

Отчёт по лабораторной работе №4
Простейший вариант

Метвалли Ахмед Фарг Набеев

Содержание

1. Цель работы
2. Выполнение лабораторной работы
- 3.Задание для самостоятельной работы
4. Выводы

Список таблиц

1.Цель работы

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

2.Выполнение лабораторной работы

Откройте терминал

Создайте каталог для работы с программами на языке ассемблера NASM:

```
ahmad-farg@ahmadfarg-VirtualBox:~$ mkdir -p ~/work/arch-pc/lab04  
ahmad-farg@ahmadfarg-VirtualBox:~$
```

Рис. 2.1: Создание каталога.

Перейдите в созданный каталог.

```
ahmad-farg@ahmadfarg-VirtualBox:~$ cd ~/work/arch-pc/lab04
```

Рис. 2.2: Переход в созданный каталог.

Создайте текстовый файл с именем hello.asm.

```
ahmad-farg@ahmadfarg-VirtualBox:~$ mkdir -p ~/work/arch-pc/lab04  
ahmad-farg@ahmadfarg-VirtualBox:~$ cd ~/work/arch-pc/lab04  
ahmad-farg@ahmadfarg-VirtualBox:~/work/arch-pc/lab04$ touch hello.asm  
ahmad-farg@ahmadfarg-VirtualBox:~/work/arch-pc/lab04$
```

Рис. 2.3: Создание файла hello.asm Откройте файл hello.asm с помощью любого текстового редактора, например, gedit.

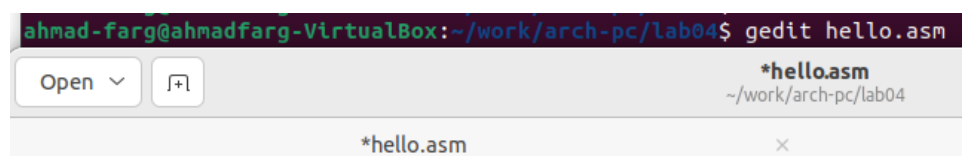


Рис. 2.4: Открытие файла hello.asm

Вводим в файл hello.asm нужный текст.

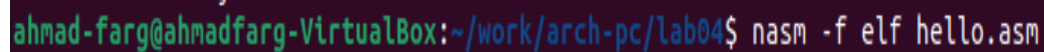
```
*hello.asm
1 SECTION .data
2 hello:db "Hello, world!",0xa
3 hellolen: equ $ - hello
4 SECTION text
5 global _start
6 _start:
7 mov eax, 4
8 mov ebx, 1
9 mov ecx, hello
10 mov edx, hellolen
11 int 0x80
12 mov eax, 1
13 mov ebx, 0
14 int 0x80
15 |
```

```
hello.asm
1 SECTION .data
2     hello: DB 'Hello world!',10
3     hellolen: EQU $- hello
4 SECTION .text
5     GLOBAL _start
6 _start:
7     mov eax,4
8     mov ebx,1
9     mov ecx,hello
10    mov edx,hellolen
11    int 80h
12    mov eax,1
13    mov ebx,0
14    int 80h
```

Рис. 2.5: Ввод текста.

Теперь напишем:

```
nasm -f elf hello.asm
```



```
ahmad-farg@ahmadfarg-VirtualBox:~/work/arch-pc/lab04$ nasm -f elf hello.asm
```

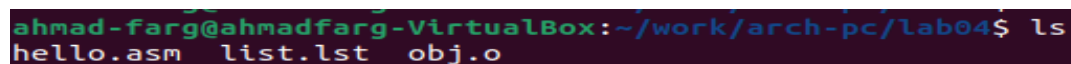
Рис. 2.6: Ввод `nasm -f elf hello.asm`

`nasm -f elf hello.asm` превращает текст программы в объектный код.

Выполните команду полной строки `nasm`.



```
ahmad-farg@ahmadfarg-VirtualBox:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst hello.asm
```

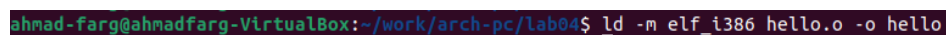


```
ahmad-farg@ahmadfarg-VirtualBox:~/work/arch-pc/lab04$ ls  
hello.asm list.lst obj.o
```

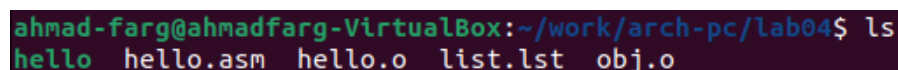
Рис. 2.7: Открытие файла

Данная команда скомпилировала исходный файл `hello.asm` в `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`). С помощью команды `ls` проверили, что файлы были созданы.

