



Institut National Polytechnique

Félix HOUPHOUET - BOIGNY

# Système d'exploitation Linux

## Module 2



# Partie 1:

# Archivage et restauration des fichiers

# I. Archivage et Restauration

La sauvegarde est un travail important de l'administrateur puisqu'en cas de gros problèmes, on passe généralement par une restauration lorsque celle-ci est encore intègre (bon fonctionnement, pas de corruption). Chaque Unix est fourni avec des commandes et des procédures de sauvegarde qui lui sont propres. On distingue tout de même quelques outils communs.

Pour la sauvegarde de fichiers et d'arborescences, utilisez les commandes **tar**. Les programmes **cpio** et **pax** peuvent aussi archiver en utilisant des redirections (< ou >)

Il est aussi très important de définir un plan de sauvegarde, en se posant les bonnes questions :

- Que faut-il sauvegarder ?
- Avec quelle fréquence ?
- La sauvegarde doit être automatique ou manuelle ?
- Quelle est la méthode de sauvegarde la plus appropriée ?

# I-1 Archivage

## Définition

**Archive:** regroupement de plusieurs fichiers froids

## I-2 Outil d'archivage tar

Pour la sauvegarde de fichiers et d'arborescences, utilisez les commandes **tar**.

## I-3 Syntaxe

**tar -cvf** nom\_archive.tar nom\_dossier/



Utilitaire d'archive    Fichier archive    dossier à archiver

- c : signifie créer une archive tar ;
- v : signifie afficher le détail des opérations ;
- f : signifie assembler l'archive dans un fichier.

## **I.2 Compression**

La compression consiste à réduire la taille d'un fichier

## **I.3 les outils de compression (gzip ou bzip2)**

**Ex:**

```
gzip mon_archive.tar
```

```
bzip2 mon_archive.tar
```

## **I.4 les outils de décompression gunzip ou bunzip2**

**Ex:**

```
gunzip mon_archive.tar.gz
```

```
bunzip2 mon_archive.tar.bz2
```

## I.4 Lister

La syntaxe est :

```
tar tvf nom_archive.tar
```

## I.5 Restauration

Pour restaurer le contenu d'une archive la syntaxe est :

```
tar xvf nom_archive.tar
```

Le paramètre x permet l'extraction de l'ensemble des fichiers de l'archive, ou du ou des fichiers spécifiés à la suite du nom de l'archive.

# Partie 2: Les redirections

# I. Les redirections

## 1. Principe

Les redirections sont l'une des plus importantes possibilités offertes par le Shell. Par redirection, on entend la possibilité de rediriger l'affichage de l'écran:

- vers un fichier ;
- une imprimante ;
- tout autre périphérique;
- les messages d'erreur vers un autre fichier ;
- de remplacer la saisie clavier par le contenu d'un fichier ;

Par défaut le canal d'entrée est le clavier, et le canal de sortie, l'écran. Un troisième canal, le canal d'erreur est aussi redirigé vers l'écran par défaut. Il est possible de rediriger ces canaux vers des fichiers.



Par défaut,

**a.** L'entrée standard est le clavier. C'est sur le clavier que les commandes sont entrées ;

**b.** la sortie standard est l'écran, Ce que renvoient les programmes sortent à l'écran ;

**c.** La sortie d'erreur est aussi l'écran. Les messages d'erreur s'affichent aussi à l'écran.

## **2. Entrée, sortie, redirection**

### **2.1 En sortie**

On se sert du caractère `>` pour rediriger la sortie standard (celle qui va normalement sur l'écran). On indique ensuite le nom du fichier où seront placés les résultats de sorties.

## 2.2 En entrée

Les commandes qui attendent des données ou des paramètres depuis le clavier peuvent aussi en recevoir depuis un fichier, à l'aide du caractère <.

Un exemple avec la commande `wc` (word count) qui permet de compter le nombre de lignes, de mots et de caractères d'un fichier.

```
$ wc < resultat.txt
```

## 3. Les canaux standards

On peut considérer un canal comme un fichier dans lequel on peut lire ou écrire.

- Le canal d'entrée standard se nomme **stdin** et porte le descripteur 0.
- Le canal de sortie standard se nomme **stdout** et porte le descripteur 1.
- Le canal d'erreur standard se nomme **stderr** et porte le descripteur 2.

On peut rediriger le canal d'erreur vers un autre fichier.

