

Dialog Tutorial

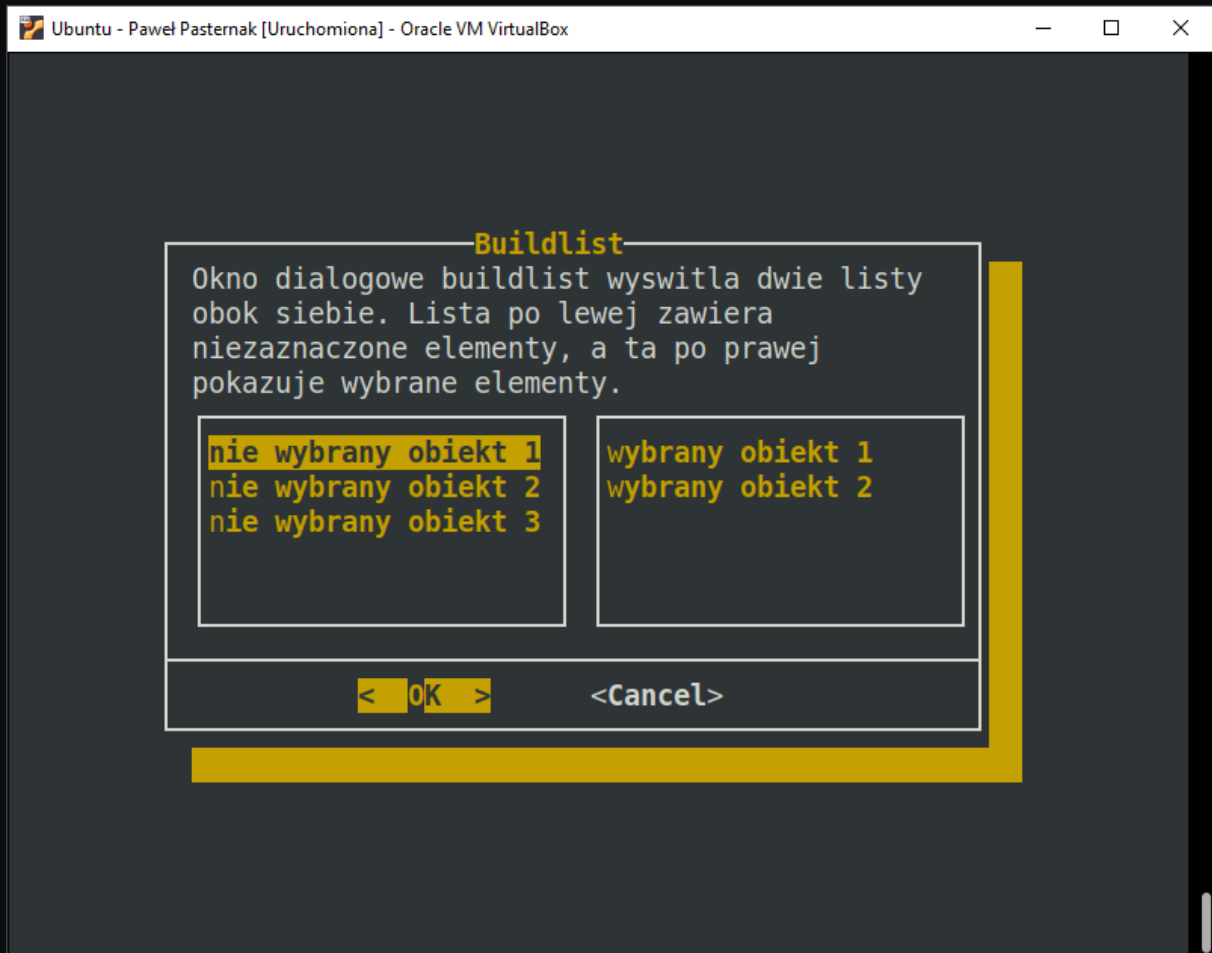
By: Paweł Pasternak

Spis treści

Rodzaje okienek:	2
1. Buildlist	2
2. Calendar	3
3. Checklist	4
4. DSelect	5
5. EditBox	6
6. FSelect	7
7. Gauge	8
8. InfoBox	11
9. InputBox	12
10. InputMenu	13
11. Menu	14
12. MixedGauge	15
13. MsgBox	16
Wyprowadzanie informacji z okienek:	17
1. Przykład wyprowadzenia informacji z InputBoxa	17
2. Przykładowe menu z oknem wiadomości	18

Rodzaje okienek:

1. Buildlist



Kod:

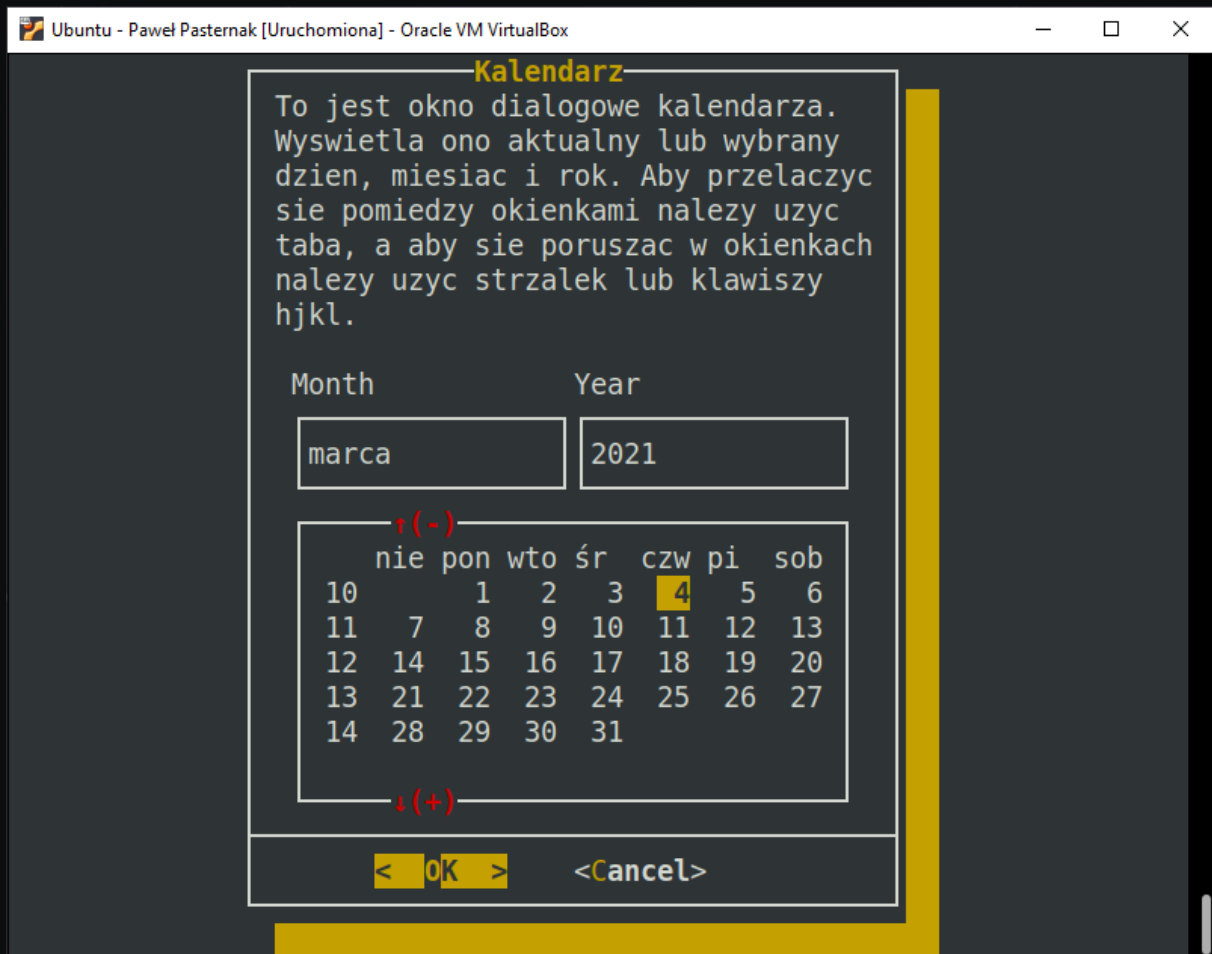
```
#!/bin/bash
```

```
dialog --title "Buildlist" \
  --buildlist "Okno dialogowe buildlist wyswitla dwie listy obok siebie. \
    Lista po lewej zawiera niezaznaczone elementy, \
    a ta po prawej pokazuje wybrane elementy." 15 50 5 \

  f1 "nie wybrany obiekt 1" off \
  f2 "nie wybrany obiekt 2" off \
  f3 "wybrany obiekt 1" on \
  f4 "nie wybrany obiekt 3" off \
  f5 "wybrany obiekt 2" on

#cyfry na koncu argumentu buildlist wyznaczaja rozmiar okna
#Aby przejsc do lewej kolumny nalezy wcisnac "^", natomiast aby przejsc do pra-
wej nale-zy wcisna "$". Aby przeniesc opcje z jednej do drugiej kolumny nalezy wy-
brac obiekt za pomocą strzałek i kliknac spacje
#Obiekty ktore obok siebie maja znacznik "off" beda sie znadowac w lewej kolum-
nie, natomiast te ze znacznikiem "on" w prawej
#tagi wybranych obiektow beda wyslane jako output do konsoli
```

