

Skrypt 1 - Menu

Skrypt ma na celu możliwość wyboru przy pomocy trybu okienkowego 4 pozostałych skryptów.

Dzięki takiemu zabiegowi użytkownik otrzymuje szybki i przejrzysty dostęp do uruchamiania skryptów.

```
#!/bin/bash

M="wybierz skrypt"

OPCJE=(1 "Kalkulator"
2 "Informacje o systemie"
3 "Wizytowka"
4 "Pole i Obwód figury" )

WYBOR=$(dialog --clear --title "Skrypty do wyboru:" --menu "$M" 20 70 4 "${OPCJE[@]}" \
2>&1 >/dev/tty)

clear
case $WYBOR in
1)
./kalk.sh ;;
2)
./info.sh ;;
3)
./wizyt.sh ;;
4)
./ob.sh ;;
esac
```

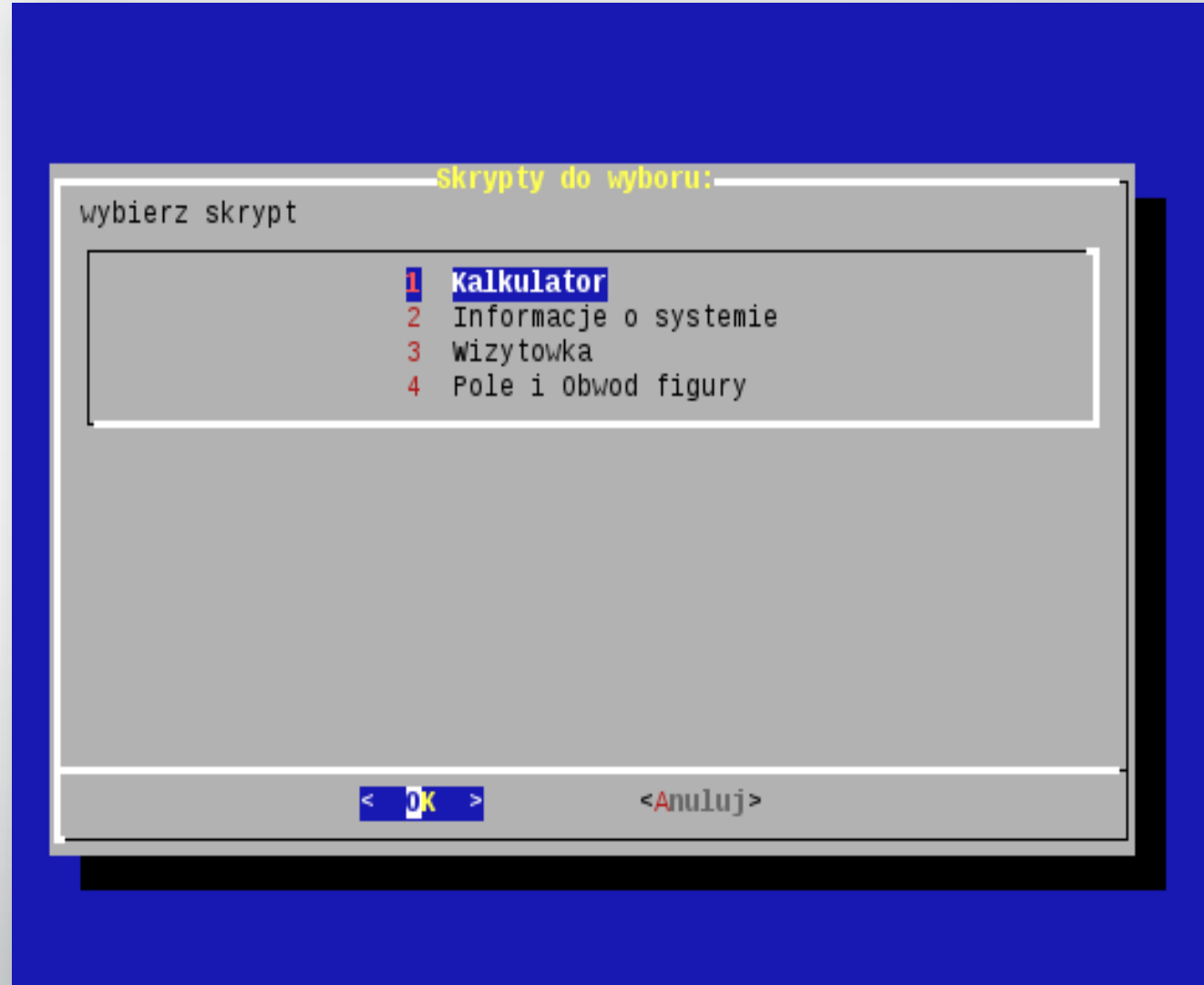
Skrypt 1 - Menu

Skrypt po uruchomieniu.

Widzimy iż możemy dokonać wyboru z 4 pozycji przesuwając się w górę lub w dół.

Wybór potwierdzamy kliknięciem w przycisk „OK”.

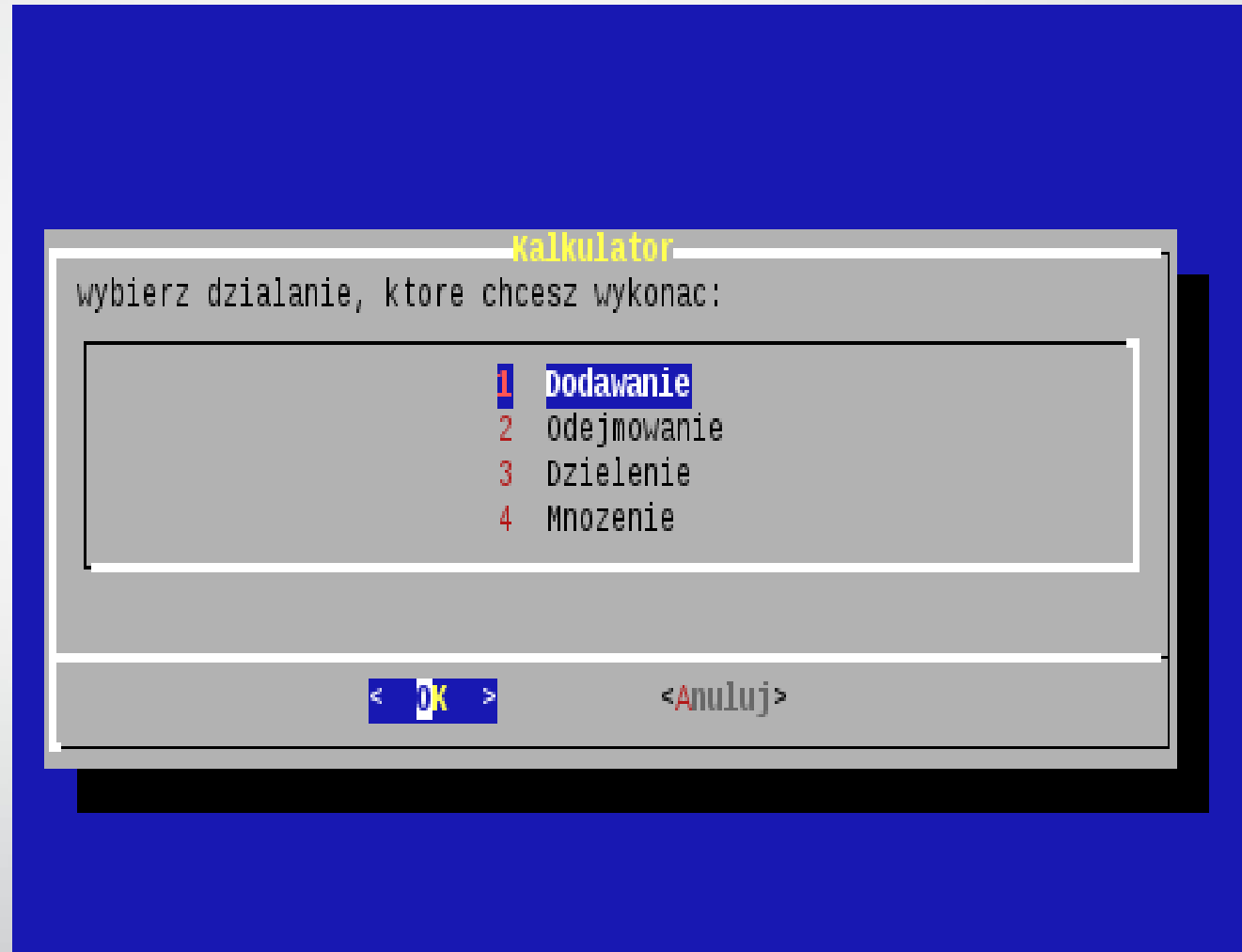
Lub wychodzimy ze skryptu przyciskiem „Anuluj”.



Skrypt 1 - Menu

Po wyborze interesującego nas skryptu w naszym przypadku jest to pozycja nr. 1 Kalkulator.

Skrypt przenosi nas bezpośrednio do następnego skryptu.



Skrypt 2 - Kalkulator

Skrypt ma na celu spełniać rolę prostego kalkulatora, który wyliczy nam działania takie jak dodawanie, odejmowanie, mnożenie i dzielenie.

Kalkulator liczby podawane przez użytkownika zapisuje w dwóch plikach tekstowych po czym zawartość ich przekształca na zmienne i za pomocą ich wylicza nam zadane działanie.

Cały skrypt działa w trybie okienkowym.

```
#!/bin/bash

M="wybierz dzialanie, ktore chcesz wykonac:"

OPCJE=(1 "Dodawanie"
2 "Odejmowanie"
3 "Dzielenie"
4 "Mnozenie" )

WYBOR=$(dialog --clear --title "Kalkulator" --menu "$M" 12 70 4 "${OPCJE[@]}" \
2>&1 >/dev/tty)

clear
case $WYBOR in

1)
dialog --inputbox "Podaj pierwsza liczbe:" 8 40 2> liczba.txt
dialog --inputbox "Podaj druga liczbe:" 8 40 2> liczba2.txt
zmiana="$(cat liczba.txt)"
zmiana2="$(cat liczba2.txt)"
echo $zmiana + $zmiana2 | bc > wynik.txt
dialog --clear --title "Wynik to:" --textbox "wynik.txt" 6 50 ;;

2)
dialog --inputbox "Podaj pierwsza liczbe:" 8 40 2> liczba.txt
dialog --inputbox "Podaj druga liczbe:" 8 40 2> liczba2.txt
zmiana="$(cat liczba.txt)"
zmiana2="$(cat liczba2.txt)"
echo $zmiana - $zmiana2 | bc > wynik.txt
dialog --clear --title "Wynik to:" --textbox "wynik.txt" 6 50 ;;

3)
dialog --inputbox "Podaj pierwsza liczbe:" 8 40 2> liczba.txt
dialog --inputbox "Podaj druga liczbe:" 8 40 2> liczba2.txt
zmiana="$(cat liczba.txt)"
zmiana2="$(cat liczba2.txt)"
```

