

# Administration Unix

Matthieu Pérotin

23 janvier 2006

## Unix

Un peu d'histoire

Pourquoi utiliser Unix ?

Linux

## Qu'est-ce qu'un administrateur ?

Rôle

## Objectifs de ce cours

Ce que nous n'allons pas traiter

Ce dont nous allons parler

## Au début régnait le Chaos

Nous sommes au début de l'informatique, dans les années 50 et au début des années 60

- ▶ Les ordinateurs sont de gros Mainframes
- ▶ Ils ne savent pas communiquer entre eux
- ▶ Les systèmes d'exploitation sont rudimentaires
- ▶ Pire : chaque ordinateur a son propre langage

## Au début régnait le Chaos

Aucune notion d'inter-opérabilité ou de compatibilité ascendente

- ▶ Chaque nouvel ordinateur, y compris d'un même constructeur, nécessite une réécriture des programmes
- ▶ Il est nécessaire de rentrer de nouveau les données "métier" à chaque changement
- ▶ Chaque mise à jour a un coût tout à fait rédhibitoire

## Multics

La frustration induite par ces changements pousse le MIT et Bell Labs, en 1965, à concevoir un système d'exploitation novateur, Multics (**MULT**iplexed **I**nformation and **C**omputing **S**ervice)

Ce système repose sur une approche tout à fait novatrice de l'informatique :

- ▶ Système à temps partagé
- ▶ Assemblage de briques de bases
- ▶ Assurant la compatibilité ascendante

## Multics

- ▶ Mais Multics se solde par un échec, Bell Labs se retire du projet en 1969 faute de résultats convaincants.
- ▶ Cependant, quelques ingénieurs de Bell continuent leur recherche sur ce que quelques un considèrent comme le Graal de l'informatique (Ken Thompson, Dennis Ritchie, Doug McIlroy, and J. F. Ossanna).

## Vers Unix

- ▶ Thomson pense que la recherche concernant les systèmes d'exploitation est fondamentale, et nécessaire à ses futurs travaux.
- ▶ A l'issue de réunions brainstorming, ils posent les bases d'un système de fichier qui sera la base d'Unix

## Space Travel

Tandis qu'il travaillait à la conception de son système d'exploitation, Thomson dans son temps libre programma un petit jeu : Space Travel. Le jeu fut ré implémenté avec Ritchie sur un PDP-7. Pour se faire ils programmèrent pour cet ordinateur :

- ▶ Une unité de calcul sur nombres flottants
- ▶ Des routines d'affichage
- ▶ Un debugger

## La génèse d'Unix

Thompson et Ritchie décident que leur machine fournira le laboratoire de leurs expériences, et commencent à implémenter leur système de fichier. Une fois ceci fait, ils disposent d'un système, mais il manque encore beaucoup d'outils pour en faire un système d'exploitation.

Thompson s'attèle à la tâche et se donne une semaine par outils

- ▶ Un shell
- ▶ Les outils pour copier, déplacer les fichiers
- ▶ Un éditeur de texte
- ▶ Un assembleur
- ▶ ...

# Unix

Unix était né, et tout s'accélère :

- ▶ Bell Labs commence à comprendre l'intérêt du nouveau système et finance l'achat d'un nouvel ordinateur
- ▶ Ritchie invente le C, Thompson réimplémente Unix dans un langage de haut niveau
- ▶ McIlroy et Thompson inventent et implémentent les "*pipes*"
- ▶ En 1976 les premiers cours universitaires d'Unix sont donnés à l'UCB

## Le choix

- ▶ Question pertinente, nous avons aujourd'hui le choix
- ▶ Microsoft débute en 1975, au début très loin de la problématique industrielle
- ▶ On peut parler aujourd'hui d'une "guerre" entre deux standards

Plusieurs éléments de réponse peuvent être apportés

## Conception rationnelle

- ▶ Unix a été pensé dès le début comme un système multi utilisateur
- ▶ Unix est un système réseau
- ▶ Unix a une conception modulaire
- ▶ Unix a l'avantage de l'ancienneté

## “Philosophie”

- ▶ Une philosophie ouverte dès ses débuts
- ▶ Respecte des standards
- ▶ Ouverture poussée à son paroxysme par le système GNU, lancé par Richard Stallman en 1983
- ▶ Aujourd'hui Unix est devenu synonyme de logiciel libre

# Parceque ça marche<sup>TM</sup>

Unix fonctionne, très bien et depuis longtemps.

- ▶ Une documentation abondante existe, il est facile de se former
- ▶ De très bonne implémentations opensource existent : **on sait pourquoi ça marche (ou pas).**
- ▶ Unix est à l'état de l'art des systèmes d'exploitation

## Qu'est-ce que Linux ?

- ▶ Linux a été inventé par un étudiant en 1991 (Linus Torvalds)
- ▶ Il s'agit d'un noyau, pas d'un système d'exploitation
- ▶ En conséquence, Linux n'est pas utilisable seul.

