



Atelier 1.1
Installation d'un système d'exploitation GNU/Linux

Vientiane, du 21 au 25 juillet 2008

TRAN DINH Minh Tri



Historique du système Unix

- 1969 Tout commence avec UNIX, créé par AT&T et Bell Labs. C'est un système multi-tâches, multi-utilisateurs et écrit en C.
- 1980 Microsoft invente le XENIX, OS basé sur UNIX et qui tourne sur PC.
- 1986-1987 MINIX est un petit OS qui tient sur une disquette de 360Ko!



Le projet GNU

GNU = GNU is Not Unix (« GNU N'est pas Unix »)

- Projet de réaliser un système à la Unix entièrement libre.
- Lancé en 1984 par Richard Stallman, un chercheur du MIT, à une époque où les sources d'Unix n'étaient plus libres d'accès.
- En 1991, le projet GNU n'avait toujours pas de noyau et tournait sur des Unix propriétaires.



Les logiciels libres (free software en anglais)

- La liberté d'exécution : tout le monde a le droit de lancer le programme, quel qu'en soit le but.
- La liberté de modification : tout le monde a le droit d'étudier le programme et de le modifier, ce qui implique un accès au code source.
- La liberté de redistribution : tout le monde a le droit de rediffuser le programme, gratuitement ou non.
- La liberté d'amélioration : tout le monde a le droit de redistribuer une version modifiée du programme.



La licence GPL

GNU General Public License

Les versions modifiées d'un logiciel sous licence GPL ne peuvent être redistribuées que sous cette même licence.



Linux

- En 1991, Linus Torvalds, un étudiant Finlandais, a créé le premier noyau Linux, semblable au noyau Unix.
- Adopte la licence GPL pour Linux.



Les concepts du système Unix

Le modèle en couche

- Noyau
- Bibliothèque C
- Bibliothèques système
- Bibliothèques d'applications
- Programmes utilisateurs



Principales caractéristiques d'Unix

- Multi-utilisateur.
- Multi-tâche.
- Orientation réseau.
- Tout est fichier.
- Gestion des droits par groupes.



Linux et ses distributions

Linux = Linus' UNIX (l'UNIX de Linus)

- **Red Hat: <http://www.redhat.com/>**
- **Suse (Novell): <http://www.suse.com/>**
- **Mandriva: <http://www.mandriva.com>**
- **Debian: <http://debian.org/>**

...



Le choix de Debian

- Stabilité
- Distribution non-commerciale
- Meilleure distribution pour serveurs



Les différentes versions de Debian

- version officielle stable Etch 4.0
- version testing Lenny
- version unstable Sid (Still In Développement)

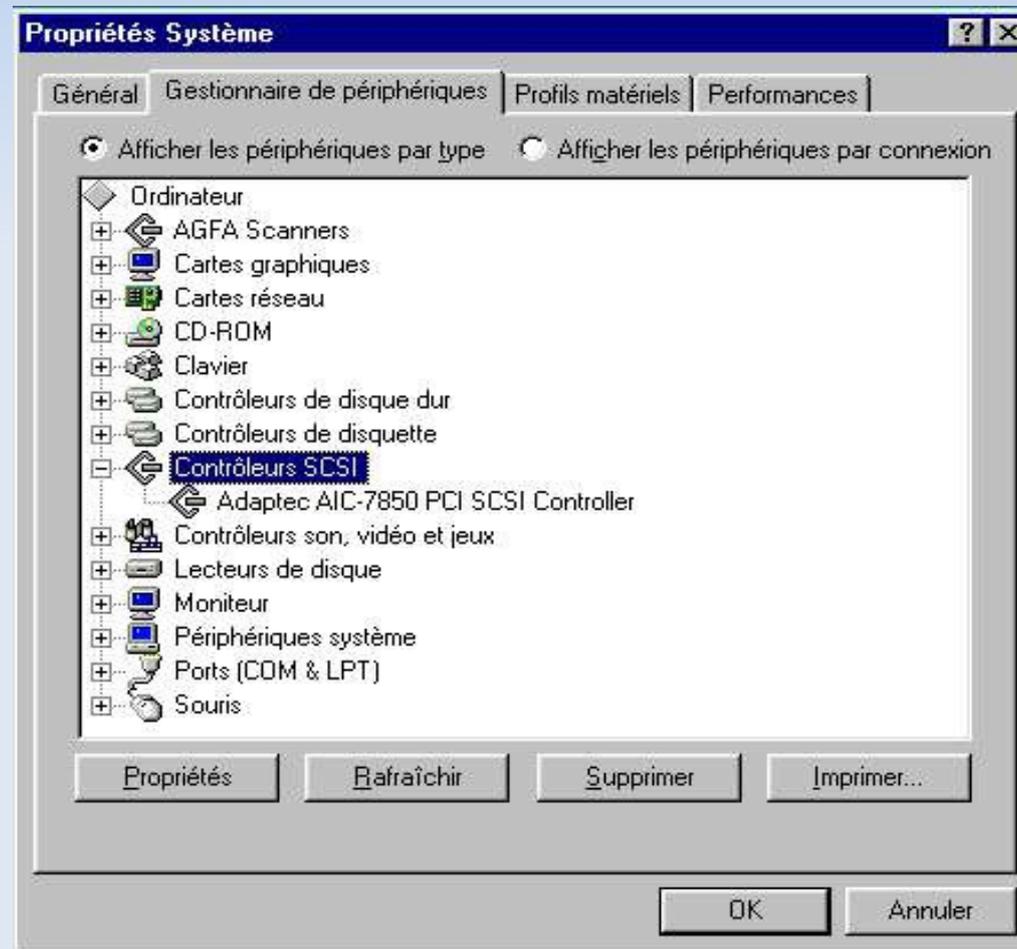


Vérification du matériel

- <http://www.tldp.org/HOWTO/Hardware-HOWTO/>
- Les manuels accompagnant chaque élément.
- Les informations sur la configuration du BIOS de votre ordinateur.
- La fenêtre « Système » dans le panneau de configuration de Windows.



Vérification du matériel (suite)





Préparation du disque dur

- Si vous souhaitez faire cohabiter Windows et Linux sur votre machine, il faut préparer une partition pour l'installation de Linux.
- Les outils de partitionnement: FIPS, Partition Magic, fdisk.



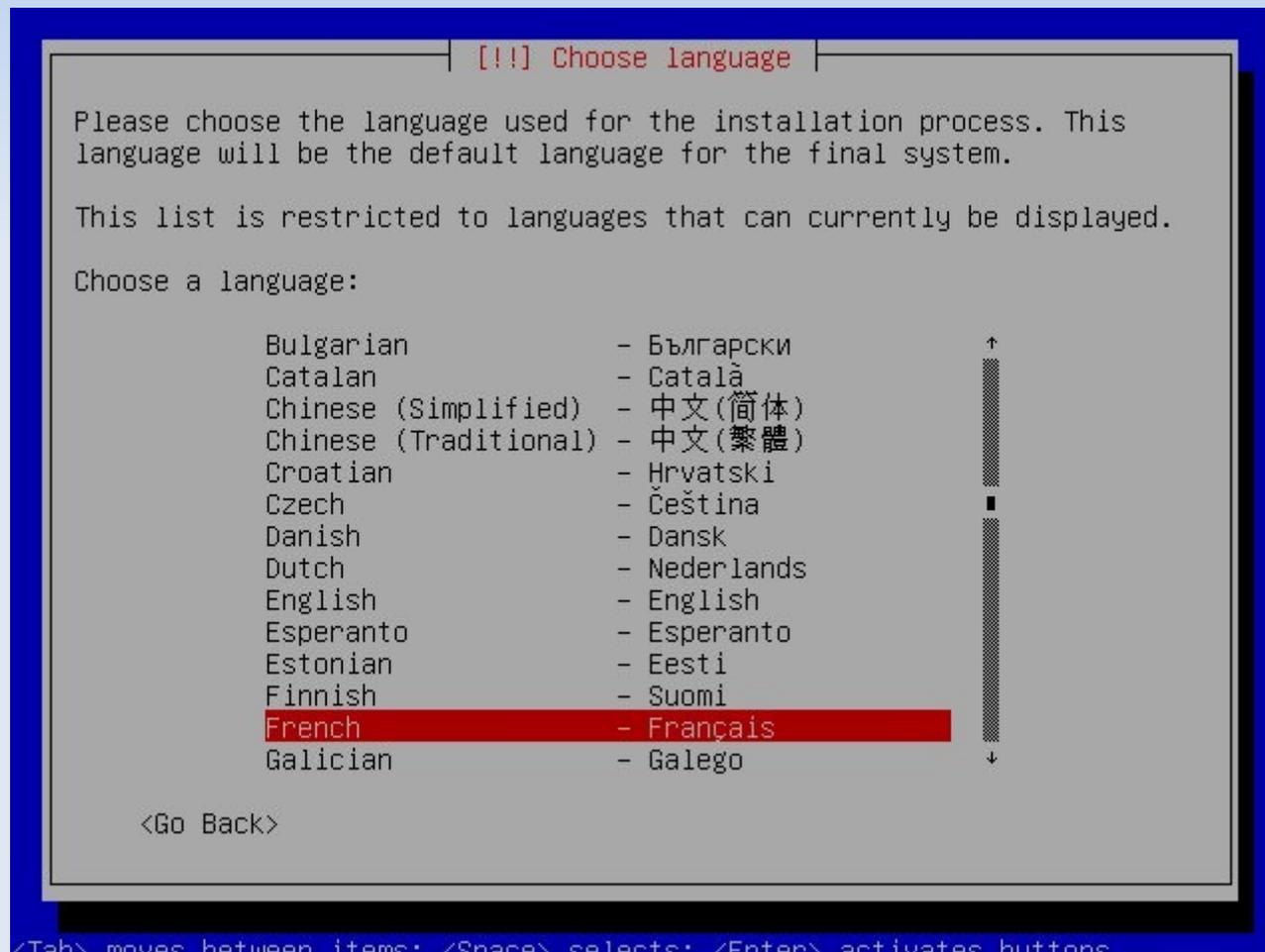
Démarrage de l'installation

Régler la séquence de boot dans le BIOS pour que votre machine démarre sur CD-ROM