Skrypt 2 - Kalkulator

Część druga skryptu.

```
3)
dialog --inputbox "Podaj pierwsza liczbe:" 8 40 2> liczba.txt
dialog --inputbox "Podaj druga liczbe:" 8 40 2> liczba2.txt
zmiana="$(cat liczba.txt)"
zmiana2="$(cat liczba2.txt)"
echo $zmiana / $zmiana2 | bc > wynik.txt
dialog --clear --title "wynik to:" --textbox "wynik.txt" 6 50

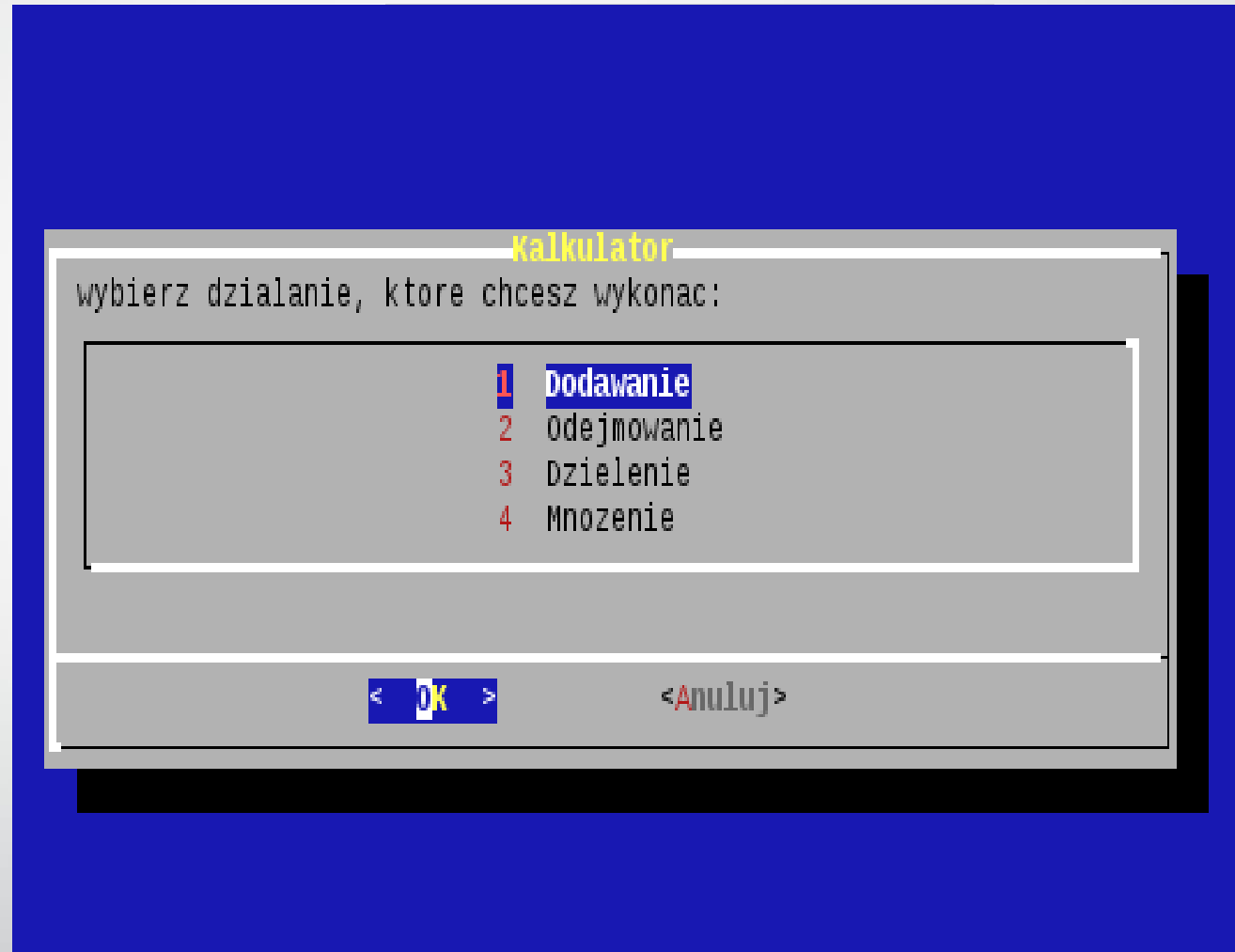
4)
dialog --inputbox "Podaj pierwsza liczbe:" 8 40 2> liczba.txt
dialog --inputbox "Podaj druga liczbe:" 8 40 2> liczba2.txt
zmiana="$(cat liczba.txt)"
zmiana2="$(cat liczba2.txt)"
echo $zmiana \* $zmiana2 | bc > wynik.txt
dialog --title "wynik to:" --textbox "wynik.txt" 6 50 ;;

esac
```

Skrypt 2 - Kalkulator

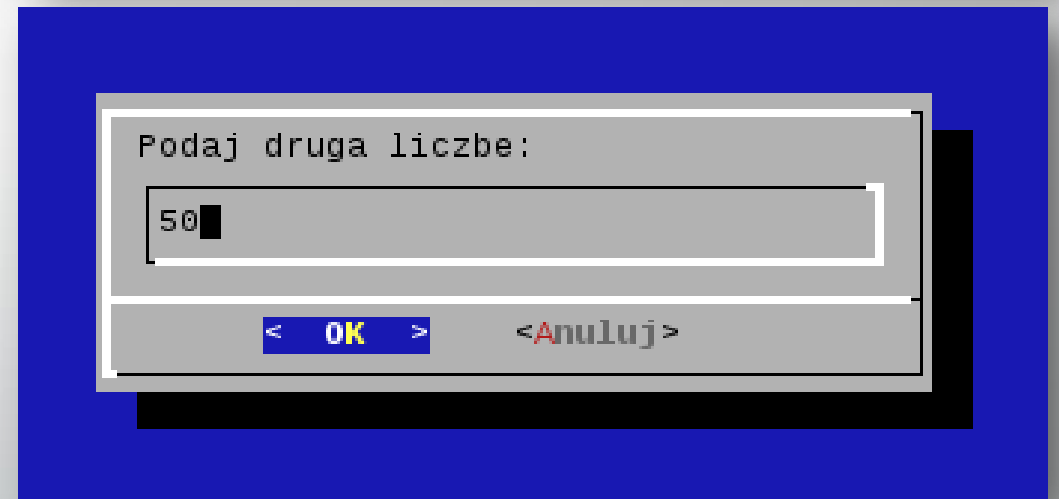
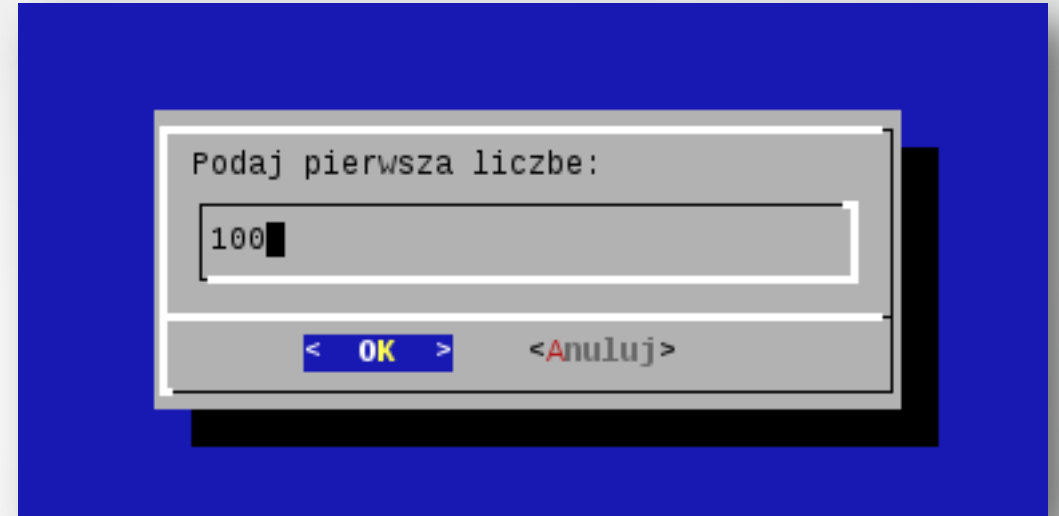
Skrypt po uruchomieniu pozwala nam na dokonanie wyboru jakie działanie chcemy wykonać. Do wyboru mamy 4 pozycje.

1. Dodawanie
2. Odejmowanie
3. Dzielenie
4. Mnożenie



Skrypt 2 - Kalkulator

Skrypt prosi nas o podanie pierwszej liczby oraz kolejno o podanie drugiej.



Skrypt 2 - Kalkulator

Po wcześniejszym podaniu liczb, które chcemy obliczyć za pomocą wybranego wcześniej działania (w naszym wypadku mnożenia) skrypt w kolejnym oknie podaje wynik działania.



Skrypt 3 – Informacje o systemie

Skrypt ma na celu dostarczać informacji o systemie oraz komputerze użytkownikowi. Możemy również dzięki niemu zajrzeć do kalendarza czy też sprawdzić aktualną datę i godzinę.

Skrypt po wyborze opcji informacji, którą chcemy otrzymać, za pomocą odpowiedniej komendy otrzymuje informacje, które zapisuje w pliku „info.txt” oraz kolejno zawartość pliku „info.txt” wyświetla informacje w okienku.

```
#!/bin/dash
```

```
M="wybierz informacje, która chcesz uzyskać:"
```

```
OPCJE=(1 "Nazwa jadra systemu"  
2 "Nazwa sieciowa komputera"  
3 "wersja jadra systemu"  
4 "Architektura komputera"  
5 "Typ procesora"  
6 "Informacje o platformie sprzętowej"  
7 "Informacje o nazwie systemu operacyjnego"  
8 "Informacje o wydaniu jadra systemu"  
9 "Data i Godzina"  
10 "Kalendarz" )
```

```
WYBOR=$(dialog --clear --title "Info:" --menu "$M" 17 70 10 "${OPCJE[@]}" \  
2>&1 >/dev/tty)
```

```
clear
```

```
case $WYBOR in
```

```
1)
```

```
uname -s > info.txt
```

```
dialog --clear --title "wynik to:" --textbox "info.txt" 6 50 ;;
```

```
2)
```

```
uname -n > info.txt
```

```
dialog --clear --title "wynik to:" --textbox "info.txt" 6 50 ;;
```

```
3)
```

```
uname -v > info.txt
```

```
dialog --clear --title "wynik to:" --textbox "info.txt" 6 50 ;;
```

```
4)
```

```
uname -m > info.txt
```

Skrypt 3 – Informacje o systemie

Część druga skryptu.

```
4)
uname -m > info.txt
dialog --clear --title "wynik to:" --textbox "info.txt" 6 50 ;;

5)
uname -p > info.txt
dialog --clear --title "wynik to:" --textbox "info.txt" 6 50 ;;

6)
uname -i > info.txt
dialog --clear --title "wynik to:" --textbox "info.txt" 6 50 ;;

7)
uname -o > info.txt
dialog --clear --title "wynik to:" --textbox "info.txt" 6 50 ;;

8)
uname -r > info.txt
dialog --clear --title "wynik to:" --textbox "info.txt" 6 50 ;;

9)
date > info.txt
dialog --title "wynik to:" --textbox "info.txt" 0 0 ;;

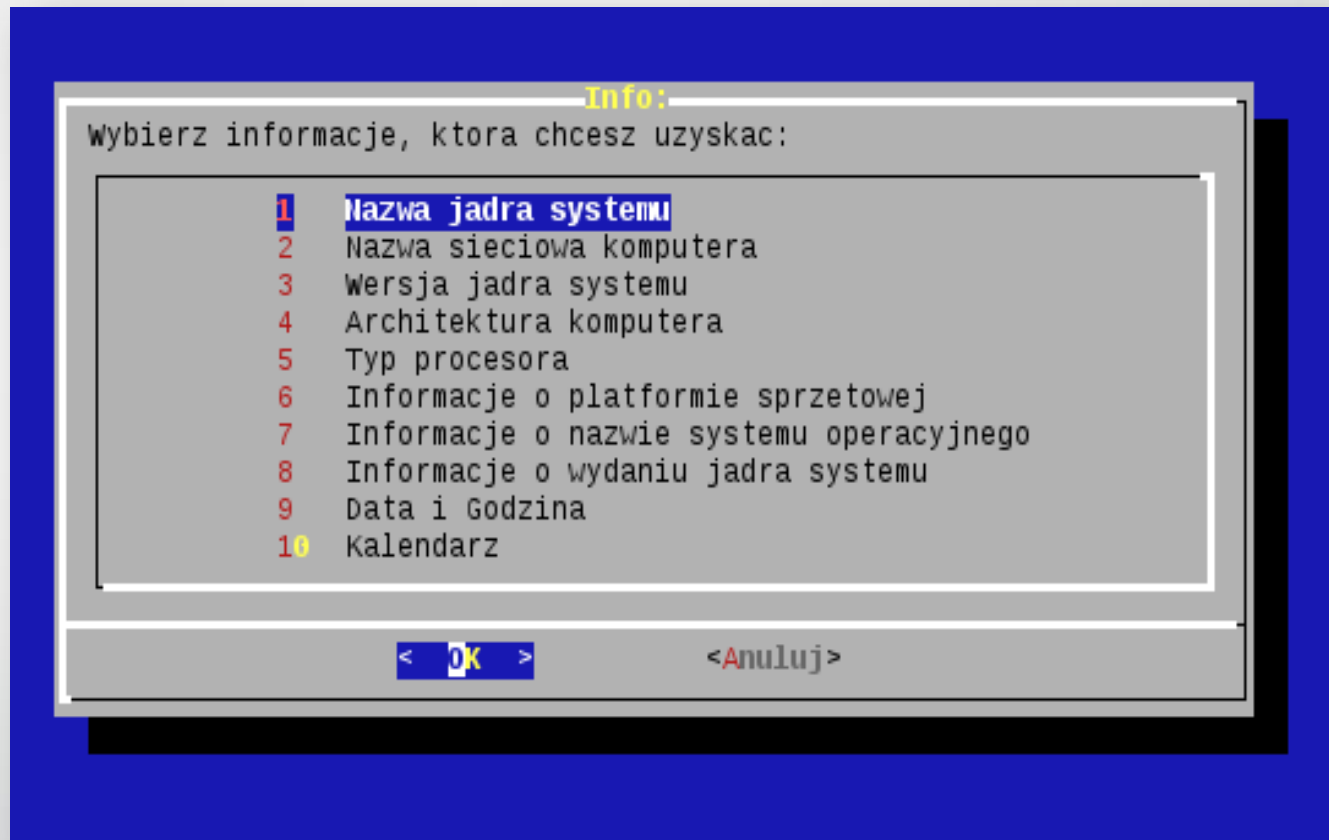
10)
cal > cal.txt
dialog --title "wynik to:" --textbox "cal.txt" 0 0 ;;

esac
```

Skrypt 3 – Informacje o systemie

Skrypt po uruchomieniu pozwala nam na dokonanie wyboru informacji, którą chcemy uzyskać. Do wyboru mam 10 opcji.

