

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4

дисциплина: Архитектура компьютера

Студент: Абдуллахи Бахара

Группа: НПИБД – 03 -23

МОСКВА

2023__ г.

Цель работы

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

Рассмотрим пример простой программы на языке ассемблера NASM. Традиционно первая программа выводит приветственное сообщение Hello world! на экран. Создайте каталог для работы с программами на языке ассемблера NASM:

Создала каталог для работы с программами на языке ассемблера NASM.

```
Setting up nasm (2.10.01-1) ...
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~$ mkdir -p ~/work/arch-pc/lab04
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~$ cd ~/work/arch-pc/lab04
```

Перейдите в созданный каталог:

```
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~$ cd ~/work/arch-pc/lab04
```

Создайте текстовый файл с именем hello.asm

Создала текстовый файл с именем hello.asm

```
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~/work/arch-pc/lab04$ touch hello.asm
```

откройте этот файл с помощью любого текстового редактора, например, gedit

Открыла файл с помощью gedit

```
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~/work/arch-pc/lab04$ gedit hello.asm
```

и введите в него следующий текст:

; hello.asm

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 ; Вызов ядра

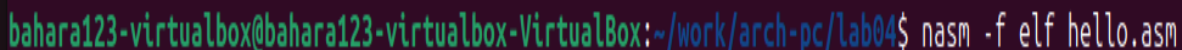
```
;hello.asm
SECTION .data                                ; Начало секции данных
hello:          DB 'Hello world!' ,10        ; 'Hello world!' плюс
                                                    ; символ перевода строки
helloLen:       EQU $-hello                  ; Длина строки hello
SECTION .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                                           ; Вызов ядра
```

NASM превращает текст программы в объектный код. Например, для компиляции приведённого выше текста программы «Hello World» необходимо написать:

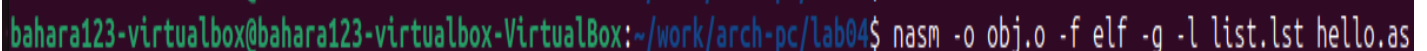
nasm -f elf hello.asm



```
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~/work/arch-pc/lab04$ nasm -f elf hello.asm
```

Выполните следующую команду:

nasm -o obj.o -f elf -g -l list.lst hello.asm



```
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst hello.asm
```

Как видно из схемы на рис. 4.3, чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику:

```
ld -m elf_i386 hello.o -o hello
```

С помощью команды `ls` проверьте, что исполняемый файл `hello` был создан.

Компоновщик `ld` не предполагает по умолчанию расширений для файлов, но принято использовать следующие расширения:

```
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~/work/arch-pc/lab04$ ls
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o list.lst obj.o
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~/work/arch-pc/lab04$
```

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

Выполните следующую команду:

```
ld -m elf_i386 obj.o -o main
```

```
:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~/work/arch-pc/lab04$ ls
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o list.lst main obj.o
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~/work/arch-pc/lab04$
```

Какое имя будет иметь исполняемый файл? Какое имя имеет объектный файл из которого собран этот исполняемый файл?

Формат командной строки `LD` можно увидеть, набрав `ld --help`. Для получения более подробной информации см. `man ld`.

Запустить на выполнение созданный исполняемый файл, находящийся в текущем каталоге, можно, набрав в командной строке:

```
./hello
```

```
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~/work/arch-pc/lab04$ ./hello
Hello world!
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~/work/arch-pc/lab04$
```

Задание для самостоятельной работы :

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

```

ox:~/work/arch-pc/lab04$ cd ..
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~/work/arch-pc$ cp lab04/hello.asm lab04/lab4.asm
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~/work/arch-pc$ ls
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~$ cd ~/work/arch-pc/lab04
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4.asm list.lst main obj.o
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~/work/arch-pc/lab04$

```

2. С помощью любого текстового редактора внесите изменения в текст программы в файле lab4.asm так, чтобы вместо Hello world! на экран выводилась строка с вашими фамилией и именем.

```

bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~/work/arch-pc/lab04$ gedit
hello.asm lab4.asm
pbe/sect/permissions: Permission denied (glibc error: quark; 2)
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~/work/arch-pc/lab04$

```

```

;hello.asm
SECTION .data
hello: DB 'Абдуллахи Бахара' ,10 ; Начало секции данных ; 'Абдуллахи Бахара'
; 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 ; Вызов ядра

```

3. Оттранслируйте полученный текст программы lab4.asm в объектный файл.

Выполните компоновку объектного файла и запустите получившийся исполняемый файл.

```

bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst lab4.asm
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~/work/arch-pc/lab04$ ld -m elf
ld: unrecognized emulation mode: elf
Supported emulations: elf_x86_64 elf32_x86_64 elf_i386 elf_iamcu i386pep i386pe
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o lab4
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4 lab4.asm lab4.o list.lst main obj.o
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~/work/arch-pc/lab04$

```

4. 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.

```
Архитектура компьютера/study_2023-2024_arh--pc$ cd
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~$ cp ~/work/arch-pc/lab04/
lab4.asm ~/work/study/2023-2024/Архитектура\ компьютера/study_2023-2024_arh--pc/
labs/lab04/
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~$ cp ~/work/arch-pc/lab04/
hello.asm ~/work/study/2023-2024/Архитектура\ компьютера/study_2023-2024_arh--pc
/labs/lab04/
bahara123-virtualbox@bahara123-virtualbox-VirtualBox:~$
```