par Katja and Guido Socher

<katja/at/linuxfocusorg
guido/at/linuxfocus.org>

(X)dialog: des shells qui parlent

Résumé:

Xdialog et dialog sont deux utilitaires classiques destinés à améliorer vos scripts shell par une interface graphique.
Quelques connaissances en programmation shell sont nécessaires pour comprendre cet article. Pour découvrir les bases de la programmation du shell vous pouvez lire l’article 216: "Programmation du Shell".

L´auteur:

Katja est l’éditrice Allemande de LinuxFocus. Elle aime Tux, le cinéma & la photographie et la mer. Sa page personnelle se trouve ici.

Guido est un fan de Linux de longue date et il apprécie ce système parce qu’il est conçu par des personnes honnêtes et ouvertes. C’est l’une des raisons pour lesquelles on le nomme "source ouverte". Sa page personnelle est à linuxfocus.org/~guido

Traduit en Français par:
Georges Tarbouriech
<gt/at/linuxfocusorg>

Introduction

Le shell d’UNIX est un environnement très productif par lui-même et fonctionne parfaitement sans

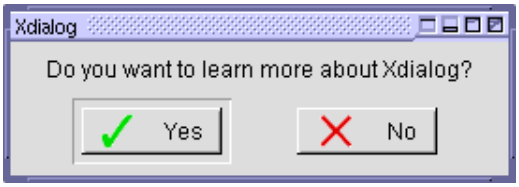
interface graphique.
Dans certains cas, toutefois, il est bien pratique de proposer un dialogue graphique avec l'utilisateur. Un bon exemple concerne le dialogue d'installation d'un programme. Vous pouvez choisir de nombreuses options sur les fonctionnalités à installer et vous pouvez sélectionner le répertoire de destination...

Entamons le (X)dialog...

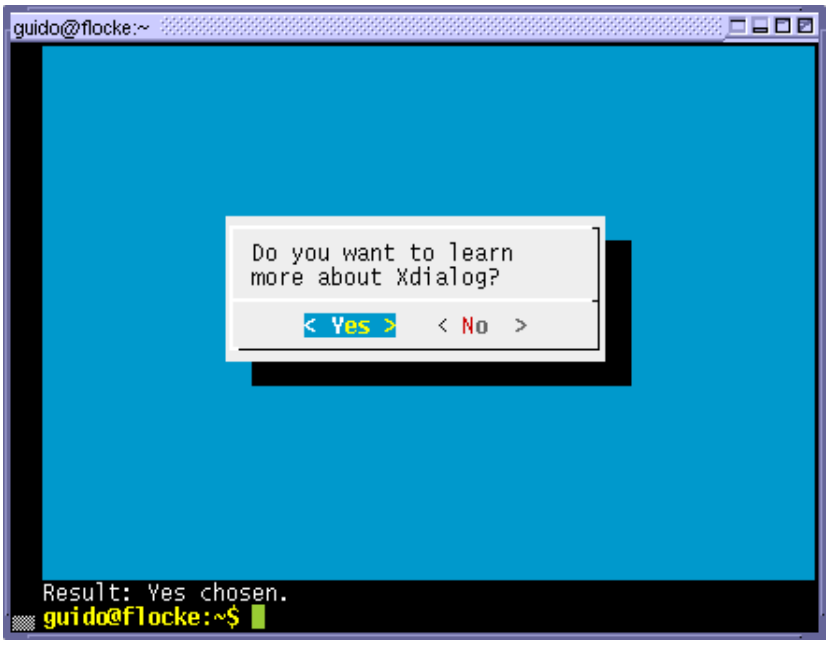
Avec dialog et Xdialog vous pouvez concevoir une application graphique en écrivant simplement un court script shell. Dialog est un programme basé sur du pur terminal et Xdialog est un programme X11.
Voici un exemple :
Tapez (ou copiez/collez) les lignes suivantes dans un shell (xterm, konsole,....):
NDT : nous conservons la langue d'origine des exemples pour que les captures d'écran correspondent. Vous pouvez bien sûr les "franciser"

```
bash
Xdialog --yesno "Do you want to learn more about Xdialog ?" 0 0;\
case $? in
0)
echo "Result: Yes chosen.";;
1)
echo "Result: No chosen.";;
255)
echo "ESC pressed.";;
esac
```