1. Nazwa jądra systemu
2. Nazwa sieciowa komputera
3. Wersja jądra systemu
4. Architektura komputera
5. Typ procesora
6. Informacje o platformie sprzętowej
7. Informacje o nazwie systemu operacyjnego
8. Informacje o wydaniu jądra systemu
9. Data i godzina
10. Kalendarz



Skrypt 3 – Informacje o systemie

Po dokonaniu wyboru interesującej nas opcji. Skrypt wyświetla nowe okno z wybraną przez nas opcją.

```
Wynik to:  
nie, 11 gru 2016, 14:26:03 CET
```

<WYJDZ>

```
Wynik to:  
3.16.6-2-default
```

<WYJDZ>

```
Wynik to:  
Linux
```

<WYJDZ>

```
Wynik to:  
grudzień 2016  
po wt śr cz pi so ni  
          1 2 3 4  
5 6 7 8 9 10 11  
12 13 14 15 16 17 18  
19 20 21 22 23 24 25  
26 27 28 29 30 31
```

<WYJDZ>

```
Wynik to:  
GNU/Linux
```

<WYJDZ>

```
Wynik to:  
i386
```

<WYJDZ>

```
Wynik to:  
i686
```

<WYJDZ>

```
Wynik to:  
#1 SMP Mon Oct 20 13:47:22 UTC 2014 (feb42ea)
```

<WYJDZ>

```
Wynik to:  
linux-cqv1.site
```

<WYJDZ>

Skrypt 4 – Generator wizytówki

Skrypt generuje wizytówkę.

Użytkownik na początku jest informowany o tym że znajduje się w generatorze wizytówki i a przejść dalej należy nacisnąć dalej

W następnym okienku użytkownika otrzymuje informację o tym że zostanie zadane mu kilka pytań i czy chce dalej kontynuować.

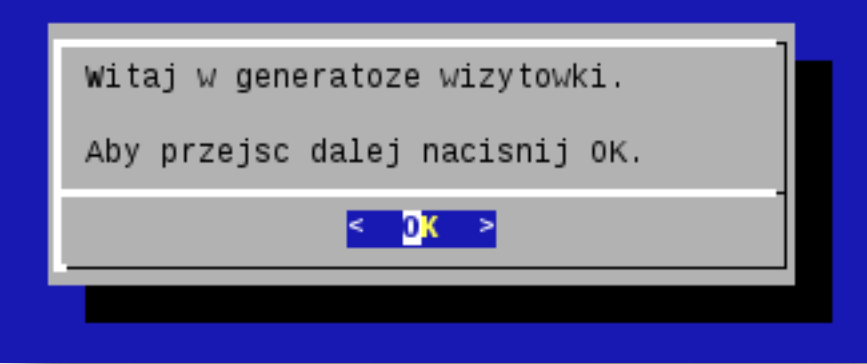
Skrypt zadaje pytania po czym wszystkie zebrane informacje zapisuje w pliku „wizytowka.txt” a następnie w oknie textbox zostaje wyświetlona zawartość pliku „wizytowka.txt”

```
#!/bin/bash
dialog --msgbox "Witaj w generatoze wizytowki. \n\nAby przejsc dalej nacisnij OK." 7 40
dialog --yesno "Generator zada Ci kilka pytan, czy chcesz kontynuowac?" 5 70 \
echo '#####' > wizytowka.txt
dialog --inputbox "Podaj swoje Imie i Nazwisko:" 10 40 2>> wizytowka.txt
echo >> wizytowka.txt
dialog --inputbox "Podaj swój adres E-Mail:" 10 40 2>> wizytowka.txt
echo >> wizytowka.txt
dialog --inputbox "Podaj Podaj swój adres zamieszkania:" 10 40 2>> wizytowka.txt
echo >> wizytowka.txt
dialog --inputbox "Podaj swój nr. telefonu:" 10 40 2>> wizytowka.txt
echo >> wizytowka.txt
echo '#####' >> wizytowka.txt
dialog --textbox "wizytowka.txt" 10 50
```

Skrypt 4 – Generator wizytówki

W oknie zostaje informacja o tym że użytkownik znajduje się w skrypcie generator wizytówki.

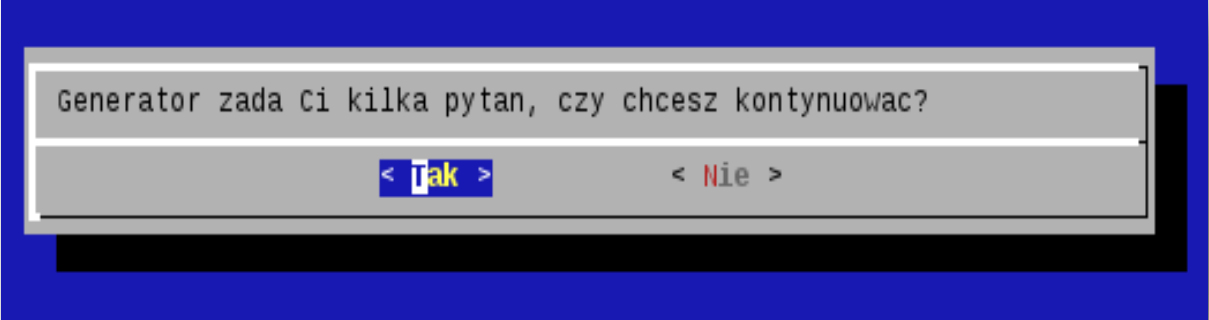
Oraz następnie otrzymujemy informację o tym że zostanie zadane nam kilka pytań i czy chcemy kontynuować dalej.



```
Witaj w generatoze wizytowki.
```

```
Aby przejsc dalej naciśnij OK.
```

```
< OK >
```



```
Generator zada ci kilka pytan, czy chcesz kontynuowac?
```

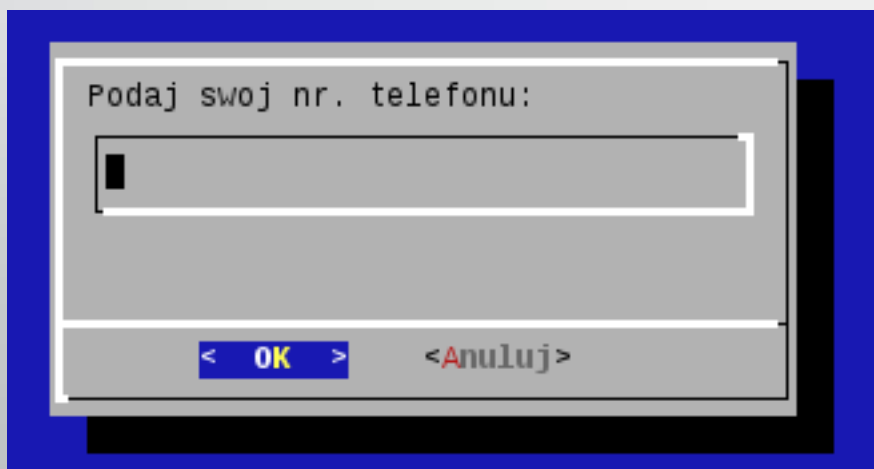
```
< Tak >
```

```
< Nie >
```

Skrypt 4 – Generator wizytówki

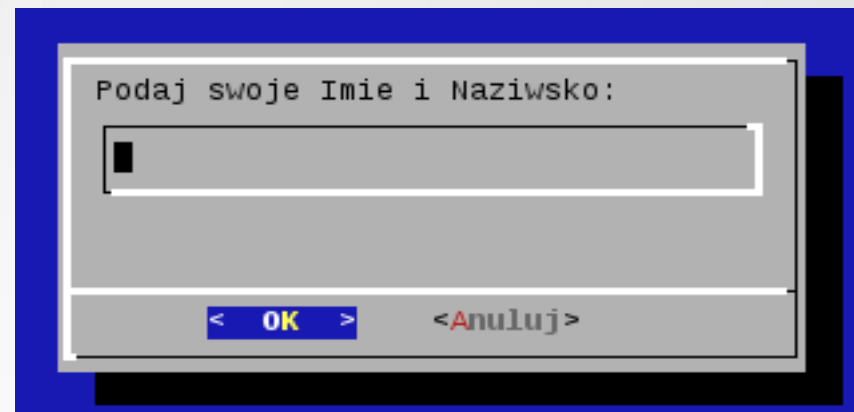
Skrypt zadaje pytanie a następnie po wpisaniu przez użytkownika wartości i kliknięciu przechodzi do następnego pytania. Skrypt zawiera 4 pytania takie jak:

1. Podaj swoje Imię i Nazwisko
2. Podaj swój adres E-Mail
3. Podaj swój adres zamieszkania
4. Podaj swój nr. telefonu



Podaj swój nr. telefonu:

< OK > <Anuluj>



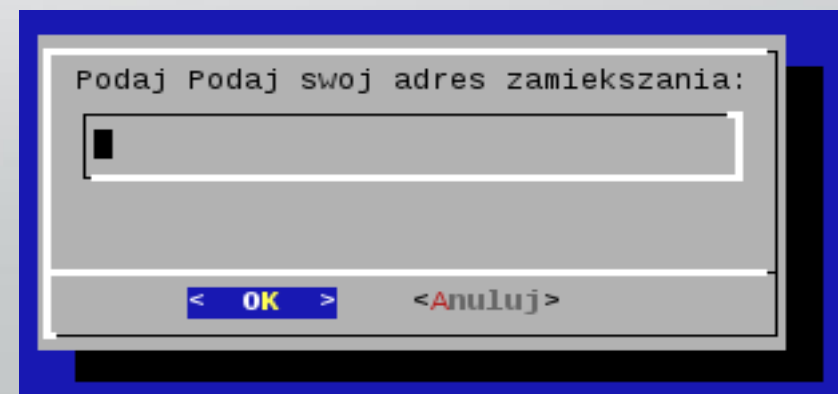
Podaj swoje Imię i Nazwisko:

< OK > <Anuluj>



Podaj swój adres E-Mail:

< OK > <Anuluj>



Podaj Podaj swój adres zamieszkania:

< OK > <Anuluj>

Skrypt 5 – Obliczanie pola i obwodu figury

Skrypt ma za zadanie obliczanie pola i obwodu wybranej przez nas figury geometrycznej. Skrypt pyta o wymiary boków, wysokość i promień i następnie zapisuje je w plikach tekstowych.

Wartości z plików tekstowych są przekształcane na zmienne.

Następnie po zastosowaniu odpowiednich wzorów wyliczany jest obwód i pole.

Na końcu wynik zapisywany jest w pliku „wynik.txt”

```
#!/bin/bash

M="wybierz figure, ktorej chcesz obliczyc obwod i pole:"

OPCJE=(1 "Kwadrat"
2 "Prostokat"
3 "Trojkat"
4 "Kolo" )

WYBOR=$(dialog --clear --title "Obwod i Pole" --menu "$M" 12 70 4 "${OPCJE[@]}" \
2>&1 >/dev/tty)

clear
case $WYBOR in

1)
dialog --inputbox "Podaj bok kwadratu:" 8 40 2> liczba.txt
zmiana="$(cat liczba.txt)"
echo 'Obwod:' > wynik.txt
echo $zmiana \* 4 | bc >> wynik.txt
echo 'Pole:' >> wynik.txt
echo $zmiana \* $zmiana | bc >> wynik.txt
dialog --clear --title "Wynik:" --textbox "wynik.txt" 10 50 ;;

2)
dialog --inputbox "Podaj pierwszy bok prostokatu:" 8 40 2> liczba.txt
dialog --inputbox "Podaj drugi bok prostokatu:" 8 40 2> liczba2.txt
zmiana="$(cat liczba.txt)"
zmiana2="$(cat liczba2.txt)"
echo 'Obwod:' > wynik.txt
echo $zmiana \* 2 + $zmiana2 \* 2 | bc >> wynik.txt
echo 'Pole:' >> wynik.txt
echo $zmiana \* $zmiana2 | bc >> wynik.txt
dialog --clear --title "Wynik:" --textbox "wynik.txt" 10 50 ;;
```


