

Разработка простых интерфейсов с помощью dialog/Xdialog (interface dialog shell gui x window)

<< Предыдущая	ИНДЕКС	Поиск в статьях	src	Установить закладку	Перейти на закладку	Следующая >>
------------------	--------	--------------------	-----	------------------------	------------------------	-----------------

Ключевые слова: interface, dialog, shell, gui, x, window, (найти похожие документы)

From: Иван Песин, Александр Куприн
Newsgroups: <http://gazette.linux.ru.net>
Date: Mon, 17 May 2004 18:21:07 +0000 (UTC)
Subject: Разработка простых интерфейсов с помощью dialog/Xdialog

Оригинал: <http://gazette.linux.ru.net/lgl01/sunil.html>

Разработка простых интерфейсов с помощью dialog/Xdialog
Автор: Sunil Thomas Thonikuzhiyil
(<http://linuxgazette.net/authors/sunil.html>)

Перевод: Иван Песин
(http://gazette.linux.ru.net/team/i_pesin.html),
Александр Куприн
(http://gazette.linux.ru.net/team/a_kuprin.html)

(Примеры, рассмотренные здесь, вы сможете найти в архиве по адресу
http://gazette.linux.ru.net/lgl01/misc/sunil/examples_sh.tar.gz. --
прим. А.К.)

1) Введение

Статья рассматривает применение программ dialog и Xdialog для реализации простых интерфейсов к скриптам. Это предполагает, что вы знакомы с написанием скриптов командного интерпретатора. Последняя версия статьи доступна по адресу
<http://gnubox.dyndns.org:8080/~sunil/dialog.php>.

dialog это утилита для построения консольных интерфейсов. Xdialog аналогичная программа для X. Обе программы более-менее совместимы и легко интегрируются в скрипты. Dialog входит в состав большинства дистрибутивов GNU/Linux. Если вы хотите собрать её из исходников, то

архив можно найти на <http://hightek.org/dialog/>. Программа Xdialog доступна на сайте <http://xdialog.dyns.net/>

Эти программы являются свободными и работают на большом количестве платформ *nix. Большинство приведённых примеров в данном руководстве являются адаптацией примеров, поставляемых с исходными кодами программ.

2) Основы

Вот первый скрипт, работу которого я проверил. Он просто выводит диалог с кнопками "Да" и "Нет".

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

$DIALOG --title " Мой первый диалог" --clear \
    --yesno "Привет! Перед вами пример программы,\ниспользующей
(X)dialog" 10 40

case $? in
    0)
        echo "Выбрано 'Да'.";;
    1)
        echo "Выбрано 'Нет'.";;
    255)
        echo "Нажата клавиша ESC.";;
esac
```

Скопируйте приведённые строки в файл, например, yesno.sh и установите атрибут выполнения.

```
$chmod u+x yesno.sh
```

Теперь попробуйте запустить его (см.рисунок [8]1).

```
$/yesno.sh
```

Изменим строку

```
DIALOG=${DIALOG=dialog}
```

на

```
DIALOG=${DIALOG=Xdialog}
```

и запустим скрипт из xterm.

Давайте детально разберём приведённую программу. Первая строка является комментарием, который также указывает, что для выполнения требуется командный интерпретатор bash. (Последовательность #! в мире

Unix называется sha-bang. Она указывает системе какой именно интерпретатор следует использовать для исполнения сценария -- <http://gazette.linux.ru.net/rus/articles/abs-guide/c112.html>. -- прим. А.К.)

```
DIALOG=${DIALOG=dialog}
```

Эта строка присваивает переменной DIALOG значение 'dialog'. Сам же диалог формируется следующей строкой:

```
$DIALOG --title " Мой первый диалог" --clear \
    --yesno "Привет! Перед вами пример
программы,\ниспользующей (X)dialog" 10 40
```

Применяемые опции:

--title задаёт заголовок диалога

