
while (defined(my $line = <>)) {
    chomp($line);
    my $elem = push(@names,$line);
    print "$elem element(s) added\n";
}
#
print "Reading process finished. Sorting ... ";
#
print "Done.\n\n";
print "Sorted set of names:\n";
foreach (sort by_name @names) {
    print "\t$_\n";
}
#
sub by_name {$a cmp $b}
```

However, when we run the program we obtain the somewhat surprising output

```
Sorted set of names:
 Borrero
 Castro
```

```
Cuesta
CÃ¡Ã±amo
Delgado
Diz
DÃaz
Mola
MÃnguez
Perales
Pilar
PÃ©rez
Ãlvarez
```

Clearly this is not the expected output if we intend to sort alphabetically (using Spanish sorting rules). The reason of this unexpected behavior is that the `cmp` function compares non-ASCII chars by codepoint number[1], which might give unexpected results. In order to sort according to a particular languague convention we should use the `locale` pragma. The previous program can be rewritten as follows

```
#!/usr/bin/perl
use strict;
use warnings;
#
##########
use locale;
use POSIX qw(locale_h);
setlocale(LC_COLLATE, 'es_ES@euro') or die "Locale es_ES\@euro not installed.\n";
##########
#
#
my @names;
#
while (defined(my $line = <>)) {
    chomp($line);
    my $elem = push(@names,$line);
    print "$elem element(s) added\n";
}
#
print "Reading process finished. Sorting ... ";
#
print "Done.\n\n";
print "Sorted set of names:\n";
foreach (sort by_name @names) {
    print "\t$_\n";
}
#
sub by_name {$a cmp $b}
```

After this change the word order is the usual one in Spanish.

```
Sorted set of names:
        Ãlvarez
        Borrero
        CÃ¡Ã±amo
        Castro
        Cuesta
        Delgado
        DÃaz
        Diz
        MÃnguez
        Mola
        Perales
        PÃ©rez
        Pilar
```

5.3.1 References

1 http://perlgeek.de/en/article/encodings-and-unicode

2 http://perldoc.perl.org/perllocale.html#USING-LOCALES

5.4 Extracting matches from a regular expressions

This can be easily done using the grouping metacharacters `'()'`. They allow the extraction of the parts of a string that matched the imposed condition. Each grouping marked by parentheses goes into a special variable `$1`, `$2`, etc. They can be used as ordinary variables.

[1] From `Wikipedia`: In character encoding terminology, a code point or code position is any of the numerical values that make up the code space. For example, ASCII comprises 128 code points in the range 0hex to 7Fhex, Extended ASCII comprises 256 code points in the range 0hex to FFhex, and Unicode comprises 1,114,112 code points in the range 0hex to 10FFFFhex.

If we want to extract the day, month and year from a date expressed as *dd/mm/yyyy* we can do the following

```
 # extract day, month, year
       if ($date =~ m!(\d\d)/(\d\d)/(\d\d\d\d)!) { # match dd/mm/yyyy format
           $day = $1;
           $month = $2;
           $year = $3;
       }
```

Note the use of the pattern match operator `m!!` to change the standard pattern delimiters. We can rewrite in a shorter form the previous code, taking advantage of the different behavior of the binding operator in scalar and list contexts.

In scalar context the binding operator returns a true or false value.

```
$answer = $date =~ m!(\d\d)/(\d\d)/(\d\d\d\d)!
```

Thus *$answer* equals to one or zero. In list context, however, the binding operator returns the list of matched values `($1, $2, $3, ...)`. Thus we can abbreviate the previous code as

```
($day,$month,$year) = ($date =~ m!(\d\d)/(\d\d)/(\d\d\d\d)!)
```

If the groupings in a regexp are nested, `$1` gets the group with the leftmost opening parenthesis, `$2` the next opening parenthesis, etc.

For more information: `man perlretut`.

5.5 Basic use of `fork` to launch a program

Apart from the `system` utility, a `Perl` script can launch child processes using the `fork` utility. Let's assume that we are interested in launching applications, called `fort_1` and `fort_2` from a script, but we are not interested in waiting for the end of the application. Using `fork` we can do the following in our script

```
defined(my $pid0 = fork) or die "Cannot fork: $!";
unless ($pid0) {
  # Child 0 process is here
  exec "fort_1";
  die "cannot exec fort_1: $!";
}
defined(my $pid1 = fork) or die "Cannot fork: $!";
unless ($pid1) {
  # Child 1 process is here
  exec "fort_2";
  die "cannot exec fort_2: $!";
}
print "Program output: \n";
# Parent processes are here
# script continues ...
waitpid($pid0, 0);
waitpid($pid1, 0);
```

Only the parent process has a non-zero value in `$pid0` and `$pid1` and skip the two `unless` conditionals. The program arrives to the `waitpid` function. This function waits for a particular child process to terminate and returns the pid of the deceased process. It is important to do so in order to get rid of zombie processes.

For more information: `man perlipc`.

5.6 `Perl` predefined variables. Some examples.

Apart from the ubiquitous `Perl` default variable, `$_` there is a large number of useful predefined variables. We give some examples of them in the following short codes.

1 `$.`

Current line number for the last filehandle accessed.

The following code displays each line of the file and the corresponding line number.