## Comportement par défaut

- Les canaux standards

- une *entrée standard* <
- une *sortie standard* 1>
- une *sortie d'erreur*. 2>

Vous pouvez rediriger les deux canaux de sortie dans un seul et même fichier, en les liant. On utilise pour cela le caractère >&. Ainsi si vous voulez grouper les deux canaux de sortie et d'erreur dans un même fichier, il faut procéder comme suit 2>&1.

```
$ ls -l > resultat.txt 2>&1
```

## 4. Ouverture de canaux

Les canaux standards sont au nombre de trois et numérotés de 0 à 2.

Ainsi 0< équivaut à < et 1> à >.

La commande **exec** permet d'ouvrir sept autres canaux numérotés de 3 à 9. On a donc en tout dix canaux.

```
$ exec 3> dump.log
```

```
$ ls -l >&3
```

```
$ cat dump.log
```

## 5. Fermeture du canal ouvert

```
$ exec 3>&-
```

## 5. TP Les redirections

### 5.1 Rediriger la sortie dans un fichier : >

La sortie standard d'une commande vers un fichier. Le résultat de la commande sera placé dans le fichier au lieu de s'afficher sur l'écran. Exemple :

- **boga~ \$ ls -l > f1**

### 5.2 Ajouter la sortie à un fichier : >>

- **boga~ \$ ls -l >> f1**

Le fichier f1 contiendra *à la suite* de son contenu le résultat de la commande

### 5.3 Rediriger l'entrée : <

On peut aussi rediriger l'entrée standard d'une commande. La commande lit dans ce cas le fichier au lieu du clavier. Exemple :

- **boga~ \$ more < f1**

## 5.4 Connecter la sortie d'une commande sur l'entrée d'une autre :

Un pipe connecte directement la sortie standard d'une commande sur l'entrée standard d'une autre commande

Exemple au lieu de taper

```
boga~ $ ls -l > f1
```

```
boga~ $ more < f1
```

On peut se passer du fichier intermédiaire f1 dans notre exemple grâce à un *pipe*

```
boga~ $ ls -l | more
```

## 5.5 Redirection des sorties standard et d'erreur

Il est parfois utile de rediriger la sortie standard et la sortie d'erreur vers un même endroit. Pour cela, on utilise le motif `2>&1` avant la redirection

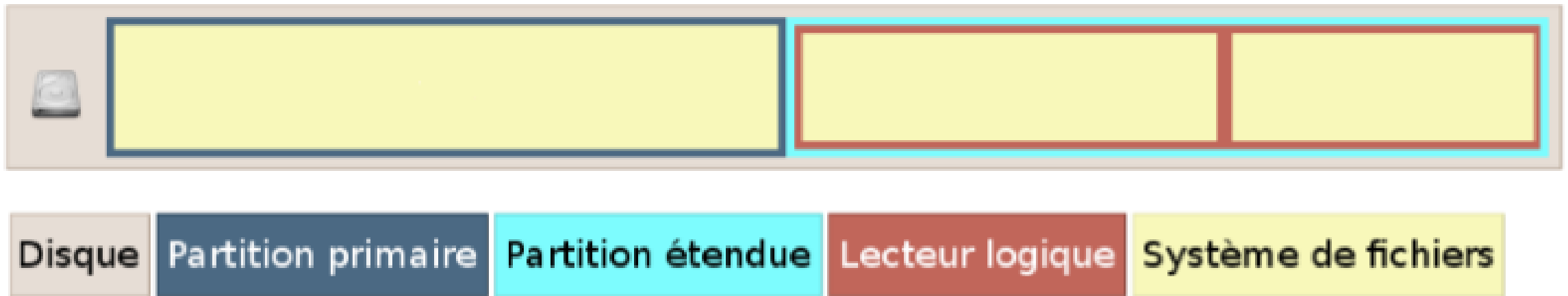
Fonction	Bourne shell (sh, bash)
Redirige la sortie d'erreur (2) et la sortie standard (1) sur l'entrée de la commande suivante	2>&1
Redirige la sortie d'erreur et la sortie standard vers fichier	>fichier 2>&1
Redirige la sortie d'erreur et la sortie standard à la fin de fichier	>>fichier 2>&1

# Partie 3: Partitionnement d'un DD



# Partitionnement de disque

- Le partitionnement est le fractionnement d'un support physique en plusieurs parties virtuelles.
- A l'intérieur d'une partition, un système de fichiers doit être créé. Celui-ci sert à organiser les données à l'intérieur de la partition,  
**Formater une partition**, c'est y créer un système de fichiers.
- Le système de fichiers permet de localiser à quel emplacement est enregistrée la donnée exacte demandée par l'utilisateur.



