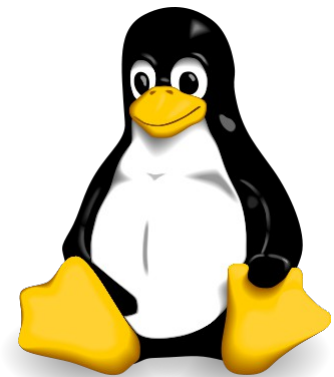




# Initiation Linux





# 1 - Introduction au système d'exploitation UNIX/Linux

---

## 1.1 - Caractéristiques d'UNIX

## 1.2 - Le Système Gnu/Linux

## 1.3 - Distribution Linux





## 1.1 - Caractéristiques d'UNIX

**Unix** est un système d'exploitation ayant les caractéristiques suivantes :

- **Multi-utilisateurs et Multitâches** : plusieurs utilisateurs peuvent accéder au système et exécuter un ou plusieurs programmes.
- **Temps partagé** : les ressources du processeur et du système sont réparties entre les utilisateurs
- **Système de fichiers hiérarchiques** : plusieurs systèmes de fichiers peuvent être rattachés au système de fichiers principal
- **Entrées-Sorties intégrées au système de fichiers** : les périphériques sont représentés par des fichiers, ce qui rend le système indépendant du matériel.
- **Gestion de la mémoire virtuelle** : un mécanisme d'échange entre la RAM et le disque permet de pallier un manque de RAM (Random Access Memory).
- **Processus réentrants** : les processus exécutant le même programme utilisent une seule copie de celui-ci en RAM.
- **Interface utilisateur interactive (shell)** : elle est constituée d'une interface séparée du noyau permettant à l'utilisateur de choisir son environnement de travail. Elle intègre un langage de commande très sophistiqué (scripts).

## 1.2 – Le Système Gnu/Linux

### « GNU's Not Unix »

L'idée d'un système d'exploitation libre est né en 1984 avec la Free Software Foundation

Linux, système Unix libre sur PC, était au départ un projet de loisir de Linus Torvalds (1991)

En 1992, Linux (version 0.96) est pleinement fonctionnel. C'est la naissance des premières distributions

Notoriété et consécration en 1998 pour Linux et les logiciels libres

Association avec le projet GNU : le système GNU/Linux (compilateur gcc)

Code source disponible (licence GPL)

Compatible POSIX (standard logiciel)

Compatible avec les autres UNIX

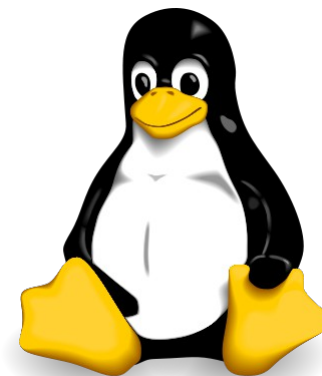
Possibilité de cohabitation avec d'autres systèmes

Implémentation complète de la pile TCP/IP

Services réseaux

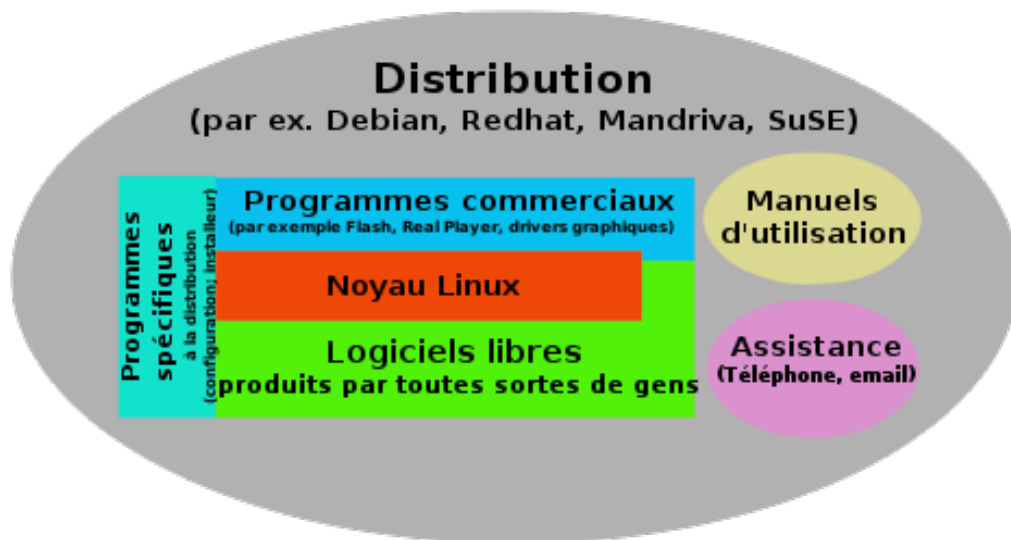
Interface graphique X Window

<http://www.linux.org>





## 1.3a - Distribution Linux

Une distribution Linux, appelée aussi distribution GNU/Linux pour faire référence aux logiciels du projet GNU, est un ensemble cohérent de logiciels, la plupart étant logiciels libres, assemblés autour du noyau Linux.








[http://fr.wikipedia.org/wiki/Distribution\\_Linu](http://fr.wikipedia.org/wiki/Distribution_Linu)

## 1.3b - Distribution Linux

	<p><u>Fedora</u> est une distribution système de gestion de paquets</p>
	<p><u>Mandriva</u> est la plus internationale d'origine grand public, elle est basée sur Red Hat</p>
	<p>est une distribution système de gestion de paquets</p>
	<p>est une distribution système de gestion de paquets</p>

## 1.3c - Distribution Linux

	<p><u>ArchLinux</u> est une <u>permanente</u>. Elle c à une communaut inspirée par <u>Crux l</u> d'outils spécifiques distribution adapté</p>
	<p><u>Debian</u> est une dis</p>
	
	
	



## 2 – Premier pas sur linux

---

2.1 - Installation linux

2.2 – Le Système Gnu/Linux : Architecture

2.3 – Premières utilisations : session connexion





Yantra Technologies

## 2.1 - Installation linux

---



**Mandriva**

Installation Mandriva 2011 :

[http://wiki.mandriva.com/en/2011.0\\_Tour](http://wiki.mandriva.com/en/2011.0_Tour)



Installation Ubuntu :

<http://doc.ubuntu-fr.org/installation>



Installation RedHat :

[https://access.redhat.com/knowledge/docs/Red\\_Hat\\_Enterprise\\_Linux/?locale=fr-FR](https://access.redhat.com/knowledge/docs/Red_Hat_Enterprise_Linux/?locale=fr-FR)

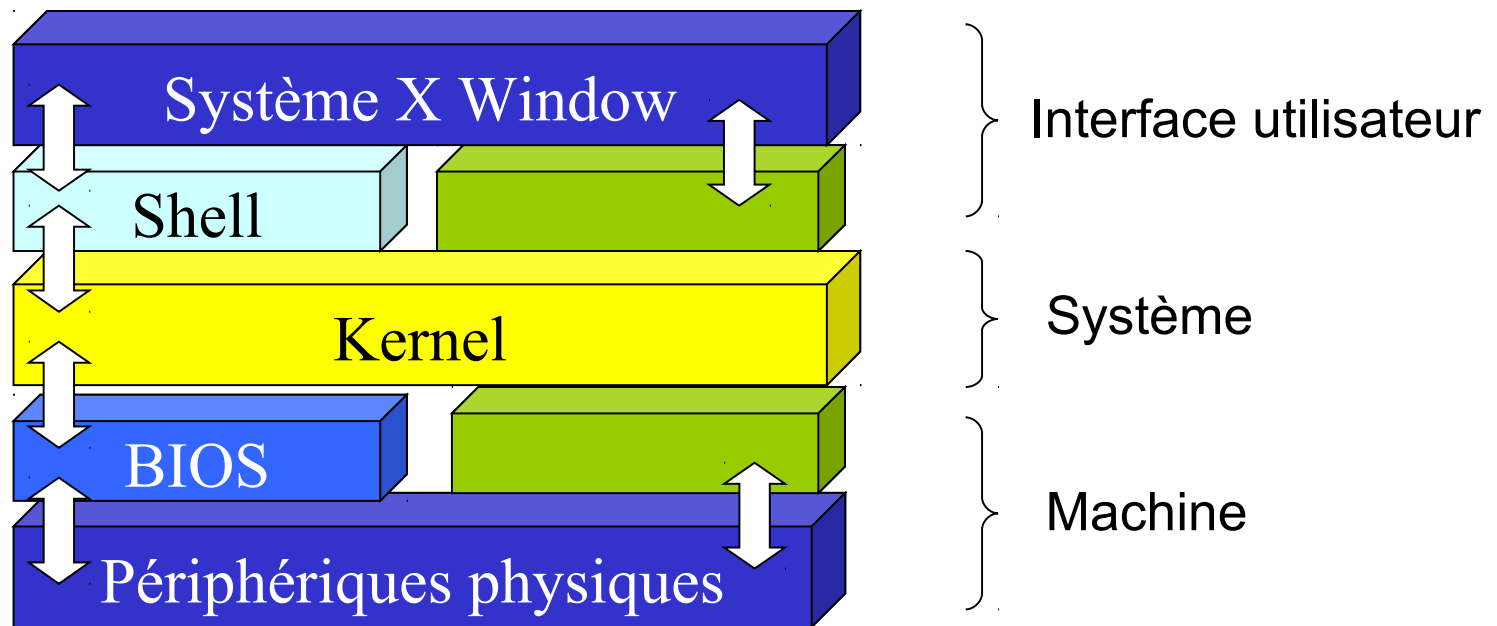


Installation Debian :

<http://www.debian.org/releases/stable/installmanual>



## 2.2 – Le Système Gnu/Linux : Architecture





## 2.3 – Premières utilisations : session connexion

---

Tout utilisateur est identifié par un nom (login name) et un mot de passe (passwd).