```
#
open(INPUT,"</etc/motd") or die "/etc/motd: $!";
#
while (<INPUT>) {
    print "Line $.: $_";
}
```

2 `$0`

Name of the program being executed.

The following code removes directories preceding the program name and stores it in a variable called `$prgname`

```
#
(my $prgname) = $0 =~ m#.*/(.+$)#;
#
```

For more information: `man perlvar`.

5.7 Using a named pipe for interprocess communication in `Perl`

A named pipe (or `fifo` file) can be used for interprocess communication between a parent process and a child process or children processes. Let's suppose that we forked and launched a couple of child processes 'Basic use of `fork` to launch a program' on the preceding page and we want to check whether each of the child processes has finished. Once the first process finishes we execute the `waitpid`. It is not efficient to directly execute `waitpid` because we do not know which of the processes will finish first.

A commented sample of code that manages to do so, launching a couple of child processes and waiting for each of them to finish is the following

```
#!/usr/bin/perl
#
# named pipe use for ipc example
#
# by Currix TM
#
use strict;
use warnings;
use POSIX qw(mkfifo);
#
# fifo definition
my $FIFOname = ".prgfifo";
unless (-p $FIFOname) { # Create the pipe if it doesn't exist
    unlink $FIFOname;
    mkfifo($FIFOname, 0700) or die "mkfifo in the current directory failed: $!";
}
#
my @pid;
#
print "This is the parent process before forking with pid $$\n";
#
defined ($pid[0]=fork) or die "Cannot fork (1): $!";
#
#
unless ($pid[0]) {
    print "fork1 pid: $pid[0]\n";
    print "fork1 ps: $$\n";
    sleep 10; # Sleeeeeeeping
    system "cat /etc/motd";
#   Child process ended. Write process number in the FIFO
    open (FIFO, ">$FIFOname") || die "can't write prgfifo: $!";
    print FIFO "$$";
    sleep 2;    # to avoid dup signals
    print "Exiting child 1\n";
    exit(0) # Remember to cleanly close the child process
} else {
    print "This is the parent process after forking 1 with pid: $pid[0]\n";
}
#
defined ($pid[1]=fork) or die "Cannot fork (2): $!";
#
unless ($pid[1]) {
    print "fork2 pid: $pid[1]\n";
    print "fork2 ps: $$\n";
    sleep 5; # Sleeeeeeeping
    system "cat /etc/fstab";
#   Child process ended. Write process number in the FIFO
    open (FIFO, ">$FIFOname") || die "can't write prgfifo: $!";
    print FIFO "$$";
    sleep 1;    # to avoid dup signals
    print "Exiting child 2\n";
    exit(0); # Remember to cleanly close the child process
```

```
} else {
    print "This is the parent process after forking 2 with pid: $pid[1]\n";
}
#
print "This are the pids from the parent process after forking: $pid[0], $pid[1]\n";
#
my $iprocess = 0;
open (FIFO, "<$FIFOname") || die "can't read prgfifo: $!";
while (1)  {
    my $kidpid = <FIFO>;
    if (defined $kidpid) {
 print "child process $kidpid ended\n";
 sleep 2; # To avoid dup signals again
 waitpid($kidpid, 0);
 last if ((++$iprocess) == 2);
    }
}
#
print "The two child processes have finished. Closing the parent process.\n";
#
unlink("$FIFOname"); # Remove the named pipe
```

For more information: `man perlipc` and references below.

5.7.1 References

1 Perldoc website (http://perldoc.perl.org/perlipc.html)

5.8 CperlMode in Emacs

The `CPerlMode` can be set as the standard mode for editing `Perl` adding the following line to the `.emacs` configuration file:

```
(defalias 'perl-mode 'cperl-mode)
```

To access the documentation about the mode use the `describe-mode` function by typing `C-h m` when in `CPerlMode`. When not in `CPerlMode` use `M-x describe-function RET cperl-mode` or `C-h f cperl-mode`.

5.8.1 References

1 Emacs wiki (http://www.emacswiki.org/emacs/CPerlMode)

5.9 Using Perl to benchmark code.

The `Benchmark` module included in the base `Perl` distribution includes a series of procedures to to benchmark running times of code.

Some of the available procedures are the following[2]

- `timethis`: run a chunk of code several times.
- `timethese`: run several chunks of code several times.
- `cmpthese`: print results of timethese as a comparison chart (*).
- `timeit`: run a chunk of code and see how long it goes (*).
- `countit`: see how many times a chunk of code runs in a given time (*).

The procedures marked with an asterisk (*) are not included by default and should be explicitly loaded.

Two of the most useful options are `timethese` and `cmpthese`.

The `timethese` procedure runs several chunks of code several times. The syntax is

[2] There are other possibilities. Check the References.

```
timethese($count, {
'Name1' => sub { ...code1... },
'Name2' => sub { ...code2... },
});
```

If the argument `$count` is a positive integer it gives the number of times the code is run, a negative intiger indicates the minimum number of CPU seconds to run[3]. The minimum in this case is 0.1 sec. If `$count` is zero a default value of 3 CPU seconds is assumed.

The output of `timethese` is an object that can be used as an input for `cmpthese`.

We apply this to the following example, comparing different ways of calculating the square of a number.

```
#!/usr/bin/perl
use strict;
use warnings;
use Benchmark qw( timethese cmpthese ) ;
my $x = 3.1;
my $CNT = -6;
my $r = timethese( $CNT, {
a => sub{$x*$x},
b => sub{$x**2},
c => sub{exp(2*log($x))}
} );
cmpthese $r;
$CNT=40_000_000;
$r = timethese( $CNT, {
a => sub{$x*$x},
b => sub{$x**2},
c => sub{exp(2*log($x))}
} );
cmpthese $r;
```

The procedures are run twice, the first with `$count=-6` and the second with `$count=40_000_000`.

In the first case the `timethese` output is the following

```
Benchmark: running a, b, c for at least 6 CPU seconds...
   a:  8 wallclock secs ( 7.07 usr +  0.00 sys =  7.07 CPU) @ 17313412.45/s (n=122405826)
   b:  7 wallclock secs ( 6.13 usr + -0.02 sys =  6.11 CPU) @ 12221032.41/s (n=74670508)
   c:  6 wallclock secs ( 6.39 usr +  0.00 sys =  6.39 CPU) @ 3914053.68/s (n=25010803)
```

In these case the real (wallclock) time is given, and also the distribution of the addition of the time spent by the user and the system to accomplish the CPU time goal[4]. In case the program spawns one or more children processes the `cusr` and `csys` times are also given. The number after the `@` symbol is the number of iterations per second and n is the total number of iterations. Thus the larger the better in these two last cases. The first version of the code can be concluded to be more efficient. This is more easily denoted using the `cmpthese` output. It gives in increasing order the number of iterations per second and the percentage of improvement (positive) or worsening (negative value) compared to the other options.

