

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

## 1 Вимоги

### 1.1 Розробник

- Висоцький Олексій Васильович;
- студент групи КІТ-320;
- 18-жовт-2020.

### 1.2 Загальне завдання

Установити середовище для подальшої роботи з предмету “Програмування”.

## Хід роботи

1. Знаходимо, завантажуюмо та інсталуємо додаток VirtualBox, що дозволить встановити операційну систему Linux та працювати з нею.

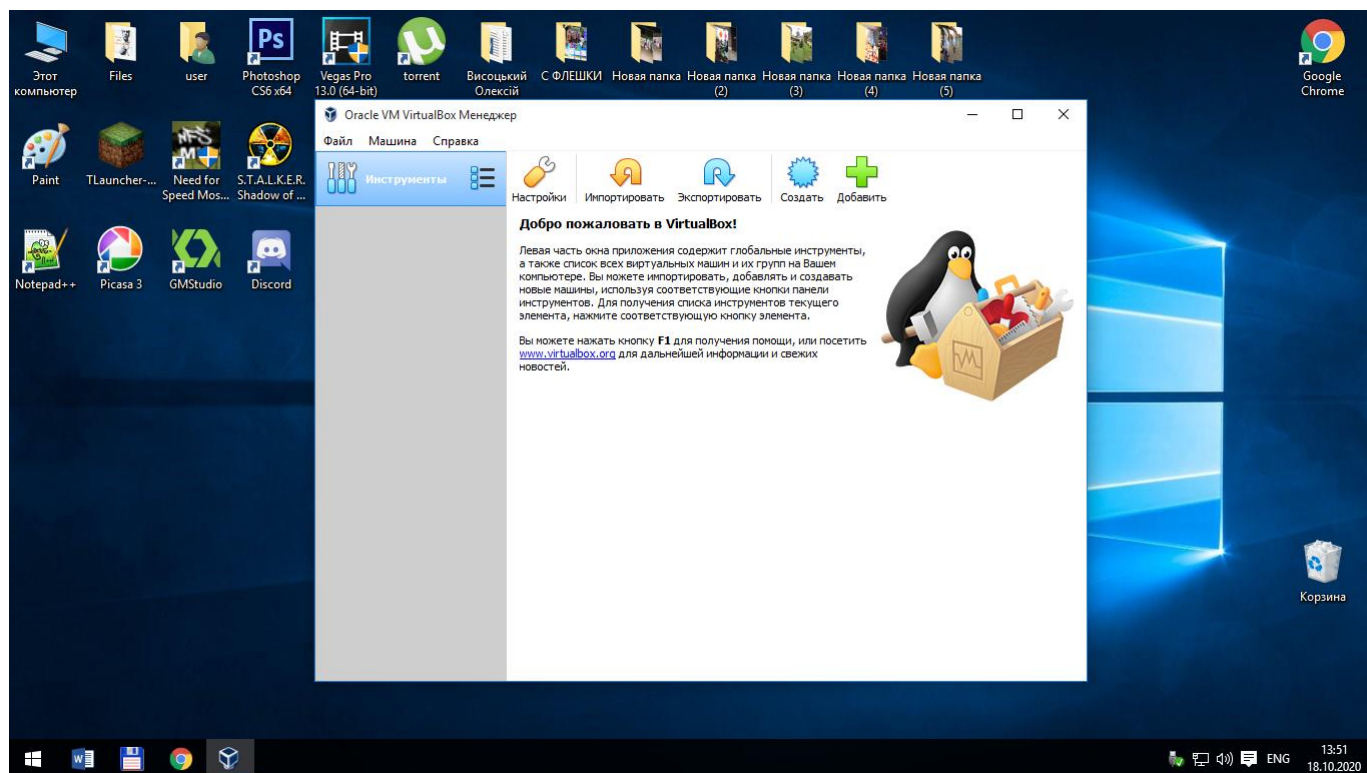


Рисунок 1 – Встановлений VirtualBox

2. Знаходимо, завантажуюмо та встановлюємо образ Linux системи для подальшої роботи.