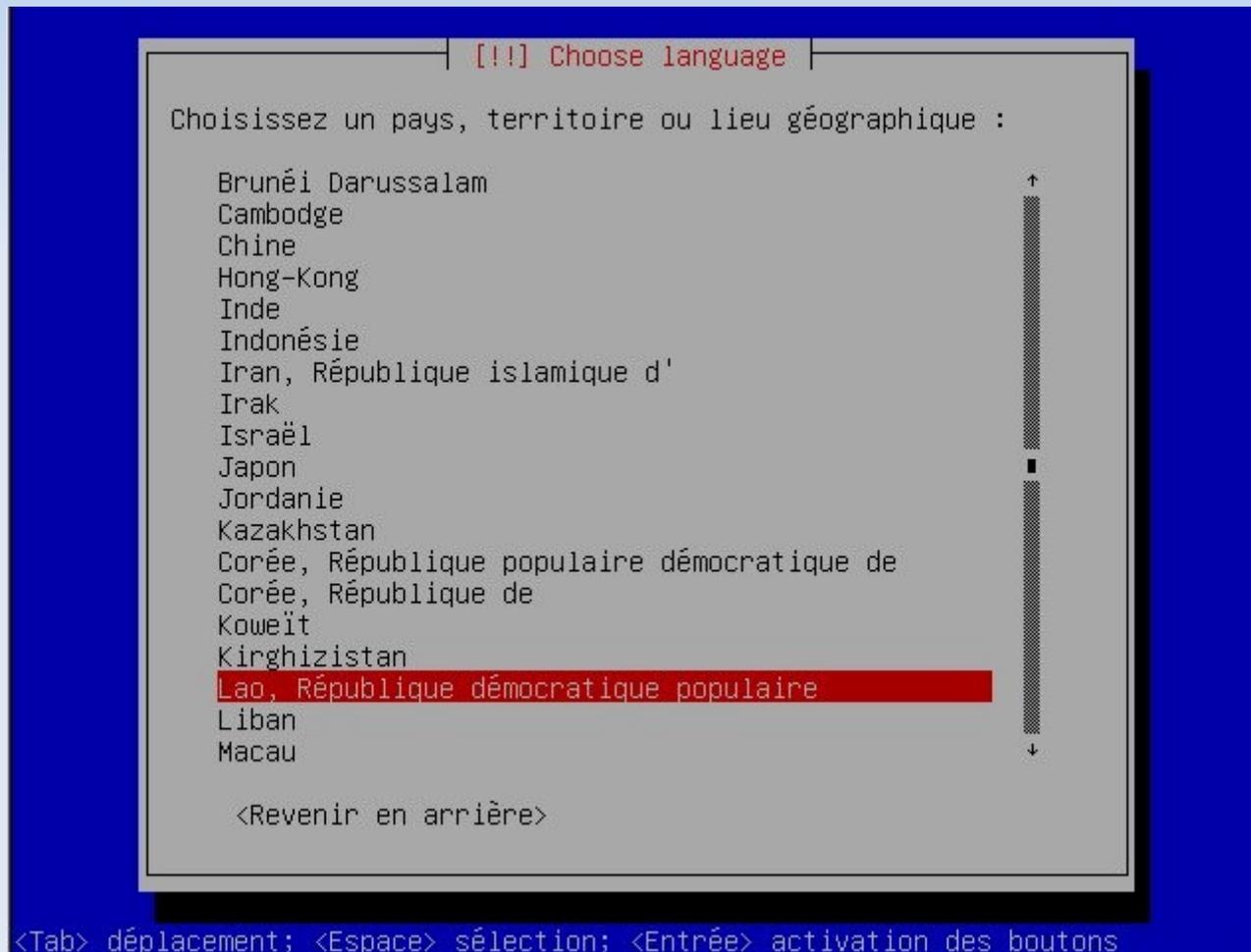


Choix de la langue





Choix du pays





Choix du clavier

```
[!] Choisir la disposition du clavier

Carte de clavier à utiliser :

Allemand (de-latin1-nodeadkeys) ↑
Amérique latine (la-latin1)
Biélorusse (by)
Belge (be2-latin1)
Brésilien (br-abnt2)
Brésilien (br-latin1)
Britannique (uk)
Bulgare (bg)
Canadien français (cf)
Croate (croat)
Danois (dk-latin1)
Dvorak
Espagnol (es)
Estonien (et)
États-Unis (us)
Finnois (fi-latin1)
Français (fr-latin9)
Grec (gr)
Hébreu (hebrew) ↓

<Revenir en arrière>
```



Détection du matériel

Cette étape est automatique et ne nécessite aucune activité de votre part!