Les comptes utilisateurs sont créés dans le fichier `/etc/passwd`

Une fois ces informations entrées l'utilisateur entre en session, il est connecté.

L'interpréteur de commande (le shell par défaut) est lancé dès la saisie de l'utilisateur.

Des scripts globaux à tous les utilisateurs d'initialisation sont lancés, ainsi que les scripts liés au compte pour personnaliser la session

Affichage du prompt, et attente de commandes

Pour sortir de session(deconnexion), utiliser la commande **exit** ou **logout**.





## 3 – Les commandes

---

- 3.1 – Commande en ligne
- 3.2 – Les scripts de connexion
- 3.3 – les commandes (1)
- 3.4 – Le Manuel : la commande man
- 3.5 – Commande d'information
- 3.6 - Autre information

### Le shell : interpréteur de commande

- Lit et exécute les commandes de l'utilisateur
- Propose un contrôle de processus
- Gère les redirections en entrée et en sortie
- Propose un véritable langage de programmation
- Plusieurs types de shell disponibles :
  - Le plus utilisé sous Linux : le bash
  - csh, ksh ...

## 3.2 – Les scripts de connexion

---

Le script **etc/profile** et tous les scripts `/etc/profile.d/*.sh`, puis il y a exécution de **\$HOME/.bash\_profile** (la variable **\$HOME** contient le chemin vers le répertoire personnel). Il s'agit ainsi d'un fichier de démarrage personnel et paramétrable.

A son tour il exécute **\$HOME/.bashrc** dans lequel il est recommandé de placer toutes les fonctions ou alias personnels.

Enfin le précédent exécute **/etc/bashrc**, dans lequel on place les alias globaux et la définition du prompt **\$PS1**.

Puis le prompt utilisateur s'affiche et le **shell** attend une commande  
...



## 3.3 – les commandes (1)

Une commande est composée d'un code mnémorique (son nom) suivi d'options et/ou de paramètres.

Une option commence par un tiret -

-c -mot

Pour obtenir la totalité des options --help

Ex : ls -l

Historique des commandes : les flèches haut et bas permettent de parcourir les dernières commandes utilisées.

Les alias : permettent de créer des noms de commandes

- Création : alias ll= "ls -l"
- Liste des alias : alias

Le **~** **seul** renvoie au **répertoire personnel** de l'utilisateur actuel.

Lorsqu'on tape une commande en ligne avec la touche **TAB** l'interpréteur cherche à compléter le nom du fichier.



## 3.4 – Le Manuel : la commande man

**man** [section] commande      *man paswwd*  
**man** -k mot\_clé                    *man -k passwd*

permet de rechercher une aide sur une commande ou un mot-clé.  
Il utilise la variable MANPATH pour effectuer la recherche des pages.  
Les pages man sont organisées en 8 sections standard comme suit :

- 1 = Commandes utilisateur
- 1M = Commandes administrateur
- 2 = Appels systèmes C.
- 3 = Fonctions C.
- 4 = Format des fichiers système.
- 5 = Divers.
- 6 = Jeux.
- 7 = Fichiers spéciaux.
- 8 = Procédures de maintenance système.



## 3.5 – Commande d'information

---

**whatis** : Cette commande permet de connaître la description d'une commande sans faire un man.

**makewhatis** : parcourt tous les pages de manuelles pour mettre à jour une base whatis. Ensuite on peut consulter cette base avec la commande whatis.

**apropos** : Rechercher une chaîne de caractère dans la base de données whatis.



## 3.5 – Commande d'information

---

**which** : localise une commande

**whereis** : Rechercher les fichiers exécutables, les sources et les pages de manuel d'une commande.

**locate** : commande permettant de localiser un fichier.



## 3.6 - Autre information

---

### ***Commandes Unix***

- [http://www.infres.enst.fr/~danzart/fiches/unix\\_abrege.html](http://www.infres.enst.fr/~danzart/fiches/unix_abrege.html)
- <http://jp.barralis.com/linux-man/>

### ***Linux et le Projet GNU***

- <http://www.gnu.org/gnu/linux-and-gnu.fr.html>

### ***Autres liens***

- <http://linux.developpez.com/>



## 4 - Arborescence du système de fichier

---

4.1 - Système de fichiers

4.2 – Arborescence des fichiers

4.3 – Désignation des fichiers

4.4 – Manipulation des répertoires

4.5 - Exemple : création de répertoires

4.6 - Manipulation des fichiers : ls

4.7 - Manipulation des fichiers

4.8 – Exemple : manipulation des fichiers

4.9 - Autres Commandes

## 4.1 - Système de fichiers

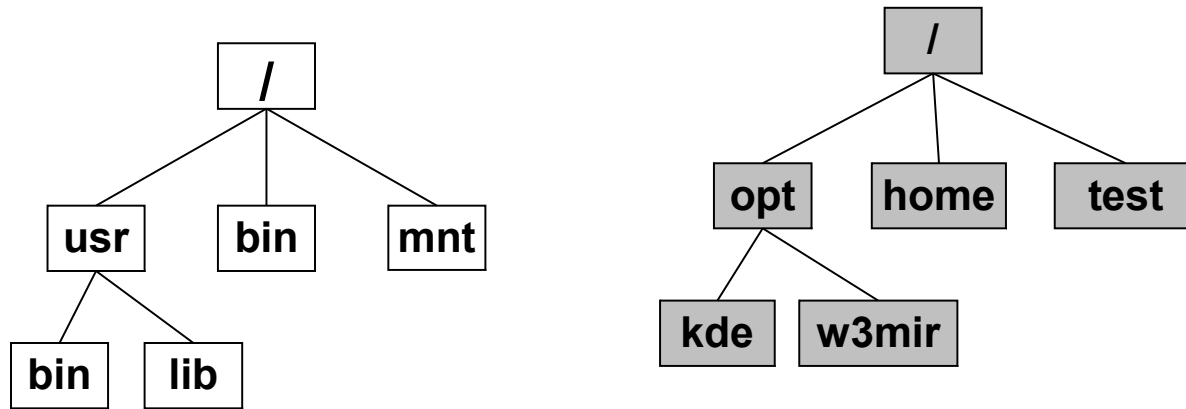
---

Le système de fichiers est organisé en une structure arborescente, dont les nœuds sont des **répertoires** et les feuilles des fichiers ordinaires.

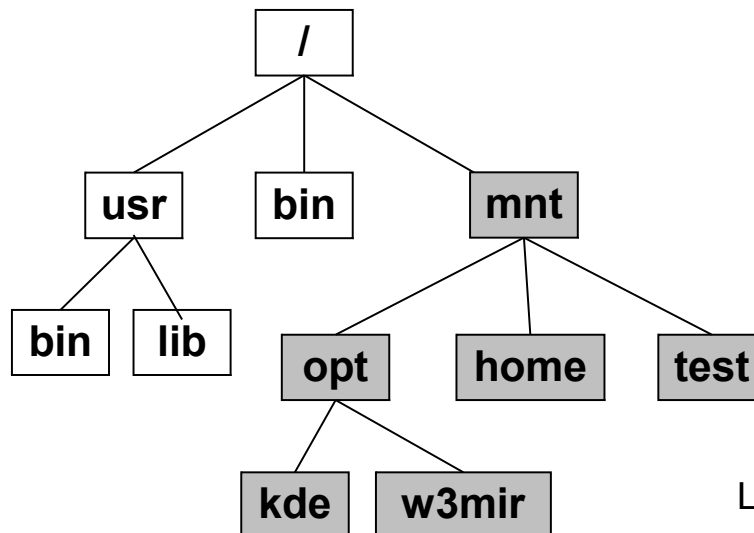
- Les fichiers ordinaires : servent à mémoriser les programmes et les données
- Les fichiers répertoires ou répertoires (**directories**) : chaque répertoire contient la liste et la référence des fichiers placés sous son contrôle et la référence du répertoire dont il dépend (répertoire père).
- Les fichiers spéciaux : les périphériques, les tubes ou autres supports de communication interprocessus.

Les noms de fichiers et les périphériques obéissent à la même syntaxe et l'accès aux périphériques est contrôlé par le mécanisme de protection des fichiers.

## 4.2 – Arborescence des fichiers



Une fois monté, le système de fichier du CD-ROM fait partie intégrante de l'arborescence



Les clés USB sont maintenant sur [/media](#)



## 4.3 – Désignation des fichiers

---

**pathname** : Le chemin d'accès du fichier est repéré par son nom et sa position dans l'arborescence.

répertoire racine: /

accès absolu : /home/utilisateur1/dev/prog1

accès relatif à partir du répertoire courant

répertoire courant : /home/utilisateur1

accès relatif dev/prog1

Dès l'entrée en session, le système se place dans un répertoire de travail initial : home directory

Les fichiers dont le nom commence par un point comme .bashrc sont des fichiers cachés.

