

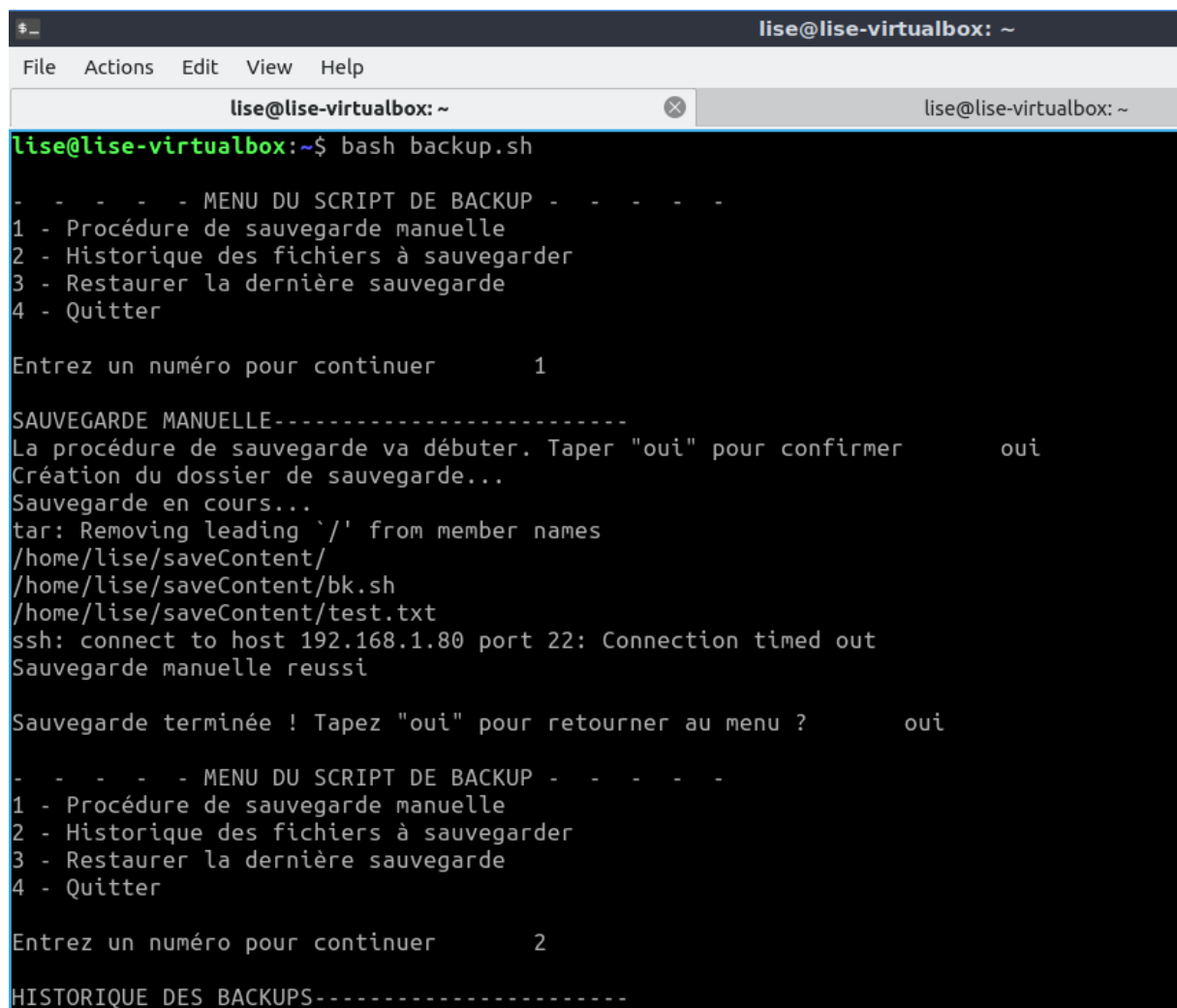
GESTION DU PATRIMOINE INFORMATIQUE

*GESTION DES
SAUVEGARDES*

DESCRIPTION DU PROJET

Dans le cadre d'un projet au cours de ma première année de BTS, j'ai réalisé un script de **sauvegarde et de restauration de fichiers** sur Linux à travers une machine virtuelle sous *Ubuntu* grâce au logiciel de virtualisation *VirtualBox*.

