

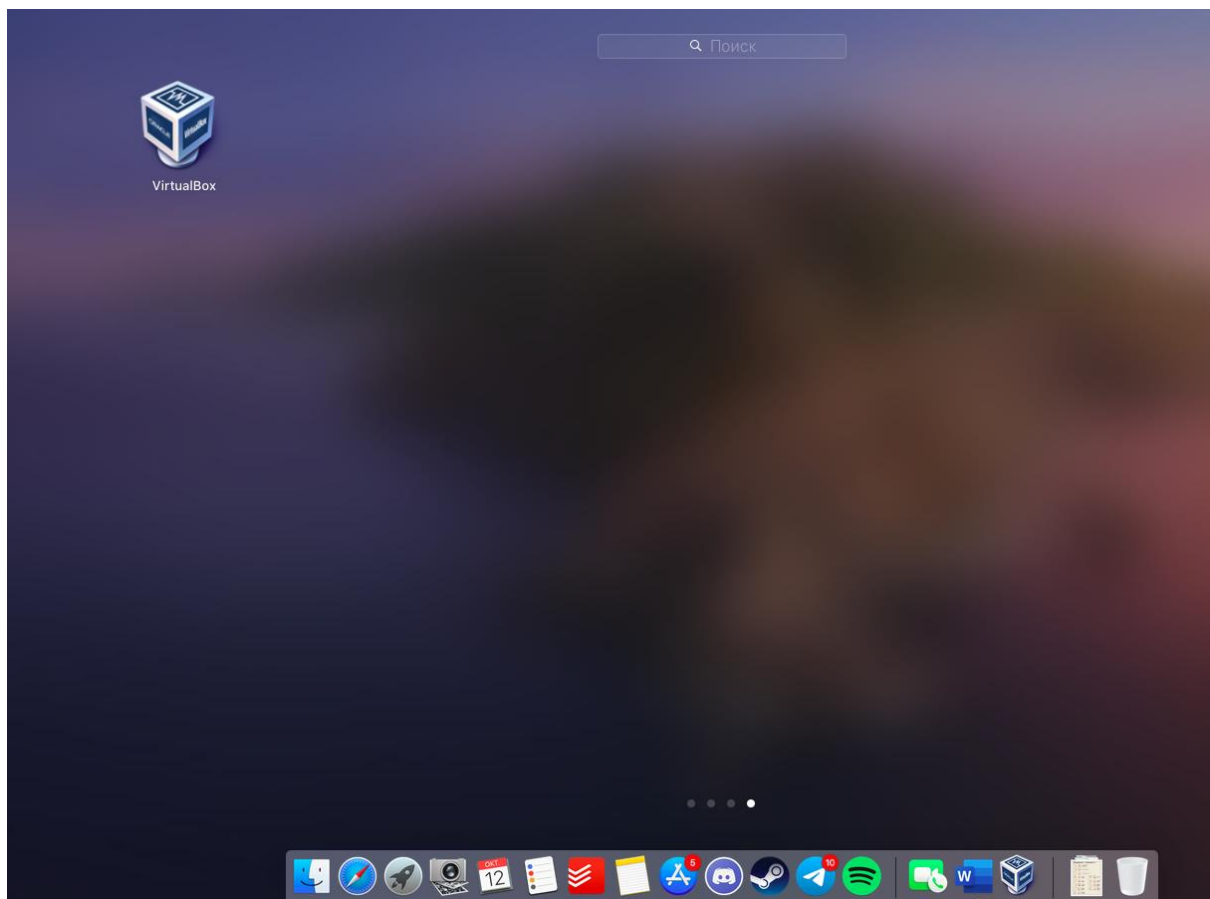
Радєвич Владислав

## Лабораторна робота №1. Вступ до програмування. Освоєння командної строки Linux

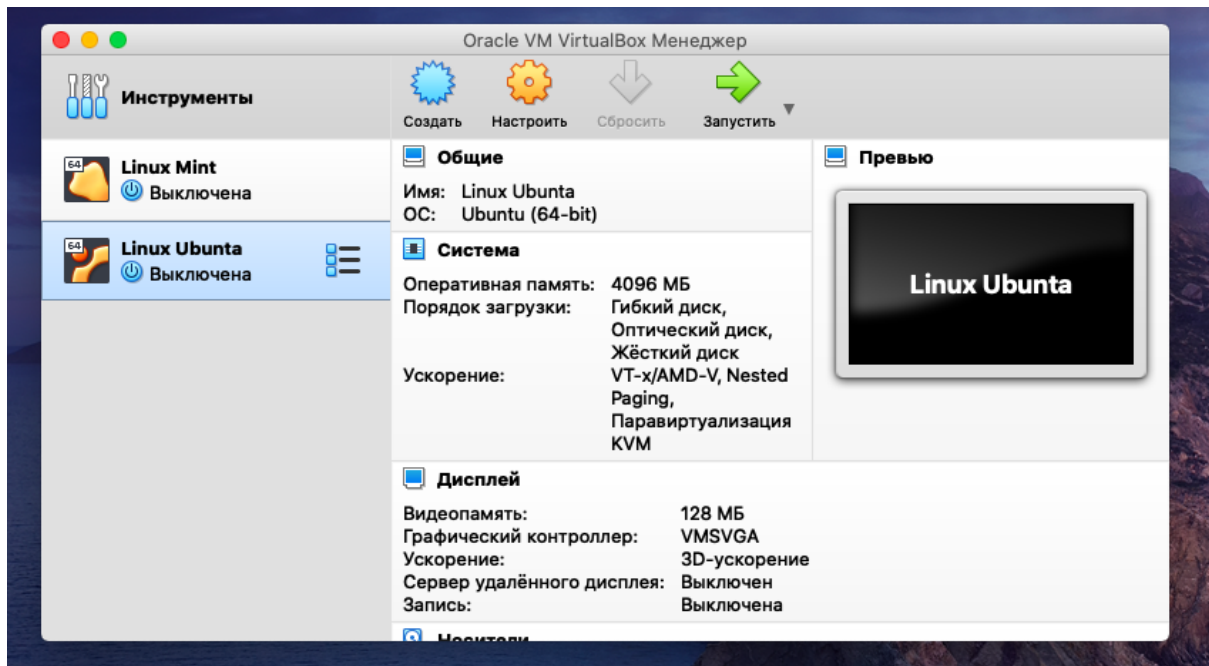
Хід роботи:

### 1. Встановлення VirtualBox

VirtualBox було встановлено, на операційну систему macOS, з офіційного сайту VirtualBox.org. (Для операційної системи macOS не підходить найновіша версія VirtualBox (6.1.14), тому для повноцінної роботи довелося встановити програму версії 6.1.6)



2. Завантаження та встановлення образу операційної системи Linux. Образ системи Linux було завантажено з сайту [osboxes.org](https://osboxes.org). Далі цей образ був добавлений в програму VirtualBox.



### 3. Встановлення програмних додатків.

Для цього пункту було необхідно запустити Linux, відкрити термінал та ввести команду «`sudo apt install [назва додатка]`» та натиснути enter. Додатки, які були встановлені: git, gcc, tree, make, ddd.