Skrypt 5 – Obliczanie pola i obwodu figury

Część druga skryptu.

```
2)
dialog --inputbox "Podaj pierwszy bok prostokatu:" 8 40 2> liczba.txt
dialog --inputbox "Podaj drugi bok prostokatu:" 8 40 2> liczba2.txt
zmiana="$(cat liczba.txt)"
zmiana2="$(cat liczba2.txt)"
echo 'Obwod:' > wynik.txt
echo $zmiana \ $\times$  2 + $zmiana2 \ $\times$  2 | bc >> wynik.txt
echo 'Pole:' >> wynik.txt
echo $zmiana \ $\times$  $zmiana2 | bc >> wynik.txt
dialog --clear --title "wynik:" --textbox "wynik.txt" 10 50 ;;

3)
dialog --inputbox "Podaj podstawie trojkata:" 8 40 2> liczba.txt
dialog --inputbox "Podaj drugi bok:" 8 40 2> liczba2.txt
dialog --inputbox "Podaj trzeci bok:" 8 40 2> liczba3.txt
dialog --inputbox "Podaj wysokosc:" 8 40 2> liczba4.txt
podstawa="$(cat liczba.txt)"
wysokosc="$(cat liczba4.txt)"
zmiana="$(cat liczba2.txt)"
zmiana2="$(cat liczba3.txt)"
echo 'Obwod:' > wynik.txt
echo $podstawa + $zmiana + $zmiana2 | bc >> wynik.txt
echo 'Pole:' >> wynik.txt
echo 0.5 \ $\times$  $podstawa \ $\times$  $wysokosc | bc >> wynik.txt
dialog --clear --title "wynik:" --textbox "wynik.txt" 10 50 ;;

4)
dialog --inputbox "Podaj promien kola:" 8 40 2> liczba.txt
promien="$(cat liczba.txt)"
echo 'Obwod:' > wynik.txt
echo 2 \ $\times$  3.14 \ $\times$  $promien | bc >> wynik.txt
echo 'Pole:' >> wynik.txt
echo $promien \ $\times$  $promien \ $\times$  3.14 | bc >> wynik.txt
dialog --clear --title "wynik:" --textbox "wynik.txt" 10 50 ;;
```

Skrypt 5 – Obliczanie pola i obwodu figury

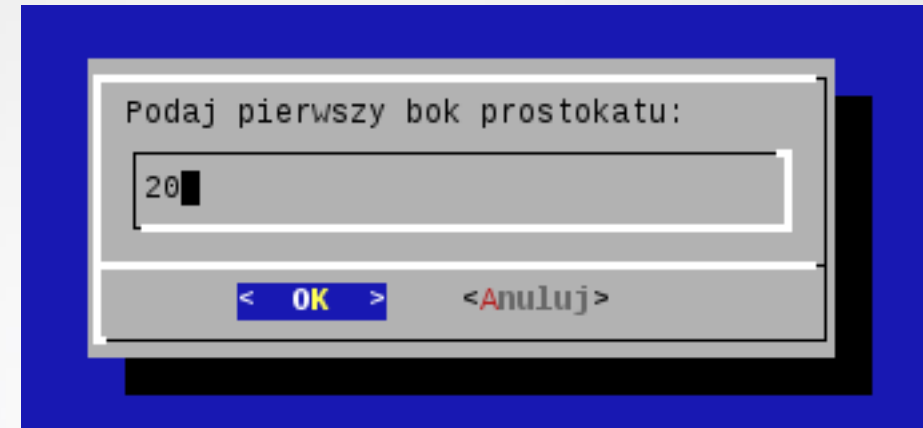
Użytkownik wybiera dla jakiej figury chce obliczyć pole oraz obwód.



Skrypt 5 – Obliczanie pola i obwodu figury

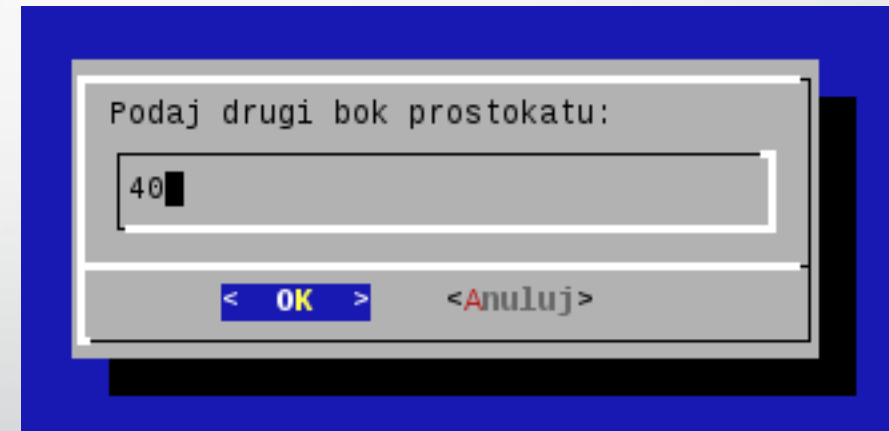
Następnie użytkownik zostaje zapytany o podanie pierwszego boku figury. W naszym wypadku prostokąta.

A następnie ponownie zostaje zapytany ale tym razem o drugi bok.



Podaj pierwszy bok prostokatu:

< OK > <Anuluj>



Podaj drugi bok prostokatu:

< OK > <Anuluj>

Skrypt 5 – Obliczanie pola i obwodu figury

Dzięki podanym wcześniej wartościom program wyliczył obwód oraz pole prostokąta.

Wyniki wyświetlane są w nowym oknie, które pobiera dane z pliku „wynik.txt”

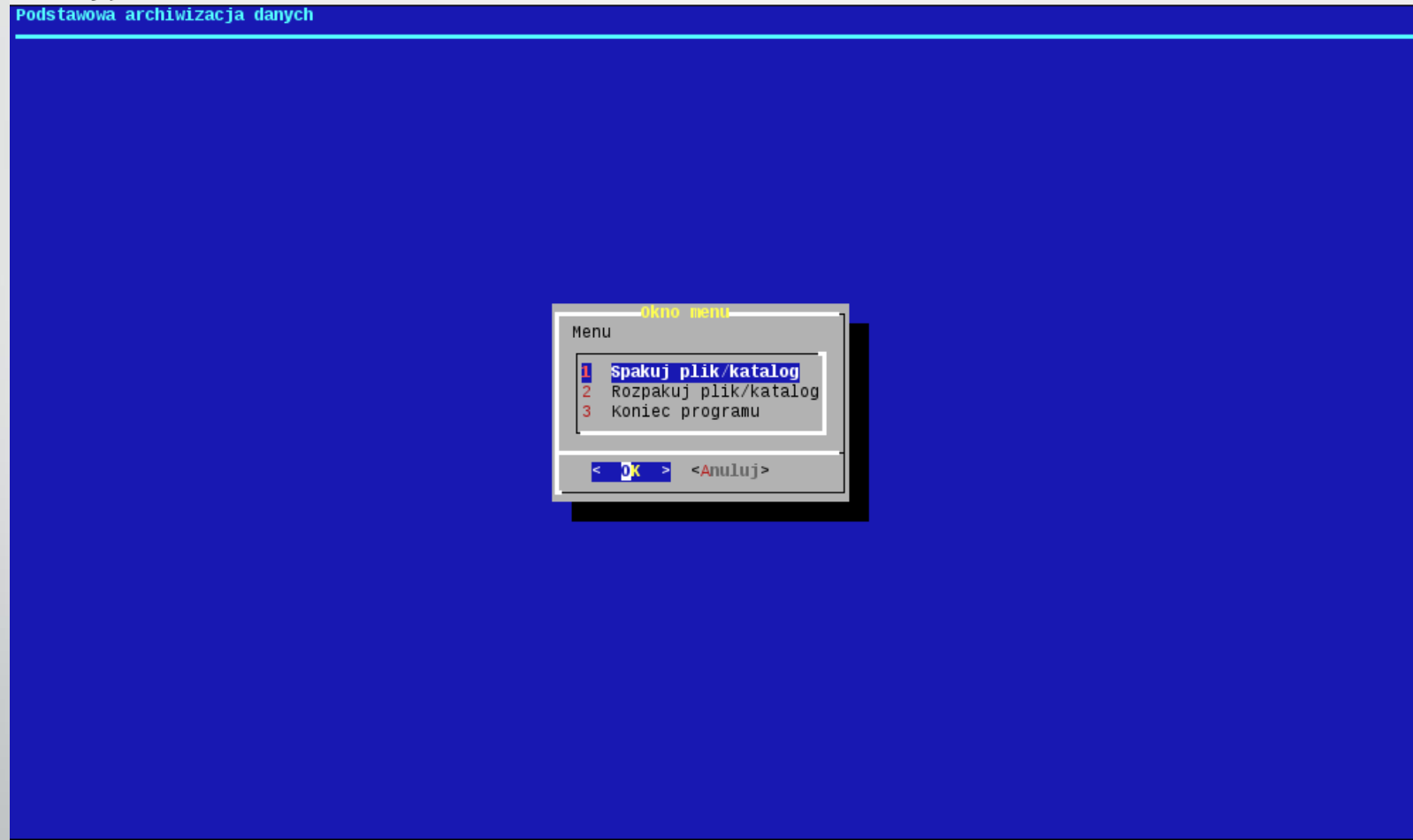


Skrypt 1

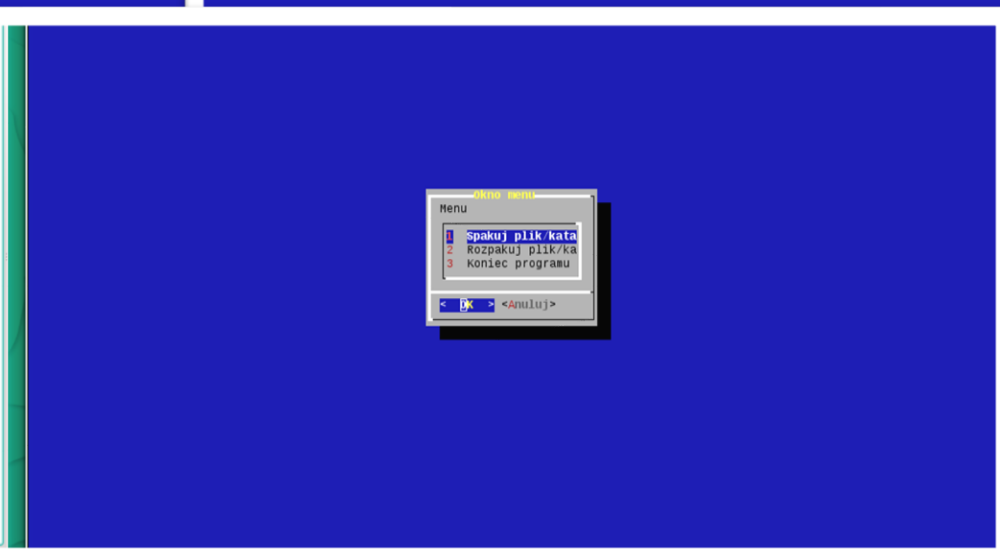
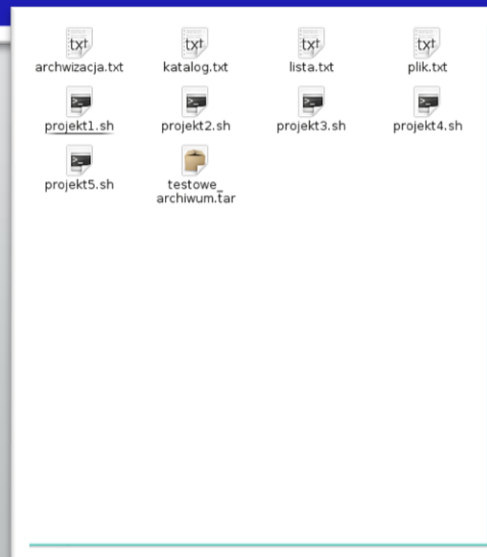
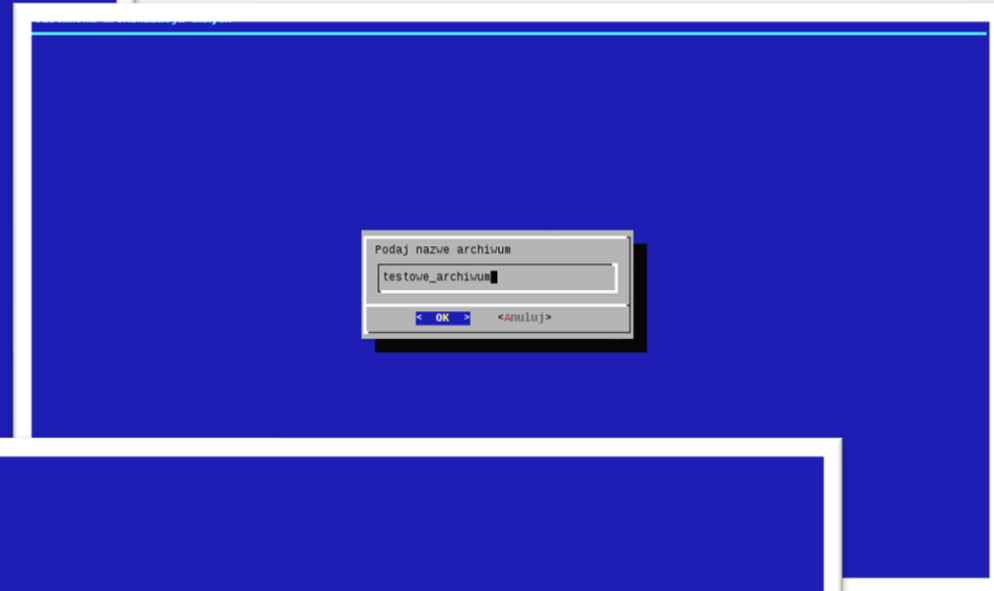
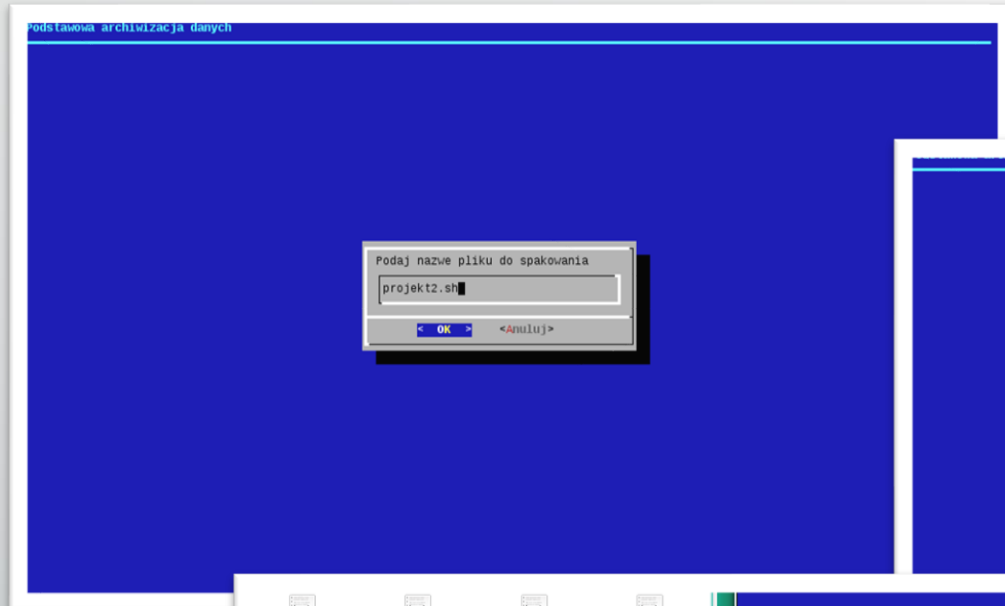
Skrypt pozwala na archiwizację danych oraz ich rozpakowanie. Skrypt jest wersji okienkowej. Najważniejsze polecenia wykorzystane w skrypcie to `tar -cf` oraz `tar -xvf`. Skrypt wykorzystuje także pętle `while` oraz instrukcje wyboru `case`.

```
#!/bin/bash
dialog --clear --title "Okno menu" \
--backtitle "Podstawowa archiwizacja danych" \
--menu "Menu" 10 30 15 \
1 "Spakuj plik/katalog" \
2 "Rozpakuj plik/katalog" \
3 "Koniec programu" 2>archwizacja.txt
opcja=$(cat archwizacja.txt)
while test $opcja != 3
do
case $opcja in
"1") echo -n > archwizacja.txt
dialog --backtitle "Podstawowa archiwizacja danych" --inputbox "Podaj nazwe pliku do spakowania" 8 40 2>archwizacja.txt
nazwa_pliku=$(cat archwizacja.txt)
echo -n > archwizacja.txt
dialog --backtitle "Podstawowa archiwizacja danych" --inputbox "Podaj nazwe archiwum" 8 40 2>archwizacja.txt
nazwa_arch=$(cat archwizacja.txt)
tar -cf $nazwa_arch.tar $nazwa_pliku ;;
"2") echo -n > archwizacja.txt
dialog --backtitle "Podstawowa archiwizacja danych" --inputbox "Podaj nazwe pliku do rozpakowania" 8 40 2>archwizacja.txt
nazwa_arch=$(cat archwizacja.txt)
tar -xvf $nazwa_arch.tar ;;
esac
dialog --clear --title "Okno menu" \
--backtitle "Podstawowa archiwizacja danych" \
--menu "Menu" 10 25 15 \
1 "Spakuj plik/katalog" \
2 "Rozpakuj plik/katalog" \
3 "Koniec programu" 2>archwizacja.txt
opcja=$(cat archwizacja.txt)
done
```

