

Как создавать диалоговые блоки в интерактивных shell скриптах

[Alexey](#) 27.12.2014 0 [Утилиты](#) [Bash](#), [shell](#), [whiptail](#)

Когда вы устанавливаете новые программы в терминале, вы могли часто видеть появляющиеся информативные диалоговые блоки для обратной связи с вами. Типы диалоговых блоков различны: от простых диалогов да/нет до полей ввода, блоков паролей, чеклистов, меню и так далее. Преимущества использования таких дружелюбных к пользователю блоков очевидны, ведь они интуитивно направляют вас для ввода необходимой информации.

Когда вы пишете интерактивный шелл скрипт, вы на самом деле можете использовать такие диалоговые блоки для принятия ввода пользователя. Будучи предварительно установленной на всех современных дистрибутивах Linux, программа под названием whiptail может упростить процесс создания шелл скриптом диалоговых и информационных блоков в терминале наподобие тех, что графических интерфейсов, что кодируют Zenity или Xdialog для скриптов. В этом уроке я опишу как с использованием whiptail создать дружелюбный пользователю диалоговые блоки в шел скрипте. Я также покажу Баш фрагменты кода различных диалоговых блоков, поддерживаемых в whiptail.

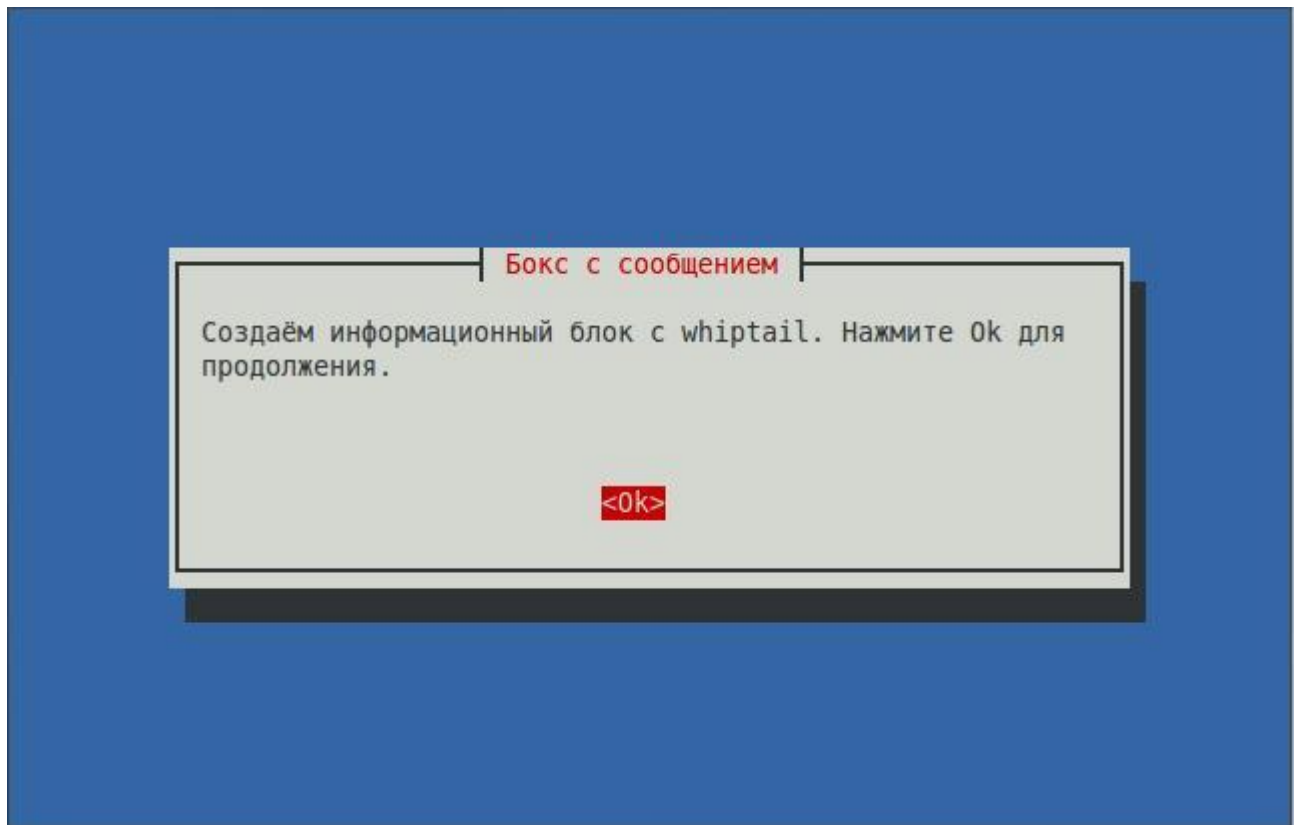
Создаём блок с сообщением

Блок с сообщением показывает произвольный текст с кнопкой подтверждения для продолжения.

```
1 whiptail --title "<название бокса>" --msgbox "<текст для показа>" <высота> <ширина>
```

