

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 ainsi pu réaliser une interface de sauvegarde qui se déclenche manuellement comme le montre l'exemple ci-dessous :

```
dodde@~$ 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...
Sauvegarde terminée ! 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-----
test.txt
```

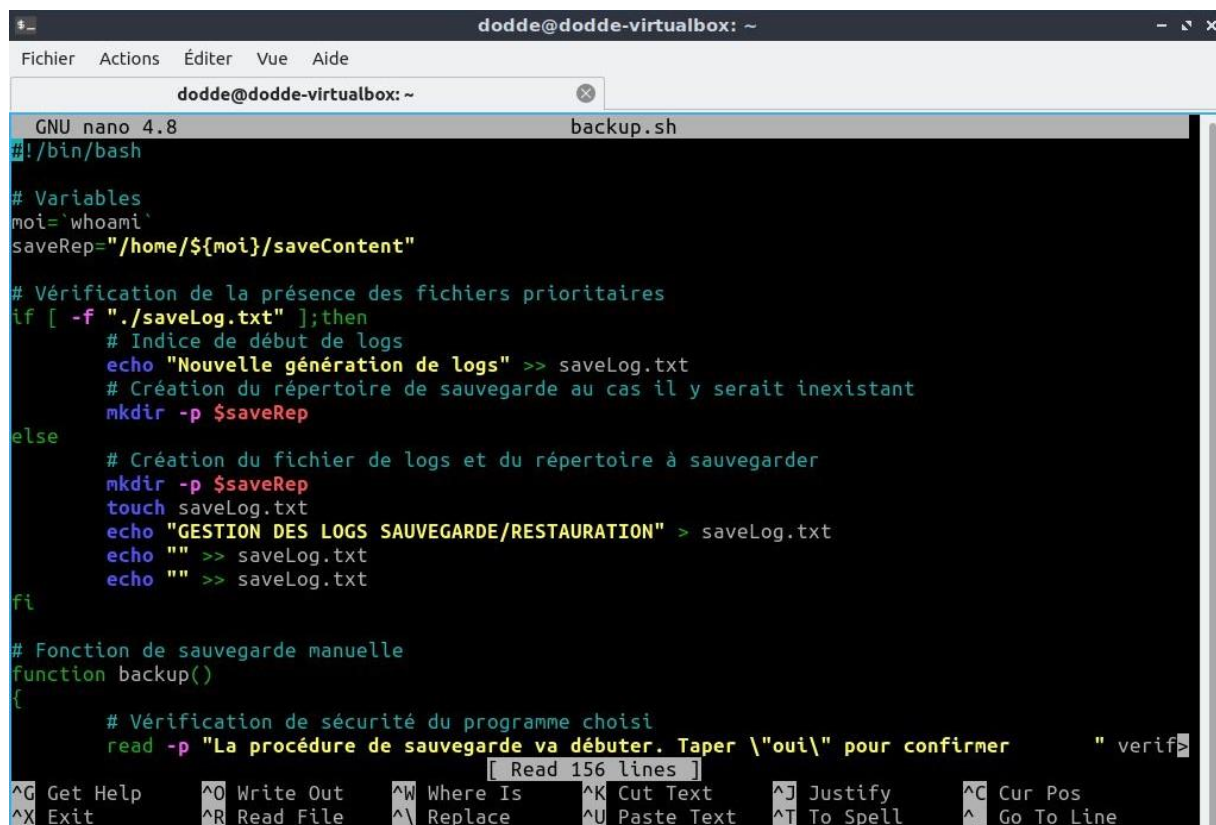
En suivant les étapes nécessaires à la sauvegarde, l'utilisateur peut ainsi lancer le processus de sauvegarde manuelle et/ou voir s'il le désire, l'historique des fichiers du répertoire cible à sauvegarder. La sauvegarde automatique, elle, se déclenche sans intervention humaine.

```
- - - - - 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      4

_____Fin du script_____
dodde@~$ █
```

Pour ce qui est du code, 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.



