

Отчет по лабораторной работе №4

**Создание и процесс обработки программ на языке ассемблера
NASM**

Кичигина Полина Евгеньевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Задание для самостоятельной работы	10
5	Выводы	12

Список иллюстраций

3.1	Создаем каталог и переходим в него	7
3.2	Создаем файл и открываем его	7
3.3	Редактируем	8
3.4	Используем команду <code>nasm</code> и проверяем работу команды	8
3.5	Преобразуем файл <code>hello.asm</code> в <code>obj.o</code> и проверяем	8
3.6	Используем команду <code>ld</code> и проверяем	9
3.7	Используем команду <code>ld</code> , создавая файл <code>main</code> и проверяем	9
3.8	Используем команду <code>./hello</code>	9
4.1	Создаем копию	10
4.2	Редактируем	10
4.3	Прописываем команды для работы файла и запускаем программу	11
4.4	Копируем файлы в локальный репозиторий	11

Список таблиц

1 Цель работы

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


2 Задание

Написать 2 программы: Hello world, lab4(Имя Фамилия))

3 Выполнение лабораторной работы

1. Программа Hello world!

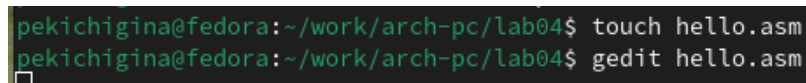
Создайте каталог для работы с программами на языке ассемблера NASM и перейдите в созданный каталог(рис. 3.1)



```
pekichigina@fedora:~$ mkdir -p ~/work/arch-pc/lab04
pekichigina@fedora:~$ cd ~/work/arch-pc/lab04
```

Рис. 3.1: Создаем каталог и переходим в него

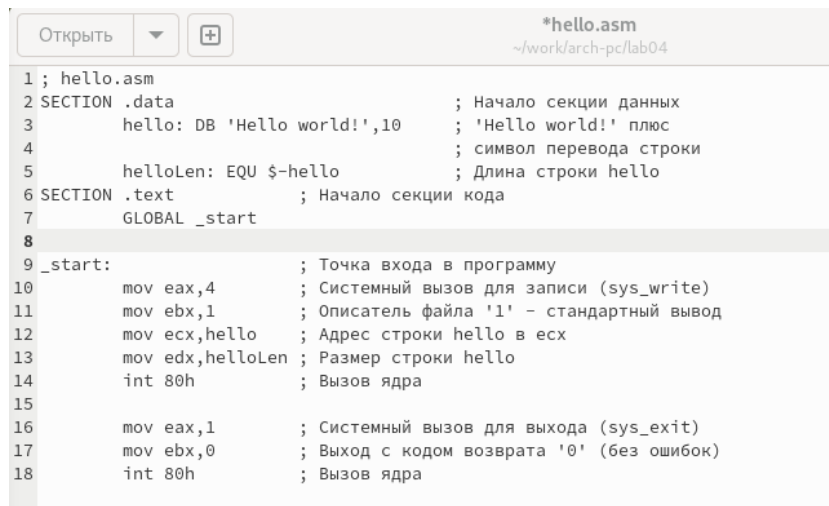
Создайте текстовый файл с именем hello.asm и откройте этот файл с помощью любого текстового редактора, например, gedit(рис. 3.2)



```
pekichigina@fedora:~/work/arch-pc/lab04$ touch hello.asm
pekichigina@fedora:~/work/arch-pc/lab04$ gedit hello.asm
```

Рис. 3.2: Создаем файл и открываем его

И введите в него следующий текст(рис. 3.3)

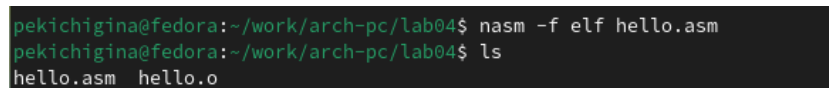


```
1; hello.asm
2SECTION .data                ; Начало секции данных
3    hello: DB 'Hello world!',10 ; 'Hello world!' плюс
4                                ; символ перевода строки
5    helloLen: EQU $-hello      ; Длина строки hello
6SECTION .text                ; Начало секции кода
7    GLOBAL _start
8
9_start:                      ; Точка входа в программу
10   mov eax,4                 ; Системный вызов для записи (sys_write)
11   mov ebx,1                 ; Описатель файла '1' - стандартный вывод
12   mov ecx,hello             ; Адрес строки hello в ecx
13   mov edx,helloLen          ; Размер строки hello
14   int 80h                   ; Вызов ядра
15
16   mov eax,1                 ; Системный вызов для выхода (sys_exit)
17   mov ebx,0                 ; Выход с кодом возврата '0' (без ошибок)
18   int 80h                   ; Вызов ядра
```

Рис. 3.3: Редактируем

2. Транслятор NASM

Преобразуем текст программы в объектный код и проверим создан ли объектный файл(рис. 3.4)

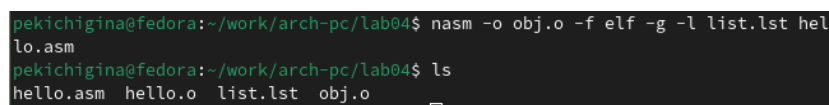


```
pekichigina@fedora:~/work/arch-pc/lab04$ nasm -f elf hello.asm
pekichigina@fedora:~/work/arch-pc/lab04$ ls
hello.asm  hello.o
```

