

# SAE 3.03 - Réseau et Application Serveur

ALLART Noah  
MANGIN Adrien

## Table des matières

<b>Introduction</b>	<b>2</b>
<b>1 Conception du mbash</b>	<b>2</b>
Principes de base . . . . .	2
Gestion des alias . . . . .	2
<b>2 Serveur de Package Debian</b>	<b>3</b>
2.1 Construction du Paquet <code>.deb</code> . . . . .	3
2.2 Gestion des Clés . . . . .	3
2.3 Structure du Serveur HTTP . . . . .	3
<b>3 Illustrations</b>	<b>4</b>
<b>Illustrations</b>	<b>4</b>
<b>4 Sources</b>	<b>5</b>
<b>Sources</b>	<b>5</b>

# Introduction

Dans le cadre de la SAE 3.03, notre objectif principal a été de concevoir et de développer un shell minimaliste, nommé **mbash**, tout en apprenant à maîtriser la création, la distribution et l'installation de paquets Debian. Ce projet nous a permis de mettre en œuvre des concepts fondamentaux liés au développement système, tels que la gestion des processus, l'interprétation des commandes utilisateur, ainsi que l'utilisation des bibliothèques de bas niveau pour exécuter et manipuler des commandes.

De plus, nous avons abordé les bonnes pratiques pour la distribution de logiciels via un serveur de paquet Debian (apache2). Cela inclut la signature des paquets pour garantir leur authenticité, ainsi que la configuration et l'administration d'un serveur HTTP pour héberger le dépôt debian.

Ce rapport présente les étapes principales de la conception du **mbash** et de sa distribution en tant que paquet Debian.

## 1 Conception du mbash

### Principes de base

Dans un premier temps, nous avons fait en sorte de pouvoir utiliser les commandes contenues dans les dossiers du `$PATH`. Pour cela, nous avons utilisé la commande `execve()` qui nous permet d'exécuter des commandes bash passées en paramètre.

Ensuite, nous avons permis à `execve()` de pouvoir utiliser et comprendre les paramètres donnés aux commandes.

Pour finir, le but a été de pouvoir exécuter des commandes built-in via notre mini bash. Nous avons donc dû recréer le principe de chaque commande built-in que nous voulions implémenter.

Voici les commandes implémentées :

- `cd` : pour se déplacer dans le `HOME`.
- `cd [chemin]` : pour se déplacer dans le répertoire donné.
- `export x=1` : pour sauvegarder une variable d'environnement (utilisable par la suite).
- `unset x` : pour supprimer une variable d'environnement.
- `help` : affiche la liste de toutes les commandes associées à une description.
- `help [nomCmd]` : affiche la description de la commande.
- `exit` : quitte le programme **mbash**.

### Gestion des alias

Nous avons aussi implémenté le fait de pouvoir utiliser des alias qui sont enregistrés dans le fichier `~/.mbashrc` (le fichier est créé automatiquement lors de la commande `sudo apt install mbash`). L'ajout des alias se fait de cette façon :

```
sudo nano ~/.mbashrc
```

Écrire dedans l'alias sous cette forme : `ll="ls -l"`.

Les `&&`, `|`, `&` et `;` sont gérés grâce à un automate qui va traiter la ligne donnée par l'utilisateur avant d'exécuter les commandes.

## 2 Serveur de Package Debian

### 2.1 Construction du Paquet .deb

La réalisation d'un paquet nécessite de suivre une convention quant à sa structure (voir Figure 1).

Le fichier `control` contient les informations sur le paquet :

```
Package: mbash-compiled
Version: 1.0-1
Section: utils
Priority: optional
Architecture: amd64
Depends: libc6 (>= 2.28)
Maintainer: allart noah, mangin adrien
Description: minibash de la SAE 3.03
```

On construit ensuite le paquet avec la commande suivante :

```
dpkg-deb --build deb_package mbash_1.0-1_amd64.deb
```

### 2.2 Gestion des Clés

On génère une paire de clés avec la commande :

```
gpg --full-generate-key
```

Choisir `RSA`, la taille par défaut et 0 pour une durée de validité infinie.

On peut vérifier les clés avec la commande :

```
gpg --list-keys
```

On exporte ensuite la clé publique afin de pouvoir la distribuer :

```
gpg --armor --export "Votre Nom" > public_key.gpg
```

On signe ensuite le paquet :

```
dpkg-sig --sign builder mbash_1.0_amd64.deb
```

### 2.3 Structure du Serveur HTTP

On crée la structure de fichier nécessaire (voir Figure 2) qu'on complète avec la commande :

```
dpkg-scanpackages . /dev/null | gzip -9 > Packages.gz
```

On place cette structure dans un répertoire `DEBIAN` situé sur un serveur `http`. Avec `apache2`, on lance le service avec la commande :

```
sudo /etc/init.d/apache2 start
```

Le répertoire `DEBIAN` se trouverait donc dans `/var/www/debian`.