```
       Rate    c    b    a
c  3914054/s   -- -68% -77%
b 12221032/s 212%   -- -29%
a 17313412/s 342%  42%   --
```

In this case the codes are ordered starting on the slowest (c in this case), giving in Rate the iterations per second and the percentages of comparison of the rate with the rate of the other codes under evaluation.

If the `$count` argument is positive the code is executed the number of times indicated by the argument. If this number is high enough the results should coincide with the previously obtained.

```
enchmark: timing 40000000 iterations of a, b, c...
   a:  2 wallclock secs ( 1.90 usr + -0.01 sys =  1.89 CPU) @ 21164021.16/s (n=40000000)
   b:  3 wallclock secs ( 3.25 usr +  0.00 sys =  3.25 CPU) @ 12307692.31/s (n=40000000)
   c: 10 wallclock secs (10.16 usr +  0.00 sys = 10.16 CPU) @ 3937007.87/s (n=40000000)
        Rate    c    b    a
c  3937008/s   -- -68% -81%
b 12307692/s 213%   -- -42%
a 21164021/s 438%  72%   --
```

The output will vary even for the same box, and several runnings are sometimes necessary to get a final answer. Also the output vary from box to box. If the same code is run in a different computer we obtain

[3]CPU seconds is, in `UNIX` terms, the user time plus the system time of the process itself, as opposed to the real (wallclock) time and the time spent by the child processes.

[4]For an explanation of the different times reported check this link Process time (http://en.wikipedia.org/wiki/Process_time)

```
Benchmark: running a, b, c for at least 6 CPU seconds...
   a:  7 wallclock secs ( 6.18 usr +  0.00 sys =  6.18 CPU) @ 19332920.23/s (n=119477447)
   b:  7 wallclock secs ( 7.25 usr +  0.00 sys =  7.25 CPU) @ 10521698.76/s (n=76282316)
   c:  8 wallclock secs ( 6.96 usr +  0.00 sys =  6.96 CPU) @ 4018543.53/s (n=27969063)
        Rate    c    b    a
c  4018544/s   -- -62% -79%
b 10521699/s 162%   -- -46%
a 19332920/s 381%  84%   --
Benchmark: timing 40000000 iterations of a, b, c...
   a:  2 wallclock secs ( 0.77 usr +  0.00 sys =  0.77 CPU) @ 51948051.95/s (n=40000000)
   b:  2 wallclock secs ( 2.40 usr +  0.00 sys =  2.40 CPU) @ 16666666.67/s (n=40000000)
   c:  9 wallclock secs ( 9.40 usr +  0.00 sys =  9.40 CPU) @ 4255319.15/s (n=40000000)
        Rate     c     b     a
c  4255319/s    --  -74%  -92%
b 16666667/s  292%    --  -68%
a 51948052/s 1121%  212%    --
```

5.9.1 References

1 Perldoc Benchmarc Entry (http://perldoc.perl.org/Benchmark.html)

2 Benchmarking in techrepublic (http://www.techrepublic.com/article/benchmarking-perl-scripts-with-b
5278558)

3 Process time (http://en.wikipedia.org/wiki/Process_time)

5.10 Accessing recursively files and directories in `Perl`

Added on November 22nd, 2012.

The easiest way in `Perl` to access files and directories recursively is making use of the `File::Find` module. For example let's assume that we want to, recursively, change the permissions of a given directory contents in such a way that files have `rw-r----` permision and directories `rwxr-x--`.

We can do this with the following script, than makes use of the `File::Find` module.

```
#!/usr/bin/perl
#
# script to process recursively a directory.
# by Currix TM.
use strict;
use warnings;
#
use File::Find;
#
sub process_files {
  my $permission_dir = 0750;
  my $permission_file = 0740;
  if (-d $_) {
    #print "processing dir $_\n";
    chmod $permission_dir, $_;
  } elsif (-f $_) {
    #print "\tprocessing file $_\n";
    chmod $permission_file, $_;
  }
}
@ARGV = qw(.) unless @ARGV;
find(\&process_files, @ARGV);
```

Notice that the `chmod` function in `Perl` needs that the permission are expressed in octal values. Note also de lack of apostrophes in the permisision variables definition.

Chapter 6

Python Tips and Templates

6.1 Creating a vector of random data

Created on April 12th, 2015.

The following snippet of `Ipython` code computes a vector, called *vectorn*, with 10 elements of normally distributed random data

```
import numpy as np

np.random.randn(10)
Out[2]:
array([-0.20490308, -0.39783301, -0.6802615 , -0.57939922, -0.10054472,
       -0.20376277, -1.48068811,  0.98628113, -0.79514919, -0.09896364])

vectorn = np.random.randn(200)
```

6.2 Logical comparison between two boolean vectors

Created on May 06th, 2015.

The following snippet of `Ipython` code defines two vector, called *vectorA* and *vectorB*, with 20 elements of normally distributed random data each and, using the `NumPy` function `logical_and` check the occurrences where the corresponding elements of the two vectors are larger than zero

```
import numpy as np

vectorA = np.random.randn(20)
vectorB = np.random.randn(20)

boolvec = np.logical_and(vectorA > 0, vectorB > 0)

vectorA[boolvec]
Out[47]: array([ 0.39058535,  1.0062992 ])

vectorB[boolvec]
Out[48]: array([ 0.87795544,  0.59063525])
```

6.3 Creating a loop iterating on a list and the list index

Created on April 16th, 2015.

The following snippet of `Ipython` code uses as a starting point a a vector, called *vn*, with 10 elements of uniformly distributed random data in the interval $[0,1)$ and in a loop, using as iterator a pair *index, value*, we build a symmetric matrix *A* such that *A_ij = vn_i*vn_j*.

