

ResEI 101 : Présentation des fonctionnalités de Linux utilisées au ResEI

Association ResEI
<gestion@resel.enst-bretagne.fr>

Réseau des Élèves de l'ENST Bretagne

23 Septembre 2005



Ces slides sont sous licence GPL (General Public Licence). Ils sont disponibles, avec leur code source sur le site de l'Association ResEI (<http://resel.enst-bretagne.fr>).

Ils ont été créés à partir de logiciels libres (\LaTeX -beamer).

Plan

- 1 Introduction
 - GNU
 - Linux
 - Linux vs. Windows
- 2 Connaissances de base
 - Arborescence type
 - Utilisateurs et groupes
 - Permissions des fichiers
 - Entrées et sorties standard
- 3 Gestion des disques durs
 - Fichiers des partitions
 - Montage
- 4 Noyau et compilation
 - Compilation
 - Noyau
 - Modules

Plan

1 Introduction

- GNU
- Linux
- Linux vs. Windows

2 Connaissances de base

- Arborescence type
- Utilisateurs et groupes
- Permissions des fichiers
- Entrées et sorties standard

3 Gestion des disques durs

- Fichiers des partitions
- Montage

4 Noyau et compilation

- Compilation
- Noyau
- Modules

GNU

- Projet Lancé en 1984 aux États-Unis par Richard Stallman
- Sous licence GPL (General Public Licence) :
 - utilisation ;
 - copie ;
 - modification ;
 - redistribution sous la même licence ;
- Partenariat avec la FSF (Free Software Fondation)

Linux

- Projet lancé en 1992 par Linus Torvalds, sous licence GPL, basé sur Unix
- Noyau le plus utilisé pour le système GNU, mais d'autres existent GNU/Hurd, NetBSD, FreeBSD
- Le noyau :
 - fait le lien entre les applications et le matériel
 - gère les processus qui tournent
 - gère les fichiers, les droits
 - et bien d'autres choses, cf. Majeure INF
- Plusieurs distributions GNU/Linux (Debian, Ubuntu, Mandriva, Fedora, etc.)

Linux vs. Windows

- Plus sécurisé et réactif (énorme communauté de développeurs)
- Choix de l'interface graphique (aucune, Gnome, KDE, etc.)
- Légèreté du système
- Le système GNU permet une mise à jour automatique de tous les logiciels
- De plus en plus de périphériques reconnus
- Mais il manque des applications professionnelles, des jeux et le support de certains périphériques

Plan

- 1 Introduction
 - GNU
 - Linux
 - Linux vs. Windows
- 2 **Connaissances de base**
 - **Arborescence type**
 - **Utilisateurs et groupes**
 - **Permissions des fichiers**
 - **Entrées et sorties standard**
- 3 Gestion des disques durs
 - Fichiers des partitions
 - Montage
- 4 Noyau et compilation
 - Compilation
 - Noyau
 - Modules

Arborescence type

- Arborescence standardisée (FHS : Filesystem Hierarchy Standard)
- Arborescence commençant par / contenant principalement :
 - /bin : binaries, contient le minimum vital des commandes
 - /etc : contient les fichiers de configuration généraux
 - /home : contient les répertoires personnels des utilisateurs
 - /root : répertoire personnel de l'utilisateur root
 - /boot : contient le noyau et le nécessaire pour le démarrage
 - /var : contient les données qui varient souvent, et les logs
 - /srv : contient les données qui se servent du système
 - /tmp : répertoire accessible par tout le monde pour stocker des fichiers temporaires
 - /dev : répertoire contenant les fichiers des périphériques
 - /usr : contient les programmes

Utilisateurs et groupes

- Gestion simple et efficace des utilisateurs, identifiés par un numéro (uid)
- Les utilisateurs peuvent se regrouper dans des groupes
- Le superutilisateur root a tous les droits
- Les autres ne peuvent pas modifier le système

Permissions des fichiers

- Chaque fichier et répertoire possède des permissions
- Les droits peuvent être : lecture, écriture, exécution
- Chaque fichier a un utilisateur et un groupe propriétaire
- Exemple :
`rw-r-xr-- eroux www-data monfichier`
 - l'utilisateur eroux a les droits d'écriture et de lecture
 - les utilisateurs du groupe www-data ont les droits de lecture et d'exécution
 - les autres ont seulement le droit en lecture
- le droit d'exécution pour un répertoire permet d'y entrer
- une gestion des droits plus fine est possible avec les ACL

Permissions des fichiers

- Chaque fichier et répertoire possède des permissions
- Les droits peuvent être : lecture, écriture, exécution
- Chaque fichier a un utilisateur et un groupe propriétaire
- Exemple :
`rw-r-xr-- eroux www-data monfichier`
 - l'utilisateur eroux a les droits d'écriture et de lecture
 - les utilisateurs du groupe www-data ont les droits de lecture et d'exécution
 - les autres ont seulement le droit en lecture
- le droit d'exécution pour un répertoire permet d'y entrer
- une gestion des droits plus fine est possible avec les ACL