2. Calendar



Kod:

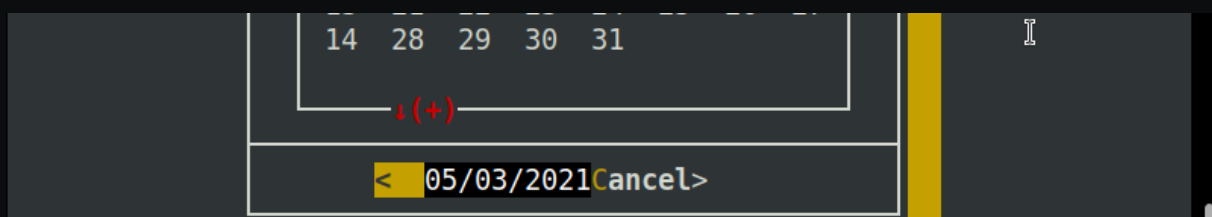
#!/bin/bash

```
dialog --title "Kalendarz" \
  --calendar "To jest okno dialogowe kalendarza. \
  Wyświetla ono aktualny lub wybrany dzień, miesiąc i rok. \
  Aby przełączyć się pomiędzy okienkami należy użyć taba, \
  a aby się poruszać w okienkach należy użyć strzałek lub klawiszy hjkl." 0 0
```

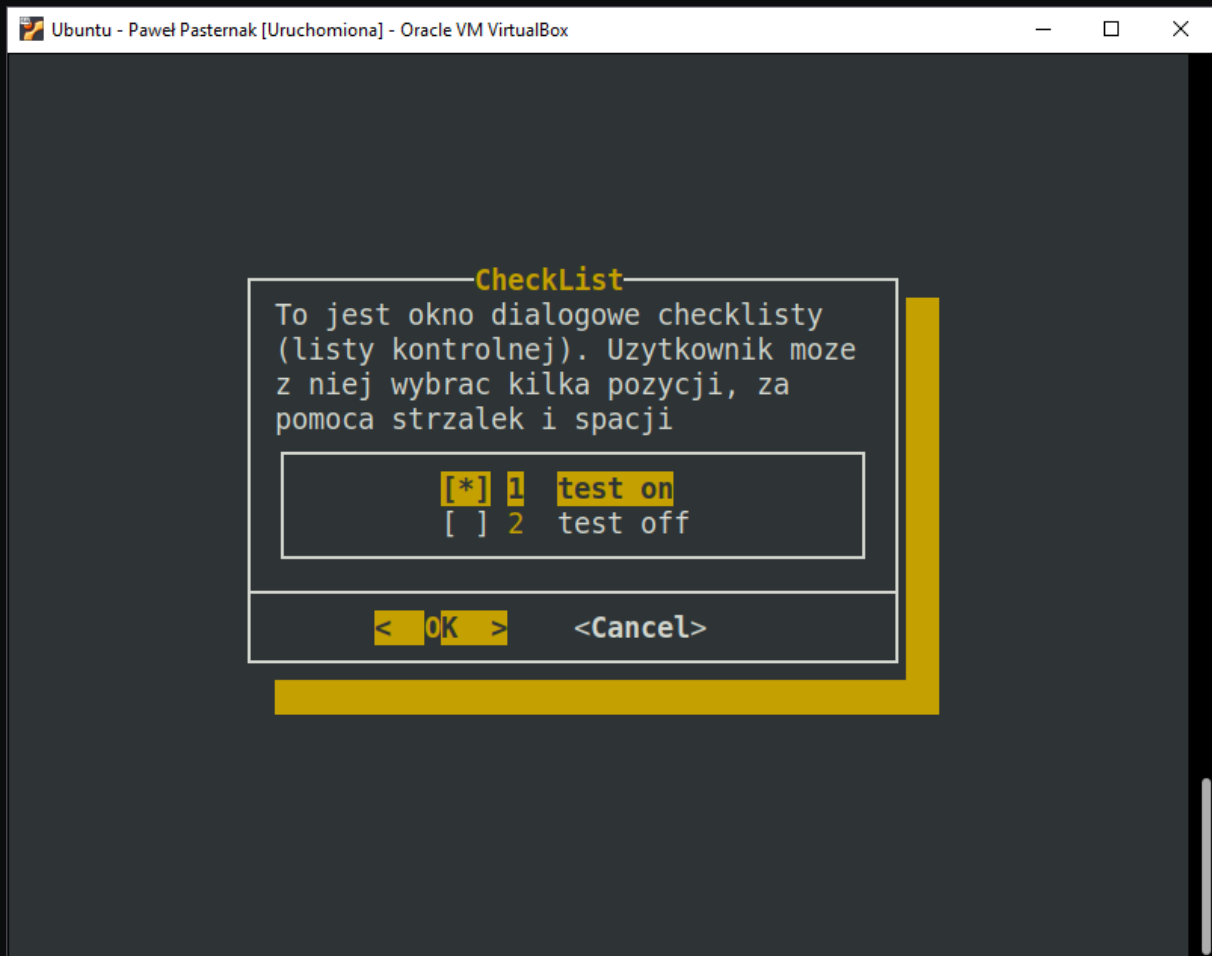
#cyfry na koncu wyznaczają rozmiar okna

jeżeli chcemy ustawić datę to po rozmiarze (2 zera na koncu) należy wprowadzić wybraną datę za pomocą cyfr

#data jest wysyłana do konsoli jako output



3. Checklist



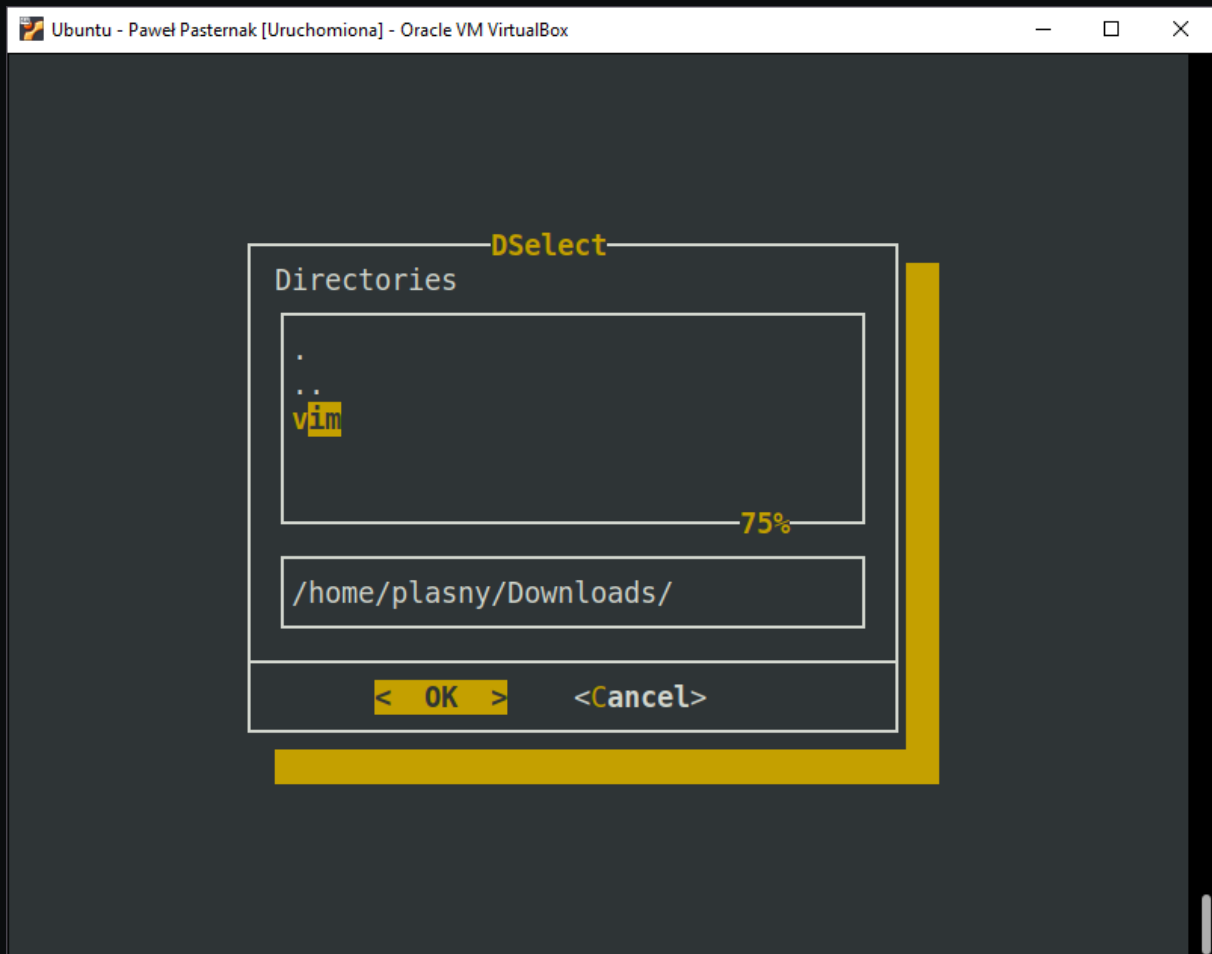
Kod:

```
#!/bin/bash
```

```
dialog --title "CheckList" \
  --checklist "To jest okno dialogowe checklisty (listy kontrolnej). \
    Uzytkownik moze z niej wybrac kilka pozycji, za pomoca strzalek i spacji" \
    10 40 0 \
    1 "test on" on \
    2 "test off" off

#cyfry na koncu wyznaczaja rozmiar okna
#za pomoca znacznikow "on" i "off", okreslamy ktore opcje beda automatycz-
nie wybrane.
```

