

**Réalisation d'un programme de gestion automatique de backups, sécurisé, et permettant la récupération d'anciens fichiers, avec possibilité de récupérer des fichiers mis à jour depuis Internet de façon sécurisée, tolérant les erreurs.**

## Présentation

**Conditionnement** : Binômes de 2 ou 3 personnes.

**Restrictions** : Pas de copier/coller entre les groupes. Google autorisé (évidemment).

**Rendu** : Mini-rapport (quelques pages) détaillant la mise en oeuvre + 10min de soutenance, à rendre avant le 16 décembre à 23:00, sur ARCHE. Un point en moins par heure de retard.

**Questions** : [guillaume@netlor.fr](mailto:guillaume@netlor.fr) avant le 12 décembre. 0,5 points en moins par question après cette date, sauf question pertinente.

Le but de ce projet est de réaliser un programme de sauvegarde et de mises à jour de fichiers avancé, qui permet de :

- Faire des backups (sauvegardes) de ses fichiers personnels ;
- Compresser ces backups pour qu'ils prennent moins de place ;
- Chiffrer ces backups pour empêcher une personne tierce de les récupérer ;
- Afficher les différences entre un backup A et un backup B ;
- Accéder à un fichier stocké dans un ancien backup et le récupérer ;
- Supprimer progressivement les anciens backups ;
- Récupérer des mises à jours de certains fichiers depuis Internet, depuis une page qui n'a pas été prévue à cet effet ;
- Vérifier que les fichiers récupérés soient valides ;
- Mettre à jour les fichiers avec les fichiers valides ;
- Gérer les erreurs de connexion ;
- Permettre l'upload de ses backups en ligne ;
- Réalise toutes ses actions régulièrement, et automatiquement ;
- Envoie un mail en cas d'erreur ou quelconque problème ;
- Possède une interface agréable à utiliser ;
- Permet de créer un utilisateur, qui est un clone d'un utilisateur existant avec les fichiers d'un état antérieur, et de le supprimer après usage.

Toutes les fonctionnalités peuvent ne pas être implémentées : on préférera une implémentation correcte de quelques fonctions plutôt qu'une implémentation bâclée de l'ensemble des fonctions. L'implémentation propre de toutes les fonctions vous donnera évidemment le plus de points. La note maximale sera échelonnée à partir du meilleur projet.

Le projet doit pouvoir être installé dans une version "propre" de Debian (c'est-à-dire l'image de base qui a été donnée au premier TP).

Votre script doit pouvoir "s'auto-installer" si des dépendances sont manquantes, en utilisant `apt-get install <package>`. Une liste de divers paquets dont vous pourriez avoir besoin est listée à la fin du sujet.

## Description des fonctionnalités

### Fonctionnalité de backup :

Cette fonctionnalité doit permettre d'effectuer une sauvegarde de dossiers spécifiés dans un fichier de configuration (par exemple, un fichier contenant les différents dossiers ligne par ligne). Ce fichier peut soit avoir un nom par défaut (`backup.conf`), soit un nom spécifié par une option dans la ligne de commande (`./monapplication.sh --conf backup.conf`).

Les fichiers de sauvegarde seront stockés dans un dossier à part, qui ne peut pas être sauvegardé lui-même. Idéalement, ce dossier serait également configurable, avec une valeur par défaut (`/var/backups`, ou une option par exemple `--backupdir /var/mesbackups`).

Le backup doit pouvoir être lancé à n'importe quel moment.

Les fichiers du backup ne doivent que pouvoir être lus par l'utilisateur qui les a créés.

### Compression des backups :

Les backups doivent être stockés dans un unique fichier au format `.tar` créés à l'aide de la commande `tar`, et compressés au format `GZip` à l'aide de la commande `gz`.

### Chiffrement des backups :

En utilisant `GPG` (implémentation GNU du standard `PGP` de chiffrement), les différents backups doivent être chiffrés le plus tôt possible, et déchiffrés uniquement si besoin. Les fichiers temporaires déchiffrés doivent être supprimés le plus tôt possible.

Pour que `GPG` fonctionne et puisse chiffrer des fichiers, il faut lui indiquer ou générer une clé qui va servir au chiffrement et au déchiffrement de ceux-ci.

**Afficher les différences entre un backup A et B :**

Tous les backups doivent être datés et incrémentaux. Le script doit permettre d'afficher les modifications (dates de modification, différence de taille uniquement, pas (forcément) le contenu du fichier) entre un même fichier ou dossier de deux backups différents, soit par date, soit par delta (le dernier, l'avant-dernier, ...). Le fichier ou le dossier en question doit exister dans les deux backups pour que la différence puisse être faite.

**Accéder à un ancien fichier stocké dans un backup :**

L'application doit permettre de pouvoir récupérer un fichier stocké dans un précédent backup, et doit pouvoir l'enregistrer sous un chemin et/ou un nom que l'utilisateur peut rentrer (afin de ne pas écraser sa copie existante du fichier).