Чтобы получить исполняемую команду необходимо передать на обработку компоновщику.



```
ahmad-farg@ahmadfarg-VirtualBox:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
```

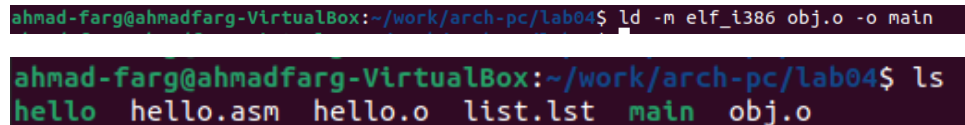


```
ahmad-farg@ahmadfarg-VirtualBox:~/work/arch-pc/lab04$ ls  
hello hello.asm hello.o list.lst obj.o
```

Рис. 2.8: Создание исполняемого файла hello.

Ключ -o с последующим значением задаёт в данном случае имя создаваемого исполняемого файла.

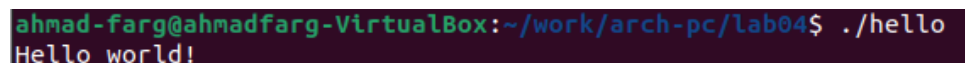
Выполните следующую команду:



```
ahmad-farg@ahmadfarg-VirtualBox:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
ahmad-farg@ahmadfarg-VirtualBox:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
```

Рис. 2.9: Выполнение команды `ld -m elf_i386 obj.o -o main`.

Запуск программы созданного исполняемого файла.



```
ahmad-farg@ahmadfarg-VirtualBox:~/work/arch-pc/lab04$ ./hello
Hello world!
```

Рис. 2.10: Запуск файла.

3. Задание для самостоятельной работы

В каталоге `~/work/arch-pc/lab04` с помощью команды `cp` создаем копию файла `hello.asm` с именем `lab4.asm`.



```
farg@ahmadfarg-VirtualBox:~/work/arch-pc/lab04$ cp hello.asm lab.asm
```

Рис. 3.1: Создание файла `lab4.asm`.

Внесите изменения в текст программы в файле lab4.asm так, чтобы вместо Hello world! на экран выводилась строка с вашими фамилией и именем.

```
1 SECTION .data
2     hello: DB 'ахмед метвалли',10
3     hellolen: EQU $- hello
4 SECTION .text
5     GLOBAL _start
6 _start:
7     mov eax,4
8     mov ebx,1
9     mov ecx,hello
10    mov edx,hellolen
11    int 80h
12    mov eax,1
13    mov ebx,0
14    int 80h
```

Рис. 3.2: Внесение изменений.

Оттранслируйте полученный текст программы lab4.asm в объектный файл.

Выполните компоновку объектного файла и запустите получившийся исполняемый файл. Рис.

```
ahmad-farg@ahmadfarg-VirtualBox:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
ahmad-farg@ahmadfarg-VirtualBox:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst lab4.asm
ahmad-farg@ahmadfarg-VirtualBox:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
ahmad-farg@ahmadfarg-VirtualBox:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o hello
ahmad-farg@ahmadfarg-VirtualBox:~/work/arch-pc/lab04$ ./hello
ахмед метвалли
```

3.3: Оттранслирование полученного текста программы lab4.asm в объектный файл.

```

ahmad-farg@ahmadfarg-VirtualBox:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
ahmad-farg@ahmadfarg-VirtualBox:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst lab4.asm
ahmad-farg@ahmadfarg-VirtualBox:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
ahmad-farg@ahmadfarg-VirtualBox:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o hello
ahmad-farg@ahmadfarg-VirtualBox:~/work/arch-pc/lab04$ ./hello
ахмед метвалли

```

Рис. 3.3: Запуск получившегося файла.

Скопируйте файлы hello.asm и lab4.asm в Ваш локальный репозиторий в каталог ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab04/. Загрузите файлы на Github.

```

ahmad-farg@ahmadfarg-VirtualBox:~/work/arch-pc/lab04$ cp hello.asm ~/work/study/2023-2024/Архитектура\
\ компьютера/arch-pc/labs/lab04/
ahmad-farg@ahmadfarg-VirtualBox:~/work/arch-pc/lab04$ cp lab4.asm ~/work/study/2023-2024/Архитектура\
\ компьютера/arch-pc/labs/lab04/
ahmad-farg@ahmadfarg-VirtualBox:~/work/arch-pc/lab04$

```

Рис. 3.3: Отправка в локальный репозиторий.

```

ahmad-farg@ahmadfarg-VirtualBox:~$ cd ~/work/study/2023-2024/Архитектура\ компьютера/arch-pc
ahmad-farg@ahmadfarg-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git commit -am
'feat(main): add files lab-4'
[master 2c08ae5] feat(main): add files lab-4
2 files changed, 24 insertions(+), 24 deletions(-)
ahmad-farg@ahmadfarg-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 3 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 609 bytes | 76.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:mohammad-hamdi/study_2023-2024_arh--pc.git
78b079d..2c08ae5 master -> master
ahmad-farg@ahmadfarg-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$

```

4. ВЫВОДЫ

В ходе выполнения лабораторной работы я освоил процедуры компиляции и сборки программ, написанных на ассемблере NASM.