## 4.4 – Manipulation des répertoires

**pwd** : donne le répertoire courant

```
utilisateur1 > pwd  
/home/utilisateur1
```

**cd** : permet de changer de répertoire de travail

```
utilisateur1 > cd ..  
/home  
utilisateur1 > cd  
/home/utilisateur1
```

**mkdir** : créer un nouveau répertoire

```
utilisateur1 > mkdir src  
utilisateur1 > cd src  
/home/utilisateur1/src
```

**rmdir** : supprime un répertoire s'il est vide

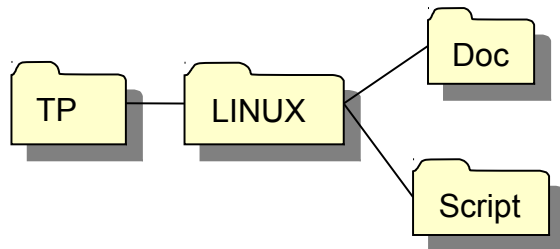
```
utilisateur1 > rmdir src
```

**find** : recherche d'un fichier à partir du répertoire donné

Pour rechercher le fichier .bash-profile chez l'utilisateur : `find . -name .bash_profile -print`

Pour rechercher tous les fichiers core et les supprimer : `find . -name core -exec rm -i {} \;`

## 4.5 - Exemple : création de répertoires



```
$ mkdir TP
$ cd TP
$ mkdir LINUX
$ cd LINUX
$ mkdir Script
$ mkdir Doc
$ ls
Doc/ Script/
```

## 4.6 - Manipulation des fichiers : ls

**ls** : donne la liste des fichiers et répertoires

```
utilisateur1 > pwd
/home/utilisateur1
utilisateur1> ls
bin dev essai tmp src
```

**ls -a** : donne tous les fichiers

```
... .bash_history .bashrc bin dev ...
```

**ls -l** : liste avec les détails

```
drwxr-xr-x 2 utilisateur1 512 jan 18 10:21 bin
drwxr-xr-x 2 utilisateur1 512 jan 15 16:05 dev
-rwxr—r— 1 utilisateur1 15 jan 20 14:40 essai
```

## 4.7 - Manipulation des fichiers

**cat** : permet d'afficher, de créer, de copier et de concaténer des fichiers

- Affichage : **cat /etc/passwd**
- Création : **cat > essai**  
**Ceci est un essai**  
**<ctrl-d>**    *\$ caractère de fin de fichier*
- Concaténation : **cat f1 f2 f3 > f123**
- Ajout : **cat f1 >> f2**    f1 est concaténé à la suite de f2

**more** : affiche un fichier page par page

**cp** : copy

- copy d'un fichier source dans un fichier destination : **cp essai essai1**
- copy d'un fichier dans un répertoire : **cp essai /home/utilisateur1/projet**
- copy d'un répertoire dans un autre : **cp -r \* /home/utilisateur1/projet**
- copy d'un lien : **cp -rl a\_sauver/ sauv/**

**mv** : move change le nom d'un fichier ou répertoire

**mv fichier1 fichier2** (fichier1 est supprimé, fichier2 est écrasé s'il existe)

**mv toto /home/utilisateur1/projet**

**rm** : supprime un (des) fichier(s)                    **rm toto**

## 4.7 - Manipulation des fichiers

**wc** : (word count ) compte des lignes, mots, caractères

- l : pour le nombre de lignes
- w : pour le nombre de mots
- c : pour le nombre de caractères

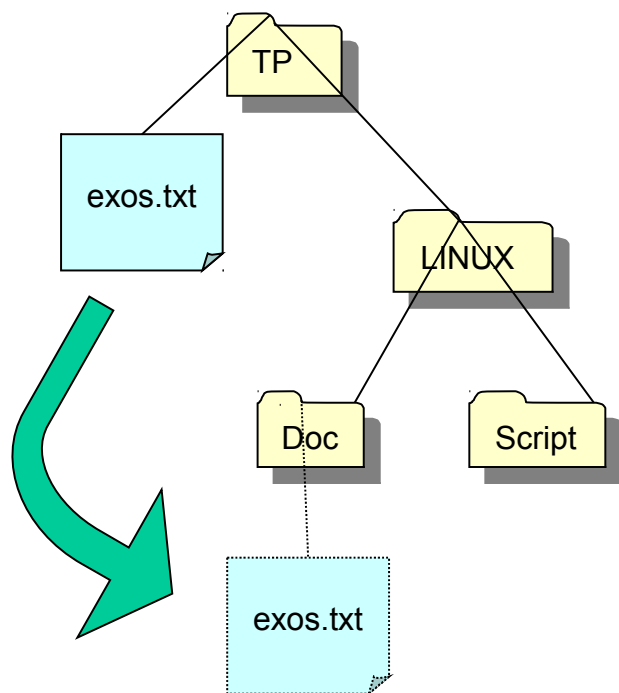
**ln** : Link liens symboliques

```
ln -s f1 new_f2
```

f1 existe, new\_f2 est créé sans occupation disque et est lié à f1

**touch** : crée un fichier vide

## 4.8 – Exemple : manipulation des fichiers



Déplacer le fichier exos.txt du répertoire TP dans le répertoire Doc

```
$ cd TP
$ mv exos.txt LINUX/DOC/.
    ou en moins bien
$ cd TP
$ cp exos.txt LINUX/DOC/.
$ rm exos.txt
```



## 4.9 - Autres Commandes

---

**clear** : effacer l'écran du terminal.

**reset** : réinitialise le terminal

**gzip** : logiciel libre de compression

**bzip2** : commande qui est à la fois le nom d'un algorithme de compression de données et d'un logiciel libre



## 5 - Travailler avec les commande

---

5.1 – les commandes (2)

5.2 - redirection & les pipes

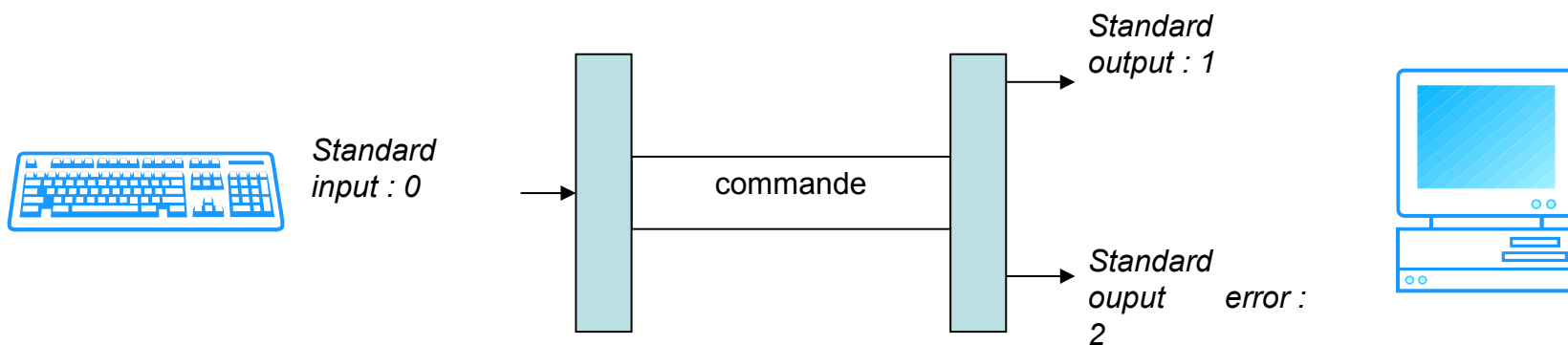
5.3 - Recherche, tri et archivage

5.4 – Exercice 1: sur les redirections

## 5.1 – les commandes (2)

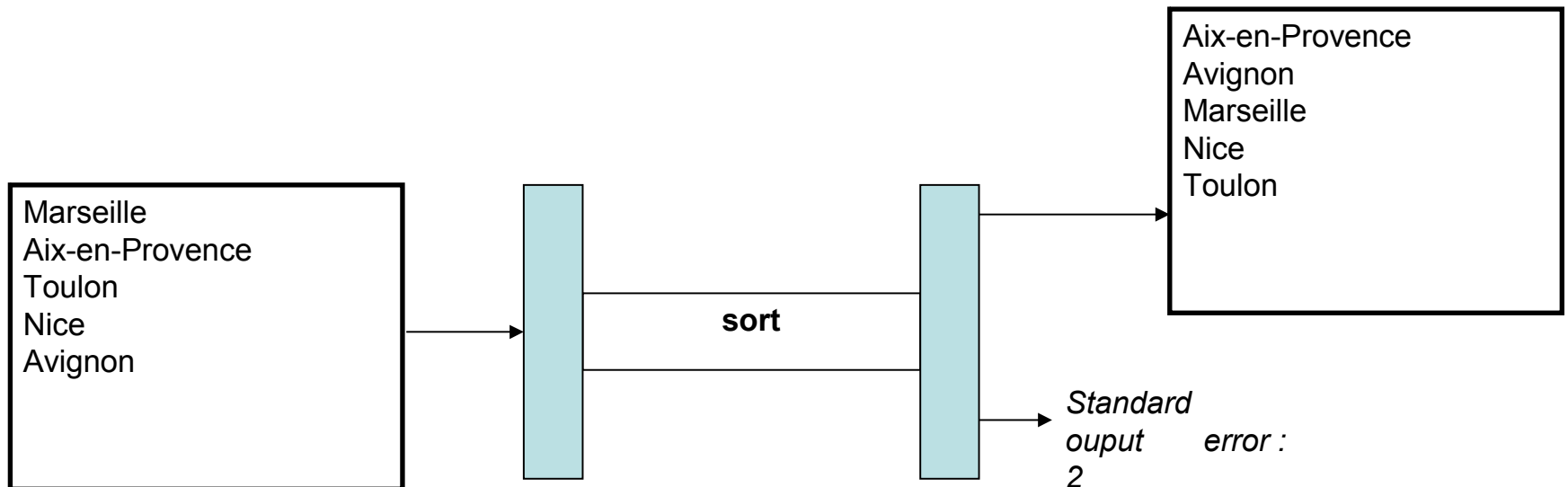
Une commande lit ses données d'entrée dans l'entrée standard (clavier) et écrit les résultats dans la sortie standard (écran).