4. DSelect



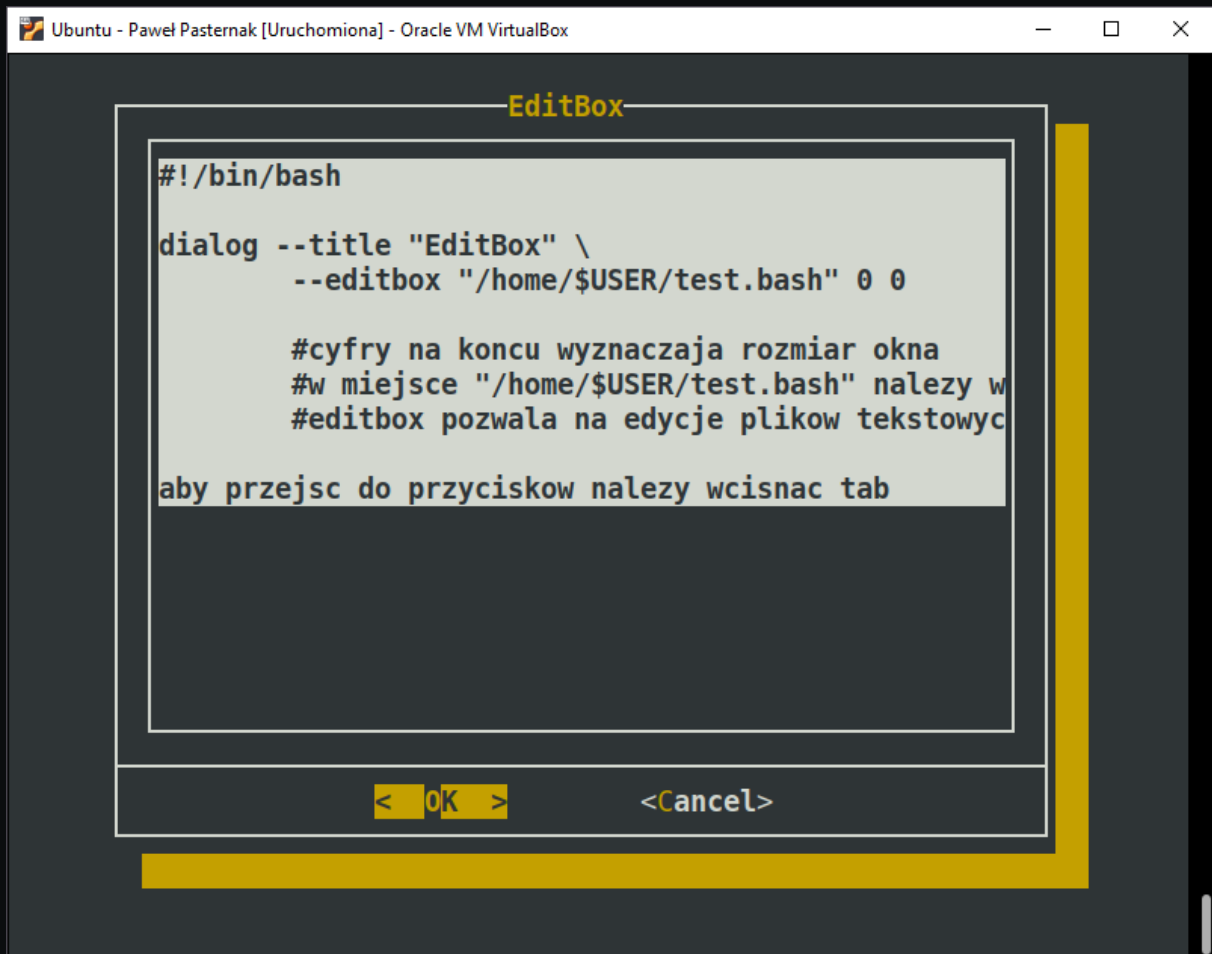
Kod:

```
#!/bin/bash
```

```
dialog --title "DSelect" \  
  --dselect "/home/$USER" 5 40
```

```
#dselect pozwala na wybranie katalogu, aby wybrac plik nalezy uzyc fselect (6)  
#w miejsce "/home/$USER" mozemy wpisac inna sciezke  
#cyfry na koncu wyznaczaja rozmiar okna
```

5. EditBox



Kod:

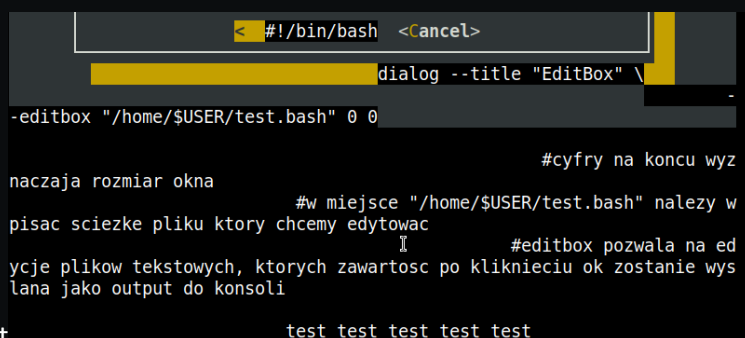
`#!/bin/bash`

```
dialog --title "EditBox" \
  --editbox "/home/$USER/test.bash" 0 0
```

#cyfry na koncu wyznaczaja rozmiar okna

#w miejsce "/home/\$USER/test.bash" nalezy wpisac sciezke pliku ktory chcemy edytowac

