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

About the authors:
Katja è la redattrice per la Germania di LinuxFocus. Ama Tux, i film & photography e adora il mare. La sua pagina personale la trovate qui. Guido è un fan di lunga data di Linux; gli piace Linux perchè è stato creato da gente onesta e disponibile.Questo è uno dei motivi per i quali si parla di opensource. La sua homepage si trova su linuxfocus.org/~guido.

(X)dialog: Shell parlanti

Abstract:

Xdialog e dialog sono due classici strumenti che dotano i vostri script per la shell di una interfaccia grafica (GUI).
Per comprendere quest'articolo avrete bisogno di qualche nozione basilare di programmazione. Per iniziare a conoscere i fondamenti della programmazione della shell potreste leggere il nostro articolo Shell Programming.

Introduzione

La shell UNIX costituisce un ambiente molto fertile di per sè stesso e lavora benissimo anche senza un'interfaccia grafica.
Tuttavia, talvolta, per alcuni utenti è di grossa importanza trovarsi di fronte a finestre di dialogo. Un esempio potrebbe essere l'interfaccia di installazione di un programma. Avere parecchie opzioni di installazione e poter scegliere la directory di installazione...

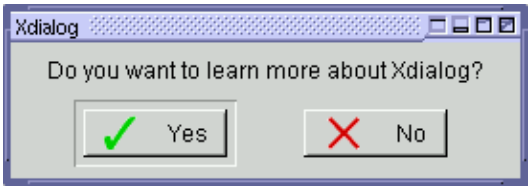
Approccio ad (X)dialog...

Con dialog e Xdialog potrete creare delle applicazioni grafiche digitando giusto qualche riga di codice. Dialog è un programma basato sul terminale puro e semplice, mentre Xdialog è un programma per X11.
Ecco un esempio:
Potete ridigitare (o fare un copia/incolla) le seguenti linee in una shell (xterm, konsole,....):

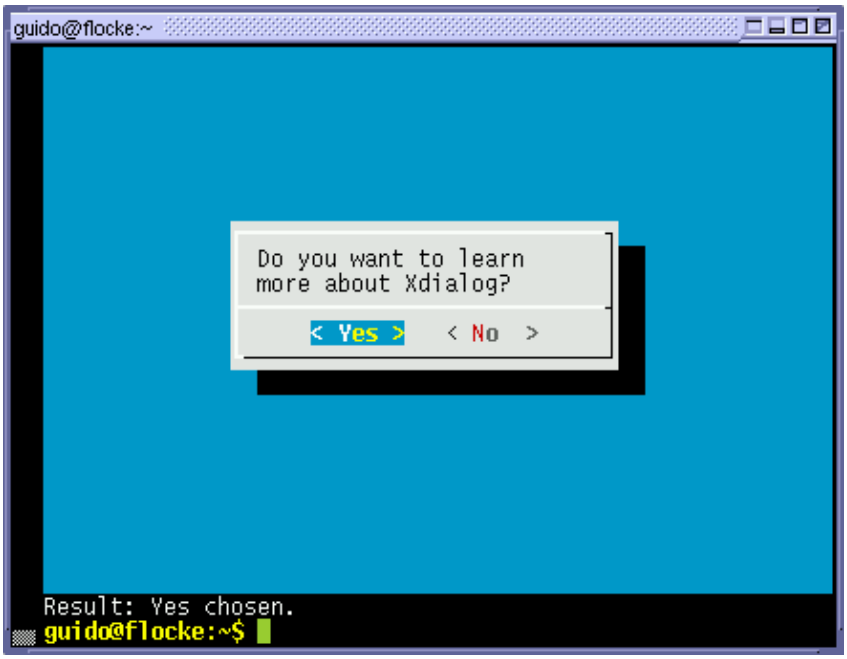
```
bash
Xdialog --yesno "Do you want to learn more about Xdialog?" 0 0;\
case $? in
```

```
0)
echo "Result: Yes chosen.";;
1)
echo "Result: No chosen.";;
255)
echo "ESC pressed.";;
esac
```

