

# Cours Linux

version 1.1

## DU Administrateur de bases de données 2010

Franck Corsini

Tome 1

*Module 1 - Prise en mains et généralités*

*Module 2 - Commandes et manipulations de base*

Franck Corsini.

Document sous licence Creative Commons Paternité-Pas d'Utilisation Commerciale-Partage des Conditions Initiales à l'Identique 2.0 France





## DU Administrateur Base de Données Fondamentaux Linux

Présentation et prise en main

### *Sommaire*

- Présentation de la formation Unix ABD
- Généralités Unix - Linux
- Premiers contacts
  - Utilisation du clavier
  - login, mot de passe, connexion
  - les shells
  - commandes Unix
- Documentation
- Editeurs de texte (vi)

- 5 jours
  - 5, 7, 12, 18 et 29 mai
- Organisation
  - Alternance cours – mise en pratique par des séances de TP
- Objectifs
  - Savoir ouvrir une session
  - Savoir lancer une commande
  - Se débrouiller avec un shell
  - Savoir manipuler les fichiers sous Unix
  - Connaître les constituants d'une arborescence unix
  - Des commandes utiles pour un DBA Unix

- Vous disposerez
  - d'un compte sur le serveur de la formation permanente
  - d'une adresse de courriel <nom>@formation.jussieu.fr
  - Webmail :  
<https://webmail.formation.jussieu.fr/squirrelmail>
- Pour se contacter
  - [franck.corsini@ipsl.jussieu.fr](mailto:franck.corsini@ipsl.jussieu.fr)
  - vos @ mail personnelles
  - liste de diffusion : [abd2010@formation.jussieu.fr](mailto:abd2010@formation.jussieu.fr)
- Cours et TP (avec corrigés) disponibles en ligne
  - <http://www.formation.jussieu.fr/abd/>

## Sommaire

- Présentation de la formation Unix ABD
- Généralités Unix - Linux
- Premiers contacts
  - Utilisation du clavier
  - login, mot de passe, connexion
  - les shells
  - commandes Unix
- Documentation
- Editeurs de texte (vi)

- Unix : un système d'exploitation
  - Ensemble de programmes de base dont le but est de :
    - Interagir avec les composantes matérielles de la machine
    - Fournir un environnement d'exécution aux programmes utilisateurs
  - **Noyau** : programme principal du SE chargé en mémoire au démarrage du système qui gère
    - la mémoire vive
    - les accès aux périphériques, aux fichiers
    - l'espace du disque dur (mémoire de masse)
    - l'exécution des programmes
    - les droits d'accès

▪ **Programme**

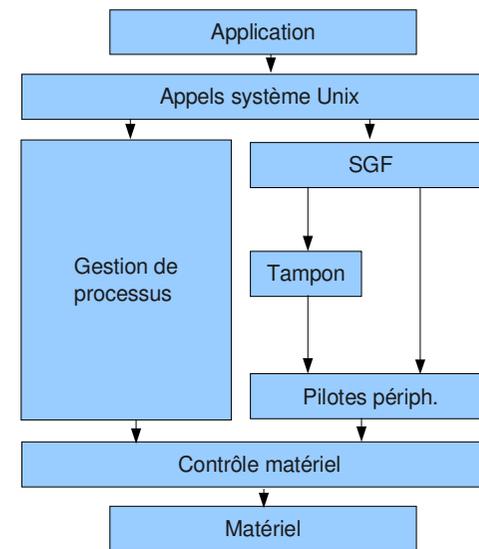
- Fichier binaire stocké sur un disque
- Chargé en mémoire centrale en cours d'exécution

▪ **Processus**

- Une instance d'un **programme** en cours d'exécution dans un **contexte** (environnement de processus)

▪ **Contexte**

- Le système d'exploitation fourni au programme des informations pour assurer une exécution correcte



▪ Terminologie

<b>Mono utilisateur</b>	Une seule personne utilise l'ordinateur
<b>Multi utilisateur</b>	Plusieurs personnes peuvent utiliser le système en même temps. Le système s'assure qu'un utilisateur n'interfère pas avec un autre
<b>Mono tâche</b>	Un seul processus tourne à un instant
<b>Multi tâche</b>	Plusieurs processus donnent l'impression de fonctionner en même temps
<b>Multi tâche préemptif</b>	Le système détermine alloue et arbitre les temps CPU à chaque processus
<b>Multi tâche non préemptif</b>	Le processus détermine lui-même le temps CPU qui lui est nécessaire

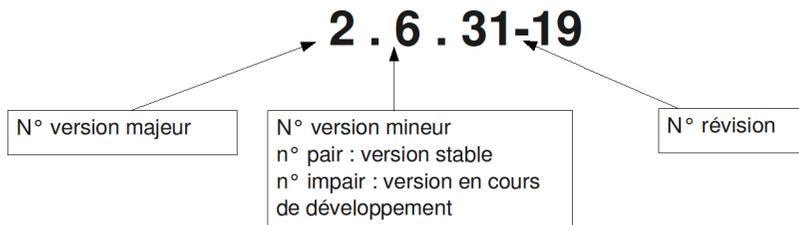
▪ Exemples

MS DOS	mono utilisateur, mono tâche
Windows 95/98/ME	mono utilisateur, multi tâche non préemptif
Windows NT, 2000, XP, 2003	mono utilisateur, multi tâche préemptif
Windows 2003 + Terminal Server	multi utilisateur
IBM OS/2	mono utilisateur, multi tâche préemptif
Unix	multi utilisateur, multi tâche préemptif

- Unix est un système **multi-tâches**
- Permettre une exécution **concurrente** et **indépendante** de plusieurs applications initiées par plusieurs utilisateurs
  - Concurrence : les applications demandent l'accès à des ressources partagées (processeur, mémoire, disques durs,...)
  - Un ordonnanceur arbitre les demandes
  - Indépendance : chaque application s'exécute de manière autonome et indépendante des autres applications

- Unix est un système **multi-utilisateurs**
- Tout programme exécuté sur un système Linux est étroitement lié à un utilisateur
  - Authentification de l'utilisateur
  - Protection de l'exécution des programmes
  - Comptabilité des ressources allouées à l'utilisateur
- Un utilisateur correspond à :
  - un identifiant (généralement avec 8 caractères) (login)
  - un mot de passe

- Version d'un noyau Linux (<http://www.kernel.org>)



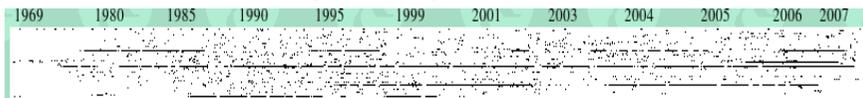
- Noyau **monolithique**
  - Les pilotes de périphériques font partie intégrante du noyau
- Noyau **modulaire**
  - Des éléments du système d'exploitation sont chargés en mémoire et exécutés à la demande

- Du point de vue utilisateur : les Un\*x se ressemblent
- Du point de vue administration : chaque Unix possède ses particularités
  - commandes dont les noms ou les options diffèrent
  - des arborescences différentes...
- Des tentatives de normalisations
  - System V Interface Definition
  - IEEE POSIX
  - X/OPEN – Open Group
- Votre rôle : connaître les mécanismes fondamentaux d'Unix afin de pouvoir vous adapter aux différents systèmes que vous pourrez rencontrer

- Historique Unix
  - ⇒ fin 60's : le laboratoire "Bell Labs" d'AT&T développe en collaboration avec le MIT un OS : Multics
  - ⇒ 1969 : Ken Thompson et Dennis Ritchie écrivent un système de gestion de fichiers, quelques utilitaires basiques (cp, rm) et un interpréteur de commande (shell)  
Ecriture en assembleur
  - ⇒ 1970 : Brian Kernighan suggère le nom de UNIX
  - ⇒ 1973 : ré-écriture d'Unix en C (premier système écrit dans un langage portable)
  - ⇒ Bell Labs distribue Unix sous forme de code source :  
apparition de différentes familles d'Unix

- ....
  - ⇒ 2 catégories d'utilisateurs d'Unix : les universités et les grands constructeurs informatiques
  - ⇒ Université de Berkeley (UCB) à l'origine d'une des plus anciennes branches d'Unix : BSD (Berkeley Software Distribution)
  - ⇒ Autre grande branche : Unix System V vendu par AT&T aux constructeurs
  - ⇒ la marque déposée UNIX® appartient aujourd'hui à l'Open Group (organisme ayant pour but de standardiser Unix)

- Historique Unix complexe (<http://www.levenez.com>)



- Aujourd'hui il reste 2 grandes familles d'UNIX
  - ⇒ la famille issue de BSD
    - FreeBSD, OpenBDS, NetBSD, Darwin (couche Unix de Mac OS X)
    - Tru64 Unix (Dec/Compaq)
  - ⇒ la famille issue de System V
    - Solaris
    - HP-UX
    - AIX
    - IRIX...

- et Linux ?

- Linux

- ⇒ n'est pas un système Unix au sens propre car son code ne provient pas du code originel (c'est un "Unix-like")
- ⇒ on parle de système GNU/Linux



```
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: Free minix-like kernel sources for 386-AT
Message-ID: <19910ct5.054106.4647@klaava.Helsinki.FI>
Date: 5 Oct 91 05:41:06 GMT
Organization: University of Helsinki
...
As I mentioned a month(?) ago, I'm working on a free version of a
minix-lookalike for AT-386 computers. It has finally reached the stage where it's
even usable (though may not be depending on what you want), and I am willing to put
out the sources for wider distribution. It is just version 0.02 (+1 (very small) patch
already), but I've successfully run bash/gcc/gnu-make/gnu-sed/compress etc under it.
Sources for this pet project of mine can be found at nic.funet.fi
(128.214.6.100) in the directory /pub/OS/Linux.
...
```

- 1991
  - Linus Torvald propose à la communauté son système d'exploitation **Linux** issu de Minix
  - En 1991, seuls quelques applications peuvent fonctionner sous Linux ( bash, gcc, sed) ...
- ...aujourd'hui Linux est devenu une plate-forme de référence pour les applications critiques des entreprises
- Raison du succès
  - Code source sous licence GPL (GNU Public Licence)
  - Communauté mondiale de développement (hackers)

- Naissance du logiciel libre
  - Richard Stallman – MIT
  - 1984 : rupture avec le modèle social du logiciel propriétaire
  - Le logiciel doit être partagé entre les utilisateurs et librement modifiable
  - Projet GNU (GNU's not Unix) - <http://www.gnu.org> - <http://www.fsf.org>
  - Logiciels : Licence GPL (GNU Public Licence)
  - Documents : Licence FDL (Free Documentation Licence)



- **Free Software Foundation (<http://www.fsf.org>)**

- Contrat entre le détenteur des droits patrimoniaux et un utilisateur (licencié)
- Richard Stallman – MIT
- 1984 : rupture avec le modèle social du logiciel propriétaire
- Le logiciel doit être partagé entre les utilisateurs et librement modifiable
- Développer un système d'exploitation Unix-like **libre** : projet GNU (GNU's not Unix) - <http://www.gnu.org>



- **Open Source Initiative**

- Autre définition du logiciel libre en 10 points

- **Définition du logiciel libre (4 libertés)**



1. **Liberté d'exécuter le programme pour quelque motif que ce soit**
2. **Liberté d'étudier le fonctionnement du programme et de l'adapter à ses besoins (nécessité de pouvoir accéder au code source pour le modifier)**
3. **Liberté de redistribuer des copies gratuitement ou contre une somme d'argent (« free » synonyme de liberté et non de gratuité)**
4. **Liberté d'améliorer le programme et de publier les améliorations pour en faire profiter la communauté**



- **Licence GPL**
  - ➔ GNU Public Licence – Copyleft ou « gauche d'auteur »
  - ➔ Empêche le logiciel libre d'être modifié et de devenir propriétaire
- **Linux étroitement lié au projet GNU**
  - ➔ Linux = noyau du système
  - ➔ Exploite dès sa création des logiciels du projet GNU
  - ➔ On parle plus généralement des systèmes « GNU/Linux »