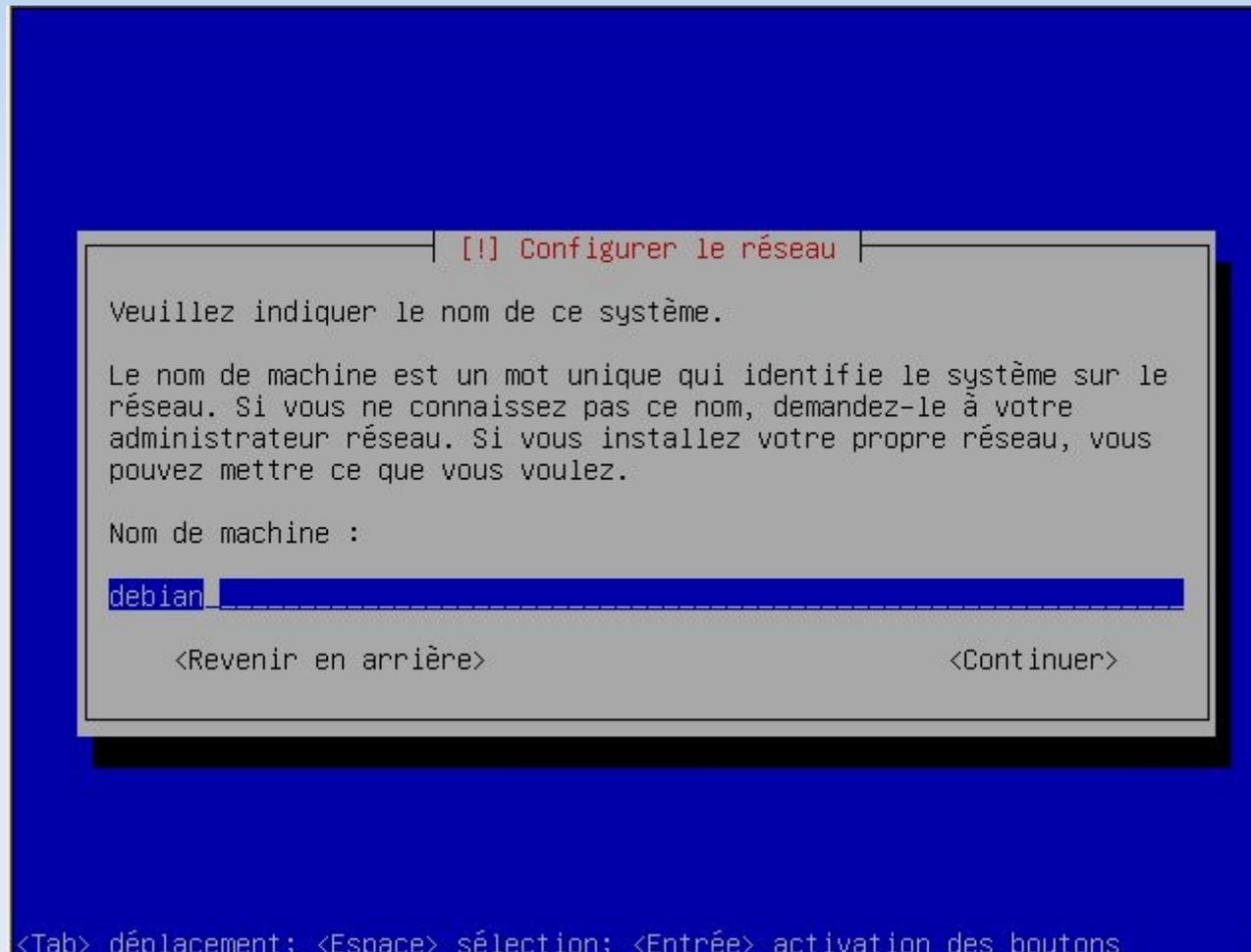


Configuration du réseau





Choix du nom de machine





Partitionner le disque

```
[!!] Partitionner les disques

Le programme d'installation peut vous assister pour le
partitionnement d'un disque (avec plusieurs choix d'organisation).
Vous pouvez également effectuer ce partitionnement vous-même. Si vous
choisissez le partitionnement assisté, vous aurez la possibilité de
vérifier et personnaliser les choix effectués.

Si vous choisissez le partitionnement assisté pour un disque complet,
vous devrez ensuite choisir le disque à partitionner.

Méthode de partitionnement :

    Assisté - utiliser un disque entier
    Assisté - utiliser tout un disque avec LVM
    Assisté - utiliser tout un disque avec LVM chiffré
    Manuel

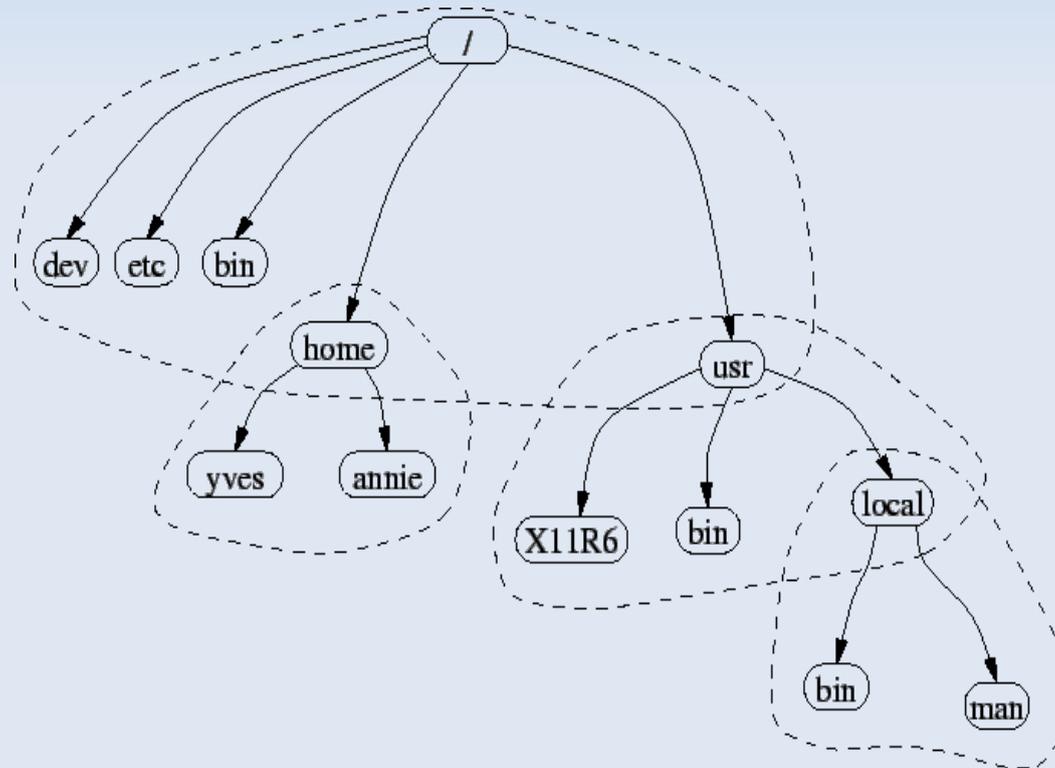
<Revenir en arrière>
```

<Tab> déplacement; <Espace> sélection; <Entrée> activation des boutons



L'arborescence du système de fichiers sous Linux

- Contrairement au système de fichiers Windows, il n'existe pas de lecteurs A:, C:, etc...
- Le système de fichiers sous Linux est organisé par l'arborescence





L'arborescence du système de fichiers sous Linux (suite)

- L'entrée du système de fichier se situe à la racine, notée / .
- Ensuite, il existe un certain nombre de répertoires présents par défaut.
- / Répertoire "racine", point d'entrée du système de fichiers.
- /boot Répertoire contenant le noyau Linux et l'amorceur
- /bin Répertoire contenant les exécutables de base.
- /dev Répertoire contenant des fichiers spéciaux nommés devices qui permettent le lien avec les périphériques de la machine
- /etc Répertoire contenant les fichiers de configuration du système
- /home Répertoire contenant les fichiers personnels des utilisateurs
- /lib Répertoire contenant les bibliothèques et les modules du noyau.
- /media Répertoire contenant les « points de montage » des médias usuels : CD, DVD, disquette, clef USB.



L'arborescence du système de fichiers sous Linux (suite)

- /root Répertoire personnel de l'administrateur.
- /sbin Répertoire contenant les exécutables destinés à l'administration du système.
- /tmp Répertoire contenant des fichiers temporaires utilisés par certains programmes.
- /usr Répertoire contenant les exécutables des programmes, la documentation, et les programmes pour le serveur graphique.
- /var Répertoire contenant les fichiers qui servent à la maintenance du système (les fichiers de logs /var/log).

Pour plus de détails, **man hier**



Les périphériques

Chaque périphérique du système (souris, disque dur, lecteur CD, carte son, etc...) est représenté par un fichier spécial dans /dev/

- /dev/psaux souris PS/2
- /dev/fd0 lecteur de disquettes
- /dev/hda lecteur maître du contrôleur IDE primaire
- /dev/hdb lecteur esclave du contrôleur IDE primaire
- /dev/hdc lecteur maître du contrôleur IDE secondaire
- /dev/hdd lecteur esclave du contrôleur IDE secondaire
- /dev/sda premier disque dur SATA, SCSI ou USB
- /dev/sdb second disque dur SATA, SCSI ou USB
- /dev/ttyS0 premier port série
- /dev/ttyS1 second port série



Les partitions

Sur un disque dur IDE ou SATA, il y a deux types de partition :

- **primaire**
- **étendue**, pouvant contenir une ou plusieurs partitions **logiques**

Sous Linux, le numéro de la partition est ajouté à la suite du nom du disque :

- les partitions **primaires** sont numéros de 1 à 4.
- les partitions **logiques** sont numéros à partir de 5.



Organisation des partitions

Selon l'usage de l'ordinateur, on va organiser les partitions de façon différente, mais il est recommandé de créer au moins les partitions suivantes :

- une partition racine (/) : de type ext3
- une partition de swap : utilisée pour la mémoire virtuelle, de taille 2 x la taille de la mémoire vive.

Si c'est pour un ordinateur personnel, il est recommandé d'avoir une partition séparé pour /home

Si c'est pour un serveur, il faut séparer aussi /boot, /var, /tmp et /usr.



Organisation des partitions (suite)

Exemples:

- Pour un ordinateur personnel:

Swap double de la mémoire vive

/ 7 Go

/home tout le reste de l'espace alloué à Linux

- Pour un serveur:

Swap Egale ou double de la mémoire vive

/ 200 Mo

/tmp 150 Mo

/var 300 Mo

/usr 2 Go

/home selon les besoins des utilisateurs



Appliquer la nouvelle table de partition

```
[!!] Partitionner les disques

Voici la table des partitions et les points de montage actuellement
configurés. Vous pouvez choisir une partition et modifier ses
caractéristiques (système de fichiers, point de montage, etc.), un
espace libre pour créer une nouvelle partition ou un périphérique
pour créer sa table des partitions.

Partitionnement assisté
Aide pour le partitionnement

IDE1 maître (hda) - 2.1 GB VBOX HARDDISK
  n° 1 primaire  1.7 GB  F ext3    /
  n° 2 primaire 444.2 MB F swap   swap

Annuler les modifications des partitions
Terminer le partitionnement et appliquer les changements

<Revenir en arrière>
```

<Tab> déplacement; <Espace> sélection; <Entrée> activation des boutons



Le réglage des comptes et mots de passe

Choisir le mot de passe root

```
[!!] Créer les utilisateurs et choisir les mots de passe

Vous devez choisir un mot de passe pour le superutilisateur, le
compte d'administration du système. Un utilisateur malintentionné ou
peu expérimenté qui aurait accès à ce compte peut provoquer des
désastres. En conséquence, ce mot de passe ne doit pas être facile à
deviner, ni correspondre à un mot d'un dictionnaire ou vous être
facilement associé.

Un bon mot de passe est composé de lettres, chiffres et signes de
ponctuation. Il devra en outre être changé régulièrement.

Par sécurité, rien n'est affiché pendant la saisie.

Mot de passe du superutilisateur (« root ») :
*****
<Revenir en arrière>                                <Continuer>
```



Le réglage des comptes et mots de passe

Créer un compte utilisateur

```
[!!!] Créer les utilisateurs et choisir les mots de passe

Un compte d'utilisateur va être créé afin que vous puissiez disposer
d'un compte différent de celui du superutilisateur (« root »), pour
l'utilisation courante du système.

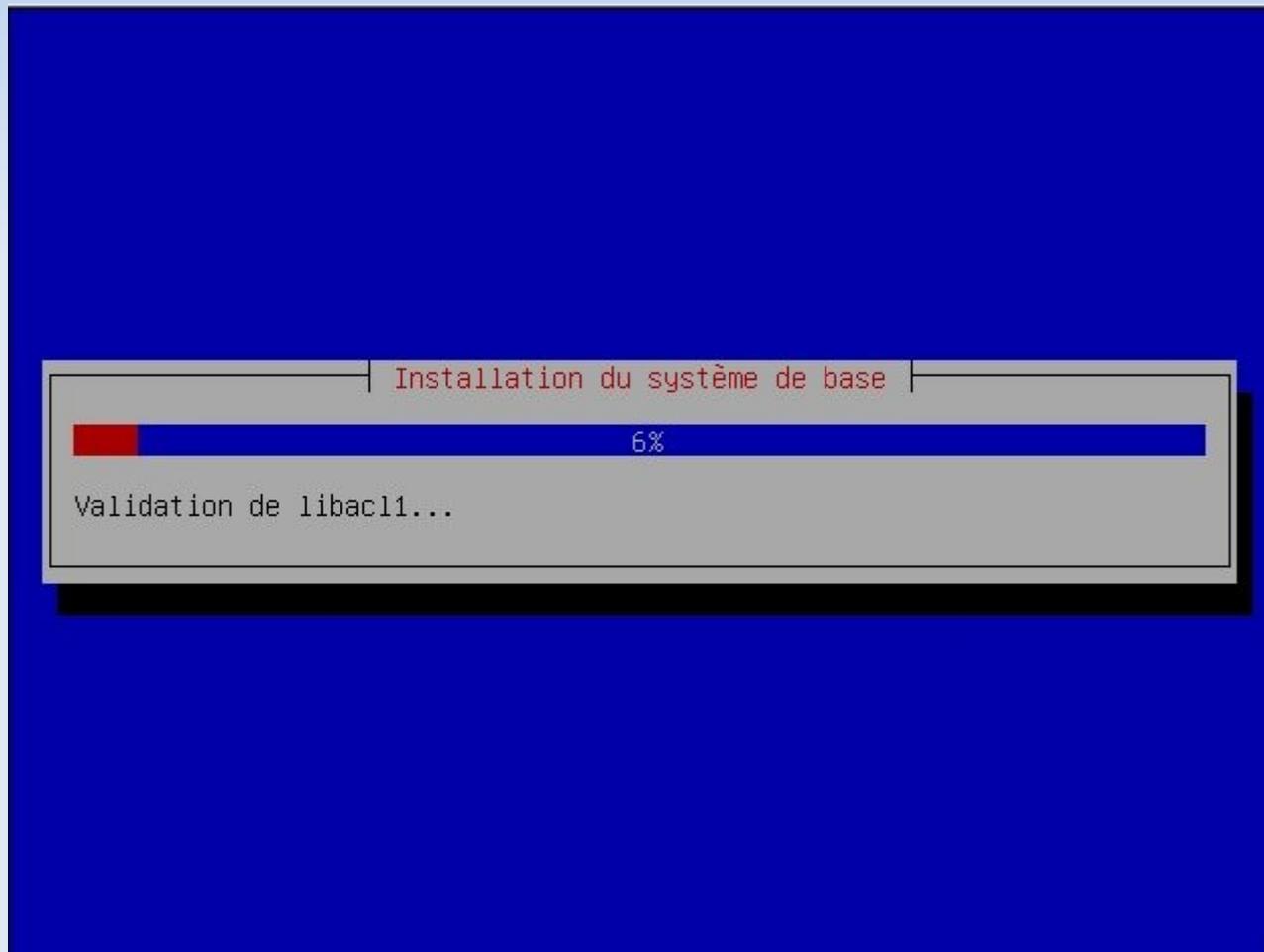
Veuillez indiquer le nom complet du nouvel utilisateur. Cette
information servira par exemple dans l'adresse origine des courriels
émis ainsi que dans tout programme qui affiche ou se sert du nom
complet. Votre propre nom est un bon choix.

Nom complet du nouvel utilisateur :
TRAN DINH Minh Tri
<Revenir en arrière>                                <Continuer>

<Tab> défilement: <Espace> sélection: <Entrée> activation des boutons
```



