

# ficheros del home

## .alias

```
# $Id: .alias,v 1.1 1999/11/27 23:01:40 sage Exp $
# Luis Francisco González <luisgh@debian.org> based on that of Vadik Vygonets
# Please check /usr/doc/tcsh/examples/alias to see other possible values.

# recopilación de alias
# opciones LS
alias l.='ls -d .* --color=tty'
alias ll='ls -l --color=tty'
alias ls='ls --color=tty'
alias lsd='ls -d */'
# evitar errores (confirmación al sobrescribir)
alias cp='cp -i'
# alias mv='mv -i'
# alias rm='rm -i'

# abreviaturas
alias rd='rmdir'
alias md='mkdir'

# errores
alias cd.='cd ..'
alias moer='more'
alias mroe='more'

# funcionalidades
alias cdo='cd $OLDPWD'
users='cut -d: -f1 /etc/passwd | sort'
users2='getent passwd|cut -d: -f1|sort'
```

## .bash\_profile

```
# ~/.bash_profile: executed by bash(1) for login shells.

# quita el permiso de escritura al others
umask 002

#prompt por defecto
PS1='[\h]$ '

# prompt de color en 2 lineas
if [ -f ".prompt" ]; then
    export PS1=`cat .prompt`
fi

if [ -f ".bashrc" ]; then
    source ~/.bashrc
fi
```

```
# variables de entorno
export EDITOR="vim"
export LANG=en_US.ISO-8859-15

# echo "> .bash_profile"
```

## .bashrc

```
# ~/.bashrc: executed by bash(1) for non-login shells.
source ~/.alias

# funciones interesantes
source ~/.functions

# exports
source ~/.export

# si dejo el ECHO da problemas con scp
# echo "> .bashrc"
```

## .export

```
# utilizar LESS en lugar de MAN y utilizar colores
export LESS_TERMCAP_mb=$'\E[01;31m'
export LESS_TERMCAP_md=$'\E[01;37m'
export LESS_TERMCAP_me=$'\E[0m'
export LESS_TERMCAP_se=$'\E[0m'
export LESS_TERMCAP_so=$'\E[01;44;33m'
export LESS_TERMCAP_ue=$'\E[0m'
export LESS_TERMCAP_us=$'\E[01;32m'
```

## .prompt

```
# para poner en PS1, 2 líneas, en color, la primera user@host pwd, la segunda el shell
\[\e]0;\w@a\]\n\[\e[32m\]\u@h \[\e[33m\]\w\[\e[0m\]\n\$\
```

## .vimrc

```
" Configuration file for vim
set
runtimepath=~/.vim,/etc/vim,/usr/share/vim/vimfiles,/usr/share/vim/addons,/usr/share/vim/vim63,/usr/share/vim/vimfiles,/usr/share/vim/addons/after,~/.vim/after

" Normally we use vim-extensions. If you want true vi-compatibility
" remove change the following statements
set nocompatible " Use Vim defaults instead of 100% vi compatibility
set backspace=indent,eol,start " more powerful backspacing
```

```

" Now we set some defaults for the editor
set autoindent      " always set autoindenting on
" set linebreak     " Don't wrap words by default
set textwidth=0     " Don't wrap lines by default
set backupcopy=yes  " Keep a backup file
set viminfo='20,\ "50  " read/write a .viminfo file, don't store more than
                        " 50 lines of registers
set history=50      " keep 50 lines of command line history
set ruler           " show the cursor position all the time

" Suffixes that get lower priority when doing tab completion for filenames.
" These are files we are not likely to want to edit or read.
set
suffixes=.bak,~, .swp, .o, .info, .aux, .log, .dvi, .bbl, .blg, .brf, .cb, .ind, .idx, .ilg, .inx
, .out, .toc

" We know xterm-debian is a color terminal
if &term =~ "xterm-debian" || &term =~ "xterm-xfree86"
    set t_Co=16
    set t_Sf=m
    set t_Sb=m
endif

" Make p in Visual mode replace the selected text with the "" register.
vnoremap p <Esc>:let current_reg = @"<CR>gvdi<C-R>=current_reg<CR><Esc>

" Vim5 and later versions support syntax highlighting. Uncommenting the next
" line enables syntax highlighting by default.
syntax on
set background=dark
" If using a dark background within the editing area and syntax highlighting
" turn on this option as well
" set background=dark

if has("autocmd")
    " Enabled file type detection
    " Use the default filetype settings. If you also want to load indent files
    " to automatically do language-dependent indenting add 'indent' as well.
    filetype plugin on

endif " has ("autocmd")

" Some Debian-specific things
augroup filetype
    au BufRead reportbug.*          set ft=mail
    au BufRead reportbug-*          set ft=mail
augroup END

" Set paper size from /etc/papersize if available (Debian-specific)
try
    if filereadable('/etc/papersize')
        let s:papersize = matchstr(system('/bin/cat /etc/papersize'), '\p*')
        if strlen(s:papersize)
            let &printoptions = "paper:" . s:papersize
        endif
    endif

```

```
    unlet! s:papersize
endif
catch /E145/
endtry

" The following are commented out as they cause vim to behave a lot
" different from regular vi. They are highly recommended though.
"set showcmd           " Show (partial) command in status line.
"set showmatch         " Show matching brackets.
"set ignorecase        " Do case insensitive matching
"set incsearch         " Incremental search
"set autowrite         " Automatically save before commands like :next and :make

" Source a global configuration file if available
if filereadable("/etc/vim/vimrc.local")
    source /etc/vim/vimrc.local
endif
```

From: <https://miguelangel.torresegea.es/wiki/> - miguel angel torres egea

Permanent link: <https://miguelangel.torresegea.es/wiki/linux:scripts:home?rev=1317200811>

Last update: 15/11/2011 01:59