La boîte qui apparaît ressemble à ça :



Si vous utilisez dialog à la place de Xdialog (supprimez le X dans la deuxième ligne du script ci-dessus), vous obtenez une application curses qui fonctionne dans le terminal et n'ouvre donc pas une nouvelle fenêtre. Dans certains cas, il est préférable d'exécuter un script shell dans la fenêtre du terminal. Particulièrement si vous souhaitez l'exécuter à distance sur plusieurs hôtes différents avec lesquels il est possible d'utiliser un routage IP direct. Dans ce cas dialog fonctionnera mais ne pourra pas lancer une application X11 comme Xdialog peut le faire.



L’application dialog/Xdialog ci-dessus était plutôt inutile mais elle montre à quel point il est facile de programmer un dialogue graphique. Il existe des boîtes de dialogue beaucoup plus intéressantes : des calendriers, des menus, des gestionnaires de fichiers, des jauges, des saisies de texte, des boîtes de message, des fenêtres d’entrée de mot de passe... Tapez

dialog --help
or
Xdialog --help

pour obtenir la liste des boîtes de dialogue disponibles. Xdialog en offre plus que dialog.

Comment ça marche

Les boîtes de dialogue se configurent par la ligne de commande.

dialog --yesno "text string" <height> <width>

Après avoir tapé dialog ou Xdialog vous devez donner le nom de la boîte choisie suivi de ses propres paramètres.

La boîte yesno accepte trois paramètres. La <hauteur> et la <largeur> peuvent être définis à zéro auquel cas la géométrie de la boîte sera automatiquement ajustée à la taille du texte. Le résultat est renvoyé au script sous forme de statut de sortie dans la variable "$?". Si d’autres résultats doivent être renvoyés, par exemple, les noms des options sélectionnées, ils le sont vers l’erreur standard. L’erreur standard est habituellement affichée à l’écran mais peut être redirigée par "2>".

Une solution simple mais efficace.

De véritables applications

Voici maintenant une application dans laquelle Xdialog/dialog offrent un réel avantage sur les scripts shell conventionnels : un menu dans lequel vous pouvez choisir entre différents Fournisseurs d'Accès à Internet et lancer la connexion. Le programme réclame les scripts ppp-on/ppp-off de l'article 192 de Mars 2001 Utiliser différents FAI pour votre accès à Internet. Le script se nomme pppdialout et affiche un menu différent selon que vous êtes connectés ou non.

```
#!/bin/sh
#
#DIALOG=Xdialog
DIALOG=dialog
#
# name of your default isp:
defaultisp=maxnet
#
error()
{
    echo "$1"
    exit 2
}
help()
{
  cat <<HELP
pppdialout -- select an ISP and dial out.
All available ISPs must have a config file in /etc/ppp/peers

pppdialout executes the ppp-on/ppp-off scripts as described
in http://linuxfocus.org/English/March2001/article192.shtml

pppdialout, copyright gpl, http://linuxfocus.org/English/November2002
HELP
  exit 0
}

# parse command line:
while [ -n "$1" ]; do
case $1 in
    -h) help;shift 1;; # function help is called
    --) shift;break;; # end of options
    -*) echo "error: no such option $1. -h for help";exit 1;;
    *)  break;;
esac
done

tempfile=/tmp/pppdialout.$$
trap "rm -f $tempfile" 1 2 5 15

# check if we have a ppp network interface
if /sbin/ifconfig | grep '^ppp' > /dev/null; then
    # we are already online
    $DIALOG --title "go offline" --yesno "Click YES to \
                     terminate the ppp connection" 0 0
    rval="$?"
    clear
    if [ "$rval" = "0" ]; then
        echo "running /etc/ppp/scripts/ppp-off ..."
        /etc/ppp/scripts/ppp-off
```

```
    fi
else
    # no ppp connection found, go online
    # get the names of all available ISP by listing /etc/ppp/peers
    for f in ‘ls /etc/ppp/peers‘; do
        if [ -f "/etc/ppp/peers/$f" ]; then
            isplist="$isplist $f =="
        fi
    done
    [ -z "$isplist" ]&&error "No isp def found in /etc/ppp/peers"
    #
    $DIALOG --default-item "$defaultisp" --title "pppdialout"  \
         --menu "Please select one of\
    the following ISPs for dialout" 0 0 0 $isplist 2> $tempfile
    rval="$?" # return status, isp name will be in $tempfile
    clear
    if [ "$rval" = "0" ]; then
        isp=‘cat $tempfile‘
        echo "running /etc/ppp/scripts/ppp-on $isp..."
        /etc/ppp/scripts/ppp-on "$isp"
    else
        echo "Cancel..."
    fi
    rm -f $tempfile
fi
# end of pppdialout
```