Пример:

```
1 #!/bin/bash
2 whiptail --title "Бокс с сообщением" --msgbox "Создаём информационный блок с whiptail" 60
```



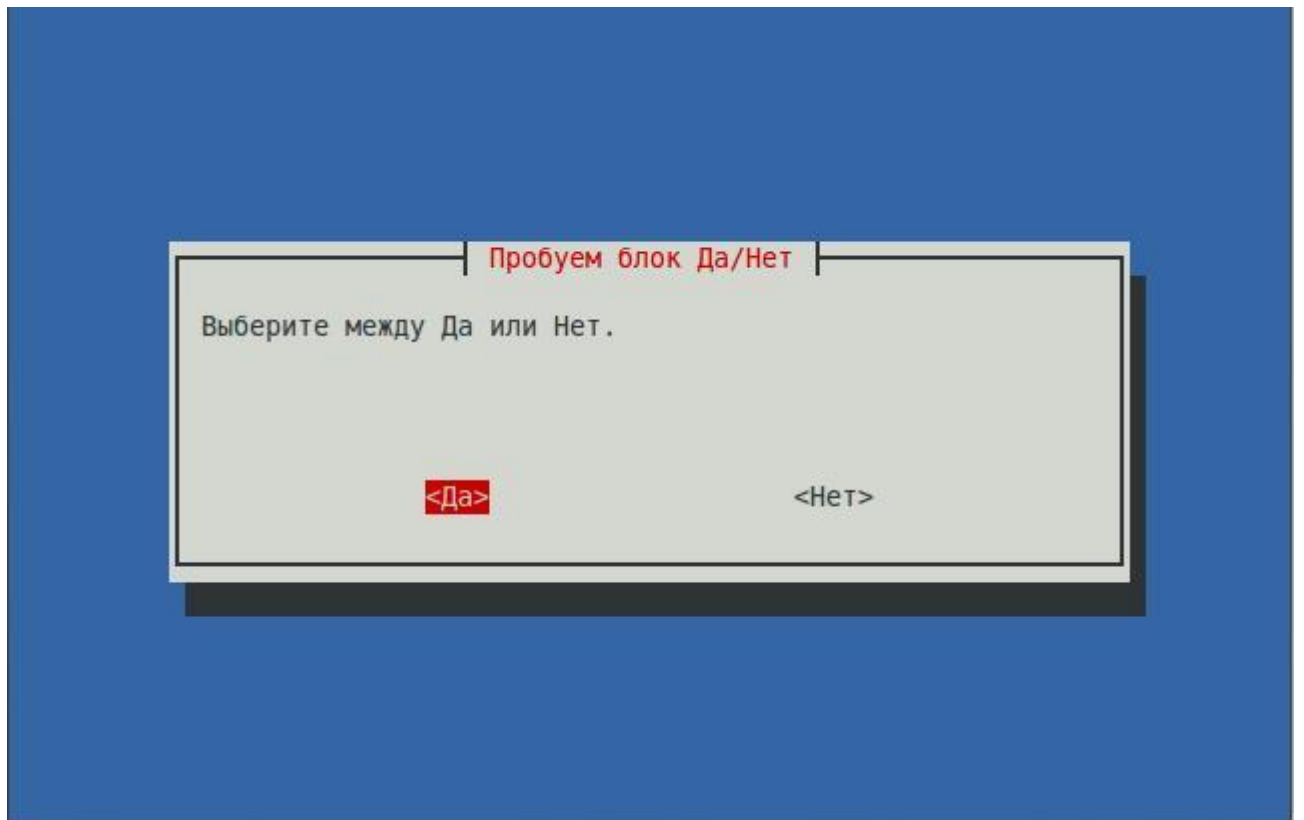
Создание Да/Нет блоков

Одним из частных пользовательских вводов является Да или Нет. В этом случае может быть использован диалоговый блок Да/Нет.

```
1    whiptail --title "<название диалогового блока>" --yesno "<текст для показа>" <выс
```

Пример:

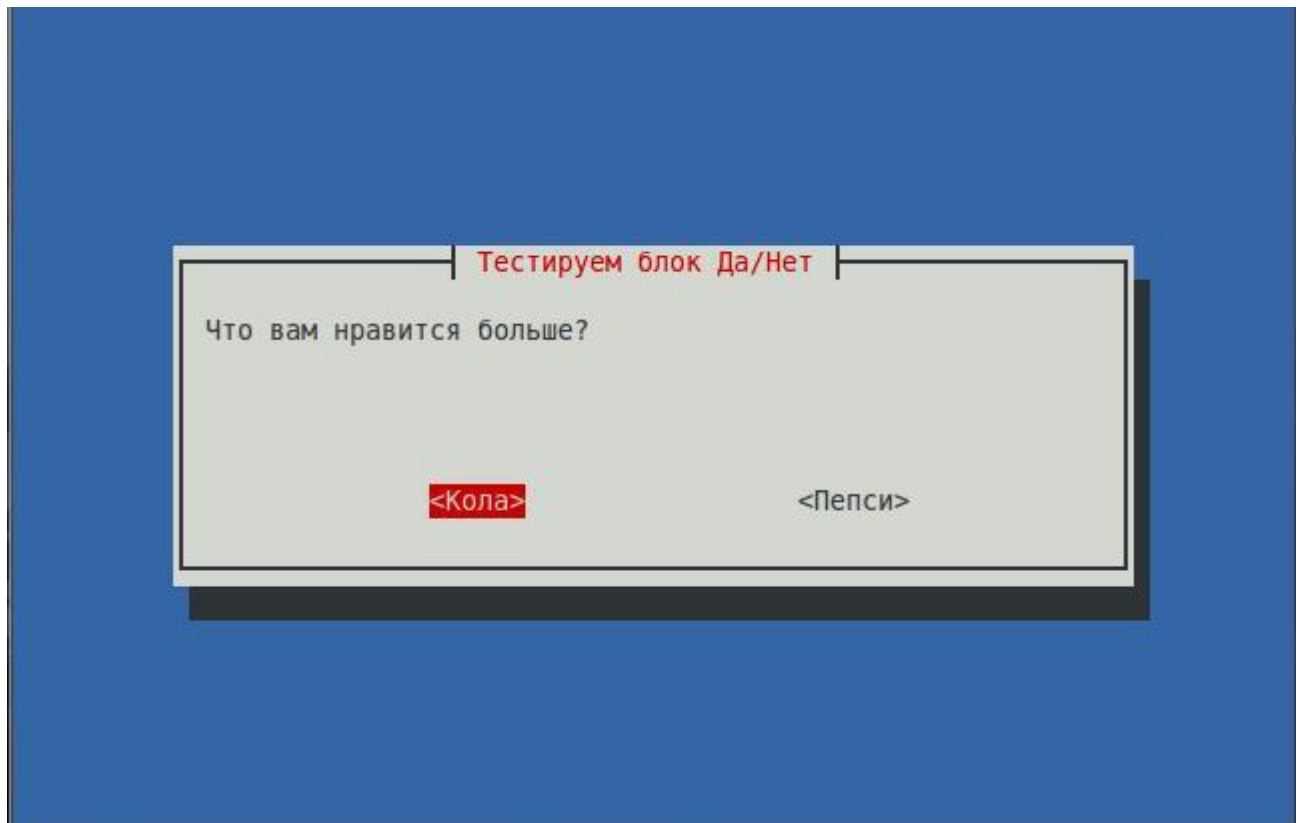
```
1    #!/bin/bash
2    if (whiptail --title "Пробуем блок Да/Нет" --yesno "Выберите между Да или Нет." 10
3    echo "Вы выбрали Да. Статус выхода был $?."
4    else
5    echo "Вы выбрали нет. Статус выхода был $?."
6    fi
```



Опционально вы можете настроить текст кнопок Да и Нет с помощью опций "--yes-button" and "--no-button"

Например:

```
1  #!/bin/bash
2  if (whiptail --title "Тестируем блок Да/Нет" --yes-button "Кола" --no-button "Пепс
60) then
3  echo "Вы выбрали Колу. Статус выхода был $?."
4  else
5  echo "Вы выбрали Пепси. Статус выхода был $?."
6  fi
```



