

Отчёт по лабораторной работе №4

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

Хохлачёва Полина Дмитриевна

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
4	Выводы	11

Список иллюстраций

3.1	Создаем каталоги с помощью команды <code>mkdir</code>	6
3.2	Переходим в каталог с помощью команды <code>cd</code>	6
3.3	Создаем текстовый файл	6
3.4	Открываем и заполняем по примеру	7
3.5	Используем команду <code>paste</code>	7
3.6	Проверяем работу команды	7
3.7	Преобразуем файл	8
3.8	Проверяем работу командой <code>ls</code>	8
3.9	Используем команду <code>ld</code>	8
3.10	Используем команду <code>ls</code>	8
3.11	Создаём файл <code>main</code>	8
3.12	Используем команду <code>ls</code>	8
3.13	Используем команду <code>./hello</code>	9
3.14	Используем команду <code>cp</code>	9
3.15	Открываем файл в текстовом редакторе	9
3.16	Редактируем файл	9
3.17	Прописываем команды	10
3.18	Копируем файлы	10

1 Цель работы

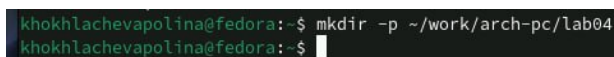
Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

2 Задание

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

3 Выполнение лабораторной работы

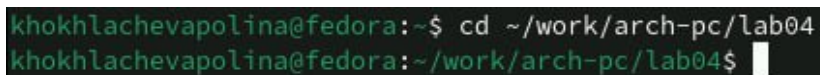
Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 3.1). Создайте каталог для работы с программами на языке ассемблера NASM



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

Рис. 3.1: Создаем каталоги с помощью команды mkdir


Переходим в созданный каталог



```
khokhlachevapolina@fedora:~$ cd ~/work/arch-pc/lab04
khokhlachevapolina@fedora:~/work/arch-pc/lab04$
```

Рис. 3.2: Переходим в каталог с помощью команды cd

Создаём текстовый файл с именем hello.asm



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

Рис. 3.3: Создаем текстовый файл

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

```

1 |; hello.asm
2 SECTION .data          ; Начало секции данных
3     hello:      DB 'Hello world!',10 ; 'Hello world!' плюс
4                                     ; символ перевода строки
5     helloLen:   EQU $-hello      ; Длина строки hello
6
7 SECTION .text          ;Начало секции кода
8     GLOBAL _start
9
10 _start:              ; Точка входа в программу
11     mov eax,4        ; Системный вызов для записи (sys_write)
12     mov ebx,1        ; Описатель файла '1' - стандартный вывод
13     mov ecx,hello    ; Адрес строки hello в ecx
14     mov edx,helloLen ; Размер строки hello
15     int 80h         ; Вызов ядра
16
17     mov eax,1        ; Системный вызов для выхода (sys_exit)
18     mov ebx,0        ; Выход с кодом возврата '0' (без ошибок)
19     int 80h         ; Вызов ядра
20

```

Рис. 3.4: Открываем и заполняем по примеру

Преобразуем текст программы в объектный код

```

khokhlachevapolina@fedora:~/work/arch-pc/lab04$ nasm -f elf hello.asm
bash: nasm: команда не найдена...
Установить пакет «nasm», предоставляющий команду «nasm»? [N/y] y

* Ожидание в очереди...
* Загрузка списка пакетов....
Следующие пакеты должны быть установлены:
nasm-2.16.01-7.fc40.x86_64      A portable x86 assembler which uses Intel-like syntax
Продолжить с этими изменениями? [N/y] y

* Ожидание в очереди...
* Ожидание аутентификации...
* Ожидание в очереди...
* Загрузка пакетов...
* Запрос данных...
* Проверка изменений...
* Установка пакетов...

khokhlachevapolina@fedora:~/work/arch-pc/lab04$

```

Рис. 3.5: Используем команду nasm

Проверяем созданся ли файл с помощью команды ls

```

khokhlachevapolina@fedora:~/work/arch-pc/lab04$ ls
hello.asm  hello.o
khokhlachevapolina@fedora:~/work/arch-pc/lab04$

```

Рис. 3.6: Проверяем работу команды

Скомпилируем исходный файл hello.asm в obj.o

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

Рис. 3.7: Преобразуем файл

Проверяем работу команды

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

Рис. 3.8: Проверяем работу командой ls

Передаём объектный файл на обработку компоновщику

```
khokhlachevapolina@fedora:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
khokhlachevapolina@fedora:~/work/arch-pc/lab04$
```

Рис. 3.9: Используем команду ld

Проверяем создание файла

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

Рис. 3.10: Используем команду ls

Передаём объектный файл на обработку компоновщику

```
khokhlachevapolina@fedora:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
khokhlachevapolina@fedora:~/work/arch-pc/lab04$
```

Рис. 3.11: Создаём файл main

Проверяем создался ли файл hello

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

Рис. 3.12: Используем команду ls

Запускаем на выполнение созданный исполняемый файл


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

Рис. 3.13: Используем команду ./hello

Создаём копию файла hello.asm с именем lab4.asm

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

Рис. 3.14: Используем команду cp

Открываем файл и редактируем его

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

Рис. 3.15: Открываем файл в текстовом редакторе

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

Рис. 3.16: Редактируем файл

Прописываем команды, как и в первой программе

```

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

```

Рис. 3.17: Прописываем команды

Копируем файлы

```

hokhlachevapolina@fedora:~/work/arch-pc/lab04$ cp hello.asm ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/lab04/
//
hokhlachevapolina@fedora:~/work/arch-pc/lab04$ cp lab04.asm ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/lab04/
//
hokhlachevapolina@fedora:~/work/arch-pc/lab04$ █

```

Рис. 3.18: Копируем файлы

4 Выводы

Мы освоили процедуры компиляции и сборки программ, познакомились с языком NASM и создали две работающие программы