Périphérique	Association par défaut	Descripteur de fichier
Standard input buffer	clavier	0
Standard output buffer	écran	1
Standard output error	écran	2



## 5.1 : les commandes (2)

Exemple : commande **sort** (qui trie les données d'entrée par ordre alphabétique)



## 5.2 - redirection & les pipes

commande < monFichier *prend comme entrée le fichier monFichier*

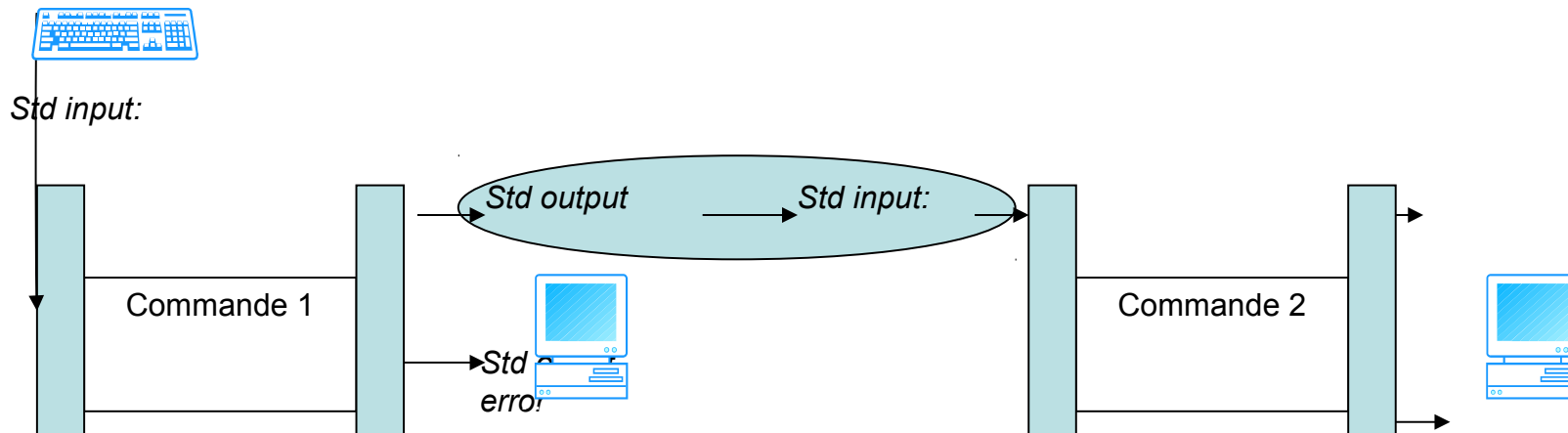
commande > monFichier *envoie la sortie dans le fichier monFichier (crée le fichier, s'il n'existe pas ; l'écrase s'il existe)*

commande >> mon\_fichier *rajoute à la fin du fichier (crée le fichier s'il n'existe pas)*

Exemple : `find / -name "villes" -print 1 > resultat 2 > erreurs`

**Les pipes** (tubes) : la sortie de la première commande devient l'entrée de la deuxième.

exemple : `cat villes | sort`



## 5.2 - redirection & les pipes

Exemple : créer un fichier de noms et le trier

1. Création du fichier lesVilles:

\$ cat > lesVilles

Marseille  
Aix-en-Provence  
Toulon  
Nice  
Avignon

2. **Tri du fichier**  
<ctrl-d> \$ *caractère de fin de fichier, permet de sortir de l'édition*

\$ sort lesVilles

Aix-en-Provence  
Avignon  
Marseille  
Nice  
Toulon

*affiche le résultat à l'écran*

\$ sort lesVilles > lesVillesTries

*sauve le résultat dans le fichier dans lesVillesTries*

2. **Visualiser le résultat**

\$ more lesVillesTries

### Tri

**sort** : permet de lire et de trier le contenu d'un fichier

```
sort [-options] [+c1 [-c2]] ...[fichier ...]
```

-c test si l'entrée est déjà triée

-m fusion des fichiers d'entrée

-d tri sur l'ordre alphabétique

-i caractères de contrôles ignorés

-n tri numérique

-r tri inversé

-tx choix du séparateur de champ x au lieu de espace ou tabulation

-o ftri résultat du tri dans le fichier ftri

```
sort -o essai.tri essai
```

# Archivage

**tar** : permet de sauvegarder une arborescence de fichier

*tar options noms*

- c create pour créer l'archive
- r append : les noms sont stockés à la fin de l'archive même s'ils existent déjà
- u update : additionne noms à l'archive s'ils n'y sont pas , ou si la date est postérieure
- t liste : les noms de l'archive
- x extract : permet d'extraire les fichiers noms de l'archive
- v verbose
- f file : permet de spécifier le nom de l'archive
- p preserve : (avec -x) permet de restaurer les droits d'accès des fichiers
- z zip: compression de l'archive (utilitaires gzip, gunzip)

## 5.3c - Recherche, tri et archivage

*Sauvegarde du répertoire alice sur l'archive /dev/fd0*

```
$ > cd /home
$ > tar -cv -f /dev/fd0 alice
alice/
alice/.bash-profile
alice/.bashrc
alice/bidon
alice/bin/ ...
```

*Extraction du répertoire alice/bidon dans /home*

```
$ > cd /home
$ > tar -xv -f /dev/fd0 alice/bidon
alice/bidon
```

*Compression dans /tmp*

```
$ > cd /home/xstra
$ > tar -cvz -f /tmp/xstra.gz .
```

*Décompression*

```
$ > tar -zx -f /tmp/xstra.gz
```

*Compression dans /tmp*

```
$ > cd /home/xstra
$ > tar -cv -f /tmp/xstra .
$ > gzip /tmp/xstra
```

*Décompression*

```
$ > gunzip /tmp/xstra.gz
$ > tar -x -f /tmp/xstra
```

## 5.3d - Recherche, tri et archivage

**grep** : recherche une chaîne de caractère dans des fichiers

**sed** : recherche de chaînes conformes à un motif donné et les modifie

`sed [-n] [-e commande_sed] fichier_source`

-n inhibe la sortie écran

`sed 's/ancien_motif/nouveau_motif/g' fichier_source`

**awk** : permet de rechercher, modifier, formater, compter, afficher, traduire

...

permet des traitements de textes complexes.

**tr** : (translate) transforme une chaîne en une autre

- **transcodage** `tr` sans option : `tr string1 string2`

Remplacer a par A, ',' par ';' et '/' par '\_'

`tr 'a,/ 'A;_' < fich1 > fich2`

- **suppression de caractères** option `-d` : `tr -d string1`

Supprimer les caractères <carriage-return> et Ctrl-z (Dos->Unix)

`Tr -d'\015\033' < fich1 > fich2`

- **suppression de répétitions** option `-s` : `tr -s string1`

Remplacer plusieurs espaces consécutifs par un seul

`tr -s ' ' < fich1 > fich2`

## 5.4 – Exercice 1: sur les redirections

Donnez la différence entre les commandes suivantes :

- `ls>toto`                    et    `ls >> toto`
- `ls titi 2>erreur`        et    `ls titi > erreur2`
- `ls titi 1>erreur3` et    `ls titi > erreur4`
- `wc -l < toto`            et    `wc -l toto`

Expliquer le résultat des commandes suivantes

- `ls -l | more`
- `who | wc -l`
- `ls | wc -w`
- `ls *.txt *.dat` et `ls *.txt *.dat 2> /dev/null`



## 6 - Administration

---

6.1 - Comptes utilisateurs

6.2 - Création d'un compte

6.3 - Suppression/Modification d'un compte d'utilisateur

6.4 – Groupe

6.5 - La commande passwd

6.6 - Outils graphiques

6.7 : Commande SU & SUDO

6.8 - Protection des fichiers

6.9 - Exemple : droits fichiers et répertoires

6.10 – Exemple : droits fichiers et répertoires

6.11 - Gestion de l'espace disque

## 6.1 - Comptes utilisateurs

---

Une machine Linux peut héberger plusieurs comptes de plusieurs personnes.

