



by Katja and Guido Socher
<katja/at/linuxfocusorg
guido/at/linuxfocus.org>

About the authors:

Katja ist die deutsche Editorin von LinuxFocus. Sie mag Tux, Film & Fotografie und das Meer. Ihre Homepage befindet sich [hier](#).

Guido ist ein langjähriger Linuxfan und er mag Linux, weil es von ehrlichen und offenen Leuten entworfen wurde. Dies ist einer der Gründe, warum wir es Open Source nennen. Seine Homepage befindet sich unter linuxfocus.org/~guido.

(X)dialog: Sprechende Shells



Abstract:

Xdialog und dialog sind zwei klassische Utilities, um deine Shellskripte mit einer grafischen Benutzerschnittstelle zu bereichern. Du brauchst ein wenig Erfahrung in Shellprogrammierung, um den Artikel verstehen zu können. Um die Grundlagen der Shellprogrammierung zu lernen, kannst du unseren Artikel über [Shell Programmierung](#) lesen.

Einführung

Die UNIX Shell ist eine sehr produktive Umgebung für sich genommen und arbeitet sehr gut ohne grafische Benutzerschnittstelle.

In ein paar Fällen macht es allerdings Sinn, einen grafischen Dialog mit der Benutzerin zu haben. Ein Beispiel wäre z.B. der Installationsdialog eines Programms. Man hat eine Reihe von Optionen für die zu installierenden Features und man kann das Zielverzeichnis auswählen....

Komm herein zu (X)dialog...

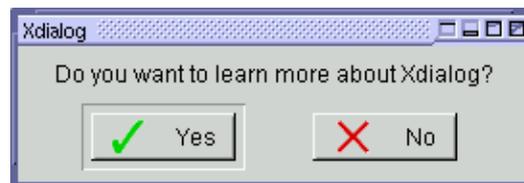
Mit dialog und Xdialog kann man eine grafische Applikation allein durch Schreiben eines kurzen Shellscripts entwerfen. Dialog ist ein rein terminal-basiertes Programm und Xdialog ist ein X11 Programm.

Hier ist ein Beispiel:

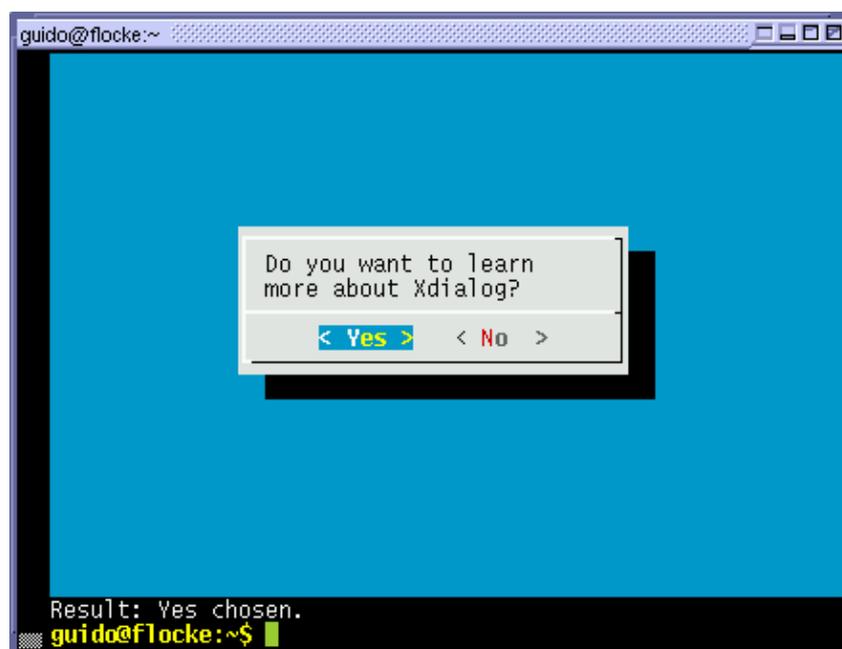
Du kannst die folgenden Zeilen in ein Shellfenster (xterm, konsole,...) tippen (oder kopieren/pasten):

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

Die Box, die erscheint, sieht so aus:



Wenn du dialog anstelle von Xdialog benutzt (entferne das X in der zweiten Zeile im oben gezeigten Skript), dann bekommst du eine auf curses basierende Applikation, die innerhalb von xterm läuft und kein separates Fenster öffnet. In einigen Fällen ist das eher für ein Shellskript angebracht, da es nur innerhalb des Terminalfensters läuft. Besonders, wenn du es remote mit einer Anzahl verschiedener hosts zwischen deinem Computer und dem entfernten host laufen lassen willst und kein direktes IP Routing verfügbar ist. In diesem Fall läuft dialog, aber du bist nicht in der Lage, eine X11 Applikation wie Xdialog zu starten.



Das obige war ein ziemlich nutzloses Beispiel von dialog/Xdialog, aber es zeigt, wie einfach es ist, einen einfachen graphischen Dialog zu programmieren. Es gibt weitere interessante Dialogboxen. Boxen wie calendar (Kalender), menus (Menüs), filemanager, progress bar, text input, message box, password dialog, ...

Laß

dialog --help

oder

Xdialog --help

laufen, um eine Liste der vorhandenen Dialogboxen zu bekommen. Xdialog hat ein paar mehr Boxen als dialog.

Wie es arbeitet

Die Dialogkästen werden auf der Kommandozeile konfiguriert.

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

Nach dem Tippen von dialog oder Xdialog muß man den Namen der gewünschten Box angeben, gefolgt von ihren spezifischen Parametern.

Die yesno Box verlangt 3 Parameter. <height> und <width> können auf 0 gesetzt werden, wodurch die Geometrie der Box sich automatisch nach der Länge des Textes richtet. Das Ergebnis wird als exit status an das Skript in der Variablen "\$?" zurückgegeben. Wenn mehr zurückgegeben werden muß, wie z.B. die Namen von selektierten Optionen, dann wird dies auf Standarderror zurückgegeben. Standarderror wird normalerweise einfach auf den Bildschirm geschrieben, kann aber mit ">2" umgeleitet werden.

Eine sehr einfache, aber wirkungsvolle Lösung.

Echte Applikationen

Jetzt eine real world Applikation, wo Xdialog/dialog wirklich einen Vorteil über konventionelle Shellscriptprogramme bieten: Ein Menü, wo man verschiedene Internet Service Provider auswählen und dann anwählen kann, um eine Internetverbindung aufzubauen. Dieses Skript erfordert ppp-on/ppp-off Skripte des März 2001 Artikels [Benutzen verschiedener ISPs für deine Internetverbindung](#). Das Skript heißt pppdialout und zeigt ein unterschiedliches Menü, abhängig davon, ob man online ist oder nicht.

```
#!/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

```

Wie das Skript arbeitet:

Zu Beginn definieren wir einige Funktionen, error und help, als nächstes überprüfen wir die Kommandozeilenargumente, dann wird ein Name für eine temporäre Datei definiert (/tmp/pppdialout.\$\$). \$\$ ist der Name des laufenden Prozesses und ist eine einzigartige Zahl für jeden Computer. Das trap statement wird ausgeführt, wenn das Programm unnormale beendet wird (z.B. wenn die Benutzerin ctrl-C drückt) und löscht in unserem Fall das tempfile. Danach überprüfen wir, ob wir schon online sind oder nicht (Befehl:

/sbin/ifconfig | grep '^ppp'). Wenn wir schon online sind, dann öffnet sich eine yesno-box, wie du oben bereits eine gesehen hast, und fragt die Benutzerin, ob sie offline gehen möchte. Wenn wir nicht online sind, wird eine Menübox geöffnet. Wir bekommen alle verfügbaren ISPs durch Auflisten der Dateien in /etc/ppp/peers (ls /etc/ppp/peers). Die Syntax der Menübox ist:

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

<height>, <width> und <menu height> werden wieder auf 0 gesetzt (automatische Länge, siehe oben) und dann erwartet das Programm Paare von Zeichenketten (<tag1> <description>). Wir haben keine wirkliche Beschreibung, deshalb setzen wir sie auf etwas Bedeutungsloses(== in diesem Fall). Die Datei in der Variablen isplist sieht so aus:

```
isp1 == isp2 == isp3 ==
```