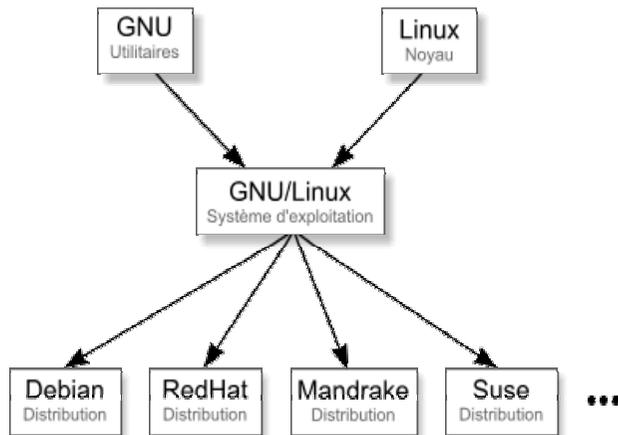
- Comparaison : logiciel libre – gratuitel – partagiciel – logiciel propriétaire
- Gratuitels et partagiciels : pas plus que le droit d'exécuter
- Télécharger « librement » un logiciel n'en fait pas un logiciel libre

	Logiciel Libre	Gratuitel (freeware)	Partagiciel (shareware)	Logiciel Propriétaire
UTILISER	OUI	OUI	LIMITE (durée ou fonctionnalités)	LIMITE (selon utilisation)
COPIER	OUI	OUI	OUI	NON (sauf copie de sauvegarde)
ETUDIER	OUI	NON	NON	NON
MODIFIER	OUI	NON	NON	NON

- Où trouver les logiciels libres ?
  - Sourceforge (<http://sourceforge.net>)
  - Framasoft (<http://www.framasoft.net>)
  - Adullact (<http://adullact.net>). Orienté collectivités, administrations
  - Les sites dédiés à une application (apache, isc,...)
- Sous quelle forme les récupérer ?
  - Archive (tar le plus souvent, zip dans le monde windows) compressée (zip, bzip, rar,...)
  - Directement depuis un dépôt versionné (cvs, subversion, ....)
- Après désarchivage, la lecture du fichier LICENSE.TXT indique le type de licence choisie pour la diffusion

- **Distributions Linux**
- **1993** : téléchargement de Linux depuis Internet sur des disquettes
- **Aujourd'hui** : DVD, Live-CD, clefs USB, machines virtuelles prêtes à l'emploi...
- Une distribution c'est :
  - Un noyau Linux - <http://www.kernel.org>
  - Un ensemble cohérent de logiciels associés compatibles
  - Approches différentes en fonction de l'utilisation : firewall, versions autonomes sur CD (*live*), poste de travail, serveur d'entreprise
  - Payant ou gratuit...ou les deux (téléchargement gratuit mais CD-ROM payant)
- <http://distrowatch.com> - 360 distributions Linux différentes!





- Les distributions Linux ont des avantages par rapport à Microsoft Windows
  - Faible consommation en mémoire vive
  - Faible consommation en espace disque
  - Très grande stabilité
  - Grande rapidité en multi-tâche, même sur un système mono-processeur
  - Evolution très (trop ?) rapide
- Totalement paramétrable (graphiquement et techniquement)
- Système très complet, utilisable dès la fin de l'installation (intègre de nombreux outils gratuits de qualité)
- Parfaitement adapté au mode multi-utilisateur

- Mais il y a aussi sur des inconvénients :
  - Manque d'applications spécifiques (jeux, video etc..) ou commerciales
  - Relative complexité
  - Problème sur le support en général
  - Problème de pilotes avec certains matériels (souvent ceux dont les constructeurs ne donnent pas les spécifications, et arrive dans le monde linux avec un retard vis à vis des pilotes Windows/Mac fournis)

- Cas particulier Linux : quelle distribution choisir ?
  - Pas simple : chaque utilisateur défend "sa" distribution Linux,
  - Gratuite ou payante ?
    - Acheter la distribution
    - Télécharger gratuitement la distribution
  - chaque distribution a ses qualités, aucune n'est exempte de défauts,
  - site disponible donnant des informations:
    - <http://distrowatch.com>
  - Bien définir ces besoins (serveurs, poste de travail multimédia, etc..)

- “La cathédrale et le bazar” : essai de **Eric Raymond**
  - Il y défend la notion de logiciel *open source* plus axé sur la qualité vis à vis de la notion de *logiciel libre* centrée sur une “éthique” du logiciel portée par Richard Stallman
  - Il oppose :
    - le « bazar » : modèle de développement du noyau Linux (communauté de développement, forte adaptabilité)
    - à la « cathédrale » : modèle de développement des logiciels propriétaires (organisation hiérarchique, adaptabilité plus faible)
- <http://www.linux-france.org/article/these/cathedrale-bazar/>

- Quelques règles de bon sens
  - le pire ennemi, c'est vous :
    - `rm /tmp *`
    - `rm -rf *`
  - pas de manipulations importantes en situation d'urgence : reculez de la machine et prenez le temps de réfléchir à ce que vous aller faire
  - pas de manip importante un vendredi après-midi
  - DOCUMENTEZ ce que vous faites

- ...

- soyez sur de revenir en arrière

```
mv fichier fichier.orig  
cp fichier.orig fichier  
vi fichier
```

si mes modifications ne fonctionnent pas comme prévu : je reviens au fichier d'origine et ... je réfléchis !

- Mieux que la copie de sauvegarde : un serveur de versionnement

- faites comme si vous ne pouviez pas revenir demain

## Sommaire

- Présentation de la formation Unix ABD
- Généralités Unix - Linux
- Premiers contacts
  - Utilisation du clavier
  - login, mot de passe, connexion
  - les shells
  - commandes Unix
- Documentation
- Editeurs de texte (vi)

- N'ayez pas peur d'expérimenter
- En mode utilisateur, les manipulations et « bêtises » ne pourront pas mettre en péril votre système
- En mode administrateur : vous avez tous les droits sur le système ... y compris celui de tout effacer !
  - ⇒ règle de base : il faut limiter le mode administrateur au minimum et l'utiliser uniquement lorsqu'il la manipulation l'exige

- Petits rappels pour les dyslexiques du clavier....



Appuyer sur cette touche donne « à »



+



Appuyer sur ces deux touches donne « 0 »

crédit images claviers : <http://www.freinet.org>



Shift  
Lock

- Attention au verrouillage des majuscules : souvent la cause de mot de passe incorrects !
- Mieux vaut ne pas l'utiliser

- touche tab, tabulation



- touche escape, esc, Echap, échappement ( eq. Ctrl+[)

Esc



- touche Alt Gr



Appuyer sur cette touche donne « à »



Appuyer sur ces deux touches donne « 0 »



Appuyer sur ces deux touches donne « @ »

- Un utilisateur Unix est équivalent à :
  - ➔ un identificateur (sur 8 caractères en général mais 32 possible) ou login
  - ➔ un mot de passe confidentiel
- Attention : utilisation frauduleuse du compte d'un autre utilisateur entraîne des sanctions pénales
  - ➔ Accès ou maintien frauduleux dans un système : 2 mois à 2 ans d'emprisonnement et de 300 à 7500 € d'amende
  - ➔ Accès ou maintien frauduleux dans un système avec suppression ou modification des données ou avec altération du fonctionnement du système : 2 mois à 2 ans d'emprisonnement et de 1500 à 15000 € d'amende
- Respectez et faites respecter les chartes informatiques en vigueur sur votre lieu d'activité

- Règles sur les mots de passe
  - ➔ ne se prête pas !
  - ➔ ne s'oublie pas !
    - exemple : Kptain[&1GD
  - ➔ ne doit pas être facile à trouver
    - évitez tout ce qui se rapporte à vous (nom, prénom, marque voiture, nom du toutou)
    - pas de mots issus d'un dictionnaire
    - doit comporter au moins 6 caractères
    - utilisez des associations de majuscules, minuscules, chiffres et caractères spéciaux

- Modification du mot de passe
  - ➔ La commande standard pour changer de mot de passe sur une machine Unix est `passwd`
  - ➔ L'authentification sur les postes de travail de la formation permanente utilise un mécanisme de centralisation des mots de passe (NIS)
    - il faudra utiliser la commande `yppasswd`

- Les comptes utilisateurs sont gérés de plusieurs façons :
  - base de compte locale
    - fichier /etc/passwd (comptes)
    - fichier /etc/shadow (mot de passe)
  - base de compte NIS
  - base de compte LDAP
  - base de compte Kerberos
  - ....



```
$ cat /etc/passwd
....
franck:x 501 501:Franck:/home/franck:/bin/bash
```

UID

GID



```
# cat /etc/shadow
....
franck:
$1$vo.BXVa1$jhdOVkmLSxxx12zu.yIkp0:12342:0:99
999:7:::
```

Mot de passe (chiffré)

- Le superutilisateur **root**
  - Utilisateur qui a tous les pouvoirs. Linux ne lui applique pas de mécanisme de protection particulier
  - Equivalent au compte Administrateur sous Windows
- Compte à utiliser pour réaliser :
  - Les tâches de maintenance du système,
  - gérer les comptes des autres utilisateurs
  - Déclarer des imprimantes...
- **Attention avec le compte root !**
  - L'utilisation de ce compte doit être réservée aux seules tâches qui le nécessitent.
  - Revenir à un compte d'utilisateur standard dès que possible.

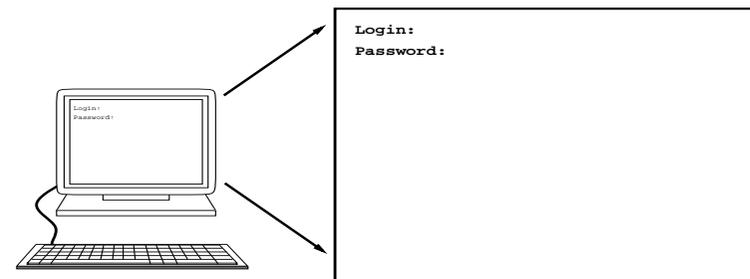
- Comment devenir root sans changer de session ?
  - Pour les session en mode texte
  - commande `su` (*substitute user*)
  - `$ su`
- Comment quitter l'identité de root ?
  - commande `exit`

```
[franck@machine ~]$ su
Mot de passe :
[root@machine franck]#
```

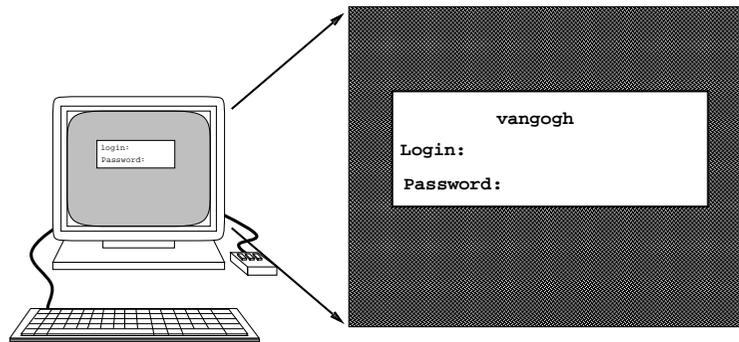
```
[root@machine franck]# exit
exit
[franck@machine ~]$
```

- Avant de pouvoir interagir avec Linux, l'utilisateur doit ouvrir une **session** sur le système
- Cela correspond à l'authentification d'un utilisateur sur le système.  
Le système entre alors en attente des commandes de l'utilisateur.
- Session textuelle ou session graphique (X-Window)

