Unix : Utilisation et programmation

Pr. Hajar IGUER
hajar.iguer@uic.ac.ma

Unix®
Operating System

PROTECTION DES FICHIERS

Introduction

- Linux est un système multi-utilisateurs
- On a donc besoin de créer plusieurs utilisateurs pour pouvoir exploiter le système d'exploitation
- D'où l'intérêt de gérer les utilisateurs, leurs droits d'accès et leur groupes.
- Les fichiers importants à connaître sont :
  - le fichier **/etc/passwd**
  - le fichier **/etc/group**

Fichier /etc/passwd

- Le fichier /etc/passwd contient toutes les informations relatives aux utilisateurs (login, mots de passe, ...).
- Seul le superutilisateur (root) doit pouvoir le modifier.
- Les droits de ce fichier doivent être en lecture seule pour les autres utilisateurs.

Fichier /etc/passwd

- **Extrait du fichier :**
  - root:x:0:0:root:/root:/bin/bash
  - bin:x:1:1:bin:/bin:
  - daemon:x:2:2:daemon:/sbin:
  - adm:x:3:4:adm:/var/adm:
  - lp:x:4:7:lp:/var/spool/lpd:
  - sync:x:5:0:sync:/sbin:/bin/sync
  - shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
  - halt:x:7:0:halt:/sbin:/sbin/halt

Fichier /etc/passwd

- Les champs :
  - Nom utilisateur
  - Mot de passe : codé, ou x donc dans /etc/shadow, ou !! Interdit de connexion
  - UID
  - GID
  - Libre : nom et prénom, adresse…
  - Le répertoire de connexion
  - Le shell

Fichier /etc/group

- Le fichier /etc/group contient la liste des utilisateurs appartenant aux différents groupes.
- Plusieurs utilisateurs peuvent accéder à plusieurs fichiers/répertoires ainsi ils sont regroupés dans une seule structure appelée groupe.

Fichier /etc/group

- Organisé en champs :
  - Le nom du groupe
  - Mot de passe (non utilisé)
  - Le numéro du groupe
  - Champs vide
  - Champs description
  - Liste des membres

CONNEXION ET DÉCONNEXION

Commandes

- logname, whoami: nom de l'utilisateur
- id: donne les informations sur l'utilisateur actuel
- uname: donne des informations sur le SE
- who: donne la liste des utilisateurs sur la machine
- passwd: modification du mot de passe

Exécuter en mode root

- Le root est désigné comme le superutilisateur de toute distribution Linux cela veut dire qu'il a tous les droits sur la machine.
- Tout autre utilisateur créé, en même temps que sa session, détient des droits limités.
- C'est un des volets de sécurité pour lequel est connu Linux.

Exécuter en mode root

- Afin d'exécuter en mode root:
  - Utilisez la commande sudo
- Afin d'exécuter en mode root et le rester:
  - Utilisez la commande sudo –su
- Notez que:
  - sudo: **s**ubstitute **u**ser **do**
  - **le mot de passe de votre session est le même que celui du superutilisateur**
  - **pour quitter le mode root, saisissez « quit » ou** **Ctrl + D**

Changer de session

- Pour changer d'utilisateur/ de session, on utilise la commande su [nom_user]
- En utilisant la commande su - , le "-" force le démarrage d'un nouveau shell de connexion, avec ajustement des variables d'environnement et de tous les réglages par défaut selon les préférences de l'utilisateur.
- Exemple: su Ginfo

GESTION DES UTILISATEURS

Gestion des utilisateurs

- Seul le superutilisateur peut gérer les autres utilisateurs
  - Pour ajouter un utilisateur: sudo adduser ginfo
  - Changer le mot de passe: passwd ginfo
  - Supprimer un compte: deluser - -remove-home ginfo

Gestion des groupes

- Tout utilisateur doit appartenir à un groupe
- Lors de la création d'un utilisateur, il est automatiquement affecté à un groupe du même nom que l'utilisateur.
  - Pour créer un groupe: addgroup
  - Pour modifier un utilisateur: usermod
    - **-l** : renomme l'utilisateur (le nom de son répertoire personnel ne sera pas changé par contre) ;
    - **-g** : change de groupe.

