

## Le système d'exploitation GNU/Linux

Nicolas Burrus <nicolas.burrus@laposte.net>



- Gestion des droits
- Gestion des processus

26 Avril 2005

## Système multi-utilisateurs ?

### Principes

- Plusieurs "personnes" différentes peuvent utiliser la machine
- Le système gère chacune des identités de manière indépendante
- Protection des informations des utilisateurs

### Intérêts

- Personnalisation du système
- Confidentialité
- Sécurité : évite les erreurs de manipulation
  - Ex: un utilisateur efface par erreur les fichiers d'un autre

## Contenu du cours

- 1 Introduction
- 2 Gestion des droits des fichiers
- 3 Gestion des processus
- 4 TP

## Notion de système multi-tâches

### Principes

- Plusieurs programmes peuvent tourner indépendamment
- Chaque programme a l'impression d'être seul
- Chaque programme a sa propre zone mémoire
- Le système assure que tous les programmes s'exécutent

### Intérêts

- Utilise au mieux les ressources de la machine
- Permet à plusieurs utilisateurs de lancer des programmes
- Robustesse : les programmes sont contrôlés par le système

## Utilisateurs et groupes sous Linux

### Chaque utilisateur a :

- Un login (nom d'utilisateur)
- Un password (mot de passe)
- Un identificateur entier unique (UID)
- Un répertoire "home"
- Stockés dans le fichier /etc/passwd
- Un groupe principal
- Il existe un utilisateur privilégié : root

### Les utilisateurs peuvent appartenir à des groupes :

- Un groupe est défini par son nom
- A chaque groupe est associé un entier unique (GID)
- Chaque groupe contient une liste d'utilisateurs
- Stockés dans le fichier /etc/group

## Commandes de base

[2/2]

**id** [username]

Affiche l'UID, le GID et les groupes de l'utilisateur courant ou de username

**groups** [username]

Liste les noms des groupes de l'utilisateur courant ou de username

**whoami**

Affiche le nom de l'utilisateur courant

## Commandes de base

[1/2]

**su** [login]

Change son identité en login (root si non spécifié)

**adduser** username

Ajoute l'utilisateur username (en root)

**deluser** username

Supprime l'utilisateur username (en root)

**usermod** [options] username

Modifie des informations sur l'utilisateur username (en root)

**addgroup** groupname

Ajoute le groupe groupname (en root)

**delgroup** groupname

Supprime le groupe groupname (en root)

## Droits des fichiers

### A chaque fichier est associé :

- Un utilisateur propriétaire
- Un groupe propriétaire
- Une politique de droits

### Politique de droits

- 3 catégories de droits:
  - Accès en lecture (read)
  - Accès en écriture (write)
  - Accès en exécution (execute)
- 3 classes d'application de ces droits :
  - L'utilisateur propriétaire
  - Le groupe propriétaire
  - Les autres utilisateurs

## Politique de droits

### Utilisation de masques numériques :

- 1 → exécution
- 2 → écriture
- 4 → lecture
- Ex: 5 → exécution + lecture
- Ex: 6 → écriture + lecture

### Pour chaque fichier :

- Un masque numérique par classe d'application
  - Utilisateur propriétaire
  - Groupe propriétaire
  - Autres utilisateurs
- Ex: 640
  - Lecture + écriture pour l'utilisateur propriétaire
  - Lecture pour le groupe propriétaire
  - Rien pour les autres

## Principes

### A chaque processus est associé :

- Un utilisateur propriétaire
- Un groupe propriétaire
- Un entier : PID (process ID)
- Une priorité entre -20 et 20 (ordre décroissant)
- Un processus parent (processus racine : `init`)
- Chaque processus a les droits de son propriétaire

## Commandes utiles

**chown** username file1 [file2] ...

Change le propriétaire des fileN en username (en root)

**chgrp** groupname file1 [file2] ...

Change le groupe propriétaire des fileN en groupname

**chmod** mode file1 [file2] ...

Change les droits des fileN selon le masque mode

Ex: chmod 644 file1

Le mode peut aussi être exprimé de façon symbolique (man chmod)

**ls -l** [file1] [file2] ...

Affiche entre autres l'utilisateur propriétaire, le groupe propriétaire et la politique de droits des fichiers. Exemple pour `/bin/chmod` :

```
-rwxr-xr-x 1 root root 31212 Jul 16 2004 /bin/chmod
```

## Lancement d'un programme

### Pour lancer un programme :

- L'utilisateur doit avoir les droits d'exécution dessus
- Par défaut, propriétaire du processus = lanceur du processus
- Exception : le `setuid` / `setgid` bit sur le fichier
  - `Setuid` (valeur 4) : le propriétaire du processus devient le propriétaire du fichier
  - `Setgid` (valeur 2), le groupe propriétaire du processus est le groupe propriétaire du fichier
  - Exemple: `chmod 4755 /bin/su`

## Lister les processus

### **top**

Liste en temps réel les processus actifs

Liste également l'occupation mémoire et processeur

### **ps**

Liste les processus

Options courantes : `ps aux`

`ps axo pid,user,command`

### **pstree**

Affiche l'arbre des processus

### **pidof process**

Donne le PID associé au processus `process`

## Les signaux

Les processus peuvent émettre et recevoir des signaux :

- SIGINT : dit au processus de s'arrêter (Ctrl-C)
- SIGSTOP : dit au processus de se mettre en pause
- SIGCONT : lui dit de reprendre
- SIGTERM : dit au processus de se terminer
- SIGKILL : tue le processus
- ...

## Manipulation des priorités

**nice** [priority] command

Lance command avec la priorité `priority` (10 par défaut)

**renice** priority PID

Change la priorité du processus ayant le PID `pid`

**ps axo nice,command**

Affiche la priorité des processus

## Commandes utiles

**kill** [-signal] PID

Envoie signal au processus ayant le PID `pid`

Ex: `kill -TERM 1234`

Ex: `kill -9 1234`

**killall** [-signal] command

Envoie signal aux processus instances de la commande `command`.

## Shell et signaux

### Le shell peut envoyer des signaux :

- Ctrl - C : SIGINT au processus actif
- Ctrl - Z : envoie SIGSTOP au processus actif et le déconnecte du terminal

### jobs

Liste les processus lancés par le shell en cours

### fg [%id]

Envoie SIGCONT au job `id` ou au dernier job stoppé et le reconnecte au terminal

### bg [%id]

Envoie SIGCONT au job `id` ou au dernier job stoppé sans le reconnecter au terminal

## Astuce

`command &`

Lance la commande `command` et la déconnecte du terminal.

<http://nburrus.objectis.net/teaching/linux/>