Les comptes utilisateur servent sur les systèmes UNIX et Linux à:

- Distinguer les différents utilisateurs qui ont accès au système, pour des raisons de sécurité. Chacun d'eux possède un compte personnel, auquel il accède par un identifiant et un mot de passe secret.
- Définir des permissions d'accès à leurs données, afin d'en autoriser ou d'en interdire l'exploitation par les autres.
- Authentifier chaque utilisateur accédant au système, ce qui permet bien d'autres actes de gestion tels que gérer les courriers électroniques des utilisateurs par exemple ou savoir qui a fait une bêtise en examinant les fichiers de trace du système.

### Utilisateur spéciaux:

le compte **root** utilisé par l'administrateur qui permet de tous faire  
le compte **news** qui permet de recevoir les news Usenet en provenance d'un autre site



## 6.2 - Création d'un compte

La création d'un compte demande plusieurs opérations :

- l'ajout d'une entrée dans le fichier **/etc/passwd**,
- la création du répertoire personnel de l'utilisateur,
- la mise en place de la configuration par défaut dans le répertoire de cet utilisateur.

Ces opérations sont prises en charge par la commande **adduser** ou **useradd** dont la syntaxe se présente comme suit :

```
adduser [-c commentaires] [-d rep_personnel]  
[-e date_expiration] [-f tps_inactivité]  
[-g groupe_initial] [-G groupe[,...]  
[-m [-k squelette_rep | -M] [-p motdepasse]  
[-s shell] [-u uid [-o]] [-n] [-r] utilisateur
```

Utilisée sans aucun argument, cette commande demande de manière interactive toutes les informations nécessaires pour identifier l'utilisateur en cours de création.

Utilisée sans option, donc uniquement avec l'argument nom d'utilisateur, cette commande applique un ensemble de valeurs par défaut qui sont consignées dans le fichier :  
**/etc/default/useradd**



## 6.3 - Suppression/Modification d'un compte d'utilisateur

---

### Supression d'un compte d'utilisateur :

Pour supprimer le compte d'un utilisateur (non connecté), on utilise la commande **userdel** :

#### **userdel [-r] utilisateur**

L'option **-r** supprime aussi le répertoire. personnel et les fichiers de l'utilisateur.

La commande supprime toute trace de l'utilisateur dans le fichier de configuration : */etc/passwd* y compris dans les groupes d'utilisateurs.

### Modification d'un compte d'utilisateur :

Pour modifier le compte d'un utilisateur, on utilise la commande **usermod** :

#### **usermod [options] utilisateur**

Les options sont les mêmes que **adduser**

**usermod -G stagiaire,prof stagex** ajoute **stagex** dans les 2 groupes stagiaire et profs (qui doivent exister au préalable)

Pour lui attribuer le nouveau mot de passe stgx : `passwd stagex`

La saisie du nouveau mot de passe sera demandée deux fois.



## 6.4 - Groupe

Un groupe sous Linux est un ensemble d'utilisateurs qui partagent les mêmes fichiers et répertoires. Les fichiers auront des droits d'accès réglables à ces groupes.

Chaque utilisateur fait partie au moins d'un groupe, son ***groupe initial ou primaire***.

Ainsi, dans ***/etc/passwd*** chaque utilisateur possède un groupe par défaut, précisé par son identifiant ***gid*** dans ce fichier.

L'appartenance au groupe primaire n'étant pas exclusive, **tout utilisateur peut faire partie de plusieurs autres groupes**, appelés ses *groupes secondaires*.

Pour lister tous les groupes (primaire et secondaires) d'un utilisateur : **groups utilisateur**

Pour créer un nouveau groupe : **groupadd groupe**

Supprimer le groupe `.stagiaires` : **groupdel stagiaires** (Le groupe est alors supprimé du fichier ***/etc/group***).

Pour ajouter un utilisateur à un groupe, le plus simple est d'éditer le fichier ***/etc/group*** et d'ajouter une liste d'utilisateurs (séparés par des virgules) sur la ligne du groupe (ou utiliser **Linuxconf**).

## 6.5 - La commande passwd

Elle est chargée du cryptage du mot de passe dans /etc/shadow

Syntaxe : **passwd [option] nom-login**

Options :

- **--stdin** la commande abandonne son caractère interactif habituel et examine son entrée standard pour s'en servir comme mot de passe.
- Très utile dans un script : **echo mot | passwd --stdin** (attention tout caractère est significatif, y compris les " ")
- **-d** pour supprimer le mot de passe, l'utilisateur pourra se connecter sans !
- **-l** pour verrouiller le compte et empêcher sa connexion.
- **-u** pour déverrouiller.



## 6.6 - Outils graphiques

Les trois outils nécessitent tous une connexion au système en tant que root.

### Linuxconf

Linuxconf est un véritable couteau suisse d'administration Linux dont l'utilisation nécessite de se connecter au système en tant que root. En général il se lance en ligne de commande (on peut utiliser **linuxconf** sous **X-KDE sur les systèmes Redhat Section Comptes utilisateurs**)

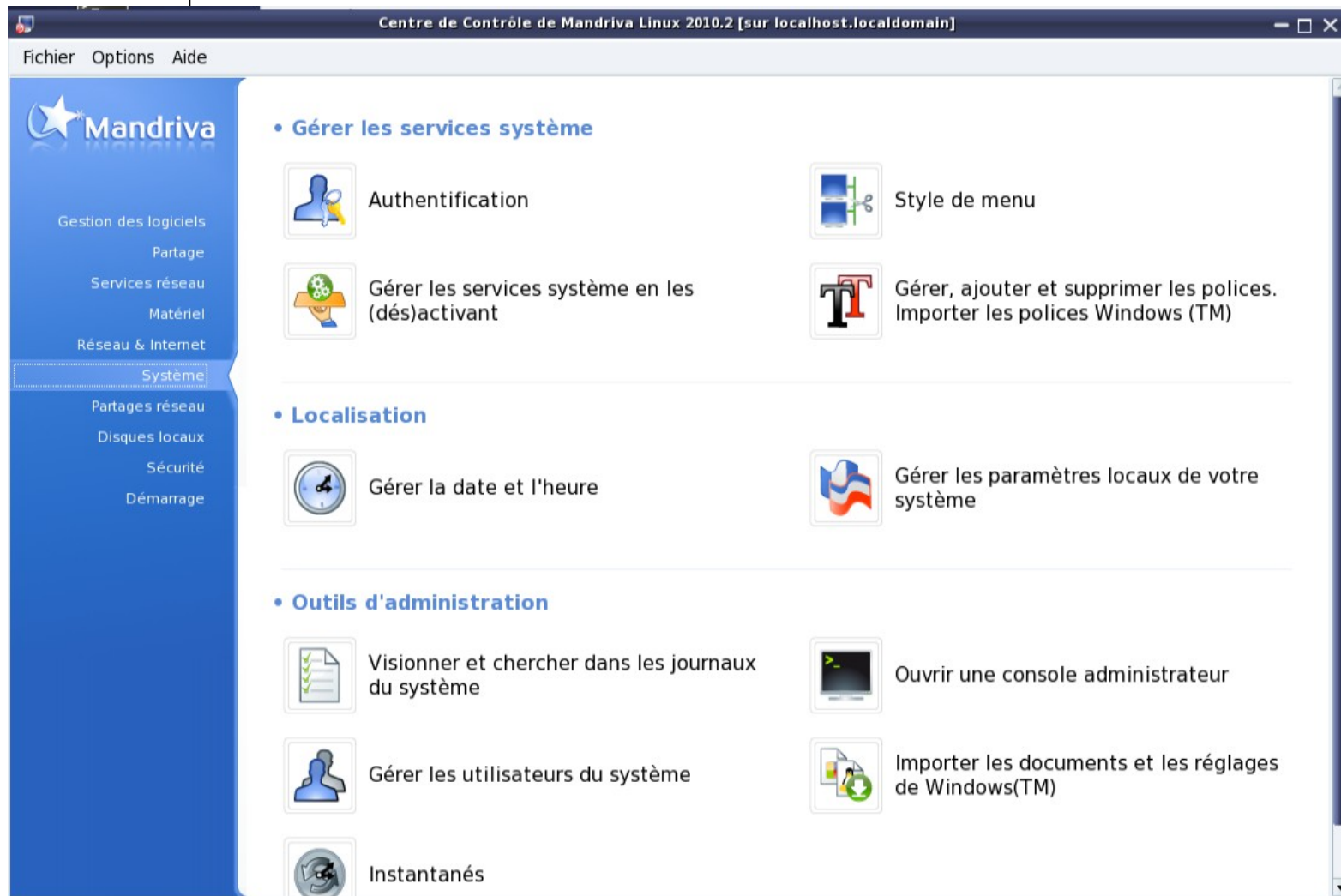
### Kuser

Kuser est une petite application graphique intégrée à l'interface graphique KDE. Pour le lancer sous Linux Mandriva, il faut exécuter la commande Lanceur d'application/Configuration/Other/User Manager ou tout simplement la commande **Kuser** dans une fenêtre console.