## Nommage des disques

- Deux types de disques : les IDE et les SCSI. Ils sont nommés « **hd** » et « **sd** » suivi d'une lettre et d'un chiffre. La lettre représente la position du disque sur le bus et le chiffre représente l'une des partitions de ce disque. Un disque dur peut être divisé en partitions.

Les partitions sur chaque disque sont représentées en ajoutant un numéro au nom du disque : **sda1** et **sda2** représentent la première et la seconde partition

## Points de montage

- Sous Linux, tout est fichier, même les périphériques. Tout périphérique est identifié à un fichier qui se trouve dans le répertoire /dev (device, périphérique en anglais).
- Les partitions créées sur les disques durs seront attachées à des "points de montage". Le point de montage est un simple répertoire vide avant le montage et qui après le montage, représente le contenu de la partition montée

# Préparation du support de stockage

## A. Création des partitions

Supposons que votre disque est détecté en tant que `/dev/sda`

1. Lancer l'interface de partitionnement `fdisk` en ligne de commande, en précisant le disque à modifier : **`fdisk /dev/sda`**
2. **`m`** affiche l'aide
3. **`n`** permet d'ajouter une nouvelle partition (affiche des informations par défaut)  
(Accepter la valeur par défaut)
4. **`p`** ou **`e`** (**`p`** pour primaire et **`e`** pour étendue)
5. Préciser la taille qu'on veut donner à la partition  
+ taille Go (ou Mo)    **ex: +5 Go**
- `w`** (pour prendre en compte la partition sauvegarder les modifications)

**Partprobe** (force le chargement de la partition)

## B. Formatage des partitions ou création du SGBF

Elle consiste à formater les partitions en ext3, ext2 ou ext4, système de fichiers conseillé sous Linux.

Pour ce faire, on utilise la commande **mkfs** en précisant le chemin du disque :

```
# mkfs -ext4 /dev/sda6 (6 partition crée)
```

## C. Monter la partition

1. On crée le répertoire de montage

```
# mkdir /backup
```

2. On fait le montage

```
Mount /dev/sda6 /backup
```

**Df -h** affiche le travail effectué

Les partitions créées ne sont pas montées automatiquement au démarrage. Pour le faire il faut modifier le fichier `/etc/fstab` (*FileSystemTable: table de système de fichiers*). Dans ce fichier se trouvent les partitions montées automatiquement au démarrage.

**Nb:** avant de modifier un fichier, il faut l'enregistrer sous un autre nom

Le fichier `/etc/fstab` liste les partitions qui seront montées au démarrage ou à la connexion du périphérique avec toujours les mêmes options.

La ligne suivante doit être ajoutée:

Laissez-y les valeurs par défaut pour les valeurs de priorité:

- 1 pour la racine
- 2 pour les autres partitions Linux

# **nano ou vi** `/etc/fstab`

# indiquer commentaire				
<b>/dev/sda6</b>	<b>/backup</b>	<b>ext4</b>	<b>default</b>	<b>1 2</b>
Nouvelle partition	nom de la partition	SGBF		Priorité
:wq				

# Partie 4: Gestion de planification des tâches

# Gestion de la planification des tâches avec l'utilitaire CRON

- La table qui contient les lignes de planification s'appelle **crontab**
- La commande qui ouvre la table à remplir est: **crontab -e**
- La table affiche des lignes de sept colonnes

**#crontab -e**

*	*	*	*	*	*	*
Minute	heure	jour	mois	semaine	default	commande

**Exemple:** toutes les 22 heures 30 minutes quelque soit le jour, le mois et la semaine.  
Créer un fichier archive du répertoire **/home** nommé **archivehome.tar** dans le répertoire **/bakup**

```
30 22 * * * tar cvf /bakup/archivehome.tar /home
```

# Script

Un script shell est un fichier en mode texte. C'est-à-dire que ce n'est pas un fichier binaire, **exécutable** directement par la machine, mais **il doit être interprété**.

## L'interprétation d'un script

L'interprétation signifie que chaque commande contenue dans un script doit être lue par un programme, appelé *interpréteur*; l'interpréteur analyse chaque commande du script et la traduit en langage machine, ce qui permet l'exécution du script.

## L'édition d'un script

Un script étant un fichier en mode texte, il doit être créé avec un éditeur de texte. Il suffit de créer un fichier vierge, de lui donner un nom,

- Sous linux cela se fait avec **nano**, **vi** ...
- Comme première ligne de ce fichier.