Skrypt 1



Skrypt 1



Skrypt 2

Skrypt pozwala na podstawowe operacje na plikach: tworzenie, edycję oraz usuwanie.

Wykorzystane polecenia to:

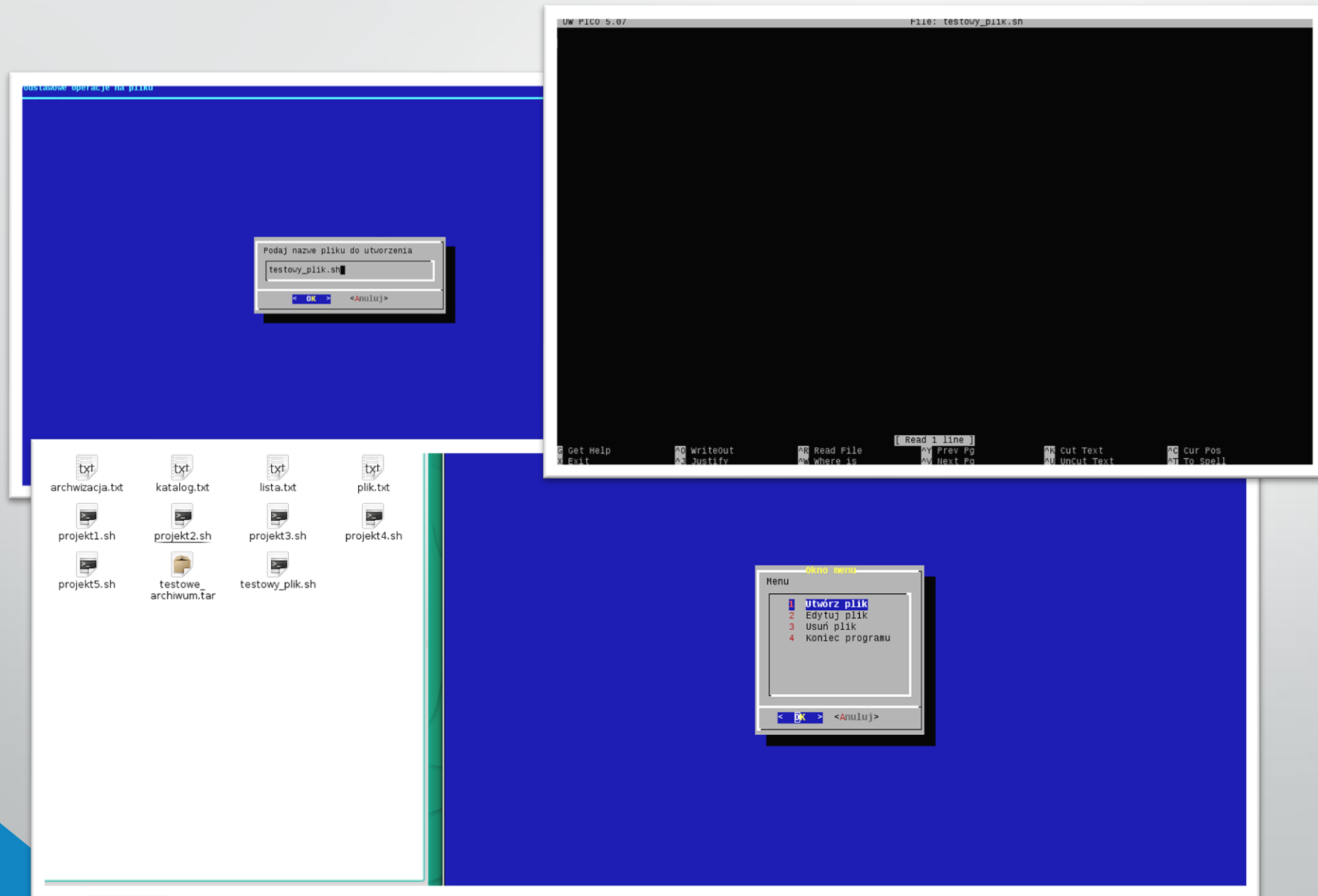
Touch

Pico

Rm

```
#!/bin/bash
dialog --clear --title "Okno menu" \
--backtitle "Podstawowe operacje na pliku" \
--menu "Menu" 15 30 15 \
1 "Utwórz plik" \
2 "Edytuj plik" \
3 "Usuń plik" \
4 "Koniec programu" 2>plik.txt
opcja=$(cat plik.txt)
while test $opcja != 4
do
case $opcja in
"1") echo -n > plik.txt
dialog --backtitle "Podstawowe operacje na pliku" --inputbox "Podaj nazwe pliku do utworzenia" 8 40 2>plik.txt
nazwa_pliku=$(cat plik.txt)
touch $nazwa_pliku ;;
"2") echo -n > archwizacja.txt
dialog --backtitle "Podstawowe operacje na pliku" --inputbox "Podaj nazwe pliku do edycji" 8 40 2>plik.txt
nazwa_pliku=$(cat plik.txt)
pico $nazwa_pliku ;;
"3") echo -n > archwizacja.txt
dialog --backtitle "Podstawowe operacje na pliku" --inputbox "Podaj nazwe pliku do usunięcia" 8 40 2>plik.txt
nazwa_pliku=$(cat plik.txt)
rm $nazwa_pliku ;;
esac
dialog --clear --title "Okno menu" \
--backtitle "Podstawowe operacje na pliku" \
--menu "Menu" 15 30 15 \
1 "Utwórz plik" \
2 "Edytuj plik" \
3 "Usuń plik" \
4 "Koniec programu" 2>plik.txt
opcja=$(cat plik.txt)
done
```


Skrypt 2



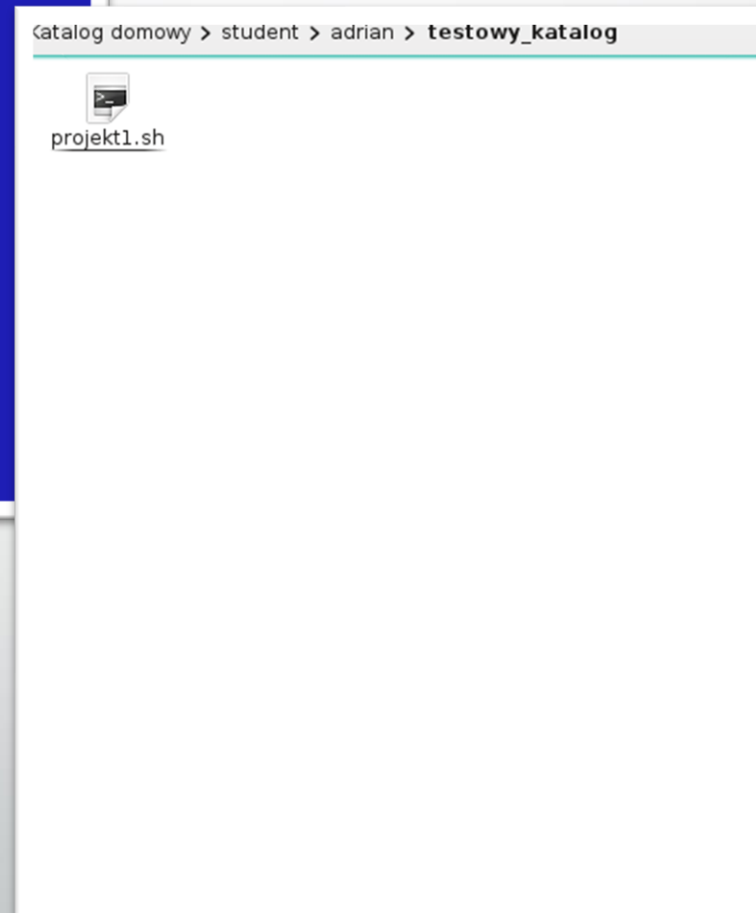
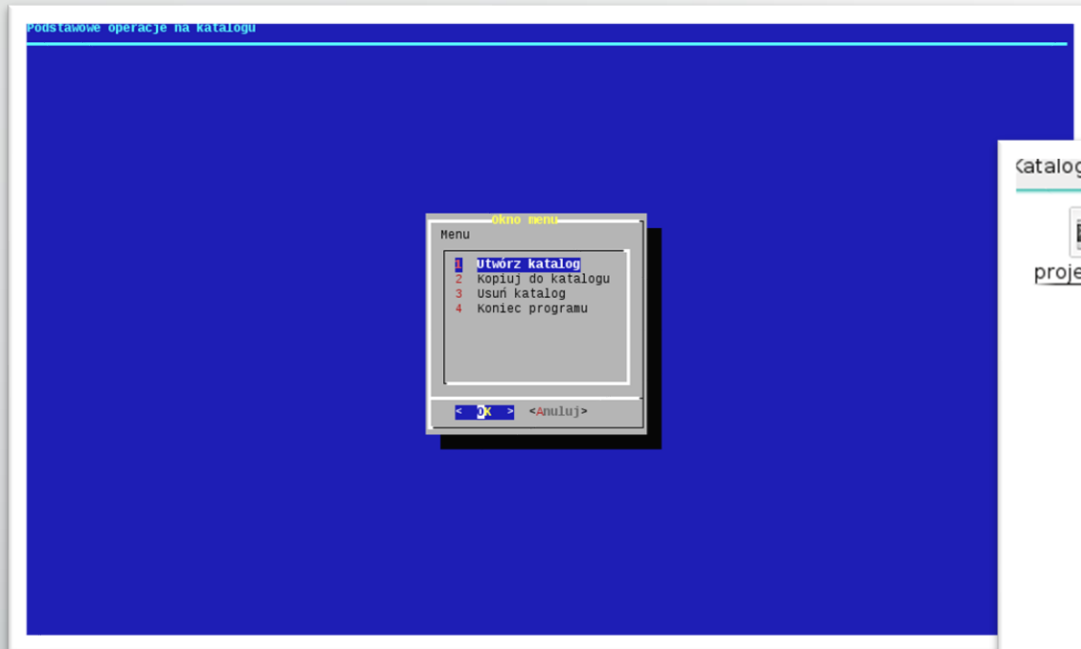
Skrypt 3

