**Київський національний університет імені Тараса Шевченка**

**факультет радіофізики, електроніки та комп’ютерних систем**

Лабораторна робота № 4

Варіант 4

Роботу виконав

студент 3 курсу

КІ-СА

Бондаренко Владислав

Київ 2020

1. **Підготовка середовища розробки**

Для виконання лабораторної роботи вам знадобиться комп'ютер (віртуальний або фізичний) архітектури AMD64/EM64T із встановленим дистрибутивом ОС Linux (будь-яким).

На систему необхідно встановити GCC, GDB, GNU Make та GNU Binutils.

Створіть окремий каталог, який будете використовувати для виконання лабораторної роботи.

Завантажте в нього файл із символами та програму-заготовку:

defs.h

exit.s





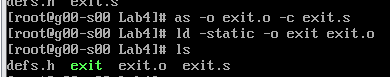
****

Виконайте асемблювання програми-заготовки та зв'язування:

as -o exit.o -c exit.s

ld -static -o exit exit.o

Пересвідчіться у тому, що виконуваний файл працездатний. Програма повинна нічого не робити і не виводити жодних помилок.





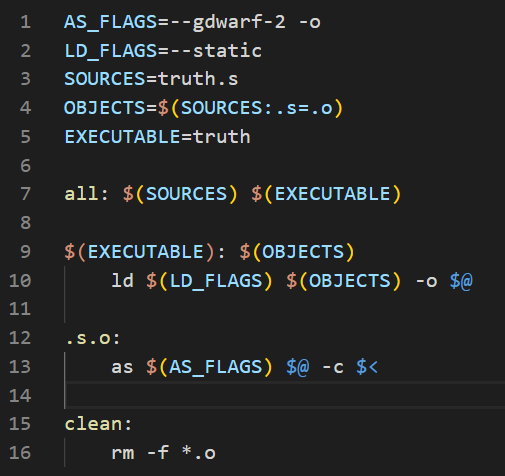
1. **Автоматизація збірки**

Створіть Makefile, який за командою make exit та make all виконає збірку, а за командою make clean очистить об'єктні та виконувані файли.

Модифікуйте Makefile так, щоб опції асемблера та лінкера задавалися змінними ASFLAGS та LDFLAGS.

Додайте опцію асемблера для генерації відлагоджувальних символів DWARF.

Використайте шаблонні правила так, щоб можна було збирати декілька асемблерних файлів в окремі виконувані файли. Це знадобиться при виконанні індивідуального завдання.

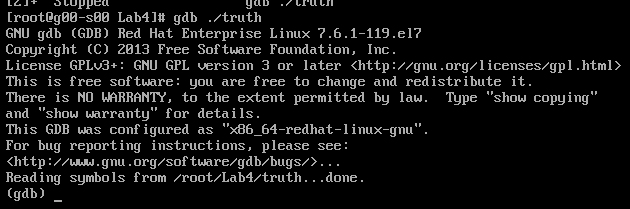




1. **Навички відлагоджування**

Завантажте одержаний виконуваний файл у відлагоджувач за допомогою команди:

gdb ./exit



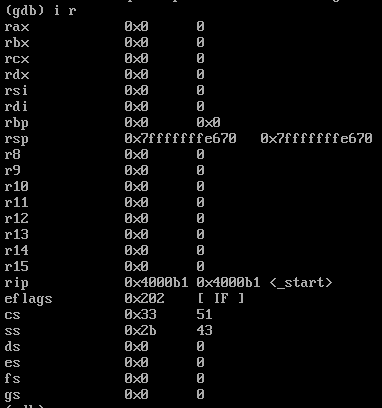
Встановіть точку зупинки на початок програми (мітка \_start):



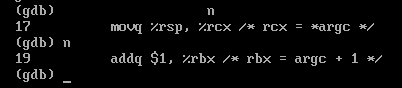
Запустіть програму.



Після зупинки виконання програми перегляньте вміст регістрів:



Переходьте до виконання наступної команди:



Для виходу із режиму покрокового виконання використовуйте команду

continue або с

Програма працюватиме до наступної точки зупинки або до повного чи аварійного завершення.

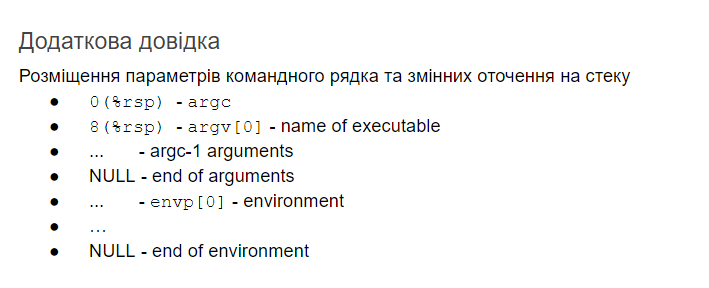


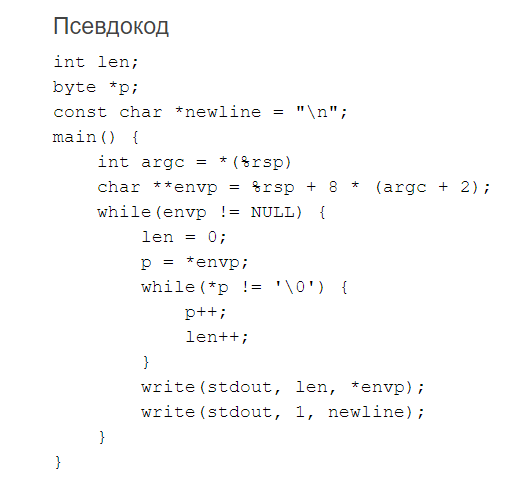
1. **Індивідуальні завдання**

Створіть програму, яка виводить вміст змінних оточення власного процесу на стандартний потік виведення.

****

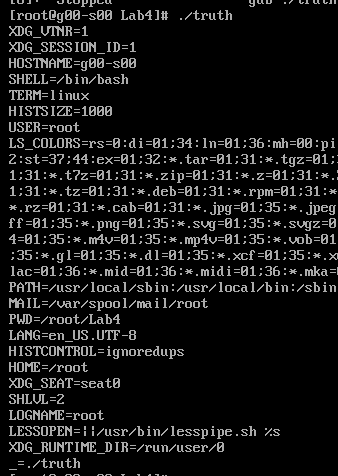
****

****

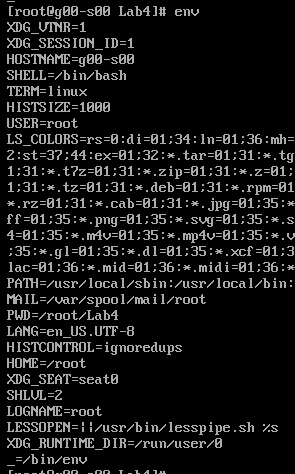
****

Код програми доступний на [github](https://github.com/Lrazerz/CompSystemsLabs/tree/master/Lab4).

Результат виконання програми:



Результат виконання стандартної команди для відображення змінних оточення **env**:



**Висновки**

В даній лабораторній роботі було ознайомлено з архітектурою x86-64, зокрема з вказівником команд RIP, флаговим регістром RFLAGS та цілочисельними регістрами загального призначення (RSP, RBP, RAX,…)

Також під час роботи було ознайомлено та отримано практичні навички роботи з GNU асемблером, який використовує AT&T синтаксис.

Було виконано автоматизацію збірки за допомогою make-файла, проведено відлагоджування коду з використанням GNU дебагера.