```
import numpy as np
##
vn = np.random.rand(20)
##
A_matrix = np.zeros((20,20))
##
for i, ival in enumerate(vn):
    for j, jval in enumerate(vn):
        A_matrix[i,j] = ival*jval
```

It can be easily optimized not computing the full matrix but the upper or lower diagonal and adding to its transpose (beware of double counting diagonal elements).

6.4 Adding a column to a `Pandas` dataframe

Created on April 21st, 2015.

If we have a `Pandas` dataframe for example, the following one, called `df0`

```
import numpy as np
import pandas as pd

vectorn = np.random.rand(20)
df0 = pd.DataFrame(data=vectorn, columns = ["s0"])
```

We can now add a second column of random data using the `pd.Series` command; the column is labeled `s1`

```
vectors = np.random.rand(20)
df0["s1"] = Series(data=vectors, index = index.df0)
```

6.5 Getting the maximum component of a vector and its index in `numpy`

Created on April 12th, 2015.

The following snippet of `Ipython` code computes a vector, called *vectorn*, with 200 elements of uniformly distributed random data in the interval $[0,1)$.

```
import numpy as np

vectorn = np.random.rand(200)

max_val, max_index = vectoru.max(), vectoru.argmax()

max_val
Out[87]: 0.99652709220203461

max_index
Out[88]: 117
```

6.6 Some easy examples of offset-aware times with `pytz`

Created on May 13th, 2015.

Dealing with imezones and the associated DST (daylight saving times) can cause a more than serious headache. Some (very limited) examples of their use.

Let's assume that we have two strings: `stdate1 = "20/03/2015 12:22"` and `stdate1 = "23/03/2015 22:22"` and we want to parse them to a `datetime` object. This is done as follows:

```
from dateutil.parser import parse
stdate1 = "20/03/2015 12:22"
stdate2 = "23/03/2015 22:22"

date1 = parse(stdate1, dayfirst=True)

date2 = parse(stdate2, dayfirst=True)

date1
Out[6]: datetime.datetime(2015, 3, 20, 12, 22)

date2
Out[7]: datetime.datetime(2015, 3, 23, 22, 22)

date2-date1
Out[8]: datetime.timedelta(3, 36000)
```

At this point we have offset-naive times. If we want to transform to a given time zone, e.g. `CET` then we use

```
import pytz

cet_tz = pytz.timezone("CET")

cet_date1 = cet_tz.normalize(cet_tz.localize(date1))
cet_date2 = cet_tz.normalize(cet_tz.localize(date2))

cet_date1
Out[12]: datetime.datetime(2015, 3, 20, 12, 22, tzinfo=<DstTzInfo 'CET' CET+1:00:00 STD>)

cet_date2-cet_date1
Out[14]: datetime.timedelta(3, 36000)
```

We can now transform these time data to `UTC`

```
utc_tz = pytz.timezone('UTC')

utc_date1 = cet_date1.astimezone(utc_tz)
utc_date2 = cet_date2.astimezone(utc_tz)

utc_date2 - utc_date1
Out[22]: datetime.timedelta(3, 36000)
```

We can transform directly to `UTC` from the initially parsed variables

```
UTC_date1 = utc_tz.normalize(utc_tz.localize(date1))

UTC_date1
Out[28]: datetime.datetime(2015, 3, 20, 12, 22, tzinfo=<UTC>)

UTC_date1 - utc_date1
Out[27]: datetime.timedelta(0, 3600)
```

These functions can be applied on lists using lambda functions.

6.7 Creating a panel array of plots with `Matplotlib`

Created on April 12th, 2015.

The following snippet of code uses a vector of length 200 with random normally distributed data (see 'Creating a vector of random data' on page 51) and plot in four panels the data, their cumulative sum, a histogram with the data, and the sum of the data to a quadratic function.

```
import numpy as np
from matplotlib import pyplot

fig,axes = pyplot.subplots(2,2) # Define plot of 2x2 panels

axes[0,0].plot(vectorn,"k-o")
Out[36]: [<matplotlib.lines.Line2D at 0x7f827af2a510>]

axes[0,1].plot(vectorn.cumsum(),"k--")
Out[37]: [<matplotlib.lines.Line2D at 0x7f827af2a1d0>]

axes[1,0].hist(vectorn,bins=30,color="r",alpha=0.3)
Out[38]:
(array([  3.,   0.,   2.,   2.,   4.,   2.,   6.,   4.,  10.,  12.,   8.,
         10.,  13.,  14.,  16.,  14.,   9.,  10.,   9.,  13.,   8.,   5.,
          8.,   3.,   5.,   6.,   0.,   1.,   1.,   2.]),
 array([-2.41379287, -2.24330459, -2.0728163 , -1.90232801, -1.73183972,
        -1.56135143, -1.39086315, -1.22037486, -1.04988657, -0.87939828,
        -0.70891   , -0.53842171, -0.36793342, -0.19744513, -0.02695684,
         0.14353144,  0.31401973,  0.48450802,  0.65499631,  0.8254846 ,
         0.99597288,  1.16646117,  1.33694946,  1.50743775,  1.67792603,
         1.84841432,  2.01890261,  2.1893909 ,  2.35987919,  2.53036747,
         2.70085576]),
 <a list of 30 Patch objects>)

axes[1,1].scatter(np.arange(200),0.01*np.arange(200)**2+10*vectorn)
Out[39]: <matplotlib.collections.PathCollection at 0x7f827aedae90>

pyplot.show()
```

6.8 Creating a panel array of plots with common axes using `Matplotlib`

Created on April 12th, 2015.

The following snippet of code add different vectors of length 200 with random normally distributed data (see 'Creating a vector of random data' on page 51) to a parabollic function (mimicking experimental errors in a object free fall) and plot the results in four panels, with common abscyssa and ordinate axes, and controlling the spacing between the panels.