Рис. 3.4: Используем команду nasm и проверяем работу команды

3. Расширенный синтаксис командной строки NASM

Компилируем исходный файл и проверяем, как сработала команда(рис. 3.5)



```
pekichigina@fedora:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst hello.asm
pekichigina@fedora:~/work/arch-pc/lab04$ ls
hello.asm  hello.o  list.lst  obj.o
```

Рис. 3.5: Преобразуем файл hello.asm в obj.o и проверяем

4. Компоновщик LD

Чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику(рис. 3.6)


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

Рис. 3.6: Используем команду ld и проверяем

Выполните следующую команду(рис. 3.7)

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

Рис. 3.7: Используем команду ld, создавая файл main и проверяем

5. Запуск исполняемого файла

Запустить на выполнение созданный исполняемый файл, находящийся в текущем каталоге, можно, набрав в командной строке(рис. 3.8)

```
pekichigina@fedora:~/work/arch-pc/lab04$ ./hello
Hello world!
pekichigina@fedora:~/work/arch-pc/lab04$
```

Рис. 3.8: Используем команду ./hello

4 Задание для самостоятельной работы

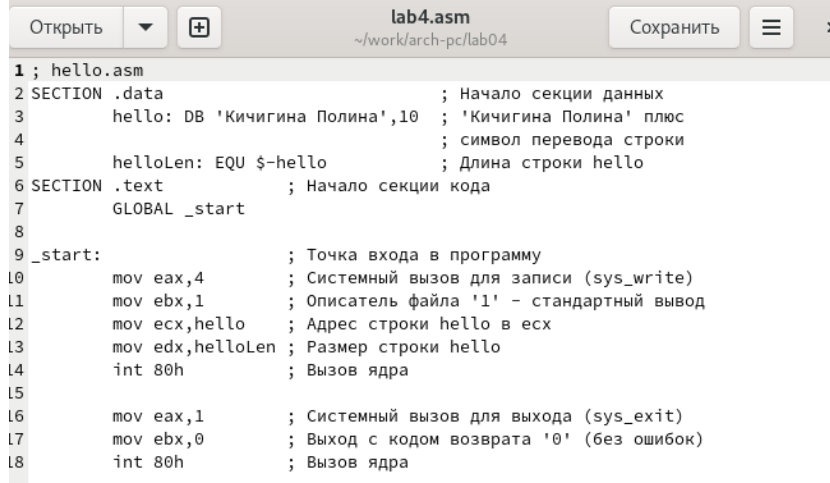
1. В каталоге ~/work/arch-pc/lab04 с помощью команды `cp` создайте копию файла `hello.asm` с именем `lab4.asm` (рис. 4.1)

```
pekichigina@fedora:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
pekichigina@fedora:~/work/arch-pc/lab04$
```

Рис. 4.1: Создаем копию

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 (рис. 4.2)

```
pekichigina@fedora:~/work/arch-pc/lab04$ gedit lab4.asm
```



```
1 ; hello.asm
2 SECTION .data ; Начало секции данных
3     hello: DB 'Пекичигина Полина',10 ; 'Пекичигина Полина' плюс
4 ; символ перевода строки
5     helloLen: EQU $-hello ; Длина строки hello
6 SECTION .text ; Начало секции кода
7     GLOBAL _start
8
9 _start: ; Точка входа в программу
10    mov eax,4 ; Системный вызов для записи (sys_write)
11    mov ebx,1 ; Описатель файла '1' - стандартный вывод
12    mov ecx,hello ; Адрес строки hello в ecx
13    mov edx,helloLen ; Размер строки hello
14    int 80h ; Вызов ядра
15
16    mov eax,1 ; Системный вызов для выхода (sys_exit)
17    mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
18    int 80h ; Вызов ядра
```

Рис. 4.2: Редактируем

3. Оттранслируйте полученный текст программы lab4.asm в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл(рис. 4.3)

```
pekichigina@fedora:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
pekichigina@fedora:~/work/arch-pc/lab04$ gedit lab4.asm
pekichigina@fedora:~/work/arch-pc/lab04$ gedit lab4.asm
pekichigina@fedora:~/work/arch-pc/lab04$ nasm -f elf hello.asm
pekichigina@fedora:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
pekichigina@fedora:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst lab4.asm
pekichigina@fedora:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
pekichigina@fedora:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
pekichigina@fedora:~/work/arch-pc/lab04$ ./lab4
Кичигина Полина
pekichigina@fedora:~/work/arch-pc/lab04$
```

Рис. 4.3: Прописываем команды для работы файла и запускаем программу

4. Скопируйте файлы hello.asm и lab4.asm в Ваш локальный репозиторий в каталог ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/(рис. 4.4)

```
pekichigina@fedora:~/work/arch-pc/lab04$ cp hello.asm ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab04/
pekichigina@fedora:~/work/arch-pc/lab04$ cp lab4.asm ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab04/
pekichigina@fedora:~/work/arch-pc/lab04$
```

Рис. 4.4: Копируем файлы в локальный репозиторий

5 Выводы

Мы познакомились с языком ассемблера NASM и создали две работающих программы.