Commander le GPIO du Raspberry Pi en mode texte

En utilisant les commandes ci-dessus, écrivez un script lisant l'état d'un bouton poussoir connecté au GPIO 16.
Ouvrez l'éditeur de texte `nano` :

```
pi@raspberrypi ~ $ nano lit_switch_modif.sh
```

Saisissez le script destiné à lire l'état d'un bouton poussoir connecté sur le GPIO16.
L'état du bouton poussoir n'est affiché que lors d'un changement.

```
#!/bin/bash
# Script lit_switch_modif.sh
# Lire l état d'un bouton poussoir
# Afficher uniquement lorsque le bouton change d état
# Ce script ne comporte pas de test destiné à l'interrompre
# Il s'arrête en appuyant sur CTRL  C

# Effacer l'écran
clear

# Rendre le répertoire /sys/class/gpio actif
cd /sys/class/gpio

# Créer l'accès au port GPIO 16
# Ne rien faire s'il existe déja
if [ -d "gpio16" ]; then
 echo "gpio16 existe dejà"
else
 echo "gpio16 : Création"
 echo 16 > export
fi

# Rendre le répertoire gpio16 actif
cd gpio16/

# Configurer le port GPIO 16 en entrée
# Normalement il y est par défaut… mais on ne sait pas ce qui a pu se passer avant
echo in > direction

# Initialiser la variable x à 1
x="1"

# Boucle while : cette boucle s'exécute tant que
# la condition est vérifiée
# ici la condition est toujours vraie :
# la boucle ne s'interrompt jamais
# Il faudra en sortir avec un CTRL C

    # L'instruction cat affiche le contenu d'un fichier à l'écran
    # Le fichier value contient l'état du bouton poussoir
    # Au lancement du programme on affiche l'état du bouton poussoir
 z="$(cat value)"
 if [ $z = 1 ]; then
     echo "Switch relâché"
 else
     echo "Switch appuyé"
 fi
```

```
while [ $x -gt 0 ]
# Le bloc inclus entre do et done est exécuté
# par la boucle while

# Début du bloc d'instructions
do

    # On mémorise l’état du bouton poussoir dans la variable mem
 let "m = z"

    # On lit à nouveau l’état du bouton poussoir
    # -ne  teste si non égal
 z="$(cat value)"
 if [ "$z" -ne "$m" ]; then
       # Si l’état est différent du précédent, on affiche l’état actuel
     if [ $z = 1 ]; then
          echo "Switch relâché"
     else
          echo "Switch appuyé"
     fi
      # Fin du premier if
 fi

    # Mémoriser nouvelle valeur
    let "m = z"

    # La commande sleep suspend l'exécution du programme
    # pendant la durée spécifiée (en secondes)
    sleep 0.1

# Fin du bloc d'instructions
done
```