Gestion des groupes

- Tout utilisateur doit appartenir à un groupe
- Lors de la création d'un utilisateur, il est automatiquement affecté à un groupe du même nom que l'utilisateur.
  - Pour créer un groupe: addgroup
  - Pour supprimer un groupe : delgroup
  - Pour modifier un utilisateur: usermod
    - **-l** : renomme l'utilisateur (le nom de son répertoire personnel ne sera pas changé par contre) ;
    - **-g** : change de groupe.
    - -**aG :** rajoute un groupe aux groupes déjà associés
  - Pour afficher la liste des groupes: groups

Gestion des groupes

- Exemples:
  - usermod –g groupe1 ginfo
  - addgroup group1
  - delgroup group2
  - groups ginfo

Propriétaire du fichier : modifier

- Pour changer le propriétaire d'un fichier
  - chown user nom_fich
- Pour changer le groupe propriétaire d'un fichier
  - chgrp group1 nom_fich
  - chown ginfo:group1 nom_fich (affectation double de l'utilisateur et du groupe)
  - chown –R : Affectera récursivement tous les sous dossiers à un utilisateur

DROITS D'ACCES

Droits : Concept

- Pour chaque fichier, il existe trois types d'utilisateurs:
  - Propriétaire du fichier
  - Les membres du groupe propriétaire du fichier
  - Les autres utilisateurs du système
- Chaque fichier est désigné par 10 attributs:
  - \- rwx rwx rwx
  - d r-x r-x r--

Droits: Signification

- **Droits:**
  - Lecture (r)
  - Ecriture (w)
  - Exécution (x)
- Utilisateur (u)
- Groupe (g)
- Autre (o)

Droits: Exemple

- c rw- r– r--: fichier spécial caractère/ lecture et écriture pour l'utilisateur/ lecture pour le groupe / lecture pour les autres.
- d rwx rw- r--:dossier/ lecture, écriture et exécution permises pour le propriétaire/ lecture et écriture pour le groupe/ seulement lecture pour les autres
- - rw- rw- rw- : fichier ordinaire/lecture et écriture pour l'utilisateur, le groupe et les autres

Affichage des informations

- Affichage des informations d'un fichier:
  - ls –l nom_fichier
  - ls –l chemin_fichier

- Affichage des informations d'un répertoire:
  - ls –dl nom_repertoire
  - ls –dl chemin_repertoire

Modification des droits d'accès aux fichiers

- Mode d'utilisation: Protection par un nombre octal
  - chmod [nb_octal] <liste_fichiers>
- **Exemple:**
  - rwx rw- r-x sera représenté par un nombre octal. Pour faire le calcul, on remplace une lettre par 1 et un tiret par 0. La représentation sera 111 110 101= 765
  - chmod 765 fich

Modification des droits d'accès aux fichiers

- Mode d'utilisation: Mode symbolique
  - chmod [who] op [droit] <liste_fichiers>
- **who?**
  - U(ser)
  - G(roup)
  - O(ther)
- **op?**
  - = force les droits
  - +ajoute les droits
  - -retire les droits

Modification des droits d'accès aux fichiers

- Exemples:
  - chmod u-w fich: supprime le droit d'écriture au propriétaire.
  - chmod g+r fich: rajoute le droit de lecture au groupe.
  - chmod ug=x fich: accès uniquement en exécution pour le propriétaire et le groupe / Aucune modification pour les autres.
  - chmod u=rwx,g=r,o=- fich : tous les accès pour l'utilisateur/ la lecture pour le groupe/ aucun droit pour les autres