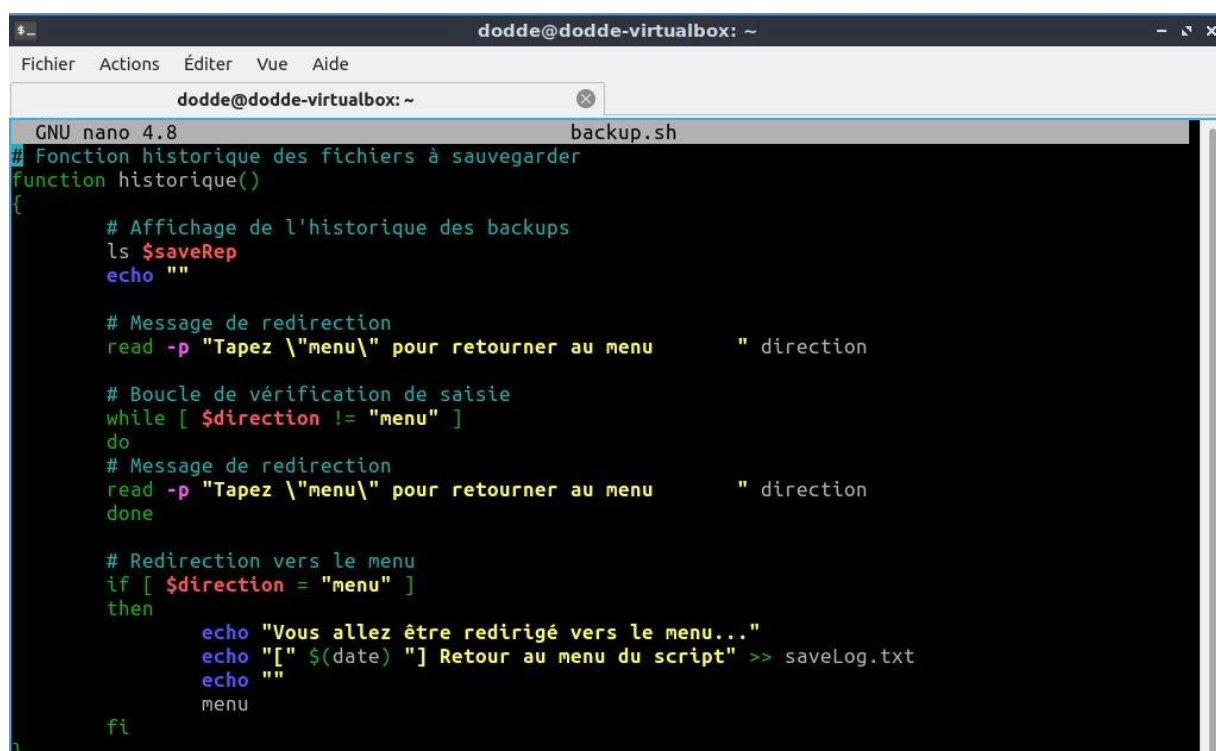
```
#!/bin/bash

# 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" verif
```

