

Linux skills

I've used Linux for a long time but I'm still learning basic stuff like keyboard shortcuts and scripting. I thought I'd write down some things as a cheat-sheet of sorts, including tools I like or tools I've been planning on testing.

Terminal shortcuts

Alt-Backspace

Cut word to the left

Alt-d

Cut word to the right

Ctrl-u

Cut out everything between the cursor position and the beginning of the line.

Ctrl-k

Cut out everything between the cursor position and the end of the line.

Ctrl-y

Paste the most recently cut segment at the position of the cursor.

Cutting and pasting is local to each bash instance. You can't cut from one terminal using these commands and paste into another terminal.

Ctrl-d

Delete the character at the position of the cursor.

Ctrl-h

Delete the character to the left of the cursor.

Ctrl-a

Move cursor to the beginning of the line.

Ctrl-e

Move cursor to the end of the line.

Ctrl-left arrow

Move cursor to the beginning of the previous word.

Ctrl-right arrow

Move cursor past the end of the next word.

Shift-page up

Scroll up one page.

Shift-page down

Scroll down one page.

Bash numerical... things

Alt-1

Repeat next keyboard input once

Alt-2

Repeat two times

...
Alt-34
Repeat 34 times

Examples:

Alt-5 1
lllll

Alt-10 Ctrl-d
<delete 10 character to the right of the cursor>

Ubuntu Unity window management

For all positions the sequence is

Ctrl-Alt-Keypad X

where X is

- 0 : Maximize window.
- 1 : Move window to bottom left quadrant of screen.
- 2 : Bottom half.
- 3 : Bottom right quadrant.
- 4 : Left half.
- 5 : Entire screen (but not formally maximized).
- etc etc.

Repeat combination for different width and sizes of the window. Not sure what the pattern is yet but they seem to cycle through some settings.

Remote access

ssh privatevm
Log in with current username to host privatevm

ssh cjp@privatevm
Log in as user cjp to host privatevm

ssh-keygen
Generate public/private key pair.

cat .ssh/id_rsa.pub > authorized_keys
Put the file authorized_keys under the directory .ssh/ in the home directory of any user whom you would like to be able to log in using the key pair. Make sure the file is readable only to the file's owner.

This is particularly practical for things like scp, which allows auto-completion if no password is required or if it is handled by ssh-agent.

scp tmp/main.c cjp@privatevm:/home/cjp/Development/C
Copy local file tmp/main.c to /home/cjp/Development/C on host privatevm using account cjp.

Search and find

Grep

```
grep include *.c
```

Find files with the suffix .c and contain the word "include"

```
grep -R "^ssh" Development
```

Find files recursively under the directory "Development" that begin with the word "ssh".

Find

```
find tmp/ -name *.pdf
```

In directory tmp, find files with suffix pdf recursively.

```
find Z/To_be_sorted/2012-05-25/ -name "*.flv" -exec ls -lh {} \;
```

In Z/To_be_sorted/2012-05-25/, feed the paths to files ending with ".flv" to the command "ls -lh".

List

```
ls -lh --sort size Downloads/
```

List long file information with human-readable size-units in the folder Downloads, and sort by size, with big files at the top. Can also be sorted by extension and time. The order can be reversed with -r.

Scripting

```
dirname $1
```

gives the path to the file in argument \$1 without trailing slash.

```
basename $1
```

gives the filename of the file in argument \$1 without previous directories being listed.

Example:

for converting flv-files to avi-files using mpeg. Invoked through 'find Media/FLV -name "*.flv" -exec ./ff.sh {} Media/AVI \;'

```
#!/bin/bash
contdir=$(dirname "$1")
oldfilename=$(basename "$1")
newfilename=$(basename "$1" | sed 's/.flv/.avi/')
thisdir=$(dirname $2)/$contdir
if [ ! -d "$thisdir" ]
then
    echo mkdir -p $thisdir
    mkdir -p $thisdir
fi

echo ffmpeg -sameq -i "$1" "$thisdir/$newfilename"
ffmpeg -sameq -i "$1" "$thisdir/$newfilename"
```

So file `Media/FLV/Documentary/Chernobyl.flv` would end up being `Media/AVI/Documentary/Chernobyl.avi`. And this should work at arbitrary depth through the directory hierarchy.

```
#!/bin/bash

filetypes=(jpg png gif)
for type in ${filetypes[@]}
do
    for f in $(find $1 -name ".*.$type")
    do
        ls "$f"
    done
done
```

Find all files in the directory indicated by the command line argument given to the script that end with jpg, png or gif. Courtesy of Linux Format. Obviously you can do more interesting things that just "ls"...

