

**Compte-rendu de la réalisation et du déploiement d'un Shell***SAE 3.03 Réseau et application serveur***SOMMAIRE**

<b>I. Généralités.....</b>	<b>2</b>
<b>II. Mbash : une version miniature de bash.....</b>	<b>3</b>
Étapes de la réalisation.....	3
A. Bilan.....	4
<b>III. Serveur de package Debian.....</b>	<b>5</b>
A. Réalisation.....	5
B. Bilan.....	8
<b>IV. Savoirs mobilisés.....</b>	<b>9</b>
A. Les savoirs.....	9
B. Les savoirs-faire.....	9
C. Les savoirs-être.....	9

## I. Généralités

Dans notre version finale de notre terminal bash miniature, nous avons décidé de pouvoir y faire les commandes suivantes:

- **ls** (liste le contenu du répertoire courant)
- **pwd** (chemin absolu du répertoire courant)
- **man** (affiche le manuel associé à une commande)
- **echo** (affiche du texte)
- **ls -l** (liste du contenu du répertoire de manière courante)
- **mkdir** (créer un dossier vide)
- **touch** (créer un fichier vide)
- **cp** (Copie des fichiers ou des répertoires)
- **mv** (Déplace ou renomme des fichiers ou des répertoires)
- **rm** (Supprime des fichiers ou des répertoires)
- **rmdir** (Supprime un répertoire vide)
- **cat** (Affiche le contenu d'un fichier)
- **nano** (Édite un fichier en utilisant l'éditeur de texte nano)
- **ps** (Affiche les processus en cours d'exécution)
- **ping** (Vérifie la connectivité réseau)
- **ifconfig** (Affiche ou configure les interfaces réseau)
- **clear** (Efface ce qu'il y a sur le terminal)
- **history** (Affiche l'historique des dernières commandes)
- **! suivit d'un chiffre** (joue la commande qui correspond dans l'historique)
- **cd** (Change de répertoire ou de fichier)
- **export PS1 ="*nouveau prompt*"** (changement du prompt)
- **exit** (Sort du terminal et ferme la fenêtre du terminal parent)

Grâce au fonctionnement de certaines fonctionnalités dans *Bash*, certaines commandes telles que *ls*, *man* ... étaient déjà compilables dans notre mini bash.

## II. Mbash : une version miniature de bash

### A. Étapes de la réalisation

Nous avons d'abord créé un dépôt GitHub et y avons importé le programme initial fourni.

Pour commencer nous avons implémenté les commandes principales avec la méthode “**execvp**”.

Pour l'exécution tout se fait dans notre méthode “**execute(char \*cmd[])**” avec **execvp** comme notre exécution par défaut.

Dans cette même méthode, nous avons commencé par faire en sorte de pouvoir sortir du terminal avec **exit**, pour cela il nous a suffit de récupérer le pid du terminal avec **getppid()** et de **kill** le processus de ce dernier.

Pour implémenter la fonction **cd**, nous avons fait une méthode qui s'occupe de changer de directory en utilisant **chdir()**.

Nous avons ensuite créé une variable **prompt** qui est affichée avant chaque commande comme sur un terminal classique. Le **prompt** a par défaut le chemin courant du terminal, il pourra être changé avec “**export PS1 =“nouveau prompt”**”, ce changement est géré par notre fonction **changePrompt(char\* newPrompt)**.

Pour finir nous avons implémenté les commandes **history** et **!**, ses appels sont liés à la gestion d'un historique. Pour ce faire, il nous a fallu créer une structure permettant de stocker la commande et son numéro. Nous avons stocké toutes les commandes effectuées dans un tableau nommé **history**. Lors de l'appel de la commande **history** execute appel la fonction **printHistory()** qui va afficher l'historique. Pour rejouer une commande il suffit d'appeler **!** avec le numéro correspondant.

### B. Bilan

Nous sommes arrivées à implémenter toutes les fonctionnalités demandées. Et cela sans utiliser la méthode **system**.

### **III. Serveur de package Debian**

#### A. Étapes de la réalisation

Dans cette partie, nous avons voulu mettre notre programme ***mbash.c*** sur un serveur pour pouvoir l'installer sur une machine cliente.