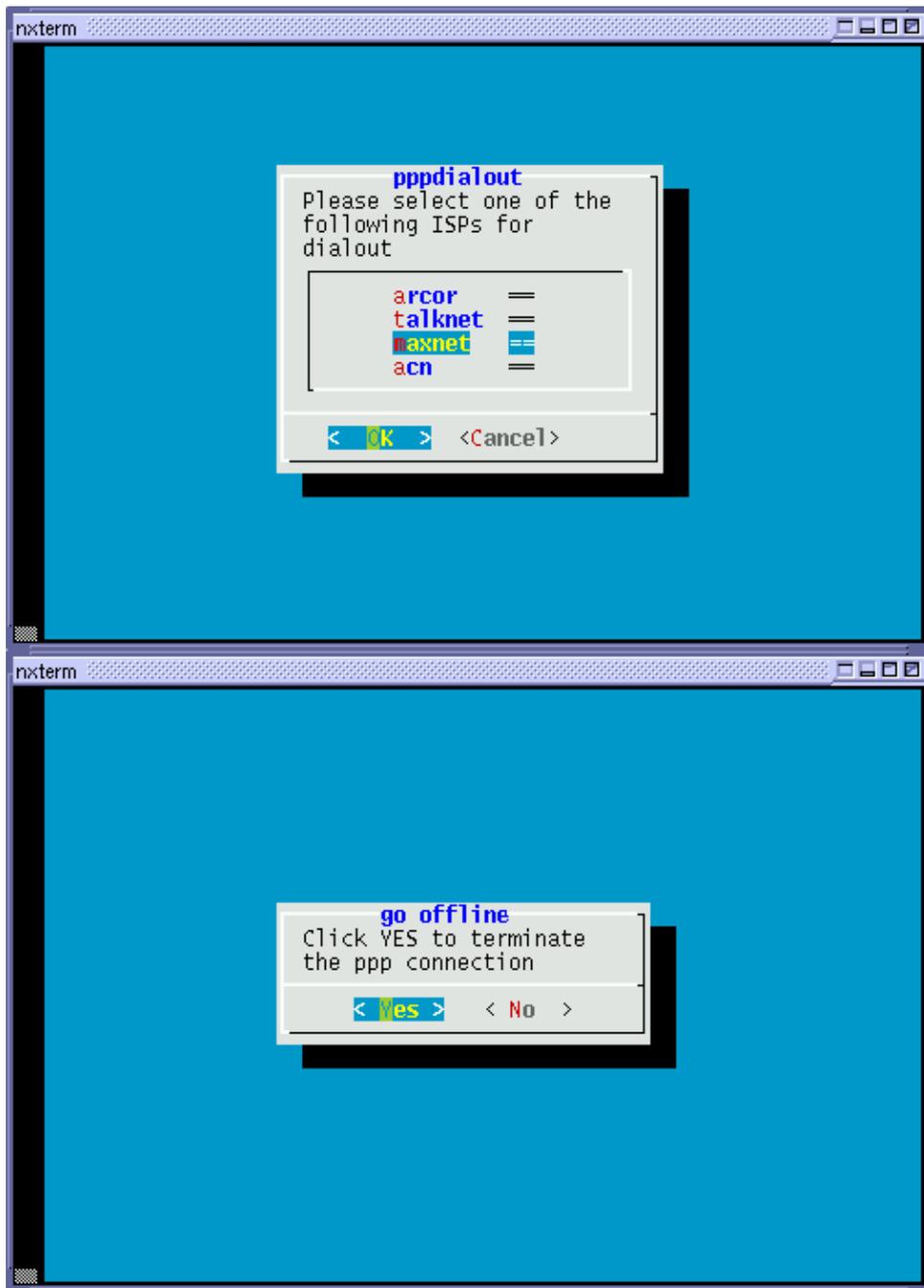
Das Ergebnis der Benutzerauswahl wird von (X)dialog in Standarderror geschrieben. Der Shellbefehl "2> \$tmpfile" schreibt es in unser tmpfile. Die Menübox bietet auch die Möglichkeit, auf Abbrechen zu drücken. Deshalb müssen wir \$? (exit status) überprüfen, um herauszufinden, welcher Knopf gedrückt wird.