**Supprimer progressivement les anciens backups :**

L'application doit pouvoir stocker jusqu'à 100 fichiers de sauvegarde. S'il existe plus de 100 fichiers, le fichier le plus ancien doit être supprimé d'abord.

**Récupérer les mises à jour depuis un site Internet non prévu à cet effet :**

Vous suivez les épisodes de Game of Thrones, et vous souhaitez conserver une synopsis de chaque épisode. Cependant, cette synopsis n'est pas disponible de façon simple via une API : vous devrez utiliser un site diabolique qui se trouve à l'adresse <https://daenerys.xplod.fr>. Le but est de récupérer, automatiquement, les synopsis des épisodes dans des fichiers distincts au fur et à mesure qu'ils deviennent disponible. Les fichiers doivent être stockés dans un dossier GoT situé dans le home de l'utilisateur, et on doit avoir un fichier nommé "Saison X Episode X.txt" pour chaque épisode, contenant le synopsis de l'épisode en question.

La synopsis de chaque épisode peut évoluer avec des corrections, il faut donc bien que lors de la synchronisation, tous les fichiers de tous les épisodes soient à jour.

Chaque synopsis contient un lien vers un fichier pour SuperSynopsis, un logiciel de synopsis, qu'on souhaite conserver également à jour.

Cette synchronisation doit se lancer avant chaque backup, pour que votre backup contienne les épisodes les plus à jour.

**Vérifier que les mises à jour soient valides :**

Les fichiers de SuperSynopsis sont signés numériquement à l'aide de GPG. La clé publique permettant de vérifier la véracité des fichiers est disponible sur le site [https://daenerys.xplod.fr/supersynopsis\\_signature.pub](https://daenerys.xplod.fr/supersynopsis_signature.pub). Il faut vérifier que les fichiers récupérés soient correctement signés, et les rejeter s'ils ne le sont pas.

Notez que votre script peut fonctionner en mode silencieux (--quiet ou -q), auquel cas les erreurs de vérification de fichiers doivent être signalés par mail. Si le script n'est pas en silencieux, les erreurs doivent apparaître sur la sortie d'erreurs.

**Gérer les erreurs de connexion :**

Il est possible que le site Daenerys ne soit pas toujours disponible. A cet effet, votre script de récupération de synopsis (aussi bien le texte, que le fichier SuperSynopsis) doit pouvoir réessayer plusieurs fois avant d'effectivement sauter la récupération.

Notez que votre script peut fonctionner en mode silencieux (--quiet ou -q), auquel cas les erreurs de connexion doivent être signalés par mail. Si le script n'est pas en silencieux, les erreurs doivent apparaître sur la sortie d'erreurs.

**Permettre l'upload de ses backups en ligne :**

Le site <https://daenerys.xplod.fr/backup> vous permet de stocker en ligne vos différents backups. L'API est expliquée sur le site en question. Le script doit permettre de pouvoir uploader un ou plusieurs backups sur ce site, et de pouvoir les re-télécharger par la suite et les ré-utiliser avec les autres fonctions (différence entre deux backups, récupération d'un fichier depuis un backup téléchargé, etc).

**Réaliser l'opération régulièrement et automatiquement :**

CRON est un daemon système permettant de réaliser des tâches de façon périodique. Configurez votre crontab pour que votre script de backup s'exécute automatiquement, de façon silencieuse (donc les erreurs envoyées par mail), toutes les heures.

Il faut donc que chaque heure (13:00, 14:00, 15:00, ...), le script synchronise les synopsis, effectue un backup, et l'upload sur le service d'upload de backups, le tout automatiquement.

**Possède une interface agréable à utiliser :**

En utilisant select et dialog, votre script doit être agréable à utiliser et les opérations doivent pouvoir se faire en tapant le moins de commandes possibles. Ce point est de très basse priorité et doit être fait plutôt si le reste est déjà fait.

**Crée un utilisateur avec les fichiers antérieurs :**

Dans le cas où on sauvegarde un dossier /home, on peut imaginer avoir une fonction créant un utilisateur en lui appliquant les fichiers antérieurs provenant d'un backup. Seuls les fichiers dans /home/xxx du backup seraient restaurés. Les permissions des fichiers restaurés pour ce nouvel utilisateur doivent être adaptées correctement.

# Ressources

## Packages utiles

Cette liste contient des paquets qui pourront vous être utile pour réaliser votre projet. Consultez leur documentation pour plus d'informations sur leur utilisation. Il se peut que vous n'ayez pas besoin de toutes les commandes, et il se peut que vous ayez besoin d'autres commandes qui ne sont pas listées ici.

- `gnupg` : Fournit la commande `gpg` pour le chiffrement
- `dialog` : Fournit la commande `dialog` pour des boîtes de dialogues en console
- `curl` : Fournit la commande `curl` pour interagir avec un serveur Web
- `wget` : Fournit la commande `wget` pour télécharger un document sur le Web
- `sendmail` : Fournit la commande `sendmail` pour envoyer un e-mail