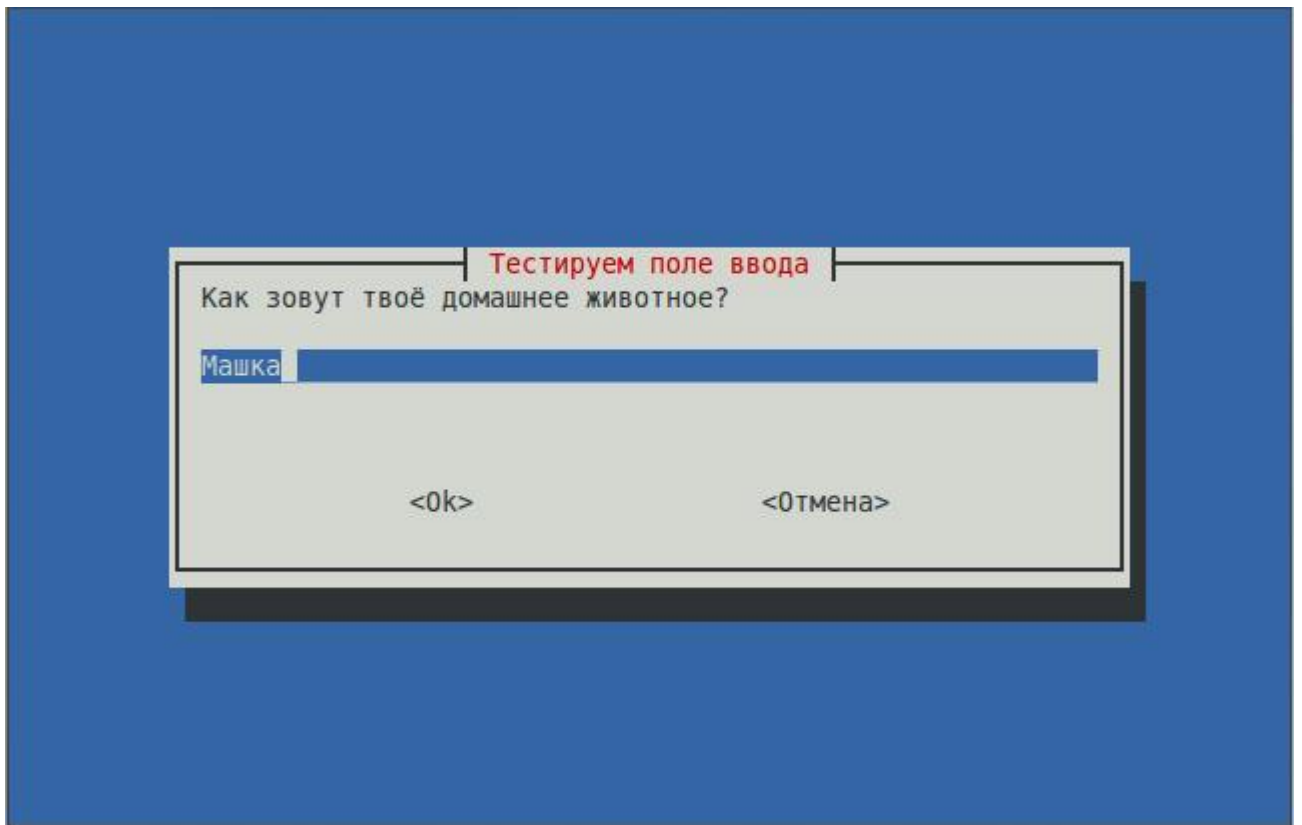
Создаём блок свободного ввода

Если вы хотите принять любой произвольный текст от пользователя, вы можете использовать этот блок ввода

```
1 whiptail --title "<Название блока ввода>" --inputbox "<Текст для показа>" <высота>
```

Например:

```
1  #!/bin/bash
2
3  PET=$(whiptail --title "Тестируем поле ввода" --inputbox "Как зовут твоё домашнее
4  exitstatus=$?
5  if [ $exitstatus = 0 ]; then
6  echo "Имя твоего домашнего животного:" $PET
7  else
8  echo "Ты выбрал отмену."
9  fi
```