Skrypt pozwala na podstawowe operacje na katalogach: tworzenie, kopiowanie plików oraz usuwanie. Wykorzystane polecenia to:

```
mkdir
cp
rm -r
```

```
#!/bin/bash
dialog --clear --title "Okno menu" \
--backtitle "Podstawowe operacje na katalogu" \
--menu "Menu" 15 30 15 \
1 "Utwórz katalog" \
2 "Kopiuj do katalogu" \
3 "Usuń katalog" \
4 "Koniec programu" 2>katalog.txt
opcja=$(cat katalog.txt)
while test $opcja != 4
do
case $opcja in
"1") echo -n > katalog.txt
dialog --backtitle "Podstawowe operacje na katalogu" --inputbox "Podaj nazwe katalogu do utworzenia" 8 40 2>katalog.txt
nazwa_kat=$(cat katalog.txt)
mkdir $nazwa_kat ;;
"2") echo -n > katalog.txt
dialog --backtitle "Podstawowe operacje na katalogu" --inputbox "Podaj nazwe katalogu do którego chcesz skopiować" 8 40 2>katalog.txt
nazwa_kat=$(cat katalog.txt)
dialog --backtitle "Podstawowe operacje na katalogu" --inputbox "Podaj co chcesz skopiować" 8 40 2>katalog.txt
nazwa_co=$(cat katalog.txt)
cp $nazwa_co $nazwa_kat;;
"3") echo -n > katalog.txt
dialog --backtitle "Podstawowe operacje na katalogu" --inputbox "Podaj nazwe katalogu do usunięcia" 8 40 2>katalog.txt
nazwa_kat=$(cat katalog.txt)
rm -r $nazwa_kat ;;
esac
dialog --clear --title "Okno menu" \
--backtitle "Podstawowe operacje na katalogu" \
--menu "Menu" 15 30 15 \
1 "Utwórz katalog" \
2 "Kopiuj do katalogu" \
3 "Usuń katalog" \
4 "Koniec programu" 2>katalog.txt
opcja=$(cat katalog.txt)
done
```

Skrypt 3



Skrypt 4

Skrypt w okienku
wyświetla wszystkie
pliki z rozszerzeniem
.sh

```
#!/bin/bash
echo -n > lista.txt
for plik in *.sh
do
echo $plik >>lista.txt
done
dialog --backtitle "Lista plików .sh" --textbox lista.txt 15 50
```

Skrypt 4

```
Lista plikow .sh
```

```
projekt1.sh
projekt2.sh
projekt3.sh
projekt4.sh
projekt5.sh
testowy_plik.sh
```

<WYJAZ>

Skrypt 5

Skrypt wyświetla
podstawowe
informacje o
komputerze takie jak:

- Nazwa użytkownika
- Aktualny katalog
- Rodzaj systemu
- Ścieżka katalogu
domowego
- Nazwa hosta

```
#!/bin/bash
echo "Nazwa użytkownika: $USER"
echo "Ścieżka katalogu domowego: $HOME"
echo "Nazwa hosta: $HOSTNAME"
echo "Rodzaj systemu operacyjnego: $OSTYPE"
echo "Aktualny katalog: $PWD"
```

Skrypt 5

```
student@linux-2kkj:~/student/adrian> ./projekt5.sh  
Nazwa użytkownika: student  
Ściszka katalogu domowego: /home/student  
Nazwa hosta: linux-2kkj.site  
Rodzaj systemu operacyjnego: linux  
Aktualny katalog: /home/student/student/adrian  
student@linux-2kkj:~/student/adrian> █
```

```

#!/bin/bash
dialog --clear --title "Okno menu" \
--backtitle "Podstawowe operacje na pliku" \
--menu "Menu" 15 50 15 \
1 "Wypisanie wszystkich plików .sh" \
2 "Utwórz plik .sh" \
3 "Usuń plik .sh" \
4 "Zmień uprawnienia danego pliku" \
5 "Edytuj plik przez PICO" \
6 "Koniec programu" 2>opcja.txt
opcja=$(cat opcja.txt)
while test $opcja != 6
do
case $opcja in
"1") echo -n > opcja.txt
for plik in *.sh
do
echo $plik>>opcja.txt
done
dialog --textbox opcja.txt 15 50;;
"2") echo -n > opcja.txt
dialog --backtitle "Podstawowe operacje na pliku" --inputbox "Podaj nazwe pliku do utworzenia" 8 40 "nazwa_pliku" 2>opcja.txt
nazwa=$(cat opcja.txt)
touch $nazwa.sh ;;
"3") echo -n > opcja.txt
dialog --backtitle "Podstawowe operacje na pliku" --inputbox "Podaj nazwe pliku do usunięcia" 8 40 "nazwa_pliku" 2>opcja.txt
nazwa=$(cat opcja.txt)
rm $nazwa.sh ;;
"4") echo -n > opcja.txt
dialog --backtitle "Podstawowe operacje na pliku" --inputbox "Podaj nazwe pliku do zmiany uprawnień" 8 40 "nazwa_pliku" 2>opcja.txt
nazwa=$(cat opcja.txt)
echo -n > opcja.txt
dialog --backtitle "Podstawowe operacje na pliku" --inputbox "Podaj uprawnienia" 8 40 "uprawnienia" 2>opcja.txt
up=$(cat opcja.txt)
chmod $up $nazwa.sh ;;
"5") echo -n > opcja.txt
dialog --backtitle "Podstawowe operacje na pliku" --inputbox "Podaj nazwe pliku do edycji" 8 40 "nazwa_pliku" 2>opcja.txt
nazwa=$(cat opcja.txt)
pico $nazwa.sh ;;
esac
dialog --clear --title "Okno menu" \
--backtitle "Podstawowe operacje na pliku" \
--menu "Menu" 15 50 15 \
1 "Wypisanie wszystkich plików .sh" \
2 "Utwórz plik .sh" \
3 "Usuń plik .sh" \
4 "Zmień uprawnienia danego pliku" \
5 "Edytuj plik przez PICO" \
6 "Koniec programu" 2>opcja.txt
opcja=$(cat opcja.txt)
done

```

Skrypt pozwala na kilka operacji takich jak:

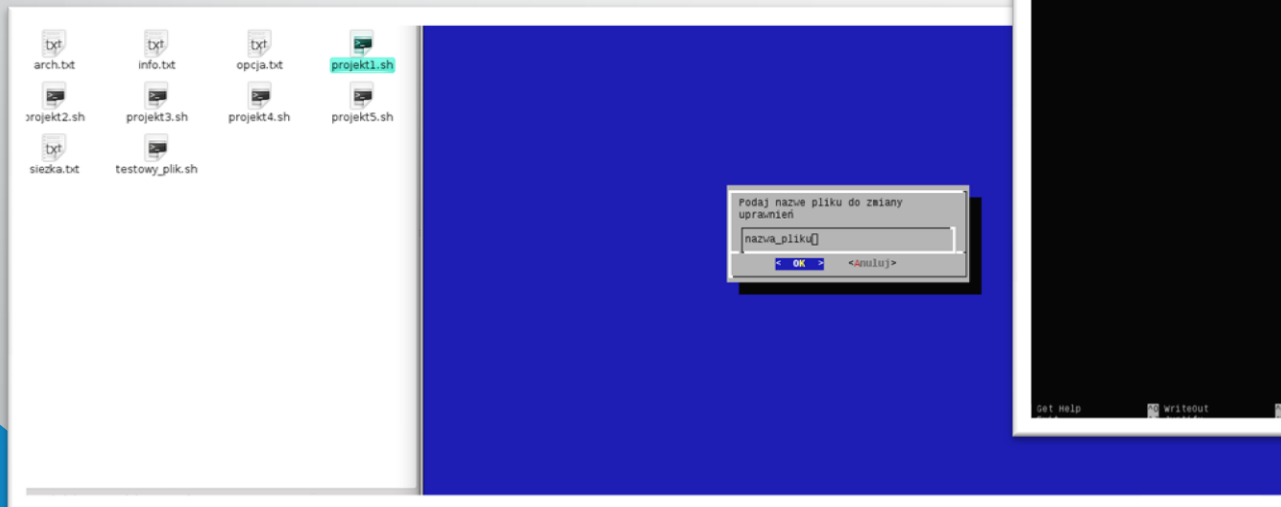
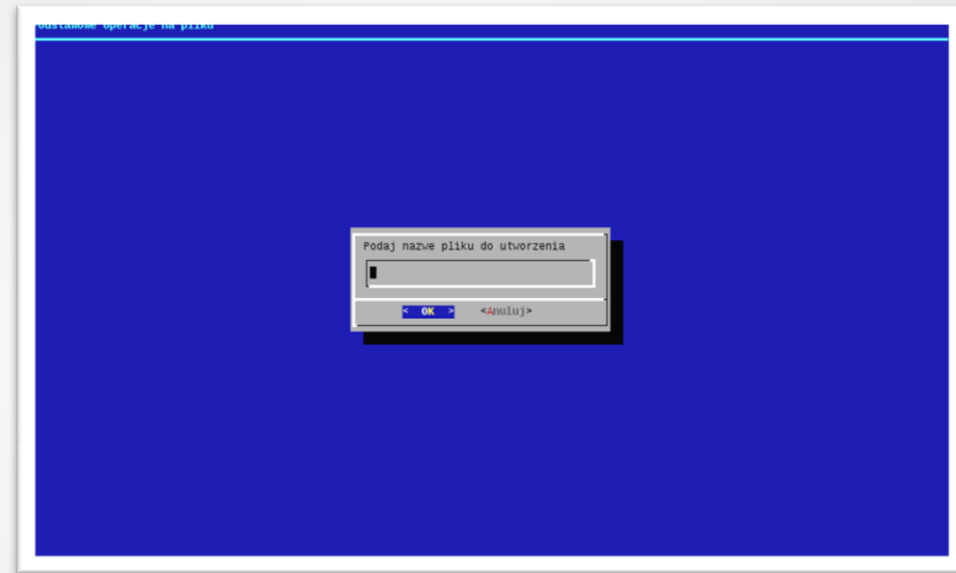
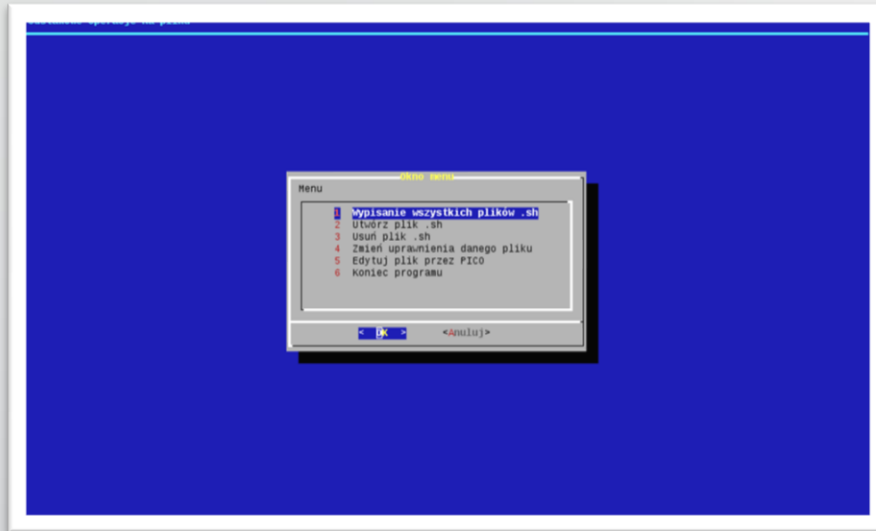
- Tworzenie pliku .sh
- Usunięcie pliku .sh
- Zmiana uprawnień do pliku
- Edycja pliku w edytorze PICO

Wykorzystuje instrukcje wyboru oraz pętle while

Podstawowe polecenia wykorzystane w skrypcie:

- touch
- rm
- chmod
- pico

Skrypt 1



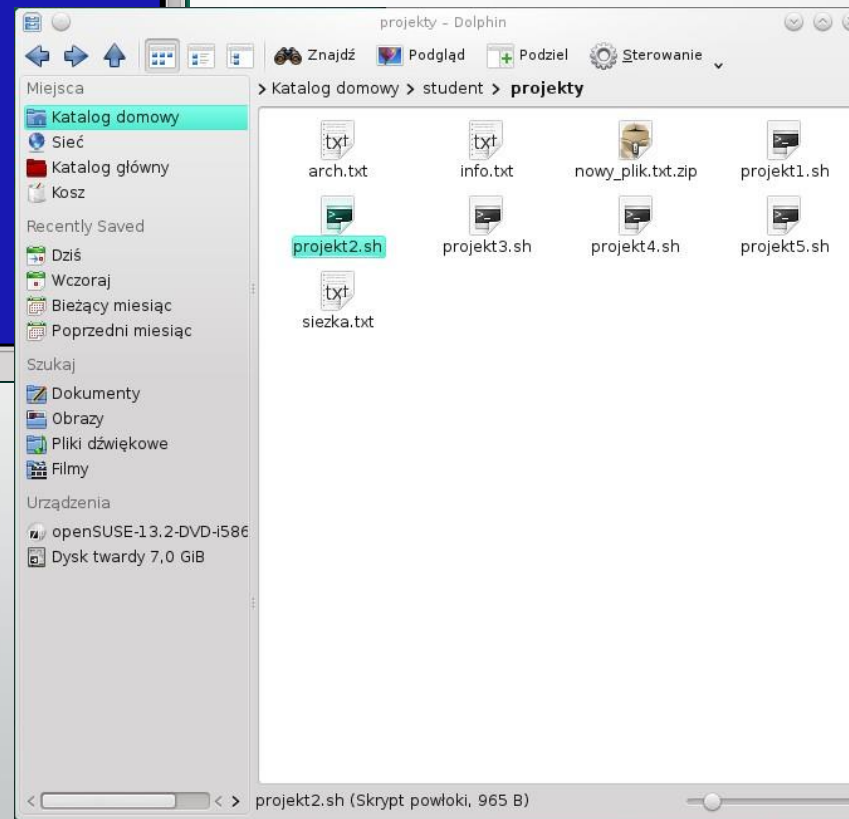
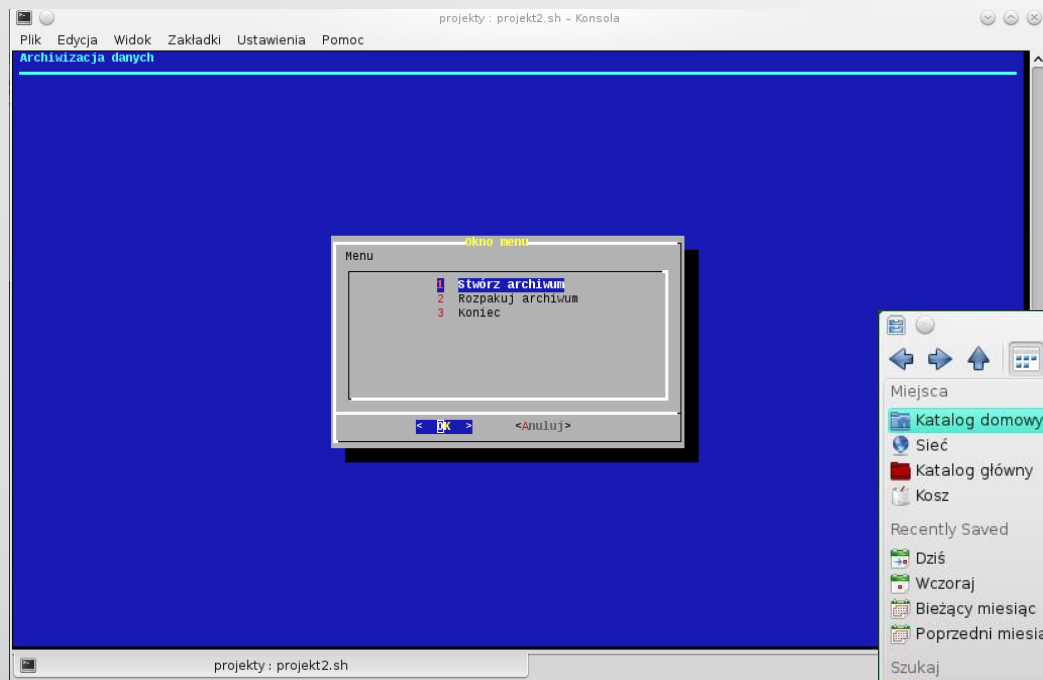
```

#!/bin/bash
dialog --clear --title "Okno menu" \
--backtitle "Archiwizacja danych" \
--menu "Menu" 15 50 15 \
1 "Stwórz archiwum" \
2 "Rozpakuj archiwum" \
3 "Koniec" 2>arch.txt
opcja=$(cat arch.txt)
while test $opcja != 3
do
case $opcja in
"1") echo -n > arch.txt
dialog --backtitle "Archiwizacja danych" --inputbox "Podaj co ma zostać zapakowane" 8 40 "nazwa_pliku" 2>arch.txt
nazwa_pliku=$(cat arch.txt)
echo -n > arch.txt
dialog --backtitle "Archiwizacja danych" --inputbox "Podaj nazwę archiwum" 8 40 "nazwa_pliku" 2>arch.txt
op=$(cat arch.txt)
tar -cf $op.tar $nazwa_pliku ;;
"2") echo -n > arch.txt
dialog --backtitle "Archiwizacja danych" --inputbox "Podaj nazwę archiwum" 8 40 "nazwa_pliku" 2>arch.txt
nazwa_arch=$(cat arch.txt)
tar -xvf $nazwa_arch.tar ;;
esac
dialog --clear --title "Okno menu" \
--backtitle "Archiwizacja danych" \
--menu "Menu" 15 50 15 \
1 "Stwórz archiwum" \
2 "Rozpakuj archiwum" \
3 "Koniec" 2>arch.txt
opcja=$(cat arch.txt)
done|

```

Skrypt pozwala na archiwizację danych oraz rozpakowanie plików .tar. Skrypt wykorzystuje polecenie tar oraz instrukcję wyboru oraz pętlę while. Skrypt jest wersji okienkowej.

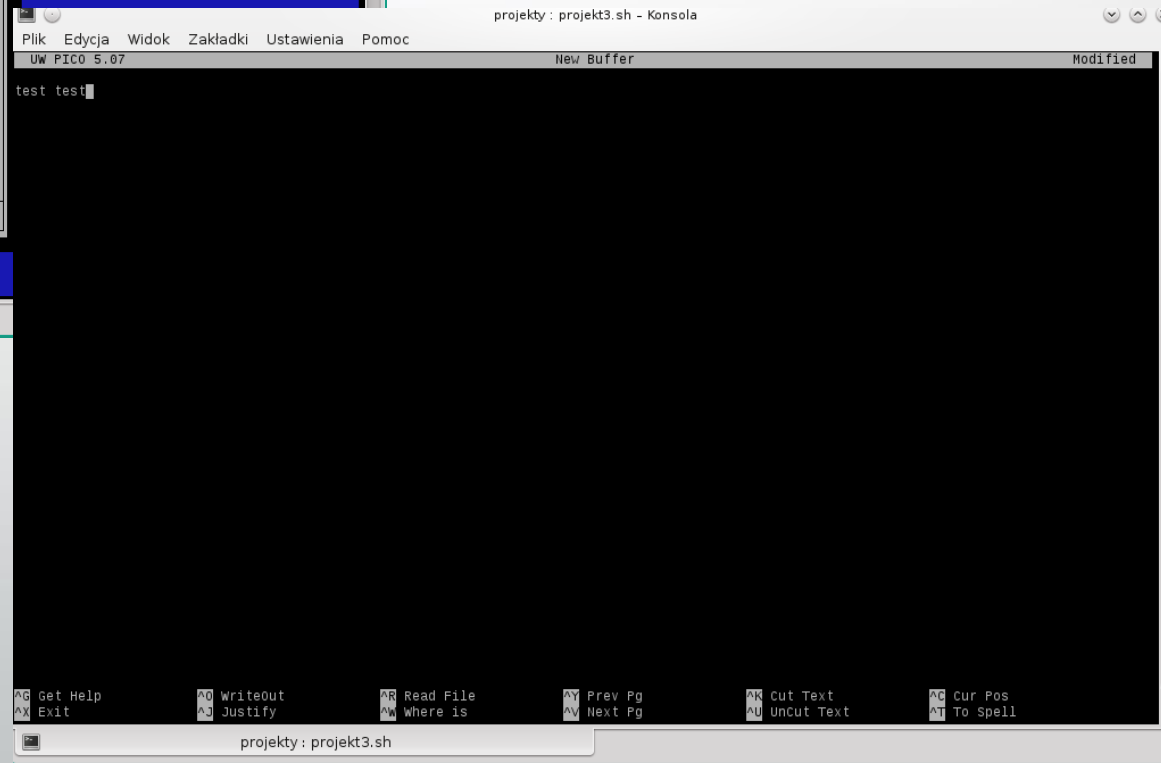
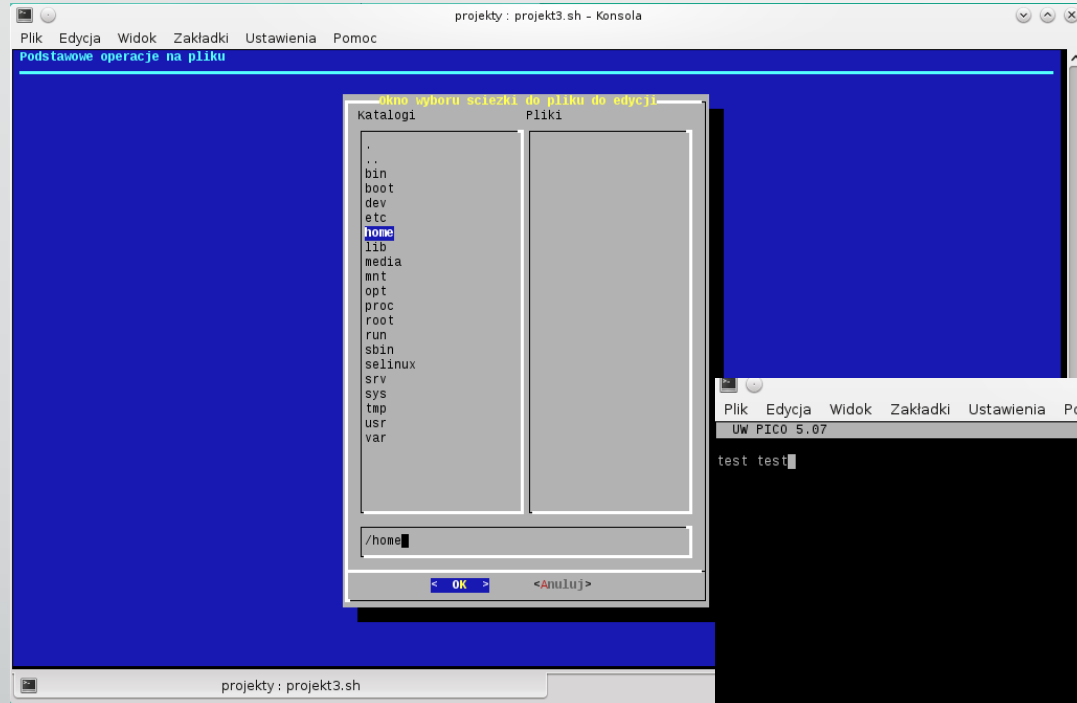
Skrypt 2



```
#!/bin/bash
dialog --clear --title "Okno wyboru ścieżki do pliku do edycji" \
--backtitle "Podstawowe operacje na pliku" \
--fselect /home 25 50 2>ściezka.txt
opcja=$(cat ściezka.txt)
echo -n > ściezka.txt
pico $opcja
```

W skrypcie określamy ścieżkę kierującą w miejsce gdzie chcemy utworzyć plik przez edytor PICO wersji okienkowej.