Perl

```
cat ls_loop.sh | perl -p -e 's/bash/dash/e;'
```

Pipe the contents of file `ls_loop.sh` to `perl` which then runs a substitution where `bash` is replaced with `dash` everywhere in the text.

Byobu

Nifty little tool based on `tmux` but you can get `screen` to be the muscle if you want. I started using it so that I could mark up text printed to standard out after commands and copy it to the clipboard. Don't get me started about the fact that only two tools exist in the year of our Lord 2012 for accomplishing this task, or how I have to first mark and copy the text into the `tmux` buffer and then issue "C-x C-v" to copy it into the X clipboard". Don't!

Config file `.byobu/keybindings.tmux` is as follows:

```
set -s escape-time 0
unbind-key -n C-y
bind-key -n C-y paste-buffer

unbind-key -n M-c
bind-key -n M-c copy-mode

unbind-key C-v
bind-key C-v run 'tmux show-buffer | xclip -selection clipboard'

bind-key -t emacs-copy C-Left previous-word
bind-key -t emacs-copy C-Right next-word-end
set -g prefix ^x,F12
unbind-key -n C-a
set-option -g pane-active-border-bg black
set-option -g pane-active-border-fg black
bind-key x send-prefix # C-x x will send C-x through to application(like emacs!)
```

And the config file `.byobu/backend` is simple enough:

```
BYOBU_BACKEND=tmux
```

Finally `.byobu/profile.tmux` contains one default line and one custom:

```
source $BYOBU_PREFIX/share/byobu/profiles/tmux
set-window-option -g mode-keys emacs
```

This result in a main set of key bindings follow by the copy-mode ones:

```
bind-key          C-Space next-window
bind-key          C-a last-window
bind-key          C-c new-window
bind-key          C-d detach-client
bind-key          C-h previous-window
bind-key          Tab select-pane -t :.+
bind-key          C-l refresh-client
bind-key          C-n next-window
bind-key          C-o rotate-window
bind-key          C-p previous-window
bind-key          C-v run-shell "tmux show-buffer | xclip -selection c
bind-key          C-w list-windows
bind-key          C-x lock-server
bind-key -n       C-y paste-buffer
bind-key -n       C-z suspend-client
bind-key -n       Space next-window
bind-key -n       ! break-pane
bind-key -n       " choose-window
bind-key -n       # list-buffers
bind-key -n       $ command-prompt -I #S "rename-session '%%'"
bind-key -n       % split-window -h
bind-key -n       & confirm-before -p "kill-window #W? (y/n)" kill-w
bind-key -n       ' command-prompt -p index "select-window -t ':%%'"
bind-key -n       ( switch-client -p
bind-key -n       ) switch-client -n
bind-key -n       * list-clients
bind-key -n       , command-prompt -I #W "rename-window '%%'"
bind-key -n       - delete-buffer
bind-key -n       . command-prompt "move-window -t '%%'"
bind-key -n       0 select-window -t :0
bind-key -n       1 select-window -t :1
bind-key -n       2 select-window -t :2
bind-key -n       3 select-window -t :3
bind-key -n       4 select-window -t :4
bind-key -n       5 select-window -t :5
bind-key -n       6 select-window -t :6
bind-key -n       7 select-window -t :7
bind-key -n       8 select-window -t :8
bind-key -n       9 select-window -t :9
bind-key -n       : command-prompt
bind-key -n       ; last-pane
bind-key -n       = choose-buffer
bind-key -n       ? list-keys
bind-key -n       A command-prompt "rename-window %%"
```

```

bind-key -n          D choose-client
bind-key -n          K confirm-before kill-window
bind-key -n          L switch-client -l
bind-key -n          [ copy-mode
bind-key -n          \ confirm-before kill-server
bind-key -n          ] paste-buffer
bind-key -n          a send-prefix
bind-key -n          c new-window
bind-key -n          d detach-client
bind-key -n          f command-prompt "find-window '%%'"
bind-key -n          i display-message
bind-key -n          k confirm-before kill-window
bind-key -n          l refresh-client
bind-key -n          n next-window
bind-key -n          o select-pane -t :.+
bind-key -n          p previous-window
bind-key -n          q display-panes
bind-key -n          r refresh-client
bind-key -n          s choose-session
bind-key -n          t clock-mode
bind-key -n          w list-windows
bind-key -n          x lock-server
bind-key -n          { swap-pane -U
bind-key -n          | split-window
bind-key -n          } swap-pane -D
bind-key -n          ~ show-messages
bind-key -n          BSpace previous-window
bind-key -n          F1 new-window -k -n config byobu-config
bind-key -n          F2 new-window
bind-key -n          F3 previous-window
bind-key -n          F4 next-window
bind-key -n          F5 source-file /usr/share/byobu/profiles/tmuxrc
bind-key -n          F6 detach-client
bind-key -n          F7 copy-mode
bind-key -n          F8 command-prompt -p (rename-window) "rename-window
bind-key -n          F9 new-window -k -n config byobu-config
bind-key -n          PPage copy-mode -u
bind-key -n          BTab select-pane -t :.-
bind-key -r          Up select-pane -U
bind-key -r          Down select-pane -D
bind-key -r          Left select-pane -L
bind-key -r          Right select-pane -R
bind-key -r          M-1 select-layout even-horizontal
bind-key -r          M-2 select-layout even-vertical
bind-key -r          M-3 select-layout main-horizontal
bind-key -r          M-4 select-layout main-vertical
bind-key -r          M-5 select-layout tiled
bind-key -n          M-c copy-mode
bind-key -n          M-n next-window -a
bind-key -n          M-o rotate-window -D
bind-key -n          M-p previous-window -a
bind-key -n          M-F11 break-pane
bind-key -n          M-IC paste-buffer
bind-key -n          M-NPage copy-mode

