

Aluminium Backup

Rémy Moll, TSMP

DFG-Freiburg

14 mai 2019



Contenu :

Idée

Réalisation

Difficultés

Problèmes

Site

1 Idée

2 Réalisation

3 Difficultés

4 Problèmes

5 Site



■ Inspiration

- Titanium Backup comme inspiration
- initialement : script + interface

■ Objectifs

- Accès facile aux fichiers sauvegardés
- Copie des programmes



■ Interface

- Une fenêtre universelle
- Fenêtre de parametres
- Fenêtre de progres

■ Moteur

- Deux modules : backup et restore
- Un script executé en cronjob



- Démarrage de `main.py`
 - Lecture de la configuration
 - Création de la fenêtre
 - Création des "listeners"
- Chargement de l'interface
 - Création des widgets
- Lors de l'event 'START'
 - 1 Fin du UI
 - 2 Sauvegarde des parametres
 - 3 Création de l'affiche de progrès
 - 4 Fichiers se font copier
 - 5 Signal de fin



Difficultés rencontrées (et résolues)

[Idée](#)

[Réalisation](#)

[Difficultés](#)

[Problèmes](#)

[Site](#)

- Interférence entre Qt et Threading
- OS et Threading



Problèmes / Bugs

"No, they're *features*!"

Idée

Réalisation

Difficultés

Problèmes

Site

- Appliquer tous les paramètres de manière efficace
 - Compression et chiffrement
 - Création de la fenêtre
- Symlinks
 - Estimation de la taille
 - Mauvaise location
- Taille des fichiers
- Redondance du code



- Destination et langue par des paramètres GET
- Chargement d'une base de données
- Header et footer statiques
- Sanitisation des entrées : erreur 404 et protection d'injections SQL