Permissions des fichiers

- Chaque fichier et répertoire possède des permissions
- Les droits peuvent être : lecture, écriture, exécution
- Chaque fichier a un utilisateur et un groupe propriétaire
- Exemple :

```
rw-r-xr-- eroux www-data monfichier
```

- l'utilisateur eroux a les droits d'écriture et de lecture
- les utilisateurs du groupe www-data ont les droits de lecture et d'exécution
- les autres ont seulement le droit en lecture
- le droit d'exécution pour un répertoire permet d'y entrer
- une gestion des droits plus fine est possible avec les ACL

Permissions des fichiers

- Chaque fichier et répertoire possède des permissions
- Les droits peuvent être : lecture, écriture, exécution
- Chaque fichier a un utilisateur et un groupe propriétaire
- Exemple :

```
rw-r-xr-- eroux www-data monfichier
```

- l'utilisateur eroux a les droits d'écriture et de lecture
- les utilisateurs du groupe www-data ont les droits de lecture et d'exécution
- les autres ont seulement le droit en lecture
- le droit d'exécution pour un répertoire permet d'y entrer
- une gestion des droits plus fine est possible avec les ACL

Entrées et sorties standard

- Chaque programme a une entrée (STDIN) et une sortie (STDOUT) standard
- le STDIN est par exemple une liste d'arguments
- le STDOUT peut être un affichage à l'écran en console, un son, etc.
- Des redirections de ces entrées et sorties peuvent être faites vers d'autres programmes ou des fichiers, par exemple
`ls > toto.text` va écrire la liste des fichiers du repertoire courant dans le fichier toto.text
- Un pipe redirige le STDOUT d'un programme vers le STDIN d'un autre, par exemple
`ls -l /dev | more`
va afficher la list des fichiers de /dev page par page

Plan

- 1 Introduction
 - GNU
 - Linux
 - Linux vs. Windows
- 2 Connaissances de base
 - Arborescence type
 - Utilisateurs et groupes
 - Permissions des fichiers
 - Entrées et sorties standard
- 3 **Gestion des disques durs**
 - **Fichiers des partitions**
 - **Montage**
- 4 Noyau et compilation
 - Compilation
 - Noyau
 - Modules

Fichiers des partitions

- En Unix tout est fichier, même les périphériques, et les partitions :
 - /dev/hda représente le premier disque dur IDE
 - /dev/hdb1 représente la première partition primaire du second disque dur IDE
 - /dev/hda5 représente la première partition logique du premier disque dur
 - /dev/mda représente le premier disque dur raid SCSI
 - etc.
- il existe même des fichiers de périphériques virtuels (/dev/zero, /dev/random, etc.)

Montage

- Une partition peut-être *montée* à un endroit de l'arborescence, c'est à dire que le système va interpréter les données sur le disque dur et faire pointer un répertoire vers ces données, par exemple
`mount -t vfat /dev/hda1 /windows`
va faire apparaître votre partition windows dans `/windows`.
- `mount` est un utilitaire puissant qui permet de monter des images iso par exemple

Plan

- 1 Introduction
 - GNU
 - Linux
 - Linux vs. Windows
- 2 Connaissances de base
 - Arborescence type
 - Utilisateurs et groupes
 - Permissions des fichiers
 - Entrées et sorties standard
- 3 Gestion des disques durs
 - Fichiers des partitions
 - Montage
- 4 **Noyau et compilation**
 - **Compilation**
 - **Noyau**
 - **Modules**

Compilation

- Dans 90% des distributions, le système et les programmes sont installés à partir de paquets déjà compilés (.deb, .rpm)
- Possibilité de compiler soi-même ses programmes à partir des sources
- Compilation en 3 étapes :
 - ./configure pour vérifier les dépendances et choisir les options de compilation, qui vont influencer le programme final
 - make qui compile avec les options choisies
 - make install qui procède à l'installation des fichiers ainsi compilés dans le système

Noyau

- Le noyau est lui-aussi un logiciel et peut-être compilé avec différentes configurations
- Les développeurs Linux sont très nombreux et issus de toutes les distributions, de nouvelles versions de plus en plus sécurisées et supportant de plus en plus de matériel sont disponibles sur `kernel.org`
- La plupart des distributions intègrent un noyau précompilé et patché pour optimiser les performances de la distribution
- Le noyau est le fichier `/boot/vmlinuz-x`, `x` étant le numéro de version

Modules

- Certaines parties du noyau peuvent prendre la forme de modules, afin d'alléger le noyau
- On peut activer ou désactiver des modules à sa guise
- Une image (initrd) des modules peut être créée pour activer le support de matériel qui n'est accessible qu'en module