Il riquadro che vi apparirà avrà più o meno quest'aspetto:



Se usate dialog invece che Xdialog (eliminate la X nella seconda linea dello script sopraesposto) otterrete un'applicazione basata su curses che funziona all'interno di xterm e che non apre una nuova finestra. In qualche caso questo è più appropriato per uno script di shell creato appositamente per la finestra di terminale. E' particolarmente indicato nel caso in cui vogliate avviarlo da remoto avendo un certo numero di host tra il vostro computer e l'host remoto: non sarebbe infatti possibile un rooting IP diretto. In questo caso dialog funzionerebbe ma non potrebbe far partire un'applicazione X11 come invece potrebbe fare Xdialog.



Quello qui sopra è un esempio carino ma senza grossa utilità di dialog/Xdialog, tuttavia dimostra quanto sia semplice programmare una semplice finestra di dialogo. Esistono finetre di dialogo molto più complesse e interessanti. Son possibili effetti tridimensionali, calendari, menu, filemanager, barre di avanzamento, riquadri di testo, di avvisi e di password... Potete digitare

dialog −−help
oppure
Xdialog −−help

per avere una lista delle finestre disponibili. Xdialog ha qualche riquadro in più di dialog.

Come funziona

I riquadri di dialog vengono configurati da linea di comando.

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

Una volta digitato "dialog" o "Xdialog" dovrete scrivere il nome del box che avete scelto, più i suoi parametri specifici.

Il box "yesno" (si – no) ammette tre variabili. Quella <heigh> (altezza) e <width> (larghezza) possono essere settate a zero se si vuole che la dimensione del box sia scelta automaticamente a seconda dello spazio che sarà occupato dal testo al suo interno. L'esito viene restituito come "exit status" nella variabile "$?" dello script. Se vengono restituiti anche altri esiti come ad esempio i nomi delle opzioni selezionate, allora verranno mostrati come standard error. Gli Standard error normalmente son stampati a schermo ma possono essere rediretti con ">2".

Un esempio semplice ma efficace.

Un esempio pratico

Creiamo adesso una applicazione in cui Xdialog/dialog forniscono un qualche vantaggio su un normale script per shell: un menu in cui si possa selezionare tra differenti ISP (Internet Service Providers) quello con cui si vuole iniziare la connessione ad Internet. Questo script richiede, per poter funzionare, lo script "ppp−on/ppp−off" descritto nell'articolo del marzo 2001 chiamato Usare diversi ISP per il vostro accesso ad Internet. Lo script si chiama pppdialout e mostra diversi menu a seconda che si sia o meno connessi.

```
#!/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
```

Come funziona lo script:
All'inizio definisce qualche funzione, errori ed help e poi controlla gli argomenti da linea di comando e viene definito un nome per il file temporaneo (/tmp/pppdialout.$$). $$ è il nome del processo corrente e consiste in un numero unico per ogni computer. L'espressione "trap" è eseguita se il programma viene chiuso in modo anomalo (ad esepio se l'utente clicca ctrl−c) a cancella il file temporaneo. Dopo controlla se siamo ancora in linea o meno (comando: /sbin/ifconfig | grep '^ppp'). Se siamo ancora in linea allora apre una yesno−box, come quella che abbiamo già visto prima, e chiede all'utente se vuol continuare offline. Se non siamo più online si apre la finestra−menu. Qui abbiamo la lista di tutti gli ISP disponibili contenuti nel file /etc/ppp/peers (ls /etc/ppp/peers). La sintassi del menu box è:

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

I parametri<height>, <width> e <menu height> vengono nuovamente settati a zero (in modo da avere un dimensionamento automatico) e quindi il programma aspetta che vengano immessi due parametri (<tag1> <description>). Non forniremo una descrizione dettagliata, perciò per qualcuno potrebbero essere discorsi di difficile comprensione. I dati nella variabile isplist dovrebbero risultare qualcosa del genere:

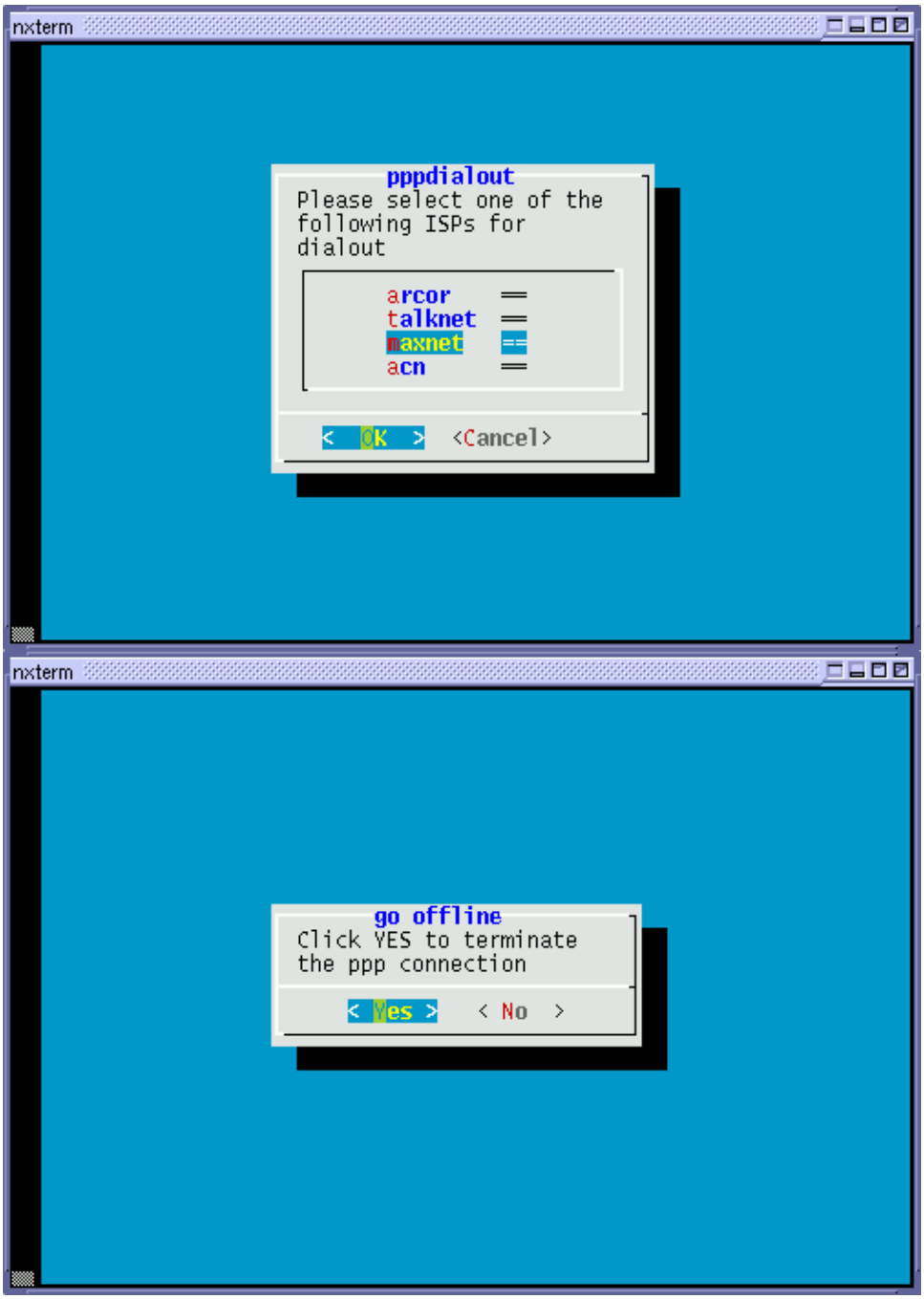
isp1 == isp2 == isp3 ==

Il risultato della scelta fatta dall'utente viene stampata da (X)dialog sullo standard error. Il comando di shell "2> $tmpfile" lo scrive nel tmpfile. Il menu box offre anche la possibilità di cancellare. Quindi dobbiamo controllare $? (exit status) per capire quale pulsante è stato premuto.

