

Rapport SAE 3.03 - Réseau et application serveur mbash

#SAE

Élèves : Valentin Knorst | Zacharie Heuertz

Introduction

Ce rapport détaille les étapes et les fonctionnalités développées dans le cadre de notre projet **mbash**, ainsi que les choix techniques effectués.

Ce projet a consisté en la création et l'installation d'un shell minimaliste, enrichi de fonctionnalités supplémentaires pour se démarquer des autres.

Installation

Nous avons configuré **mbash** pour qu'il puisse être installé directement à partir d'un dépôt, via la commande suivante :

```
sudo apt install mbash
```

Cela permet une installation rapide et une gestion simplifiée du programme, imitant le processus des packages standards sur Linux.

Nous détaillerons plus tard comment nous sommes arrivés à ce résultat.

Fonctionnalités principales

Commandes implémentées

1. **ls** : Reproduction partielle de la commande `ls` servant à lister le contenu d'un dossier.
option : `-l`, qui a nécessité des recherches approfondies sur les appels système et privilèges.
2. **cd** : Permet de naviguer dans les répertoires.
3. **help** : Fournit une aide contextuelle sur les commandes disponibles.
4. **exit** : Quitte le shell proprement.
5. **pwd** : Affiche le chemin courant du répertoire de travail.
6. **info** : Permet d'activer ou de désactiver le module info.
Ce dernier a été imaginé par nos soins afin de donner des informations complémentaires à l'utilisateur.
Il détaille par exemple ce que fait une commande et donne aussi en combien de temps cette dernière s'est exécutée.

Exécution des autres commandes Linux

Grâce à l'utilisation de **execve**, nous avons permis à toutes les commandes Linux non implémentées directement dans mbash d'être exécutées.

Grâce à un parcours du path bin de l'utilisateur, mbash garantit une compatibilité étendue avec l'environnement Linux.

Mode arrière-plan

Nous avons implémenté la gestion de l'arrière-plan avec le caractère `&`.

Cette fonctionnalité permet d'exécuter des processus en arrière-plan, améliorant ainsi la flexibilité et la performance du shell.

Dans un cas normal, mbash attend le retour de la commande effectué pour redonner la main.

Dans un cas d'arrière plan, il peut lancer d'autre tâches tout en gardant l'exécution de son fils.

Normalement, les zombies sont évités.

Module info

Pour nous démarquer, nous avons développé un **module info** qui peut être activé ou désactivé à volonté. Ce module offre :

- Un suivi détaillé des requêtes exécutées.
- L'affichage du temps d'exécution des commandes.

Cette fonctionnalité est particulièrement utile pour analyser et optimiser les performances des commandes.

Points techniques notables

Travail sur `ls -l`

La reproduction de la commande `ls -l` a été le point le plus complexe du projet. Nous avons dû :

- Comprendre les structures des fichiers et répertoires via des appels système comme `stat`.
- Interpréter les informations obtenues pour les afficher dans un format lisible (permissions, propriétaires, tailles, etc.).

Gestion des chemins

L'exploration des chemins des binaires grâce à **execve** a été une étape formatrice, nous permettant de mieux comprendre comment Linux localise et exécute les programmes.

Recherches approfondies

Nos recherches ont couvert des domaines variés, comme :

- Les appels système (`fork`, `execve`, `wait`, etc.).
 - La gestion des processus.
 - Les redirections et la gestion des descripteurs de fichiers.
-

Conclusion

Ce projet nous a permis de développer un shell fonctionnel avec des fonctionnalités innovantes tout en renforçant nos connaissances sur les bases du système Linux. Nous avons particulièrement apprécié :

- L'exploration des appels système.
- La conception d'un shell modulaire et performant.
- Les défis liés à la reproduction des commandes classiques.

Nous espérons que ce rapport reflète les efforts et les compétences mobilisés pour ce projet.

Nous aurions aimé pouvoir faire un shell plus travaillé et innovateur mais malheureusement avec un membre du binôme ayant été atteint par la grippe durant 3 jours de la SAE, notre progression a été fortement ralentie. Mais nous restons largement très satisfait de notre travail dans ce petit temps de projet et sommes fier de nos efforts fournis

Tutoriel installation serveur avec un dépôt Debian

Partie Serveur : Création d'un dépôt Debian

```
sudo apt update
```

```
sudo apt install dpkg-sig reprepro apache2 gnupg
```

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

Choisir :

- RSA and RSA
- 4096
- Date d'expiration : 0 (pas d'expiration).

mot de passe choisie : loutre

```
gpg: revocation certificate stored as '/home/kali/.gnupg/openpgp-  
revocs.d/50479A6DACDE6BFCBDEC4E193D8B631BD9CB95BB.rev'  
public and secret key created and signed.
```

```
pub rsa4096 2025-01-17 [SC]  
50479A6DACDE6BFCBDEC4E193D8B631BD9CB95BB  
uid valentin gugug3293@gmail.com  
sub rsa4096 2025-01-17 [E]
```

Créer le répertoire du dépôt : `mkdir -p ~/debian-repo/{conf,incoming}`

Configurer reprepro : Créez un fichier distributions dans `~/debian-repo/conf` :

Codename: stable

Components: main

Architectures: amd64

SignWith: 50479A6DACDE6BFCBDEC4E193D8B631BD9CB95BB

Créer un package Debian pour mbash :

- Préparez le binaire ou le script mbash.
- Organisez un répertoire source pour le package :

```
mkdir -p mbash-0.1/usr/bin
cp mbash mbash-0.1/usr/bin/
mkdir -p mbash-0.1/DEBIAN
```

Créez un fichier mbash-0.1/DEBIAN/control :

Package: mbash

Version: 0.1

Section: utils

Priority: optional

Architecture: amd64

Maintainer: Valentin

Description: mbash

Construire le package :

```
dpkg-deb --build mbash-0.1
```

Signer le package :

```
dpkg-sig --sign builder mbash-0.1.deb
```

Ajouter le package au dépôt :

```
reprepro -b ~/debian-repo includedeb stable ~/Desktop/mbash-0.1.deb
```

Liez le dépôt à Apache :

```
sudo ln -s ~/debian-repo /var/www/html/debian
```

```
sudo systemctl restart apache2
```

(parce que j'en ai eu beaucoup à cause du port occupé par une ancienne SAE)
si erreur avec apache :

```
sudo netstat -tuln | grep ':80|:443'
```

```
sudo kill -9
```

```
sudo lsof -i :80
```

```
ps -fp 48524
```

(ancienne SAE)

```
sudo systemctl stop myweb.service
```

```
systemctl restart apache2
```

Le dépôt est maintenant accessible à <http://127.0.0.1/debian>.

Exporter la clé publique :

```
gpg --export --armor ABCD1234 > public.key
```

Partie Client : Requête au dépôt Debian

ajouter dans /etc/apt/sources.list

deb [signed-by=/usr/share/keyrings/mbash.gpg] <http://127.0.0.1/debian> stable main

Importer la clé publique :

wget -O - <http://127.0.0.1/debian/public.key> | sudo tee /usr/share/keyrings/mbash.gpg

sudo apt update

et **ENFIN** :

```
sudo apt install mbash
```

Gestion cycle de vie :

Créer une nouvelle version du package :

- Modifiez la version dans le fichier control (par exemple version: 0.2).
- Recréez et signez le package :

```
dpkg-deb --build mbash-0.2  
dpkg-sig --sign builder mbash-0.2.deb
```

Ajoutez la nouvelle version au dépôt :

reprepro -b ~/debian-repo includedeb stable mbash-0.2.deb

Mettre à jour le client :

sudo apt update

sudo apt upgrade