**#!/bin/bash**

- Avant de lancer le script, il est nécessaire de modifier les permissions le concernant et de le rendre exécutable. Cela se fait avec la ligne de commande suivante : **chmod u+x nom\_du\_script**



Pour pouvoir utiliser mes scripts en tapant directement leur nom (sans le "./") depuis n'importe quel répertoire de mon ordinateur, il me suffit d'indiquer au shell de chercher aussi dans ce nouveau dossier en l'ajoutant au PATH. Pour ceci, il suffit de faire :

```
# export PATH=$PATH: /le_chemin_du_nouveau_répertoire_contenant_le_script
```

Ex:

```
# export PATH=$PATH: /projets/mesutilitaires (mesutilitaires contient le script)
```

## Principe de fonctionnement de la tâche planifiée

1. Créer le script en fonction du travail qu'on veut exécuter et de l'enregistrer sous un nom.exe
2. Ajouter le script nom.exe dans la table **crontab** en tenant compte de la période d'exécution

# Partie 13: Gestion Réseau

# Gestion Réseau

❑ Pour obtenir les paramètres réseau de base: (Affichage des interfaces)

- Windows: **ipconfig**
- Linux: **ifconfig** ou **ip** à partir de la version 17.10 de Ubuntu (interface activés)  
**ifconfig -a** (interface activés ou non)

❑ Contacter une autre machine (Windows ou Linux) **ping** « adresse »

Les fichiers de configuration des cartes réseau se trouvent:

- Sous red'hat et descendant : /etc/sysconfig/network-script/

Il y'a deux fichiers: **ifcfg-enxxxxxxxx** et **ifcfg-lo**

Modifier le fichier : ifcfg-eno**xxxxxxxx**

Nano : /etc/sysconfig/network-script/ ifcfg-eno**xxxxxxxx**

Type

# Bootproto = « DHCP » (commentaire)

def root = « yes »

peeDNS = « yes »

IPV4

IPADDR =

NETMASK =

- Sous Debian et descendants : /etc/network/interfaces

Il y'a un fichier: interfaces

Modifier le fichier : interfaces

Nano : /etc/network/interfaces

```
# network manager #
```

```
auto eth0
```

```
if face eth0 inet DHCP ou Static (pour fixer les adresses)
```

```
address      x.x.x.x
```

```
Netmask      x.x.x.x
```

```
gateway      x.x.x.x
```

**NB:** avant de modifier les fichiers de configuration, il faut d'abord sauvegarder le fichier original sous un autre nom

**Copie distantes d'un serveur à un autre serveur**

```
#scp username @ ip_source: /chemin/fichier username @ ip_destination /chemin/fichier
```

## Utiliser ifconfig

1. Définir votre adresse IP

```
root@Baobab ~# ifconfig eth0 192.168.1.2 netmask 255.255.255.0 up
```

2. Définir votre passerelle par défaut

```
root@Baobab ~# route add default gw 192.168.1.1
```

3. Définir votre serveur DNS

```
[root@Baobab ~]# echo "nameserver 1.1.1.1" > /etc/resolv.conf
```

## Utiliser la commande ip et netplan

Ifconfig étant en cours de suppression, il est temps de s'habituer au nouveau système. Par défaut, Ubuntu 18.04 n'utilise plus « ifconfig », mais utilise les nouvelles commandes « ip » et « netplan ».

1. Afficher votre IP en utilisant la commande ip

```
[root@Baobab ~]# ip addr show
```

2. Activer ou désactiver une interface réseau

```
[root@Baobab ~]# ip link set eth0 up
```

```
[root@Baobab ~]# ip link set eth0 down
```

3. Afficher la table de routage

```
[root@Baobab ~]# ip route show
```

## Modifier les informations de réseau

Pour Ubuntu, voici l'alternative de **/etc/network/\*** dans l'ancien système. L'ensemble du système utilise désormais les fichiers de configuration YAML sous **/etc/netplan**, puis la commande **netplan** applique ces configurations au système.

```
root@Baobab ~# nano /etc/netplan/*.yaml
```

```
network:
```

```
version: 2
```

```
ethernets:
```

```
eth0:
```

```
addresses: [192.168.1.2/24]
```

```
gateway4: 192.168.1.1
```

```
dhcp4: true
```

```
optional: true
```

```
nameservers:
```

```
addresses: [8.8.8.8,8.8.4.4]
```

ONBOOT détermine si l'interface doit être automatiquement activée au démarrage de la machine.