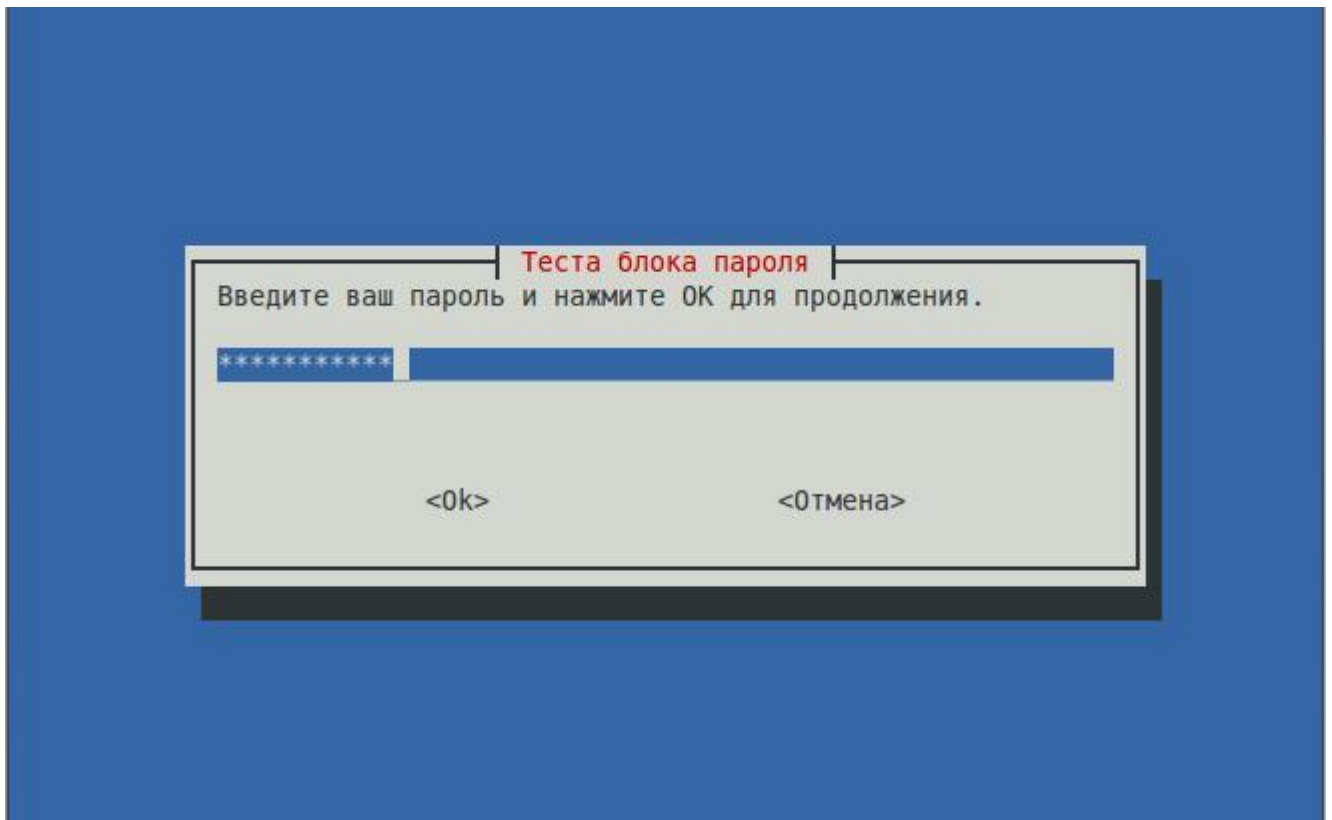
Создание блока для ввода пароля

Блок пароля полезен, когда вы хотите получить чувствительную информацию от пользователя.

```
1 whiptail --title "<Название блока пароля>" --passwordbox "<текст для отображения>"
```

Пример:

```
1  #!/bin/bash
2  PASSWORD=$(whiptail --title "Теста блока пароля" --passwordbox "Введите ваш пароль"
3>&1 1>&2 2>&3)
3  exitstatus=$?
4  if [ $exitstatus = 0 ]; then
5  echo "Ваш пароль:" $PASSWORD
6  else
7  echo "Вы выбрали отмену."
8  fi
```



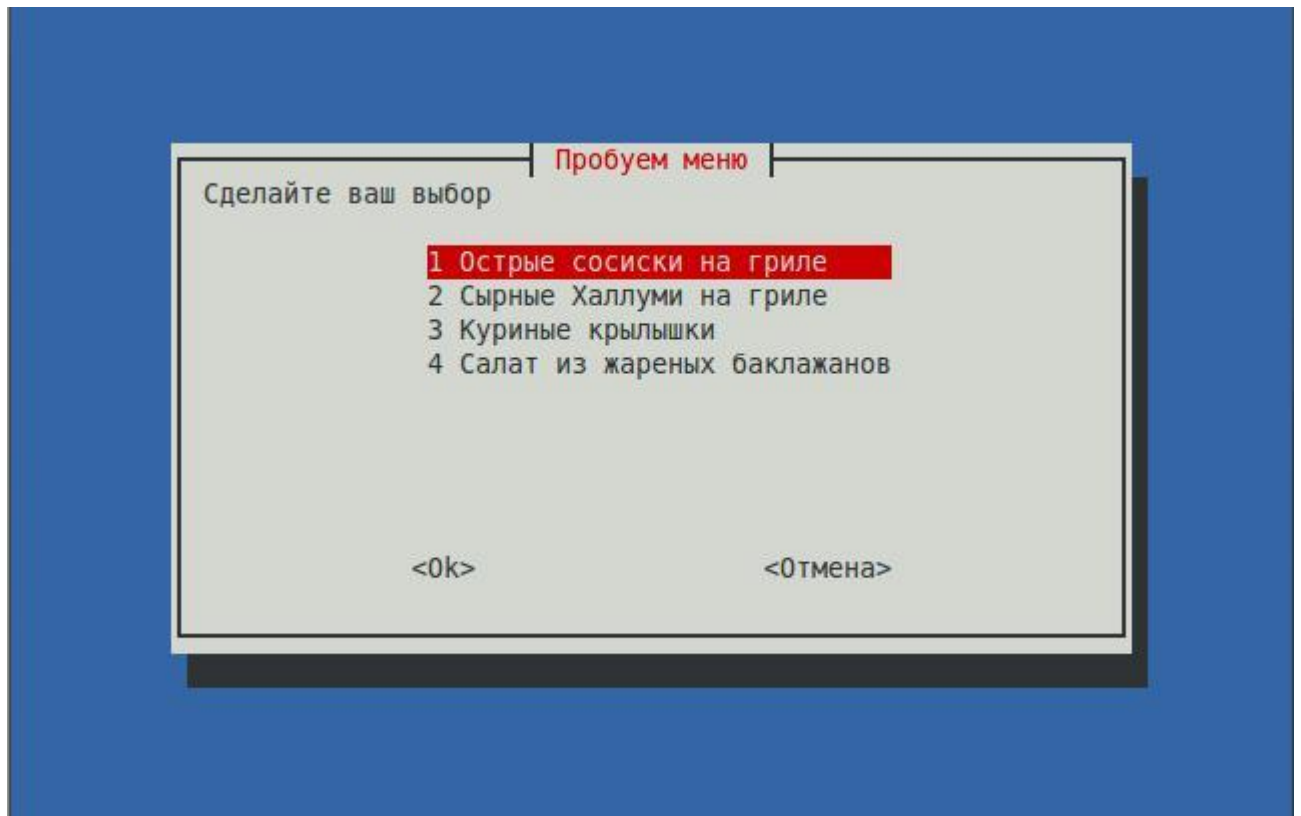
Создание меню

Когда вы хотите спросить пользователя выбрать один из нескольких произвольных вариантов, вы можете использовать меню.

```
1 whiptail --title "<название меню>" --menu "<текст для отображения>" <высота> <ширина>
```