```
import numpy as np
from matplotlib import pyplot

vectorn = np.random.randn(100)

result1 = 0.5*9.8*time_grid**2 + 2*vectorn

result2 = 0.5*9.8*time_grid**2 + 4*vectorn

result3 = 0.5*9.8*time_grid**2 + 8*vectorn

result4 = 0.5*9.8*time_grid**2 + 16*vectorn

fig,axes = pyplot.subplots(2,2,sharex=True,sharey=True)

axes[0,0].plot(result1,"k-o")
Out[85]: [<matplotlib.lines.Line2D at 0x7f827aa7ae10>]

axes[0,1].plot(result2,"k-o")
Out[86]: [<matplotlib.lines.Line2D at 0x7f827aa7ae90>]

axes[1,0].plot(result3,"k-o")
Out[87]: [<matplotlib.lines.Line2D at 0x7f827aa7a4d0>]

axes[1,1].plot(result4,"k-o")
Out[88]: [<matplotlib.lines.Line2D at 0x7f827aaaff10>]

pyplot.subplots_adjust(wspace=0,hspace=0)

pyplot.show()
```

6.9 Combining several plots in a figure

Created on June 10th, 2015.

The following snippet of `Ipython` code computes three vectors, called *vector1*, *vector2*, and *vector3*, with 100 elements of normally distributed random data with the same mean value (2) and different standard deviations (0.1, 0.2, and 0.4). We then plot the three vectors in a single graph controlling the line styles and labels and ticks font sizes.

```
import numpy as np

meanval = 2
vector1 = np.random.normal(loc = meanval, scale = 0.2, size = 100)
vector2 = np.random.normal(loc = meanval, scale = 0.4, size = 100)
vector3 = np.random.normal(loc = meanval, scale = 0.8, size = 100)

ax = pyplot.subplot(111)

ax.plot(vector1,"o--b",lw=3)
ax.plot(vector2,"x:r",lw=2)
ax.plot(vector3,"g",lw=2)

ax.set_xlabel(r'X axis Label (a.u.)',fontsize=16)
ax.set_ylabel(r'Y axis Label $v_1, v_2, v_3$',fontsize = 16)
pyplot.setp(ax.get_xticklabels(), fontsize=14)
pyplot.setp(ax.get_yticklabels(), fontsize=14)
```

Chapter 7

Text Processing and Formatting

7.1 `pdftk` application examples

Updated on December 1st, 2011.

We present several examples of transforming `PDF` files using the program `pdftk`, a powerful and simple application to work with `PDF` files.

1 Removing pages. If we have a file named `text.pdf` with ten pages, the following commands transform the file removing certain pages and saving the transformed output in file `out_text.pdf`:

- Remove first two pages:

```
$ pdftk text.pdf cat 3-10 output out_text.pdf
```

- Remove pages 2, 3, and 6:

```
$ pdftk text.pdf cat 1 4-5 7-10 output out_text.pdf
```

- Merge `PDF` files.

```
$ pdftk 1.pdf 2.pdf 3.pdf 4.pdf cat output file1234.pdf
```

  This command merges four files into a final file `file1234.pdf`.

- Split the pdf file into pages. If you want to create a pdf including only the first page of file `jpcA.pdf`

```
pdftk jpcA.pdf cat 1 output jpcA_p1.pdf
```

  In case you want to extract pages 1 to 3, leave out pages 4 and 5, and include the rest of the pages you can execute

```
pdftk jpcA.pdf cat 1-3 6-end output jpcA_partial.pdf
```

- (To Do) Decrypt input or encrypt output.
- (To Do) Generate FDF data stencil from PDF forms.
- (To do) Apply a background watermark or a foreground stamp.

7.2 Merge two `PostScript` or `PDF` files

Updated on December 1st, 2011.

We explain an alternative way to merge `PDF` files, to avoid using `pdftk`. This is also applicable to `PostScript` files.

You can also merge two `PostScript` or `PDF` files using `gs`

```
$gs -q -dNOPAUSE -dBATCH -sDEVICE=pswrite \
       -sOutputFile=bla.ps -f foo1.ps foo2.ps

$gs -q -dNOPAUSE -dBATCH -sDEVICE=pdfwrite \
       -sOutputFile=bla.pdf -f foo1.pdf foo2.pdf
```

7.2.1 References

1 Debian Reference Guide (http://qref.sourceforge.net/Debian/reference/ch-tips.en.html#s8.6.24)

7.3 Include *et al.* in bibtex

You can include the term *et al.* to replace a long set of authors in a bibtex reference using and others. Thus, if you want to include only the first two authors and replace the rest by others, the bibtex author field should be written as

```
author =    {First Author and Second Author and others},
```

7.4 Include bibliographical info in each chapter using LaTeX

You can include the bibliography after each chapter in LaTeX using the package chapterbib. You can even change the citation style for each chapter. In order to do so include at the beginning of the main tex source file the line

```
\usepackage{chapterbib}
```

Then at the end of each chapter include

```
\bibliographystyle{alpha}
\bibliography{texfilename}
```

For each chapter you can change the bibliography style if necessary. If you are using \include{*filename* do not forget to run bibtex to each of the included files possessing a bibliography entry.

7.5 Comment paragraphs in LaTeX

In order to comment one or several paragraphs in a LaTeX source file instead of making use of the % character you can use the verbatim package including at the beginning of your tex file

```
\usepackage{verbatim}
```

And then you can simply comment several lines of the file making use of \begin{comment} and \end{comment}

```
\begin{comment}
Commented fragment ...
\end{comment}
```

7.6 Installing appropriately a LaTeX package or style.

The way to install a LaTeX package in the right path is shown. The first step is to look for the place where the package should be installed.

```
$ kpsewhich -expand-var "\$TEXMFLOCAL"
/usr/local/share/texmf
```

Unpack the latex package that is going to be installed and copy the directories (bibtex/, doc/, tex/ ...) to the previous path, /usr/local/share/texmf and rehash the database.

```
$ sudo cp -r doc tex source bibtex  /usr/local/share/texmf
$ texhash
texhash: Updating /usr/local/share/texmf/ls-R...
texhash: Updating /var/lib/texmf/ls-R-TEXMFMAIN...
texhash: Updating /var/lib/texmf/ls-R-TEXLIVE...
texhash: Updating /var/lib/texmf/ls-R...
texhash: Done.
```

The style is now ready and installed.

7.7 Change the selected language in DebianDoc files

In order to change the selected language for a DebianDoc sgml file the option -l is employed. This affects the different tags and labels in the text.

For example, assuming that the default language in the system is English and we want to generate a version in Spanish in ASCII format of the file test.sgml we should employ

```
debiandoc2text -les_ES test.sgml
```

In this case we suppose that the locale es_ES is available[1].

7.8 Avoid Page %d may be too complex to print errors.

When preparing pdf files from LaTeX files sometimes the dvipdf script takes a very long time to complete the pdf file and outputs some errors like in the following example.

