

## Le système d'exploitation GNU/Linux

Nicolas Burrus <nicolas.burrus@laposte.net>



- Edition de texte
- Scripts shell

26 Avril 2005

## Editer du texte brut

### Pourquoi éditer du texte brut ?

- Lisible universellement
- Fichiers de configuration
- Code source
- Documentation
- Mails
- ...

### Editeurs populaires

- Vi / Vim / Gvim
- Emacs / Xemacs
- Gedit
- Jed
- ...

## Contenu du cours

- 1 Edition de texte
- 2 Scripts shell
- 3 TP

## Utilisation de Vim

### Contraintes initiales :

- Disponible partout
- Léger
- Adaptable à tout type de clavier (sans touches de fonction, etc.)
- Mode texte

### Résultat :

- 2 modes d'utilisation
  - Un mode de commande
  - Un mode d'insertion ou d'édition proprement dit
- Beaucoup de raccourcis clavier
- Peu intuitif au départ

## Survie sous Vim

```
vim [file1] [file2] ...
```

Edite les fichiers `fileN`

### Changement de mode :

- [Esc] : revenir au mode de commande
- [i] : entrer dans le mode d'édition

### En mode commande :

- x : supprime un caractère
- dd : supprime une ligne
- :q! : quitte sans sauvegarder
- :wq : quitte en sauvegardant

## Utilisation d'Emacs

### Utilisation du minibuffer

- M-x : active le minibuffer
- Possibilité d'appeler les fonctions d'Emacs
- Ex: M-x `save-buffer` : sauvegarde le fichier courant

### Raccourcis claviers usuels :

- `http://refcards.com/refcards/gnu-emacs/`
- Sélectionner une zone : C-space puis déplacement
- Copier : M-w
- Couper : C-w
- Coller : M-y

## L'éditeur Emacs

### Introduction :

- Part du projet GNU, écrit par Stallman
- Très extensible (en Lisp) ...
- ... et donc un peu "usine à gaz"
- Utilisation intensive des raccourcis claviers

### Syntaxe des raccourcis clavier :

- C-g : control + g
- M-g : Alt + g
- C-g s : control + g puis s

## Script shell ?

### Définition :

- Fichier contenant des commandes à destination du shell
- Le shell lit le fichier, et exécute toutes les commandes

### Intérêt :

- Automatisation de tâches de maintenance du système
- Répétition de tâches
- Regroupement de tâches compliquées sous une seule commande

### Généralités :

- Usuellement : `extension .sh`
- En-tête : `#!/bin/sh`
- Le fichier doit avoir les droits d'exécution
- Les commandes peuvent ensuite être rentrées

## Exemple de script

```
#!/bin/sh
# This is a comment.
echo "Starting backup ..."

cd
tar cfz etc.tar.gz /etc
tar cfz var.tar.gz /var
tar cfz home.tar.gz $HOME

mkdir -p archives/
mv *.tar.gz archives/

echo "Ok, backup successfully completed."
```

<http://nburrus.objectis.net/teaching/linux/>

## Astuces

### En vrac :

- \$1, \$2, ... : arguments du scripts
- \$# : nombre d'arguments
- ; peut être remplacé par un retour à la ligne

```
if cond; then commands1; else commands2; fi
Exécute commands1 si la commande cond réussit, sinon exécute
commands2
```

### **test** cond

```
Réussit si cond est vraie
Ex: test $1 = "toto"
Ex: test -f /bin/lis
```