## Rappel : Qu'est-ce qu'un SE (OS) ?

Un SE remplit le rôle de couche d'abstraction entre le matériel et l'utilisateur. En gros un système d'exploitation. Deux grandes tâches sont attendues d'un système d'exploitation :

- ▶ L'aspect machine étendue
- ▶ Le gestionnaire de ressources

## Rappel : Qu'est-ce qu'un noyau ?

Le noyau est la couche la plus basse du système d'exploitation.

C'est un composant logiciel qui est le seul à parler directement au matériel

Un système composé d'un noyau seul n'est pas utilisable.

## GNU/Linux

Le véritable système d'exploitation est bien GNU, et non Linux.

C'est cependant un abus de langage fréquent.

Une distribution Linux correspond à un système d'exploitation, elle contient le noyau Linux, les outils GNU ainsi que des outils de haut niveau pour maintenir le système.

# Rôle

- ▶ Permettre
  - ▶ La cohabitation des personnes
  - ▶ Le fonctionnement des outils de travail

## **Aspect relations humaines important**

- ▶ Gérer les ressources (matérielles et logicielles)
- ▶ Mise en place du réseau, choix des configurations matérielles, maintenance, surveillance, formation . . .

## Aspect relationnel

- ▶ Trois attitudes possibles
  - ▶ Escalve
  - ▶ Tortionnaire
  - ▶ Interlocuteur
- ▶ Évidemment la meilleur attitude est la troisième. Mais
  - ▶ Nécessite du temps et de la diplomatie
  - ▶ Responsabiliser les utilisateurs
  - ▶ Justifier sa démarche en l'expliquant

## Aspect Technique

L'administrateur doit :

- ▶ Assurer la maintenance matérielle
- ▶ Assurer la maintenance logicielle
- ▶ Rester aux aguets
  - ▶ Nouveaux besoins
  - ▶ Nouvelles technologies
  - ▶ Surveillance (usages illicites)
  - ▶ Protection (Sécurité, virus ...)

## En particulier

- ▶ Gérer les utilisateurs (ouverture, fermeture de compte)
- ▶ Gérer les sauvegardes
- ▶ Gérer les licences (responsabilité légale)
- ▶ Inventorier les moyens logiciels et matériels
- ▶ S'occuper des commandes
- ▶ Gérer les imprimantes, les consommables
- ▶ Gérer les données (obsolètes, espace disque ...)

## En bref

- ▶ L'administrateur doit faire en sorte que tout fonctionne correctement
- ▶ Il s'agit d'un véritable travail demandant un grand nombre de compétences distinctes
- ▶ Rôle réellement clef de toute organisation

## Ce dont nous n'allons pas traiter

- ▶ L'aspect relationnel
- ▶ Des différences avec l'administration sous Windows
- ▶ Le dimensionnement matériel

## Ce dont nous allons parler

- ▶ Administrer une station de travail sous Unix
- ▶ Administrer un réseau de stations de travail sous Unix

Une partie théorique présentera un certain nombre de connaissances générales sur le sujet

L'accent sera donné sur la mise en pratique.

## Preliminaires

Les bonnes questions à se poser  
Choix d'une distribution

## Avant de se lancer

Comme d'habitude lorsque l'on entreprend quelque chose, la rédaction d'un cahier des charges est nécessaire.

Il sert :

- ▶ À planifier le déploiement dans le temps
- ▶ À évaluer la liste des services que doit fournir la machine
- ▶ De base de travail

Ce cahier des charges doit être validé par les utilisateurs potentiels.

## Avant de se lancer

Un certain nombre de questions doivent être posées et doivent trouver une réponse dans le CdC

- ▶ À quoi va servir cette machine ?
- ▶ À qui va-t-elle servir ?
- ▶ Sera-t-elle connectée à un réseau ?

## À qui va-t-elle servir ?

La réponse à cette question va déterminer plusieurs choses :

- ▶ Le niveau d'accessibilité de l'interface
- ▶ Le nombre d'utilisateurs potentiels ce qui influe directement sur
  - ▶ L'espace disque nécessaire
  - ▶ La technologie utilisée pour l'authentification
- ▶ La politique de sécurité

## À qui va-t-elle servir ? - Exemple 1

### Exemple

*La machine va servir aux personnes de passage à la cafétéria pour accéder à leur boîte aux lettres.*

- ▶ Politique de sécurité "niveau critique"
- ▶ Grand nombre d'utilisateurs potentiels, outils adaptés.
- ▶ Utilisateurs novices, accentuer la simplicité

## À qui va-t-elle servir ? - Exemple 2

### Exemple

*La machine va servir à un unique étudiant effectuant des travaux sur la simulation d'écoulements liquides.*

- ▶ Politique de sécurité normale
- ▶ Peu d'utilisateurs
- ▶ Utilisateur expert

## À quoi va servir cette machine ?

Cette question va servir à déterminer les besoins matériels et logiciels

Une machine destinée à fournir un terminal web ne requiert pas les mêmes moyens qu'une machine destinée à faire tourner des simulations complexes.

C'est sans doute la réflexion la plus importante à mener.

Cette phase se fait en concertation avec les utilisateurs potentiels, c'est l'expression de leur besoin.