### Userdrake

Userdrake est une utilitaire graphique développé par l'éditeur Mandrakesoft. Pour le lancer sous Linux Mandriva, il faut exécuter la commande Lanceur d'application /Configuration/Other/Userdrake ou tout simplement la commande **userdrake** dans une fenêtre console.

## 6.6 - Outils graphiques





## 6.7 : Commande SU & SUDO

---

**su** : Commande qui permet d'exécuter un shell avec un User-ID et un Group-ID différents

**sudo** : Commande qui permet à l'administrateur système d'accorder à certains utilisateurs (ou groupes d'utilisateurs) la possibilité de lancer une commande en tant qu'administrateur, ou comme autre utilisateur, tout en conservant une trace des commandes saisies et des arguments.



## 6.8a - Protection des fichiers

**groupe d'utilisateurs** : pour permettre le partage des fichiers et le travail en équipe

**droits d'accès** : définis par le propriétaire

Pour chaque fichier il existe 3 types d'utilisateurs :

- le propriétaire du fichier (**u**)
- les membres du groupe propriétaire du fichier (**g**)
- les autres utilisateurs (**o**)

Pour chaque fichier et par type d'utilisateur il existe 3 modes d'accès :

- Autorisation de lecture (**r** ou **4**)
- Autorisation d'écriture (**w** ou **2**)
- Autorisation d'exécution (**x** ou **1**)

Ce qui fait 9 bits : rwx rwx rwx

Les différents types de fichiers :

fichier ordinaire (-) , fichier répertoire (d),  
périphérique accédé en mode caractère (c),  
périphérique accédé en mode bloc (b),  
tube nommé (pipe) (p), lien symbolique (l),  
socket (s).



## 6.8b - Protection des fichiers

Exemple : `ls -l /etc/passwd`

```
- rw- r-- r-- 1 root bin 2055 jul 28 18:03 /etc/paswd
```



`chmod` : modification des droits d'accès en mode symbolique

- `chmod u-w file` supprime le droit d'écriture au propriétaire
- `chmod g+r file` ajoute le droit de lecture pour le groupe
- `chmod ug=x` accès uniquement en exécution pour le propriétaire et le groupe, met à vide les autres champs



## 6.8c - Protection des fichiers

`chmod` : modification des droits d'accès en mode octal

Exemple `rw- r-x` est représenté par le nombre octal 765

Une lettre équivaut à un 1, un tiret à 0

111 110 101 = 765

`chmod 777` donne tous les droits à tout le monde

### Droits d'accès aux répertoires

`ls -dl bin`

```
drwx r-x r-x 3 utilisateur1 staff 1024 jul 28 18:04 bin
```

r autorisation en lecture

x autorise l'accès au répertoire

w autorise la création, la suppression et le changement d'un élément du répertoire

## 6.9 - Exemple : droits fichiers et répertoires

Soit un fichier f1 dans le répertoire user1

1. Le répertoire user1 appartenant à l'utilisateur user1 a les protections suivantes:

`drwx --- ---`

Le fichier f1 de user1 a les protections suivantes :

`-rwx --- ---`

☞ Seul user1 peut modifier et supprimer le fichier f1.

1. Le répertoire user1 appartenant à l'utilisateur user1 a les protections suivantes :

`dr-x --- ---`

Le fichier f1 de user1 a les protections suivantes :

`-rwx --- ---`

☞ Seul user1 peut modifier f1, mais il ne pourra le supprimer, car le répertoire n'est pas en w (autorisation de création, suppression, modification du nom de l'élément).



## 6.10 – Exemple : droits fichiers et répertoires

---

En utilisant les commandes `mkdir` et `cat` créer dans un nouveau répertoire le fichier « bienvenue » contenant la ligne de commandes :

```
echo Bienvenue dans le monde Linux
```

Exécutez le fichier (`./bienvenue`).



## 6.11 - Gestion de l'espace disque

du : donne l'occupation du disque ou répertoire en bloc de 1Ko

df : donne le nombre de blocs de 1Ko disponibles sur les disques

### Partition du disque :

- Créer une partition de swap de taille égale à 2 fois le mémoire RAM
- Pour un système Linux personnel, la solution la plus simple est d'attribuer tout le reste à une seule partition
- Sinon, pour un petit groupe, mettre au minimum:
  - /tmp 200 Mo
  - /var 400 Mo
  - /home 200 Mo par utilisateur
  - / tout l'espace restant 3 Go



## 7 - Editeurs

---

7.1 – VI

7.2 - Éditeurs graphiques

Vi est un des éditeurs de texte les plus populaires sous Linux (avec Emacs et pico) malgré son ergonomie très limitée.

Vi est un éditeur entièrement en mode texte, ce qui signifie que chacune des actions se fait à l'aide de commandes texte.

Cet éditeur est très puissant et peut être très utile en cas de non fonctionnement de l'interface graphique.

La syntaxe pour lancer Vi est la suivante :

```
vi nom_du_fichier
```

Une fois le fichier ouvert, il vous est possible de vous déplacer à l'aide des curseurs, ainsi que des touches *h*, *j*, *k* et *l* (au cas où le clavier ne posséderait pas de curseurs fléchés).



## 7.1.1 - Les modes de Vi

---

Vi possède 3 modes de fonctionnement :

**Le mode normal:** celui dans lequel vous êtes à l'ouverture du fichier. Il permet de taper des commandes

**Le mode insertion:** Ce mode permet d'insérer les caractères que vous saisissez à l'intérieur du document. Pour passer en mode insertion, il suffit d'appuyer sur la touche *Insert* de votre clavier, ou à défaut sur la touche *i*

**Le mode de remplacement:** Ce mode permet de remplacer le texte existant par le texte que vous saisissez. Il vous suffit de réappuyer sur *insert* (ou *i*) pour passer du mode insertion au mode remplacement, et d'appuyer sur la touche *Echap* pour revenir en mode normal

## 7.1.2 - Les commandes de base

Commande	Description
<code>:q</code>	Quitte l'éditeur (sans sauvegarder)
<code>:q!</code>	Force l'éditeur à quitter sans sauvegarder (même si des modifications ont été apportées au document)
<code>:wq</code>	Sauvegarde le document et quitte l'éditeur
<code>:file <i>nom du fichier</i></code>	Sauvegarde le document sous le <i>nom</i> spécifié

## 7.1.3 - Les commandes d'editions

Commande	Description
<code>x</code>	Efface le caractère actuellement sous le curseur
<code>dd</code>	Efface la ligne actuellement sous le curseur
<code>dx</code>	Efface <i>x</i> lignes à partir de celle actuellement sous le curseur
<code>nx</code>	Efface <i>n</i> caractères à partir de celui actuellement sous le curseur
<code>x&gt;&gt;</code>	Indente <i>x</i> lignes vers la droite à partir de celle actuellement sous le curseur
<code>x&lt;&lt;</code>	Indente <i>x</i> lignes vers la gauche à partir de celle actuellement sous le curseur

## 7.1.4 - Recherche/ remplacer

---

Pour rechercher un mot dans un document, il vous suffit (en mode normal) de taper / suivi de la chaîne à rechercher, puis de valider par la touche entrée. Il est alors possible d'aller d'occurrence en occurrence grâce à la touche *n*.

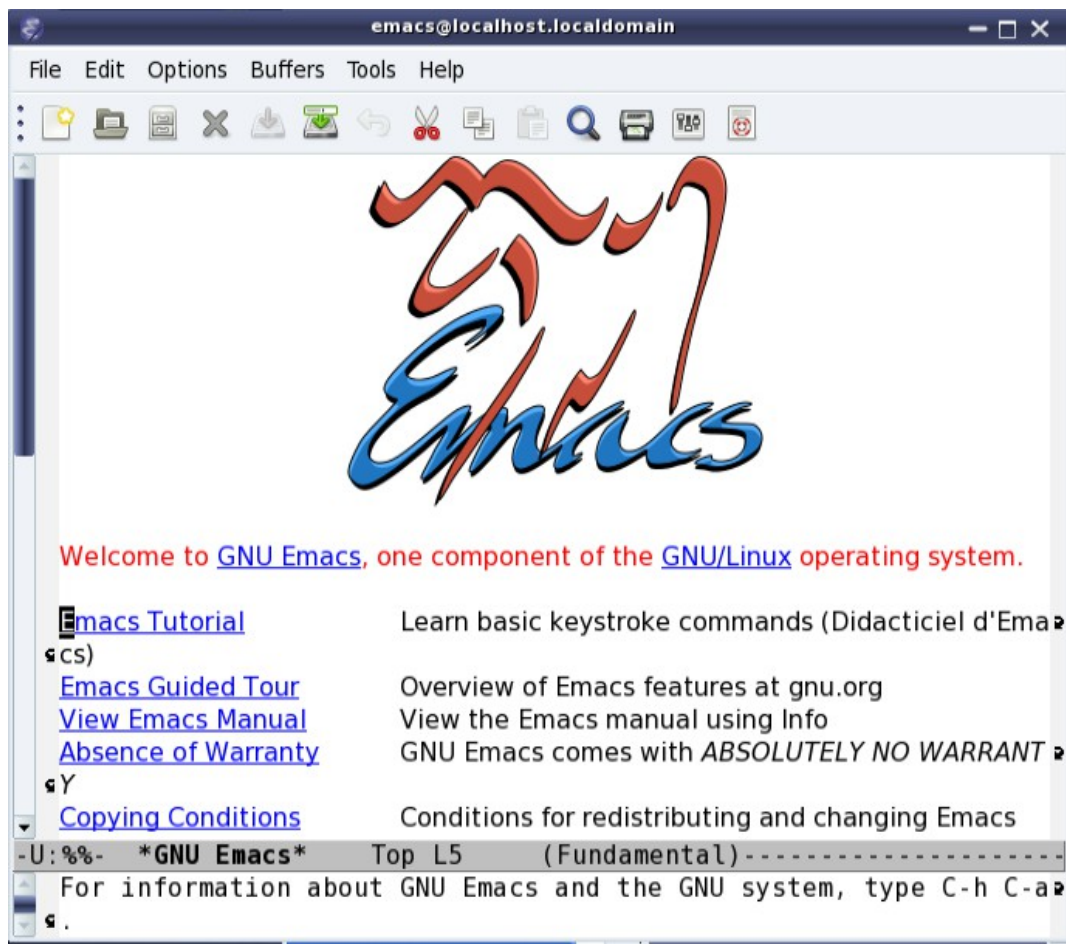
Pour remplacer une chaîne de caractères par une autre sur un ligne, il existe une commande très puissante sous *Vi* utilisant les **expressions régulières**. Voici sa syntaxe :

```
:s/chaine_a_replacer/chaine_replacante/
```