```
--clear очищает экран перед отображением диалога
--yesno задаёт тип диалога и текст для отображения.
```

Выводимый текст нужно брать в двойные кавычки. Текст переносится в зависимости от ширины диалогового окна. Можно использовать символ \n

для указания принудительного перевода строки. Последние два числа задают ширину и высоту диалога. (Размеры окна задаются в символах. Это

верно как для dialog, так и Xdialog. При этом для вывода текстовой информации Xdialog использует моноширинный шрифт. -- прим. А.К.)

Между

кнопками можно переключаться при помощи клавиши табуляции.

Теперь программа ждёт пользовательского выбора. В зависимости от того,

нажмёте кнопку "Да" или "Нет", или нажмёте клавишу Escape, переменная

командного интерпретатора \$? будет содержать код завершения программы,

который можно так или иначе обработать.

2) Ввод данных

Следующая программа ожидает ввода строку и затем отображает её на экране.

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

$DIALOG --title "Ввод данных" --clear \
        --inputbox "Привет! Перед вами пример ввода данных\nВведите своё имя:" 16 51 2> $tempfile
```

```
retval=$?
```

```
case $retval in
```

```
0)
    echo "Вы ввели `cat $tempfile`"
    ;;
```

```
1)
    echo "Отказ от ввода.";;
```

```
255)
    if test -s $tempfile ; then
        cat $tempfile
    else
        echo "Нажата клавиша ESC."
    fi
    ;;
```

```
esac
```

Запустите программу в консоли и под X (после замены dialog на Xdialog)

Эта программа немного сложнее, чем предыдущая. Следующие строки определяют временный файл и его удаление при завершении программы:

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

(В оригинальном скрипте для формирования имени временного файла использовалась утилита `tempfile`, но такую обнаружить не удалось, поэтому пришлось её заменить на `mktemp`. Хотя скрипт работать будет

в

любом случае, т.к. в случае ошибки конструкция `||` позволяет сформировать имя файла вида `/tmp/test$$`, где `$$` -- значение генератора

случайных чисел. Хотя, на мой взгляд, корректней было бы использовать переменную `$RANDOM`. -- прим. А.К.)

В первой строке делается попытка создать временный файл с помощью утилиты `mktemp`. Если это не получается, он создается вручную, в каталоге `/tmp`. Вторая строка определяет обработчик сигналов. При завершении скрипта (вне зависимости, корректном или не корректном) обработчик удаляет временный файл. Числа -- это номера обрабатываемых сигналов.

После этого вызывается программа `dialog`:

```
$DIALOG --title "Ввод данных" --clear \
--inputbox "Привет! Перед вами пример ввода данных\nВведите своё имя:" 16 51 2> $tempfile
```

Программа по умолчанию выводит результат в файл ошибок (`stderr` - прим.

И.П.). Благодаря этому вы можете перехватить введённый текст для последующей его обработки.

3) Организация меню

Следующая программа позволяет вам организовать список с возможностью выбора одного из элементов:

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

$DIALOG --clear --title "Мои любимые исполнители" \
--menu "Все любят песни хинди, поэтому выбирайте:" 20 51 4 \
"Rafi" "Mohammed Rafi" \
"Mukesh" "Mukesh" \
"Kishore" "Kishore Kumar" \
"Saigal" "K L Saigal" \
"Lata" "Lata Mangeshkar" \
"Yesudas" "K J Yesudas" 2> $tempfile

retval=$?
```

```
choice=`cat $tempfile`

case $retval in
  0)
    echo "Да вы эстет! '$choice' -- это лучшее, что вы слышали в своей
жизни!";;
  1)
    echo "Отказ от ввода.";;
  255)
    echo "Нажата клавиша ESC.";;
esac
```

Логика работы скрипта аналогична той, что реализована в скрипте `inputbox.sh` -- результат выполнения скрипта перенаправляется во временный файл, откуда он может быть взят для дальнейшей обработки.

4) Списки зависимых кнопок (radiolist) и флажков (checklist).

Формирование таких списков аналогично организации меню, описанного в предыдущем разделе.

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

$DIALOG --backtitle "Не стесняйтесь, выберите любимого певца" \
  --title "Выбор исполнителя" --clear \
  --radiolist "Мой любимый певец, это... " 20 61 5 \
  "Rafi" "Mohammed Rafi" off \
  "Lata" "Lata Mangeshkar" ON \
  "Hemant" "Hemant Kumar" off \
  "Dey" "MannaDey" off \
  "Kishore" "Kishore Kumar" off \
  "Yesudas" "K. J. Yesudas" off 2> $tempfile

retval=$?
```

```
choice=`cat $tempfile`
case $retval in
  0)
    echo "Ого! Кто бы мог подумать, но выбор пал на '$choice'";;
  1)
    echo "Отказ от ввода.";;
  255)
    echo "Нажата клавиша ESC.";;
esac
```

