

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5

дисциплина: *Архитектура компьютера*

Студент: Ужаков Магомед

Группа: НПИбд-02-22

МОСКВА

2022 г.

Цель работы:

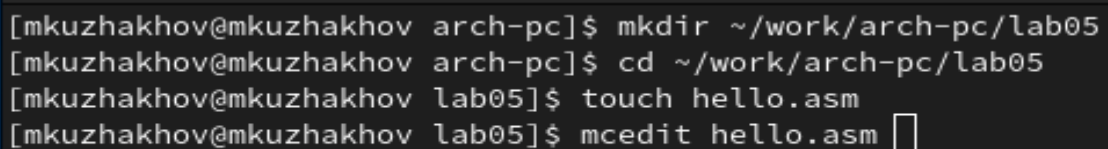
Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

Порядок выполнения лабораторной работы:

1. Программа Hello world!

Рассмотрим пример простой программы на языке ассемблера NASM. Традиционно первая программа выводит приветственное сообщение Hello world! на экран.

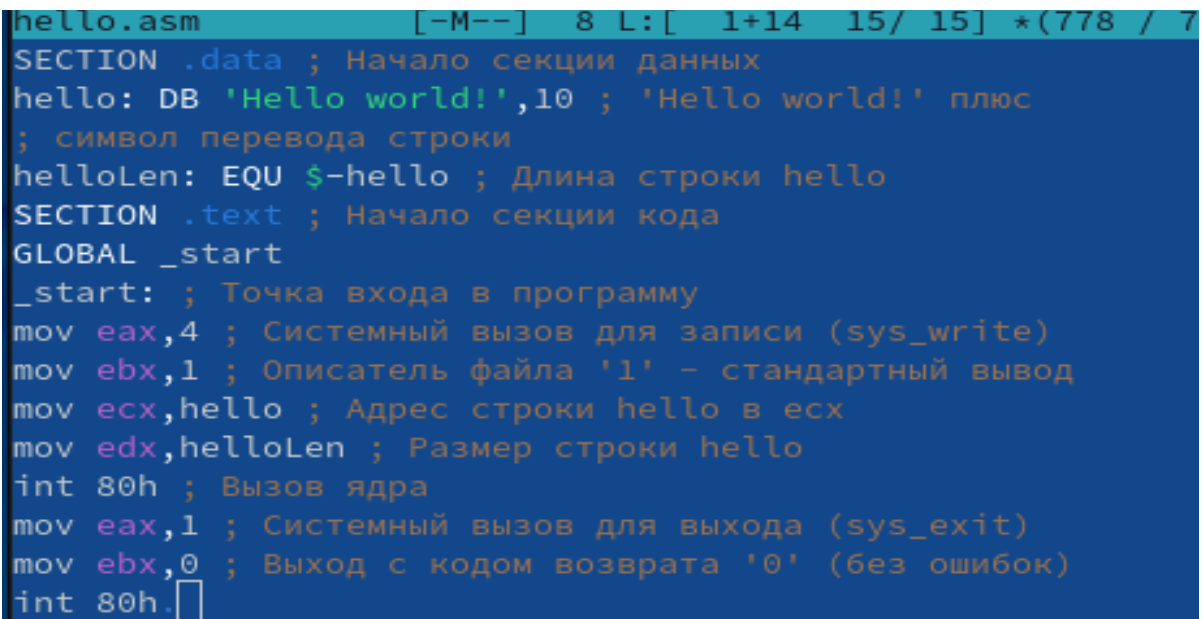
Создадим каталог для работы с программами на языке ассемблера NASM, перейдем в него, создадим текстовый файл с именем hello.asm и откроем его (рис. 1).



```
[mkuzhakhov@mkuzhakhov arch-pc]$ mkdir ~/work/arch-pc/lab05
[mkuzhakhov@mkuzhakhov arch-pc]$ cd ~/work/arch-pc/lab05
[mkuzhakhov@mkuzhakhov lab05]$ touch hello.asm
[mkuzhakhov@mkuzhakhov lab05]$ mcedit hello.asm
```

Рис. 1. Создание файла hello.asm

Введём в него следующий текст (рис. 2).



```
hello.asm [-M--] 8 L:[ 1+14 15/ 15] *(778 / 7
SECTION .data ; Начало секции данных
hello: DB 'Hello world!',10 ; 'Hello world!' плюс
; символ перевода строки
helloLen: EQU $-hello ; Длина строки hello
SECTION .text ; Начало секции кода
GLOBAL _start
_start: ; Точка входа в программу
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла '1' – стандартный вывод
mov ecx,hello ; Адрес строки hello в ecx
mov edx,helloLen ; Размер строки hello
int 80h ; Вызов ядра
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
int 80h.
```

Рис. 2. Код программы hello

2. Транслятор NASM.