Comment fonctionne le script :
Au début nous définissons quelques fonctions, erreurs et aide, et ensuite nous vérifions les arguments de la ligne de commande et le nom d’un fichier temporaire est attribué (/tmp/pppdialout.$$). $$ est le nom du processus courant et il possède un numéro unique pour chaque ordinateur. "Trap" est exécuté si le programme se termine anormalement (si l’utilisateur presse Ctrl-C, par exemple) et supprime le fichier temporaire. Ensuite, nous vérifions si nous sommes déjà en ligne (commande: /sbin/ifconfig | grep ’^ppp’). Si nous sommes en ligne, nous ouvrons une boîte yesno (ouinon), celle que nous avons déjà vu plus haut, et demandons à l’utilisateur s’il souhaite se déconnecter. Si nous ne sommes pas en ligne une boîte menu est ouverte. Nous y obtenons tous les FAI disponibles en listant les fichiers contenus dans /etc/ppp/peers (ls /etc/ppp/peers). La syntaxe de la boîte menu :

dialog --menu "text" <height> <width> <menu height> <tag1> <description> ...

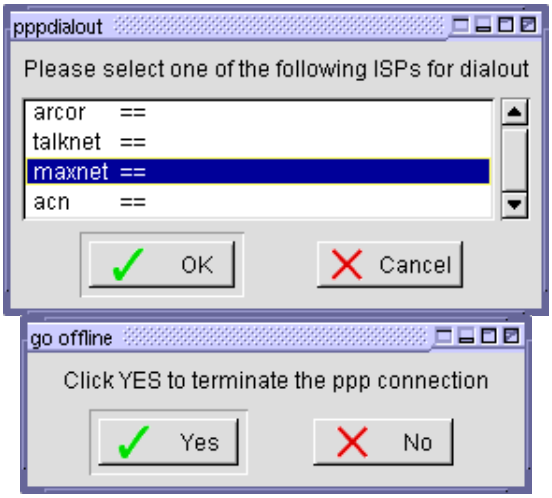
La <hauteur>, la <largeur> et la <hauteur du menu> sont de nouveau définis à zéro (taille automatique, voir plus haut) et le programme attend des couples de chaînes de caractères (<tag1> <description>). Nous n’avons pas vraiment de description, nous définissons donc quelque chose sans signification (== dans ce cas). Les données dans la variable iplist ressembleront à ce qui suit :