```

```

bind-key -n M-PPage copy-mode
bind-key -r M-Up resize-pane -U 5
bind-key -n M-Up choose-session
bind-key -r M-Down resize-pane -D 5
bind-key -n M-Down choose-session
bind-key -r M-Left resize-pane -L 5
bind-key -r M-Right resize-pane -R 5
bind-key -n C-F2 display-panes
bind-key -n C-F3 display-panes
bind-key -n C-F4 display-panes
bind-key -n C-F5 send-keys ". /usr/bin/byobu-reconnect-sockets" ;
bind-key -n C-F6 kill-pane
bind-key -n C-F8 new-window -k "byobu-layout restore; clear; /bin
bind-key -n C-F11 join-pane -h -s :. -t :-1
bind-key -r C-Up resize-pane -U
bind-key -n C-Up resize-pane -U
bind-key -r C-Down resize-pane -D
bind-key -n C-Down resize-pane -D
bind-key -r C-Left resize-pane -L
bind-key -n C-Left resize-pane -L
bind-key -r C-Right resize-pane -R
bind-key -n C-Right resize-pane -R
bind-key -n S-F1 new-window -k -n help "$BYOBU_PAGER $BYOBU_PREFI
bind-key -n S-F2 display-panes
bind-key -n S-F3 display-panes
bind-key -n S-F4 display-panes
bind-key -n S-F5 new-window -k /usr/lib/byobu/include/cycle-statu
bind-key -n S-F6 run-shell "exec touch $BYOBU_RUN_DIR/no-logout"
bind-key -n S-F8 next-layout
bind-key -n S-F11 join-pane -v -s :. -t :-1
bind-key -n S-F12 source-file /usr/share/byobu/keybindings/f-keys.
bind-key -n S-Up display-panes
bind-key -n S-Down display-panes
bind-key -n S-Left display-panes
bind-key -n S-Right display-panes
bind-key -n C-S-F2 new-session
bind-key -n C-S-F3 swap-window -t :-1
bind-key -n C-S-F4 swap-window -t :+1
bind-key -n C-S-F5 new-window -d "byobu-select-profile -r"
bind-key -n C-S-F8 command-prompt -p "Save byobu layout as:" "run-s
bind-key -n C-S-F12 new-window /usr/lib/byobu/include/mondrian
bind-key -n C-S-Left previous-window
bind-key -n C-S-Right next-window

bind-key -t emacs-copy C-Space begin-selection
bind-key -t emacs-copy C-a start-of-line
bind-key -t emacs-copy C-b cursor-left
bind-key -t emacs-copy C-c cancel
bind-key -t emacs-copy C-e end-of-line
bind-key -t emacs-copy C-f cursor-right
bind-key -t emacs-copy C-g clear-selection
bind-key -t emacs-copy C-k copy-end-of-line
bind-key -t emacs-copy C-n cursor-down
bind-key -t emacs-copy C-p cursor-up