Затем скомпилируем программу Hello world! (рис. 3).

```
[mkuzhakhov@mkuzhakhov lab05]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
[mkuzhakhov@mkuzhakhov lab05]$ ls
hello.asm list.lst obj.o
[mkuzhakhov@mkuzhakhov lab05]$
```

Рис. 3. Компиляция программы

Как видим, образовался объектный файл, значит компиляция прошла успешно.

3. Расширенный синтаксис командной строки NASM.

Полный вариант командной строки nasm выглядит следующим образом

Выполним следующую команду, а затем проверим, что файлы были созданы (рис. 4).

```
[mkuzhakhov@mkuzhakhov lab05]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
[mkuzhakhov@mkuzhakhov lab05]$ ls
hello.asm list.lst obj.o
[mkuzhakhov@mkuzhakhov lab05]$
```

Рис. 4. Команда nasm

Данная команда скомпилирует исходный файл hello.asm в obj.o (опция -o позволяет задать имя объектного файла, в данном случае obj.o), при этом формат выходного файла будет elf, и в него будут включены символы для отладки (опция -g), кроме того, будет создан файл листинга list.lst (опция -l).

4. компоновщик LD.

Чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику, затем проверим, что исполняемый файл был создан (рис. 5).

```
[mkuzhakhov@mkuzhakhov lab05]$ ld -m elf_i386 obj.o -o hello
[mkuzhakhov@mkuzhakhov lab05]$ ld -m elf_i386 obj.o -o main
[mkuzhakhov@mkuzhakhov lab05]$ ls
hello hello.asm list.lst main obj.o
[mkuzhakhov@mkuzhakhov lab05]$
```

Рис. 5. Исполняемый файл

Затем запустим созданный исполняемый файл с помощью следующей команды (рис. 6).

```

[mkuzhakhov@mkuzhakhov lab05]$ ./hello
Hello world!
[mkuzhakhov@mkuzhakhov lab05]$ ./main
Hello world!
[mkuzhakhov@mkuzhakhov lab05]$
[mkuzhakhov@mkuzhakhov lab05]$

```

Рис. 6. Запуск программы

Как видим, все сработало корректно.

Порядок выполнения самостоятельной работы:

В том же каталоге создадим копию файла hello.asm с именем lab5.asm и внесем в него изменения, чтобы программа выводила на экран мои фамилию и имя.

```

lab5.asm      [-M--] 21 L:[ 1+ 1  2/ 15] *(7
SECTION .data ; Начало секции данных
hello: DB 'mkuzhakhov',10 ; 'Hello world!' плюс
; символ перевода строки
helloLen: EQU $-hello ; Длина строки hello
SECTION .text ; Начало секции кода
GLOBAL _start
_start: ; Точка входа в программу
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла '1' - стандартный вывод
mov ecx,hello ; Адрес строки hello в ecx
mov edx,helloLen ; Размер строки hello
int 80h ; Вызов ядра
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
int 80h.

```

Рис. 7. Изменения в программе

Затем оттранслируем полученный текст программы в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл (рис. 11).

```
[mkuzhakhov@mkuzhakhov lab05]$ cp hello.asm lab5.asm
[mkuzhakhov@mkuzhakhov lab05]$ mcedit lab5.asm

[mkuzhakhov@mkuzhakhov lab05]$ nasm -o lab5.o -f elf -g -l lab5.asm
lab5.o: fatal: no input file specified
Type nasm -h for help.
[mkuzhakhov@mkuzhakhov lab05]$ nasm -o lab5.o -f elf -g lab5.asm
[mkuzhakhov@mkuzhakhov lab05]$ ld -m elf_i386 lab5.o -o lab5
[mkuzhakhov@mkuzhakhov lab05]$ ./lab5
mkuzhakhov
[mkuzhakhov@mkuzhakhov lab05]$
```

Рис. 8. Программа lab5

Как видим, все работает.

Вывод:

Во время лабораторной работы были освоены процедуры компиляции и сборки программ, написанных на ассемблере NASM.