

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: Фролота Т.М

Группа: НММбд-02-24

МОСКВА

2024 г.

Содержание

1. Цель работы.....3

2. Задания.....4

4.Выполнение лаборотарной работы.....5

5.Вывод:.....7

1. Цель работы.

Научиться компилировать и собирать программы на ассемблере NASM.

2. Задания.

- 1) Создать программу "Hello world!" на ассемблере NASM;
- 2) Скомпилировать и собрать программу с помощью транслятора NASM и компоновщика LD;
- 3) Запустить файл программы.

4.Выполнение лаборотарной работы.

1.Открываем терминал и вводим следующую команду для создания каталога.

```
tmfrolova@dk8n64 ~ $ mkdir -p ~/work/arch-pc/lab04
```

Рис.1.1 Создание каталога.

Входим в созданный каталог

```
tmfrolova@dk8n64 ~ $ cd ~/work/arch-pc/lab04
```

Рис.1.2 Вход в каталог .

Создаём нужный для работы файл.

```
tmfrolova@dk8n64 ~/work/arch-pc/lab04 $ touch hello.asm
```

Рис.1.3 Создание файла.

Проверим его наличие.

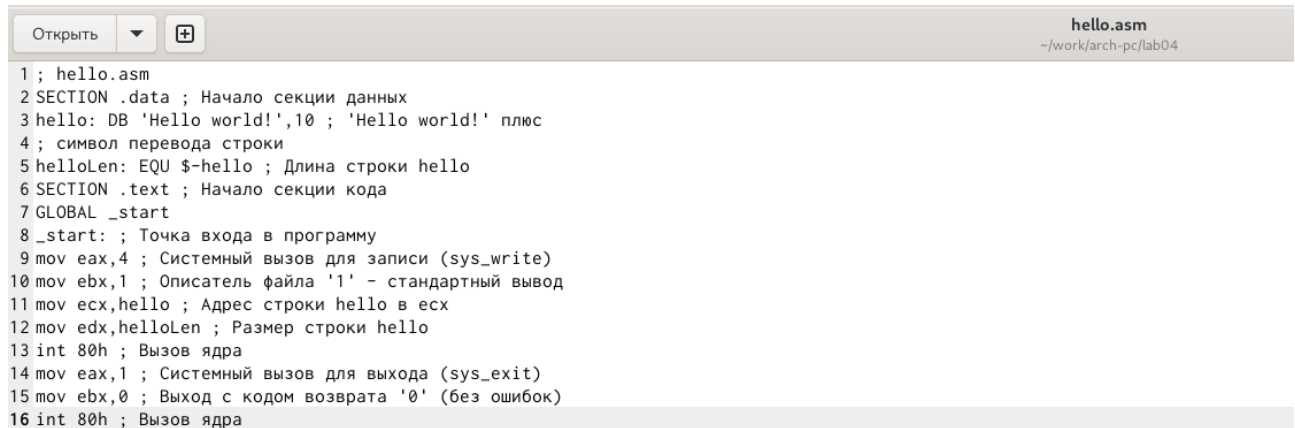
```
tmfrolova@dk8n64 ~/work/arch-pc/lab04 $ ls  
hello.asm
```

Откроем файл с помощью текстового редактора.

```
tmfrolova@dk8n64 ~/work/arch-pc/lab04 $ gedit hello.asm
```

Рис.1.4 Открытие файла.

Введём туда следующий текст.



```
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 SECTION .text ; Начало секции кода  
7 GLOBAL _start  
8 _start: ; Точка входа в программу  
9 mov eax,4 ; Системный вызов для записи (sys_write)  
10 mov ebx,1 ; Описатель файла '1' - стандартный вывод  
11 mov ecx,hello ; Адрес строки hello в ecx  
12 mov edx,helloLen ; Размер строки hello  
13 int 80h ; Вызов ядра  
14 mov eax,1 ; Системный вызов для выхода (sys_exit)  
15 mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)  
16 int 80h ; Вызов ядра
```

Рис.1.5 Текст в файле.

Далее вводи команды и проверяем их исполнение.

```
tmfrolova@dk8n64 ~/work/arch-pc/lab04 $ nasm -f elf hello.asm
```

Рис.1.6 Команда nasm -f elf hello.asm

```
tmfrolova@dk8n64 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o
```

Рис.1.7 Проверка выполнения.

```
tmfrolova@dk8n64 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
```

Рис.1.8 Команда `nasm -o obj.o -f elf -g -l list.lst hello.asm`

```
tmfrolova@dk8n64 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o  list.lst  obj.o
```

Рис.1.9 Проверка выполнения.

```
tmfrolova@dk8n64 ~/work/arch-pc/lab04 $ ld -m elf_i386 hello.o -o hello
```

Рис.1.10 Команда `ld -m elf_i386 hello.o -o hello`

```
tmfrolova@dk8n64 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  list.lst  obj.o
```

Рис.1.11 Проверка выполнения.

```
tmfrolova@dk8n64 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
```

Рис.1.12 Команда `ld -m elf_i386 obj.o -o main`

```
tmfrolova@dk8n64 ~/work/arch-pc/lab04 $ ./hello
Hello world!
```

Запустим созданный во время работы файл.

Рис.1.13 Запуск файла.

5.Вывод.

В ходе работы была освоена процедура компиляции и сборки программ, написанных на ассемблере NASM.