```
$ dvipdf test.dvi
Page 1 may be too complex to print
Page 3 may be too complex to print
Warning:  no %%Page comments generated.
```

Moreover, when the final pdf file is opened with a viewer there is a long series of error messages

```
$ xpdf test.pdf
Error: Bad bounding box in Type 3 glyph
Error: Bad bounding box in Type 3 glyph
Error: Bad bounding box in Type 3 glyph
Error: Bad bounding box in Type 3 glyph
Error: Bad bounding box in Type 3 glyph
Error: Bad bounding box in Type 3 glyph
```

This is due to the inclusion of the command \usepackage[T1]{fontenc} in the input file. This is convenient when writing in Spanish because in this way you have access to special characters (as accented letter).

The solution to the problem consists in making use of the latin-modern fonts. The heading of the tex file should include

```
\usepackage[T1]{fontenc}
\usepackage{lmodern}
```

The Debian package that includes this fonts should be installed and its name is lmodern.deb.

7.9 Diverse LaTeX lists

LaTeX distinguishes between three different list environments: enumerate, itemize, and description. Each environment provides four levels, which implies you can have nested lists of up to four levels. The description of the three environments is the following.

1 *Enumerate*

The syntax in this case is

```
\begin{enumerate}
\item ...
\end{enumerate}
```

The enumerate environment permits the definition of numbered lists. If you like to change the appearance of the enumerator, the simplest way to change is to use the enumerate-package, giving you the possibility to optionally choose an enumerator.

[1] You can check the available locales in the file /etc/locale.gen.

```
\usepackage{enumerate}
...
\begin{enumerate}[I]%for capital roman numbers.
\item
\end{enumerate}

\begin{enumerate}[(a)]%for small alpha-characters within brackets.
\item
\end{enumerate}
```

2 *Itemize*

Itemization is probably the mostly used list in LaTeX. It also provides four levels.

```
\begin{itemize}
\item ...
\end{itemize}
```

The bullets marking each item can be changed for each level using the following command:

```
\renewcommand{\labelitemi}{$\bullet$}
\renewcommand{\labelitemii}{$\cdot$}
\renewcommand{\labelitemiii}{$\diamond$}
\renewcommand{\labelitemiv}{$\ast$}
```

Amongst the more commonly used symbols are `$\bullet$`, `$\cdot$`, `$\diamond$`, `$-$`, `$\ast$`, and `$\circ$`.

3 *Description*

The description list is very handy if you need to explain notations or terms. Its neither numbered nor bulleted. The user can define the string marking each item.

```
\begin{description}
\item[] ...
\end{description}
```

In the three environments the space between different items can be controlled with the `\itemsep` command that can only be added just after `begin`

```
\begin{itemize}\itemsep2pt
\item
\end{itemize}
```

7.9.1 References

1 LaTeX Lists Environments (http://texblog.wordpress.com/2008/10/16/lists-enumerate-itemize-descrip

7.10 Use Unicode encoding in LaTeX

In order to use `utf` encoding in LaTeX the following line should be added to the tex file

```
\usepackage[utf8]{inputenc}
```

If this option does not work use `utf8x` instead `utf8`.

7.11 Use color in LaTeX

Added on November 30th, 2011.

In order to use colors in your in LaTeX the following line should be added to the file preamble

```
\usepackage{color}
```

The text can be colored in different ways

```
\textcolor{declared-color}{text}
{black text\color{declared-color} text}
```

where declared-color is a color that was defined before by `\definecolor`. You can change the background color of the whole page by:

```
\pagecolor{declared-color}
```

7.11.1 References

1 LaTeX Color in LaTeX Wiki Book (http://en.wikibooks.org/wiki/LaTeX/Colors#Adding_the_color_package)

7.12 Easy way of defining smaller margins in LaTeX

Added on January 20th, 2012.

Updated on February 1st, 2012

In order to define a LaTeX document with smaller margins than the default without tampering too much with measures and sizes the `fullpage` package can be used adding the your document header

```
\usepackage[options]{fullpage}
```

Possible options for this package are

1 `in`: (*default*) margins set to 1 in.

2 `cm`: margins set to 1.5 cm.

3 `plain`: (*default*) selects plain page style.

4 `empty`: neither headers nor footers.

5 `headings`: both headers and footers.

6 `myheadings`: both headers and footers.

```
\usepackage[cm]{fullpage}
```

With `A4` papersize another possibility is

```
\usepackage{a4wide}
```

7.13 Using the same footnote mark in LaTeX

Added on February 5th, 2012.

Sometimes in LaTeX it is necessary to make reference to the same footnote several times in a page. The following syntax allows for this

```
Text that has a footnote\footnote{This is the footnote} looks like this. Later text referring to same footnote\footnotemark[\value{fo
```

It is important to take into account that this doesn't work if there are other footnotes between the first reference and any of the other *duplicates*.

7.13.1 References

1 LaTeX wikibooks (http://en.wikibooks.org/wiki/LaTeX/Formatting#Footnotes)

7.14 Using the euro symbol in LaTeX

Added on February 7th, 2012.

The euro currency symbol in LaTeX is added making use of the package eurosym, which is part of all the major GNU/Linux distributions. The package has to be loaded in the document header

```
\usepackage{eurosym}
```

Then there are two possible ways of including the euro currency symbol.