Installation du système de base





Sélection des logiciels à installer

```
[!] Sélection des logiciels

Actuellement, seul le système de base est installé. Pour adapter
l'installation à vos besoins, vous pouvez choisir d'installer un ou
plusieurs ensembles prédéfinis de logiciels.

Logiciels à installer :

[ ] Environnement graphique de bureau
[ ] Serveur web
[ ] Serveur d'impression
[ ] Serveur DNS
[ ] Serveur de fichiers
[ ] Serveur de courrier
[ ] Bases de données SQL
[ ] Ordinateur portable
[ ] Système standard

<Continuer>
```



Installation du bootloader

[!] Installer le programme de démarrage GRUB sur un disque dur

Il semble que cette nouvelle installation soit le seul système d'exploitation existant sur cet ordinateur. Si c'est bien le cas, il est possible d'installer le programme de démarrage GRUB sur le secteur d'amorçage du premier disque dur.

Attention : si le programme d'installation ne détecte pas un système d'exploitation installé sur votre ordinateur, la modification du secteur principal d'amorçage empêchera temporairement ce système de démarrer. Toutefois, le programme de démarrage GRUB pourra être manuellement reconfiguré plus tard pour permettre ce démarrage.

Installer le programme de démarrage GRUB sur le secteur d'amorçage ?

<Revenir en arrière>

<Oui>

<Non>



Redémarrage et premier boot!

```
[!!!] Terminer l'installation
      Installation terminée
L'installation est terminée et vous allez pouvoir maintenant démarrer
votre nouveau système. Veuillez vérifier que le support
d'installation (CD, disquettes) est bien retiré afin que le nouveau
système puisse démarrer et éviter de relancer la procédure
d'installation.

<Revenir en arrière>                <Continuer>
```

Bravo! L'installation terminée;)



Se connecter à Linux...

Linux est système:

- multi-utilisateur
- multi-tache

Pour se connecter on a besoin d'un:

- compte utilisateur
- mot de passe

```
Debian GNU/Linux 4.0 debian tty1
debian login: _
```

Se connecter avec le compte root ou le compte utilisateur que vous avez créé.



Se connecter à Linux...(suite)

- Si vous êtes en root, vous aller voir ce prompt:

```
vaio:~#
```

- Si vous êtes un utilisateur normal:

```
tri@vaio:~$
```

Pour arrêter le système, utiliser la commande **poweroff**

Attention:

Il faut éviter d'arrêter un système Linux!



Sous Linux:

- Tout est fichier
- Système de fichier est organisé par l'arborescence
- Dans un système de fichier Linux, il y a trois types de bloc:
 - * Superblock: contient les informations sur la taille des blocs, la tables des inodes,
 - * inode: chaque inode contient les informations d'un fichier comme les droits d'accès, la taille...
 - * Bloc de données: contient les données elles-même.



- **pwd** (Print Working Directory): afficher le répertoire actuel
- **cd** (Change Directory): changer de répertoire.

cd nom_du_répertoire

Chemin absolue: /home/user/docs

Chemin relative: si on est déjà dans /home => user/docs

Cas particuliers :

cd ~ renvoie vers le répertoire utilisateur de l'utilisateur courant.

cd . = vers répertoire courant

cd .. = vers répertoire parent

- **ls** Lister le contenu d'un répertoire



- `ls -a` («all»: tous)

Affiche tous les fichiers (y compris les fichiers caché .*)

- `ls -i`

Afficher les inodes des fichiers

- `ls -l` (long)

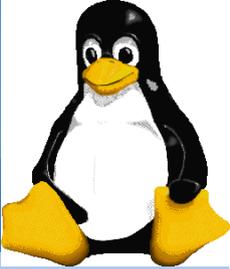
Affichage en format long (type, date, taille, propriétaire, permissions)

```
tri@vaio:~$ ls -l
```

```
-r--r--r-- 1 tri tri 2641561 2004-05-07 18:06 1 Installation Linux.pdf
-rw----- 1 tri tri 653173 2008-07-14 10:27 Award Request.pdf
-rw-r--r-- 1 tri tri 880 2007-06-21 17:33 comptes.txt
-rw-r--r-- 1 tri tri 68 2008-07-14 10:03 conge
drwxr-xr-x 9 tri tri 4096 2008-07-16 11:02 Desktop
```



- le premier caractère:
 - le caractère - représente un simple fichier ;
 - la lettre d représente un dossier (directory) ;
 - la lettre l représente un lien ;
- la première colonne fournit des informations sur les droits ;
- la colonne suivante est le nombre de liens pointant sur ce fichier ;
- la suivante donne le nom du propriétaire du fichier.
- la quatrième indique le nom du groupe qui peut accéder au fichier selon les actions autorisées par le propriétaire.
- la cinquième nous donne la taille du fichier en octets.
- la sixième donne la date et l'heure de la dernière modification du fichier.
- la dernière colonne donne le nom du fichier. S'il s'agit d'un lien, la référence du fichier est indiqué par un -> [source]



Substitutions sur noms de fichiers

Il est possible d'utiliser des caractères “joker” (des wildcards) pour remplacer un caractère ou une chaîne de caractères

* remplace n'importe quelle chaîne de caractères (même vide)

? remplace un caractère

[xx] remplace les caractères ou plage de caractères particuliers

ex : 1[89]99 => 1899 et 1999

1[1-9]99 => 1199, 1299, 1399, 1499, 1599, 1699, ... 1999

1*ab => 1ab, 1agagagaggdegzgab, 111111111ab,

a (tous les noms contenant la lettre a)

[A-Z][a-z] tous les mots de 2 lettres commençant par une majuscule.

{ensemble de mots séparés avec des virgules} essayer avec chaque mot



Exemple:

- `$ ls *txt`

L'interpréteur remplace d'abord `*txt` par tous les noms de fichiers et de répertoires finissant par `txt` (y compris `.txt`), sauf ceux commençant par `.`, et enfin exécute la ligne de commande `ls`.

- `$ ls -d .*`

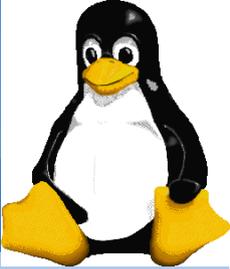
Affiche tous les fichiers et les répertoires commençant par `.`
`-d` indique à `ls` de ne pas afficher le contenu des dossiers `.*`

- `$ cat ?.log`

Affiche le contenu de tous les fichiers dont le nom commence par `l` caractère et finit par `.log`

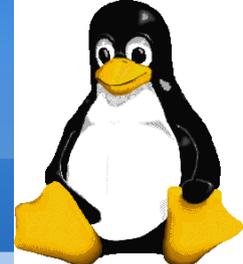
- `$ ls /etc/{p,gr}*`

Affiche tous les fichiers et les répertoires commençant par `p` ou `gr`



La commande cp

- `cp fichier_orig fichier_dest`
Crée une copie d'un fichier d'origine
- `cp fich1 fich2 fich3 ... rep`
Copie tous les fichiers vers le répertoire de destination
- `cp -i` (interactif)
Demande confirmation à l'utilisateur dans le cas où le fichier de destination existe déjà
- `cp -r rep_orig rep_dest` (récursive)
Copie du répertoire tout entier



La commande **tar** Archiver des fichiers

- Création d'une archive:

\$ tar cvzf archive fichiers ou répertoires

c: créer

v: verbeux. Utile pour suivre la progression de l'archivage

z: compresser avec gzip

f: fichier. Archive créée dans un fichier (sinon utilise une bande)

Extraire tous les fichiers d'une archive:

\$ tar xvzf archive

- Extraire seulement quelques fichiers d'une archive:

\$ tar xvzf <rchive fichiers ou rép

- Les fichiers ou répertoires sont donnés avec un chemin relatif au répertoire racine de l'archive.



La commande **rm** (remove) Effacer un fichier :

\$ rm fichier

Par défaut, rm demande confirmation (=rm -i), on peut le forcer à supprimer un fichier grâce à rm -f

- rm fich1 fich2 fich3 ...