Skrypt 3



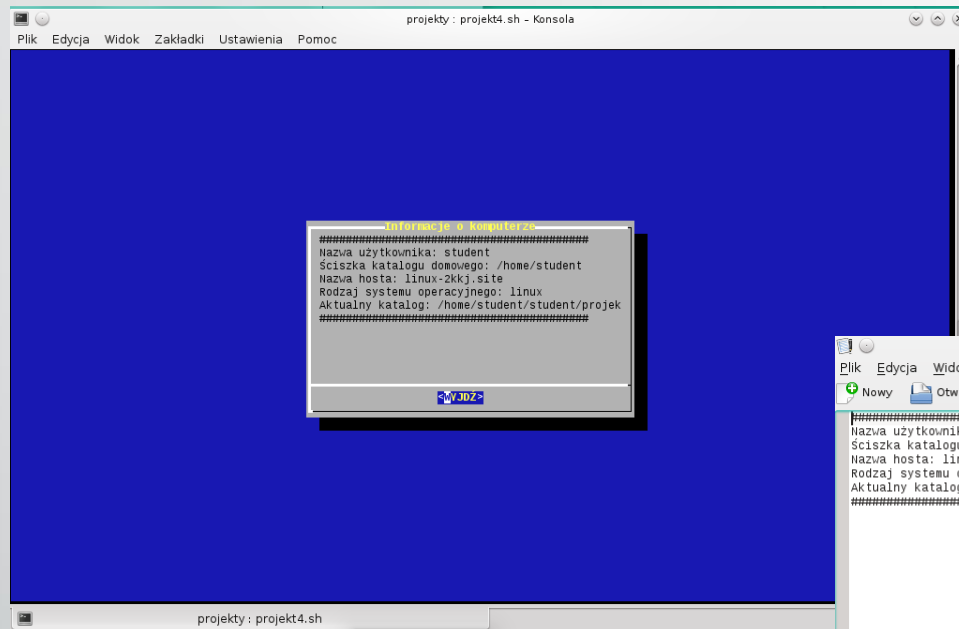
```
#!/bin/bash
echo -n > info.txt
echo "#####" >> info.txt
echo "Nazwa użytkownika: $USER" >> info.txt
echo "Ściszka katalogu domowego: $HOME" >> info.txt
echo "Nazwa hosta: $HOSTNAME" >> info.txt
echo "Rodzaj systemu operacyjnego: $OSTYPE" >> info.txt
echo "Aktualny katalog: $PWD" >> info.txt
echo "#####" >> info.txt
dialog --title "Informacje o komputerze" --textbox info.txt 15 50
```

Skrypt zapisuje do pliku info.txt podstawowe informacje o komputerze. Wykorzystuje zmienne środowiskowe takie jak:

- \$USER
- \$HOME
- \$HOSTNAME
- \$OSTYPE
- \$PWD

Informacje wyświetlane są w postaci okienkowej.

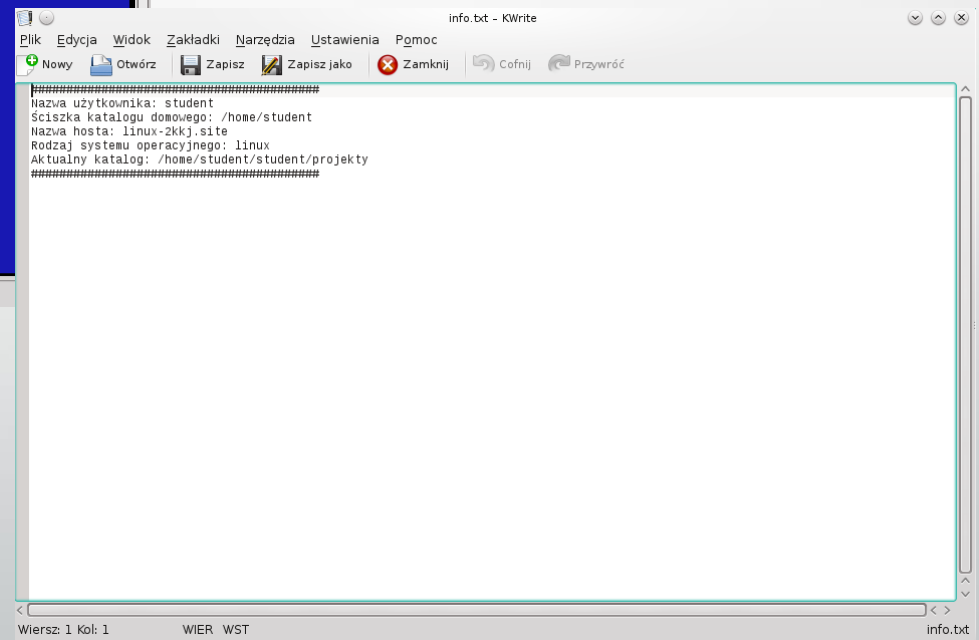
Skrypt 4



A terminal window titled "projekty : projekt4.sh - Konsola" with a menu bar containing "Plik", "Edycja", "Widok", "Zakładki", "Ustawienia", and "Pomoc". The terminal background is blue. A dialog box titled "Informacje o komputerze" is overlaid on the terminal, containing the following text:

```
#####  
Informacje o komputerze  
#####  
Nazwa użytkownika: student  
Ściszka katalogu domowego: /home/student  
Nazwa hosta: linux-2kkj.site  
Rodzaj systemu operacyjnego: linux  
Aktualny katalog: /home/student/student/projek  
#####
```

The dialog box has a "OK" button at the bottom. The terminal window has a taskbar at the bottom with the label "projekty : projekt4.sh".



A KWrite editor window titled "info.txt - KWrite" with a menu bar containing "Plik", "Edycja", "Widok", "Zakładki", "Narzędzia", "Ustawienia", and "Pomoc". The window contains the following text:

```
#####  
Nazwa użytkownika: student  
Ściszka katalogu domowego: /home/student  
Nazwa hosta: linux-2kkj.site  
Rodzaj systemu operacyjnego: linux  
Aktualny katalog: /home/student/student/projekty  
#####
```

The window has a toolbar with icons for "Nowy", "Otwórz", "Zapisz", "Zapisz jako", "Zamknij", "Cofnij", and "Przywróć". The status bar at the bottom shows "Wiersz: 1 Kol: 1" and "WIER WST". The file name "info.txt" is visible in the bottom right corner.

```
#!/bin/bash
echo "Opcje:"
echo "1 - Tworzenie katalogu"
echo "2 - Zmiana nazwy katalogu"
echo "3 - Przenieś plik do katalogu"
echo "4 - Koniec"
read opcja
while test $opcja != 4
do
case $opcja in
"1")
echo "Podaj nazwe katalogu do utworzenia:"
read nazwa_katalogu
mkdir $nazwa_katalogu ;;
"2")
echo "Podaj stara nazwe katalogu"
read stare
echo "Podaj nową nazwe katalogu"
read nowa
mv $stare $nowa ;;
"3")
echo "Podaj nazwe katalogu:"
read katalog
echo "Podaj nazwe pliku"
read plik
cp $plik $katalog ;;
esac
echo "Opcje:"
echo "1 - Tworzenie katalogu"
echo "2 - Zmiana nazwy katalogu"
echo "3 - Przenieś plik do katalogu"
echo "4 - Koniec"
read opcja
done
```

Skrypt pozwala na kilka operacji takich jak:

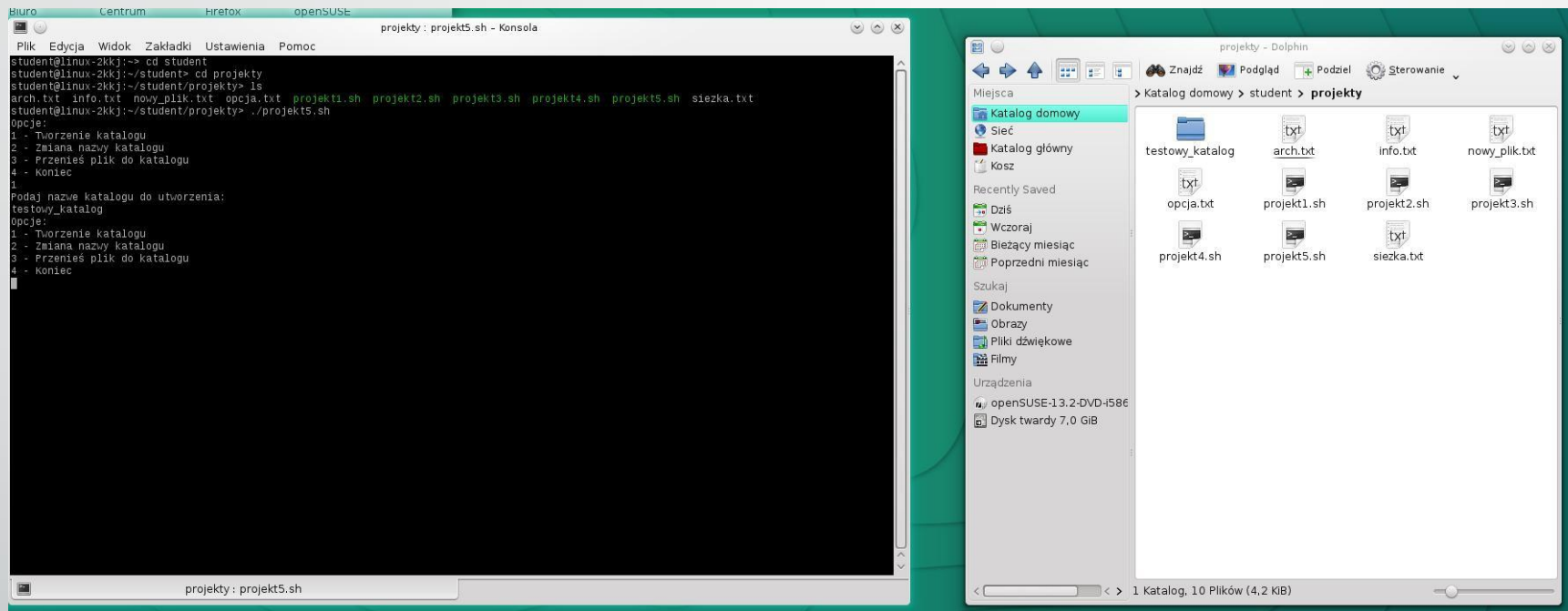
- Tworzenie katalogu
- Zmiana nazwy katalogu
- Przenoszenie pliku do danego katalogu

Wykorzystuje instrukcje wyboru oraz pętle while

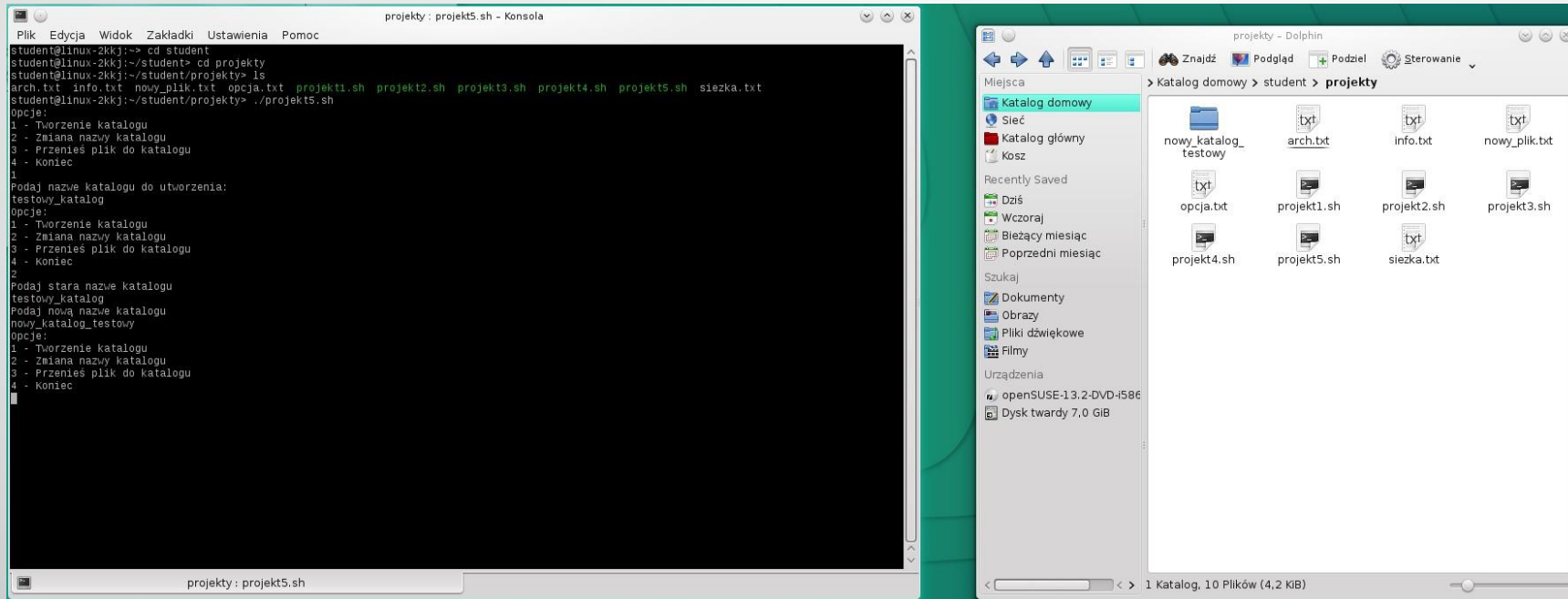
Podstawowe polecenia wykorzystane w skrypcie:

- mkdir
- mv
- cp

Skrypt 5



Skrypt 5



Skrypt 5