Partie Serveur:

<b>Mise en place du dépôt Debian</b>	
<b>Action</b>	<b>Commande</b>
Installation de reprepro, dpkg-sig, apache2	<i>sudo apt-get install dpkg-sig reprepro apache2</i>
Mise en place de l'arborescence	<i>mbash / DEBIAN / control mbash / usr / bin / exécutable mbash</i>
Création du fichier control	Voir image 1
Mise en place de l'arborescence	var / www / html / debian / conf / distribution
Création du fichier distributions	Voir image 2
Génération des répertoires dists et db	<i>reprepro -b . export</i>
Création du fichier Release	<i>apt-ftparchive release . &gt; Release</i>
Vérifier que reprepro soit dans le répertoire usr / bin	<i>which reprepro</i>
Génération de la clé	<i>gpg --gen-key</i>
Génération du fichier .deb	<i>dpkg-deb --build mbash</i>
Garantir que gpg utilise bien le terminal approprié	<i>export GPG_TTY=\$(tty)</i>
Signature du fichier .deb	<i>dpkg-sig -k (ID de la clé créée) --sign builder mbash.deb</i>
Inclusion du fichier Debian (mbash.deb) dans le dépôt géré par Repro	<i>reprepro -b . includedeb generic /home/etudiant/mbash.deb</i>

Image 1:

```
Linux [En fonction] - Oracle VM VirtualBox
GNU nano 6.2
control *

Package: mbash
Version: 0.1
Section: main
Priority: standard
Architecture: amd64
Maintainer: Maëlle Bitsindou Célie Ponroy
Description: Terminal Bash miniature_
```

Image 2:

```
Linux [En fonction] - Oracle VM VirtualBox
GNU nano 6.2
distributions *

Origin: mbash
Label: mbash apt
Suite: stable
Codename: generic
Version: 0.1
Architectures: amd64
Components: main
Description: Installation d'un terminal bash miniature
SignWith: yes_
```

Arborescence du package:

```
etudiant@ubuntu1804:~/debpkgs$ tree
.
├── mbash-package
│   ├── DEBIAN
│   │   └── control
│   └── usr
│       └── bin
│           └── mbash
└── mbash-package.deb

4 directories, 3 files
```

Partie Client:

<b>Mise en place du serveur et du programme</b>	
<b>Action</b>	<b>Commande</b>
Ajout du serveur à la liste des sources APT (Voir image 3)	<code>echo "deb <a href="http://adresse-du-serveur/mbash_repo/">http://adresse-du-serveur/mbash_repo/</a>"   sudo tee -a /etc/apt/sources.list</code>
Authentification du package avec la clé publique (Voir image 4)	<code>wget -O - <a href="http://adresse-du-serveur/mbash_repo/mbash_public_key.asc">http://adresse-du-serveur/mbash_repo/mbash_public_key.asc</a>   sudo apt-key add</code>
Installation du programme mbash	<code>sudo apt update sudo apt install mbash</code>
Mise à jour vers une nouvelle version	<code>sudo apt update sudo apt upgrade</code>

Image 3:

```
# deb-src http://fr.archive.ubuntu.com/ubuntu/ jammy main restricted
## Major bug fix updates produced after the final release of the
## distribution.
deb http://fr.archive.ubuntu.com/ubuntu/ jammy-updates main restricted
# deb-src http://fr.archive.ubuntu.com/ubuntu/ jammy-updates main restricted

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team. Also, please note that software in universe WILL NOT receive any
## review or updates from the Ubuntu security team.
deb http://fr.archive.ubuntu.com/ubuntu/ jammy universe
# deb-src http://fr.archive.ubuntu.com/ubuntu/ jammy universe
deb http://fr.archive.ubuntu.com/ubuntu/ jammy-updates universe
# deb-src http://fr.archive.ubuntu.com/ubuntu/ jammy-updates universe

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
## your rights to use the software. Also, please note that software in
## multiverse WILL NOT receive any review or updates from the Ubuntu
## security team.
deb http://fr.archive.ubuntu.com/ubuntu/ jammy multiverse
# deb-src http://fr.archive.ubuntu.com/ubuntu/ jammy multiverse
deb http://fr.archive.ubuntu.com/ubuntu/ jammy-updates multiverse
# deb-src http://fr.archive.ubuntu.com/ubuntu/ jammy-updates multiverse

## N.B. software from this repository may not have been tested as
## extensively as that contained in the main release, although it includes
## newer versions of some applications which may provide useful features.
## Also, please note that software in backports WILL NOT receive any review
## or updates from the Ubuntu security team.
deb http://fr.archive.ubuntu.com/ubuntu/ jammy-backports main restricted universe multiverse
# deb-src http://fr.archive.ubuntu.com/ubuntu/ jammy-backports main restricted universe multiverse

deb http://security.ubuntu.com/ubuntu jammy-security main restricted
# deb-src http://security.ubuntu.com/ubuntu jammy-security main restricted
deb http://security.ubuntu.com/ubuntu jammy-security universe
# deb-src http://security.ubuntu.com/ubuntu jammy-security universe
deb http://security.ubuntu.com/ubuntu jammy-security multiverse
# deb-src http://security.ubuntu.com/ubuntu jammy-security multiverse

# This system was installed using small removable media
# (e.g. netinst, live or single CD). The matching "deb cdrom"
# entries were disabled at the end of the installation process.
# For information about how to configure apt package sources,
# see the sources.list(5) manual.
deb [signed-by=/etc/apt/keyrings/mbash.gpg] http://192.168.156.185/apt/debian/mBash generic main
```

