

Практична робота № 19.

Розробка сценаріїв *bash* для графічних робочих столів. Використання пакета *zenity*

Графічні оболонки Linux

Графічна оболонка середовища, являє собою оболонку операційної системи, яка забезпечує зв'язок користувача з різними функціями системи, тобто, просто кажучи, дозволяє управляти нею.

У **Windows**, по суті, всього одна графічна оболонка, тобто користувач може змінювати графічну тему, змінювати налаштування деяких графічних елементів (наприклад, змінити іконку у папки), але сам графічний інтерфейс буде залишатись тим же.

У **Linux** же таких графічних оболонок кілька і користувач може встановити їх всі, а під час входу в систему вибрати ту, яка йому більше подобається. Найбільш поширені середовища в Linux – це **KDE** та **GNOME**. Вони являють собою два конкуруючих середовища, тобто якщо користувач має більш-менш потужний комп'ютер, то вибір графічної оболонки зупиниться на цих двох середовищах, а вже яку встановити – буде залежати тільки від нього.

Обираючи графічне середовище, користувач обирає набір програм, з якими буде працювати. Середовище **KDE** використовує для роботи бібліотеку **Qt**, а **GNOME** – **GTK**. Якщо був обраний KDE, то будуть встановлені програми, що працюють на бібліотеці Qt, якщо в GNOME, то відповідно встановлюються програми, засновані на бібліотеці GTK. Наприклад, як файловий менеджер під час вибору KDE буде встановлено Dolphin, а у результаті вибору GNOME – Nautilus. Якщо дозволяє дисковий простір, то звичайно ж можна встановити їх разом і у результаті завантаження операційної системи можна вибирати, в якій графічній оболонці ми сьогодні захочемо працювати, але запустити програму під невласливим їй середовищем не вийде.

Графічне середовище XFCE

XFce є графічною оболонкою, що побудована на основі інструментального пакета **GTK+**, який використовується в **Gnome**, але набагато легша і призначена для тих, кому потрібен простий, ефективно працюючий стіл, який легко використовувати і налаштувати. Переваги **XFce**:

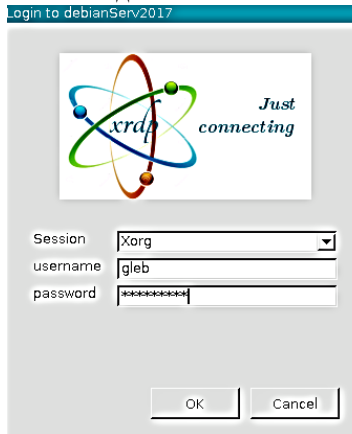
– простий, легкий у користуванні робочий стіл, який повністю налаштовується за допомогою миші, з інтерфейсом «drag and drop»;

- інтегрований віконний менеджер, менеджер файлів, управління звуком, модуль сумісності з **Gnome** та інше;
- стандартизовані меню і панелі інструментів, комбінації клавіш, кольорні схеми та ін.;
- можливість використання тем, оскільки використовує **GTK+**;
- **XFce** швидка, легка і ефективна: ідеальна для застарілих / слабких машин або машин з обмеженою пам'яттю.

На сервері Debian в університеті встановлене графічне середовище **XFce**. Кожен з користувачів, зареєстрованих у системі, може віддалено під'єднатись до нього за допомогою оснастки **mstsc** (підключення до віддаленого робочого столу), увівши IP-адресу сервера Linux,



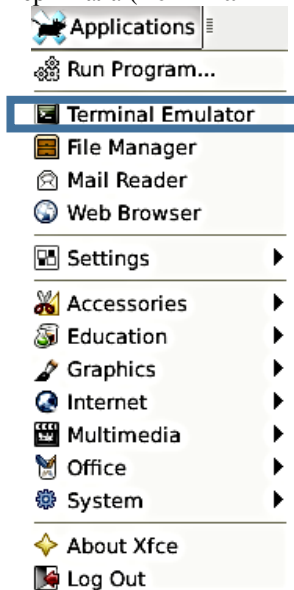
а потім після встановлення підключення ввести свій логін та пароль.