Droits aux dossiers: Interprétation

- r: autorise la lecture du contenu du répertoire et permet de voir la liste des fichiers
- x: autorise l'accès au répertoire (à l'aide de la commande cd)
- w: autorise la création, la suppression et le changement du nom d'un élément du répertoire. Cette permission reste indépendante de l'accès aux fichiers dans le répertoire

MÉCANISMES DE REDIRECTION/ TUBES/ FILTRES

Structure d'une commande

- Toute commande utilisée dans le système d'exploitation linux est structurée de la manière suivante:

Mécanisme de redirection

- Il est possible de rediriger le résultat d'une commande et d'effectuer des enchainements de commandes.
- Cette redirection se fait dans un fichier ou en entrée d'une autre commande

Mécanisme de redirection

- Nous présentons deux flux de redirection dans les fichiers
  - > fich1: redirige dans un fichier et l'écrase s'il existe déjà sans demande de confirmation
  - >> fich1 : redirige à la fin d'un fichier et le crée s'il n'existe pas

- NB: ces symboles spéciaux sont appelés flux de redirection

Mécanisme de redirection

- Une commande exécutée peut parfois générer des erreurs.
- Par défaut la redirection se fait en entier dans la sortie standard (fichier).
- Il y a une possibilité de séparer les résultats dans deux fichiers différents (deux sorties)

Mécanisme de redirection

- Pour cela on utilise le symbole suivant:
  - 2> fichier : redirige les erreurs éventuelles dans le fichier ,il sera écrasé s'il existe déjà.
  - 2>> fichier : redirige les erreurs éventuelles à la fin d'un fichier et le crée s'il n'existe pas.
  - 2>&1 fichier : redirige les erreurs éventuelles dans le même fichier que la sortie standard.

Mécanisme de redirection

- Nous présentons deux flux de redirection qui permettent de lire depuis un fichier
  - < fich1 : envoie le contenu d'un fichier à une commande
  - << fich1 : permet de lire une entrée progressivement depuis le clavier ligne par ligne
- NB: ce symbole < indique l'entrée envoyée à la commande

Mécanisme de redirection

- Exemples:
  - commande > fich1.txt
  - commande >> fich1.txt
  - commande > fich1.txt 2> erreurs.txt
  - commande > fich1.txt 2>> erreurs.txt
  - commande > fich.txt 2>&1
  - commande < fich.txt
  - commande << FIN

Exercice

- En utilisant les redirections, réalisez les manipulations suivantes:
  - Copie d'un fichier dans un nouveau fichier
  - Concaténation de deux fichiers
  - Ajout du contenu d'un fichier dans un autre
  - Créer un fichier par saisie au clavier

