

Systèmes d'exploitation

Cours – introduction – informatique,
E2I, EPU
Dimitri Galayko

1

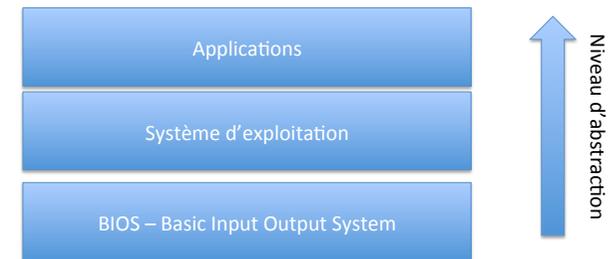
Sommaire

- Système d'exploitation – généralités
- Introduction au linux
 - Principes
 - Utilisateurs, droits
 - Système de fichiers
 - Interface graphique, distributions
 - Outils spécifiques
- Linux versus Windows

2

Logiciel : les couches

- 3 couches logicielles à distinguer :
 - BIOS: se trouve dans la mémoire morte de l'ordinateur, sert à faire démarrer le système d'exploitation et à configurer l'ordinateur
 - Système d'exploitation: une couche permettant de manipuler les fichiers et les principales ressources de l'ordinateur par l'utilisateur et surtout par les applications
 - Les applications : les programmes ayant une utilité finale.



3

BIOS : Démarrage de l'ordinateur

- Initialiser/identifier les composantes de la carte mère
- Modifier l'heure de l'ordinateur
- L'amorçage : Power-on Self Test, puis recherche d'un système d'exploitation
- L'ordre d'amorçage : à partir de quelle périphérique démarrer le système (disque dur, clé USB, DVD, réseau)
- Pour entrer dans le bios: appuie de la touche Del au démarrage de l'ordinateur
- Protection par mot de passe : impossibilité de faire démarrer l'ordinateur autrement que par le moyen prévu

4

Système d'exploitation

- Nécessaire lorsqu'il y a plusieurs programmes à lancer (téléphones mobiles, consoles de jeu primitifs: pas de SE)
- Gestion des ressources de l'ordinateur (disques dur, mémoire, affichage)
- Réalisation des fonctions « simples » mais néanmoins complexes et répétitives (saisi sur le clavier, affichage d'un texte, copie de fichiers, etc.)
- Interface avec utilisateur
- Gestion des utilisateurs, gestion des programmes
- Fonctionnement multi-tâches: depuis les grands calculateurs, afin d'optimiser l'utilisation des ressources
- Système de stockage: les fichiers

5

UNIX : raison du succès

- Bases simples: il y a deux notions clé, « fichier » et « processus »
- Architecture modulaire: un programme=une fonction, moyens d'assemblage très puissants (Shell)
- Le noyau minimaliste
- Le système est écrit en langage C, donc, indépendant de l'architecture du matériel
- Dès le départ, le système est orienté pour un travail en réseau: avantage très important dès 1990 (Internet, réseaux locaux...)
- Multi-utilisateur: protection, sécurité

7

Système Unix : la genèse

- Début des 60s: plusieurs systèmes d'exploitations, plus d'une centaine
- Multics (1970) : compatible avec systèmes multiprocesseur
- 1969: Unix – a d'abord été adopté par les universitaires puis par l'industrie
- Fin des 70s : BSD = Berkley Systems Distribution, support de TCP/IP
- Familles des OS standardisés et ouverts: Free BSD, Open BSD, GNU/Linux
- GNU = GNU Not Unix (équation à récurrence infinie, une blague d'informaticiens)

6

Les fichiers dans l'Unix

- Historiquement, un fichier est une collection de données binaires arrangées sous forme d'une liste (l'ordre est important).
- Par exemple, on peut imaginer un fichier qui stocke les 100 premiers chiffres du nombre Pi
- Les fichiers sont traditionnellement associés au stockage sur un disque dur/disquette/ bande magnétique