Після введення необхідних даних стане можливою робота на віддаленому робочому столі **XFce**.



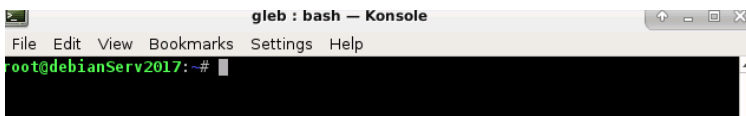
Подібно до меню «Пуск» у Windows, у графічній оболонці **Xfce** існує меню «Applications», що дозволяє відкрити певні програми, які розподілені за категоріями. Для виконання цієї лабораторної роботи знадобиться емулятор терміналу (**Terminal Emulator**).



Відкрити емулятор термінала також можна і за допомогою меню швидкого доступу внизу робочого стола **XFce**.



Після відкриття емулятора термінала ми бачимо звичний інтерфейс командного рядка.



Написання сценаріїв з використанням візуальних елементів графічних середовищ

Колись сценарії командного інтерпретатора представляли собою один з найбільш поширених способів організації роботи в системі, але з часом вони стали сприйматися як похмурі і нудні. Але так не має бути, якщо сценарій призначений для виконання в графічному середовищі. Передбачено широкий спектр можливостей забезпечення взаємодії з користувачем сценарію, не базованих виключно на інструкціях **read** і **echo**.

Утиліта **dialog** – одна з небагатьох, які допомагають створювати доброзичливі до користувача (в плані призначеного для користувача інтерфейсу) сценарії і програми. Вона може конструювати різні діалогові вікна, не потребуючи при цьому запущений графічний інтерфейс – достатньо лише консолі, адже для малювання використовується бібліотека **ncurses**. Як наслідок, **dialog** добре пристосований для використання у сценаріях **bash**, в яких він і застосовується в більшості випадків.

Варто відзначити, що існують аналоги **dialog**'а, спеціально розраховані на використання в графічному оточенні – універсальний **Xdialog** (повністю сумісний з **dialog**), **gtkdialog** (вікна малюються за допомогою графічної бібліотеки GTK), **kdialog** (для роботи у графічному середовищі **KDE**) і **zenity** (призначений для використання разом з графічним середовищем **Gnome**). Слід підкреслити, що, принаймні, **Xdialog**, крім стандартних діалогових вікон з **dialog**'а, має ще й трохи своїх: деревовидні списки, поле для редагування багаторядкових текстових даних та ін.

У команді **dialog** використовуються параметри командного рядка для визначення того, який графічний елемент Windows повинен бути

сформований. Термін «графічний елемент» (widget) застосовується в пакеті dialog для позначення графічного об'єкта, приблизно відповідного одному з об'єктів Windows.

Ось короткий список діалогових вікон, які здатний малювати dialog:

- вікна з кнопками Так / Ні;
 - меню;
 - вікна з полями введення;
 - інформуючі вікна і просто вікна з текстом;
 - поля з радіокнопками;
 - вікна для вибору файлів;
 - вікна для вибору необхідного часу, дати;
 - спеціальні вікна для запиту пароля;
- і багато іншого.

Огляд утиліти zenity

Zenity – це засіб створення діалогових вікон в режимі командного рядка. Слід зазначити, що насправді діалогові вікна створюються засобами **Gtk+**, тому в системі повинні бути встановлені відповідні бібліотеки. Від аналогічних програм **zenity** відрізняють більш витончені засоби реалізації GUI-елементів.

Завантажити актуальну версію цієї утиліти можна на Web-сайті розробників [<http://library.gnome.org/users/zenity/>]. Також цю програму можна знайти в репозиторіях деяких дистрибутивів Linux.

Застосування **zenity** для окремих команд в інтерактивному режимі не настільки ефективне, як під час написання сценаріїв. У сценаріях командної оболонки часто потрібна взаємодія з користувачем, щоб повідомити якусь інформацію, наприклад, про виникнення «нестандартної» ситуації. Також потрібно відображати інформацію про стан виконання операції, що триває досить довго. Крім цього, іноді сценарієм необхідно отримати деяку інформацію від користувача: вибір варіанта відповіді на поставлене запитання, вибір файлу із запропонованого списку та ін. Все це можна організувати за допомогою **zenity**.