isp1 == isp2 == isp3 ==

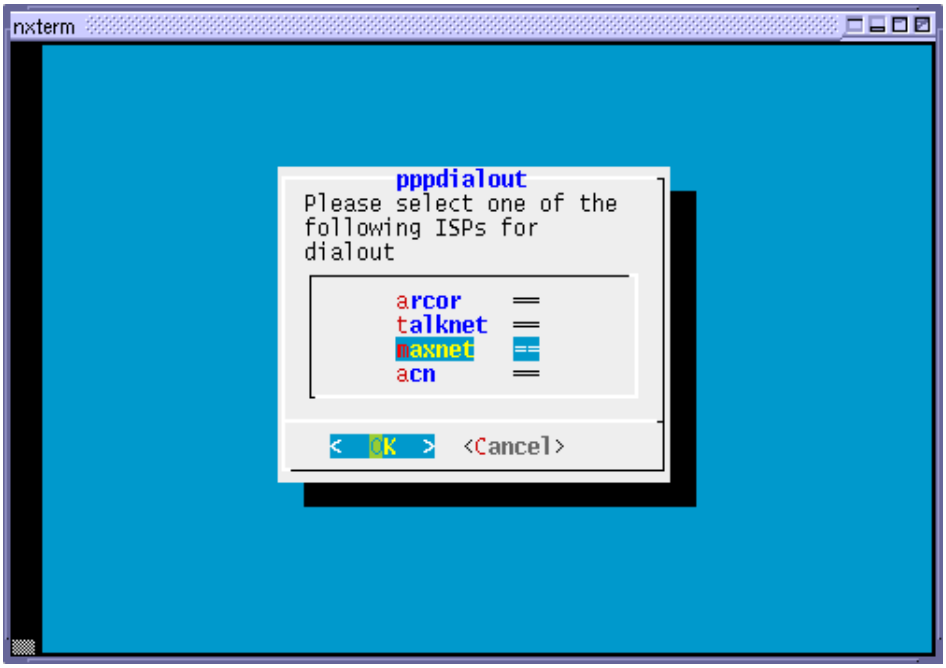
Le résultat du choix de l’utilisateur est affiché par (X)dialog sur l’erreur standard. La commande shell "2> $tmpfile" l’écrit dans notre fichier temporaire (tempfile). La boîte menu offre également la possibilité de presser le bouton cancel (Annuler). Nous devons donc vérifier $? (le statut de sortie) pour savoir quel bouton a été pressé.

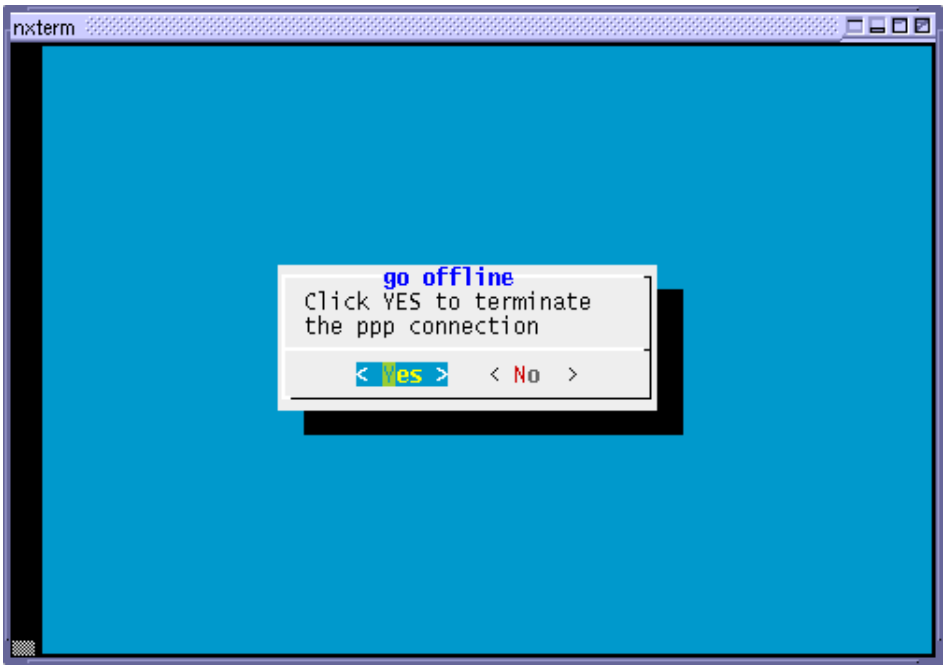
Bon, assez de théorie. Voilà à quoi ça ressemble.