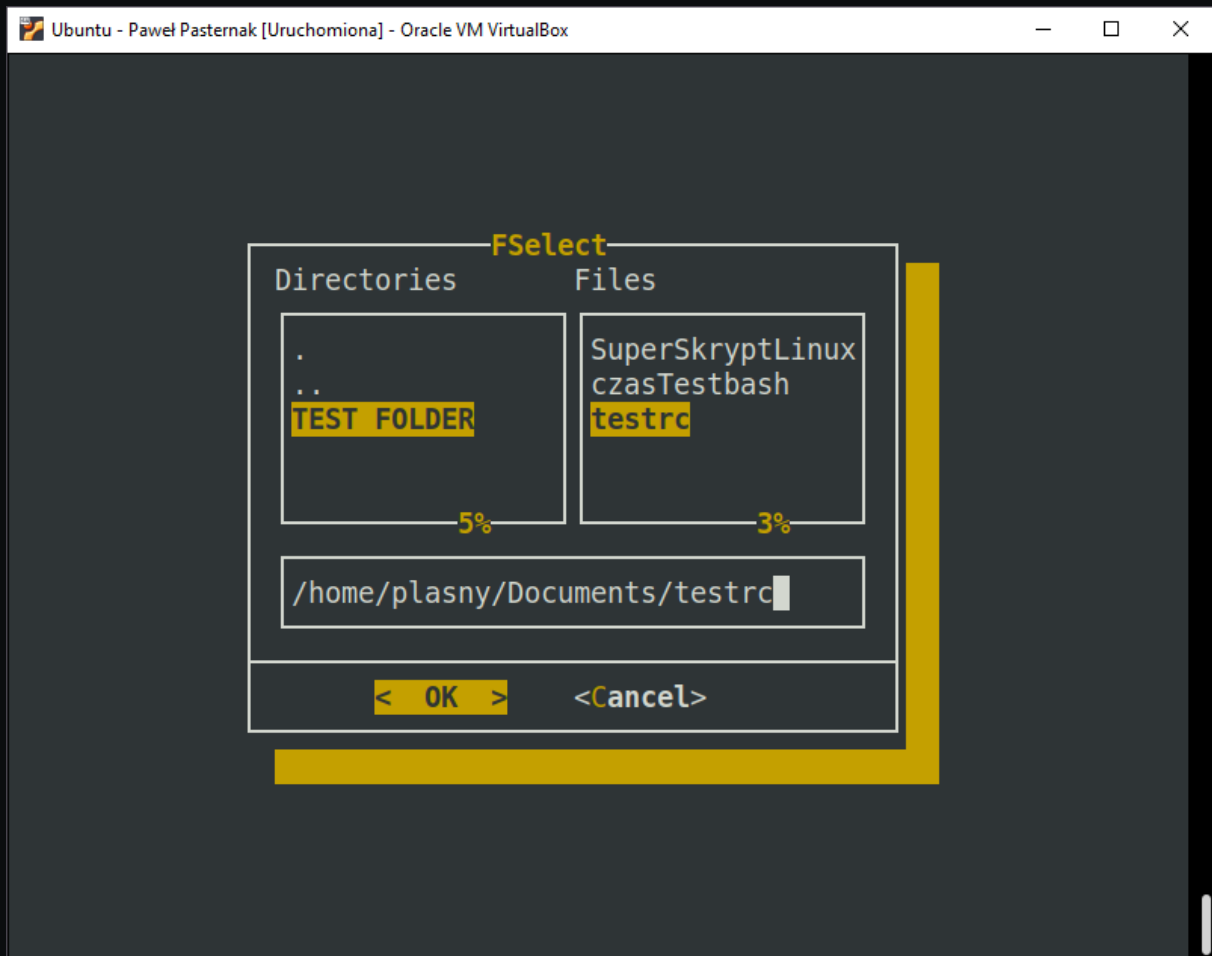
#editbox pozwala na edycje plikow tekstowych, ktorych zawartosc po kliknieciu ok zostanie wyslana jako output do konsoli



Przykładowy output

test test test test test

6. FSelect



Kod:

```
#!/bin/bash
```

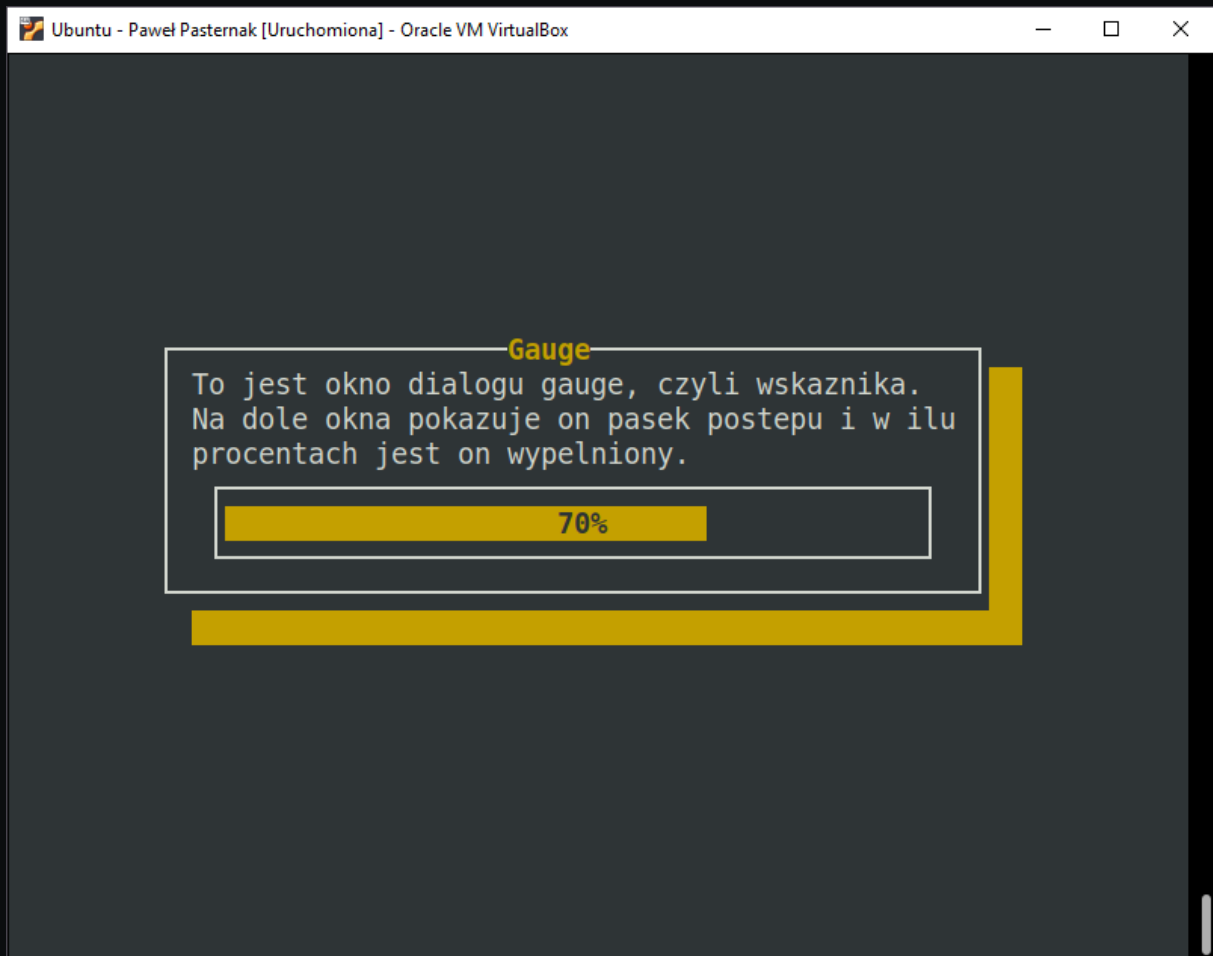
```
dialog --title "FSelect" \  
  --fselect "/home/$USER" 5 40
```

#fselect pozwala na wybranie pliku

#w miejsce "/home/\$USER" mozemy wpisac inna sciezke

#cyfry na koncu wyznaczaja rozmiar okna

7. Gauge



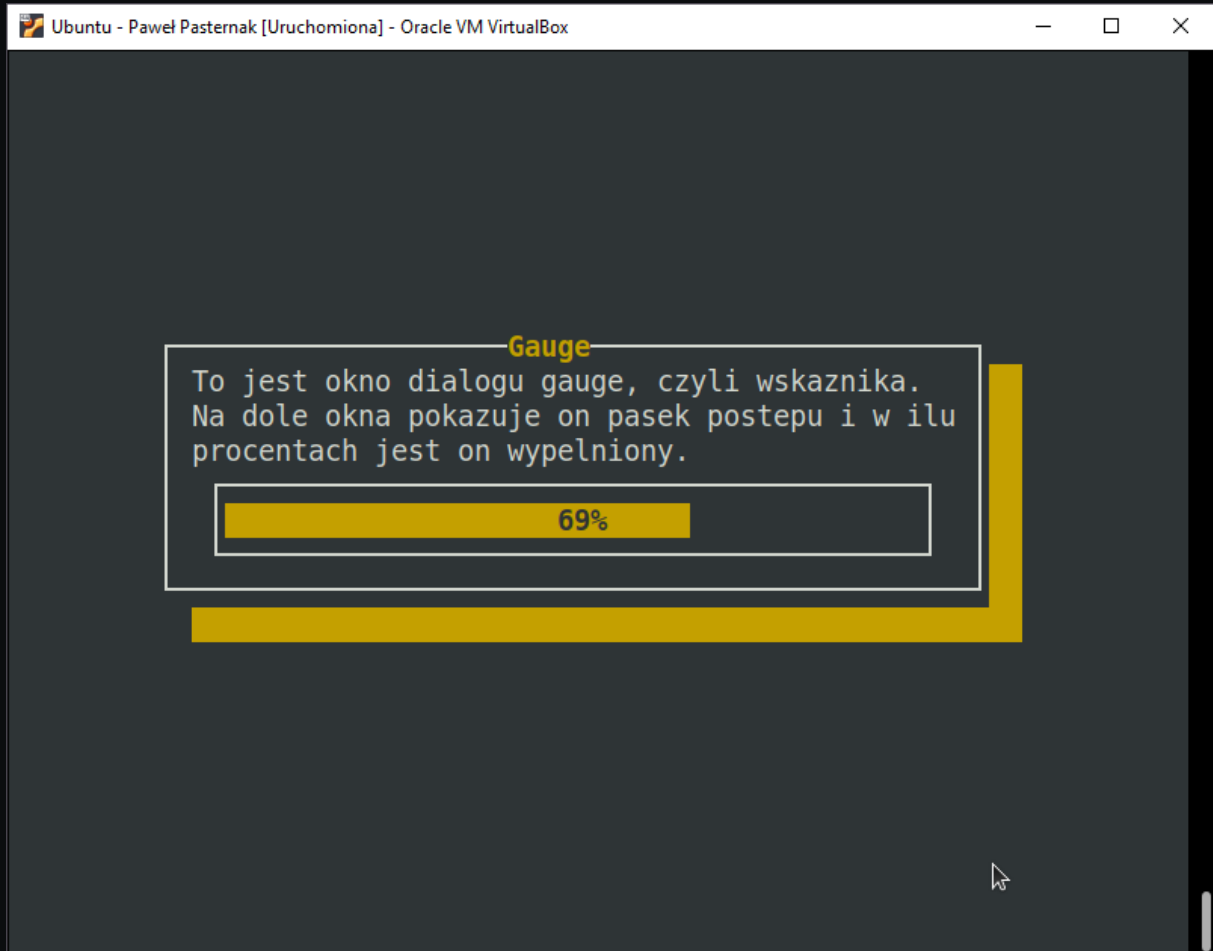
Kod:

```
#!/bin/bash
```

```
dialog --title "Gauge" \  
  --gauge "To jest okno dialogu gauge, czyli wskaźnika. \  
  Na dole okna pokazuje on pasek postępu i w ilu procentach jest on wypeł-  
niony." 8 50 70
```

```
# 3 cyfra odpowiada za procent wypełnienia
```