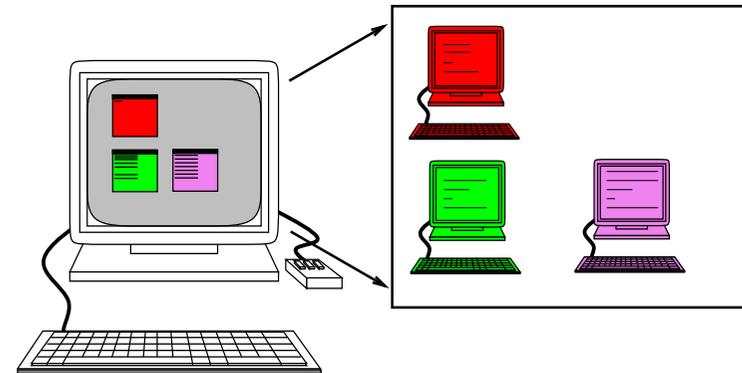
- Connexion sur un terminal texte



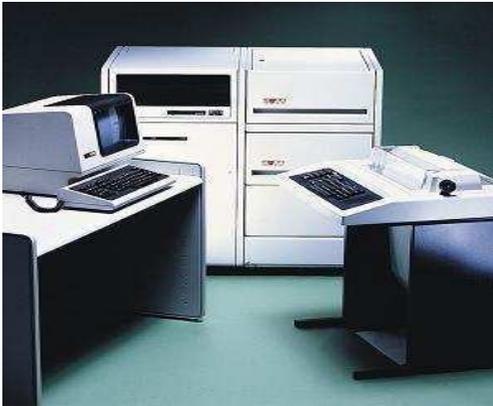
- Connexion sur un terminal graphique



- Une fois connecté via l'interface graphique, on utilise un programme d'émulation de terminal type texte qui fournit dans une fenêtre une connexion comme sur un terminal texte



- A l'origine : des teletypes puis des consoles texte
- Ici pas d'interface graphique : l'interaction utilisateur se fait par l'intermédiaire de mots frappés au clavier

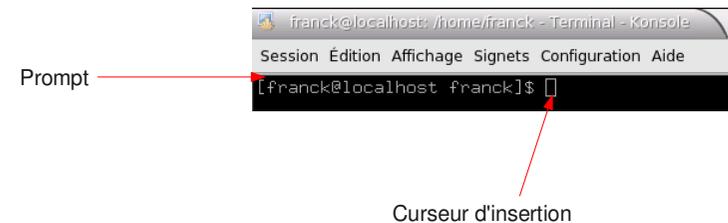


Console texte Digital VT100  
et teletype Digital

- Le shell (« coquille ») est un programme qui permet la saisie et l'interprétation de ce qui est tapé. C'est l'interface entre le système et l'utilisateur.
- Ici pas d'interface graphique : l'interaction utilisateur se fait par l'intermédiaire de mots frappés au clavier
- C'est à la fois...
  - Un interpréteur de commandes (un programme qui boucle en attente des commandes utilisateur)
  - Un langage de programmation (interprété) offrant les structures de base comme tout autre langage
  - Sous Unix, le shell est un programme au même titre qu'un autre. On peut donc changer de shell si besoin.

- shells les plus répandus :
  - ➔ sh (bourne shell) : disponible sur toute plate-forme Unix
  - ➔ bash (bourne Again shell) : Distribué sur la majeure partie des Linux; version améliorée de sh et csh
  - ➔ ksh (korn shell) : Bourne shell amélioré par AT&T
  - ➔ csh (C shell) : shell développé pour BSD
- Un shell est associé par défaut à chaque utilisateur. Il est défini dans le fichier `/etc/passwd`

- Tous les shells se présentent sous la même forme :
  - ➔ une chaîne de caractères affiche que le shell attend que l'utilisateur tape une commande au clavier : le prompt
  - ➔ un curseur qui va se déplacer au fur et à mesure de la saisie des commandes



```
franck@localhost: /home/franck - Terminal - Konsole
Session Édition Affichage Signets Configuration Aide
[franck@localhost franck]$
```

1. Attente d'une commande.  
Invite de commande (prompt)

```
franck@localhost: /home/franck - Terminal - Konsole
Session Édition Affichage Signets Configuration Aide
[franck@localhost franck]$ ls
```

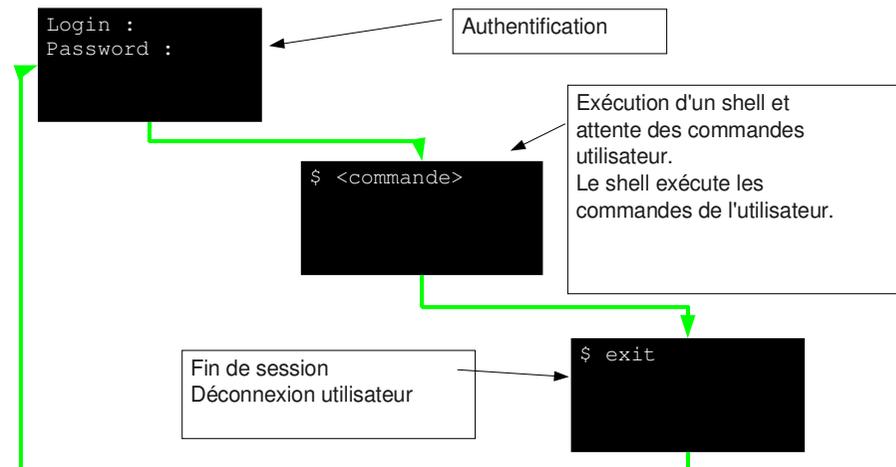
2. Saisie d'une commande.  
Validation avec [Entrée]

```
franck@localhost: /home/franck - Terminal - Konsole
Session Édition Affichage Signets Configuration Aide
[franck@localhost franck]$ ls
comptes          evolution/
cours/           netscape-1686-pc-linux-gnu-ins
cours-linux-mk/  netscape-installer/
Desktop/         tmp/
Documents/      translog.20040125211824.log
[franck@localhost franck]$
```

3. Lecture puis analyse de la ligne  
de commande.  
Exécution du programme et attente  
de la fin d'exécution

4. Retour à l'invite de  
commande

Globalement, une session utilisateur c'est :



- La suite du cours comportera des exemples comme :

```
[franck@machine] $ ls
```

- Attention ! Il ne faudra pas taper la chaîne du prompt lorsque vous testerez les commandes indiquées
- Pour terminer une session shell, on tape la commande commune à tous les shells :

```
$ exit
```

- ➔ dans le contexte d'un terminal texte, `exit` retourne sur la demande de login
- ➔ dans le contexte d'une émulation de terminal texte en environnement graphique, `exit` ferme le programme d'émulation uniquement

- commande unix = un ensemble de mots séparés par des caractères blancs (espace, tabulation)
  - ➔ premier mot : nom de la commande
  - ➔ reste des mots : paramètres de la commande
- parmi ces paramètres, les options, modifient le comportement de la commande
- format d'une commande type :

```
$ commande [options] parametres
```

- ➔ les crochets [ et ] encadrent les éléments facultatifs de la commande
- ➔ les crochets ne doivent pas apparaître dans la ligne de commande

- La syntaxe pour spécifier une option a été définie dans le programme de la commande
- Il existe pourtant des notations que l'on retrouve dans la plupart des commandes (mais il existe des exceptions)
  - ⇒ une option est introduite par le signe `-` et est souvent constituée d'une lettre
    - \$ `cp -r -v source dest`
  - ⇒ l'ordre des options n'a pas d'importance et on peut les cumuler
    - \$ `ls -a -l = $ ls -l -a = $ ls -al`
  - ⇒ les options constituée d'un mot entier sont souvent introduite par le signe `--`
    - \$ `urpmi --auto-update`

- Erreur classique : oubli de l'espace entre la commande, les options et les paramètres
  - ⇒ le shell ne reconnaît pas la commande et affiche un message d'erreur
  - ⇒ les paramètres ou options ne sont pas valides pour la commande : la commande affiche une erreur
  - ⇒ résultat inattendu de la commande

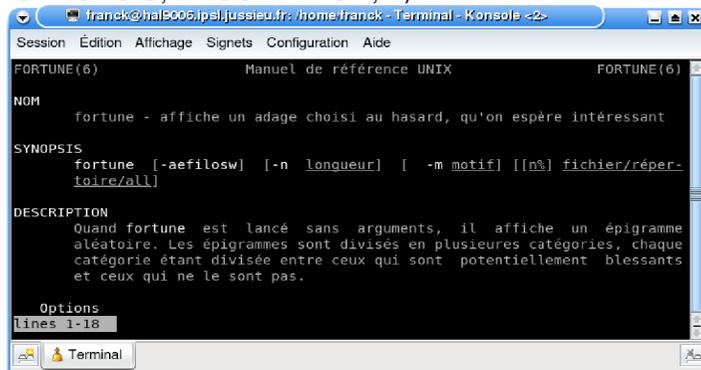
```
[franck@localhost ~]$ ls -al
total 6011
drwxr-xr-x 63 franck franck   3072 mar  7 21:33 ./
drwxr-xr-x  6 root   root    1024 fév 20 12:30 ../
-rw-rw-r--  1 franck franck 95684 déc 13 16:57 24488-
linuxevolution.tar.bz2
.....
[franck@localhost ~]$ ls-al
bash: ls-al: command not found
```

## Sommaire

- Présentation de la formation Unix ABD
- Généralités Unix - Linux
- Premiers contacts
  - Utilisation du clavier
  - login, mot de passe, connexion
  - les shells
  - commandes Unix
- Documentation
- Editeurs de texte (vi)

- Il existe une aide en ligne sur les commandes accessibles pendant le fonctionnement du système : les pages de manuel  
  
man
- Syntaxe de la commande man
  - `$ man [options] commande`
- Inconvénient : il faut connaître le nom de la commande (nom anglais très souvent). Les pages de manuels sont plus une référence sur les nombreuses options des commandes et leur syntaxe particulière
- Les pages de manuel sont LA référence de l'utilisateur débutant à l'expert

- Exemple : `$ man fortune`
  - affichage des informations page par page
  - rubriques constantes pour chaque manuel (NAME, SYNOPSIS, DESCRIPTION,...)



```
franck@hal5006.lpsl.jussieu.fr: /home/franck - Terminal - Konsole <2>
Session Édition Affichage Signets Configuration Aide
FORTUNE(6) Manuel de référence UNIX FORTUNE(6)
NOM
fortune - affiche un adage choisi au hasard, qu'on espère intéressant
SYNOPSIS
fortune [-aefilow] [-n longueur] [-m motif] [[n%] fichier/ré-
pertoire/all]
DESCRIPTION
Quand fortune est lancé sans arguments, il affiche un épigramme
aléatoire. Les épigrammes sont divisés en plusieurs catégories, chaque
catégorie étant divisée entre ceux qui sont potentiellement blessants
et ceux qui ne le sont pas.
Options
lines 1-18
```

- navigation dans les pages
  - barre d'espace ou touche « `f` » pour avancer
  - touche « `b` » pour reculer
  - on quitte le manuel lorsqu'on arrive à la fin
  - on quitte à tout moment en appuyant sur « `q` »
- pour plus d'informations sur l'utilisation de la commande man
  - `$ man man`

- man (manual)
- Synopsis  

```
man [options] [section] nom-objet
man [options] nom-fichier-man
```
- But : visualiser des pages de manuel (manpages)
- Fichier de conf : `/etc/man.config`
- Utilisation d'outils de mise en page de pages textes (*pager*)
  - ➔ `more` (« one more page » )
  - ➔ `less` (« less is more » )
- Sur les Linux récents, man est associé à la commande `less -is` pour la visualisation des pages
- Définition du pager par la variable PAGER du fichier `/etc/man.config`

- Format des pages de manuel (parties les plus courantes)

En-tête	Description
Name	Nom suivi d'une courte description
Synopsis	Syntaxe de la commande
Description	Description de la commande
Options	les différents options possibles
Return values	pour ce qui est lié à la programmation
See also	Références complémentaires
Bugs	....
Files	Fichiers associés à l'objet (fichiers de conf d'une commande par exemple)
Authors	les auteurs et traducteurs le cas échéant

- Commande `info`
- La FSF (Free software foundation) a fondé la documentation des commandes GNU sous format Texinfo accessibles depuis la commande `info`
- Format Texinfo commun à l'affichage et à l'impression
- Navigation à l'aide de liens hypertext comme avec un navigateur (utilise les commandes emacs)
- `info info` OU `man info`

```

franck@localhost:~
Fichier  Édition  Affichage  Terminal  Onglets  Aide
File: info.info, Node: Top, Next: Getting Started, Up: (dir)

Info: An Introduction
*****

The GNU Project distributes most of its on-line manuals in the "Info
format", which you read using an "Info reader".  You are probably using
an Info reader to read this now.

  There are two primary Info readers: `info', a stand-alone program
designed just to read Info files, and the `info' package in GNU Emacs,
a general-purpose editor.  At present, only the Emacs reader supports
using a mouse.

  If you are new to the Info reader and want to learn how to use it,
type the command `h' now.  It brings you to a programmed instruction
sequence.

  To read about advanced Info commands, type `n' twice.  This brings
you to `Advanced Info Commands', skipping over the `Getting Started'
chapter.

--zz-Info: (info.info.gz)Top, 29 lines --Top-----
Welcome to Info version 4.11. Type ? for help, m for menu item.