Для того, чтобы использовать список флажков, вместо радиокнопок, замените в скрипте опцию `--radiolist` на `--checklist`.

5) Создание индикатора

Такой элемент позволяет визуализировать процесс выполнения вашего скрипта:

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

COUNT=10
(
while test $COUNT != 110
do
echo $COUNT
echo "XXX"
echo "Новое сообщение ($COUNT процентов)"
echo "Строка 2"
echo "XXX"
COUNT=`expr $COUNT + 10`
sleep 1
done
) |
$DIALOG --title "Индикатор" --gauge "А вот пример простейшего
индикатора" 20 70 0
```

Особенность реализации индикатора заключается в том, что программа dialog получает данные через конвейер от кода, который заключён внутри

круглых скобок. Есть два момента, на которые необходимо обратить внимание. Первый -- это обязательное использование переменной \$COUNT.

Именно из неё dialog/Xdialog считывает текущее значение индикатора.

При этом желательно, чтобы значение переменной колебалось в диапазоне

от 0 до 100. Второй -- это использование строк вида "XXX" в качестве

ограничителей сообщения, выводящегося на экран.

(В реализации этого элемента есть ошибка. Если используется dialog, то

всё отрабатывается корректно, а в случае с Xdialog строка "Строка 2"

не переносится на новую строку. Использование тоже \n не помогает.

-- прим. А.К.)

6) Выбор файла

Вот пример простейшего диалога для выбора файла:

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

FILE=`$DIALOG --stdout --title "Выберите файл" --fselect $HOME/ 10 60`

case $? in
0)
echo "Выбран \"$FILE\"";;
```

```
1)
    echo "Отказ от ввода.>";;
255)
    echo "Нажата клавиша ESC.>";;
esac
```

Обратите внимание, что в этом примере используется другой механизм получения данных от dialog. По умолчанию большинство элементов, представленных в dialog, возвращают значение через stderr, поэтому в предыдущих примерах использовалось перенаправление данных с stderr в файл. Используя опцию --stdout, можно сразу перенаправлять данные на стандартный вывод, что и демонстрируется в последнем примере.

Диалоговое окно для выбора файла состоит из панелей. Вы можете перемещаться между ними при помощи клавиши "Tab". Кроме этого, у вас есть возможность вводить данные непосредственно в строке ввода, расположенной под панелями.

(Правда есть пара "но", на которые следует обратить внимание. На первое "но" вы наступите, если попробуете последний пример выполнить, используя Xdialog. Получается, что добиться 100%-го портирования из текста в графику невозможно -- приходится менять размер окна. Второе "но" -- это навигация по файловому дереву. В текстовом режиме (dialog) у меня это не получилось -- клавиша "Enter" воспринимается как окончательный выбор и использовать её для перемещения по подкаталогам не удаётся. В графике (Xdialog) с этим проще -- там поддерживается мышка и перемещение по подкаталогам не вызывает проблем. -- прим. А.К.)

7) Календарь и настройка часов

а) Календарь

Информация о годе, месяце и дне выводится на отдельных панелях. Если

значение дня, месяца или года не указано, либо оно отрицательное, то используются системная дата. (Работает только в dialog. При частичном

отсутствии начальных значений даты (например, вы не указали год) Xdialog выдаёт сообщение об ошибке. Указать в качестве начального значения, например, 1000-й год не получится -- dialog воспринимает это как неверное значение и указывает текущую дату. Xdialog в этом случае выдаёт сообщение об ошибке. -- прим. А.К.) Для изменения значений можно использовать стрелки управления курсором, либо воспользоваться горячими клавишами, используемыми в vi при навигации