Sur la machine client, il faut ajouter le serveur `http` dans la liste des serveurs de packages en éditant le fichier :

```
sudo nano /etc/apt/sources.list.d/mbash.list
```

En ajoutant les lignes suivantes :

```
deb [arch=amd64] http://127.0.0.1/debian stable main
deb [signed-by=/usr/share/keyrings/mondepot.gpg] http://127.0.0.1/debian stable main
```

On met à jour la liste des serveurs avec les commandes :

```
sudo apt upgrade
sudo apt update
```

On installe ensuite le paquet (voir Figure 3) :

```
sudo apt install mbash
```

### 3 Illustrations



FIGURE 1 – Structure du paquet Debian



FIGURE 2 – Structure sur le serveur HTTP

```

e4415u@e4415u-VirtualBox:/var/www/html$ sudo apt install mbash
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
mbash est déjà la version la plus récente (1.0).
Les paquets suivants ont été installés automatiquement et ne sont plus nécessaires :
  linux-headers-5.15.0-67-generic linux-hwe-5.15.0-67 linux-image-5.15.0-67-generic linux-modules-5.15.0-67-generic linux-modules-extra-5.15.0-67-generic
Veuillez utiliser « sudo apt autoremove » pour les supprimer.
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
e4415u@e4415u-VirtualBox:/var/www/html$ mbash
e4415u-VirtualBox:/var/www/html$ help
Liste des commandes disponibles :
cd : Change le répertoire courant. Utilisation : cd [répertoire]
pwd : Affiche le répertoire courant.
echo : Affiche des arguments ou du texte. Utilisation : echo [texte]
exit : Quitte le shell. Utilisation : exit [code_de_sortie]
env : Affiche les variables d'environnement.
unset : Supprime une variable d'environnement. Utilisation : unset [variable]
export : Ajoute ou modifie une variable d'environnement. Utilisation : export [variable=valeur]
help : Affiche cette liste d'aide ou des détails sur une commande. Utilisation : help [commande]
e4415u-VirtualBox:/var/www/html$ ^C
e4415u@e4415u-VirtualBox:/var/www/html$
e4415u@e4415u-VirtualBox:/var/www/html$ help
Liste des commandes disponibles :
cd : Change le répertoire courant. Utilisation : cd [répertoire]
pwd : Affiche le répertoire courant.
echo : Affiche des arguments ou du texte. Utilisation : echo [texte]
exit : Quitte le shell. Utilisation : exit [code_de_sortie]
env : Affiche les variables d'environnement.
unset : Supprime une variable d'environnement. Utilisation : unset [variable]
export : Ajoute ou modifie une variable d'environnement. Utilisation : export [variable=valeur]
help : Affiche cette liste d'aide ou des détails sur une commande. Utilisation : help [commande]
e4415u-VirtualBox:/var/www/html$ help cd
cd : Change le répertoire courant. Utilisation : cd [répertoire]
e4415u-VirtualBox:/var/www/html$ cd
e4415u-VirtualBox:/home/e4415u$ ^C
e4415u@e4415u-VirtualBox:/var/www/html$
  
```

FIGURE 3 – Illustration du fonctionnement

## 4 Sources

### Sources générales

- **Debian New Maintainers' Guide** : Guide officiel pour apprendre à créer des packages Debian. Comprend des informations détaillées sur la structure des fichiers, les dépendances et les scripts.
- **Debian Policy Manual** : Normes et conventions pour les packages Debian.
- **GPG Documentation** : Documentation officielle pour générer et utiliser des clés GPG.
- **dpkg Documentation** : Informations sur les outils `dpkg` et `dpkg-deb` utilisés

pour construire et gérer les packages Debian.

## Commandes spécifiques

- **dpkg-deb** : Pour construire les packages Debian (Référence : [dpkg-deb Manual](#)).
- **debsign** : Pour signer les packages Debian (Référence : [debsign Manual](#)).
- **gcc** : Compilation du code source (Référence : [GCC Documentation](#)).

## Tutoriaux communautaires et autres ressources

- **Creating Debian Packages** - Linuxize.
- **Packaging Tutorial** - Ubuntu Wiki.
- **Stack Overflow** : Pour résoudre des problèmes spécifiques et des astuces pratiques.