Pour plus de performance, j'ai opté pour la création de fonctions de traitement des différentes fonctionnalités dont ci-dessous celles de l'historique des fichiers et de la sauvegarde des fichiers :



```
## 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
}
```

```
dodde@dodde-virtualbox: ~
Fichier Actions Éditer Vue Aide
dodde@dodde-virtualbox: ~
GNU nano 4.8 backup.sh
# Fonction de sauvegarde manuelle
function backup()
{
    # Vérification de sécurité du programme choisi
    read -p "La procédure de sauvegarde va débuter. Taper \"oui\" pour confirmer" "verif"

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

    # 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}"
        mkdir -p $saved || echo "[" $(date) "]" Erreur lors de la création du dossier de sa>

        # Sauvegarde des fichiers
        echo "Sauvegarde en cours..."
        cp -r $saveRep $saved || echo "[" $(date) "]" Erreur lors de la sauvegarde" >> save>
        echo "[" $(date) "]" -- Sauvegarde manuelle -- Tâche terminée " >> saveLog.txt

        # Message de redirection

^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 Line
1 2 3 4          dodde@dod...ualbox: ~ 11:03
```

```
dodde@dodde-virtualbox: ~
Fichier Actions Éditer Vue Aide
dodde@dodde-virtualbox: ~
GNU nano 4.8 backup.sh
# Fonction d'affichage du 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 ""

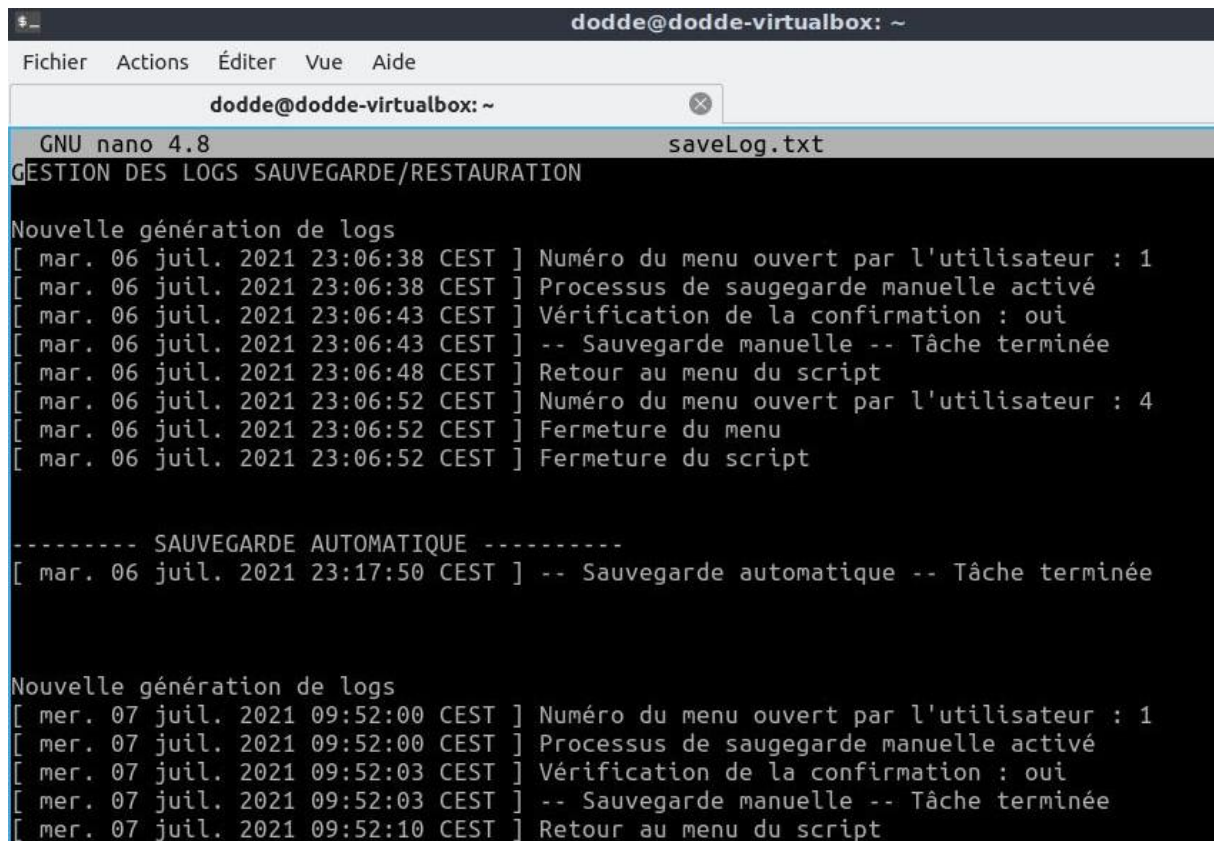
    # Demande de choix et lecture de la saisie
    read -p "Entrez un numéro pour continuer" "choix"

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

    # 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"
    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 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.



The screenshot shows a terminal window titled 'dodde@dodde-virtualbox: ~' with a menu bar (Fichier, Actions, Éditer, Vue, Aide) and a tab labeled 'dodde@dodde-virtualbox: ~'. The editor is GNU nano 4.8 editing 'saveLog.txt'. The file content is as follows:

```
GESTION DES LOGS SAUVEGARDE/RESTAURATION

Nouvelle génération de logs
[ mar. 06 juil. 2021 23:06:38 CEST ] Numéro du menu ouvert par l'utilisateur : 1
[ mar. 06 juil. 2021 23:06:38 CEST ] Processus de sauvegarde manuelle activé
[ mar. 06 juil. 2021 23:06:43 CEST ] Vérification de la confirmation : oui
[ mar. 06 juil. 2021 23:06:43 CEST ] -- Sauvegarde manuelle -- Tâche terminée
[ mar. 06 juil. 2021 23:06:48 CEST ] Retour au menu du script
[ mar. 06 juil. 2021 23:06:52 CEST ] Numéro du menu ouvert par l'utilisateur : 4
[ mar. 06 juil. 2021 23:06:52 CEST ] Fermeture du menu
[ mar. 06 juil. 2021 23:06:52 CEST ] Fermeture du script

----- SAUVEGARDE AUTOMATIQUE -----
[ mar. 06 juil. 2021 23:17:50 CEST ] -- Sauvegarde automatique -- Tâche terminée

Nouvelle génération de logs
[ mer. 07 juil. 2021 09:52:00 CEST ] Numéro du menu ouvert par l'utilisateur : 1
[ mer. 07 juil. 2021 09:52:00 CEST ] Processus de sauvegarde manuelle activé
[ mer. 07 juil. 2021 09:52:03 CEST ] Vérification de la confirmation : oui
[ mer. 07 juil. 2021 09:52:03 CEST ] -- Sauvegarde manuelle -- Tâche terminée
[ mer. 07 juil. 2021 09:52:10 CEST ] Retour au menu du script
```

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