En apprenant et en utilisant le langage **bash**, j'ai créé un fichier nommé « backup.sh » qui permet à l'utilisateur de lancer la sauvegarde en tapant la commande « bash backup.sh » et puis il va suivre l'instruction affichée sur l'écran pour exécuter les actions comme le montre l'exemple ci-dessous :



```
lise@lise-virtualbox: ~  
File Actions Edit View Help  
lise@lise-virtualbox: ~  
lise@lise-virtualbox:~$ bash backup.sh  
- - - - - MENU DU SCRIPT DE BACKUP - - - - -  
1 - Procédure de sauvegarde manuelle  
2 - Historique des fichiers à sauvegarder  
3 - Restaurer la dernière sauvegarde  
4 - Quitter  
Entrez un numéro pour continuer      1  
SAUVEGARDE MANUELLE-----  
La procédure de sauvegarde va débuter. Taper "oui" pour confirmer      oui  
Création du dossier de sauvegarde...  
Sauvegarde en cours...  
tar: Removing leading `/' from member names  
/home/lise/saveContent/  
/home/lise/saveContent/bk.sh  
/home/lise/saveContent/test.txt  
ssh: connect to host 192.168.1.80 port 22: Connection timed out  
Sauvegarde manuelle réussi  
Sauvegarde terminée ! Tapez "oui" pour retourner au menu ?      oui  
- - - - - MENU DU SCRIPT DE BACKUP - - - - -  
1 - Procédure de sauvegarde manuelle  
2 - Historique des fichiers à sauvegarder  
3 - Restaurer la dernière sauvegarde  
4 - Quitter  
Entrez un numéro pour continuer      2  
HISTORIQUE DES BACKUPS-----
```

Les actions possibles sont accessibles en tapant le numéro correspondant :

1. Lancer la sauvegarde
2. Voir l'historique des fichiers à sauvegarde
3. Restaurer la dernière sauvegarde
4. Quitter le processus de sauvegarde

```

- - - - - MENU DU SCRIPT DE BACKUP - - - - -
1 - Procédure de sauvegarde manuelle
2 - Historique des fichiers à sauvegarder
3 - Restaurer la dernière sauvegarde
4 - Quitter

Entrez un numéro pour continuer

```

Pour ce qui est du code, tapez vous « nano backup.sh » pour accéder au code source. Il est intégralement fait en bash et s'adapte à chaque machine, créant si nécessaire les répertoires de sauvegarde dont il aura besoin pour effectuer sa tâche.

```

lise@lise-virtualbox: ~
File Actions Edit View Help
lise@lise-virtualbox: ~ backup.sh
GNU nano 4.8
#!/bin/bash

# Variables
moi=`whoami`
saveRep="/home/${moi}/saveContent"

# Vérification de la présence des fichiers prioritaires
if [ -f "./saveLog.txt" ];then
    # Indice de début de logs
    echo "Nouvelle génération de logs" >> saveLog.txt
    # Création du répertoire de sauvegarde au cas il y serait inexistant
    mkdir -p $saveRep
else
    # Création du fichier de logs et du répertoire à sauvegarder
    mkdir -p $saveRep
    touch saveLog.txt
    echo "GESTION DES LOGS SAUVEGARDE/RESTAURATION" > saveLog.txt
    echo "" >> saveLog.txt
    echo "" >> saveLog.txt
fi

# Fonction de sauvegarde manuelle
function backup()
{
    # Vérification de sécurité du programme choisi
    read -p "La procédure de sauvegarde va débuter. Taper \"oui\" pour confirmer

    # Log de la confirmation

```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
 ^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To L

Pour plus de performance, j'ai opté pour la création de fonctions de traitement des différentes fonctionnalités :

1. Pour faire la sauvegarde

```

# Fonction de sauvegarde manuelle
function backup()
{
    # Vérification de sécurité du programme choisi
    read -p "La procédure de sauvegarde va débiter. Taper \"oui\" pour confirmer

    # Log de la confirmation
    echo "[ $date ] Vérification de la confirmation : $verification" >> saveLog.t

    # Vérification de la confirmation
    if [ $verification = "oui" ]
    then

        # Répertoire de sauvegarde
        echo "Création du dossier de sauvegarde..."
        aujourd'hui=$(date +%Y-%m-%d)
        repertoire="${aujourd'hui}_${moi}"
        saved="/home/${moi}/saveBackup/${repertoire}"

        # Sauvegarde des fichiers
        echo "Sauvegarde en cours..."
        tar zcvf - $saveRep | sshpass -p "Sontrapkp" ssh lise@192.168.1.80 "cat
        echo "[" $(date) "]" -- Sauvegarde manuelle -- Tâche terminée " >> sav
        echo "Sauvegarde manuelle réussi"
        echo ""

        # Message de redirection

```

2. Pour afficher l'historique des sauvegardes

```

# Fonction historique des fichiers à sauvegarder
function historique()
{
    # Affichage de l'historique des backups
    ls $saveRep
    echo ""

    # Message de redirection
    read -p "Tapez \"menu\" pour retourner au menu" direction

    # Boucle de vérification de saisie
    while [ $direction != "menu" ]
    do
        # Message de redirection
        read -p "Tapez \"menu\" pour retourner au menu" direction
    done

    # Redirection vers le menu
    if [ $direction = "menu" ]
    then
        echo "Vous allez être redirigé vers le menu..."
        echo "[" $(date) "]" Retour au menu du script" >> saveLog.txt
        echo ""
        menu
    fi
}

