# 1 Problématique : maintenance des systèmes d'exploitation

En fonction des cas d'utilisation, maintenir des systèmes d'exploitation peut nécessiter de penser : mises à jour, indisponibilité, sauvegardes, tests, instantanés, restaurations, recettes de configuration.

## 1.1 Systèmes de fichiers, installés sur partitions, avec accès en écriture

### 1.1.1 Système de fichiers conventionnel : ext2, ext3, ext4, jfs, xfs

- **avantages** : instantanéité de toutes les modifications apportées aux fichiers du système
- **inconvénients** : nécessité de régulièrement réaliser et tester des sauvegardes du système

### 1.1.2 Système de fichiers géré par des recettes de configuration : ansible, chef, puppet

- **avantages** : possibilité de remettre rapidement en état certains pans entiers du système
- **inconvénients** : pas de résolution des écarts de configuration non gérés par les recettes

### 1.1.3 Système de fichiers avec gestion d'instantanés : btrfs, zfs

- **avantages** : permet de sauvegarder et restaurer un état des fichiers du système à un instant
- **inconvénients** : réduit progressivement l'espace disponible, pas encore utilisé par défaut

## 1.2 Images autonomes, sans installation, avec accès en lecture seule

### 1.2.1 Amorçage sans gestion de persistance

- **avantages** : démarrer sur un système autonome dans un état ayant été figé au préalable
- **inconvénients** : perdre au redémarrage toutes modifications faites aux fichiers du système

### 1.2.2 Amorçage avec gestion de persistance

- **avantages** : conservation sur une partition marquée des fichiers modifiés depuis le démarrage
- **inconvénients** : pas de séparation entre la persistance des fichiers systèmes et des données

---

# 2 Proposition : fonctionnement autonome incrémental

Mettre en œuvre un système d'exploitation hybride entre un système installé et un système autonome : cumuler les avantages des deux, en images incrémentales ou complètes, sans les divers inconvénients.

- **avantages** : redémarrage = restauration, mise à jour = sauvegarde, séparation système/données
- **inconvénients** : maintenance exhaustive si effectuée régulièrement et d'une façon manuelle

## 2.1 Miroirs de dépôts officiels distribution et éditeurs

- synchronisation locale pour accès rapide, stable et hors-ligne : **apt-mirror**, **debmirror**, **ftpsync**
- vérification d'intégrité des dépôts locaux avant utilisation de leurs paquets logiciels synchronisés

## 2.2 Construction d'un système de fichiers autonome (debian gnu/linux)

- prise en compte du type de machine hôte pour le choix des paquets de base : physique, virtuelle
- création d'un système de fichiers de base minimal à partir des dépôts locaux : **debootstrap**
- intégration des paquets nécessaires à la construction d'autres systèmes autonomes, si besoin
- transformation effective en système d'exploitation autonome : **live-boot**, **update-initramfs**
- détermination des autres paquets logiciels à installer et à configurer, en fonction des besoins
- déport des données à rendre persistantes, avec des liens symboliques pointant vers partition(s)

### 2.3 Encapsulation dans un fichier image

- utilisation d'un format de fichier amorçable adapté au montage en lecture seule : **squashfs**
- choix d'un des divers algorithmes de compression disponibles : **gzip**, **lzma**, **lzo**, **lz4**, **xz**, **zstd**
- niveau supplémentaire d'encapsulation avec un format de fichier amorçable hybride : **iso**

### 2.4 Sécurité du fichier image produit

- assurer l'intégrité du fichier final par le calcul d'une somme de contrôle : **sha256**, **sha512**
- garantir l'authenticité de l'image grâce à une signature numérique associée au fichier : **gpg**

### 2.5 Amorçage de fichier(s) image(s) sécurisé(s)

- chargeur de démarrage avec gestion de signature numérique : **grub**, **bios**, **uefi**, **secure boot**
- création d'un menu de démarrage à choix multiple d'images : **grub.cfg**, **squash4**, **iso9660**
- vérification d'authenticité et d'intégrité de fichiers images : **gcry_sha256**, **gcry_sha512**, **pgp**
- chargement d'image(s) en mémoire vive d'une machine hôte : complet, partiel avec **overlayfs**

### 2.6 Mise à niveau incrémentale

- fabrication d'une nouvelle image, à partir de la plus récente, pour le prochain redémarrage
- si le redémarrage est différé, mise à jour du système d'exploitation actuellement en mémoire
- si le redémarrage est nécessaire et critique, réduction de sa durée effective : **kexec-tools**

## 3 Automatisations potentiellement implémentables

- vérification d'intégrité des dépôts, voire le processus de synchronisation, de façon parallélisée
- construction de systèmes de fichiers autonomes complets, à partir de différents profils versionnés
- création de nouveaux fichiers images, par la mise à jour d'images amorçables déjà existantes
- génération à la volée de menus de démarrage, à choix multiples d'images amorçables détectées