по тексту: h, j, k и l. (Верно для dialog. В Xdialog используются управление при помощи мышки и только навигация по дням месяца возможна

при помощи клавиш управления курсором. -- прим. А.К.) Если год устанавливается равным 0, то по умолчанию используется значение текущего года. Результат выводится в формате день/месяц/год

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

USERDATE=`$DIALOG --stdout --title "Календарь" --calendar "Выберите
дату..." 00 7 7 1981`

case $? in
  0)
    echo "Выбрано: $USERDATE.>";;
  1)
    echo "Отказ от ввода.>";;
  255)
    echo "Нажата клавиша ESC.>";;
esac
```

б) Настройка часов

Этот диалог позволяет вам выбирать время:

```
#!/bin/sh

DIALOG=${DIALOG=Xdialog}
USERTIME=`$DIALOG --stdout --title "Настройка часов" \
    --timebox "Укажите, пожалуйста, время..." 0 0 12 34 56`

case $? in
  0)
    echo "Указано время: $USERTIME.>";;
  1)
    echo "Отказ от ввода.>";;
  255)
    echo "Нажата клавиша ESC.>";;
esac
```

8) Другие возможности

Кроме этого, Xdialog располагает такими элементами, как деревья (tree-view), выбор значения из заданного диапазона (range-box), редактор текстовых файлов (edit-box) и т.п. За детальной информацией

обращайтесь по адресу <http://thgodef.nerim.net/xdialog/doc/box.html>. Не забудьте заглянуть в справочное руководство для dialog -- оно содержит интересную информацию о таких возможностях как ввод пароля (password box), просмотр файла (tailbox) и т.д. Также у вас есть возможность манипулировать внешним видом окна, меняя цвета, добавляя/убирая тени и т.п.

9) Подсказки

Ваш скрипт сможет самостоятельно делать выбор между dialog и Xdialog, если в его начале дописать следующую конструкцию:

```
if [ -z $DISPLAY ]
then
    DIALOG=dialog
else
    DIALOG=Xdialog
fi
```

Попробуйте запустить скрипт, предложенный ниже, в консоли и "иксах".

```
#!/bin/sh
if [ -z $DISPLAY ]
then
    DIALOG=dialog
else
    DIALOG=Xdialog
fi
$DIALOG --yesno "Забавно, не правда ли?" 0 0
```

10) Ссылки

1) Страницы справочного руководства dialog:

<http://hightek.org/dialog/manual-0.9a-20010429.html>

Обязательно прочтите их (или `man dialog`), если планируете писать скрипты, используя dialog.

2) Примеры скриптов: <http://www.fifi.org/doc/dialog/examples/>.

Все примеры, представленные здесь, являются модифицированными скриптами, взятыми по этому адресу. Если вы используете Debian GNU/Linux, то эти примеры вы найдёте в `/usr/share/doc/dialog/examples`.

3) Страница Thomas'a Dickey: <http://dickey.his.com/dialog/>

4) Страница Vincent'a Stemen'a: <http://hightek.org/dialog/>. Эта страница содержит исчерпывающую информацию о различных версиях dialog.

5) Документация по Xdialog:

<http://thgodef.nerim.net/xdialog/doc/index.html>.

На этой странице вы найдёте полную информацию о функциональных возможностях Xdialog.

Sunil Thomas Thonikuzhiyi

Я работаю консультантом по информационным технологиям при

законодательном собрании штата Керала, Тривандрам, Индия. На Linux
я работаю с 1996. Получил высшее образование на факультете
компьютерных
наук университета города Кочин (Cochin). Интересуюсь всеми типами
операционных систем. В свободное время люблю слушать народную
индийскую музыку.

Copyright © 2004, Sunil Thomas Thonikuzhiyil. Copying license
<http://linuxgazette.net/copying.html>
Published in Issue 101 of Linux Gazette, April 2004

<< Предыдущая	ИНДЕКС	Поиск в статьях	src	Установить закладку	Перейти на закладку	Следующая >>
------------------	--------	--------------------	-----	------------------------	------------------------	-----------------