```

3. Pour restaurer la dernière sauvegarde

```

function restaurer(){
    # Restauration des sauvegardes
    server="lise@192.168.1.80"
    sshpass -p "Sontrapkp" scp -r $server:~/saveBackup ~/
    echo "Restauration terminée !"
    echo ""
    echo "[" $(date) "] Restauration terminée">>saveLog.txt

    # Message de redirection
    read -p "Tapez \"menu\" pour retourner au menu" direction
    # Boucle de vérification de saisie
    while [ $direction != "menu" ]
    do
        # Message de redirection
        read -p "Tapez \"menu\" pour retourner au menu" direction
    done

    # Redirection vers le menu
    if [ $direction = "menu" ]
    then
        echo "Vous allez être redirigé vers le menu..."
        echo "[" $(date) "] Retour au menu du script" >> saveLog.txt
        echo ""
        menu
    fi
}

```

4. Pour afficher le menu

```

function menu()
{
    # Contenu du menu
    ligne1="Procédure de sauvegarde manuelle"
    ligne2="Historique des fichiers à sauvegarder"
    ligne3="Restaurer la dernière sauvegarde"

    # Affichage du menu
    echo ""
    echo "- - - - - MENU DU SCRIPT DE BACKUP - - - - -"
    echo "1 - $ligne1"; echo "2 - $ligne2"; echo "3 - $ligne3"; echo "4 - Quitter"
    echo ""

    # Demande de choix et lecture de la saisie
    read -p "Entrez un numéro pour continuer" choix

    # Log du menu ouvert
    echo "[" $(date) "] Numéro du menu ouvert par l'utilisateur : $choix" >> save

    # Boucle sur le choix de l'utilisateur
    while [ ! $choix = 1 -a ! $choix = 2 -a ! $choix = 3 -a ! $choix = 4 ]
    do
        # Message affiché dans la boucle
        read -p "Entrez un numéro valide pour continuer !" choix
        # Log de chaque saisie de l'utilisateur dans la boucle
        echo "[" $(date) "] Numéro du menu ouvert dans la boucle par l'utilisateur : $choix" >> save
    done
}

```

GESTION DES LOGS

Pour contrôler le processus de traitement et intervenir en cas de problème, un système de log a été intégré au code afin d'enregistrer les interactions de l'utilisateur, de récupérer et de stocker des données à chaque évolution du script de manière transparente et ainsi de pouvoir l'étudier pour réagir plus efficacement en cas de bug. Ils sont tous loggés dans le fichier nommé « saveLog.txt », juste tapez la commande « nano saveLog.txt » pour le consulter.

```
lise@lise-virtualbox: ~  
File Actions Edit View Help  
lise@lise-virtualbox: ~  
lise@lise-virtualbox:~$ ls  
backupauto.sh backup.sh Documents home Pictures saveContent Templates  
backup_dirs.txt Desktop Downloads Music Public saveLog.txt Videos  
lise@lise-virtualbox:~$
```

```
GNU nano 4.8 saveLog.txt  
GESTION DES LOGS SAUVEGARDE/RESTAURATION  
  
[ mer. 07 juil. 2021 15:29:01 CEST ] Numéro du menu ouvert par l'utilisateur :  
Nouvelle génération de logs  
[ mer. 07 juil. 2021 15:36:01 CEST ] Numéro du menu ouvert par l'utilisateur :  
Nouvelle génération de logs  
[ mer. 07 juil. 2021 15:44:01 CEST ] Numéro du menu ouvert par l'utilisateur :  
Nouvelle génération de logs  
[ mer. 07 juil. 2021 15:49:01 CEST ] Numéro du menu ouvert par l'utilisateur :  
Nouvelle génération de logs  
[ ven. 25 mars 2022 15:44:53 CET ] Numéro du menu ouvert par l'utilisateur : 1  
[ ven. 25 mars 2022 15:44:53 CET ] Processus de sauvegarde manuelle activé  
[ ] Vérification de la confirmation : oui  
Nouvelle génération de logs  
[ ven. 25 mars 2022 15:46:19 CET ] Numéro du menu ouvert par l'utilisateur : 1  
[ ven. 25 mars 2022 15:46:19 CET ] Processus de sauvegarde manuelle activé  
[ ] Vérification de la confirmation : oui  
[ ven. 25 mars 2022 15:48:31 CET ] -- Sauvegarde manuelle -- Tâche terminée  
[ ven. 25 mars 2022 15:50:53 CET ] Retour au menu du script  
[ ven. 25 mars 2022 15:50:56 CET ] Numéro du menu ouvert par l'utilisateur : 2  
[ ven. 25 mars 2022 15:50:56 CET ] Affichage de l'historique des fichier à sauvegarde  
[ ven. 25 mars 2022 15:51:03 CET ] Retour au menu du script  
[ ven. 25 mars 2022 15:57:21 CET ] Numéro du menu ouvert par l'utilisateur : 4  
[ ven. 25 mars 2022 15:57:21 CET ] Fermeture du menu  
[ ven. 25 mars 2022 15:57:21 CET ] Fermeture du script
```

La sauvegarde automatique implémente elle aussi ce système de log qui gère également les erreurs.