Ok, genug Theorie. Hier ist, wie es aussieht

... als nettesGTK GUI mit Xdialog:



... mit dem curses basierten dialog in einem Terminal:



Weitere Applikationen

Wir haben noch eine weitere Anwendung für dich. Sie heißt mktgz und benutzt die checklist Box von Xdialog. Das einfache terminal-basierte dialog besitzt keine Checkliste, deshalb läuft sie nur mit Xdialog. Du kannst mktgz benutzen, um tar.gz Pakete zu bauen.

mktgz yourpackage .

Dies zeigt alle Dateien des aktuellen Arbeitsverzeichnisses (".") und du kannst auswählen, welche in dein yourpackage.tar.gz Paket mit eingeschlossen werden sollen. Du kannst es [hier \(mktgz.txt\)](#) herunterladen. Wir

gehen nicht Zeile für Zeile durch den Code, da du schon genug wissen solltest, um das Skript lesen zu können.

Xdialog und dialog kommen mit einem Verzeichnis namens "samples", wo du weitere Beispiele finden kannst (Redhat 7.3 speichert sie unter /usr/share/doc/Xdialog-2.0.5/samples). Sei aber vorsichtig, da einige von ihnen wirklich etwas tun und nicht bloß reine Demoapplikationen sind.

Schlußbemerkung

Xdialog und dialog bieten eine Reihe verschiedener Dialogboxen. Nicht alle von ihnen sind immer für alle Typen von Shellscripten angebracht. Die Shell selbst ist auch eine sehr "mächtige" Umgebung. Vervollständigen eines Pfades mit der tab-Taste kann sehr viel schneller sein, als das Durchsuchen verschiedener Verzeichnisse in einem GUI und sie dann durch Drücken auszuwählen. Besonders die Möglichkeit des Pipelinens und Verbindens von Befehlen macht die Shell zu einem sehr mächtigen Werkzeug. Etwas wie:

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

(für diejenigen, die nicht so erfahren sind mit Unixshells: Dies zählt alle unterschiedlichen Zeilen in file.txt, die die Zeichenkette "somestring" enthalten).

Solche Pipelinekonstrukte sind möglich, weil alle Befehle von Kommandozeilenargumenten gesteuert werden. In anderen Worten: sie halten nicht an und fragen die Benutzerin, wie sie weitermachen möchte. Es gibt jedoch Applikationen, wo grafische Dialoge sehr nützlich sind. Xdialog und dialog sind sehr einfach zu benutzen, aber natürlich nicht so mächtig wie eine echte grafische Anwendung. Sie schließen die Lücke zwischen einem reinen ASCII Shellscript und einer vollständigen grafischen Anwendung.

Wo bekommt man Xdialog und dialog?

Die CDs deiner Linuxdistribution sind der erste Platz, um nach dialog und Xdialog zu suchen. Es könnte sein, daß sie schon auf deinem Computer installiert sind (frag deinen Computer z.B. rpm -qil Xdialog, dpkg -L Xdialog). Die Homepage von Xdialog ist:

<http://www.chez.com/godefroy/>

und dialog ist unter

<http://hightek.org/dialog/>

Du kannst dialog/Xdialog auch von dort herunterladen.

Referenzen

- Xdialog: <http://www.chez.com/godefroy/>
dialog: <http://hightek.org/dialog/>
- Xdialog Dokumentation: <http://www.chez.com/godefroy/doc/index.html>
- Andere LinuxFocus Artikel:
 - ◆ [Benutzen verschiedener ISPs für deinen Internetzugang](#)
 - ◆ [Shellprogrammierung](#)
- [pppdialout script](#)
- [mktgz script](#)

<p><u>Webpages maintained by the LinuxFocus Editor</u> <u>team</u></p>	<p>Translation information:</p>
--	---------------------------------

© Katja and Guido Socher
"some rights reserved" see linuxfocus.org/license/
<http://www.LinuxFocus.org>

en --> -- : Katja and Guido Socher <katja/at/linuxfocusorg
guido/at/linuxfocus.org>
en --> de: Katja Socher <katja(at)linuxfocus.org>

2005-01-11, generated by lfparsr_pdf version 2.51