

Conception d'interfaces graphiques avec Dialog

1. Les bases

Dialog est un utilitaire permettant de construire des interfaces graphiques basées sur la console. Dialog est présent dans la plupart des distribution GNU/Linux.

Premier script dialog. Il affiche une simple fenêtre OUI/NON.

```
#!/bin/bash
DIALOG=${DIALOG=dialog}

$DIALOG --title " Mon premier dialog" --clear \
        --yesno "Bonjour, ceci est mon premier programme dialog" 10 30

case $? in
    0)    echo "Oui choisi. ";;
    1)    echo "Non choisi. ";;
    255)  echo "Appuyé sur Echap. ";;
esac
```

Les options utilisées sont :

- `--title` : Cette option définit le titre de la fenêtre
- `--clear` : Cette option efface l'écran avant de dessiner
- `--yesno` : Cette option dessine une fenêtre à choix binaire (oui/non) avec le texte indiqué à l'intérieur de la fenêtre.

Le texte à afficher dans la fenêtre *oui/non* est indiqué entre guillemets. Le texte s'étend en fonction de la largeur de la fenêtre ; faites appel à `\n` pour forcer un saut de ligne. Les deux derniers nombres spécifient la largeur et la hauteur de la fenêtre. Pour vous déplacer entre *oui* et *non*, utilisez la touche **Tab**.

Le programme dialog attend maintenant une entrée utilisateur. Quand vous appuyez sur **Entrée** sur *oui* ou *non*, ou si vous appuyez sur la touche **Echap**, le programme se termine et la valeur de retour est disponible dans la variable shell `$?` que vous pouvez traiter par la suite.

2. Lire une entrée

Le programme suivant lit une chaîne de caractères que vous entrez et la réaffiche.

```
#!/bin/sh
DIALOG=${DIALOG=dialog}
fichtemp=`tempfile 2>/dev/null` || fichtemp=/tmp/test$$
trap "rm -f $fichtemp" 0 1 2 5 15
$DIALOG --title "Ma fenêtre d'entrée" --clear \
    --inputbox "Bonjour, voici un exemple de fenêtre d'entrée\n
Essayez de saisir votre nom ci-dessous:" 16 51 2> $fichtemp

valret=$?

case $valret in
    0)
        echo "La chaîne d'entrée est `cat $fichtemp`";
    1)
        echo "Appuyé sur Annuler.";;
    255)
        if test -s $fichtemp ; then
            cat $fichtemp
        else
            echo "Appuyé sur Echap."
        fi
    ;;
esac
```

Essayez de lancer le programme dans la console.

Le programme est un peu plus complexe que notre précédent programme de fenêtre *oui/non*. Les lignes suivantes créent un fichier temporaire :

```
fichtemp=`tempfile 2>/dev/null` || fichtemp=/tmp/test$$
trap "rm -f $fichtemp" 0 1 2 5 15
```

La première ligne ci-dessus essaie de créer un fichier temporaire à l'aide de l'utilitaire `tempfile`. Si ce dernier échoue, un fichier temporaire est configuré manuellement dans `/tmp`. La seconde ligne met en place une routine d'interception. Quand le script se termine (sur erreur ou non), la commande **trap** supprime le fichier temporaire. Les nombres affichés sont les signaux qui seront interceptés.

dialog est alors lancé comme suit :

```
$DIALOG --title "Ma fenêtre d'entrée" --clear \
    --inputbox "Bonjour, voici un exemple de fenêtre d'entrée\n
    Essayez de saisir votre nom ci-dessous : " 16 51 2> $fichtemp
```

Le programme `dialog` écrit sa sortie sur la sortie d'erreur standard par défaut. Par conséquent, la chaîne d'entrée que vous saisissez est envoyée vers la sortie d'erreur standard que nous redirigeons vers notre fichier temporaire. Vous pourrez reprendre le texte saisi dans le fichier temporaire pour le traiter ultérieurement.

3. Construire un menu

Essayez le programme suivant à la fois dans une console :

```
#!/bin/sh
DIALOG=${DIALOG=dialog}
fichtemp=`tempfile 2>/dev/null` || fichtemp=/tmp/test$$
trap "rm -f $fichtemp" 0 1 2 5 15
$DIALOG --clear --title "Mon jeu prefere" \
    --menu "Bonjour, choisissez votre jeu prefere :" 60 50 10 \
        "Gradius" "Gradius III - SNES" \
        "Probotector" "Probotector - SNES" \
        "Castlevania" "Castlevania - NES" \
        "Alex" "Alex Kid - Master System" \
        "Zelda" "Zelda, A link to the past - SNES" \
        "Slug" "Metal Slug - Neo Geo" 2> $fichtemp
valret=$?
choix=`cat $fichtemp`
case $valret in
    0)      echo "'$choix' est votre jeu préféré";;
    1)      echo "Appuyé sur Annuler";;
    255)    echo "Appuyé sur Echap.";;
esac
```

La logique est exactement la même que pour la fenêtre d'entrée. Nous redirigeons le choix que vous avez fait vers un fichier temporaire, puis nous traitons la valeur de retour de `dialog` et le contenu du fichier temporaire.