```
The book is 10 \euro.
The book is \euro 10.
The book is \EUR{10}.
\textbf{The book is \EUR{10}.}
\textit{The book is \EUR{10}.}
```

7.15 Changing pages to landscape orientation in LaTeX texts

Added on November 16th, 2012.

The occurrence of a large table or figure in LaTeX sometimes hamper the display in the default portrait orientation. This can be solved changing one or various pages to landscape orientation.

The geometry package allows to change the full document to landscape orientation adding to the document header

```
\usepackage[landscape]{geometry}
```

The lscape package allows to change to landscape orientation a section of the document. To do so add the following line to the document header

```
\usepackage{lscape}
```

And whenever it is needed to switch to landscape orientation, e.g. to include a large table or figure the region affected by the change is defined as

```
\begin{landscape}
... table or figure here ...
\end{landscape}
```

This is specially suited for printing. To change also the orientation in the pdf file and for better screen readibility use the package pdflscape in the header instead of lscape

```
\usepackage{pdflscape}
```

And proceed as before

```
\begin{landscape}
... table or figure here ...
\end{landscape}
```

7.15.1 References

1 Landscape in LaTeX (http://texblog.org/2007/11/10/landscape-in-latex)

7.16 Including single column figures or tables in a double column LaTeX document

Added on November 18th, 2012.

Sometimes when writing a two-column document the occurrence of a large table or figure in LaTeX forces its display in a single-column way. This can be solved for figures and tables using the -* variant

```
\begin{table*}

\end{table*}
\begin{figure*}

\end{figure*}
```

In this way figures and tables will occupy the full page.

Chapter 8

Graphic Edition

8.1 Include greek characters in `inkscape`

Find the unicode number corresponding to the greek character you want to include. Tables of codes can be found in Unicode Charts (http://unicode.org/charts/PDF/U0370.pdf). For example, the unicode for the alpha symbol is `03b1`.

In a text cell type `Ctrl-u`, include the unicode key of the greek letter and press enter.

8.2 Include greek characters in `Gimp`

Find the unicode number corresponding to the Greek character you want to include. Tables of codes can be found in Unicode Charts (http://unicode.org/charts/PDF/U0370.pdf). For example, the unicode for the alpha symbol is `03b1`.

In a text cell type `Ctrl-Shft-u`, include the unicode key of the Greek letter and press enter.

8.3 Change the text baseline in `inkscape`

In order to include sub and super-indexes in `inkscape` you can change the text base line in a text cell using the keys:

`Alt-Up` and `Alt-Down`.

8.4 Use `ImageMagick` to transform graphic files

Updated on July 06th, 2014.

You can easily transform a graphic file from the console using the tool `convert` from the `ImageMagick` suite. Imagine for example that we start with the file `foto_0.png`.

```
$ identify foto_0.png
foto_0.png PNG 1209x1710 1209x1710+0+0 DirectClass 8-bit 2.63038mb
```

We have made use of the command `identify`, that is also part of the `ImageMagick` suite and describes the format and characteristics of image files. There are several ways to resize the file. The common syntax is `convert -resize` *geometry*. From the many possible ways of expressing the geometry change we present three useful options:

1 *scale*`%`: Height and width scaled by the specified percentage *scale*.

2 *width*: Width fixed to *width* and height automatically rescaled to conserve aspect ratio.

3 `x`*height*: Height fixed to *height* and width automatically rescaled to conserve aspect ratio.

We present examples of the three possibilities[1]:

```
$ identify foto_0.png
foto_0.png PNG 1209x1710 1209x1710+0+0 DirectClass 8-bit 2.63038mb
$ convert -depth 8 -resize 50% foto_0.png foto_1.png
$ identify foto_1.png
foto_1.png PNG 605x855 605x855+0+0 DirectClass 8-bit 681.391kb
$ convert -depth 8 -resize 1024 foto_0.png foto_2.png
$ identify foto_2.png
foto_2.png PNG 1024x1448 1024x1448+0+0 DirectClass 8-bit 1.82886mb
$ convert -depth 8 -resize x800 foto_0.png foto_3.png
$ identify foto_3.png
foto_3.png PNG 566x800 566x800+0+0 DirectClass 8-bit 601.271kb
```

Apart from resizing images we can also transform from an image format to a different one. For example, to convert from encapsulated postscript (eps) to png format:

```
$ identify bsplot_N40.eps
bsplot_N40.eps PS 613x661 613x661+0+0 16-bit DirectClass 20.4KB 0.000u 0:00.000
$ convert bsplot_N40.eps bsplot.png
$ identify bsplot.png
bsplot.png PNG 613x661 613x661+0+0 8-bit PseudoClass 6c 8.64KB 0.000u 0:00.000
```

8.4.1 References

1 Linux Journal, issue 185, Sept. 2009 (http://www.linuxjournal.com/article/10531)

8.5 Enhancing colors and resolution in Gimp

A Gimp filter that improves and sharpen lines when colors are too weak can be found in

```
Filters -> Enhance -> Unsharp Mask
```

8.6 Remove the background of an image in Gimp

In order to remove the background of an image using Gimp a possible (easy) recipe is the following-,

1 Using the *lasso* tool (also known as *Free Select Tool*) make a rough selection around the image that you want to remove the background from.

2 Activate the quick mask tool by selecting the little square at the bottom left side of the main image window. Using this tool the selection can be refined.

3 Using the pencil tool shade the background more precisely around your image. If you make a mistake you can either use Ctrl Z to undo your last action or change the pencil to white which will remove the mask.

4 Toggle the quick mask off, copy the image, and paste as new. This will result in a background-free copy of your image.

5 To refine the final result, add alpha to the selection (from the layers tool). Then, from the Select menu: (a) invert the selection, (b) feather the selection by 1 or 2 pixels, (c) Ctrl + k to clear, and (d) Ctrl + Shift + A to unselect.

6 Save your image.

8.7 Take a screenshot using Gimp

In order to take a screenshot of the full display, a window, or part of a window launch Gimp and select

