Лабораторна робота 1. Вступ до програмування. Освоєння командної строки Linux.

Виконав: Гладков Костянтин Сергійович, студент ХПІ, КІТ-320

Завдання:

1) Використовував ВіртуалБокс

2) Було встановлено Убунту (gnome)

3) комманди для установки репозиторіїв - sudo apt-get install (name) (sudo - superuser)

name - ім'я сховища, який встановлюємо. В даному випадку це git, make, ddd, tree, gcc

(Буде потрібно після комманди ввести пароль вiд superuser)

4) для клонування з гітхаба потрібно використовувати комманду git clone (link)

5) tree - комманда для перегляду директорії та їх вмісту

6) makefile знаходиться в директорії sample\_project, куди можна перейти коммандой cd sample\_project, а далі використовувати дану нам комманду make clean prep compile

7) виконав компіляцію, побачив зміни і зробив висновки

8) до компіляції:

├── sample\_project

│   ├── Doxyfile

│   ├── Makefile

│   ├── README.md

│   ├── src

│   │   ├── lib.c

│   │   ├── lib.h

│   │   └── main.c

│   └── test

│   └── test.c

после компиляции:

├── dist

│   ├── main.bin

│   └── test.bin

├── Doxyfile

├── Makefile

├── README.md

├── src

│   ├── lib.c

│   ├── lib.h

│   └── main.c

└── test

└── test.c

9) в test.c я змінив "Congratulations! All tests succeed! \ N" на "Congratulations man! All tests succeed! \ N"

10) просто ввести комманду make clean prep compile після чого запустити test.c і ось різниця (оригінальний варіант і після змінений):

Running test test\_get\_animal\_by\_name ...

Running test test\_generate\_animal ...

Congratulations! All tests succeed!

Running test test\_get\_animal\_by\_name ...

Running test test\_generate\_animal ...

Congratulations man! All tests succeed!

повний код даного фрагмента:

int main ()

{

srand ((unsigned int) time (0));

unsigned short is\_success = 1;

is\_success & = test\_get\_animal\_by\_name ();

is\_success & = test\_generate\_animal ();

if (is\_success) {

printf ( "Congratulations man! All tests succeed! \ n");

} Else {

printf ( "Some tests failed. Check logs, fix errors and try again \ n");

}

return! is\_success;

}

10) просто ввести комманду make clean prep compile після чого запустити test.c і ось різниця (оригінальний варіант і після змінений):

Running test test\_get\_animal\_by\_name ...

Running test test\_generate\_animal ...

Congratulations! All tests succeed!

Running test test\_get\_animal\_by\_name ...

Running test test\_generate\_animal ...

Congratulations man! All tests succeed!

повний код даного фрагмента:

int main ()

{

srand ((unsigned int) time (0));

unsigned short is\_success = 1;

is\_success & = test\_get\_animal\_by\_name ();

is\_success & = test\_generate\_animal ();

if (is\_success) {

printf ( "Congratulations man! All tests succeed! \ n");

} Else {

printf ( "Some tests failed. Check logs, fix errors and try again \ n");

}

return! is\_success;

}

11)

diff --git a / test / test.c b / test / test.c

index b8d80bf..5ad80fc 100644

--- a / test / test.c

+++ b / test / test.c

@@ -25,7 +25,7 @@ unsigned short test\_get\_animal\_by\_name ()

unsigned short is\_success = 1;

enum animal\_type input\_data [] = {CAT, DOG, COW, PIG, ANIMAL\_TYPE\_COUNT, ANIMAL\_TYPE\_COUNT + 1};

char \* expected\_values ​​[] = { "Кіт", "Собака", "Корова",

- "Свиня", "N / A", "N / A"};

+ "Свиня", "N / A", "N / A"};

for (int i = 0; i < DATA\_SIZE; i++) {

char \*actual\_value = get\_animal\_type\_name(input\_data[i]);

@@ -102,9 +102,10 @@ int main()

is\_success &= test\_generate\_animal();

if (is\_success) {

- printf("Congratulations! All tests succeed!\n");

+ printf("Congratulations man! All tests succeed!\n");

} else {

printf("Some tests failed. Check logs, fix errors and try again\n");

}

return !is\_success;

}

+

12) документація була зроблена в цьому файлі

Висновки: Лінукс багато в чому залежить від терміналу (консолі). З її допомогою можна встановити пакети, репозиторії, а також дані з інтернету (git clone). Під час використання Убунту, я освоївся в її середовищі (гном)

також я вивчив комманди cd (переміщення по системі), nano (текстовий редактор), git (для установки програм) , apt-get install (установник пакетів) та іншi. Завдяки компіляції коду, кід видозмінився, видавши нові файли, які можна відкрити і запустити. Кінцевий результат

може бути змінений, як я і зробив, помінявши кінцеву видачу коду у файлі test.c. Також я зміг за допомогою команди дізнатися які саме зміни були виконані.