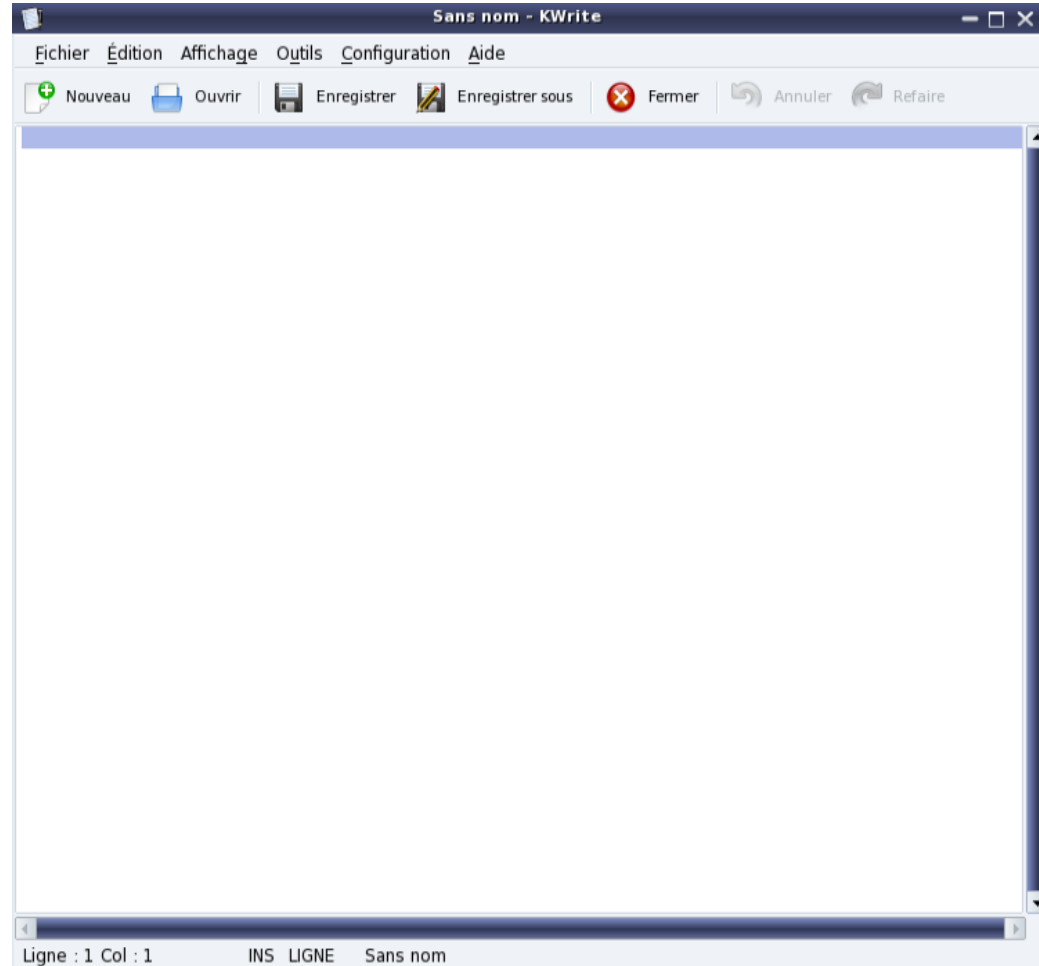
Il est possible de la généraliser à tout le document grâce à la syntaxe suivante :

```
:%s/chaine_a_replacer/chaine_replacante/
```

## 7.2 - Éditeurs graphiques : emacs



## 7.2 - Éditeurs graphiques : kwrite



## 7.2 - Éditeurs graphiques : gedit



The screenshot shows the 'Manuel de gedit - Navigateur d'aide' window. The title bar includes window control buttons and the text 'Manuel de gedit - Navigateur d'aide'. The menu bar contains 'Fichier', 'Édition', 'Aller à', 'Signets', and 'Aide'. The toolbar features 'Précédent', 'Suivant', 'Sommaire de l'aide', and a search box labeled 'Rechercher :'. The main content area is titled 'Manuel de gedit' and contains a table of contents with ten items: I. Introduction, II. Premiers pas, III. Travailler avec des fichiers, IV. Travailler avec du texte, V. Recherche et remplacement de texte, VI. Impression, VII. Aides à la programmation, VIII. Raccourcis clavier, IX. Préférences, and X. Greffons. A right-hand sidebar lists the same items under the heading 'À propos de ce document'. The status bar at the bottom shows 'Ligne : 1 Col : 1', 'INS LIGNE', and 'Sans nom'.

Manuel de gedit - Navigateur d'aide

Fichier Édition Aller à Signets Aide

Précédent Suivant Sommaire de l'aide Rechercher :

### Manuel de gedit

- I. Introduction
- II. Premiers pas
- III. Travailler avec des fichiers
- IV. Travailler avec du texte
- V. Recherche et remplacement de texte
- VI. Impression
- VII. Aides à la programmation
- VIII. Raccourcis clavier
- IX. Préférences
- X. Greffons

À propos de ce document

- Manuel de gedit
- Introduction
- Premiers pas
- Travailler avec des fichiers
- Travailler avec du texte
- Recherche et remplacement de texte
- Impression
- Aides à la programmation
- Raccourcis clavier
- Préférences
- Greffons

Ligne : 1 Col : 1    INS LIGNE    Sans nom

## 7.2 - Éditeurs graphiques : vim

```
VIM - Vi Amélioré

version 7.2.284
par Bram Moolenaar et al.
Vim est un logiciel libre

Aidez les enfants pauvres d'Ouganda !
tapez :help iccf<Entrée> pour plus d'informations

tapez :q<Entrée>           pour sortir du programme
tapez :help<Entrée> ou <F1> pour accéder à l'aide en ligne
tapez :help version7<Entrée> pour lire les notes de mise à jour

0,0-1          Tout
```



## 8 - Démarrage

---

8.1 – Démarrage système

8.2 – Commande : runlevels

8.3 – programme d'amorçage





## 8.1 – Démarrage système

Au démarrage ou à l'arrêt d'une machine Linux plusieurs tâches doivent être effectuées.

Ces tâches peuvent correspondre à des fonctionnalités de base ou à des tâches plus ????. Le premier processus à être lancé sur un système Unix est le programme **init**, dont le but est, entre autres, de lancer tous les autres processus nécessaires à l'utilisation d'une machine.

La gestion de tous les services installés sur une machine passe par des scripts situés dans le répertoire */etc/init.d/*. Ce répertoire contient un script par service

Ces scripts peuvent être exécutés en leur passant un paramètre spécifique :

- start : démarre le service
- stop : arrête le service
- restart : redémarre le service
- reload : force la relecture du fichier de configuration sans arrêter le service
- status : affiche l'état actuel du service (démarré, éteint, ...)

## 8.1 – Démarrage système

Exemple : pour lancer un serveur apache, il faut exécuter cette commande : `/etc/init.d/apache2 start` ou pour le redémarrer : `/etc/init.d/apache2 restart`.

Cette liste n'est pas exhaustive et n'est pas applicable à tous les scripts.

Ces scripts sont le moyen de contrôler les différents services installés sur votre machine. Lors de l'installation d'un paquet correspondant à un service, un script est automatiquement ajouté dans ce répertoire.

Chaque état de la machine est associé à un runlevel. il y a 7 runlevels différents sur un système Unix Voici les runlevels des systèmes

Debian/Ubuntu :

- Runlevel 0 : Arrêt de la machine
- Runlevel 1 : Mode maintenance (mono-utilisateur en ligne de commande)
- Runlevel 2 à 5 : Mode multi-utilisateur complet avec serveur graphique si installé
- Runlevel 6 : Redémarrage de la machine



## 8.2 – Commande : runlevels

---

**dmesg** : est une commande sur les systèmes d'exploitation de type Unix qui affiche la mémoire tampon de message du noyau.