Une fois le fichier correctement renseigné, on utilise les commandes ifup/ifdown :

# **ifdown** désactive la carte réseau

# **ifup** active la carte réseau avec rechargement des paramètres modifiés

Ou

# service network restart

Les paramètres parlent d'eux-mêmes. Les valeurs NETWORK et BROADCAST sont optionnelles si IPADDR et NETMASK sont renseignés (dans ce cas le calcul est automatique) ou si DHCP est utilisé. BOOTPROTO indique comment monter l'interface, soit static, soit dhcp. La valeur bootp peut aussi être utilisée. Dans le cas de DHCP, le fichier peut ressembler à ceci :

DEVICE=eth0 ONBOOT=yes BOOTPROTO=dhcp

Dans le cas d'une configuration statique, IPADDR et NETMASK sont obligatoires :

DEVICE=eth0 IPADDR=192.168.1.2 NETMASK=255.255.255.0 ONBOOT=yes

BOOTPROTO=static

ONBOOT

# LINUX (SE)

## Serveur DHCP

### Présentation

Le service DHCP (Dynamic Host Configuration Protocole) permet aux hôtes d'un réseau de demander et recevoir des informations de configuration.

### Éléments de configuration détenus:

- Plage d'adresse à allouer
- Plage d'adresses réservées
- Bail
- Adresse IP passerelle et masque réseau
- Adresse IP des serveurs DNS

**Package:** dhcp-server

**Pour l'installer:** # apt-get install isc-dhcp-server

**Le fichier de configuration:** /etc/dhcp/**dhcpd.conf**



## ■ Serveur dhcpd

Le serveur dhcpd est un service (daemon) lancé à l'aide d'un script (/etc/init.d/dhcpd).

### a. Démarrage

# service dhcpd start                      **Ou**                      # /etc/init.d/dhcpd start

### b. Exemple de Configuration

Informations de base

```
ddns-update-style none;    # pas de mise à jour du DNS par DHCP
option domain-name "toto.fr"; # nom de domaine transmis au client
option domain-name-servers 192.168.1.254; # liste des DNS séparés par des virgules
default-lease-time 21600; # durée du bail par défaut en secondes sans demande
                           explicite
max-lease-time 43200; # durée max du bail si la demande du client est plus élevée
```

*Comme dhcpd peut gérer plusieurs sous-réseaux, on doit lui préciser les règles à appliquer pour chaque sous-réseau. Généralement dans le cadre d'un petit réseau un seul bloc sera présent mais tous les cas sont envisageables. Si vous êtes certain de n'avoir qu'un seul réseau, vous pouvez omettre la déclaration du subnet (sous-réseau).*

```
# Gestion du sous-réseau 192.168.1.0
subnet 192.168.1.0 netmask 255.255.255.0
{
option routers 192.168.1.254;      # passerelle pour ce réseau
option subnet-mask 255.255.255.0;  # masque de sous-réseau
range 192.168.1.2 192.168.1.250;  # Configuration de l'intervalle DHCP
```

### **# Cas d'attributions d'IP statiques**

```
host station1
{
    { hardware ethernet 00:A0:ad:41:5c:b1; # Adresse MAC
      fixed-address 192.168.1.1;          # cette machine aura l'IP 192.168.1.1
    }
}
```

*# Certains clients DHCP ignorent totalement le fait qu'un serveur DHCP peut dynamiquement allouer un nom (hostname) à l'hôte. Dans l'exemple précédent, la machine avec l'IP 192.168.1.1 devrait obtenir le nom station1.*

# LINUX (SE)

## Serveur DNS

### Présentation

Le Système de Noms de Domaines **DNS** (Domain Name System) transforme:

- Les noms d'hôte en adresses IP : c'est la résolution de nom.
- Les adresses IP en noms d'hôte : c'est la résolution inverse.

**NB:** Le DNS remplace la configuration du fichier `/etc/hosts` qui devait être ajouter à tous les ordinateurs qu'on administre.

**Package:** **Bind** (Berkeley Internet Name Daemon). **Bind 9** supporte l'IPv6.

### Configuration générale

La configuration globale de **Bind** est placée dans le fichier `/etc/named.conf`.

La configuration détaillée des zones est placée dans `/var/lib/named`.

`/etc/named.conf` est composé de deux parties:

- La première concerne la configuration globale des options de Bind.
- La seconde est la déclaration des zones pour les domaines individuels.

**NB:** Les commentaires commencent par un `#` ou `//`.