Пример:

```
1    #!/bin/bash
2    OPTION=$(whiptail --title "Пробуем меню" --menu "Сделайте ваш выбор" 15 60 4 \
3    "1" "Острые сосиски на гриле" \
4    "2" "Сырные Халлуми на гриле" \
5    "3" "Куриные крылышки" \
6    "4" "Салат из жареных баклажанов" 3>&1 1>&2 2>&3)
7    exitstatus=$?
8    if [ $exitstatus = 0 ]; then
9    echo "Вы выбрали:" $OPTION
10   else
11   echo "Вы нажали отмену."
```



Создание диалога радиолист

Блок радиолист схож по смыслу с меню в том смысле, что вы можете выбрать только одну опцию из списка доступных опций. В отличие от блока меню, тем не менее, вы можете отобразить, какая опция выбрана по умолчанию, определив её статус.

```
1 whiptail --title "<заголовок радиолиста>" --radiolist "<текст для показа>" <высота>
```

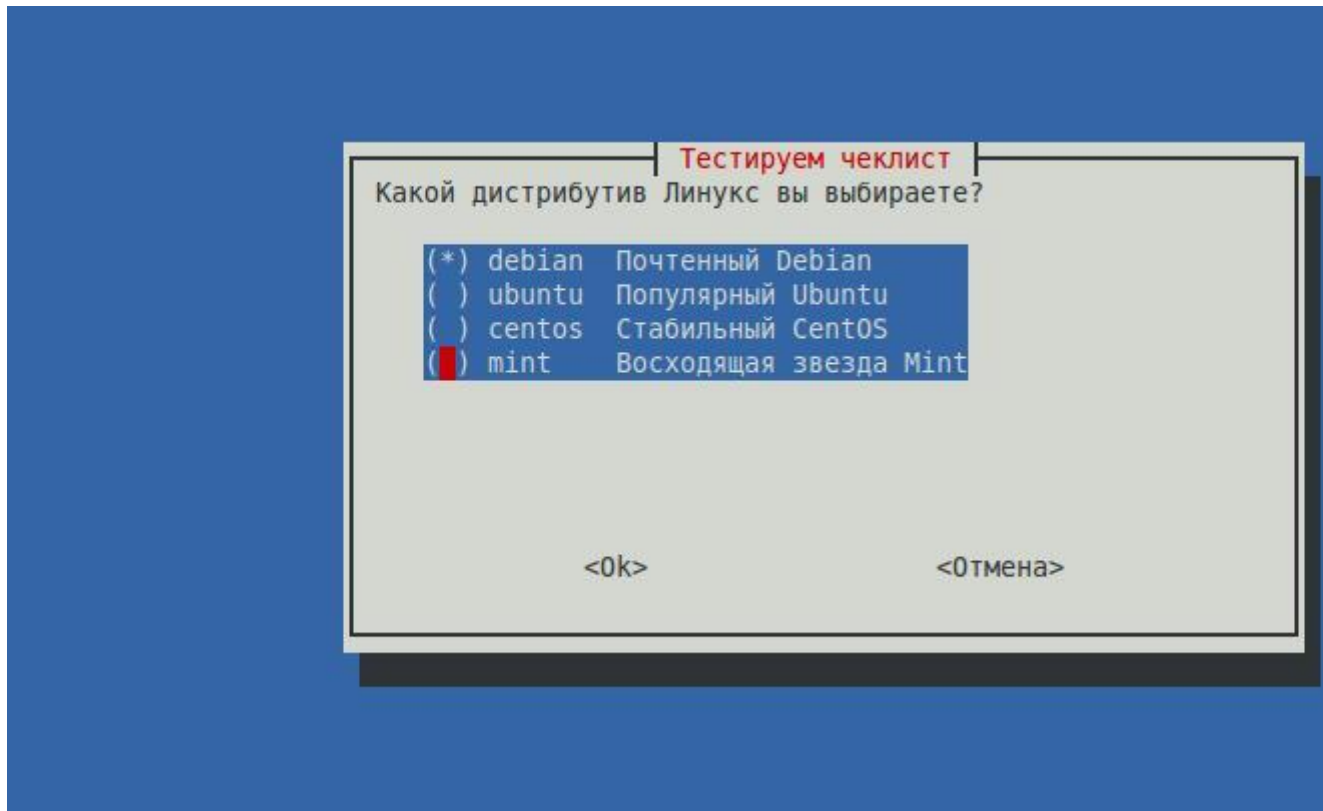
Например:

```
1  #!/bin/bash
2  DISTROS=$(whiptail --title "Тестируем чеклист" --radiolist \
3    "Какой дистрибутив Линукс вы выбираете?" 15 60 4 \
4    "debian" "Почтенный Debian" ON \
5    "ubuntu" "Популярный Ubuntu" OFF \
6    "centos" "Стабильный CentOS" OFF \
7    "mint" "Восходящая звезда Mint" OFF 3>&1 1>&2 2>&3)
8  exitstatus=$?
9  if [ $exitstatus = 0 ]; then
    echo "Выбранный дистрибутив: "$DISTROS
```

```

10  else
11  echo "Вы нажали отмену."
12  fi
13