**runlevel** : lit le fichier système utmp (typiquement /var/run/utmp) pour trouver l'enregistrement du niveau d'exécution et affiche sur sa sortie standard les niveaux d'exécution courant et précédent, séparés par une espace. S'il n'y a pas de niveau d'exécution précédent, la lettre N sera affichée à la place.

## 8.3 – programme d’amorçage

---

**GNU GRUB** (acronyme signifiant en anglais « GRand Unified Bootloader ») est un programme d'amorçage de micro-ordinateur. Il s'exécute à la mise sous tension de l'ordinateur, après les séquences de contrôle interne et avant le système d'exploitation proprement dit, puisque son rôle est justement d'en organiser le chargement. Lorsque le micro-ordinateur héberge plusieurs systèmes (on parle alors de multi-amorçage), il permet à l'utilisateur de choisir quel système démarrer

**LILO** (Linux Loader) est, tout comme **GRUB**, un chargeur d'amorçage libre, diffusé sous la licence BSD. Il permet de choisir la partition à amorcer lors du démarrage de l'ordinateur.



Yantra Technologies

## 9 - monitoring des ressources

---

9.1 - Les processus

9.2 - Les processus : le job control

9.3 - Les processus : quelques commandes

9.4 - Les processus : la commande ps

9.5 - Les processus : quelques commandes  
(suite)



## 9.1 - Les processus

- Programmes en cours d'exécution
- Identifié par un numéro unique PID (processus Identifier)

### Deux sortes de processus :

1. Les processus systèmes (daemons) : init, crond, inetd
2. Les processus utilisateurs : interpréteur de commande, exécution d'un programme utilisateur

### Exécution d'une commande

1. Le mode interactif (foreground) *commande*
2. Le mode en arrière-plan (background) *commande &*
3. Le mode différé *at 20:00:00 20/01/07 < commande*
4. Le mode batch : on place la commande dans une file d'attente *batch*
5. Le mode cyclique : *crontab*



## 9.2 - Les processus : le job control

- Le **job** est une ligne de commandes shell. Il est composé d'un ou plusieurs processus.
- Chaque job est numéroté de 1 à N par le shell.
- Un job peut se trouver dans 3 états :
  - avant-plan (foreground) on n'a plus la main en shell
  - arrière-plan (background) on a la main en shell
  - suspendu (suspend), le job est en attente.

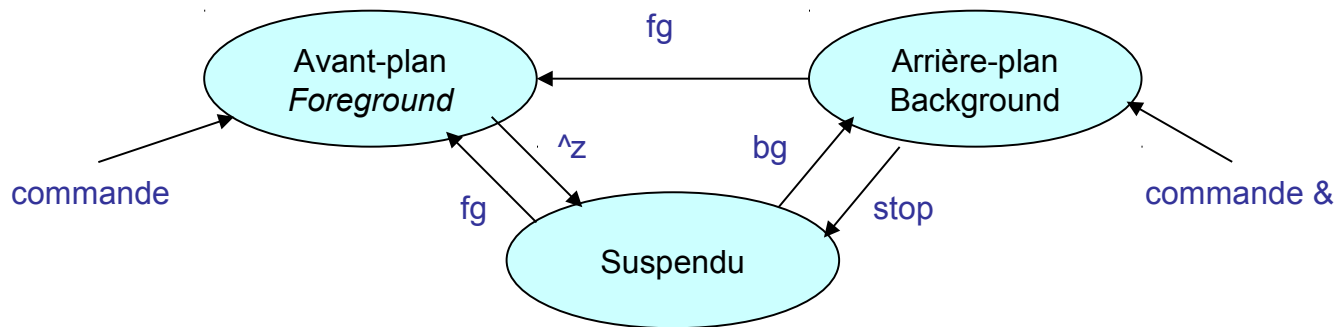
*fg %numéro-job* fait passer le processus de background en foreground

*<ctrl-z>* fait passer le processus de foreground en stoppé (suspendu)

*fg %numéro-job* fait passer le processus stoppé en foreground

*bg %numéro-job* fait passer le processus stoppé en background

*<ctrl-c>* interrompt le processus interactif





## 9.3 - Les processus : quelques commandes

### La commande kill

kill [-signal] numéro\_PID  
kill [-signal] numero-job  
kill -l donne la liste des signaux

```
kill -9 1941  
kill -1 %2
```

<ctrl-c> pour arrêter le processus en cours d'exécution

### La commande jobs

```
jobs -l  
[1] 2655 Stopped sleep 100  
[2] - 2975 Running beta &  
fg %1 la commande sleep est en avant-plan
```

```
sleep 60 &
```

### La commande sleep

### La commande nohup

Pour éviter qu'un processus en background soit interrompu quand on ferme une session :

```
essai &  
nohup essai &
```

## 9.4 - Les processus : la commande ps

Permet d'avoir l'état des processus en cours

Par défaut seuls les processus de l'utilisateur sont affichés.

-u nom\_utilisateur permet de sélectionner un utilisateur

### Les options

-e tous les processus en cours

-f donne :

- *Le nom de l'utilisateur: UID*
- *Le numéro du processus : PID*
- *Le numéro du processus père : PPID*
- *L'heure de lancement : STIME*
- *Le nom du terminal TTY*
- *Le temps d'exécution : TIME*

ps

```
PID   TTY   TIME  CMD
pts/1 0:00  ps
pts/1 0:20  -ksh
```

ps -ef

```
UID   PID  PPID  C  STIME  TTY   TIME  CMD
root   1    0    0  may 18  ?    0:00  init
root  359   1    0  may 18  ?    0:00  inetd

uti1 11784 11731 0  13:45  ?    0:00  /bin/bash
```

## 9.5 - Les processus : quelques commandes (suite)

---

**top** : affiche les processus linux

**netstat** : Affiche les connexions réseau, les tables de routage, les statistiques des interfaces, les connexions masquées, les messages netlink, et les membres multicast.

**lspci** : est un utilitaire permettant d'afficher des informations sur les bus PCI du système ainsi que sur les périphériques qui y sont connectés.

**lsusb** : est un utilitaire qui vous permet d'afficher des informations relatives aux bus USB du système et aux périphériques qui y sont connectés.



## 10 - Communications

---

- 10.1 - Les communications
- 10.2 - Les communications
- 10.3 - Des outils TCP/IP
- 10.4 - Les fichiers associés



# ftp : Transferts de fichiers

ftp [option] [nom\_machine]

```
prompt> ftp beta.u-strasbourg.fr
prompt> name:  passwd:
ftp> ls
  file1
ftp> binary           transfert de type binaire
ftp> get file1 tempo  transfert dans tempo
ftp> quit
prompt> ls
  adm/ bin/ tempo src/
```

**telnet** : Emulation d'un terminal sur le réseau

**telnet** nom\_machine

```
prompt> telnet beta.u-strasbourg.fr
prompt> login:   passwd:
beta> ls
  file1
beta> logout
prompt>
```

**ssh** : en réseau plus sécurisé que telnet

**rlogin, rsh** : Emulation d'un terminal sur le réseau

**mail, talk** : courrier électronique



## 10.3 - Des outils TCP/IP

**ping** : permet de vérifier si une machine distante répond.

`ping -c 5 192.168.0.1` pour envoyer 5 pings à la machine dont l'adresse IP est 192.168.0.1

**ifconfig** : permet de connaître la configuration des cartes réseau, de changer celle-ci. On peut utiliser `linuxconf` aussi.

**route** : permet de voir, d'ajouter ou d'enlever les routes.

`route -n`

**netstat** : permet de connaître les ports utilisés

`netstat -ntl` ports ouverts

`netstat -nr | grep 195.98.246.28`

**traceroute** : permet de voir la route prise par un paquet.

**nslookup** : permet d'interroger un serveur de nom (DNS).

**nmap** : permet de connaître les ports ouverts sur une machine distante.

`nmap 192.168.0.*`



## 10.4 - Les fichiers associés

---

`/etc/hosts` : Liste des couples numéro IP, nom symbolique

`/etc/resolv.conf` : liste des serveurs de noms disponibles et leur adresse IP

`/etc/hosts.equiv` : équivalences de noms

`/etc/services` : services réseaux disponibles (nom, numéro de port, protocole)

Sous `$HOME` environnements spécifiques de l'utilisateur

`.rhosts`

`.netrc` identique à `rhosts` pour ftp et telnet

`.forward` pour rediriger les messages

### **Des livres**

- *Linux Initiation et utilisation* Armspach, Colin, Ostré-Waerzeggers Dunod
- *Programmation système en C sous Linux* Christophe Blaess Eyrolles
- *Le noyau Linux* P.Bovet, Cesati O'Reilly

### **Commandes Unix**

- [http://www.infres.enst.fr/~danzart/fiches/unix\\_abrege.html](http://www.infres.enst.fr/~danzart/fiches/unix_abrege.html)

### **Shell scripts**

- <http://www.accreteil.fr/reseaux/systemes/linux/shelscripts/shell-bash.html>
- **Avanced bash scripting guide** : <http://www.tldp.org/LDP/abs/html/>
- **C Shell** : <http://www.shellunix.com/csh.html>
- **Bourne Shell** : <http://www.shellunix.com/sh.html>
- **Korn Shell** : <http://www.shellunix.com/ksh.html>