Supprime les fichiers donnés

- rm -i (interactif)

Demande toujours à l'utilisateur de confirmer les suppressions

- rm -r rep1 rep2 rep3 (récursif)

Suppression des répertoires donnés et de tout leur contenu



La commande **mv** (move) Déplacer un fichier:

`mv fichiers_source destination`

Si plusieurs fichiers, destination = répertoire

- `mv ancien_nom nouveau_nom` (“move”: déplacer)

Change le nom du fichier ou du répertoire donné

- `mv -i` (interactif)

Si le fichier existe déjà, demander confirmation à l'utilisateur



Les liens symboliques et les liens physiques

Un lien symbolique est un fichier spécial qui est juste une référence au nom d'un autre (fichier ou répertoire)

Utile pour simplifier et réduire l'utilisation du disque quand deux fichiers ont le même contenu.

Exemple:

`biographie_anakin_skywalker -> biographie_darth_vador`

Comment distinguer les liens symboliques:

- `ls -l` affiche `->` et le fichier référencé par le lien
- GNU `ls` affiche les liens avec une couleur différente



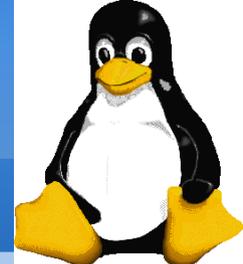
Création des liens symboliques

- Pour créer un lien symbolique (même ordre que dans cp):
`ln -s nom_fichier nom_lien`
- Pour créer un lien vers un fichier dans un autre répertoire, avec le même nom:
`ln -s ../LISEZ_MOI.txt`
- Pour créer plusieurs liens d'un coup dans un dossier donné:
`ln -s fich1 fich2 fich3 ... rep`
- Pour supprimer un lien:
`rm nom_lien`
Bien-sûr, cela ne supprime pas le fichier référencé par le lien!



Liens physiques

- Par défaut, ln crée des liens physiques
- Un lien physique vers un fichier est un fichier ordinaire, avec exactement le même contenu physique
- Bien qu'ils économisent toujours de la place, les liens physiques sont indiscernables des fichiers d'origine.
- Si vous supprimez le fichier d'origine, cela n'affecte pas le contenu du lien physique.
- Le contenu est supprimé quand il n'y a plus aucun fichier (lien physique) qui y fait référence



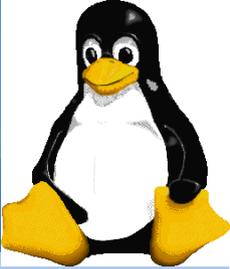
Création et suppression de répertoires

- `mkdir rep1 rep2 rep3 ...` (“make dir”: créer rép.)
Crée des répertoires avec les noms spécifiés
- `rmdir rep1 rep2 rep3 ...` (“remove dir”: suppr. rép.)
Supprime les répertoires spécifiés
Sécurité: ne fonctionne que quand les répertoires sont vides
Alternative: `rm -r`



Manipulation de base des fichiers

- Avant de modifier le contenu d'un fichier, vous voudrez sans doute en afficher le contenu :
`cat nom_du_fichier`
- De nombreux fichiers dépassent la taille d'un écran. Pour que l'affichage s'arrête à la fin de la page:
`more nom_du_fichier` ou `less nom_du_fichier`
- Pour créer un nouveau fichier vide:
`touch nom_du_fichier`



Afficher le contenu de fichier

Plusieurs façons d'afficher le contenu de fichiers

- `$ cat fich1 fich2 fich3 ...` (concaténer)
Met bout à bout et affiche le contenu des fichiers donnés
- `$ more fich1 fich2 ...` (plus de détails)
A chaque page, demande à l'utilisateur d'appuyer sur une touche pour continuer. Peut aussi aller directement à la première apparition d'un mot clé (commande `"/`)
- `$ less fich1 fich2 fich3 ...` (moins)
Fait plus que `more` avec moins!
Ne lit pas le fichier entier avant de commencer à afficher
Permet de remonter en arrière dans le fichier (commande `"?"`)



Les commandes head et tail

- `$ head [-n] fichier (tête)`
Affiche les n premières lignes (ou 10 par défaut) du fichier donné
N'a pas besoin d'ouvrir le fichier en entier pour le faire!
- `$ tail [-n] fichier (queue)`
Affiche les n dernières lignes (ou 10 par défaut) du fichier donné
Ne charge pas tout le fichier en mémoire. Très utile pour les gros fichiers.
- `$ tail -f fichier (follow: suivre)`
Affiche les 10 dernières lignes du fichier donné en continue.

Exemples

```
$ head bogues.txt
```

```
$ tail -f /var/log/messages
```



La commande grep

- `$ grep motif fichiers`
Parcourt les fichiers donnés et affiche les lignes qui correspondent au motif spécifié.
- `$ grep erreur *.log`
Affiche toutes les lignes contenant erreur dans les fichiers *.log
- `$ grep -i erreur *.log`
Idem, mais indifférent aux majuscules / minuscules
- `$ grep -ri erreur`
Idem, mais récursivement dans . et ses sous-répertoires
- `$ grep -v info *.log`
Affiche toutes les lignes des fichiers, sauf celles qui contiennent info



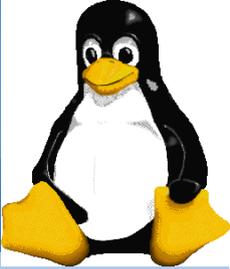
La commande sort

- `$ sort fichier (trier)`
Trie les lignes du fichier selon l'ordre des caractères et les affiche.
 - `$ sort -r fichier ("reverse": inverse)`
Idem, mais en ordre inverse
 - `$ sort -ru fichier`
u: unique. Idem, mais ne sort qu'une seule fois les lignes identiques.
- Exemples:
- ```
$ sort /etc/passwd
```
- ```
$ sort -r /etc/passwd
```



La commande wc

- `$ wc fichier`
Afficher le nombre d'octets, de mots et de lignes du fichier
 - `$ wc -l fichier`
Afficher le nombre de lignes du fichier
 - `$ wc -c fichier`
Afficher le nombre d'octets du fichier
 - `$ wc -w <fichier>`
Afficher le nombre de mots du fichier
- Exemples:
- ```
$ wc /etc/passwd
```
- ```
$ wc -c /etc/passwd
```



Comparaison de fichiers et répertoires

- `$ diff fichier1 fichier2`
Affiche les différences entre 2 fichiers, ou rien si les fichiers sont identiques.
- `$ diff -r rep1/ rep2/`
Affiche les différences entre fichiers de même nom dans les 2 répertoires.



La commande find

Commande pour chercher les fichiers

Quelques exemples:

- `$ find .-name "*.pdf"`

Recherche tous les fichiers *.pdf dans le répertoire courant (.) et ses sous-répertoires.

- `$ find docs -name "*.pdf" -exec xpdf {} ';'`

Recherche tous les fichiers *.pdf dans le répertoire docs et les affiche l'un après l'autre.

- `$ find /home/user -name *.txt -size +100k`

Recherchera dans le sous le répertoire de l'utilisateur user tous les fichiers qui finissent par ".txt" et qui ont une taille supérieure à 100 kilo-octets.



Les commandes whereis et which

- **whereis** Rechercher les fichiers exécutables, les fichiers de configuration, les sources et les pages de manuel de commande.
\$ whereis last
last: /usr/bin/last /usr/share/man/man1/last.1.bz2
- **which** Rechercher le chemin vers les fichiers exécutables
\$ which last
/usr/bin/last



La commande echo

Outil pour afficher ce qu'on lui donne comme argument

- `$ echo 1`
1
- `$ echo ~root`
/root
- `$ echo ~tri`
/home/tri
- `$ echo $(date)`
mercredi 16 juillet 2008, 14:31:46 (UTC+0700)
- `$ echo ${5+5}`
10



La commande locate

Outil de recherche à base d'expressions régulières, une alternative à find beaucoup plus rapide.

- locate clé

Affiche tous les fichiers sur votre système contenant clé dans leur nom.

- locate “*.pdf”

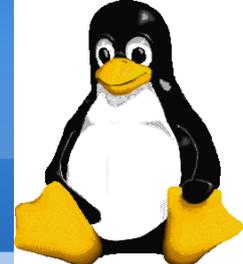
Affiche tous les fichiers *.pdf existant sur votre système.

- locate “/home/frigo/*mousse*”

Affiche tous les fichiers *mousse* dans le répertoire indiqué

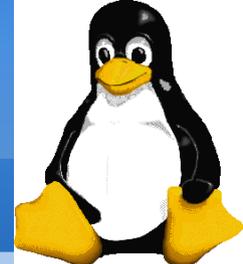
locate est bien plus rapide grâce à l'indexation de tous les fichiers dans une base de données dédiée, qui est mise à jour régulièrement.

find est plus adapté aux recherches sur les fichiers créés récemment.



Les commandes man et apropos

- `$ man commande` Afficher le manuel d'une commande, d'un fichier de configuration...
- `$ apropos mot_clé` de chercher une commande, à partir d'un mot-clé
`apropos = man -k`



Éditeurs de texte vi

- Les deux éditeurs de texte les plus connus et les plus utilisés sont vi et emacs. Les bons administrateurs préfèrent vi ;) vi est disponible sur tous les systèmes Unix.
- Difficile à apprendre pour les débutants habitués aux éditeurs graphiques.
- Très productif pour les utilisateurs avancés
- Souvent incontournable pour modifier des fichiers de configuration quand vous ne disposez que d'une console texte.