Przykładowy sposób działania:



Kod:

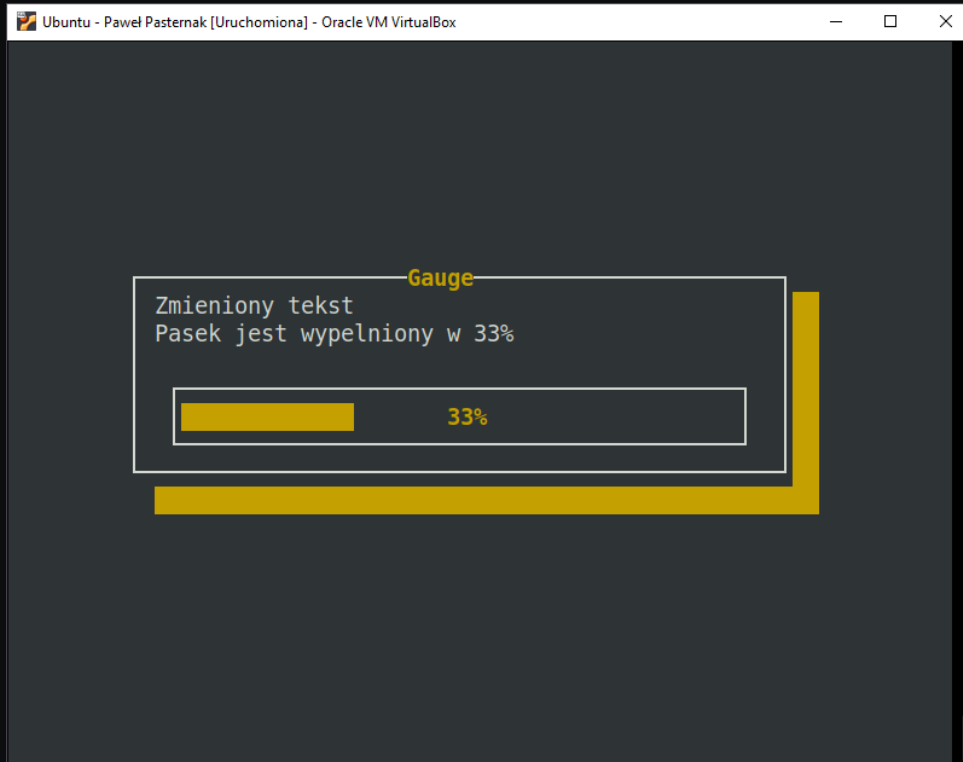
```
#!/bin/bash
```

```
#aby pasek postępu działał możemy utworzyć petle która co 0.1s będzie wysyłała w ilu procentach jest wypełniony
```

```
(i=0
while [ $i -ne 100 ]; do
    echo $i
    ((i+=1))
    sleep 0.1s
done) |
```

```
dialog --title "Gauge" \
    --gauge "To jest okno dialogu gauge, czyli wskaźnika. \
    Na dole okna pokazuje on pasek postępu i w ilu procentach jest on wypełniony." \
    8 50 0
```

Zmieniający się tekst wraz z paskiem:



Kod:

```
#!/bin/bash
```

```
#aby pasek postępu działał możemy utworzyć pętlę która co 0.1s będzie wysyłała w ilu procentach jest wypełniony
```

```
(i=0
while [ $i -ne 100 ]; do
    echo $i
```

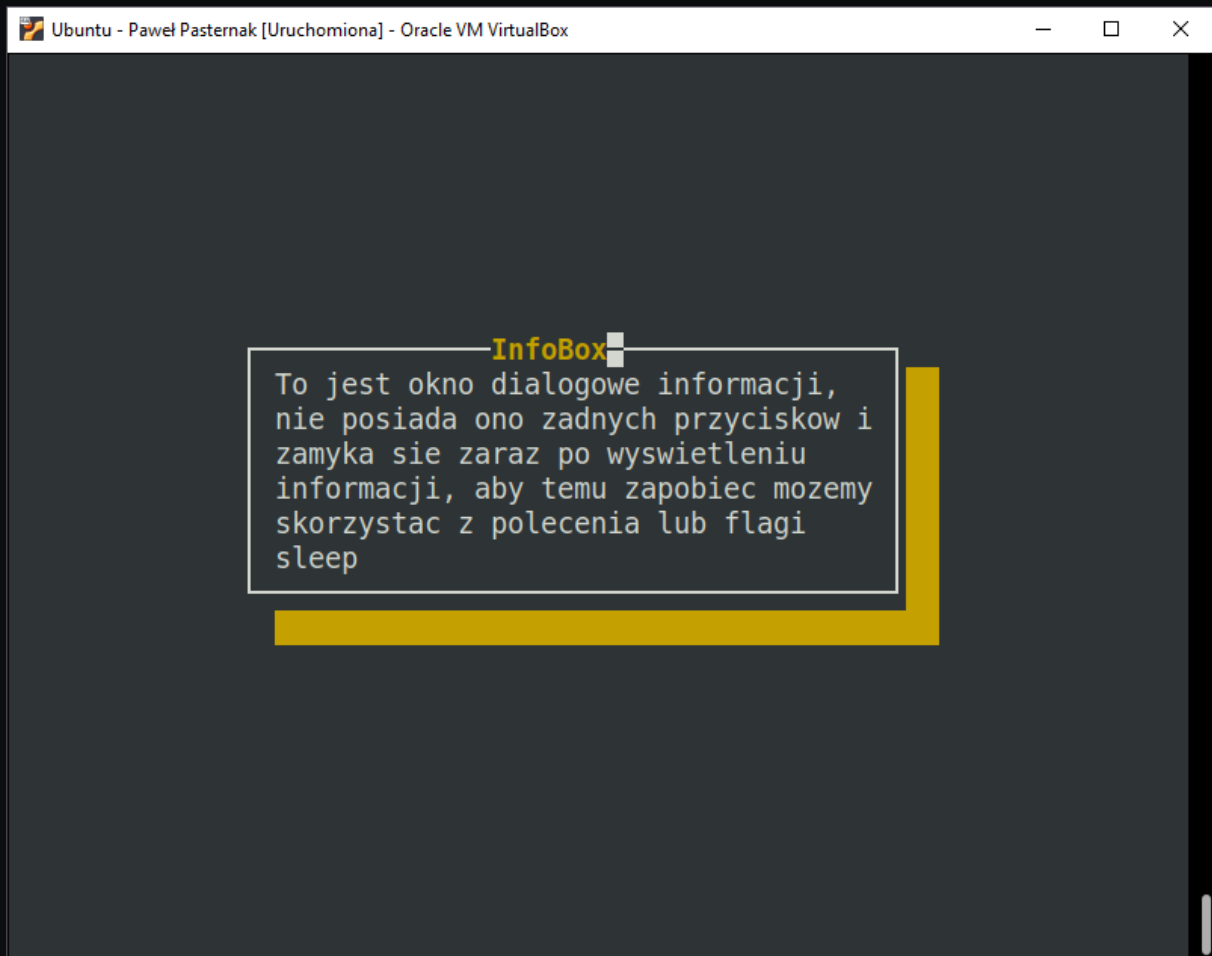
```
    #jeżeli chcielibyśmy zmienić także wyświetlany nad paskiem tekst to możemy to zrobić w następujący sposób
```

```
    echo "XXX"
    echo "Zmieniony tekst \nPasek jest wypełniony w $i%"
    echo "XXX"
```

```
    ((i+=1))
    sleep 0.1s
done) |
```

```
dialog --title "Gauge" \
    --gauge "To jest okno dialogu gauge, czyli wskaźnika. \
    Na dole okna pokazuje on pasek postępu i w ilu procentach jest on wypełniony." \
    8 50 0
```