```

- Commande `whatis`
- Recherche des mots-clefs à l'intérieur d'une base de données
- Elle est constituée des mots présents dans la section NAME de la page de manuel
- Base de données alimentée par la commande `makewhatis`
- `makewhatis` est appelée régulièrement par l'intermédiaire de script présent dans `/etc/cron.daily` et `/etc/cron.monthly`
- `makewhatis -w` utilise la commande `manpath` (ou `man - path`) pour connaître les répertoires dans lequel il doit rechercher les informations
- Cette base se trouve dans le fichier `/var/cache/man/whatis`

```
[franck@localhost ~]$ whatis mkfs
mkfs          (8) - build a Linux file system
mkfs          (8) - Créer un système de fichiers Linux
mkfs.minix [mkfs] (8) - Créer un système de fichiers Minix
```

```
[franck@localhost ~]$ man -f mkfs
mkfs          (8) - build a Linux file system
mkfs          (8) - Créer un système de fichiers Linux
mkfs.minix [mkfs] (8) - Créer un système de fichiers Minix
```

```
[franck@localhost ~]$ more /var/cache/man/whatis
3ds2m          (1) - Convertit les mailles (meshes) d'un fichier 3DS en un
n m-file (fichier contenant les mailles)
3dsdump        (1) - Affiche l'information sur la structure interne d'un
fichier 3DS
411toppm       (1) - convert Sony Mavica .411 image to PPM
9wm            (1) - un gestionnaire de fenêtres pour X ressemblant à
a2p           (1) - Awk to Perl translator
a2ps          (1) - formate des fichiers pour impression sur une imprima
.....
```

- Commande `apropos`
- Recherche d'une chaîne de caractères dans la base de données `whatis`
- Le résultat est donc beaucoup moins précis
- `man -k` est équivalent à `apropos`

```
[franck@localhost ~]$ whatis mkfs
mkfs          (8) - build a Linux file system
mkfs          (8) - Créer un système de fichiers Linux
mkfs.minix [mkfs] (8) - Créer un système de fichiers Minix
[franck@localhost ~]$ apropos mkfs
mke2fs [mkfs]  (8) - create an ext2/ext3 filesystem
mkfs          (8) - build a Linux file system
mkfs          (8) - Créer un système de fichiers Linux
mkfs.minix [mkfs] (8) - Créer un système de fichiers Minix
```

```
[franck@localhost ~]$ whatis mk
mk: nothing appropriate
[franck@localhost ~]$ apropos mk
ExtUtils::Mkbootstrap (3pm) - make a bootstrap file for use by DynaLoader
ExtUtils::Mksymlists (3pm) - write linker options files for dynamic extension
glib-mkenums      (1) - C language enum description generation utility
growisofs        (1) - combined mkisofs frontend/DVD recording program
mkbootdisk       (8) - créer une disquette de démarrage autonome pour le système en place
mk_cmds          (1) - error table compiler
.....
```

- `/usr/share/doc` ou `/usr/doc`
- Contient de la documentation dans un format différent des pages man : ASCII, postscript, html, pdf.
- On y trouve plutôt des tutoriels, des documentations techniques complémentaires, des README, des HOWTO

```
[franck@localhost doc]$ find /usr/share/doc -type f | wc -l
14037
[franck@localhost doc]$ find /usr/share/doc -name *.pdf | wc -l
20
[franck@localhost doc]$ find /usr/share/doc -name *.html | wc -l
5681
[franck@localhost doc]$ find /usr/share/doc -name *.ps | wc -l
18
```

- ... et quand on a rien trouvé comme documentation interne (le développeur n'a pas fait son travail correctement) il reste l'aide "codée" dans la commande
- Beaucoup d'informations dans les commentaires des fichiers de configuration
- Affichage de l'aide intégrée à la commande avec option `-h` ou `-help` ou `-?`

- Le projet LDP (Linux Documentation Project) regroupe un grand nombre de doc (HOWTO, Readme, guides de référence,...)
- Url à connaître : <http://www.tldp.org>
- <http://www.gnu.org>
- <http://www.kernel.org>
- Newsgroups
  - comp.os.linux.\*
  - alt.os.linux
- Archives newgroups
  - Rechercher sur <http://groups.google.com>
- Mailing lists (listes de diffusion)

## Sommaire

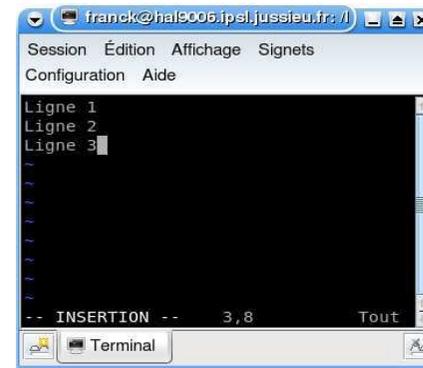
- Présentation de la formation Unix ABD
- Généralités Unix - Linux
- Premiers contacts
  - Utilisation du clavier
  - login, mot de passe, connexion
  - les shells
  - commandes Unix
- Documentation
- Editeurs de texte (vi)

- Panorama d'éditeurs de fichier textes
- Il existe beaucoup d'éditeurs de texte mais seuls quelques uns (sufisament stables et robustes) suffisent
  - ➔ `vi` : le seul éditeur de texte standard sous Unix
  - ➔ `emacs` : puissant mais plus complexe que `vi`
  - ➔ `xemacs` : version graphique d'`emacs`
  - ➔ `nedit` : graphique et simple d'utilisation
- En pratique : il faut savoir au minimum utiliser `vi`.
  - ➔ C'est celui que vous retrouverez sur tout Unix
  - ➔ inconvénient : il demande de la pratique
- `vim` : version améliorée de `vi` (<http://www.vim.org>)

- Vi fonctionne sous **3 modes**
  - ➔ **Commande** : saisie possible d'une commande (souvent une touche) pour agir sur le document.  
C'est le mode par défaut au démarrage
  - ➔ **Edition** (*Insert*): permet la saisie du texte
  - ➔ **Ex** (*mode dernière ligne*) : saisie des commandes relatives à la gestion de fichiers (entre autres)
- Vi est un éditeur en mode texte
  - ➔ vous utilisez le clavier à la fois pour éditer le texte du document (insérer-supprimer des caractères) et pour lancer les commandes du logiciel (gestion des fichiers par exemple)

- Le plus difficile au début : savoir dans quel mode on se trouve !
  - ➔ voir le mode en bas de l'écran à gauche.  
INSERT ou INSERTION apparaît si vous êtes en mode Edition
  - ➔ s'il n'y a rien, vous êtes en mode Commande
  - ➔ si le curseur est en bas de l'écran après « : », vous êtes en mode Ex

- Démarrer vi
  - ➔ `$ vi` : démarre vi sur un nouveau document
  - ➔ `$ vi exemple.txt` : démarre vi en ouvrant le document *exemple.txt*



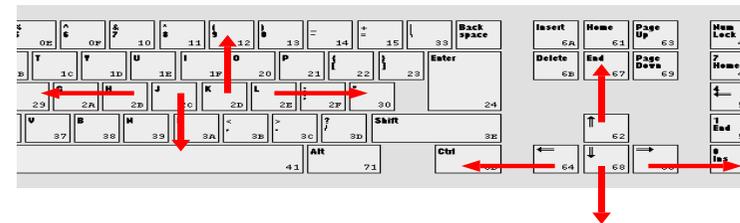
• Commandes passant en mode saisie de texte

Séquence	Action
i	Insérer à la position courante du curseur
a	Insérer à la position suivante du curseur
I	Insérer en début de ligne
A	Insérer en fin de ligne
o	Ouvrir une nouvelle ligne en dessous du curseur
O	Ouvrir une nouvelle ligne au dessus du curseur
cw	Changer un mot
c\$	Changer jusqu'à la fin de la ligne

• Sortie du mode saisie de texte et passage au mode Commande  
⇨ touche ESC

• Commande de déplacement

Séquence	Action
h ou gauche	déplacer le curseur d'un caractère à gauche
l ou droit	déplacer le curseur d'un caractère à droite
j ou bas	déplacer le curseur d'une ligne vers le bas
k ou haut	déplacer le curseur d'une ligne vers le haut
nombre G	aller à la ligne « nombre »
CTRL + G	affiche le numéro de la ligne courante



- Commandes principales

Séquence	Action
x	détruire le caractère sous le curseur
r	remplacer le caractère sous le curseur
dd	effacer la ligne courante
dw	effacer le mot sous le curseur
.	répéter la dernière commande
J	joindre la ligne suivante avec la ligne courante
/toto n	rechercher le mot toto dans le texte en avant répéter la dernière recherche
u	annulation de la dernière commande (undo)
CTRL + L	rafraîchir l'écran

- Commandes de sauvegarde et de sortie de vi

Séquence	Action
:w	sauver le fichier éditer
:w toto	sauver le fichier dans toto
:q	quitter vi
:q!	quitter vi sans sauvegarder
:wq ou ZZ	sauver puis quitter vi
:e toto	éditer maintenant le fichier toto
:r toto	importer maintenant le fichier toto
:e!	éditer la copie courante du fichier (abandon des modifications)

- Attention : toutes ces commandes se font en mode Ex
  - touche ESC + « : » pour passer en mode Ex

- Commandes de copier/coller

Séquence	Action
yy	copie la ligne courante
p	colle dans le texte le contenu du presse-papier
nombre y	copie « nombre » lignes
:n,my	copie de la ligne n à la ligne m (mode Ex)

- Commandes de substitution

Séquence	Action
:s/toto/titi/	sur la ligne du curseur, remplacer le premier mot « toto » par « titi »
:s/toto/titi/g	sur la ligne du curseur, remplacer tous les mots « toto » par « titi »
:1,\$s/toto/titi/	de la ligne 1 à la dernière ligne (\$), remplacer le premier mot « toto » par « titi »
:1,\$s/toto/titi/g	de la ligne 1 à la dernière ligne (\$), remplacer tous les mots « toto » par « titi »
:%s/toto/titi/g	remplace de la première à la dernière ligne chaque mot « toto » par « titi »



- Bibliographie

- cours AOI Thierry Besançon  
<http://www.formation.jussieu.fr>

- Une brève histoire D'Unix – Marie-Lan Nguyen  
<http://www.tuteurs.ens.fr/unix/histoire.html>

- tout sur   
<http://www.gnu.org>

## DU Administrateur Base de Données Fondamentaux Linux

Commandes et manipulations de base

### *Sommaire*

- Notions d'objets Unix
- Notions de chemins
- Manipulation de base des objets
  - cd, pwd, ls, mkdir, mv, rm
- Liens hard, liens symboliques
- Commandes de base
  - cat, more, less, wc, diff, head, tail, touch, file

- Sous Unix, plusieurs type d'objets
  - ➔ fichiers
  - ➔ répertoires
  - ➔ objets associés aux disques durs, clefs USB, bandes
  - ➔ objets destinés à la communication entre applications
- On manipule le plus souvent
  - ➔ fichiers
  - ➔ répertoires
- Une règle que l'on vérifiera souvent
  - ➔ « Sous Unix, tout est fichier »

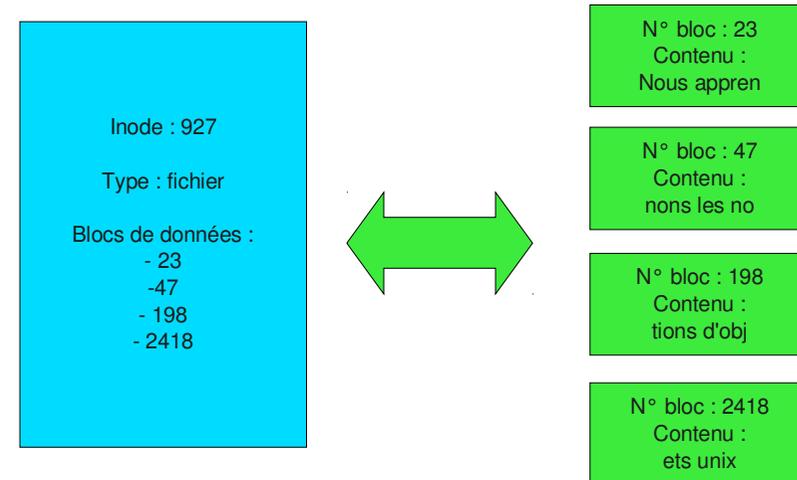
- Sous Unix, la « casse » est importante : le système fait la différence entre les majuscules et les minuscules

```
[aoi@test]$ ls -l
total 0
-rw-r--r-- 1 franck franck 0 mar 13 10:27 exemple.txt
-rw-r--r-- 1 franck franck 0 mar 13 10:26 exeMPLE.TXT
-rw-r--r-- 1 franck franck 0 mar 13 10:26 Exemple.txt
-rw-r--r-- 1 franck franck 0 mar 13 10:26 EXEMPLE.txt
```

- Sous Unix
  - ➔ on évite les caractères espace, apostrophe, guillemets, lettres accentuées dans les noms d'objet
  - ➔ on utilise généralement : les lettres minuscules, les lettres majuscules, le tiret « - », le point « . », l'underscore « \_ »

- Sur un système de fichier Unix, les objets (fichiers) sont gérés par l'intermédiaire d'une « meta-donnée » stockée dans le système de fichier appelée **inode**
  - chaque inode dispose d'un numéro identifiant le fichier
  - il renseigne sur le type d'objet (fichier, répertoire,...)
  - il possède la liste des blocs de données qui correspondent à son contenu
  - Attention : le nom de l'objet ne se trouve pas dans l'inode. Le nom du fichier est géré par le répertoire.

