



- 0. term - il terminale, strumento utile per fare ogni cosa
- 1. SCRIPT - e' un file che contiene i nostri comandi
- 2. vim - L'editor che ci permette di scrivere il nostro codice
- 3. AWK - Un parser semplice e potente che converte del testo "A" in testo "B"
- 4. SED - stream editor
- 5. FLUXBOX - Un Window Manager come GNOME, ma mooolto più leggero
- 6. MPD - Un sistema che ci permette di gesire la musica a basso livello
- 7. ncmpcpp - Un programma per ascoltare la musica
- 8. others - Altri strumenti come tilda, xscreensaver, conky

```
#####  
##### SCRIPT #####  
#####
```

Gli script sono sequenze di comandi messe in un file.  
Il file a sua volta viene reso eseguibile e chiamato ad ogni necessita'.

Vediamo ad esempio uno script:

```
> file: example/1/halt
```

lo rendo eseguibile con il comando:

```
chmod +x <nome file>
```

e lo eseguo:

```
./halt
```

Examples:

> Nella cartella example ci sono una serie di script utili da vedere e commentare per fare esperimenti e prove.

More info:

<http://www.pluto.it/files/ildp/guide/abs/index.html> [ Advanced Guide]

<http://www.shell-fu.org/> [ Bash tips&tricks ]

```
#####  
##### VIM #####  
#####
```

E' sempre utile conoscere un editor da terminale, perche' puo' sempre capitare di trovarsi senza grafica e dover fare delle operazioni su file.

VI e' installato di default anche nei sistemi minimali.  
VIM e' il suo successore (la M sta per VI-iMproved ) e' presente su  
Ubuntu/Debian/Slakware/Redhat/ecc...

Per utilizzarlo basta dare il comando :

```
$ vim nome_file
```

e per uscire basta digitare

```
:wq ( write & quit) oppure :q! (quit without saving)
```

trucchi:

[copia e incolla]

yy : yank copia tutta la riga

dd: taglia riga

100p <invio>: copia 100 volte la stessa riga

p: paste incolla

[aiuta a leggere]

:syntax on - modifica il testo colorandolo, permettendo a chi  
programma di sapere cosa scrive

:number on - fa vedere i numeri di riga

:autoindent - capisce quale codice stiamo scrivendo e ci aiuta ad  
indentare il codice

[modifica caratteri]

gU\$ : rendere maiuscola tutta la riga

gu\$ : rendere minuscola tutta la riga

:%s/pippo/pluto/ - sostituisce in tutto il file le parole "pippo",  
con la parola "pluto"

[autocompletamento]

Control+N : autocompleta la parola che stiamo scrivendo cercando  
parole simili nel nostro file

more info:

La cosa migliore per imparare ad usare vim è usare il suo tutorial:  
> vimtutor

Poi esiste un ottimo wiki:

<http://vim.wikia.com>

```
#####  
##### AWK #####  
#####
```

E' uno strumento fondamentale per il parsing di file

Parsing significa che prendi un file, lo filtri e ti cambio l'output

mandandoti solo la parte interessante.

AWK usa delle macro (delle funzioni) del linguaggio c, infatti è molto utile da usare per chi conosce un po' di programmazione c. Pero' e' molto più intuitivo del C!

AWK e' presente su tutti in sistemi operativi unix based con tanto di manuale e nelle distribuzioni piu' nuove anche esempi.

```
$ man awk
$ ls /usr/share/doc/mawk/
```

Spesso indispensabile sapere qualche comando quindi impariamo qualche trucco per evitare di fare operazioni ripetitive sui file.

Examples:

```
1. stampo la prima colonna
awk '{ print $1 }'
2. stampo la seconda colonna
awk '{ print $2 }'
3. stampo i caratteri finali di ogni riga
awk '{ print $NF}'
```

```
##### SED #####
```

SED e' un editor non interattivo.  
SED non apre il file completamente per modificarlo, ma esegue le operazioni nelle linee specifiche.

E' come pensare a VIM pero' senza vedere cosa state andando a modificare.

Perchè bisognerebbe pensare ad una follia del genere ?  
Perchè spesso gli script hanno bisogno anche di modificare i files e quindi è meglio lanciare un comando :)

Se ad esempio abbiamo un file di dimenstione 4GB, non possiamo aprirlo immediatamente con VIM ma nemmeno con un qualsiasi editor perchè ci metterebbe troppo tempo per caricare in memoria tutto il contenuto del file. SED serve per eseguire delle operazione sulle righe specifiche di un file.

Generalmente si usa SED per sostituzioni/inserimenti/cancellazioni la sua codifica e' molto intuitiva e per aiutare nella ricerca di pattern (schemi di ricerca) si usano delle espressioni regolari.

Examples:



caratteri importanti:

^ : inizio riga  
\$ : fine riga  
. : qualsiasi carattere  
+ : carattere ripetuto (quale carattere? quello specificato precedentemente)  
\* : qualsiasi carattere ripetuto (quale carattere, quello specificato precedentemente)  
[abc] : un insieme di caratteri (in questo caso: a,b oppure c)  
\t : tabulatore  
\n : a capo

```
#####  
#####  
##### FLUXBOX #####  
#####
```

Fluxbox e' un Window Manager, come GNOME, KDE, XFCE o altro  
La differenza e' che fluxbox e' leggero e permette di fare tutto quello che si vuole

Appena installato fluxbox crea una cartella .fluxbox che va permette di essere modificata.

i file più importanti sono

.fluxbox/keys - modifico le scorciatoie da tastiera  
.fluxbox/init - modifico come il mio sistema deve iniziare e dove mettere le cose  
.fluxbox/startup - specifico quali applicazioni far partire quando lancio fluxbox  
.fluxbox/style - modifico l'aspetto e scelgo i colori delle finestre

more info:  
[fluxbox-wiki.org](http://fluxbox-wiki.org)

```
#  
#  
# samba slide ended  
#  
#
```