Maëlle Bitsindou S3D	Célie Ponroy S3D
----------------------	------------------

#### Image 4:

```
root@Linux:~# wget -qO- http://192.168.185/apt/debian/mBash/key/public.key | gpg --dearmor | sudo tee /etc/apt/keysings/mbash.gpg
-----BEGIN PGP PUBLIC KEY BLOCK-----
-----END PGP PUBLIC KEY BLOCK-----
root@Linux:~# apt-key add /etc/apt/keysings/mbash.gpg
root@Linux:~# apt update
root@Linux:~# apt install mbash
root@Linux:~# mbash
mbash 1.0.0 (2018-07-10) - [https://github.com/mbash-project/mbash]
Copyright (C) 2018 mbash project

Usage: mbash [-c command] [-f file] [-l log_level] [-v] [-V] [-h]

Options:
  -c command      Run command as root
  -f file         Run file as root
  -l log_level    Set log level (info, warning, error, critical)
  -v              Print version information
  -V              Print version information and exit
  -h              Print help information and exit

Version: 1.0.0 (2018-07-10)
```

## B. Bilan

En résumé, la mise en place du serveur de packages Debian a été un succès, avec la création d'un dépôt sécurisé, la configuration efficace d'un serveur HTTP via Apache2, et la génération de clés pour assurer l'intégrité des packages. Ces mesures garantissent un déploiement fiable et une gestion efficace des mises à jour du programme ***mbash*** pour les utilisateurs clients.

**IV. Savoirs mobilisés****A. Les savoirs**

Dans le cadre de cette SAE, nous avons acquis une compréhension approfondie des concepts fondamentaux de programmation en langage C.

Nous avons développé des connaissances approfondies sur la mise en œuvre de commandes de base du shell, telles que cd, pwd, et la gestion des processus en arrière-plan vu en cours de système.

Maintenant, nous sommes familiarisés avec les différentes méthodes d'exécution de commandes en utilisant execve, execvp, et la manipulation des variables d'environnement.

Notre savoir inclut également une compréhension des aspects de base du fonctionnement du système Debian et la création de packages Debian que nous avons pu mettre en œuvre pour l'installation du programme.

**B. Les savoirs-faire**

Nous avons développé des compétences solides en programmation en C, en particulier dans la création d'un programme compact et fonctionnel répondant aux exigences spécifiques de la version miniature de Bash (mbash).

La mise en œuvre de commandes telles que cd, pwd, et la gestion des processus en arrière-plan a renforcé nos compétences en manipulation des systèmes d'exploitation. La création d'un package Debian nous a permis de maîtriser les outils tels que dpkg-sig et reprepro, ainsi que de comprendre le processus de signature des packages avec des clés publique et privée.

Enfin, nous avons acquis des compétences en configuration de serveurs HTTP, comme Apache 2, pour rendre le serveur de package accessible.

**C. Les savoirs-être**

Notre engagement dans ce projet en binôme a renforcé notre capacité à collaborer efficacement, en partageant des idées et en résolvant des problèmes de manière collaborative.

La gestion du temps a été cruciale pour atteindre les objectifs fixés, ce qui a renforcé notre capacité à prioriser les tâches et à respecter les délais.

Nous avons démontré une attitude proactive en explorant différentes approches pour résoudre les problèmes, en utilisant tant la créativité pour le montage de la vidéo, que la rigueur technique pour l'élaboration du projet en lui-même.

La gestion des défis techniques et la recherche de solutions nous ont permis de développer de la résilience et de la persévérance dans un environnement de programmation assez complexe.