Utilisation de vi

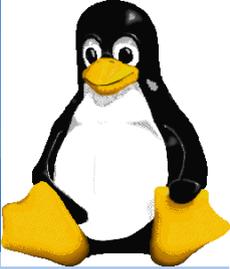
Pour éditer un fichier texte existant ou créer un nouveau fichier texte:

```
$ vi nom_du_fichier
```

Les modes de fonctionnement :

- Le mode Commande, dans lequel vous vous trouvez quand vous ouvrez vi. Si vous êtes dans un autre mode et que vous voulez revenir au mode commande, appuyez sur la touche Echap.
- Le mode Insertion, à partir du mode Commande, on passe en mode Insertion en tapant au choix I, i, a, A, o, O. Le texte saisi sera inséré:

i...	avant le curseur
I...	début de ligne courante
a...	après le curseur
A...	ajout en fin de ligne courante
o...	au-dessous de la ligne courante
O...	au-dessus de la ligne courante



Utilisation de vi (suite)

Dans le mode Commande, vous pouvez taper:

- :h pour accéder à l'aide
- :w pour enregistrer
- :w nom_du_fichier pour faire enregistrer-sous nom_du_fichier
- :q pour quitter
- :wq pour enregistrer et quitter
- :q ! pour quitter sans enregistrer les modifications
- /mot_clef pour faire rechercher un mot (n pour passer à l'itération suivante)
- :numéro_de_ligne pour aller directement à cette ligne
- G aller à la dernière ligne



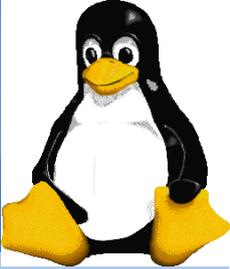
Utilisation de vi (suite)

- `y nombre_de_lignes y` (sans espace) pour copier ce nombre de ligne à partir du curseur (`yy` pour copier une ligne ou un groupe de mots en mode visuel),
- `d nombre_de_lignes d` (sans espace) pour couper ce nombre de ligne à partir du curseur (`dd` pour couper une ligne en mode commande ou un groupe de mots en mode visuel),
- `p` pour coller après le curseur
- `u` pour annuler la dernière modification. Vous pouvez appuyer plusieurs fois sur `u` pour annuler les dernières modifications.
- `Ctrl-r` pour annuler la dernière annulation. Vous pouvez renouveler la combinaison de touches pour annuler les annulations antérieures.
- `:%s/toto/tata/g` pour remplacer toutes les occurrences de la chaîne de caractères `toto` par la chaîne de caractère `tata`.



Manipulation des utilisateurs

- Par défaut, un utilisateur fait partie du groupe principal portant le même nom que le User. Il peut faire partie d'un autre groupe principal et de groupes secondaires. Le groupe principal est celui dont l'id apparaît dans le fichier `/etc/passwd`, et définit le groupe « possédant » le fichier lors de sa création par un utilisateur. En UNIX (et donc en LINUX), les fichiers n'appartiennent qu'à un seul groupe. L'utilisateur fera donc généralement partie de plusieurs groupes afin de faciliter la gestion des droits d'accès.
- Pour la gestion des utilisateurs, vous devez être connectés comme superutilisateur. Vous utilisez donc la commande `su` – qui vous donne les droits de root et son environnement de travail, comme s'il s'était connecté directement. `su` vous donne ses droits sans changer d'environnement.



Les utilisateurs

- Ajouter un utilisateur : **useradd** [options] user

Liste des options principales:

-g groupe : crée le user dans un groupe principal

-G groupe : crée le user dans un(des) groupe(s) secondaire(s)

-d répertoire : crée une home directory différente de celle créée par défaut

ex : # useradd -g users -G management mon_user

- Il faut ensuite établir le mot de passe de l'utilisateur.

passwd user

- Pour changer les paramètres du user : **usermod** avec les mêmes options que useradd

- Supprimer un utilisateur : **userdel** user

l'option -r permet de supprimer le répertoire utilisateur en même temps.



Les groupes

- Créer un groupe : **groupadd** groupe
- Modifier le nom d'un groupe : **groupmod** -n nouveaunom groupe
- Supprimer un groupe : **groupdel** groupe

NB: les fichiers appartenant à ce groupe ne seront pas supprimés et ne seront pas modifiés. Si le groupe à supprimer est le groupe principal d'un utilisateur, le groupe ne sera pas supprimé



Exemples

- # useradd tri
- # passwd tri
Changing password for user tri.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
- # groupadd vientiane
- # usermod -G vientiane tri
- # less /etc/group | grep vientiane
vientiane:x:504:tri,root
- # groups tri
tri : tri vientiane



Les permissions sur les fichiers

Pour afficher les permissions des fichiers d'un répertoire : `ls -l`

```
-rw-r--r--    1 root    root    4288 May 14 22:17    scmp-jpc
drwx-----    2 root    root    1024 Jun  4 07:12    mail
```

Le premier caractère décrit l'objet:

- pour les fichiers

l pour les liens symboliques

d pour directory

s, b ou c pour les périphériques

Ensuite 3 groupes de 3 lettres pour le propriétaire, le groupe et pour les autres



Les permissions sur les fichiers (suite)

3 types de droits d'accès:

- Accès en lecture (r: read)
- Accès en écriture (w: write)
- Droits d'exécution (x)

3 types de niveaux d'accès:

- Utilisateur (u): pour le propriétaire du fichier
- Groupe (g): tout fichier a un attribut "groupe", qui correspond à une liste d'utilisateurs
- Autres (o: others): pour tous les autres (propriétaire et groupe exclus)



Contraintes de droits d'accès

- x sans r est autorisé mais sans valeur.
Vous devez pouvoir lire un fichier pour l'exécuter.
- Les répertoires requièrent à la fois les droits r et x: x pour entrer, r pour accéder au contenu.
- Vous ne pouvez pas renommer, supprimer ou copier des fichiers dans un répertoire si vous n'avez pas accès en écriture à ce répertoire.
- Si vous avez accès en écriture à un répertoire, vous **POUVEZ** supprimer un fichier même si vous ne disposez pas de droits d'écriture pour ce fichier (souvenez-vous qu'un répertoire est juste un fichier décrivant une liste de fichiers). Cela permet même de modifier un fichier (le supprimer et le recréer) même protégé en écriture.



Exemples

- `-rw-r--r--`

Lisible et modifiable pour le propriétaire, seulement lisible pour les autres.

- `-rw-r-----`

Lisible et modifiable pour le propriétaire, seulement lisible pour les utilisateurs appartenant au groupe du fichier.

- `drwx-----`

Répertoire seulement accessible par son propriétaire

- `-----r-x`

Fichier exécutable seulement par les autres, mais ni par vos amis ni par vous-même. Droits d'accès parfaits pour un piège...



Modifier les permissions

La commande **chmod** permet de modifier les permissions sur les fichiers

chmod <à qui><opération><permission> <nomdufichier>

<à qui> :
 u (user)
 g (group)
 o (other)
 a (all)

<opération>:
 + (ajoute)
 - (supprime)
 = (remplace)

<permission> :
 r (read)
 w (write)
 x (execute)



Modifier les permissions (suite)

2 formats pour les droits d'accès:

- Format en base 8 (abc):

$a,b,c = r*4+w*2+x$ (r, w, x: booléens)

Exemple: `chmod 644 <fichier>`

(rw pour u, r pour g et o)

- Format symbolique. Facile à comprendre par des exemples:
`chmod go+r` : ajouter droit en lecture au groupe et aux autres
`chmod u-w` : supprimer droit d'écriture pour le propriétaire
`chmod a-x` : (a: all = tous). Enlever les droits d'exécution à tous les utilisateurs.



Autres options de chmod

```
$ chmod -R a+rX linux/
```

Rend linux et tout ce qu'il contient accessible à tout le monde!

- **R**: applique les changements récursivement
- **X**: x, mais seulement pour répertoires et fichiers déjà exécutables. Très pratique pour ouvrir récursivement l'accès à des répertoires, sans ajouter des droits d'exécution à tous les fichiers.

```
$ chmod a+t /tmp
```

- **t**: (“sticky”: collant). Permission spéciale pour les répertoires, autorisant uniquement l'effacement d'un fichier par son propriétaire ou par celui du répertoire.
- Utile pour les répertoires accessibles en écriture par plusieurs utilisateurs, comme /tmp.



Changement de propriétaire

Vous pouvez changer le propriétaire (owner = utilisateur ou groupe) d'un (ou plusieurs) fichier(s) ou groupe(s).

Le propriétaire étant l'utilisateur qui a créé le fichier.

La commande est

chown nouveaupropriétaire fichier



Changement de groupe

Vous pouvez aussi changer le groupe d'un fichier grâce à
chgrp nouveaugroupe fichier

ex : # chgrp users testfile

NB : on peut attribuer en une fois un nouveau propriétaire et un
nouveau groupe:

chown nouveaupropriétaire.nouveaugroupe fichier

ex : # chown test.users testfile



Les redirections: sortie standard

Plus de détails sur les sorties des commandes

- Toutes les commandes qui sortent du texte sur votre terminal le font en écrivant sur leur sortie standard.
- La sortie standard peut être écrite (redirigée) dans un fichier en utilisant le symbole >
- La sortie standard peut être rajoutée à la fin d'un fichier existant par le symbole >>



Les redirections: sortie standard (suite)

Exemples de redirection de sortie

- `ls ~toto/* > ~temp/plaisr.txt`
- `cat text1_text2.txt > docs.txt`
`cat han_solo.txt >> biographies_starwars.txt`
- `echo "LISEZ_MOI: Aucun fichier ou répertoire de ce type" > LISEZ_MOI`

Moyen facile de créer un fichier sans éditeur de texte.

Exemple:

```
ls -l > filelist
```

envoie le listing du dossier vers le fichier filelistL



Entrée standard (suite)

On peut rediriger l'entrée standard:

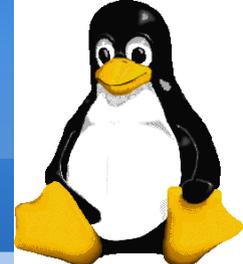
- `sort < participants.txt`

L'entrée standard de `sort` est prise dans le fichier indiqué.