- Un fichier correspond à un inode de type fichier



- Sous unix, un répertoire est aussi un fichier
  - ce fichier particulier contient une liste d'associations : nom de fichier - inode



N° bloc : 56  
Contenu :  
(toto, 789)

N° bloc : 72  
Contenu :  
(titi, 6754)

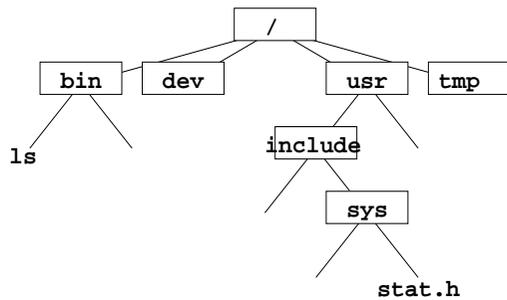
N° bloc : 12048  
Contenu :  
(tata, 8756)

N° bloc : 1768  
Contenu :  
(tutu, 59)

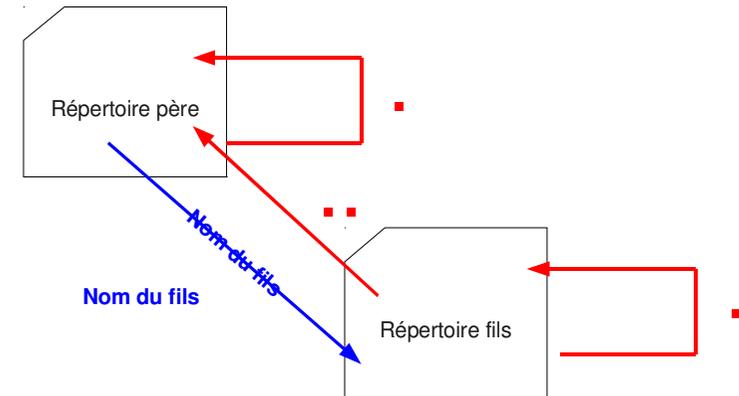
- Le répertoire contient les noms de fichiers ainsi que l'inode correspondant permettant d'y accéder

```
$ vi repertoire1
"
=====
" Netrw Directory Listing (netrw v98)
" /home/franck/UNIX/repertoire1
" Sorted by name
" Sort sequence: [V]$,*,\bak$,\.o$,\.h$,\.info$,\.swp$,\.obj$
" Quick Help: <F1>:help -:go up dir D:delete R:rename s:sort-by x:exec
"
=====
../
./
tata
titi
toto
```

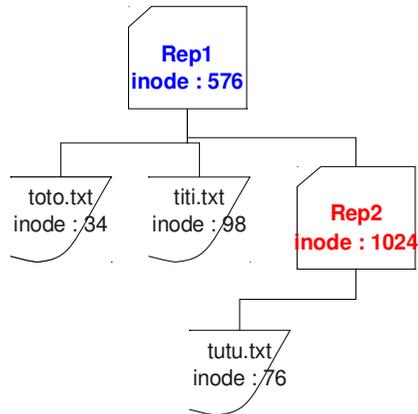
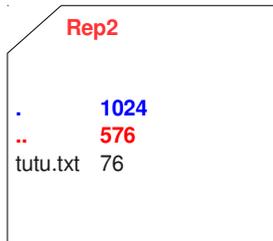
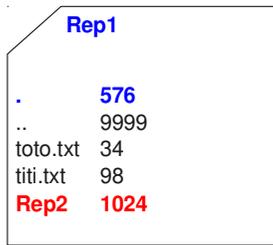
- Un répertoire peut renvoyer sur un autre répertoire et ainsi de suite
  - le système de fichier unix peut être ainsi représenté par un arbre où chaque noeud correspond à un répertoire et chaque feuille à un fichier
  - la racine de l'arbre s'appelle « / » (« slash »)



- Il existe une notion de « descendance » dans les répertoires



- Donc où se trouve stocké le nom d'un répertoire s'il ne se trouve pas dans l'inode ?...



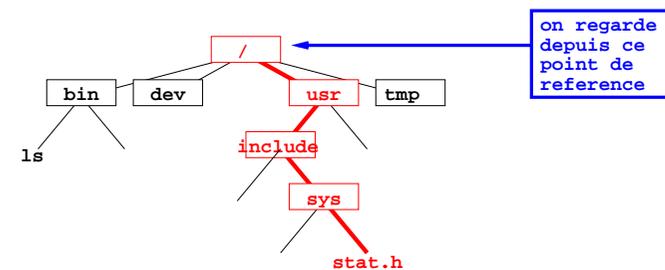
## Sommaire

- Notions d'objets Unix
- Notions de chemins
- Manipulation de base des objets
  - cd, pwd, ls, mkdir, mv, rm
- Liens hard, liens symboliques
- Commandes de base
  - cat, more, less, wc, diff, head, tail, touch, file

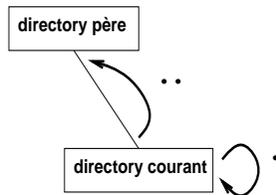
- On accède à un fichier à travers son « chemin » dans l'arborescence
  - le chemin est constitué d'une liste de noms de répertoires et se termine par le nom du fichier
  - le caractère « / » permet de séparer les différents répertoires qui constituent le chemin

`/repertoire1/répertoire2/fichier`

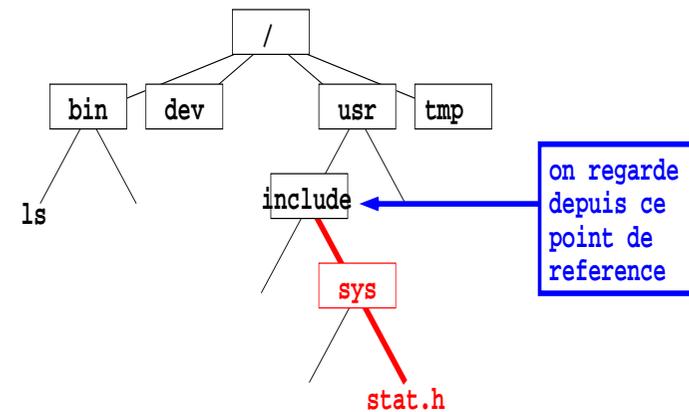
- Chemin d'accès « **absolu** »
  - C'est le chemin qui permet d'accéder à un fichier et **qui commence par la racine de l'arbre**
  - Un chemin absolu doit toujours commencer par « / »**
  - exemple : « `/usr/include/sys/stat.h` »



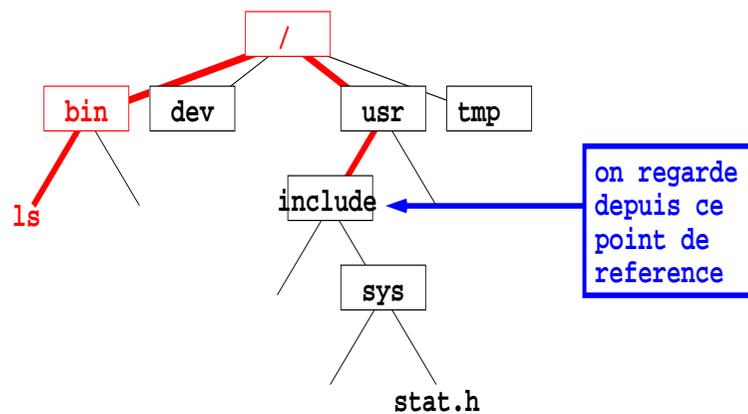
- Chemin d'accès « **relatif** »
  - ⇒ C'est le chemin qui permet d'accéder à un fichier et **qui peut commencer à n'importe quel endroit de l'arbre excepté la racine**
  - ⇒ **Un chemin relatif est « relatif » à la position de référence (le répertoire courant)**
  - ⇒ le répertoire courant est noté « . »
  - ⇒ le répertoire parent du répertoire courant est noté « .. »



- exemple 1 : depuis « /usr/include/ », le chemin relatif du fichier « stat.h » est « sys/stat.h »



- exemple 2 : depuis « /usr/include/ », le chemin relatif du fichier « ls » est « ../../bin/ls »



- Importance des écritures « . » et « .. »
  - commande « find » pour lancer une recherche à partir de l'endroit courant
    - \$ find . -name fichier -print
  - pour lancer une commande qui se trouve dans le répertoire courant
    - \$ ./macommande
- Comparaison Windows/Unix
  - Windows : plusieurs volumes « C:, D: »; « \ » comme séparateur de répertoires
  - Unix : une arbre unique; « / » comme séparateur de répertoires

## Sommaire

- Notions d'objets Unix
- Notions de chemins
- Manipulation de base des objets
  - cd, pwd, ls, mkdir, mv, rm
- Liens hard, liens symboliques
- Commandes de base
  - cat, more, less, wc, diff, head, tail, touch, file

- Commande « cd » (change directory)
  - syntaxe : cd répertoire
  - \$ cd /etc
  - \$ cd ../home/albert
- ⇒ lorsque le répertoire n'existe pas, le shell vous affiche une erreur

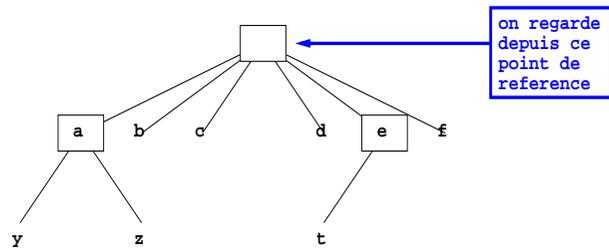
```
[aoi@test]$ cd vide  
bash: cd: vide: Aucun fichier ou répertoire de ce type
```

- Commande « `pwd` » (present working directory)
  - ➔ syntaxe : `pwd`
  - ➔ `$ cd /etc`  
`$ pwd`  
`/etc`
  - ➔ `$ cd /home/albert`  
`$ pwd`  
`/home/albert`

- Commande « `ls` » (list)
  - ➔ syntaxe : `cd [options] objets`
  - ➔ principales options (cumulables)
    - « `-l` » : affiche au format détaillé les informations
    - « `-R` » : liste récursive du contenu des sous-répertoires
    - « `-d` » : affiche les répertoires comme les fichiers sans afficher leurs contenus
    - « `-F` » : affiche les objets avec un suffixe désignant le type de l'objet
    - « `-a` » : affichage des fichiers « cachés » (dont le nom commence par « `.` »)
    - « `-i` » : affiche le numéro d'inode correspondant
    - « `-t` » : affiche les fichiers triés par date
    - « `-S` » : afficher les fichiers triés par taille