## Exemple

### Exemple

*La machine va servir aux personnes de passage à la cafétéria pour accéder à leur boîte aux lettre.*

- ▶ Outils minimaux (Interface graphique, Firefox)
- ▶ Faible nécessité d'espace disque

## À quoi va-t-elle servir ? - Exemple 2

### Exemple

*La machine va servir à un unique étudiant effectuant des travaux sur la simulation d'écoulements liquides.*

- ▶ Outils de développement
- ▶ Grand besoin d'espace disque

## Sera-t-elle connectée à un réseau ?

Cette question détermine directement la politique de sécurité. De plus elle complète les questions précédentes car un certain nombre de services tournent déjà très certainement sur le réseau

- ▶ Authentification
- ▶ Espace disque
- ▶ Accès à Internet
- ▶ ...

## Unix sans réseau ...

En fait la question précédente n'en est pas vraiment une, car un système Unix sans accès réseau n'a pas réellement de sens.

La vraie question à se poser est à quel type de réseau est-on connecté, et le niveau d'hostilité de ce réseau.

## Une distribution

Une distribution est un ensemble complet de logiciels permettant d'utiliser GNU/Linux

Un grand nombre de distributions existent (plusieurs centaines!)

Il n'y a cependant que 10 majeures

## Qu'est-ce qui différencie les distributions ?

Mis à part le logo et le nom...

- ▶ le système de package
- ▶ les outils de configuration
- ▶ le packaging
- ▶ parfois, l'utilité

## Question utilisation

Il y a deux grand types de distributions :

- ▶ Les distributions live
- ▶ Les distributions à installer

## Distribution live

Les distributions “live” sont souvent fournies sous la forme de cdrom bootables (live CD). Tout le système est sur le cdrom, et le disque dur des machines n'est pas touché.

Avantages :

- ▶ Permet d'utiliser linux ponctuellement, sans avoir à réinstaller son poste windows
- ▶ L'intégrité du système est garantie en toute circonstances

Inconvénients :

- ▶ Personnalisation difficile
- ▶ Toutes les modifications sont perdues !

## Distribution normales

Les distributions normales sont des systèmes complets qu'il faut installer sur le disque dur de son ordinateur. Avantages :

- ▶ Beaucoup plus performant
- ▶ Facile à personnaliser

Inconvénients :

- ▶ Une erreur peut rendre le système inexploitable
- ▶ Les changements sont difficiles

## Spécificités

- ▶ Certaines distributions sont spécifiques à un usage donné
  - ▶ Utilisation sur vieilles machines (peu de RAM, peu d'espace disque)
  - ▶ Calcul scientifique, calcul distribué. . .
  - ▶ Utilisation sur ordinateurs spécifiques (Macintosh, stations SUN, . . .)
- ▶ En fait il est possible d'arriver au même résultat quelque soit la distribution
- ▶ Mais on peu gagner du temps en choisissant la bonne distribution

## Outils de configuration

- ▶ Rien n'est standard en terme d'outils de configuration
- ▶ A charge de chaque distribution d'en développer, ou non.
- ▶ Il est toujours possible d'éditer des fichiers de configuration quelque soit la distribution, mais l'emplacement de ces fichiers est susceptible de changer de l'un à l'autre

### Exemple

*La configuration du réseau se fait dans `/etc/network/interface` sous Debian, et est située quelque part dans `/etc/sysconfig` sous Fedora.*

# Packages

La notion de package est importante dans le monde Unix.

Un package est une archive contenant l'ensemble des fichiers nécessaires au fonctionnement d'un programme donné.

Mais un package contient aussi d'autres informations comme la liste des dépendances, c'est à dire les autres logiciels dont a besoin le programme pour fonctionner.

## Packages - plusieurs standards

Il existe plusieurs standard de packages :

- ▶ RPM (RedHat, Fedora, Mandriva, ...)
- ▶ DEB (Debian, Ubuntu, Progeny, ...)
- ▶ TGZ (Slackware)
- ▶ sources (Gentoo)

Il n'y a pas de différence fondamentale entre un `.rpm`, un `.deb` et un `.tgz`

En revanche les distribution basées sur les sources ont une particularité intéressante.

## Distributions basées sur les sources

- ▶ Dans la plupart des distribution les paquets contiennent des logiciels déjà compilés, et prêt à être copiés sur le disque dur.
- ▶ Le problème est que l'architecture des ordinateurs est loin d'être uniforme.
- ▶ Les distributions sources fonctionnent différemment.
- ▶ Dans ce type de distributions les paquets contiennent les sources du programme à installer
- ▶ Le programme est compilé au moment de son installation.
- ▶ Ceci permet de prendre en compte les spécificités du matériel
- ▶ Le prix à payer est le temps de la compilation

## Bilan

- ▶ Le choix de la bonne distribution pour le bon usage est important, mais pas capital
- ▶ La plupart sont équivalentes
- ▶ À catégorie d'utilisation donnée, le choix est sentimental plus que raisonné

# Bilan

Pour la suite de ce cours nous travaillerons avec :

- ▶ Debian pour tout ce qui est aspect serveur
- ▶ Ubuntu pour les postes clients
- ▶ Knoppix comme système live