Le caractère `<` lit le contenu d'un fichier et l'envoie vers l'entrée d'une commande comme si c'était l'utilisateur qui tapait au clavier

- `mail user@company.com < filelist`

créé un mail avec comme corps de message le contenu du fichier `filelist`



Les tuyaux (pipeline)

- Les tuyaux Unix sont très utiles pour rediriger la sortie standard d'une commande vers l'entrée standard d'une autre commande.
- Exemples
 - `cat *.log | grep -i erreur | sort`
 - `grep -ri error /var/log | grep -v "syntax" | sort -u > erreurs_synt.txt`
 - `grep -ri erreur . | grep -v "ignore" | sort -u \ > erreurs_graves.log`
 - `cat /home/*/devoirs.txt | grep note | more`
- Il s'agit d'une des fonctionnalités les plus puissantes des shells Unix!



L'erreur standard

- Les messages d'erreur sont normalement envoyés (si le programme est bien écrit) vers l'erreur standard au lieu de la sortie standard.
- L'erreur standard peut être redirigée par `2>` ou `2>>`
- Exemple:
`cat f1 f2 faux_fich > nouv_fich 2> fich_erreur`
- Note: 1 est le descripteur de la sortie standard, donc `1>` est équivalent à `>`
- On peut rediriger à la fois la sortie et l'erreur standard vers le même fichier en utilisant `&>`
`cat f1 f2 faux_fich &> fich_entier`



Séquence de commandes

- Possibilité de taper la prochaine commande dans votre terminal même si la commande courante n'est pas terminée.
- Possibilité de séparer plusieurs commandes par le symbole ;
echo "Vous êtes le plus beau"; sleep 10; echo "des menteurs"
- Conditions: utiliser `||` (ou) ou `&&` (et):
ls abc || echo "Désolé, abc n'existe pas"
N'exécute echo que si la première commande échoue.
ls ~sd6 && cat ~sd6/* > ~sydney/recettes.txt
N'affiche le contenu des fichiers du répertoire que si la commande ls réussit (indique un accès en lecture).



Exemples de redirection de sortie

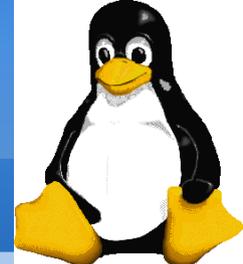
- `ls ~toto/* > ~temp/plaisr.txt`
- `cat text1_text2.txt > docs.txt`
`$ cat han_solo.txt >> biographies_starwars.txt`
- `echo "LISEZ_MOI: Aucun fichier ou répertoire de ce type" > LISEZ_MOI`

Moyen facile de créer un fichier sans éditeur de texte.

Exemple:

```
ls -l > filelist
```

envoie le listing du dossier vers le fichier filelist



Entrée standard

De nombreuses commandes, quand on ne leur donne pas d'arguments en entrée, peuvent chercher leurs entrées sur l'entrée standard.

```
$sort
```

```
windows
```

```
linux
```

```
[Ctrl][D]
```

```
linux
```

```
windows
```

sort prend l'entrée standard comme entrée: dans ce cas, ce que vous tapez dans le terminal



Entrée standard (suite)

On peut rediriger la sortie standard:

- `$ sort < participants.txt` (terminé par `[Ctrl][D]`)

L'entrée standard de `sort` est prise dans le fichier indiqué.

Le caractère `<` lit le contenu d'un fichier et l'envoie vers l'entrée d'une commande comme si c'était l'utilisateur qui tapait au clavier.

- `$ mail user@company.com < filelist`

crée un mail avec comme corps de message le contenu du fichier `filelist`



Les tuyaux (pipeline)

- Les tuyaux Unix sont très utiles pour rediriger la sortie standard d'une commande vers l'entrée standard d'une autre commande.
- Exemples:
 - `cat *.log | grep -i erreur | sort`
 - `grep -ri error /var/log | grep -v "syntax" | sort -u > erreurs_synt.txt`
 - `grep -ri erreur . | grep -v "ignore" | sort -u \ > erreurs_graves.log`
 - `cat /home/*/devoirs.txt | grep note | more`
- Il s'agit d'une des fonctionnalités les plus puissantes des shells Unix!



L'erreur standard

- Les messages d'erreur sont normalement envoyés (si le programme est bien écrit) vers l'erreur standard au lieu de la sortie standard.
- L'erreur standard peut être redirigée par `2>` ou `2>>`
- Exemple:
`cat f1 f2 faux_fich > nouv_fich 2> fich_erreur`
- Note: 1 est le descripteur de la sortie standard, donc `1>` est équivalent à `>`
- On peut rediriger à la fois la sortie et l'erreur standard vers le même fichier en utilisant `&>`
`cat f1 f2 faux_fich &> fich_entier`



Séquence de commandes

- Possibilité de taper la prochaine commande dans votre terminal même si la commande courante n'est pas terminée.
- Possibilité de séparer plusieurs commandes par le symbole ;
echo "Vous êtes le plus beau"; sleep 10; echo "des menteurs"
- Conditions: utiliser `||` (ou) ou `&&` (et):
ls abc || echo "Désolé, abc n'existe pas"
N'exécute echo que si la première commande échoue.
ls ~sd6 && cat ~sd6/* > ~sydney/recettes.txt
N'affiche le contenu des fichiers du répertoire que si la commande ls réussit (indique un accès en lecture).



Gestion des processus

- Sous Unix, chaque tâche s'exécute au sein d'un ou plusieurs processus. Chaque processus a un PID (Processus ID) qui lui est propre. Si un processus plante, les autres processus ne sont pas affectés.
- La commande **ps** permet de lister les processus
 - **ps -ux**
Affiche tous les processus appartenant à l'utilisateur courant.
 - **ps -aux**
Affiche tous les processus existant sur le système



Gestion des processus (suite)

```
tri@vaio:~$ ps -aux | grep tri | grep bash
```

```
tri 11680 0.1 0.7 6332 3680 pts/0 Ss 22:35 0:00 bash
tri 11805 0.2 0.7 6332 3680 pts/1 Ss 22:36 0:00 bash
tri 11886 4.0 0.7 6332 3676 pts/2 S+ 22:37 0:00 bash
tri 11913 0.0 0.1 3012 764 pts/0 S+ 22:37 0:00 grep bash
USER, PID, %CPU, %MEM, VSZ, RSS, TTY, STAT, START, TIME
COMMAND
```

PID: (Process ID) Identifiant du processus

VSZ: (Virtual SiZe) Taille virtuelle du processus

RSS: (ReSident Size) Nombre de Ko occupés en mémoire

TTY: (TeleTYpe) Terminal

STAT: Statut: R (Runnable: exécutable), S (Sleep: endormi), W (paging: en cours de pagination), Z (Zombie)...



Arrêt de processus

- **kill pids**

Envoie un signal d'arrêt aux processus spécifiés. Cela permet aux processus de sauvegarder leurs données et s'arrêter eux-mêmes.

- **kill -9 pids**

Envoie un signal d'arrêt immédiat. Le système lui-même se charge d'arrêter les processus. Utile quand une tâche est vraiment bloquée

- **killall commande**

Arrête toutes les tâches exécutant commande.

Exemple:

```
killall bash
```



Montage et démontage un système de fichier

- La commande **mount** sert à intégrer un système de fichier dans le système de fichier de la racine. Elle doit être exécutée en tant que root.

```
# mount options /dev/periphérique /mnt/répertoire_de_montage à
```

 condition que le type de système de fichier soit supporté par le noyau et que le répertoire `/mnt/répertoire_de_montage` existe déjà.
- Pour démonter ce système de fichier, la commande **umount**:

```
# umount /mnt/répertoire_de_montage
```
- Par contre, n'importe quel utilisateur peut taper la commande `mount` tout court pour savoir quels sont les systèmes de fichiers montés à l'instant d'exécution de la commande.



Les variables d'environnement

La commande **env** liste les variables d'environnement du user.

Pour afficher le contenu d'une variable, utilisez la commande suivante

```
echo $nom_de_la_variable
```

```
tri@vaio:~$ echo $PATH
```

```
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/bin/X11
```

Quelques variables d'environnement standards:

- **\$DISPLAY** Écran sur lequel afficher les applications graphiques
- **\$HOME** Répertoire de l'utilisateur courant
- **\$HOSTNAME** Nom de la machine locale
- **\$USER** Nom de l'utilisateur courant
- **\$PATH** Spécifie l'ordre de recherche de commandes pour le shell



Mise en garde sur PATH

- Il est fortement recommandé de ne pas avoir le répertoire “.” dans votre variable d’environnement PATH, en particulier pas au début.
- Un intrus pourrait placer un fichier ls malveillant dans vos répertoires. Il serait exécuté à chaque appel de ls depuis ces répertoires et pourrait s’attaquer à vos données personnelles.
 - Si vous avez un fichier exécutable de nom test dans un répertoire, il sera utilisé à la place du programme test par défaut et certains scripts ne fonctionneront plus correctement.
 - Chaque fois que vous entrez dans un nouveau répertoire, le shell perdra du temps à mettre à jour sa liste de commandes disponibles. Lancez vos propres commandes ainsi: ./test



Alias

Les shells vous permettent de définir des alias: des raccourcis pour des commandes que vous utilisez très souvent

- `alias ls='ls -la'`

Utile pour toujours lancer des commandes avec certains paramètres

- `alias rm='rm -i'`

Utile pour faire que rm demande toujours une confirmation

- `alias tor='trouver_objet_rambaldi --vite --risque'`

Utile pour remplacer de longues commandes utilisées régulièrement.