```



Создаём диалок чеклиста

Чеклист полезен, когда вы хотите попросить пользователя выбрать более чем один вариант из списка опций, он отличается от радиолиста, в котором можно сделать только один выбор.

```

1  whiptail --title "<заголовок чеклиста>" --checklist "<текст для показа>" <высота>

```

Пример:

```

1  #!/bin/bash
2  DISTROS=$(whiptail --title "Тестируем чеклист" --checklist \
3  "Выберите предпочитаемый дистрибутив Линукс" 15 60 4 \
4  "debian" "Почтенный Debian" ON \
5  "ubuntu" "Популярная Ubuntu" OFF \
6  "centos" "Стабильная CentOS" ON \
7  "mint" "Восходящая звезда Mint" OFF 3>&1 1>&2 2>&3)
8  exitstatus=$?

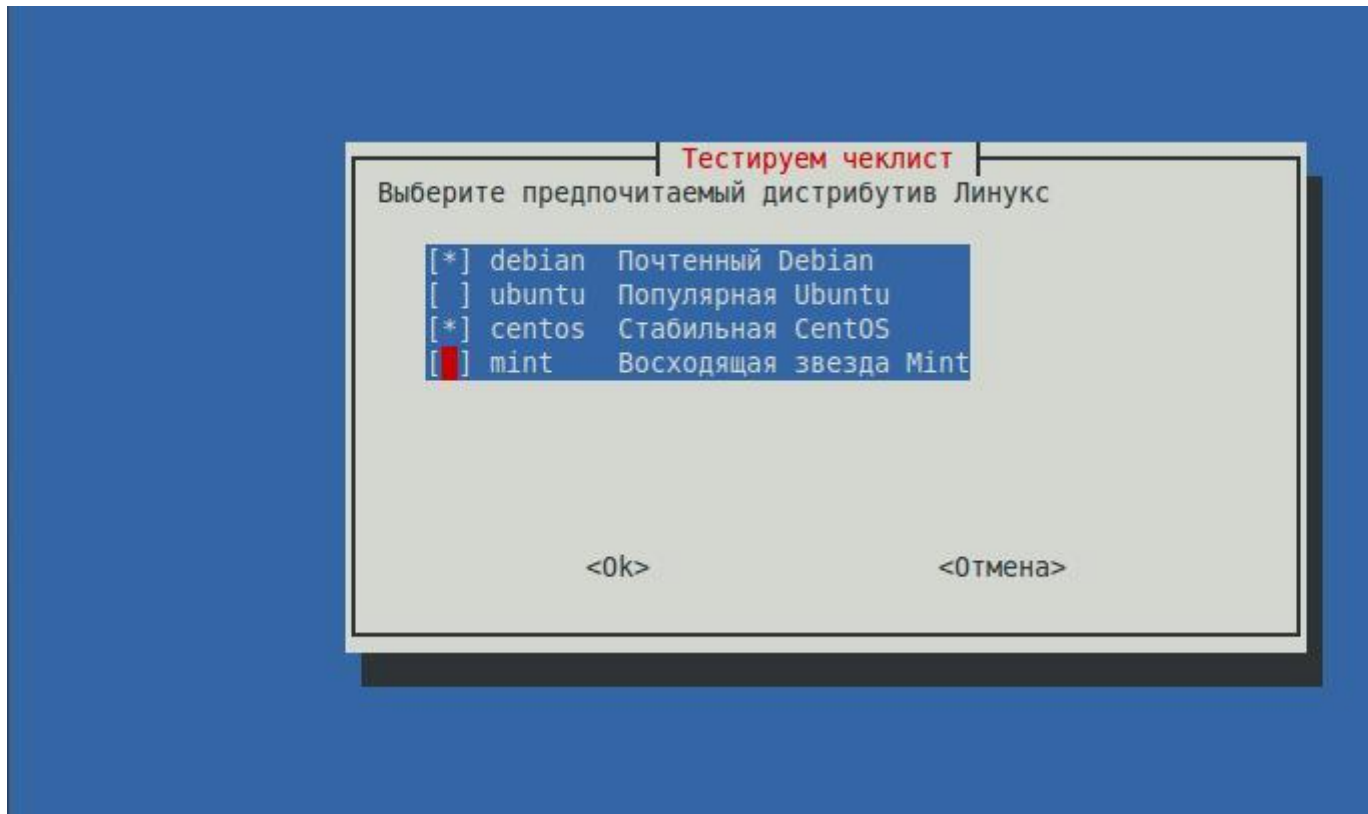
```



```

8     if [ $exitstatus = 0 ]; then
9         echo "Ваш любимый дистрибутив:" $DISTROS
10        else
11            echo "Вы выбрали отмену."
12        fi
13