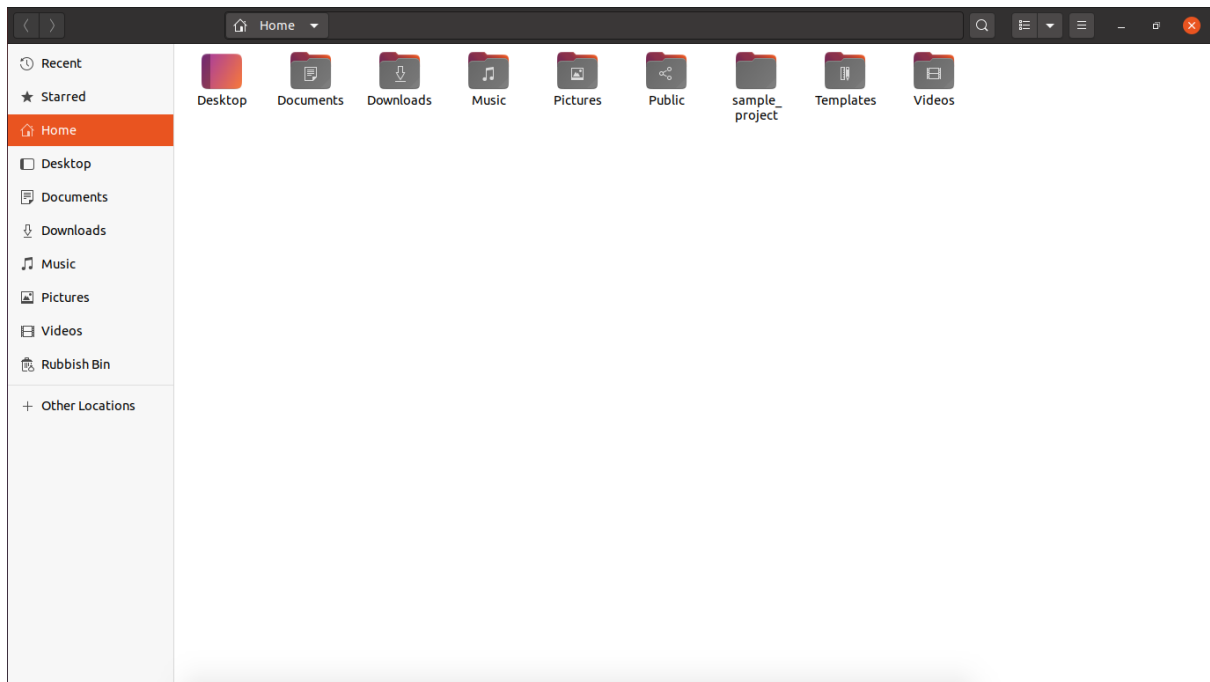
```

osboxes@osboxes:~$ sudo apt install git
[sudo] password for osboxes:
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.25.1-1ubuntu3).
0 to upgrade, 0 to newly install, 0 to remove and 282 not to upgrade.
osboxes@osboxes:~$ sudo apt install gcc
Reading package lists... Done
Building dependency tree
Reading state information... Done
gcc is already the newest version (4:9.3.0-1ubuntu2).
0 to upgrade, 0 to newly install, 0 to remove and 282 not to upgrade.
osboxes@osboxes:~$ sudo apt install tree
Reading package lists... Done
Building dependency tree
Reading state information... Done
tree is already the newest version (1.8.0-1).
0 to upgrade, 0 to newly install, 0 to remove and 282 not to upgrade.
osboxes@osboxes:~$ sudo apt install make
Reading package lists... Done
Building dependency tree
Reading state information... Done
make is already the newest version (4.2.1-1.2).
0 to upgrade, 0 to newly install, 0 to remove and 282 not to upgrade.
osboxes@osboxes:~$ sudo apt install ddd
Reading package lists... Done
Building dependency tree
Reading state information... Done
ddd is already the newest version (1:3.3.12-5.2build1).
0 to upgrade, 0 to newly install, 0 to remove and 282 not to upgrade.
osboxes@osboxes:~$
osboxes@osboxes:~$

```

### 4. Клонування файлу з інтернету

За допомогою команди «`git clone [посилання на репозиторій]`» ми скопували файл «sample\_project» за посиланням [https://github.com/davydov-vyacheslav/sample\\_project](https://github.com/davydov-vyacheslav/sample_project)



## 5. Застосування команди «tree»

Щоб вивести структуру каталогів проекту `sample_project` в термінал Linux необхідно ввести такі команди :

```
$ cd sample_project
```

```
$ tree
```

```

osboxes@osboxes:~$
osboxes@osboxes:~$ cd sample_project
osboxes@osboxes:~/sample_project$ tree
.
├── dist
│   ├── main.bin
│   └── test.bin
├── Doxyfile
├── Makefile
├── README.md
├── src
│   ├── lib.c
│   ├── lib.h
│   └── main.c
└── test
    └── test.c

3 directories, 9 files
osboxes@osboxes:~/sample_project$

```

#### 6. Компіляція програми sample\_project

Щоб скомпілювати програму sample\_project через термінал, ми використовуємо команду «make», а саме:

\$ make clean prep compile

```

3 directories, 9 files
osboxes@osboxes:~/sample_project$ make clean prep compile
rm -rf dist
mkdir dist
gcc -std=gnu11 -g -Wall -Wextra -Werror -Wformat-security -Wfloat-equal -Wshadow -Wconversion -Wlogical-not-parentheses -Wnull-dereference -I./src src/lib.c src/main.c -o ./dist/main.bin
gcc -std=gnu11 -g -Wall -Wextra -Werror -Wformat-security -Wfloat-equal -Wshadow -Wconversion -Wlogical-not-parentheses -Wnull-dereference -I./src src/lib.c test/test.c -o ./dist/test.bin
osboxes@osboxes:~/sample_project$