- `alias cia='. /home/sydney/env/cia.sh'`

Utile pour initialiser rapidement un environnement (. est une commande shell pour exécuter le contenu d'un script shell)



Le système de gestion des paquets Debian

Trois programmes s'occupent de la gestion des paquets Debian:

- dpkg
- apt-get
- dselect



dpkg

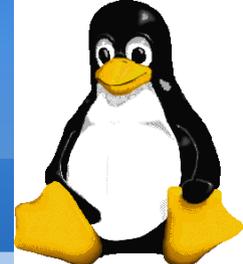
- Il faut éviter de l'utiliser en temps normal pour installer et désinstaller des paquets, puisqu'il ne gère pas les dépendances entre paquets.
- Par contre, c'est souvent le seul moyen d'installer des paquets qui ne sont pas présents dans la distribution. Il faut alors télécharger les fichiers correspondant aux paquets et les installer avec la commande dpkg.



dpkg (suite)

Utilisation:

- Installe les paquets paquet1 et paquet2 (comme dpkg ne gère pas les dépendances, il faut installer en même temps les paquets qui dépendent l'un de l'autre, si une ancienne version du paquet est déjà installée, elle sera remplacée)
dpkg -i paquet1_0.1_i386.deb paquet2_0.2_i386.deb
- Désinstalle le paquet paquet1 mais ne supprime pas ses fichiers de configuration :
dpkg -r paquet1
- Désinstalle le paquet paquet1 et supprime ses fichiers de configuration
dpkg -r --purge paquet1



dpkg (suite)

- Reconfigure le paquet paquet1 qui est déjà installé :
`# dpkg-reconfigure paquet1`
- Affiche la liste des fichiers installés par le paquet vim :
`% dpkg -L vim`
- Affiche la liste des paquets installés :
`% dpkg -l`

Pour plus d'informations ou pour avoir la liste complète des options disponibles, consultez le manuel de dpkg : `man dpkg`



apt-get

- Gère l'installation et le retrait des paquets en tenant compte des dépendances ainsi que le téléchargement des paquets s'ils sont sur une source réseau.
- Les sources des paquets sont définies dans le fichier de configuration /etc/apt/sources.list

```
$ cat /etc/apt/sources.list
```

```
deb http://ftp.hk.debian.org/debian/ etch main contrib non-free
```

```
deb http://security.debian.org/ etch/updates main contrib non-free
```

deb pour définir une source de paquets binaires ;

deb-src pour définir une source de paquets sources (cela n'intéressera que les développeurs).

Pour plus de détails, man sources.list.



Utilisation de apt-get

- Mettre à jour la liste des paquets disponibles (pour les sources réseau ou fichier, il doit aller voir si elles ont été mises à jour; pour les sources CD, il ne fait rien de particulier) :

```
# apt-get update
```

- Mettre à jour tous les paquets déjà installés à la dernière version disponibles dans les sources :

```
# apt-get upgrade
```

- Idem que la commande précédente, mais cette commande est optimisée pour les migrations vers une version supérieure de Debian (par exemple passer de la version stable à la version instable) :

```
# apt-get dist-upgrade
```



Utilisation de apt-get (suite)

- Installe les paquets paquet1 et paquet2 et tous les paquets dont ils dépendent :
apt-get install paquet1 paquet2
- Désinstalle le paquet paquet1 sans effacer ses fichiers de configuration :
apt-get remove paquet1
- Idem que la commande précédente mais ses fichiers de configuration sont supprimés :
apt-get remove --purge paquet1
- Efface du disque dur les paquets téléchargés pour être installés (inutile quand la source est un CD ou un fichier du système de fichiers local) :
apt-get clean

Outil plus évolué que apt-get: **aptitude**



Rechercher un paquet et afficher sa description

- Pour chercher un paquet dans la base des paquets disponibles:
`$ apt-cache search liste_de_mots_clés`
Par exemple, pour chercher un compilateur Fortran, tapez :
`$ apt-cache search fortran compiler`
Il recherche alors les termes "fortran" et "compiler " dans la description de tous les paquets disponibles. Dans les réponses, vous obtenez notamment :
g77 - The GNU Fortran 77 compiler.
- Pour afficher les caractéristiques et la description du paquet paquet1:
`$ apt-cache show paquet1`



Configurer l'interface réseau d'une machine Linux

Éditer le fichier `/etc/network/interfaces`

```
# vi /etc/network/interfaces
```

```
auto lo
```

```
iface lo inet loopback
```

```
auto eth0
```

```
iface eth0 inet static
```

```
    address 192.168.1.50
```

```
    netmask 255.255.255.0
```

```
    broadcast 192.168.1.255
```

```
    gateway 192.168.1.1
```



Configurer l'interface réseau d'une machine Linux (suite)

- Pour que le système tienne compte des modifications, il faut redémarrer le service networking:

```
# /etc/init.d/networking restart
```

```
Reconfiguring network interfaces: done.
```

- On peut aussi configurer directement l'interface réseau directement avec la commande **ifconfig**



L'accès à distance par SSH

- SSH signifie Secure SHell. C'est un protocole qui permet de faire des connexions sécurisées (i.e. cryptées) entre un serveur et un client SSH. Nous allons utiliser le programme OpenSSH, qui est la version libre du client et du serveur SSH.

- Installation du client et du serveur SSH

Le client SSH est disponible dans le paquet openssh-client, qui est préinstallé.

Pour pouvoir vous connecter à distance, il faut installer le serveur SSH :

```
# apt-get install openssh-server
```

- Se logger par SSH

```
$ ssh login@nom_du_serveur_SSH
```



Transfert de fichiers par SSH

- SSH signifie Secure SHell. C'est un protocole qui permet de faire des connexions sécurisées (i.e. cryptées) entre un serveur et un client SSH. Nous allons utiliser le programme OpenSSH, qui est la version libre du client et du serveur SSH.

- Installation du client et du serveur SSH

Le client SSH est disponible dans le paquet openssh-client, qui est préinstallé.

Pour pouvoir vous connecter à distance, il faut installer le serveur SSH :

```
# apt-get install openssh-server
```

- Se logger par SSH

```
$ ssh login@nom_du_serveur_SSH
```



Transfert de fichiers par SSH (suite)

Utiliser SCP (Ssh CoPy)

La syntaxe:

- `scp utilisateur1@machine1:/fichier_source
utilisateur2@machine2:/fichier_destination`

Exemples:

- pour transférer le fichier `test1.txt` situé dans le répertoire courant vers le home du compte `toto` de la machine `ordi1.exemple.org`
`$ scp test1.txt toto@ordi1.exemple.org:`
- pour récupérer le fichier `test2.txt` situé le home de l'utilisateur `toto` de la machine `ordi2.exemple.org` et l'écrire dans le répertoire courant :
`$ scp toto@ordi2.exemple.org:test2.txt .`



Transfert de fichiers par SSH (suite)

- pour récupérer tous les fichiers ayant l'extension .txt situés dans le répertoire /usr/local de la machine ordi2.exemple.org et l'écrire dans le sous-répertoire test-scp du répertoire courant:

```
$ scp toto@ordi2.exemple.org:/usr/local/*.txt test-scp
```
- pour transférer l'intégralité du sous-répertoire test-scp du répertoire courant vers le sous répertoire incoming du home de l'utilisateur toto de la machine ordi1.exemple.org :

```
$ scp -r test-scp toto@ordi1.exemple.org:/incoming
```



Telnet

Telnet est un protocole qui permet d'accéder à distance à une machine, mais la connexion n'est pas sécurisée : le mot de passe et les données sont transférés en clair ! Telnet ne permet pas de faire des transferts de fichiers. Il est donc conseillé de ne pas utiliser Telnet mais uniquement SSH.

```
$ telnet nom_du_serveur_telnet
```

Comme il n'est pas sécurisé, on n'utilise telnet que pour faire des tests. Par exemple, pour voir si le services web d'un serveur quelconque fonctionne:

```
$ telnet nom_du_serveur 80
```



Partage de fichiers sous Unix

- NFS (Network File System) est un protocole standard de partage de répertoires sous Unix/Linux.



NFS côté client

- Pour monter le répertoire `/home/ftp/` partagé par la machine dont le nom DNS est `ordi1.exemple.org` dans le répertoire `/mnt/test` déjà créé, utilisez la commande **mount**:

```
# mount -t nfs ordi1.exemple.org:/home/ftp /media/test
```

- Une fois que vous n'avez plus besoin de ce partage, vous pouvez le démonter :

```
# umount /media/test
```

- Pour que ce répertoire soit monté à chaque démarrage, rajoutez la ligne suivante dans le fichier de configuration `/etc/fstab` :

```
ordi1.exemple.org:/home/ftp          /media/test          nfs  
soft,timeo=5,intr,rsize=8192,wsiz=8192
```

- Pour comprendre les options, **man mount**.



NFS côté serveur

- Il faut installer le paquet **nfs-kernel-server**:
apt-get install nfs-kernel-server
- Editez le fichier /etc/exports et rajoutez la ligne suivante pour partager le répertoire /home/test/ à la machine ordi2.exemple.org:
/home/test ordi2.exemple.org(rw,root_squash)
- Enfin, demandez à nfs-kernel-server de démarrer:
/etc/init.d/nfs-kernel-server start
Exporting directories for NFS kernel daemon...done.
Starting NFS kernel daemon: nfsd mountd.
- Par la suite, il suffira de lui dire de relire son fichier de configuration après chaque modification :
/etc/init.d/nfs-kernel-server reload
Re-exporting directories for NFS kernel daemon...done.



Monter un serveur Web

- Installer le paquet

Le serveur web OpenSource le plus utilisé dans le monde s'appelle Apache (<http://www.apache.org/>). Pour l'installer :

```
# apt-get install apache2
```

- Configurer Apache

Les fichiers de configuration d'Apache sont dans le répertoire `/etc/apache/`.

```
#vi /etc/apache2/sites-enabled/000-defaults
```

```
# /etc/init.d/apache2 reload
```

Reloading apache configuration.



Installation de la distribution Ubuntu 8.04 Hardy Heron



- Basé sur Linux Debian unstable (sid)
- Nouvelle version chaque 6 mois
- 18 mois de support pour la version précédente
- Mises à jour faciles en utilisant les outils et la philosophie de Debian
- Environnement de travail GNOME
- Dernières versions des applications les plus populaires
- Mise à jour de sécurité régulières