8

Fichiers sous Unix

- Sous Unix, « tout ce qui n'est pas un fichier est un processus »
- C'est à dire, que la notion du fichier est bien plus large que sous Windows
- Le disque dur, un répertoire, le clavier, la souris: tout est fichier
- On parle également d'un flux [de données]
- Tous les fichiers sont organisés dans une seule arborescence (avec une seule racine), contrairement à Windows où les racines correspondent aux partitions des disques durs (C:, D:, ...)
- La racine s'appelle « root » et se désigne /

9

Types de fichiers

- Fichiers réguliers: collection des bits dans le sens de Windows
- Répertoires: une liste de fichiers
- Raccourcis (liens vers d'autres fichiers)
- Fichiers spéciaux : représentent les dispositifs (port USB, clavier, souris, disques durs, lecteurs DVD, etc.)

10

Organisation de données sur un disque dur sous Linux

- Un disque dur: est partitionné
- La raison: séparer les données d'utilisateur des données systèmes/programmes afin de les préserver en cas du crash du système
- Exemple: l'utilisateur ou un programme (virus?) remplit « bêtement » l'espace disque: le système ne peut plus fonctionner s'il y a qu'une seule partition
- Une partition spécifique pour la mémoire virtuelle (*Swap*), 1-2 GO
- Une partition pour les programmes (min. 10 GO)
- Une ou plusieurs partitions pour les données – aussi grandes que nécessaire

11

L'arborescence des fichiers systèmes de Linux: un « standard »

- /usr : programmes installés par l'utilisateur
- /home : les données des utilisateurs
- /etc : fichiers texte de configuration
- /mnt : les dispositifs éjectables (clés USB, CD, DVD)
- /tmp : fichiers temporaires
- /dev : les dispositifs d'entrée-sortie
- /bin : programmes de système (ex., manipulation fichiers)
- /boot : programmes de démarrage du système

Plus d'infos : http://www.tldp.org/LDP/intro-linux/html/sect_03_01.html

Fichiers spéciaux: . – le répertoire courant, .. – le répertoire parent, / - la racine

12

Les utilisateurs et les droits d'accès

- Le travail multi-utilisateur et multi-tâches: la principale force de Unix
- Les utilisateurs sont organisés en groupes
- Un utilisateur: une identité « virtuelle » qui peut correspondre à une personne réelle ou à un logiciel/processus
- Chaque fichier appartient à un groupe et à un utilisateur (faisant partie de ce groupe)
- Système de gestion fine des droits

13

Les utilisateurs et les droits d'accès

- Vis-à-vis de chaque fichier, il y a 3 types d'utilisateurs:
 - Le propriétaire du fichier
 - Les membres du groupe du fichier
 - Les autres utilisateurs
- Les droits de lecture, d'écriture et d'exécution sont définis individuellement pour chaque catégorie.
- Ca rend possible une gestion très fine des ressources, de l'accès et de la sécurité
- Par exemple, un étudiant « bêta » n'a aucun moyen d'abimer le système informatique installé sur l'ordinateur de l'école. Au pire, il supprimera tous les fichiers de son compte. (On ne parle pas des hackers !)

14

Les droits des fichiers

- Exemple de réponse de la commande ls -l (lister le contenu du répertoire courant)
- Tous les fichiers appartient à l'utilisateur galayko et au groupe galayko
- Ensuite vient la taille en octet, la date de modification et puis le nom

```
[marazli:~/Documents/tmp] galayko% ls -lt
total 30320
-rw-r--r--@ 1 galayko  galayko  443516 Mar  2  2013 asmodeus.epub
-rwxr-xr-x  1 galayko  galayko   13839 Jun 26  2010 aineptepub.pyw
-rwxr-xr-x  1 galayko  galayko    8504 Jun 26  2010 ineptkey.pyw
-rwxr-xr-x  1 galayko  galayko 14536192 Jun 26  2010 python-2.6.2.msi
-rwxr-xr-x  1 galayko  galayko  511069 Jan  9  2010 pycrypto-2.1.0.win32-py2.6.exe
[marazli:~/Documents/tmp] galayko%
```