```

#### 7. Виконати скомпільовані файли

За для того, щоб виконати файли та отримати результат їх виконання потрібно в термінал ввести команду «make (run)»

```

osboxes@osboxes:~/sample_project$ make
rm -rf dist
mkdir dist
gcc -std=gnu11 -g -Wall -Wextra -Werror -Wformat-security -Wfloat-equal -Wshadow -Wconversion -Wlogical-not-parentheses -Wnull-dereference -I./src src/lib.c src/main.c -o ./dist/main.bin
gcc -std=gnu11 -g -Wall -Wextra -Werror -Wformat-security -Wfloat-equal -Wshadow -Wconversion -Wlogical-not-parentheses -Wnull-dereference -I./src src/lib.c test/test.c -o ./dist/test.bin
./dist/main.bin
Інформація про тварину №01: Корова: зріст = 123 см, маса = 121 гр.
Інформація про тварину №02: Свиня: зріст = 125 см, маса = 103 гр.
Інформація про тварину №03: Собака: зріст = 87 см, маса = 97 гр.
Інформація про тварину №04: Свиня: зріст = 91 см, маса = 34 гр.
Інформація про тварину №05: Корова: зріст = 83 см, маса = 118 гр.
Інформація про тварину №06: Свиня: зріст = 72 см, маса = 64 гр.
Інформація про тварину №07: Собака: зріст = 5 см, маса = 119 гр.
Інформація про тварину №08: Собака: зріст = 97 см, маса = 29 гр.
Інформація про тварину №09: Собака: зріст = 122 см, маса = 75 гр.
Інформація про тварину №10: Кіт: зріст = 69 см, маса = 62 гр.
osboxes@osboxes:~/sample_project$

```

#### 8. Знаходження нового файлу в проєкті «sample\_project»

Після компілювання, завдяки команді «tree», ми можемо зауважити, що у даному проєкті створилася нова папка «dist» з файлами «main.bin» та «test.bin».

```

Інформація про тварину №10: Кіт: зріст = 69 см, маса = 62 гр.
osboxes@osboxes:~/sample_project$ tree
.
├── dist
│   ├── main.bin
│   └── test.bin
├── Doxyfile
├── Makefile
├── README.md
├── src
│   ├── lib.c
│   ├── lib.h
│   └── main.c
├── test
│   └── test.c
└──
3 directories, 9 files
osboxes@osboxes:~/sample_project$

```

#### 9. Завдати змін у коді

Я хочу змінити назву одну з твари, наприклад замість «свиня» поставити «вовк». Для цього я відкрив файл

lib.c : у 25-ій строчці замість «PIG» → «WOLF», та замість «Свиня» → «Вовк».

lib.h : у 28-ій строчці замість «PIG» → «WOLF», «Свиня» → «Вовк».

test.c : у 26-ій строчці замість «PIG» → «WOLF».

#### 10. Продемонструвати зміни

Далі в терміналі прописуємо «\$ make clean prep compile», після цього пишемо команду «\$ make (run)».

```
osboxes@osboxes:~/sample_project$ make clean prep compile
rm -rf dist
mkdir dist
gcc -std=gnu11 -g -Wall -Wextra -Werror -Wformat-security -Wfloat-equal -Wshadow -Wconversion
-Wlogical-not-parentheses -Wnull-dereference -I./src src/lib.c src/main.c -o ./dist/main.bin
gcc -std=gnu11 -g -Wall -Wextra -Werror -Wformat-security -Wfloat-equal -Wshadow -Wconversion
-Wlogical-not-parentheses -Wnull-dereference -I./src src/lib.c test/test.c -o ./dist/test.bin
osboxes@osboxes:~/sample_project$ make
rm -rf dist
mkdir dist
gcc -std=gnu11 -g -Wall -Wextra -Werror -Wformat-security -Wfloat-equal -Wshadow -Wconversion
-Wlogical-not-parentheses -Wnull-dereference -I./src src/lib.c src/main.c -o ./dist/main.bin
gcc -std=gnu11 -g -Wall -Wextra -Werror -Wformat-security -Wfloat-equal -Wshadow -Wconversion
-Wlogical-not-parentheses -Wnull-dereference -I./src src/lib.c test/test.c -o ./dist/test.bin
./dist/main.bin
Інформація про тварину №01: Собака: зріст = 102 см, маса = 76 гр.
Інформація про тварину №02: Вовк: зріст = 90 см, маса = 33 гр.
Інформація про тварину №03: Вовк: зріст = 31 см, маса = 112 гр.
Інформація про тварину №04: Корова: зріст = 88 см, маса = 33 гр.
Інформація про тварину №05: Кіт: зріст = 7 см, маса = 63 гр.
Інформація про тварину №06: Корова: зріст = 106 см, маса = 125 гр.
Інформація про тварину №07: Вовк: зріст = 123 см, маса = 109 гр.
Інформація про тварину №08: Вовк: зріст = 15 см, маса = 103 гр.
Інформація про тварину №09: Кіт: зріст = 3 см, маса = 102 гр.
Інформація про тварину №10: Кіт: зріст = 25 см, маса = 117 гр.
osboxes@osboxes:~/sample_project$
```