Необхідно уточнити, що після закриття діалогового вікна, **zenity** повертає числовий код завершення операції:

- 0 – означає, що користувач натиснув в діалоговому вікні кнопку «ОК» або «Закрити» (Close);
- 1 – означає, що користувач натиснув кнопку «Скасувати» (Cancel) або скористався функціями (кнопками) вікна, щоб закрити його;
- -1 – повідомляє про те, що операція завершилася з помилкою;

– 5 – діалогове вікно було закрито після закінчення інтервалу часу очікування.

Створення діалогового вікна для виведення повідомлень

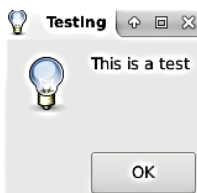
Щоб почати використовувати **zenity**, на практиці не потрібно володіти особливими знаннями або вміннями, досить просто познайомитися з різними опціями (ключами), що дозволяють сповна використовувати можливості цієї програми.

У **zenity** визначено чотири типи діалогових вікон для виводу повідомлень:

- помилка (ключ **--error**);
- інформація (ключ **--info**);
- питання (ключ **--question**);
- попередження (ключ **--warning**).

Просте повідомлення визначається такою командою:

```
$ zenity --info --title="Testing" --text="This is a test"
```



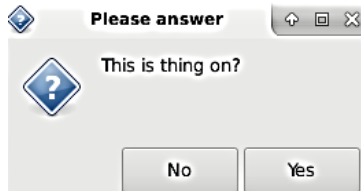
Якщо застосований емулятор терміналу підтримує мишу, то з'являється додаткова можливість клацати на кнопці OK для закриття діалогового вікна. Для моделювання натиснення можна також використовувати команди клавіатури; в нашому випадку достатньо натиснути клавішу <Enter>.

Створення діалогового вікна для вибору варіанта відповіді

Діалогове вікно з питанням являє собою більш складний варіант графічного елемента повідомлення у **zenity**, який дозволяє користувачеві відповісти на питання, що відображене у вікні, позитивно або негативно. Цей елемент створює в нижній частині вікна дві кнопки, на однію з яких є напис «Yes», а на іншій – «No». Користувач може переходити від однієї кнопки до іншої за допомогою миші, клавіші табуляції або клавіш зі стрілками. Щоб вибрати кнопку, користувач може натиснути клавішу пробілу або <Enter>. Для створення такого діалогового вікна до команді **zenity** потрібно вказати ключ **--question**.

Нижче наведено приклад використання графічного елемента `zenity`.

```
$ zenity --question --title="Please answer" --text="Is this thing on?"
```



```
$ echo $? (була натиснута кнопка «No»)
```

```
1
```

```
$
```

Статус виходу команди `zenity` встановлюється залежно від того, яку кнопку обрав користувач. Якщо обрана кнопка **No**, статусом виходу є 1, а якщо обрана кнопка **Yes**, статус виходу набуває значення 0.

Приклад. Файл `yesno.sh`

```
#!/bin/bash

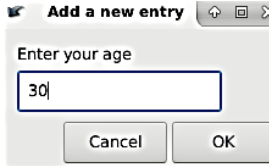
zenity --question \
--title="Displays the contents of the current \
directory" \
--text="Do you want to use a long format?"

case $? in
  0)
    ls -l;;
  1)
    ls;;
  -1)
    echo "There was an error!"
    exit 1;;
esac
```

Введення та редагування тексту

Під час взаємодії з користувачем неможливо обійтися без засобів введення текстової інформації. Звісно, `zenity` надає і ці засоби. Для цього разом з командою потрібно використати ключ `--entry`. При цьому графічний елемент створює дві кнопки – **OK** і **Cancel**. Якщо обрана кнопка **Cancel**, то статусом виходу команди є 1; в іншому випадку статус виходу дорівнюватиме 0, а введені користувачем дані при цьому виводяться у стандартний потік **STDOUT**.

```
$ zenity --entry --text="Enter your age"
```



```
30
```

```
$ echo $? (була натиснута кнопка «ОК»)
```

```
0
```

```
$
```