- Commande « ls »

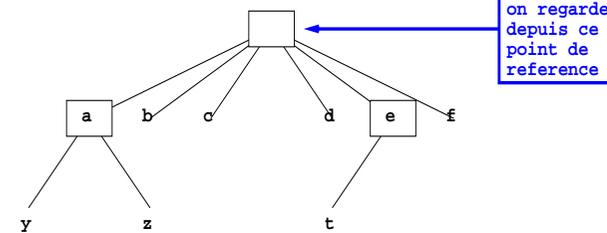
➔ renvoie la liste des objets



```
[aoi@test]$ ls
a/ b c d e/ f
```

- Commande « ls -l »

➔ renvoie la liste des objets avec informations

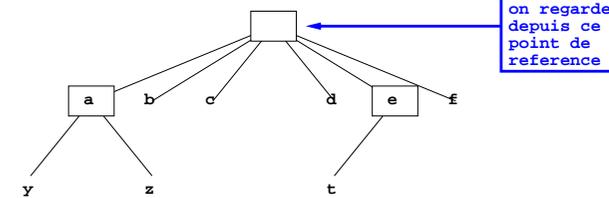


```
[aoi@test]$ ls -l
total 2
drwxr-xr-x 2 franck franck 1024 mar 13 15:59 a/
-rw-r--r-- 1 franck franck 0 mar 13 15:58 b
-rw-r--r-- 1 franck franck 0 mar 13 15:58 c
-rw-r--r-- 1 franck franck 0 mar 13 15:58 d
drwxr-xr-x 2 franck franck 1024 mar 13 15:59 e/
-rw-r--r-- 1 franck franck 0 mar 13 15:58 f
```

- Attention : affichage différent selon que les fichiers sont vieux de plus de 6 mois
  - ➔ affichage de l'année mais pas de l'heure

```
[aoi@test]$ ls -l
total 2
drwxr-xr-x 2 franck franck 1024 mar 13 15:59 a/
-rw-r--r-- 1 franck franck  0 mar 13 15:58 b
-rw-r--r-- 1 franck franck  0 fév 12 2006 c
-rw-r--r-- 1 franck franck  0 mar 13 15:58 d
drwxr-xr-x 2 franck franck 1024 mar 13 15:59 e/
-rw-r--r-- 1 franck franck  0 fév 12 10:56 f
```

- Commande « ls -R »
  - ➔ renvoie le contenu des sous-arborescences



```
[aoi@test]$ ls -R
.:
a/ b c d e/ f

./a:
y z

./e:
t
```

▪ Commande « ls -lR »

- combinaison des option -l et -R

```
[aoi@test]$ ls -lR
.:
total 2
drwxr-xr-x 2 franck franck 1024 mar 13 15:59 a/
-rw-r--r-- 1 franck franck  0 mar 13 15:58 b
-rw-r--r-- 1 franck franck  0 mar 13 15:58 c
-rw-r--r-- 1 franck franck  0 mar 13 15:58 d
drwxr-xr-x 2 franck franck 1024 mar 13 15:59 e/
-rw-r--r-- 1 franck franck  0 mar 13 15:58 f

./a:
total 0
-rw-r--r-- 1 franck franck 0 mar 13 15:59 y
-rw-r--r-- 1 franck franck 0 mar 13 15:59 z

./e:
total 0
-rw-r--r-- 1 franck franck 0 mar 13 15:59 t
```

▪ Commande « ls -F »

- associe un suffixe désignant l'objet
- « / » : désigne un répertoire
- « \* » : désigne un exécutable
- « @ » : désigne un lien symbolique

```
[aoi@test]$ ls -F
a/ b c d e/ f
```

- Les émulateurs de terminal récents associent des couleurs aux différents type de fichiers

- Commande « ls -a »

- ➔ affiche tous les fichiers (également ceux qui commencent par un point – fichiers « cachés »)

```
[aoi@test]$ ls -la
total 6
drwxr-xr-x  4 franck franck 1024 mar 13 16:22 ./
drwxr-xr-x 66 franck franck 3072 mar 13 15:20 ../
drwxr-xr-x  2 franck franck 1024 mar 13 15:59 a/
-rw-r--r--  1 franck franck   0 mar 13 15:58 b
-rw-r--r--  1 franck franck   0 fév 12  2006 c
-rw-r--r--  1 franck franck   0 mar 13 16:22 .config
-rw-r--r--  1 franck franck   0 mar 13 15:58 d
drwxr-xr-x  2 franck franck 1024 mar 13 15:59 e/
-rw-r--r--  1 franck franck   0 fév 12 10:56 f
```

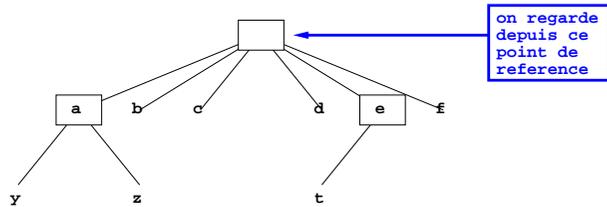
- Commande « ls -t »

```
[aoi@test]$ ls -lt
total 6
-rw-r--r--  1 franck franck  111 mar 13 19:34 c
-rw-r--r--  1 franck franck   40 mar 13 19:26 d
-rw-r--r--  1 franck franck 1826 mar 13 19:26 b
drwxr-xr-x  2 franck franck 1024 mar 13 15:59 e/
drwxr-xr-x  2 franck franck 1024 mar 13 15:59 a/
-rw-r--r--  1 franck franck   0 fév 12 10:56 f
```

- Commande « ls -S »

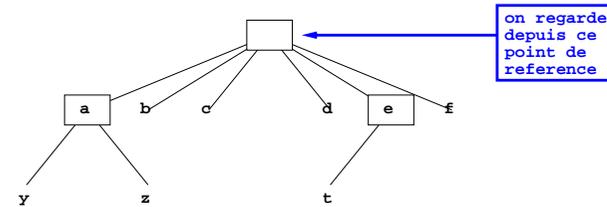
```
[aoi@test]$ ls -lS
total 6
-rw-r--r--  1 franck franck 1826 mar 13 19:26 b
drwxr-xr-x  2 franck franck 1024 mar 13 15:59 a/
drwxr-xr-x  2 franck franck 1024 mar 13 15:59 e/
-rw-r--r--  1 franck franck  111 mar 13 19:34 c
-rw-r--r--  1 franck franck   40 mar 13 19:26 d
-rw-r--r--  1 franck franck   0 fév 12 10:56 f
```

- Différence entre le contenu et le contenant
  - affichage du contenu



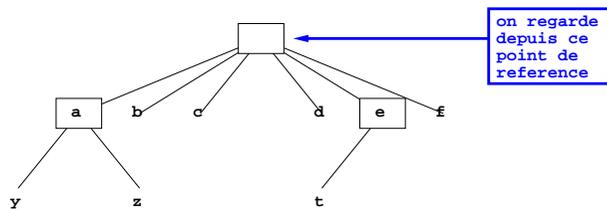
```
[aoi@test]$ ls a
y z
[aoi@test]$ ls -l a
total 0
-rw-r--r-- 1 franck franck 0 mar 13 15:59 y
-rw-r--r-- 1 franck franck 0 mar 13 15:59 z
```

- Différence entre le contenu et le contenant
  - affichage du contenant : utiliser l'option « -d »



```
[aoi@test]$ ls -d a
a/
[aoi@test]$ ls -ld a
drwxr-xr-x 2 franck franck 1024 mar 13 15:59 a/
```

- Différence entre le contenu et le contenant
  - affichage du contenu : « ls » sans option renvoie la liste des fichiers contenus dans le répertoire courant



```
[aoi@test]$ ls
a/ b c d e/ f
[aoi@test]$ ls .
a/ b c d e/ f
```

- Commande « mkdir » (make directory)
  - syntaxe : mkdir [options] répertoires

```
[aoi@test]$ mkdir z
[aoi@test]$ ls -ld z
drwxr-xr-x 2 franck franck 1024 mar 13 21:02 z/
```

- création de répertoires emboîtés

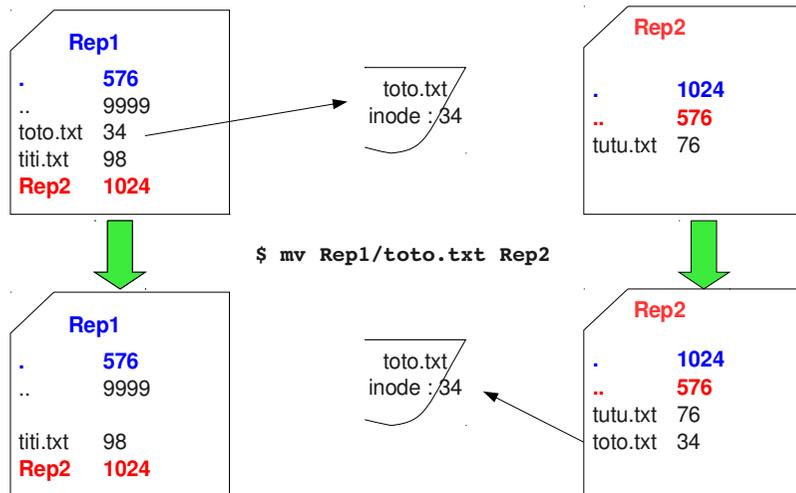
```
[aoi@test]$ mkdir z/w
[aoi@test]$ mkdir z/w/y
```

- option « -p » : pour créer une sous-arborescence

```
[aoi@test]$ mkdir -p rep1/rep2/rep3
```

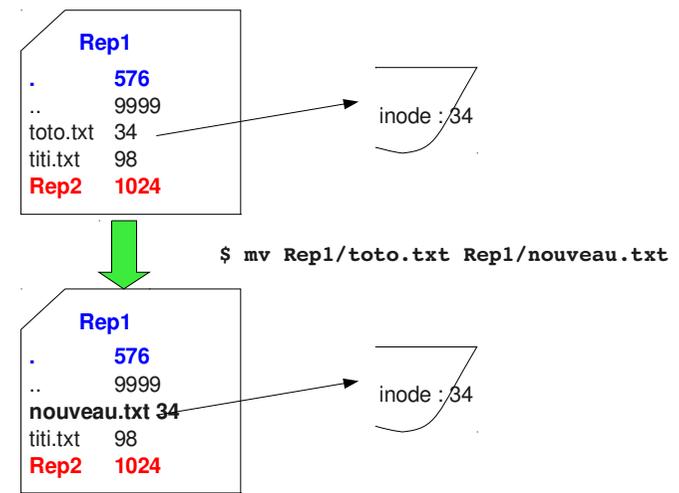
Déplacer et renommer des objets : mv

- Déplacer un objet : le rattacher à un autre répertoire de l'arborescence



Déplacer et renommer des objets : mv

- Renommer un objet : modifier la valeur du nom associé à l'inode



- Commande « mv » (move)
  - ➔ syntaxe : `mv [options] départ destination`
  - ➔ deux utilisations
    - changer le rattachement d'un objet
    - changer le nom du rattachement
  - ➔ l'option « -i » demande la confirmation dans le cas d'un écrasement

Source	Destination	Signification
Objet	Objet différent d'un répertoire	Renomme l'objet de départ en nom de destination
Objet(s)	Répertoire	Déplace les objets vers le répertoire de destination en conservant les noms d'origine

- Renommage

```
[aoi@test]$ ls -l
total 9
-rw-r--r-- 1 franck franck 40 mar 13 19:26 d
drwxr-xr-x 2 franck franck 1024 mar 13 15:59 e/
-rw-r--r-- 1 franck franck 0 fév 12 10:56 f
[aoi@test]$ mv f nouveau
[aoi@test]$ ls -l
total 9
-rw-r--r-- 1 franck franck 40 mar 13 19:26 d
drwxr-xr-x 2 franck franck 1024 mar 13 15:59 e/
-rw-r--r-- 1 franck franck 0 fév 12 10:56 nouveau
```

- Déplacement

```
[aoi@test]$ mv nouveau e/
[aoi@test]$ ls -l e/
total 0
-rw-r--r-- 1 franck franck 0 fév 12 10:56 nouveau
-rw-r--r-- 1 franck franck 0 mar 13 15:59 t
```