... en tant qu'interface GTK avec Xdialog:



... avec le dialogue curses dans le terminal:





D’autres applications

Nous avons une autre application à vous proposer. Elle se nomme mktgz et utilise la boîte "checklist" de Xdialog. Dialog (basé sur le terminal) ne possède pas de "checklist", par conséquent cela ne fonctionnera qu’avec Xdialog. Pour construire vos paquetages tar.gz vous pouvez utiliser mktgz.

mktgz votrepaquetage .

Cette commande affiche tous les fichiers du répertoire courant (".") et vous pouvez choisir lesquels inclure dans votre paquetage tar.gz. Vous pouvez le télécharger ici (mktgz.txt) Nous n’analyserons pas le code ligne à ligne puisque vous en savez déjà assez pour comprendre le script.

Conclusion

Xdialog et dialog offrent de nombreuses boîtes de dialogue différentes. Elles ne sont pas toujours appropriées à tous les types de script shell. Le shell proprement dit est déjà un environnement très puissant. Compléter un chemin par la touche de tabulation est beaucoup plus rapide que de rechercher les différents répertoires dans une application graphique en cliquant sur chacun. La possibilité d’imbriquer et de combiner les commandes en fait un outil d’une puissance étonnante. Par exemple :

```
grep -i "somestring" file.txt | sort | uniq | wc -l
```

(pour ceux qui manquent d’expérience avec le shell UNIX : cette commande compte les lignes du fichier file.txt contenant la chaîne de caractères "somestring")
De telles constructions imbriquées sont rendues possibles parce que toutes les commandes sont contrôlées par des arguments. En d’autres termes : elles ne s’arrêtent pas pour demander à l’utilisateur

s’il veut continuer.
Toutefois les dialogues graphiques peuvent être très utiles dans certaines applications. Xdialog et dialog sont très faciles à utiliser sans, bien évidemment, bénéficier de la puissance d’une véritable application graphique. Elles comblent le vide entre le script shell en ASCII pur et l’application graphique.

Xdialog et dialog contiennent un répertoire "samples" dans lequel vous trouverez d’autres exemples (Redhat 7.3 les stocke dans /usr/share/doc/Xdialog-2.0.5/samples). Soyez prudents parce que certains sont vraiment fonctionnels et ne doivent plus être assimilés à des "demos".

Où se procurer Xdialog et dialog?

Les CD de votre distribution Linux sont le premier endroit où chercher dialog et Xdialog. Ils sont peut-être déjà installés sur votre ordinateur (demandez-le lui : rpm -qil Xdialog, dpkg -L Xdialog). Le site de Xdialog se trouve à :
http://www.chez.com/godefroy/
et dialog à
http://hightek.org/dialog/
Vous pouvez y télécharger dialog/Xdialog.

Références

- Xdialog: http://www.chez.com/godefroy/
  dialog: http://hightek.org/dialog/
- Documentation Xdialog : http://www.chez.com/godefroy/doc/index.html
- article 192: Utiliser differents FAI pour votre accès à Internet
- script pppdialout
- script mktgz
- article 216: Programmation du Shell

Site Web maintenu par l´équipe d´édition LinuxFocus	Translation information:
© Katja and Guido Socher	en --> -- : Katja and Guido Socher <katja/at/linuxfocusorg guido/at/linuxfocus.org>
"some rights reserved" see linuxfocus.org/license/	en --> fr: Georges Tarbouriech <gt/at/linuxfocusorg>
http://www.LinuxFocus.org	

2005-01-14, generated by lfparser_pdf version 2.51