Також можна присвоїти результат виконання команди *kdialog* певній змінній.

```
$ age=`zenity --entry --text="Enter your age"`
```

```
$ echo $age
```

```
30
```

Після заголовку (перший параметр) можна задати другий параметр, який представляє значення за замовчуванням:

```
$ age=`kdialog --inputbox «Enter your age:» 30`
```

У полі введення може міститись слово, визначене за замовчуванням (ключ *--entry-text*). Також введені в поле дані можна приховати, використавши ключ *--hide-text*.

Приклад. Файл *inputPassword.sh*

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

```
if zenity --entry --title="Verify access rights" \
  --text="Input your password:" \
  --entry-text="password" \
  --hide-text
```

```
then
```

```
  echo $?
```

```
else
```

```
  zenity --error --text="Password not entered"
```

```
fi
```



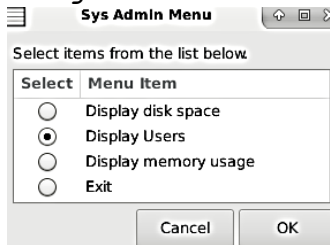
Списки

У **zenity** не залишені без уваги і такі важливі елементи інтерфейсу, як списки (ключ **--list**). Поряд зі звичайним текстовим списком є можливість створення списків з перемикачами і радіокнопками. Елементи для діалогового вікна, що містить список, повинні бути вказані в такому порядку: стовпці формують рядок, далі символ нового рядка, після чого вводяться стовпці, що становлять наступний рядок та ін. Рядки елементів списку можуть бути передані в діалогове вікно через потік стандартного введення. Новий стовпець вказується за допомогою ключа **--column**.

Перемикачі (ключ **--checklist**) і радіокнопки (**--radiolist**) можуть розташовуватися тільки в першому стовпці, тому в рядках для цих варіантів списку найпершим елементом обов'язково повинен бути рядок «TRUE» або «FALSE».

За допомогою списків з радіокнопками у першому стовпці можна реалізувати інтерактивне меню.

```
$ zenity --list --radiolist --title «Sys Admin Menu» --column "Select" --column "Menu Item" FALSE "Display disk space" FALSE "Display users" FALSE "Display memory usage" FALSE "Exit"
```



Display users

```
$
```

Приклад. Файл menu.sh

```
#!/bin/bash
command=`zenity --list --radiolist --title «Sys Admin Menu» \
--column "Select" --column "Menu Item" \
FALSE "Display disk space" \
FALSE "Display users" \
FALSE "Display memory usage" \
FALSE "Exit"`

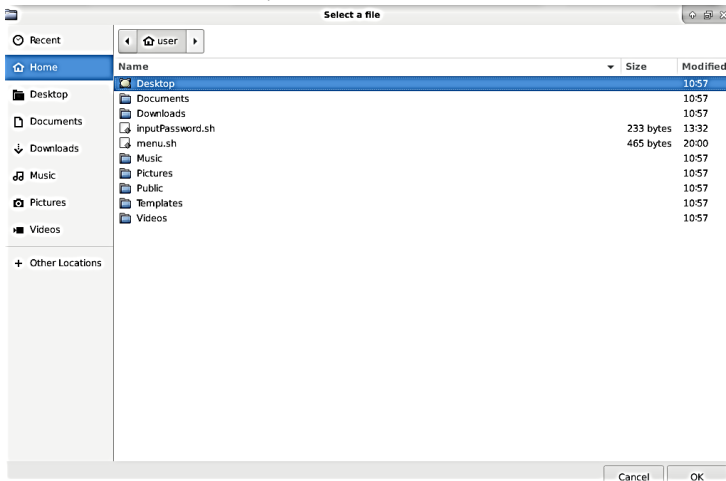
if [ $? -eq 0 ]
```

```
then
  case $command in
    1)
      df ;;
    2)
      who ;;
    3)
      vmstat ;;
    4)
      exit ;;
  esac
fi
```