```

```

bind-key -t emacs-copy      C-r search-backward
bind-key -t emacs-copy      C-s search-forward
bind-key -t emacs-copy      C-v page-down
bind-key -t emacs-copy      C-w copy-selection
bind-key -t emacs-copy      Escape cancel
bind-key -t emacs-copy      Space page-down
bind-key -t emacs-copy      , jump-reverse
bind-key -t emacs-copy      ; jump-again
bind-key -t emacs-copy      F jump-backward
bind-key -t emacs-copy      N search-reverse
bind-key -t emacs-copy      R rectangle-toggle
bind-key -t emacs-copy      f jump-forward
bind-key -t emacs-copy      g goto-line
bind-key -t emacs-copy      n search-again
bind-key -t emacs-copy      q cancel
bind-key -t emacs-copy      NPage page-down
bind-key -t emacs-copy      PPage page-up
bind-key -t emacs-copy      Up cursor-up
bind-key -t emacs-copy      Down cursor-down
bind-key -t emacs-copy      Left cursor-left
bind-key -t emacs-copy      Right cursor-right
bind-key -t emacs-copy      M-1 start-number-prefix
bind-key -t emacs-copy      M-2 start-number-prefix
bind-key -t emacs-copy      M-3 start-number-prefix
bind-key -t emacs-copy      M-4 start-number-prefix
bind-key -t emacs-copy      M-5 start-number-prefix
bind-key -t emacs-copy      M-6 start-number-prefix
bind-key -t emacs-copy      M-7 start-number-prefix
bind-key -t emacs-copy      M-8 start-number-prefix
bind-key -t emacs-copy      M-9 start-number-prefix
bind-key -t emacs-copy      M-< history-top
bind-key -t emacs-copy      M-> history-bottom
bind-key -t emacs-copy      M-R top-line
bind-key -t emacs-copy      M-b previous-word
bind-key -t emacs-copy      M-f next-word-end
bind-key -t emacs-copy      M-m back-to-indentation
bind-key -t emacs-copy      M-r middle-line
bind-key -t emacs-copy      M-v page-up
bind-key -t emacs-copy      M-w copy-selection
bind-key -t emacs-copy      C-Up scroll-up
bind-key -t emacs-copy      C-Down scroll-down
bind-key -t emacs-copy      C-Left previous-word
bind-key -t emacs-copy      C-Right next-word-end

```

To launch a new byobu-session directly in Unity go to CompizConfig Settings Manager and enable the General->Commands plugin. Set a Command line of choice to "gnome-terminal -x /home/cjp/byobu_launcher.sh" where the script in question is:

```

#!/bin/bash
byobu -S "`date +%F'|' %k:%M`"

```

That creates a new session with a name like "2012-06-03|21:46".

Then set the corresponding keybinding to something good. I use `<Control><Alt>b`. `<Control><Alt>t` starts a plain `gnome-terminal`, so `<Control><Alt>b` starting a `gnome-terminal` with a new `byobu`-session makes sense mnemonically. `<Control><Alt>a` might be good for when you want to start `byobu` and attach to an old session.

Surfraw

I don't like using the `surfraw` program for one reason and one reason only: Julian Assange. Much like with the meager choice of tools with which I can mark text in a terminal with keyboard shortcuts, it is best not to get me started on Julian. Note that I almost invariably refer to people as `mr Gates` or `ms Hooker` out of courtesy, even when the latter describes her profession and not her name. Not so for Julian.

I can take some solace in the fact that he hasn't worked on the project for some time, having been busy - I presume - with undermining US efforts to conduct international relations without the use of bombs and helping Afghan extremists find people who oppose them. Busy, busy!

So... `Surfraw`. I thought it would print out Google search results if I typed in a query at the command line. Not quite it seems. By default `Surfraw` turns your query into a properly formatted URL which it then feeds to a reasonably chosen web browser. In my case `Google Chrome`. I don't quite see that point of that. `Google Chrome` is great and use it to the exclusion of almost any other browser, but if I'm driven to perform a Google search from the command line I'm probably not using `Google Chrome` for a reason. The closest thing to my imagined functionality is achieved by using this config file placed in `.config/surfraw/conf` :

```
SURFRAW_text_browser=/usr/bin/w3m
SURFRAW_graphical=no
SURFRAW_text_browser_args=-dump
```

This is slightly impractical as `w3m` stays true to the HTML formatting and doesn't aspire to the kind of succinct summary I have in mind. So I comment out the last line and use `w3m` straight up, which is darned clever. `w3m` has a very subtle approach and acts kind of like a pager such as `"less"` and `"more"`, only with color and the ability to follow links. It seems to adhere to many of Emacs key bindings and by pressing `Tab` you proceed through links. For some reason `C-u` is the binding for moving backwards through the links.