15

Les droits des fichiers

- A gauche, vous avez des traits '-' et des lettres r, w et x: r=read, w=write, x=execute qui donne le type et les droits associés au fichier
- Première position: le type du fichier (d pour répertoire, - pour fichier régulier)
- Deuxième, troisième, quatrième position: les droits à la lecture, écriture, exécution, dans l'ordre pour l'utilisateur
- '-' signifie « pas de droit », r, w ou x signifie que l'utilisateur a le droit de lire, écrire ou exécuter
- Positions 4-7, 8-10 : la même chose pour les autres membres du groupe, et pour tous les autres membres.

```
[marazli:~/Documents/tmp] galayko% ls -lt
total 30320
-rw-r--r--@ 1 galayko  galayko  443516 Mar  2  2013 asmodeus.epub
-rwxr-xr-x  1 galayko  galayko   13839 Jun 26  2010 aineptepub.pyw
-rwxr-xr-x  1 galayko  galayko    8504 Jun 26  2010 ineptkey.pyw
-rwxr-xr-x  1 galayko  galayko 14536192 Jun 26  2010 python-2.6.2.msi
-rwxr-xr-x  1 galayko  galayko  511069 Jan  9  2010 pycrypto-2.1.0.win32-py2.6.exe
[marazli:~/Documents/tmp] galayko%
```

16

Les droits des fichiers

- Seul l'utilisateur qui possède le fichier peut modifier les droits du fichier
- Mais également il y a un super-utilisateur : le *root*
- Il peut tout faire dans le système: tout modifier, tout lire, tout supprimer
- L'utilisateur *root* doit uniquement entrer dans le système pour effectuer les tâches d'administration, jamais pour travailler
- Le *root*: un grand pouvoir, de grandes responsabilités

17

Les *shell* : automatisation des manipulations système

- Une *shell* : un outil permettant d'écrire et d'exécuter les scripts afin d'automatiser les actions système
- Par exemple: on souhaite renommer tous les fichiers .txt en .dat
- Le script peut être le suivant :

```
#!/bin/tcsh
foreach file (*.txt)
    set newname=`echo $file | sed -e "s/txt/dat/"`
    mv $file $newname
end
```

18

Le travail en terminal

- Sous linux, le mode naturel de travail est d'utiliser un terminal
- Pourquoi ?
 - La configuration du système se fait par des fichiers texte
 - Une très grande modularité des commandes : un commande=un processus=un programme
- Le terminal : une interface d'utilisateur qui permet de saisir les commandes en version texte. Un grand avantage pour le travail en réseau.
- Un grand nombre d'outils très puissants mais les utilisateurs sont « gâtés » par Windows, par la souris et par les interfaces « intuitives ».
- Seuls les utilisateurs professionnels gagnent à utiliser un terminal
- Très différent de Windows, où le terminal est exclusivement réservé aux tâches d'administration
- Les TPs: le travail avec le terminal

19

Le linux aujourd'hui

- Unix a traditionnellement été réservé aux utilisateurs professionnels (par ex., pas d'interface graphique)
- De nos jours, de progrès substantiels ont été fait pour rendre Linux accessible au grand public.
- Les distributions: Ubuntu, Scientific Linux, Fedora, ...
- Une distribution: un paquet de logiciels libres et gratuits, pas seulement le système d'exploitation (bureautique, outils scientifiques, outils graphiques, etc).
- L'installation de linux est incroyablement simple (essayez ! Au moins sur une machine virtuelle)
- Interface graphique de plus en plus évoluée, et on peut choisir son interface, contrairement à Windows (les plus répandus, Ubuntu, KDE et GNOME)
- Une suite bureautique Libre Office équivalent à Microsoft Office
- Deux avantages imbattables : **Linux ne souffre pas de virus et Linux est gratuit !!**

20

Le programme pour les TPs

- Prise en main de l'environnement linux,
- Travail sous terminal
- Edition de texte version *geek* : l'éditeur *vim*
- Connaissance de shell : éléments de programmation (avancé)