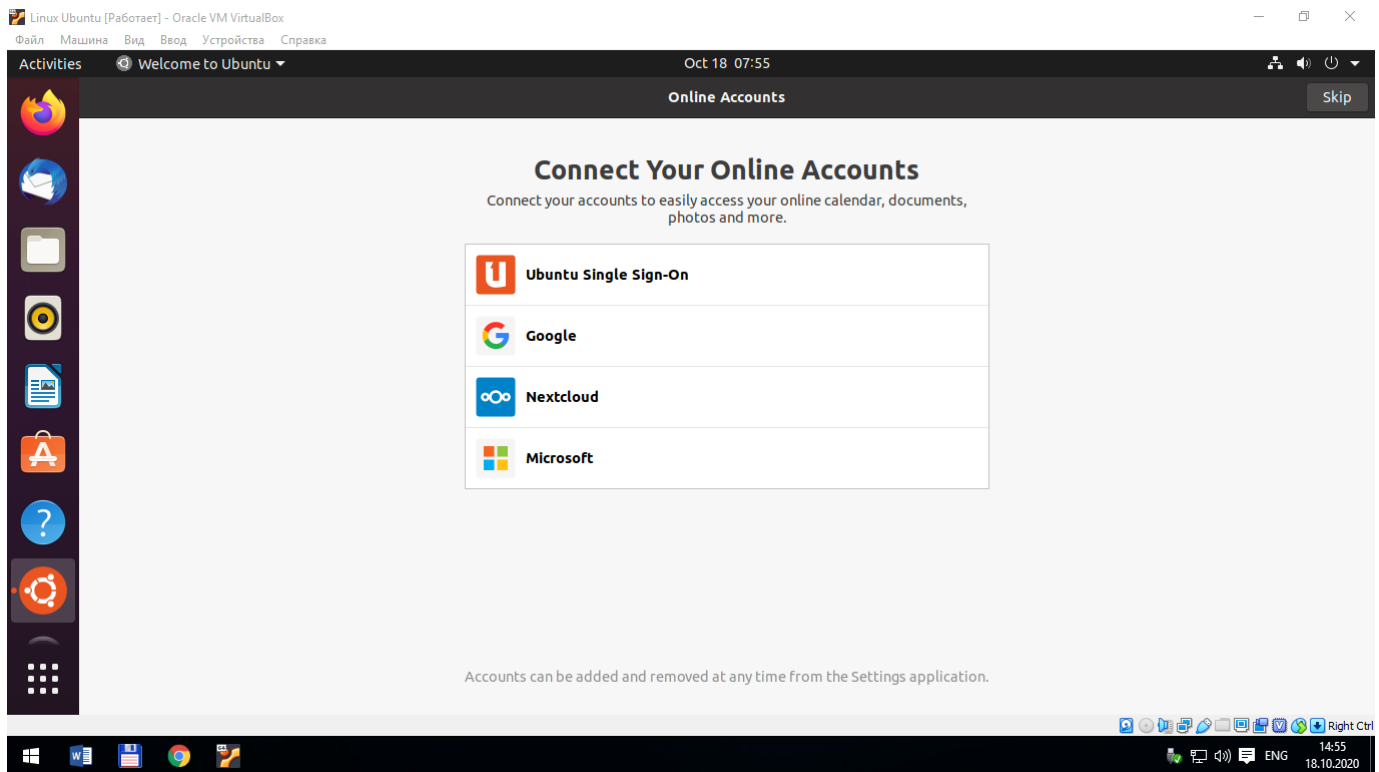


Рисунок 2 - Встановлений Linux

3. Інсталюємо мінімально-необхідні пакети (програмні додатки): *git*, *gcc*, *tree*, *make*, *ddd*. Для інсталяції пакетів використаємо пакетний менеджер *apt-get*.
4. За допомогою системи контролю версіями *git* та команди *git clone* виконуємо клонування проекту (репозиторія), що знаходиться в інтернеті за посиланням [https://github.com/davydov-vyacheslav/sample\\_project](https://github.com/davydov-vyacheslav/sample_project).

```
osboxes@osboxes:~$ git clone https://github.com/davydov-vyacheslav/sample_project
Cloning into 'sample_project'...
remote: Enumerating objects: 43, done.
remote: Counting objects: 100% (43/43), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 43 (delta 14), reused 29 (delta 10), pack-reused 0
Unpacking objects: 100% (43/43), 13.74 KiB | 182.00 KiB/s, done.
osboxes@osboxes:~$
```

Рисунок 4 – Клонування проекту (репозиторія)

5. Ознайомлюємося з утилітою *tree*. Заходимо в директорію сконованого проекту та завдяки утиліті *tree* виводимо на екран структуру каталогів проекту.

```

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

2 directories, 7 files
osboxes@osboxes:~/sample_project$

```

Рисунок 5 – Структура каталогів проекту

6. Компілюємо проект командою *make clean prep compile*.

```

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.bi
n
osboxes@osboxes:~/sample_project$