8. InfoBox



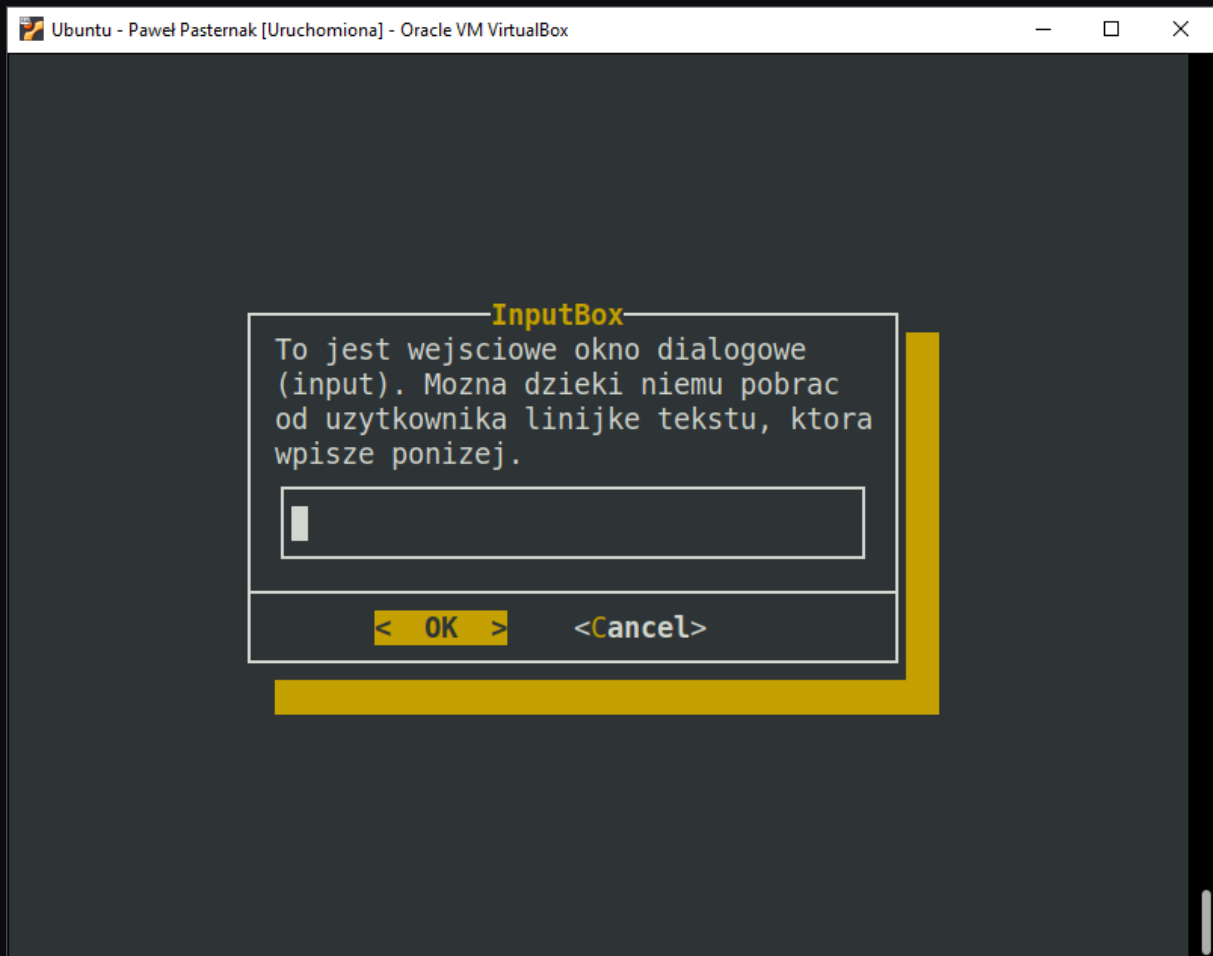
Kod:

```
#!/bin/bash
```

```
#flaga --sleep sprawia że program zatrzymuje się na 5 sekund
```

```
dialog --title "InfoBox" \  
  --sleep 5 \  
  --infobox "To jest okno dialogowe informacji, nie posiada ono \  
  żadnych przycisków i zamyka się zaraz po wyświetleniu informacji, \  
  aby temu zapobiec możemy skorzystać z polecenia lub flagi sleep" 8 40
```

9. InputBox



Kod:

```
#!/bin/bash
```

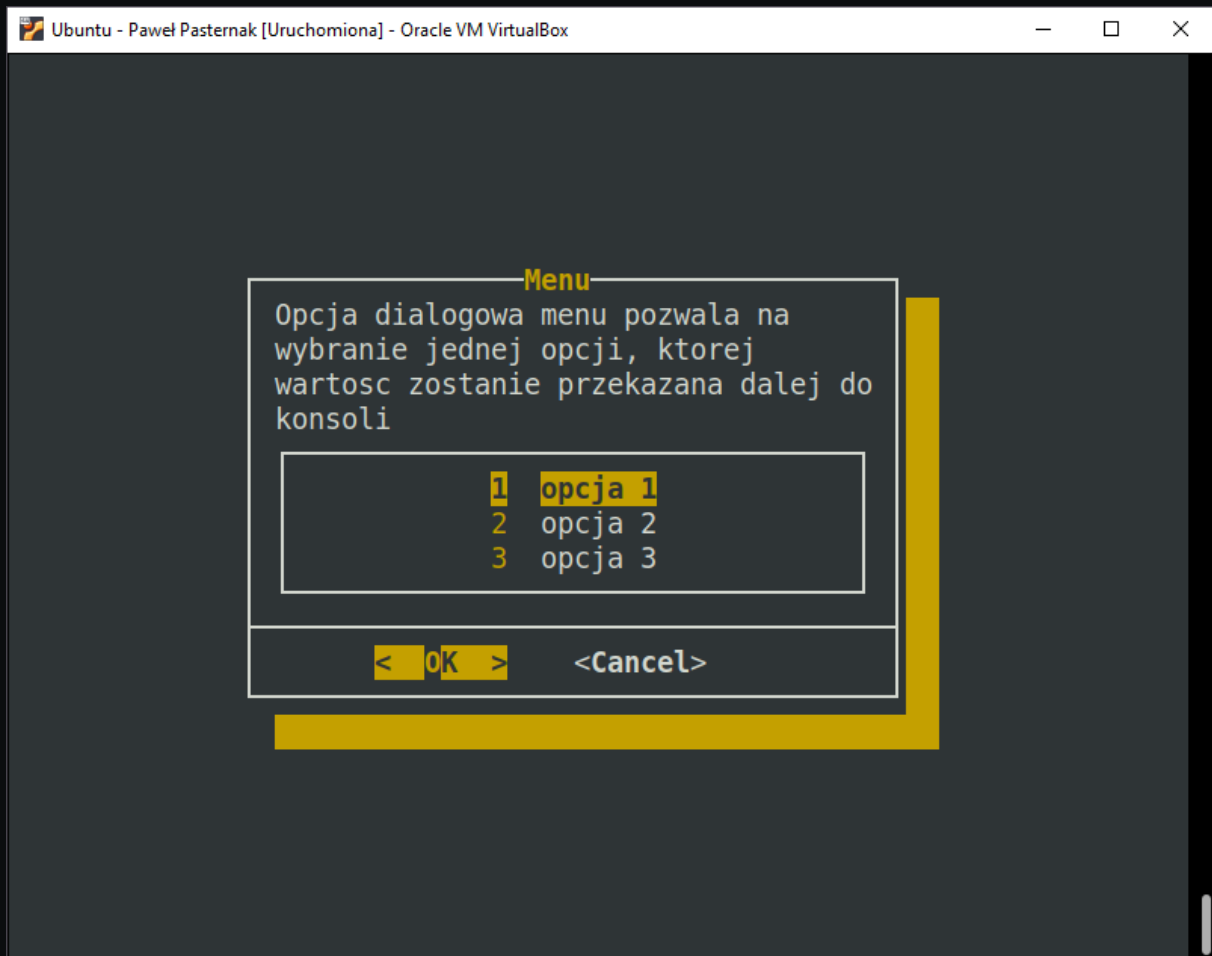
```
dialog --title "InputBox" \  
  --inputbox "To jest wejsciowe okno dialogowe (input). \  
  Moza dzięki niemu pobrac od uzytkownika linijke tekstu, \  
  ktora wpisze ponizej." 11 40
```

```
#to co wprowadzi uzytkownik zostanie przeslane do konsoli
```

