

# LE BILAN

---

## PROGRAMMATION SHELL

**Un gestionnaire de « To-do *list* »**

**Projet réalisé par :**

**BOUDRISS Meriem 11808654 et LAMHATTAT Reda 11809054**

**1<sup>ere</sup> année double licence mathématiques et informatique**

**Professeurs :**

**Hiba Ouni, Pierre Rousselin**

**23/04/2019**

## 1. Le but de notre projet

Notre projet propose un gestionnaire de Todolist avec des options basiques (ajouter des tâches, supprimer, afficher la liste) et d'autres un peu plus intéressantes (supprimer une tâche avec un délai, crypter la liste avec un mot de passe et la décrypter). Ce programme permet de gérer plusieurs listes, par exemple on peut se situer dans une liste précise et au même temps ajouter des entrées dans une autre liste ou afficher son contenu.

Nous avons choisi le sujet de la Todolist puisqu'il offre plusieurs options et permet d'utiliser son imagination.

## 2. Représentation générale du programme

### - Le script principale

Les options disponibles :

Notre programme propose 22 options pour pouvoir manipuler les différentes listes créées par l'utilisateur, on présente ici quelques unes :

➔ Cryptage d'une liste :

```
*crypt*) #crypter une Todolist avec un mot de passe
echo ""
ls ${PWD}/les_listes
echo "-----"
read -p "Choisir la liste à crypter: " crplist
if [ -r ${PWD}/les_listes/${crplist} ];then
    read -s -p "saisir le mot de passe: " mdp; echo
    # -s permet de ne pas afficher le mot de passe
    # echo pour un saut de ligne
    if [ -e ${PWD}/les_listes/.mm ];then
        chmod 644 ${PWD}/les_listes/.mm
    else
        touch ${PWD}/les_listes/.mm
    fi
    echo "${crplist}=${mdp}" >> ${PWD}/les_listes/.mm
    #le fichier .mm contient les mots de passe et il est invisible
    chmod 0 ${PWD}/les_listes/.mm
    #protéger le fichier .mm
    chmod 0 "${PWD}/les_listes/${crplist}"
    echo "----La Todolist ${crplist} a été protégé----"
else
    echo "----${crplist} est déjà cryptée!----"
fi;;
```

Cette option permet de protéger une Todolist par un mot de passe qui sera demandé à l'utilisateur, le mot de passe sera invisible grâce à l'option `-s` de la commande `read`.

Une fois le mot de passe saisi la liste concernée est privée de tous ses droits, ainsi le mot de passe est enregistré dans un fichier caché et lui aussi sera privé des permissions.

## ➔ Décryptage d'une liste:

```
*decrypt*) #décrypter une Todolist
echo ""
ls ${PWD}/les_listes
echo "-----"
read -p "Choisir la liste à décrypter: " crplist
if ! [ -e ${PWD}/les_listes/${crplist} ];then
    echo "Cette liste n'existe pas"
else
    if [ -r ${PWD}/les_listes/${crplist} ];then
        echo "Cette liste n'est pas cryptée!"
    else
        read -s -p "Saisir le mot de passe: " mdpbis; echo
        chmod 644 ${PWD}/les_listes/.mm
        recup=$( grep "${crplist}" ${PWD}/les_listes/.mm | cut -d= -f2 )
        while [[ ${mdpbis} != ${recup} ]];do
            echo "MOT DE PASSE INCORRECT!!"
            read -s -p "Saisir le mot de passe: " mdpbis; echo
        done
        echo "mot de passe valide"
        chmod 644 "${PWD}/les_listes/${crplist}"
        echo "----${crplist} a été décrypté----"
        sed -i "${crplist}/d" ${PWD}/les_listes/.mm
        chmod 0 ${PWD}/les_listes/.mm #re-protéger le fichier des MDP
    fi
fi;;
```

Pour décrypter une liste, on teste d'abord si la liste est déjà cryptée, sinon un message d'avertissement est affiché, une fois le test est valide on demande à l'utilisateur de saisir le mot de passe, qui doit être identique à celui sauvegardé dans le fichier (on attribue les droits nécessaires au fichier pour pouvoir récupérer le mot de passe), si le mot de passe est correct, la liste concernée est décryptée on lui ajoutant les permissions par défaut d'un fichier, ensuite on supprime la ligne qui contient l'ancien MDP pour éviter d'avoir un fichier avec plusieurs MDP. Finalement, on re-protège le fichier des MDP.