4. Liste radio et liste à choix multiples

Les listes radio et les listes à choix multiples peuvent être programmées exactement comme des menus. Un exemple de liste radio simple est donné ci-dessous.

```

#!/bin/sh
DIALOG=${DIALOG=dialog}
fichtemp=`tempfile 2>/dev/null` || fichtemp=/tmp/test$$
trap "rm -f $fichtemp" 0 1 2 5 15

$DIALOG --backtitle "Choisissez votre jeu prefere" \
  --title "Mon jeu prefere" --clear \
  --radiolist "Bonjour, vous pouvez choisir votre jeu prefere ici  "
20 61 5 \
  "Gradius" "Gradius III - SNES" off \
  "Probotector" "Probotector - SNES" off\
  "Castlevania" "Castlevania - NES" off\
  "Alex" "Alex Kid - Master System" ON\
  "Zelda" "Zelda, A link to the past - SNES" off\
  "Slug" "Metal Slug - Neo Geo" off 2> $fichtemp
valret=$?
choix=`cat $fichtemp`
case $valret in
  0)
    echo "'$choix' est votre jeu préféré";;
  1)
    echo "Appuyé sur Annuler.";;
  255)
    echo "Appuyé sur Echap.";;
esac

```

Pour créer des listes à choix multiples, changez simplement l'option `--radiolist` en `--checklist`.

5. Construire une barre d'avancement

Une barre d'avancement basée sur dialog peut servir à indiquer la progression du chargement de votre programme. La construction d'une barre d'avancement n'est pas évidente. Jetez un coup d'œil à l'exemple suivant :

```
#!/bin/sh
DIALOG=${DIALOG=dialog}
compteur=10
(
while test $compteur != 110
do
echo $compteur
echo "XXX"
echo "Le nouveau message ($compteur pourcent)"
echo "XXX"
compteur=`expr $compteur + 10`
sleep 1
done
) |
$DIALOG --title "Ma barre d'avancement" --gauge "Bonjour, ceci est une
barre d'avancement" 20 70 0
```

Ici, le programme dialog prend son entrée dans le code affiché entre parenthèses. Ce code donne le nombre à utiliser pour la barre d'avancement et le message à afficher. Le message à afficher dans la fenêtre de progression doit être entouré par **echo "XXX"**.

6. Sélection de fichier

Voici le code typique d'une fenêtre dialog de sélection de fichiers :

```
#!/bin/sh
DIALOG=${DIALOG=dialog}

fichier=`$DIALOG --stdout --title "Choisissez un fichier" --fselect $HOME/
14 48`

case $? in
0)
echo "\"$fichier\" choisi";;
1)
echo "Appuyé sur Annuler.";;
255)
echo "Fenêtre fermée.";;
esac
```

Notez que le programme ci-dessus utilise une technique différente pour reprendre le fichier sélectionné. Comme mentionné précédemment, toutes les sorties de dialog sont envoyées par défaut sur la sortie d'erreur standard par défaut. Toutefois, l'option `--stdout` permet d'envoyer les informations de sortie à la sortie standard qui à son tour peut être affectée à une variable. Cette astuce peut également servir dans d'autres fenêtres dialog comme les menus, les fenêtres *oui/non*, etc.

7. Calendrier et horloges

Calendrier

Une fenêtre calendrier affiche le jour, le mois et l'année dans des fenêtres ajustables séparément. Si la valeur du jour du mois ou de l'année est absente ou négative, les valeurs actuelles correspondantes sont employées. Vous pouvez augmenter ou diminuer n'importe laquelle de ces valeurs à l'aide des touches **Gauche**, **Haut**, **Droite** et **Bas**. Utilisez les touches **h**, **j**, **k** et **l** dans le style vi pour vous déplacer dans la zone des jours du mois. Faites appel à la touche **Tab** ou **Retour** pour vous déplacer d'un champ à l'autre. Si l'année est mise à zéro, la date actuelle est prise comme valeur initiale. En quittant, la date est affichée sous la forme jour/mois/année.

```
#!/bin/sh
DIALOG=${DIALOG=dialog}

datechoisie=`$DIALOG --stdout --title "CALENDRIER" --calendar "Choisissez
une date..." 0 0 7 7 1981`

case $? in
  0)
    echo "Date saisie : $datechoisie.>";
  1)
    echo "Appuyé sur Annuler.>";
  255)
    echo "Fenêtre fermée.>";
esac
```

L'horloge

La fenêtre horloge de dialog vous permet de choisir une heure. Essayez le programme suivant et observez comment il fonctionne :

```
#!/bin/sh

DIALOG=${DIALOG=dialog}
heure=`$DIALOG --stdout --title "HORLOGE" --timebox "Saisissez l'heure..."
0 0 12 34 56`

case $? in
  0)
    echo "Heure saisie : $heure.>";
  1)
    echo "Appuyé sur Annuler.>";
  255)
    echo "Fenêtre fermée.>";
esac
```