10. InputMenu

TODO

11. Menu



Kod:

```
#!/bin/bash
```

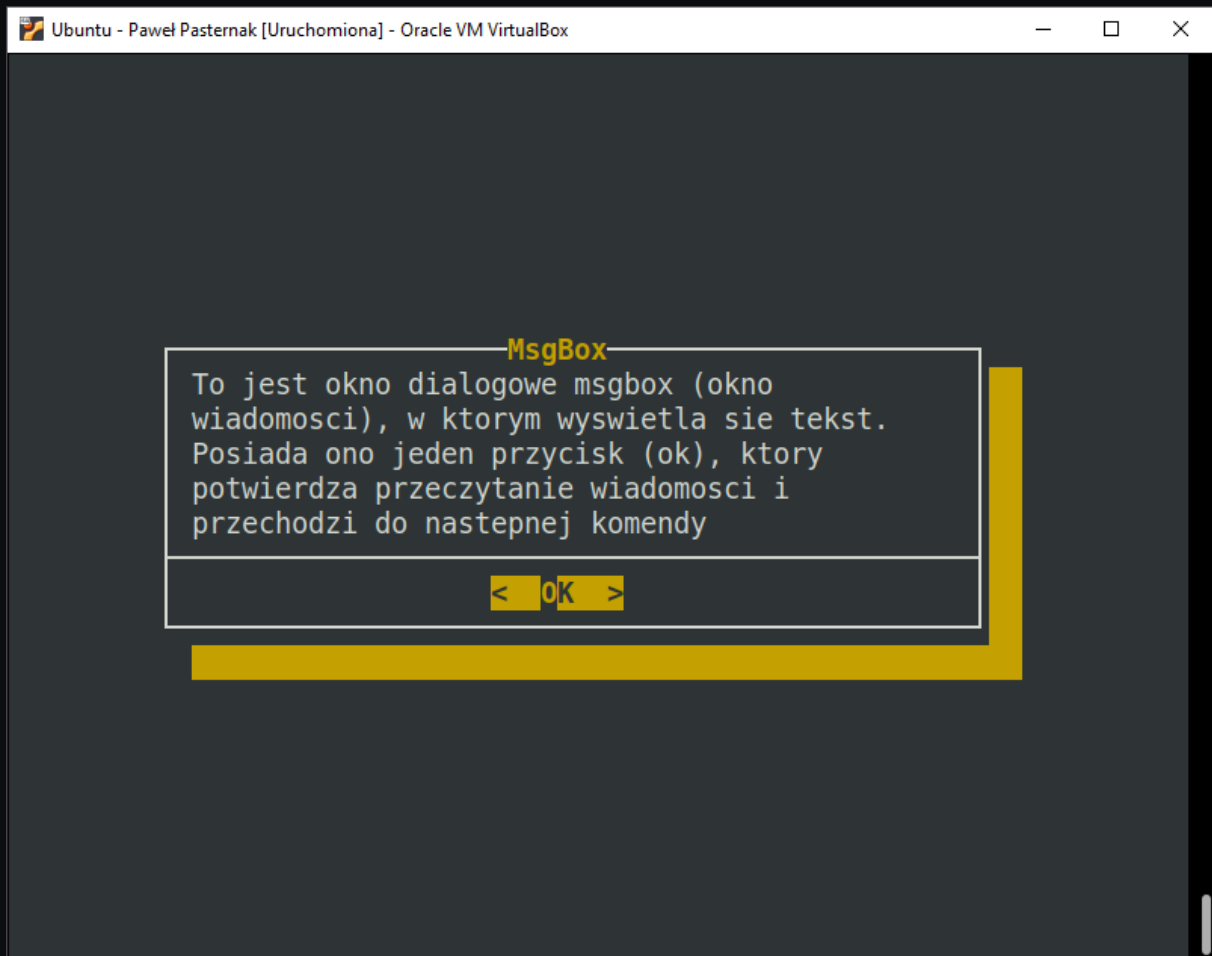
```
dialog --title "Menu" \
  --menu "Opcja dialogowa menu pozwala na wybranie jednej opcji, \
  ktorej wartosc zostanie przekazana dalej do konsoli" 0 0 0 \
  1 "opcja 1" \
  2 "opcja 2" \
  3 "opcja 3"
```

#tekst (tutaj cyfry) z prawej zostaną wysłane jako output do konsoli, natomiast tekst z prawej jest to "opis"

12. MixedGauge

*to do

13. MsgBox



Kod:

```
#!/bin/bash
```

```
dialog --title "MsgBox" \  
  --msgbox "To jest okno dialogowe msgbox (okno wiadomosci),\  
  w którym wyświetla się tekst. Posiada ono jeden przycisk (ok),\  
  który potwierdza przeczytanie wiadomosci \  
  i przechodzi do następnej komendy" 9 50\  

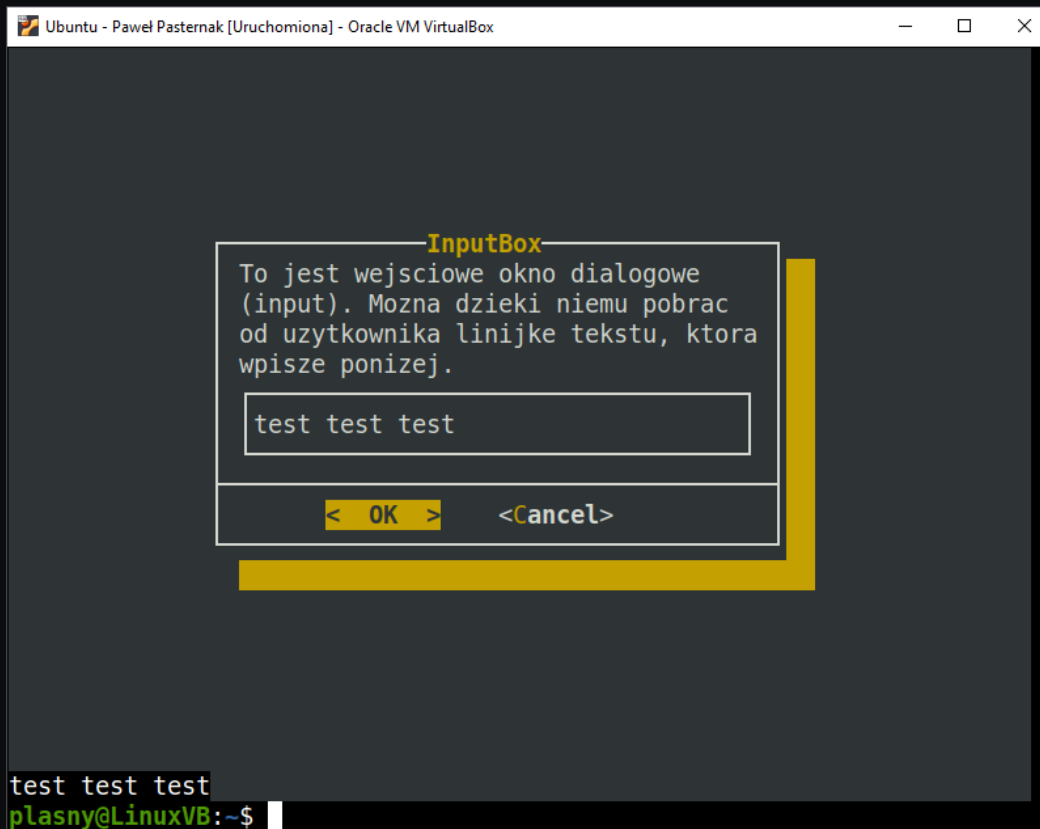
```

```
#ostatnie dwie cyfry odpowiadaja za rozmiar okna
```


Wyprowadzanie informacji z okienek:

Wyprowadzanie informacji podanych przez użytkownika w dialogu do innego polecenia lub pliku:

1. Przykład wyprowadzenia informacji z InputBoxa



```
#!/bin/bash
```

#Jeżeli chcemy wyprowadzić to co wpisał użytkownik do dialogu to możemy to zrobić na 2 znane nam sposoby:

#1 sposób, z użyciem flagi --stdout

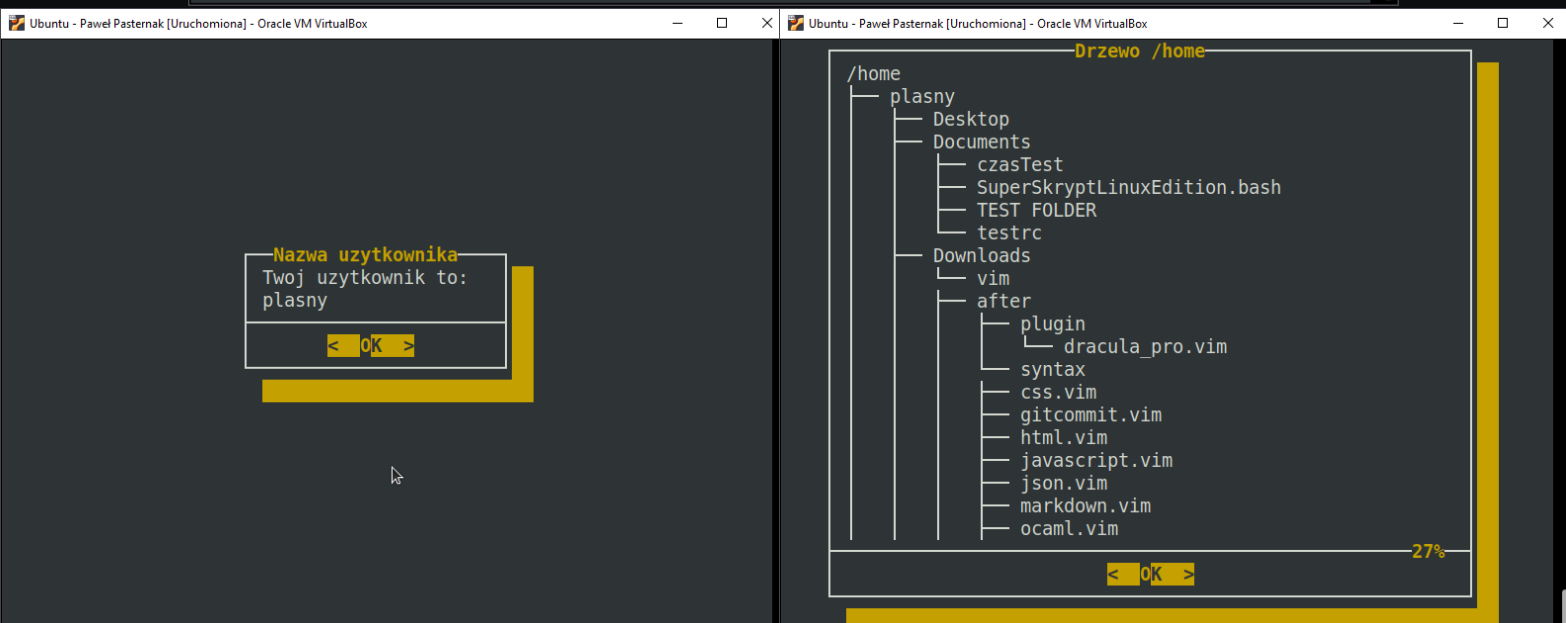
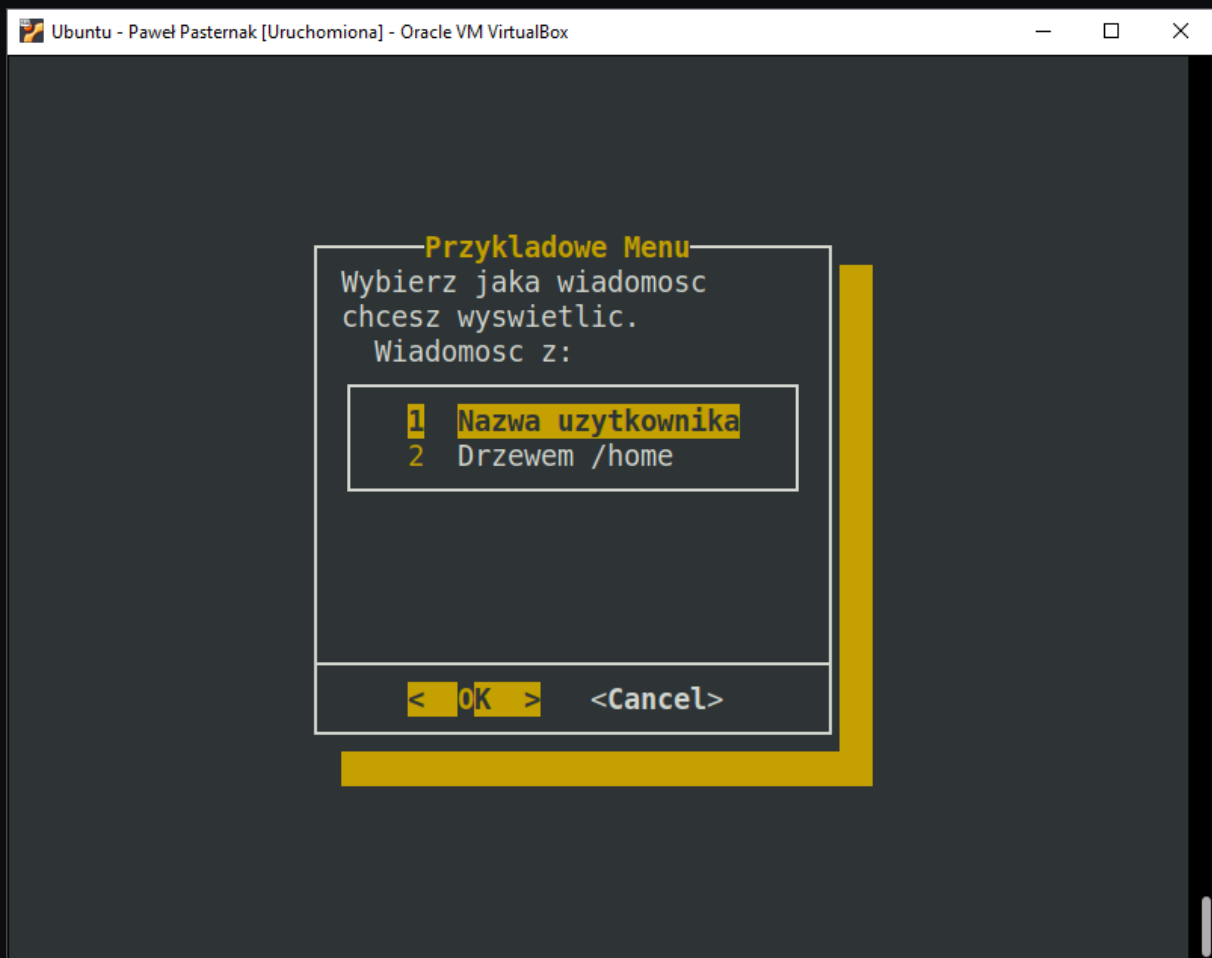
```
input=$(dialog --stdout \
  --title "InputDialog" \
  --inputbox "To jest wejsciowe okno dialogowe (input). \
    Można dzięki niemu pobrać od użytkownika linijkę tekstu, \
    która wpisze poniżej." 11 40)
echo $input
```

#2 sposób, z użyciem outputu do pliku po wciśnięciu ok

```
UserInputToFile=/tmp/UITF.sh.$$
```

```
dialog --title "InputDialog" \
  --inputbox "To jest wejsciowe okno dialogowe (input). \
    Można dzięki niemu pobrać od użytkownika linijkę tekstu, \
    która wpisze poniżej." 11 40 2>"${UserInputToFile}"
input=`cat $UserInputToFile`
echo $input
```

2. Przykładowe menu z oknem wiadomości



Kod:

```
#!/bin/bash
```

#Aby skrypt był funkcjonalny należy zainstalować aplikację "tree"

#1 sposób, z użyciem flagi --stdout

```
menu=$(dialog --stdout \
  --title "Przykładowe Menu" \
  --menu "Wybierz jaką wiadomość chcesz wyświetlić. \n
  Wiadomość z: " 0 0 0 \
    1 "Nazwa użytkownika" \
    2 "Drzewem /home")

case $menu in
  1) dialog --title "Nazwa użytkownika" \
    --msgbox "Twój użytkownik to: \n$USER" 6 25;;
  2) tree=$(tree /home) #zapis wyniku komendy 'tree /home' do zmiennej tree
    dialog --title "Drzewo /home" \
      --msgbox "$tree" 50 60;;
esac
```

#2 sposób z zapisem do plików

```
UserInputToFile=/tmp/UITF.sh.$$
```

```
CommandInputToFile=/tmp/CITF.sh.$$
```

```
dialog --title "Przykładowe Menu" \
  --menu "Wybierz jaką wiadomość chcesz wyświetlić. \n
  Wiadomość z: " 0 0 0 \
    1 "Nazwa użytkownika" \
    2 "Drzewem /home" 2>"${UserInputToFile}"
```

```
menu=`cat $UserInputToFile`
```

```
case $menu in
  1) dialog --title "Nazwa użytkownika" \
    --msgbox "Twój użytkownik to: \n$USER" 6 25;;
  2) (tree /home) > "${CommandInputToFile}"
    dialog --title "Drzewo /home" \
      --textbox "${CommandInputToFile}" 50 60;; #text-
box pozwala na wyświetlanie zawartości plików, podobnie do msgbox
esac
```

```
rm -f /tmp/UITF.sh.* && rm -f /tmp/CITF.sh.*
```