## ➔ Suppression avec delai :

```
*auto*) #supprimer des tâches avec un délai précis
if ! [ -r ${nom} ];then
    echo ""
    echo "Cette liste est protégée!"
    echo "-----pour l'ouvrir utiliser: --decrypte "
else
    cat -n $nom
    echo ""
    read -p "quel est le numero de la ligne que vous voulez supprimer avec un délai: " num
    if [ $num -gt $(grep -c ".*" $nom) ] ; then
        echo "cette ligne n'existe pas!"
    else
        read -p "veuillez entrez le mois (1-12): " mois
        read -p "veuillez entrez le jour (1-31): " jour
        read -p "veuillez entrez l'heure (00-23): " heure
        read -p "veuillez entrez la munité (1-59): " minute
        echo "la tache -- $(sed -n "$num p" $nom) -- sera supprimer le $jour/
$mois à $heure:$minute"
        sed -i "$num i $(sed -n "$num p" $nom) -- sera supprimer le $jour/
$mois à $heure:$minute" $nom
        sed -i "$(($num+1)) d" $nom
        crontab < <(crontab -l ; echo "$minute $heure $jour $mois 0 sed -i \"\$
$(grep -n \"sera supprimer le $jour/$mois à $heure:$minute\" $nom | cut -d: -
f1) d\" $nom" )
    fi
fi;;
```

Pour sélectionner la tâche à supprimer avec un délai, on teste d'abord si la liste est protégée, sinon on demande à l'utilisateur le numéro correspondant à la ligne souhaitée, ensuite on a mis en place un autre test pour bien s'assurer du numéro communiquée, si tout est valide, l'utilisateur doit entrer la date précise, par conséquent la tâche sera supprimée, on attendant elle sera marquée avec la phrase suivante : « ...sera supprimée le (date et heure exacte) » .

→ L'ajout d'une nouvelle liste :

```
*newlist*) #créer une nouvelle liste
read -p "Quel est le nom de la nouvelle liste " namelist
touch ${namelist}
mv ${namelist} ${PWD}/les_listes
nom="${PWD}/les_listes/${namelist}"
echo "-----Vous êtes dans la liste ${namelist}-----";;
```

L'option « newlist » permet de créer une nouvelle liste qui sera ajoutée dans le dossier qui contient toutes les Todolist (il contient aussi la liste qui fait du bien), la liste sera donc créée dans le répertoire dédié (les\_listes) et elle deviendra la liste courante de l'utilisateur.

→ Afficher toutes les Todolist :

```
*show*) #Afficher les titres de toutes les Todolist
if [ $(ls ${PWD}/les_listes | wc -w) = "0" ];then
echo "vous avez aucune liste de tâches!!"
else
echo " "
echo "total : $( ls ${PWD}/les_listes | wc -w )"
echo " "
ls ${PWD}/les_listes
echo " "
echo "(pour afficher le contenu d'une liste utilisez la commande \"--
catlist\") "
fi;;
```

L'option « show » permet d'afficher les titres de toutes les Todolist du répertoire « les\_listes », si le répertoire est vide un message d'information est affiché.

=> Nous vous invitons à découvrir les autres options dans le script.

#### - La page help du script

Une page d'aide est associée au script pour permettre à l'utilisateur de bien manipuler les options, cette page se représente comme une page du manuel.

ATTENTION ! IL FAUT METTRE LA PAGE « help » DANS LE MEME REPERTOIRE QUE LE SCRIPT.

```

Todoliste (script shell)                                # La page d'aide de la todolist

NAME
    todolist.sh - un gestionnaire de To-Do List

DESCRIPTION
    ./todolist.sh - pour exécuter le programme
    -> choix du nom de la Todolist
    ->options:
        =>choisir entre 24 options disponibles (ajouter, supprimer,
        afficher...)

OPTIONS
    Ce programme contient les options suivante:"

    --aj => permet d'ajouter des tâches dans la Todolist, la date d
    est aussi sauvegardée.
    pour ajouter des tâches dansla liste tapez --aj

    --sup => permet de supprimer des tâches de la Todolist, les tâc
    supprimées sont ajoutées automatiquem
    (voir plus loin).
    pour supprimer une tâche de la liste tapez --s

    --aff => permet d'afficher le contenu de la liste courante, le
    total des tâches est aussi affiché.