`Surfraw` has support for many more services than `Google`:

```
$ sr -elvi
GLOBAL ELVI:
acronym      -- Look for acronyms definitions (www.acronymfinder.com)
ads          -- Search SAO/NASA Astrophysics Data System
alioth       -- Search Alioth (alioth.debian.org)
amazon       -- Search the amazon.com bookstore
archpkg      -- Search Arch Linux Packages (www.archlinux.org/packages/)
archwiki     -- Search the Arch Linux Wiki
arxiv        -- Search arXiv E-Print Archive for articles
ask          -- Question the web using Ask Jeeves (www.ask.com)
aur          -- Search aur.archlinux.org for PKGBUILDs
austlii      -- Search Australian Law docs (www.austlii.edu.au)
bbcnews      -- Search BBC News (news.bbc.co.uk)
bing         -- Search the web using Microsoft's Bing (www.bing.com)
bookfinder   -- Search for books using www.bookfinder.com
```

bugmenot -- Bypass compulsory web registration with bugmenot.com
bugzilla -- Search for bugs on Bugzilla bugtrackers
cablesearch -- search openports for OpenBSD packages
cia -- Search CIA documents at www.cia.gov
cite -- Search computer science papers (citeseerx.ist.psu.edu)
cliki -- Search the common lisp wiki
cnn -- Search on CNN (cnn.com)
codesearch -- Search source code using Google Code Search
(www.google.fr/codesearch)
comlaw -- Search Australian Law using Comlaw(www.comlaw.gov.au)
ctan -- Search the Comprehensive TeX Archive Network (ctan.org)
currency -- Convert currencies with the Universal Currency Converter
(www.xe.net/ucc)
cve -- Search for CAN assignments in CVE
debugs -- Search the debian BTS (bugs.debian.org)
debcontents -- Search contents of debian/ubuntu packages
(packages.debian.org/packages.ubuntu.com)
deblists -- Search debian mailing lists
(lists.debian.org/search.html)
deblogs -- Show changelogs for a package in Debian main
(changelogs.debian.net)
debpackages -- Search debian/ubuntu packages
(packages.debian.org/packages.ubuntu.com)
debpkghome -- Visit the home page for a Debian package
debpts -- Search the Debian Package Tracking System
(packages.qa.debian.org)
debsec -- Search the Debian Security Tracker for CVE ids or
package names
debvcsbrowse -- Browse the VCS repository for a Debian package
debwiki -- Search the Debian Wikis (wiki.debian.org &
women.debian.org/wiki)
deja -- Search usenet using Google Groups (groups.google.com)
deli -- Search Delicious bookmarks
discogs -- Search the Discogs database of music information
(www.discogs.com)
dmoz -- Search the Open Directory Project web directory
(dmoz.org)
duckduckgo -- Securely search the web using duckduckgo
(www.duckduckgo.com)
ebay -- Search the Ebay auction site
etym -- Look up word origins at www.etymonline.com
excite -- Search on Excite (www.excite.com)
finkpkg -- Search Fink packages (pdb.finkproject.org)
foldoc -- The Free On-Line Dictionary Of Computing (foldoc.org)
freebsd -- Search FreeBSD related information (www.freebsd.org)
freedb -- Search for cd track listings in FreeDB (www.freedb.org)
freshmeat -- Search Freshmeat (www.freshmeat.net)
fsfdir -- Search the FSF/UNESCO Free Software Directory
(directory.fsf.org)
gcache -- Search the web using Google cache (www.google.com)
genbugs -- Search the Gentoo bug tracker (bugs.gentoo.org)
genportage -- Search gentoo-portage.com for packages
google -- Search the web using Google (www.google.com)
gutenberg -- Search for books on Project Gutenberg (gutenberg.org)

```

happypenguin    -- Search the Linux Game Tome (www.happypenguin.org)
imdb            -- Search the Internet Movie Database (www.imdb.com)
ixquick        -- Search the web using ixquick [HTTPS] (www.ixquick.com)
jamendo        -- Search Jamendo: free music with Creative Commons
licenses (www.jamendo.com)
javasun        -- Search Java API docs (java.sun.com)
lisp           -- Search lisp documentation
lastfm         -- Search last.fm
leodict        -- Search Leo's German <-> English dictionary
(dict.leo.org)
lsm            -- Search the Linux Software Map
macports       -- Search macports packages (macports.org)
mathworld     -- Search Wolfram MathWorld
mininova      -- Search the mininova bittorent source.
musicbrainz   -- Search MusicBrainz (musicbrainz.org)
netbsd        -- Search NetBSD related information (www.netbsd.org)
ntrs          -- Search the NASA Technical Report Server
openbsd       -- Search OpenBSD related information (www.openbsd.org)
openports     -- search openports for OpenBSD packages
opensearch    -- Search an OpenSearch-enabled website
pasearch      -- Search the unofficial Penny Arcade archives
(pipefour.org/pa)
pgpkeys       -- Search the PGP key database
piratebay     -- Search thepiratebay.org for torrents
pubmed        -- Search medical/molbio databases (www.ncbi.nlm.nih.gov)
rae           -- Spanish Dictionary
rfc           -- Search RFCs (internet standards documents)
rhyme         -- Search for rhymes et al using Lycos Rhyme
(rhyme.lycos.com)
rpmsearch     -- Search for RPMs in various distros
scholar       -- Search Google Scholar (scholar.google.com)
scicom       -- Search Scientific Commons
scirus        -- Search for science using Scirus (scirus.com)
scitopia     -- Search for science with scitopia.org
scpan        -- Search the Comprehensive Perl Archive Network
(search.cpan.org)
scroogle      -- Search Google anonymously via Scroogle
(www.scroogle.org)
slashdot     -- Search stories on Slashdot (www.slashdot.org)
slinuxdoc    -- Search entries in LDP (www.linuxdoc.org)
sourceforge  -- Search SourceForge (www.sourceforge.net)
springer     -- Search Springer for Books and Articles
stack        -- Search Stack Overflow
stockquote   -- Get a single stock quote (multiple providers)
sunonesearch -- Search Sun One Search (onesearch.sun.com)
thesaurus    -- Look up word in Merriam-Webster's Thesaurus (www.m-
w.com)
translate    -- Translate human languages (various providers)
urban        -- Search urbandictionary.com for a definition
W           -- Activate Surfraw defined web-browser
w3css        -- Validate a CSS URL with the w3c CSS validator
(jigsaw.w3.org/css-validator)
w3html       -- Validate a web page URL with the w3c validator
(validator.w3.org)