- Demande de confirmation avant écrasement

```
[aoi@test]$ ls -l
total 0
-rw-r--r-- 1 franck franck 0 mar 13 15:59 y
-rw-r--r-- 1 franck franck 0 mar 13 15:59 z
[aoi@test]$ mv -i y z
mv: écraser `z'?y
[aoi@test]$ ls -l
total 0
-rw-r--r-- 1 franck franck 0 mar 13 15:59 z
```

- Commande « cp » (copy)
  - ➔ cp [options] départ destination
  - ➔ deux utilisations
    - dupliquer en changeant le nom de destination (cas où la destination n'est pas un répertoire)
    - dupliquer vers un répertoire destination en conservant les noms
  - ➔ l'option « -i » confirmation en cas d'écrasement
  - ➔ l'option « -r » copie récursive
  - ➔ l'option « -p » copie avec conservation des dates

## ▪ Copie de fichier

```
[aoi@test]$ ls
fichier.txt
[aoi@test]$ cp fichier.txt fichier.txt.sav
[aoi@test]$ ls
fichier.txt  fichier.txt.sav
```

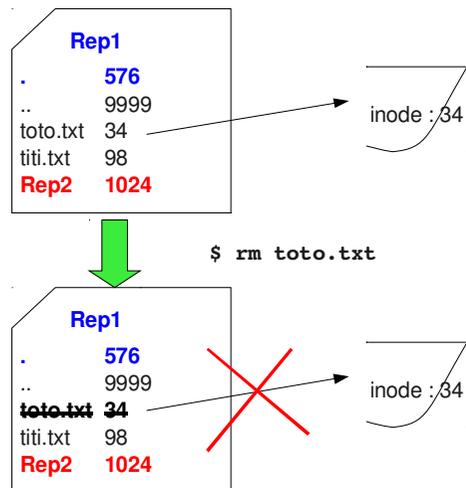
## ▪ Copie de fichier dans un répertoire

```
[aoi@test]$ ls
fichier.txt  sauve/
[aoi@test]$ cp fichier.txt sauve/
[aoi@test]$ ls
fichier.txt  sauve/
[aoi@test]$ ls sauve/
fichier.txt
```

## ▪ Copie récursive

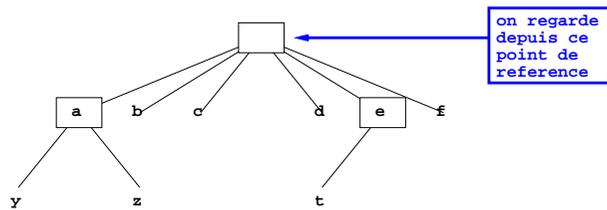
```
[aoi@test]$ ls -l
total 2
drwxr-xr-x 2 franck franck 1024 mar 16 22:25 dest/
drwxr-xr-x 2 franck franck 1024 mar 15 17:30 sauve/
[aoi@test]$ cp -r sauve/ dest/
[aoi@test]$ ls -R dest/
dest/:
sauve/

dest/sauve:
fichier.txt
```



- Commande « rm » (remove)
  - ➔ rm [options] objets
  - ➔ l'option « -i » confirmation de suppression
  - ➔ l'option « -r » suppression récursive
  - ➔ l'option « -f » suppression en force d'un objet même si ses droist ne s'y prêtent pas
  - ➔ rm -rf répertoires
    - supprime récursivement fichiers et répertoires sans demande de confirmation : pratique mais dangereux.

Destruction d'objet : rm

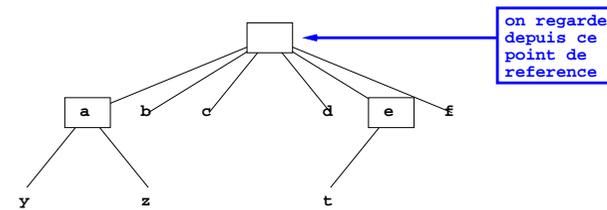


```
[aoi@test]$ ls -F
a/ b c d e/ f

[aoi@test]$ rm f

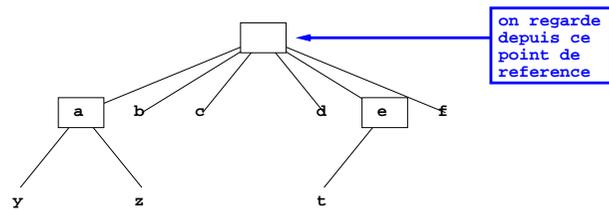
[aoi@test]$ ls -F
a/ b c d e/
```

Destruction d'objet : rm



```
[aoi@test]$ rm a
rm: ne peut détruire le répertoire `a': est un répertoire
[aoi@test]$ ls a
y z
```

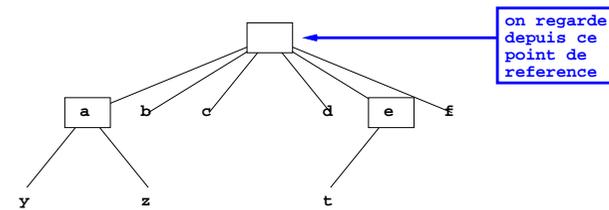
Destruction d'objet : rm



```
[aoi@test]$ rm -rf a
```

```
[aoi@test]$ ls -F
b c d e/
```

Destruction d'objet : rm



```
[aoi@test]$ rm -i b
```

```
rm: détruire fichier régulier `b'? y
```

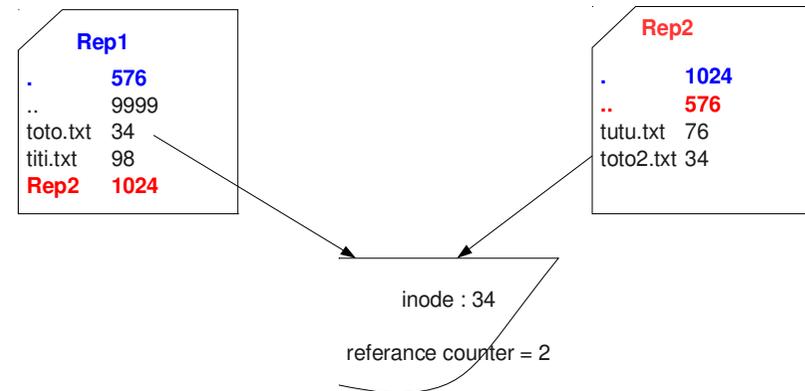
```
[aoi@test]$ ls -F
c d e/
```

## Sommaire

- Notions d'objets Unix
- Notions de chemins
- Manipulation de base des objets
  - cd, pwd, ls, mkdir, mv, rm
- Liens hard, liens symboliques
- Commandes de base
  - cat, more, less, wc, diff, head, tail, touch, file

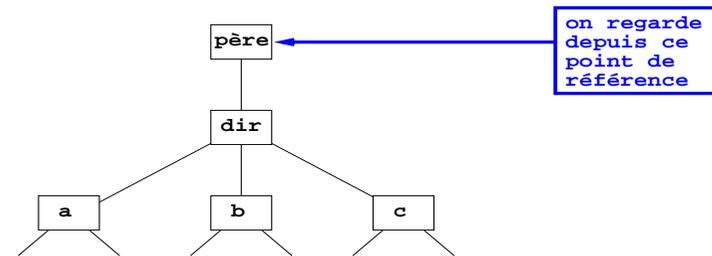
## Liens sur objets

- Rappel : les noms sont stockés dans les répertoires
- un nom est appelé *un lien* sur l'objet
- Sur Unix, chaque objet peuvent être associés plusieurs noms



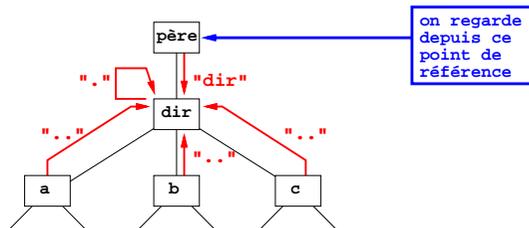
- Dans l'inode d'un objet, il y a un compteur de liens :
  - ➔ compteur incrémenté lors de la création d'un nouveau lien
  - ➔ compteur décrétementé lors de la suppression d'un lien
  - ➔ l'objet est détruit lorsque le dernier lien sur l'objet est supprimé
  - ➔ « . » et « .. » sont des liens

```
[aoi@test]$ ls -l
total 5
-rw-r--r-- 3 franck franck 111 mar 13 19:34 c
-rw-r--r-- 3 franck franck 111 mar 13 19:34 c-1
-rw-r--r-- 3 franck franck 111 mar 13 19:34 c-2
-rw-r--r-- 1 franck franck 40 mar 13 19:26 d
drwxr-xr-x 2 franck franck 1024 mar 15 15:56 e/
```



```
[aoi@test]$ ls -l dir
total 3
drwxr-xr-x 2 franck franck 1024 mar 16 23:23 a/
drwxr-xr-x 2 franck franck 1024 mar 16 23:23 b/
drwxr-xr-x 2 franck franck 1024 mar 16 23:23 c/
[aoi@test]$ ls -ld dir
drwxr-xr-x 5 franck franck 1024 mar 16 23:23 dir/
```

- Il y a en effet 5 liens sur l'objet « dir » :



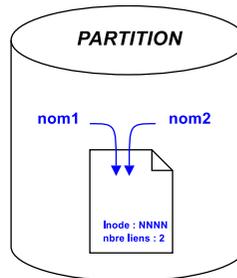
- lien « /chemin/dir »
- lien « /chemin/dir/. »
- lien « /chemin/dir/a/.. »
- lien « /chemin/dir/b/.. »
- lien « /chemin/dir/c/.. »

- la commande « `ls -ldi` » permet de vérifier que l'inode correspondant à chaque nom relève bien du même fichier

```
[aoi@test]$ ls -ldi dir dir/. dir/a/.. dir/b/.. dir/c/..
2165772 drwxr-xr-x 5 franck franck 1024 mar 16 23:23 dir/
2165772 drwxr-xr-x 5 franck franck 1024 mar 16 23:23 dir/./
2165772 drwxr-xr-x 5 franck franck 1024 mar 16 23:23 dir/a/..
2165772 drwxr-xr-x 5 franck franck 1024 mar 16 23:23 dir/b/..
2165772 drwxr-xr-x 5 franck franck 1024 mar 16 23:23 dir/c/..
```

- Il existe deux types de liens
  - lien **hard**
  - lien **symbolique**

- Le lien **hard** utilise le numéro d'inode pour identifier l'objet
  - un numéro unique par partition
  - confiné à l'intérieur d'une même partition (unicité de l'inode)
- Contraintes
  - hard link impossible vers une autre partition (risque de perte de l'unicité de l'inode)
  - hard link impossible vers un répertoire (risque de boucles invisibles dans l'arborescence)
- Utilisation courante : liens hard dans un environnement « chrooté »



- Commande « ln » (link)
  - ln original synonyme

```
[aoi@test]$ ls -l fichier1
-rw-r--r-- 1 franck franck 0 mar 17 08:31 fichier1

[aoi@test]$ ln fichier1 fichier2

[aoi@test]$ ls -l fichier1 fichier2
-rw-r--r-- 2 franck franck 0 mar 17 08:31 fichier1
-rw-r--r-- 2 franck franck 0 mar 17 08:31 fichier2

[aoi@test]$ ls -li fichier1 fichier2
1666053 -rw-r--r-- 2 franck franck 0 mar 17 08:31
fichier1
1666053 -rw-r--r-- 2 franck franck 0 mar 17 08:31
fichier2
```

- Copies et déplacement vers un lien

```
[aoi@test]$ cp c fichier1
cp: écraser `fichier1'?y
```

```
[aoi@test]$ ls -li fichier1 fichier2
1666053 -rw-r--r-- 2 franck franck 111 mar 17 08:34 fichier1
1666053 -rw-r--r-- 2 franck franck 111 mar 17 08:34 fichier2
```

```
[aoi@test]$ ls -li fichier3
1666055 -rw-r--r-- 1 franck franck 136 mar 17 08:43 fichier3
```

```
[aoi@test]$ mv fichier3 fichier2
mv: écraser `fichier2'?y
```

```
[aoi@test]$ ls -li fichier1 fichier2
1666053 -rw-r--r-- 1 franck franck 111 mar 17 08:34 fichier1
1666055 -rw-r--r-- 1 franck franck 136 mar 17 08:43 fichier2
```