Діалогове вікно вибору файлу

Команда **zenity** дозволяє скористатися ще кількома привабливими вбудованими графічними елементами. Під час роботи з іменами файлів надзвичайно зручним є графічний елемент **--file-selection**, який дозволяє позбутися від необхідності введення імен використовуваних файлів, оскільки з його допомогою можна перейти безпосередньо до того каталогу, де знаходиться файл, і зробити вибір.

```
$ zenity --file-selection --title="Select a file" --filename="$HOME/"
```



Додатковий ключ **--filename** задає місце розташування каталогу, з якого починається перехід по каталогам у вікні, або файл, який позначається вибраним за замовчуванням. Вікно графічного

елемента **--file-selection** містить лістинг каталогів, що знаходиться зліва, та лістинг файлів, який представлений справа. Повне ім'я обраного файлу у результаті виводиться до стандартного потоку виведення.

Нижче наведений приклад сценарію, який містить вікно вибору файлу, який потрібно видалити.

Приклад. Файл Deletefile.sh

```
#!/bin/bash

delete_file()
{
    local f=$1
    local m="$0: file $f failed to delete."
    if [ -f $f ]
    then
        rm $FILE && m="$0: $f file deleted."
    else
        m="$0: $f is not a file."
    fi

    zenity --info --title=" Remove file" --
text="$m"
}

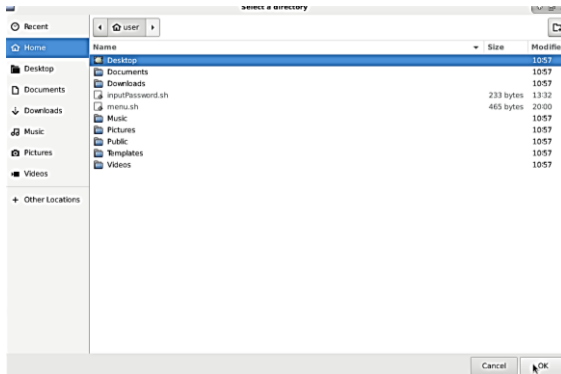
file=`zenity --file-selection --title="Select a
file" --filename=" $HOME/"`

if [ ! -z $file ] then
    delete_file $file
```

Вікно вибору каталогу

Подібно до діалогового вікна вибору існуючого файлу, у **zenity** передбачена опція вибору існуючого каталогу. Для цього під час створення діалогового вікна вибору файлу необхідно вказати додатковий ключ **--directory**. Оброблюється таке діалогове вікно так само, як і діалогове вікно вибору файлу.

```
$ kdialog -title «Select a file» --
getexistingdirectory ~
```



Завдання

1. Написати сценарій для графічних робочих столів, який шукає заданий користувачем у діалоговому вікні рядок у всіх файлах заданого за допомогою діалогового вікна вибору певного каталогу, що знаходиться у вашому домашньому каталозі. Передбачити перевірку, чи є елемент з обраним іменем каталогом. Якщо це каталог, вивести у інформаційному діалоговому вікні повний шлях і імена файлів, у вмісті яких присутній заданий рядок, і їх розмір, якщо ні, вивести повідомлення, що це не каталог.
2. Написати сценарій для графічних робочих столів, який приймає від користувача ім'я файлу (за допомогою діалогового вікна вибору файлу) та надає користувачеві обрати одну з дій над ним (переглянути, скопіювати, перемістити, видалити), використовуючи інтерактивне меню. Під час вибору користувачем певної команди передбачити необхідні параметри для її виконання (назва файлу або каталогу у разі копіювання або переміщення), а також вивести для користувача вікно для додаткового підтвердження, чи дійсно він бажає виконати цю дію.

Контрольні питання

1. Які елементи графічного інтерфейсу можна використати за допомогою команди `kdialog`?
2. Як обробити введення даних користувачем у діалозі?
3. Як додати до сценарію вікно з кнопками Yes / No?
4. Як додати до сценарію поле для введення даних?
5. Як додати до сценарію інтерактивне меню?
6. Як додати до сценарію вікно для вибору файлу та вікно для вибору каталогу?