```
File -> Create -> ScreenShot...
```

It is convenient to fix a time delay to be able to select the right window to grab the region of interest.

[1] Note that we added the option -depth 8 to avoid the image depth to be changed to 16, the default system value.

8.8 Change the color of markers in `inkscape`

Added on July 06th, 2014.

In order to change the color of markers (e.g. arrows end and start) in `inkscape` to coincide with the color of the stroke of the object they belong to the following effect should be enabled.

`Extensions > Modify Path > Color Markers to Match Stroke`

If

8.8.1 References

1 Inkscape FAQ (http://wiki.inkscape.org/wiki/index.php/Frequently_asked_questions#How_do_I_change_the_color_of_markers_.28e.g._arrow_ends.29.3F)

Chapter 9

Miscelanea

9.1 Compute the average from the results of `Moodle` test grading

The starting point is a set of txt files with the grading obtained by students in `Moodle`. This set should include all students, not only the students that have participated in the test. The format of these files[1] is like the one in the following example

```
XXX XXXXXX XX  29 de January de 2009, 11:27 17 minutos 40 segundos 7
```

For those students that did not participate in the test, the format is the following

```
XXX XXXXXX XX       -       -       -
```

The name of the files with the results are *test_i.txt* with *i = 1, 2, 3*. The number of files is not limited in principle. The following perl oneliner extract the required info, computes the average students' grade, and save the output in a `CSV` file format

```
perl -e 'while (<>) {
$v{$1}+=$2 if (/^(.*)\b\s+\d+\s+de.*s\s+(\d+).*$/);
$v{$1}+=0 if (/^(.*)\b\s+-\s+-\s+-.*$/);};
foreach (sort keys(%v)) { print "$_ ;",$v{$_}/3,"\n"}' \
test_1.txt test_2.txt test_3 .txt  > grades.csv
```

9.2 Splitting in two lines a cell content in `Openoffice/simpress`

From `OpenOffice Calc`'s help pages

Writing multi-line text:

1. The Ctrl+Enter key combination inserts a manual line break. This shortcut only works directly in the cell, not in the input line.
2. If you want the text to automatically break at the right border of the cell, proceed as follows: select all the cells where you want the text to break at the right border, and in `Format - Cells - Alignment`, mark the `Automatic line break` option and click OK.

9.3 Solving `BADSIG` errors in `UBUNTU (Oneiric Ocelot)`

Added on January 18th, 2012.

In `Oneiric Ocelot`, for reasons unknown to me, errors concerning the signature key plage the software update process. For example, the output obtained updating with `apt-get` with this problem is

[1] The `Moodle` output is formatted using `UTF-8`, thus it may be necessary to convert, using `emacs`, to `latin-1` formatting.

```
$ sudo apt-get update
Ign http://archive.canonical.com oneiric/partner Translation-en_US
Ign http://extras.ubuntu.com oneiric/main Translation-en
Hit http://sunsite.rediris.es oneiric-updates Release
Ign http://archive.canonical.com oneiric/partner Translation-en
Hit http://sunsite.rediris.es oneiric-backports Release
Hit http://sunsite.rediris.es oneiric-security Release
Ign http://sunsite.rediris.es oneiric-security Release
Hit http://sunsite.rediris.es oneiric/main Sources
Hit http://sunsite.rediris.es oneiric/restricted Sources
Hit http://sunsite.rediris.es oneiric/universe Sources
Hit http://sunsite.rediris.es oneiric/multiverse Sources
Hit http://sunsite.rediris.es oneiric/main i386 Packages
Hit http://sunsite.rediris.es oneiric/restricted i386 Packages
Hit http://sunsite.rediris.es oneiric/universe i386 Packages
Hit http://sunsite.rediris.es oneiric/multiverse i386 Packages
Hit http://sunsite.rediris.es oneiric/main TranslationIndex
Hit http://sunsite.rediris.es oneiric/multiverse TranslationIndex
Hit http://sunsite.rediris.es oneiric/restricted TranslationIndex
Hit http://sunsite.rediris.es oneiric/universe TranslationIndex
Hit http://sunsite.rediris.es oneiric-updates/main Sources
Hit http://sunsite.rediris.es oneiric-updates/restricted Sources
Hit http://sunsite.rediris.es oneiric-updates/universe Sources
Hit http://sunsite.rediris.es oneiric-updates/multiverse Sources
Hit http://sunsite.rediris.es oneiric-updates/main i386 Packages
Hit http://sunsite.rediris.es oneiric-updates/restricted i386 Packages
Get:2 http://sunsite.rediris.es oneiric-updates/universe i386 Packages [82.3 kB]
99% [2 Packages bzip2 0 B] [Waiting for headers] [Waiting for headers]
bzip2: Data integrity error when decompressing.
        Input file = (stdin), output file = (stdout)

It is possible that the compressed file(s) have become corrupted.
You can use the -tvv option to test integrity of such files.

You can use the 'bzip2recover' program to attempt to recover
data from undamaged sections of corrupted files.

Get:3 http://sunsite.rediris.es oneiric-updates/multiverse i386 Packages [4,976 B]
```

In order to fix this error follow the instructions below. *Notice that this implies moving directories and removing recursively as superuser. Be extremely cautious...*

In first place you open an interactive session as superuser and download the public key from the `Ubuntu` key server.

```
$ sudo -i
# apt-key adv --recv-key --keyserver keyserver.ubuntu.com 40976EAF437D05B5
```

Then proceed to make a backup copy of some directories and remove part of the stored info (remember, do not take the name of root in vain...)

```
# cp -arf /var/lib/dpkg /var/lib/dpkg.backup
# cp /var/lib/dpkg/status-old /var/lib/dpkg/status
# cp /var/lib/dpkg/available-old /var/lib/dpkg/available
# rm -rf /var/lib/dpkg/updates/*
# rm -rf /var/lib/apt/lists
```

Create the `lists/partial` directory, clean the cache and update again.

```
# mkdir -p /var/lib/apt/lists/partial
# apt-get clean
# apt-get update
```

This should be enough to fix the `BADSIG` problem. Hopefully.

Chapter 10

Translations of this document

The present document has been translated from (not-so-good) English to other languages:

1 Translation to Spanish of Chapter 3 (Perl Language) by MarÃa Ramos. (http://www.webhostinghub.com/support/es/misc/instalacion-de-perl)

2 Translation to Frech of the full document by Kate Bondareva. (http://www.autoteiledirekt.de/science/certains-mini-howtos-dinteret)