- Suppression d'un lien par rm

```
[aoi@test]$ ls -li fichier1 fichier2
1666053 -rw-r--r-- 2 franck franck 111 mar 17 08:34 fichier1
1666053 -rw-r--r-- 2 franck franck 111 mar 17 08:34 fichier2
```

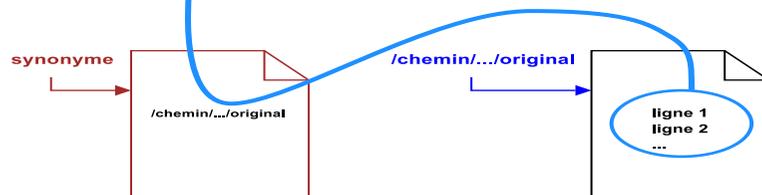
```
[aoi@test]$ rm fichier1
```

```
[aoi@test]$ ls -li fichier1 fichier2
ls: fichier1: Aucun fichier ou répertoire de ce type
1666053 -rw-r--r-- 1 franck franck 111 mar 17 08:34 fichier2
```

- Place occupée (les liens ne prennent pas de place)

```
[aoi@test]$ ls -la
total 4
drwxr-xr-x 2 franck franck 1024 mar 17 08:56 ./
drwxr-xr-x 5 franck franck 1024 mar 17 08:56 ../
-rw-r--r-- 2 franck franck 111 mar 17 08:34 fichier1
-rw-r--r-- 2 franck franck 111 mar 17 08:34 fichier2
[aoi@test]$ du -b
1135 .
[aoi@test]$ rm fichier2
[aoi@test]$ du -b
1135 .
```

- Le lien **symbolique** est un fichier spécial contenant le chemin d'accès à un autre objet
  - un numéro unique par partition
- Utilisation courante :
  - fichier commun à plusieurs utilisateurs sur une partition différente
  - liens sur des répertoires



- Commande « ln » (link)
  - ln -s original synonyme

```
[aoi@test]$ ls -l
total 1
-rw-r--r-- 1 franck franck 294 mar 18 22:43 fichier1

[aoi@test]$ ln -s fichier1 fichier2

[aoi@test]$ ls -li fichier1 fichier2
1674243 -rw-r--r-- 1 franck franck 294 mar 18 22:43 fichier1
1674242 lrwxrwxrwx 1 franck franck 8 mar 18 22:45 fichier2 ->
fichier1

[aoi@test]$ ls -lL fichier1 fichier2
-rw-r--r-- 1 franck franck 294 mar 18 22:43 fichier1
-rw-r--r-- 1 franck franck 294 mar 18 22:43 fichier2
```

- Suppression d'un lien symbolique « rm »

```
[aoi@test]$ ls -li fichier1 fichier2
1674243 -rw-r--r-- 1 franck franck 294 mar 18 22:43 fichier1
1674242 lrwxrwxrwx 1 franck franck 8 mar 18 22:45 fichier2 ->
fichier1
```

```
[aoi@test]$ rm fichier1
rm: détruire fichier régulier `fichier1'? y
```

```
[aoi@test]$ ls -li fichier2
1674242 lrwxrwxrwx 1 franck franck 8 mar 18 22:45 fichier2 ->
fichier1
```

```
[aoi@test]$ ls -liL fichier2
ls: fichier2: Aucun fichier ou répertoire de ce type
[aoi@test]$ cat fichier2
cat: fichier2: Aucun fichier ou répertoire de ce type
```

- Les systèmes Unix imposent les droits « lrwxr-xr-x » ou « lrwxrwxrwx » sur le lien
  - ➔ ils ne peuvent être modifiés
  - ➔ on peut seulement modifier les droits du fichier pointé par un lien symbolique

```
[aoi@test]$ ls -l fichier1 fichier2
-rw-r--r-- 1 franck franck 307 mar 18 23:03 fichier1
lrwxrwxrwx 1 franck franck 8 mar 18 22:45 fichier2 ->
fichier1
```

```
[aoi@test]$ chmod 600 fichier2
```

```
[aoi@test]$ ls -l fichier1 fichier2
-rw----- 1 franck franck 307 mar 18 23:03 fichier1
lrwxrwxrwx 1 franck franck 8 mar 18 22:45 fichier2 ->
fichier1
```

## Sommaire

- Notions d'objets Unix
- Notions de chemins
- Manipulation de base des objets
  - cd, pwd, ls, mkdir, mv, rm
- Liens hard, liens symboliques
- Commandes de base
  - cat, more, less, wc, diff, head, tail, touch, file

- Commande « cat » (concatenate)

→ `cat fichiers`

```
[aoi@test]$ cat /etc/modules
# /etc/modules: kernel modules to load at boot time.
#
# This file should contain the names of kernel modules that
are
# to be loaded at boot time, one per line. Comments begin
with
# a `#', and everything on the line after them are ignored.

nvram
hw_random
evdev
[aoi@test]$
```

- Commande « more »
  - ➔ more fichiers
  - ➔ affichage du contenu d'un fichier page par page
  - ➔ Syntaxe :
    - espace ou « f » pour avancer d'une page
    - « b » pour reculer d'une page
    - « q » pour quitter
  - ➔ le commande « man » affiche les pages de manuel à l'aide de la commande « more »

- Commande « less »
  - ➔ less fichiers
  - ➔ même principe que more avec certains avantages
    - déplacement ligne par ligne à l'aide des touches de déplacement
    - ne nécessite pas la lecture de la totalité du fichier pour commencer : intérêt pour les fichiers longs
  - ➔ commandes basées à la fois sur more et sur vi

## ▪ Commande « wc » (word count)

➔ `wc [options] fichiers`

➔ options :

- « `-c` » compte le nombre de caractères uniquement
- « `-l` » compte le nombre de lignes uniquement
- « `-w` » compte le nombre de mots uniquement

```
[aoi@test]$ wc /etc/modules
9 50 266 /etc/modules
[aoi@test]$ wc -l /etc/modules
9 /etc/modules
```

## ▪ Commande « diff » (difference)

➔ `diff [options] fichier1 fichier2`

➔ comparaison (fichiers texte) ligne à ligne de fichier2 par rapport à fichier1

➔ options :

- « `-c` » affiche quelques lignes du contexte
- « `-u` » mode unifié

➔ commande très utile en administration : isoler rapidement les différences entre différentes versions de fichier de configuration

- Modification d'une ligne

```
[aoi@test]$ cat fichier1
option1 = texte1
option2 = texte2
option3 = texte3
```

```
[aoi@test]$ cat fichier2
option1 = texte1
option2 = modification
option3 = texte3
```

```
[aoi@test]$ diff fichier1 fichier2
2c2
< option2 = texte2
---
> option2 = modification
```

- Traduction

⇒ « 2c2 » : la ligne 2 de fichier2 est changée par rapport à la ligne 2 de fichier1

- Décalage et ajout de lignes

```
[aoi@test]$ cat fichier1
option1 = texte1
option2 = texte2
option3 = texte3
```

```
[aoi@test]$ cat fichier2
```

```
option1 = texte1
option2 = modification
option3 = texte3
```

```
option4 = option4
```

```
[aoi@test]$ diff fichier1 fichier2
```

```
0a1
>
2c3
< option2 = texte2
---
> option2 = modification
3a5,6
>
> option4 = option4
```

- « 0a1 » : ligne 1 fichier2 ajoutée à la ligne 0 du fichier1

- « 3a5,6 » : lignes 5 à 6 fichier2 ajoutées à la ligne 3 fichier1

- option « -c »

```
[aoi@test]$ cat fichier1
option1 = texte1
option2 = texte2
option3 = texte3
```

```
[aoi@test]$ cat fichier2
```

```
option1 = texte1
option2 = modification
option3 = texte3
option4 = option4
```

```
[aoi@test]$ diff -c fichier1 fichier2
```

```
*** fichier1 2007-03-19 09:04:37.000000000
+0100
--- fichier2 2007-03-19 09:02:53.000000000
+0100
*****
*** 1,3 ***
! option2 = texte2
option3 = texte3
--- 1,6 ----
+
option1 = texte1
! option2 = modification
option3 = texte3
+
option4 = option4
```

- « ! » : ligne modifiée

- « + » : ligne ajoutée

- option « -u »

```
[aoi@test]$ cat fichier1
option1 = texte1
option2 = texte2
option3 = texte3
```

```
[aoi@test]$ cat fichier2
```

```
option1 = texte1
option2 = modification
option3 = texte3
option4 = option4
```

```
[aoi@test]$ diff -u fichier1 fichier2
```

```
--- fichier1 2007-03-19
09:04:37.000000000 +0100
+++ fichier2 2007-03-19
09:02:53.000000000 +0100
@@ -1,3 +1,6 @@
+
option1 = texte1
-option2 = texte2
+option2 = modification
option3 = texte3
+
+option4 = option4
```

- exemple d'utilisation
  - copie de pages php d'une machine à une autre...les pages php ne fonctionnent plus correctement
  - Peut être un problème de configuration de php mais le fichier php.ini compte 1355 lignes et 7230 mots à comparer

```
[aoi@test]$ diff -u php-v1.ini php-v2.ini
--- php-v1.ini 2007-03-19 08:44:28.000000000 +0100
+++ php-v2.ini 2007-03-19 08:45:29.000000000 +0100
@@ -395,7 +395,7 @@
 ; You should do your best to write your scripts so
 that they do not require
 ; register_globals to be on; Using form variables as
 globals can easily lead
 ; to possible security problems, if the code is not
 very well thought of.
-register_globals = Off
+register_globals = On
```

- Commande « head »
  - head [-nombre] fichiers
  - extrait les « nombre » premières ligne d'un fichier

```
[aoi@test]$ head -5 /etc/php.ini
[PHP]

;;;;;;;;;;
; WARNING ;
;;;;;;;;;;
```

- Commande « tail »
  - `tail [-nombre] fichiers` : affiche les n dernières lignes
  - `tail [+nombre] fichiers` : affiche de la nième ligne à la dernière
  - `tail [-f] fichier` : affichage « en continu » des dernières lignes  
Particulièrement intéressant pour suivre « en live » l'évolution d'un fichier journal.  
Sortie avec Ctrl + C

```
[aoi@test]$ tail -5 /etc/php.ini
;hphp.upload.verification_script = /home/hphp/verify_script

; Local Variables:
; tab-width: 4
; End:
```

- Commande « touch »
  - `touch fichiers`
  - A la base, la commande « touch » est destinée à modifier l'horodatage d'un fichier
  - Si le fichier n'existe pas, la commande touch crée un fichier vide

```
[aoi@test]$ ls -l fichier3
ls: fichier3: Aucun fichier ou répertoire de ce type
[aoi@test]$ touch fichier3
[aoi@test]$ ls -l fichier3
-rw-r--r-- 1 franck franck 0 mar 19 09:58 fichier3
```

```
[aoi@test]$ touch -m -t 200701011200 fichier3
[aoi@test]$ ls -l fichier3
-rw-r--r-- 1 franck franck 0 jan  1 12:00
fichier3
```

- Commande « file »
  - file fichiers
  - Détermine le type d'un fichier ou à quelle application elle correspond
  - utilise un fichier « magic » qui contient les motifs de reconnaissance des fichiers

```
[aoi@test ~]$ file *
toto.tar.bz2:          bzip2 compressed data, block size = 900k
titi.tar.gz:          gzip compressed data
Admin:                directory
certificat.pl2:       data
exemple.txt:          ASCII text
images:               JPEG image data, JFIF
standard 1.01
document.pdf: PDF document, version 1.3
fichierword.doc:     Microsoft Office Document
sign.jpg:             JPEG image data, JFIF
standard 1.00, comment: "LEAD"
```

- Copyrights et crédits
  - Ce cours Unix est fondé sur le cours UNIX ARS de Thierry Besançon publié sous licence OPL (Open Publication Licence). Le document original a été modifié pour l'adapter au public AOI. La version originale est disponible à l'adresse <http://www.formation.jussieu.fr>