11.

```
osboxes@osboxes:~/sample_project$ git diff
diff --git a/src/lib.c b/src/lib.c
index 56582cf..5baed65 100644
--- a/src/lib.c
+++ b/src/lib.c
@@ -22,8 +22,8 @@ char *get_animal_type_name(enum animal_type type)
     case COW:
         result = "Корова";
         break;
-    case PIG:
-        result = "Свиня";
+    case WOLF:
+        result = "Вовк";
         break;
     default:
         result = "N/A";
diff --git a/src/lib.h b/src/lib.h
index 2d3728e..d3a1f8e 100644
--- a/src/lib.h
+++ b/src/lib.h
@@ -25,7 +25,7 @@
 * Тип тварини
 */
enum animal_type {
-    PIG, /**< Свиня */
+    WOLF, /**< Вовк */
    COW, /**< Корова */
    DOG, /**< Собака */
}; ...skipping...
diff --git a/src/lib.c b/src/lib.c
index 56582cf..5baed65 100644
--- a/src/lib.c
+++ b/src/lib.c
```

```

@@ -22,8 +22,8 @@ char *get_animal_type_name(enum animal_type type)
     case COW:
         result = "Корова";
         break;
+     case PIG:
+         result = "Свиня";
+     case WOLF:
+         result = "Вовк";
         break;
     default:
         result = "N/A";
diff --git a/src/lib.h b/src/lib.h
index 2d3728e..d3a1f8e 100644
--- a/src/lib.h
+++ b/src/lib.h
@@ -25,7 +25,7 @@
 * Тип тварини
 */
enum animal_type {
-    PIG, /**< Свиня */
+    WOLF, /**< Вовк */
+    COW, /**< Корова */
    DOG, /**< Собака */
    CAT, /**< Кіт */
diff --git a/test/test.c b/test/test.c
index b8d80bf..a2f7f84 100644
--- a/test/test.c
+++ b/test/test.c
@@ -23,7 +23,7 @@ unsigned short test_get_animal_by_name()
#define DATA_SIZE 6

    unsigned short is_success = 1;
-    enum animal_type input_data[] = { CAT, DOG, COW, PIG, ANIMAL_TYPE_COUNT, ANIMAL_TYPE_COUNT + 1 };
+    enum animal_type input_data[] = { CAT, DOG, COW, WOLF, ANIMAL_TYPE_COUNT, ANIMAL_TYPE_COUNT + 1 };
    char *expected_values[] = { "Кіт", "Собака", "Корова",
                                "Свиня", "N/A", "N/A" };

```

#### Контрольні запитання:

1. Операційна система – сукупність програм, завдяки яким можливе керування ресурсами комп'ютера, обчислювальними процесами та використання апаратури користувачем. Основні ОС: Windows, macOS, Linux.
2. Лідерами в сфері виробництва засобів віртуалізації є компанії VMware, Microsoft, SWSoft, XenSource, VirtualBox і InnitEK.
3. Менеджер паків – це програма, яка слідкує за встановленими програмами, дозволяє легко встановлювати нові та легко оновлювати старі програми завдяки набору файлів. Є такі менеджери пакетів як apt, Synaptic та існує встроєний в ОС менеджер пакетів.
4. Система контролю версіями – це система, яка записує зміни, які відбулися у файлі або в декількох файлах протягом деякого часу так, щоб ви змогли відкотитися до попередніх версій цього файлу (файлів). Існують локальні системи контролю версій, централізовані та децентралізовані системи контролю версій.
5. Makefile – це файл, в якому заздалегідь прописані інструкції для команди «make». Тобто це спосіб автоматизації процедури створення програмного забезпечення. Щоб використати Makefile



необхідно, встановити утиліту «make» в термінали перейти в директорію з файлом Makefile та завдяки команді «make (clean) (prep) (compile) (run)» ми можемо видалити папку «dist», якщо вона є, додати папку «dist», скомпілювати проект в папку «dist» та запустити проект, відповідно.

6. Команда «tree» робить деревоподібну структуру всіх каталогів, які є в директорії в якій ви знаходитесь.
7. Потрібно в терміналі перейти в директорію з зміненими файлами та ввести команду «git diff».