Ok, basta teoria. Qui vediamo come dovrebbe sembrare il tutto.

... con una bella GTK GUI con Xdialog:

... con la finestra di dialogo basata sulle curses nel terminale:



Più applicazioni

Abbiamo un'altro programmino pronto per voi. Si chiama mktgz ed utilizza la checklist box di Xdialog. La finestra di dialogo per terminale non ha una checklist, perciò mktgz funziona solo con Xdialog. Potete usare mktgz per creare pacchetti tar.gz.

mktgz yourpackage .

Questo comando mostra tutti i file nella directory corrente (".") e potete selezionarna altre da includere nel vostro pacchetto vostropacchetto.tar.gz. Potete scaricare qui (mktgz.txt) mktgz. Non ci addentreremo nelle

linee di codice perchè quello che avete letto qui dovrebbe bastarvi per capire da soli il contenuto di questo script.

Xdialog e dialog hanno una directory chiamata "samples" dove troverete molti più esempi (Redhat 7.3 li raccoglie sotto la cartella /usr/share/doc/Xdialog−2.0.5/samples). Considerate, tuttavia, che alcune samples non son delle mere applicazioni demo ma eseguono dei compiti ben precisi.

Conclusioni

Xdialog e dialog offrono un numero consistente di dialog boxes. Non tutte queste finestre di dialogo son sempre applicabili a qualsiasi script di shell. La shell stessa costituisce un "ambiente" molto versatile e potente. Completare un percorso ad un file può essere molto più conveniente e veloce in shell con il tasto TAB, piuttosto che cercare quello stesso file cliccando nelle cartelle contenute in una finestra grafica. In particolare la possibilità offerta da pipeline e dalla combinazione dei comandi rende la shell uno strumento insuperabile. Qualcosa tipo:

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

(per chi non avesse abbastanza esperienza con le shell UNIX: Questo comando conta quante linee del file file.txt contengono il termine "somestring")
Tale articolazione di pipeline è possibile poichè tutti i comandi son controllati dagli argomenti di linea di comando. In altre parole: chiedono in continuazione all'utente se voglia continuare.
Comunque ci son applicazioni dove le finestre di dialogo sono realmente utili. Xdialog e dialog son molto facili da usare ma certamente non quanto una applicazione grafica. Essi colmano il dislivello tra uno script per shell ASCII e una applicazione grafica.

Dove trovare Xdialog e dialog?

I CD della vostra distribuzione Linux sono il primo posto dove dovreste cercare dialog e Xdialog. Potrebbe anche darsi che siano già installati sul vostro computer (chiedete al vostro computer: ad esempio rpm −qil Xdialog, dpkg −L Xdialog). La homepage di Xdialog è:
http://www.chez.com/godefroy/
e dialog si trova su
http://hightek.org/dialog/
Nel caso in cui non li abbiate trovati installati o nei vostri CD potrete anche cercarli nei loro siti ufficiali.

Riferimenti

- Xdialog: http://www.chez.com/godefroy/
  dialog: http://hightek.org/dialog/
- Xdialog documentazione: http://www.chez.com/godefroy/doc/index.html
- Altri articoli di LinuxFocus:
  - Using different ISPs for your Internet access
  - Shell Programming
- pppdialout script
- mktgz script

en −−> −− : Katja and Guido Socher <katja/at/linuxfocusorg guido/at/linuxfocus.org>
en −−> it: Kikko <kikko/at/linuxfocus.org>

2005−01−10, generated by lfparser_pdf version 2.51