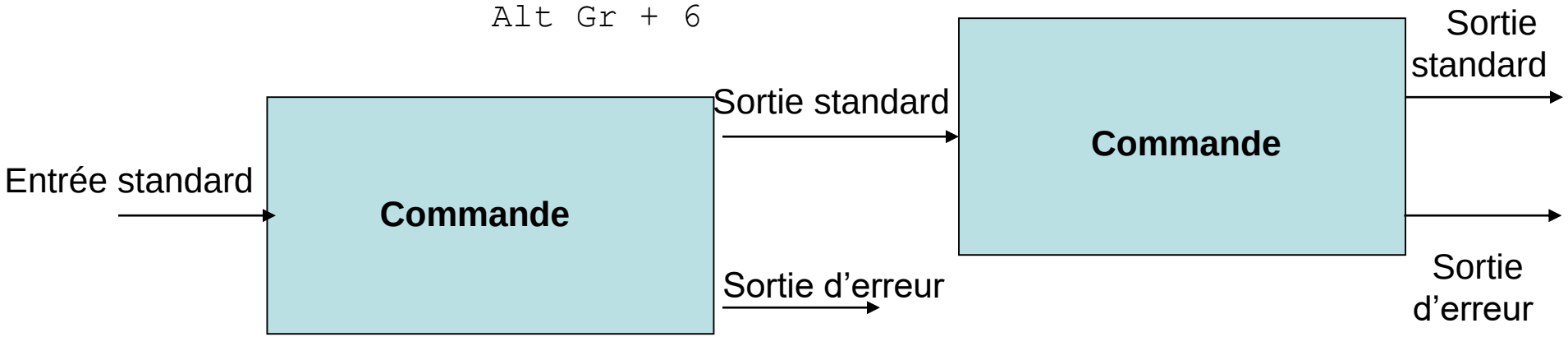
Correction

- Copie d'un fichier dans un nouveau fichier
  - cat info.txt > test.txt
  - cat <info.txt > test.txt
- Concaténation de deux fichiers
  - cat info.txt info2.txt > test.txt
  - cat <info.txt info2.txt > test.txt
- Ajout du contenu d'un fichier dans un autre
  - cat info.txt info2.txt >> info3.txt
- Créer un fichier par saisie au clavier
  - Cat > test3.txt
  - 1
  - 2
  - 3
  - CTRL + d

Tubes

- Un tube est un flot de données qui permet de relier la sortie standard d'une commande à l'entrée standard d'une autre commande.
- commande1 | commande2 | commande3

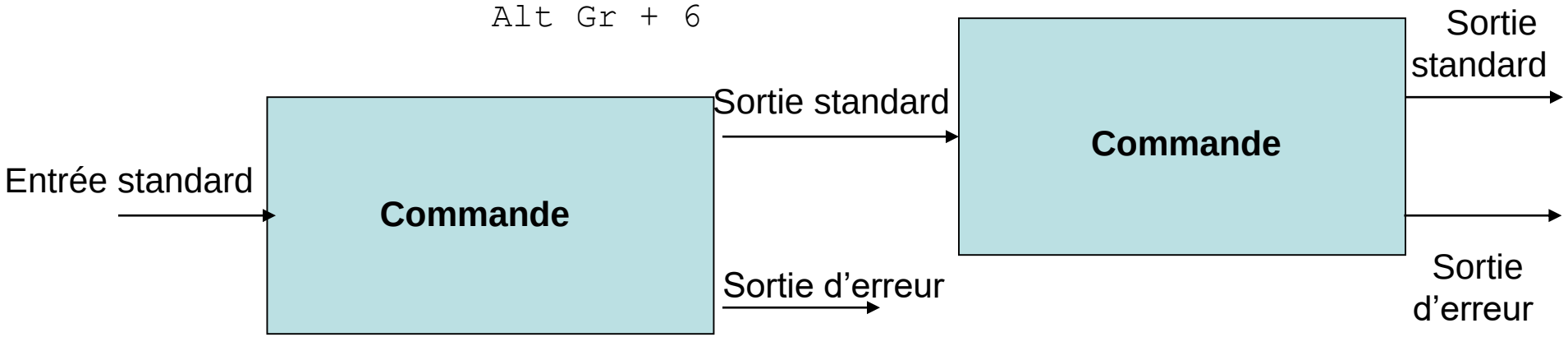
Alt Gr + 6



Tubes

- Un tube est un flot de données qui permet de relier la sortie standard d'une commande à l'entrée standard d'une autre commande.
- commande1 | commande2 | commande3

Alt Gr + 6



Filtres

- C'est une commande qui lit les données sur l'entrée standard, les traite et les écrit sur la sortie standard.
- Les filtres les plus utilisés:
  - grep: recherche les occurrences d'une chaine de caractère
  - wc: word count compte le nombre de caractère
  - less: affiche le contenu d'un fichier page par page
  - sort: filtre de tri

Filtres

- Exemple:
  - cat /etc/passwd | grep /bin/bash | wc –l

- Quelles commandes Linux exécuter pour obtenir à l'écran: Il y a xxx utilisateurs de ce système dont le login shell est bash?
  - cat /etc/passwd | wc -l