```

Рисунок 6 – Компіляція проекту

7. Компіляція проекту, що була виконана в попередньому пункті, створює виконуючі файли, які можна запустити та побачити результат роботи проекту. Знаходимо ці файли та виконуємо їх.

```

osboxes@osboxes:~/sample_project$ ./dist/main.bin
Інформація про тварину №01: Свиня: зріст = 83 см, маса = 17 гр.
Інформація про тварину №02: Свиня: зріст = 85 см, маса = 45 гр.
Інформація про тварину №03: Корова: зріст = 76 см, маса = 122 гр.
Інформація про тварину №04: Собака: зріст = 100 см, маса = 122 гр.
Інформація про тварину №05: Корова: зріст = 32 см, маса = 107 гр.
Інформація про тварину №06: Корова: зріст = 2 см, маса = 50 гр.
Інформація про тварину №07: Корова: зріст = 120 см, маса = 53 гр.
Інформація про тварину №08: Кіт: зріст = 13 см, маса = 123 гр.
Інформація про тварину №09: Корова: зріст = 11 см, маса = 72 гр.
Інформація про тварину №10: Корова: зріст = 66 см, маса = 111 гр.
osboxes@osboxes:~/sample_project$ ./dist/test.bin
Running test test_get_animal_by_name ...
Running test test_generate_animal ...
Congratulations! All tests succeed!
osboxes@osboxes:~/sample_project$

```

Рисунок 7 – Результат виконання створених файлів

8. За допомогою утиліти `tree` виводимо на екран структуру каталогів файлів проекту після компіляції. Бачимо, що при зборці проекту був доданий каталог `dist` та два виконуючі файли – `main.bin` та `test.bin`.

```
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$
```

Рисунок 8 – Структура каталогів проекту після компіляції

9. Виконуємо зміни коду при умові, що:
- 1) наступна компіляція проекту буде виконана без помилок;
  - 2) зміни можна побачити у даних, що виводяться на екран проектом.
10. Перекомпілюємо проект командою `make clean prep compile` та продемонструємо зміни, що помітні при виведенні на екран.

```
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$ ./dist/main.bin
Інформація про тварину №01: Кіт: зріст = 42 см, маса = 85 гр.
Інформація про тварину №02: Людина: зріст = 53 см, маса = 41 гр.
Інформація про тварину №03: Корова: зріст = 104 см, маса = 59 гр.
Інформація про тварину №04: Кіт: зріст = 21 см, маса = 46 гр.
Інформація про тварину №05: Свиня: зріст = 56 см, маса = 117 гр.
Інформація про тварину №06: Людина: зріст = 53 см, маса = 109 гр.
Інформація про тварину №07: Корова: зріст = 83 см, маса = 97 гр.
Інформація про тварину №08: Свиня: зріст = 57 см, маса = 113 гр.
Інформація про тварину №09: Кіт: зріст = 112 см, маса = 114 гр.
Інформація про тварину №10: Кіт: зріст = 80 см, маса = 3 гр.
Інформація про тварину №11: Свиня: зріст = 98 см, маса = 45 гр.
Інформація про тварину №12: Кіт: зріст = 26 см, маса = 90 гр.
Інформація про тварину №13: Свиня: зріст = 38 см, маса = 68 гр.
Інформація про тварину №14: Людина: зріст = 7 см, маса = 89 гр.
Інформація про тварину №15: Собака: зріст = 70 см, маса = 10 гр.
osboxes@osboxes:~/sample_project$
```

Рисунок 10 – Перекомпіляція проекту

11. За допомогою команди `git diff` покажемо виконані зміни у файлах.

```
osboxes@osboxes:~/sample_project$ git diff
diff --git a/src/lib.c b/src/lib.c
index 56582cf..0178b65 100644
--- a/src/lib.c
+++ b/src/lib.c
@@ -25,6 +25,9 @@ char *get_animal_type_name(enum animal_type type)
     case PIG:
         result = "Свиня";
         break;
+    case HUMAN:
+        result = "Людина";
+        break;
     default:
         result = "N/A";
 }
diff --git a/src/lib.h b/src/lib.h
index 2d3728e..92e0380 100644
--- a/src/lib.h
+++ b/src/lib.h
@@ -19,7 +19,7 @@
 /**
  * Кількість тварин у масиві
  */
-#define ANIMAL_COUNT 10
+#define ANIMAL_COUNT 15
```

Рисунок 11 – Виконані зміни коду у файлах

## Висновки

При виконанні даної лабораторної роботи було набуто практичного досвіду роботи з командною строкою Linux, програмними додатками `git`, `tree`, `make`; освоєно інтерфейс системи Linux.