

# Plan du projet

## Objectif

Créer un script `sauvegarde.sh` permettant de faire l'archive des fichiers de votre répertoire personnel.

## Fonctionnalités

1. **Afficher l'usage du script**
2. **Tester la présence d'au moins un argument**
3. **Afficher l'aide à partir d'un fichier texte**
4. **Afficher le nombre de fichiers et la taille totale des fichiers modifiés dans les dernières 24 heures**
5. **Archiver les fichiers modifiés dans les dernières 24 heures**
6. **Options du script**

## Script complet

```
#!/bin/bash

# Fonction pour afficher l'usage
show_usage() {
    echo "sauvegarde.sh: [-h] [-g] [-m] [-v] [-n] [-r] [-a] [-s] chemin.."
}

# Fonction pour afficher l'aide
HELP() {
    cat help.txt
}

# Fonction pour afficher le nombre de fichiers et la taille totale des fichiers
# modifiés dans les dernières 24 heures
afficher_nombre_taille() {
    local chemin="$1"
    find "$chemin" -type f -mtime -1 -exec ls -lh {} + | awk '{ print $9 ": " $5 }'
    find "$chemin" -type f -mtime -1 -exec du -ch {} + | grep total$
}

# Fonction pour archiver les fichiers modifiés dans les dernières 24 heures
archiver_fichiers() {
    local chemin="$1"
    local archive_name="backup_$(date +%Y%m%d_%H%M%S).tar.gz"
    find "$chemin" -type f -mtime -1 -print0 | tar --null -czvf "$archive_name" --
files-from -
    echo "Fichiers archivés dans $archive_name"
}

# Fonction pour renommer l'archive avec la date et l'heure de la modification
renommer_archive() {
    local archive_name="$1"
    local new_archive_name="backup_$(date +%Y%m%d_%H%M%S)_modified.tar.gz"
```

```

mv "$archive_name" "$new_archive_name"
echo "Archive renommée en $new_archive_name"
}

# Fonction pour sauvegarder les informations sur les fichiers archivés
sauvegarder_infos() {
    local fichier="$1"
    local chemin="$2"
    find "$chemin" -type f -mtime -1 -exec ls -lh {} + > "$fichier"
    echo "Informations sauvegardées dans $fichier"
}

# Fonction pour afficher le menu textuel
afficher_menu() {
    while true; do
        echo "Menu :"
        echo "1. Afficher l'usage"
        echo "2. Afficher l'aide"
        echo "3. Afficher le nombre de fichiers et la taille totale"
        echo "4. Archiver les fichiers"
        echo "5. Renommer l'archive"
        echo "6. Sauvegarder les informations"
        echo "7. Quitter"
        read -p "Choisissez une option : " choix
        case $choix in
            1) show_usage ;;
            2) HELP ;;
            3) read -p "Entrez le chemin : " chemin
                afficher_nombre_taille "$chemin" ;;
            4) read -p "Entrez le chemin : " chemin
                archiver_fichiers "$chemin" ;;
            5) read -p "Entrez le nom de l'archive : " archive_name
                renommer_archive "$archive_name" ;;
            6) read -p "Entrez le fichier de sauvegarde : " fichier
                read -p "Entrez le chemin : " chemin
                sauvegarder_infos "$fichier" "$chemin" ;;
            7) exit 0 ;;
            *) echo "Choix invalide" ;;
        esac
    done
}

# Fonction pour afficher un menu graphique (utilisation de YAD)
afficher_menu_graphique() {
    yad --form --title "Sauvegarde" --field "Option":CB "Afficher l'usage!Afficher
l'aide!Afficher le nombre de fichiers et la taille totale!Archiver les
fichiers!Renommer l'archive!Sauvegarder les informations!Quitter" | while read
choix; do
        case $choix in
            "Afficher l'usage") show_usage ;;
            "Afficher l'aide") HELP ;;
            "Afficher le nombre de fichiers et la taille totale")
                chemin=$(yad --entry --title "Chemin" --text "Entrez le chemin :")
                afficher_nombre_taille "$chemin" ;;
        esac
    done
}

```

```

        "Archiver les fichiers")
            chemin=$(yad --entry --title "Chemin" --text "Entrez le chemin :")
            archiver_fichiers "$chemin" ;;
        "Renommer l'archive")
            archive_name=$(yad --entry --title "Archive" --text "Entrez le nom
de l'archive :")
            renommer_archive "$archive_name" ;;
        "Sauvegarder les informations")
            fichier=$(yad --entry --title "Fichier de sauvegarde" --text
"Entrez le fichier de sauvegarde :")
            chemin=$(yad --entry --title "Chemin" --text "Entrez le chemin :")
            sauvegarder_infos "$fichier" "$chemin" ;;
        "Quitter") exit 0 ;;
        *) yad --text "Choix invalide" ;;
    esac
done
}

# Vérifier la présence d'au moins un argument
if [ $# -lt 1 ]; then
    show_usage >&2
    exit 1
fi

# Traiter les options
while getopts ":hnargms:v" opt; do
    case ${opt} in
        h) HELP ;;
        n) read -p "Entrez le chemin : " chemin
            afficher_nombre_taille "$chemin" ;;
        a) read -p "Entrez le chemin : " chemin
            archiver_fichiers "$chemin" ;;
        r) read -p "Entrez le nom de l'archive : " archive_name
            renommer_archive "$archive_name" ;;
        s) read -p "Entrez le fichier de sauvegarde : " fichier
            read -p "Entrez le chemin : " chemin
            sauvegarder_infos "$fichier" "$chemin" ;;
        m) afficher_menu ;;
        g) afficher_menu_graphique ;;
        v) echo "Nom des auteurs : John Doe, Version du code : 1.0" ;;
        *) show_usage >&2
            exit 1 ;;
    esac
done

```

## Détails des étapes

### 1. Afficher l'usage du script :

- La fonction `show_usage` affiche un message d'utilisation sur la sortie standard.

### 2. Tester la présence d'au moins un argument :

- Si aucun argument n'est passé, `show_usage` est appelé et le script se termine avec une erreur.

### 3. Afficher l'aide à partir d'un fichier texte :

- La fonction `HELP` affiche le contenu d'un fichier texte `help.txt`.

### 4. Afficher le nombre de fichiers et la taille totale des fichiers modifiés dans les dernières 24 heures :

- La fonction `afficher_nombre_taille` utilise `find` et `ls` pour afficher les fichiers modifiés et leur taille.

### 5. Archiver les fichiers modifiés dans les dernières 24 heures :

- La fonction `archiver_fichiers` crée une archive `tar.gz` des fichiers modifiés dans les dernières 24 heures.

### 6. Options du script :

- Les options du script sont gérées avec `getopts` pour inclure toutes les fonctionnalités demandées.