```

« Help » contient toutes les options du programme avec des explications courtes et le mode d'utilisation

### 3. Un exemple de déroulement et mode d'utilisation :

Pour commencer le programme, il faut d'abord rendre le script exécutable :

```
~$ chmod u+x todolist.sh
```

Ensuite, il faut exécuter le script.

```
~$ ./todolist.sh
```

Le programme commence par demander à l'utilisateur le nom de la Todolist, il faut donc saisir le titre de la liste pour pouvoir après continuer avec les différentes options.

```

quel est le nom de votre liste: liste

Todo --aff

|total : 0 |
-----
aucune tâche à afficher!
-----

Todo ----aj

Ajouter la tâche bonjour tout le monde
-- bonjour tout le monde -- added

Todo --

```

Si la liste est vide, alors le nombre total des tâches est 0 et il y a aucune tâche à afficher.  
Ensuite, on ajoute des tâches on tapant « aj ».

```
Todo --ajj

Ajouter la tâche hello every body
-- hello every body -- added

Todo --aff

|total : 2|
-----
1  bonjour tout le monde -- ajoutée le jeudi 25/04/2019 à 16:27:00
2  hello every body -- ajoutée le jeudi 25/04/2019 à 16:32:11
-----

Todo --fait

bonsoir tout le monde -- ajoutée le lundi 04/22/19 à 23:47:40  a été faite le l
undi 04/22/19 à 23:47:49
une nouvelle liste -- ajoutée le lundi 04/22/19 à 23:49:21  a été faite le lund
i 04/22/19 à 23:49:37
--> hello every body -- ajoutée le lundi 04/22/19 à 23:51:11  a été faite le lu
ndi 04/22/19 à 23:51:59
--> partir chez ma tante -- ajoutée le mercredi 04/24/19 à 13:05:40  -- sera su
pprimer le 04/04 à 22:29 a été faite le mercredi 04/24/19 à 22:31:47
```

Pour afficher la liste du bien, on utilise l'option « fait », on a donc la liste de toutes les tâches déjà faites avec la date d'ajout et aussi la date quand elle a été faite.

```
Todo --vide

etes-vous sur de vouloir vider la liste du bien?(o\n)o

-----la liste fait a été vidé!-----

Todo --
```

Et pour vider la liste du bien, on utilise l'option vide qui demande d'abord la confirmation.

```
Todo --pri

1  bonjour tout le monde -- ajoutée le jeudi 25/04/2019 à 16:27:00
2  hello every body -- ajoutée le jeudi 25/04/2019 à 16:32:11

Quel est le numero de la ligne que vous voulez priorisée: 1
le niveau d'importance (A/a) ou (B/b) ou (C/c) ou (D/d) : A
A bonjour tout le monde -- ajoutée le jeudi 25/04/2019 à 16:27:00  -- has prior
itized

Todo --aff

|total : 2|
-----
1  A bonjour tout le monde -- ajoutée le jeudi 25/04/2019 à 16:27:00
2  hello every body -- ajoutée le jeudi 25/04/2019 à 16:32:11
-----
```

L'option « pri » permet de donner une certaine priorité à une tâche, on choisi donc la ligne concernée et le niveau d'importance. Par conséquent, la lettre qui représente le niveau est ajoutée au début de la ligne. En plus, cette option trie automatiquement les lignes selon l'ordre de priorité, on peut aussi changer la priorité attribuée à une tâche on lui affectant une nouvelle.

=>Nous vous invitons à tester le reste des options.

#### 4. Des commentaires sur des passages du code :

Ce bout de code concerne les listes cryptées, il faut toujours tester si la Todolist est protégée avant de tenter d'ajouter du contenu, l'afficher ou même le supprimer, pour cela on utilise un test avec l'option -r qui permet de vérifier si le fichier a le droit de lecture, on a décidé de tester un seul droit vu que le cryptage enlève tous les droits.

```
if ! [ -r ${nom} ];then
    echo ""
    echo "Cette liste est protégée!"
    echo "-----pour l'ouvrir utiliser: --decrypte "
else
```

Le code source est commenté pour faciliter la lecture