```

```
w3link          -- Check web page links with the w3c linkchecker
                 (validator.w3.org/checklink)
w3rdf           -- Validate a RDF URL with the w3c RDF validator
                 (validator.w3.org)
wayback        -- Search The Internet Archive's Wayback Machine for a URL
                 (archive.org)
webster        -- Look up word in Merriam-Webster's Dictionary (www.m-
w.com)
wetandwild     -- Real time weather information (many sources)
wikipedia      -- Search the free encyclopedia wikipedia
woffle        -- Search the web using Woffle (localhost:8080)
worldwidescience  -- Search for science with www.worldwidescience.org
yahoo         -- Search Yahoo categories (www.yahoo.com)
yandex        -- Search the web using Yandex (yandex.ru)
youtube       -- Search YouTube (www.youtube.com)
yubnub        -- Use the social command-line for the web (yubnub.org)
```

Useful text editors

nano

Basic text editor with nice explanation of commands at the bottom of the screen.

emacs

Mother of all text editors, the keyboard shortcuts of which serves as a template for various shells. Used to write the raw text for this document.

emacs -nw

Invocation of emacs for terminal use.

gedit

This is a text editor much reviled by hard core Linux people, presumably because it's easy to use. It can highlight text based on the syntax of dozens of programming languages and has a big set of plugins available.

Trivia

I should really have called this **nix skills* because it will refer to things that are applicable to most Unix-style platforms. Linux, Unix, BSD, Solaris and more. But I don't dare name folders or files with a star in them because in a Linux-shell a star has special properties. If I type "ls *.pdf" it's equivalent to performing the "ls"(list) operation of all files ending with the suffix pdf. So if the directory contains

```
Alpha.pdf
Beta.pdf
Gamma.txt
```

then the command "ls *.pdf" will be expanded to "ls Alpha.pdf Beta.pdf Gamma.pdf". Apparently "mkdir *nix" works just fine and if you type "ls *n" and press the tab-key for autocompletion, the shell escapes the * with a backslash. However, typing "mkdir *.plt" didn't work because it matched a file in the directory and you obviously can't create a directory with the same name as a file.