```



Сооздание строки прогресса

Другой дружественный пользователю диалоговый блок это прогресс бар. `whiptail` читает из стандартного ввода проценты (от 0 до 100) и отображает соответствующий индикатор датчика.

```

1     whiptail --gauge "<текст для отображения>" <высота> <ширина> <начальный процент>

```

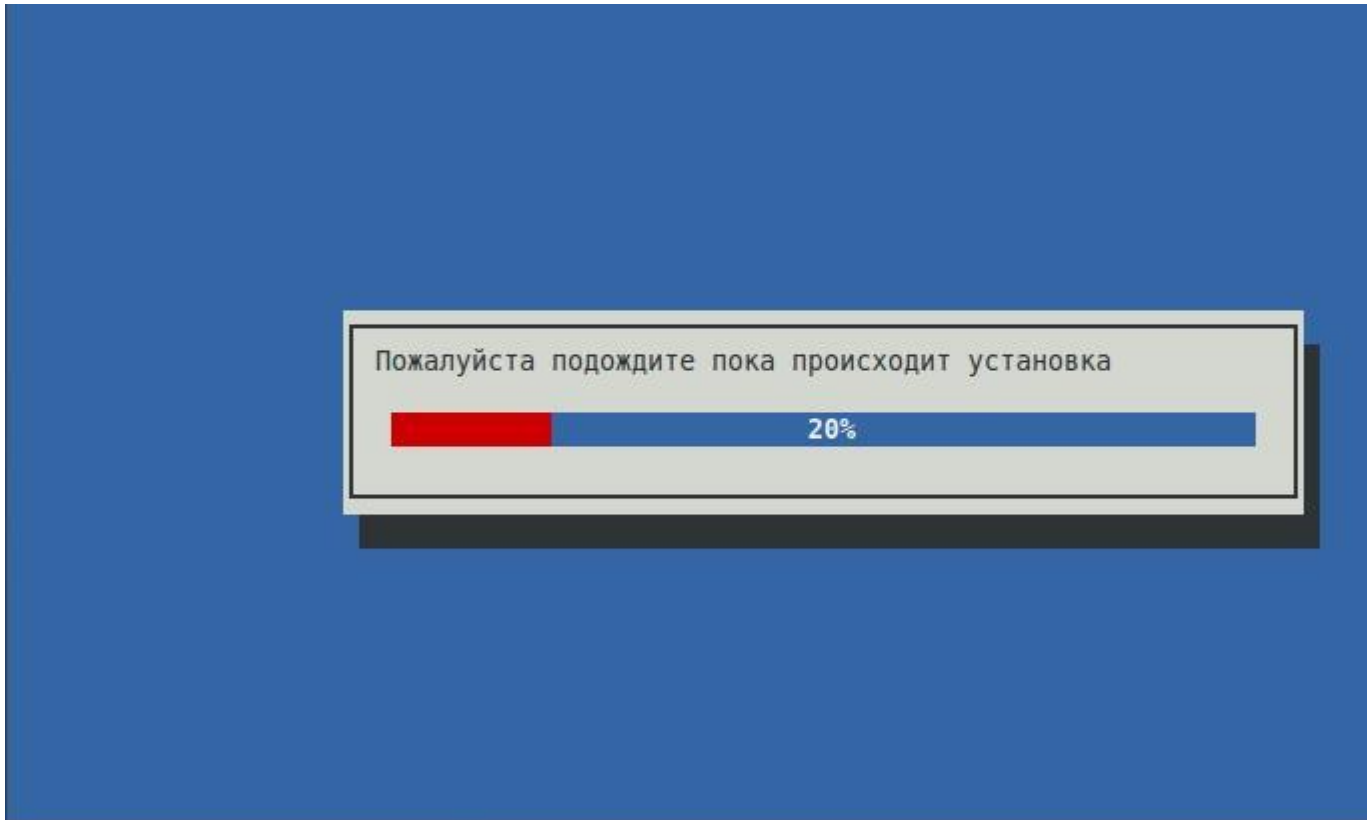
Пример:

```

1     #!/bin/bash
2     {
3         for ((i = 0 ; i <= 100 ; i+=20)); do
4             sleep 1
5             echo $i
6         done
7     }

```

```
5     done
6 } | whiptail --gauge "Please wait while installing" 6 60 0
7
```



Теперь вы увидели, как легко создавать полезные диалоговые блоки в интерактивных шелл скриптах. В следующий раз, когда вам понадобится написать шелл скрипт для кого-то,

почему бы не попробовать whiptail и не впечатлить его или её?

п.с. Все примеры, кроме последнего, одинаково хорошо работают и в bash, и в sh, Но последний пример не будет работать в простом sh из-за конструкции ($i = 0$; $i \leq 100$; $i+=20$). Поэтому для простого sh подойдёт следующий код:

```
1  #!/bin/sh
2  {
3      for i in $(seq 0 20 100); do
4          sleep 1
5          echo $i
6      done
7  } | whiptail --gauge "Please wait while installing" 6 60 0
```

