

Отчёт по лабораторной работе№4

Создание и процесс обработки программ на языке ассемблера NASM

Шмавонян Диана Арамовна

Содержание

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

Список иллюстраций

3.1	Создание каталога	7
3.2	Переход в каталог	7
3.3	Создание текстового файла	7
3.4	Открытие файла	8
3.5	Ввод текста	8
3.6	Компиляция текста	8
3.7	проверка, что объектный файл был создан	9
3.8	Создание файлов	9
3.9	Передача файла на компоновку	9
3.10	Зададим имя создаваемого исполняемого файла	9
3.11	Запуск на выполнение созданный исполняемый файл	10
3.12	Создание копии файла с именем lab4.asm	10
3.13	Внесение изменения в текст программы	10
3.14	Оттранслирование, компоновка, запуск	11

Список таблиц

1 Цель работы

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

2 Задание

Здесь приводится описание задания в соответствии с рекомендациями методического пособия и выданным вариантом.

3 Выполнение лабораторной работы

1. Создайте каталог для работы с программами на языке ассемблера NASM:

```
dashmavonyan@dk2n25 ~ $ mkdir -p ~/work1/arch-pc1/lab04  
dashmavonyan@dk2n25 ~ $
```

Рис. 3.1: Создание каталога

2. Перейдём в созданный каталог:

```
dashmavonyan@dk2n25 ~ $ cd ~/work1/arch-pc1/lab04  
dashmavonyan@dk2n25 ~/work1/arch-pc1/lab04 $
```

Рис. 3.2: Переход в каталог

3. Создадим текстовый файл с именем hello.asm:

```
dashmavonyan@dk2n25 ~/work1/arch-pc1/lab04 $ touch hello.asm  
dashmavonyan@dk2n25 ~/work1/arch-pc1/lab04 $
```

Рис. 3.3: Создание текстового файла

4. Откроем этот файл с помощью текстового редактора

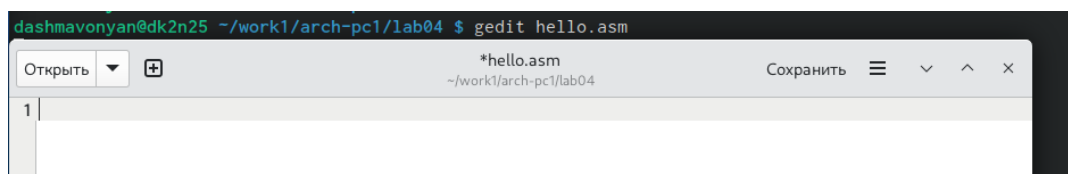


Рис. 3.4: Открытие файла

5. Введём в него текст:

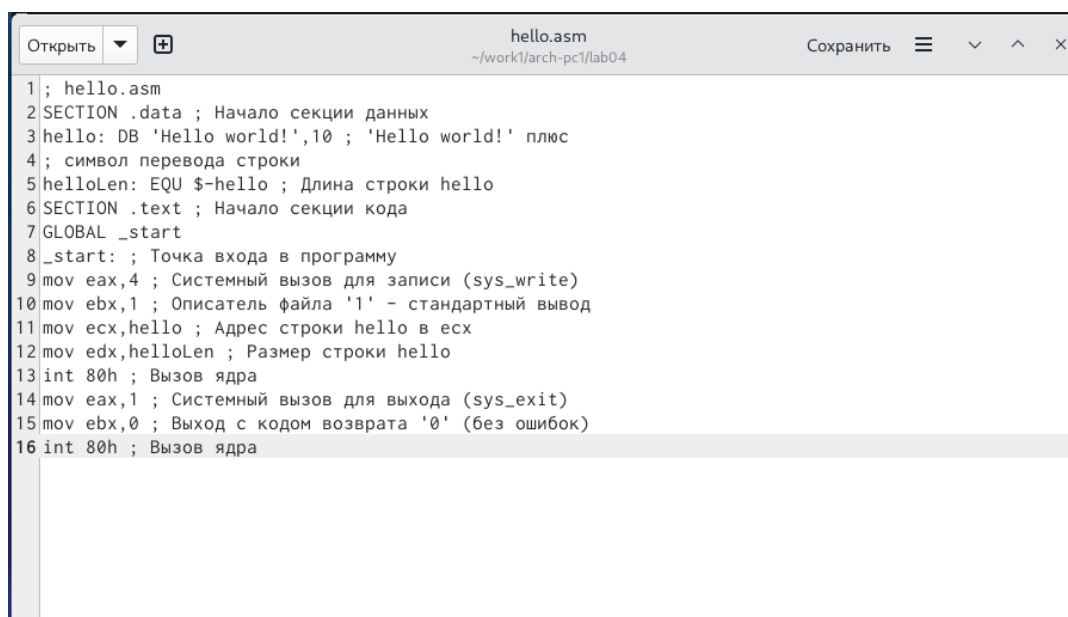


Рис. 3.5: Ввод текста

6. Скомпилируем данный текст

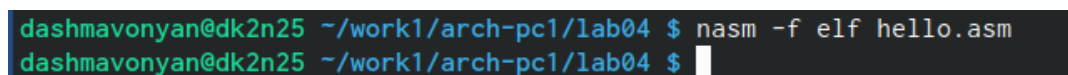


Рис. 3.6: Компиляция текста

7. Проверим, что объектный файл был создан:


```
dashmavonyan@dk2n25 ~/work1/arch-pc1/lab04 $ ls
hello.asm  hello.o
dashmavonyan@dk2n25 ~/work1/arch-pc1/lab04 $
```

Рис. 3.7: проверка, что объектный файл был создан

8. Скомпилируем исходный файл hello.asm в obj.o и создадим файл листинга list.lst и проверим

```
dashmavonyan@dk2n25 ~/work1/arch-pc1/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
dashmavonyan@dk2n25 ~/work1/arch-pc1/lab04 $ ls
hello.asm  hello.o  list.lst  obj.o
dashmavonyan@dk2n25 ~/work1/arch-pc1/lab04 $
```

Рис. 3.8: Создание файлов

9. Передадим объектный файл на обработку компоновщику и проверим.

```
dashmavonyan@dk2n25 ~/work1/arch-pc1/lab04 $ ld -m elf_i386 hello.o -o hello
dashmavonyan@dk2n25 ~/work1/arch-pc1/lab04 $ ls
hello  hello.asm  hello.o  list.lst  obj.o
dashmavonyan@dk2n25 ~/work1/arch-pc1/lab04 $
```

Рис. 3.9: Передача файла на компоновку

10. Зададим имя создаваемого исполняемого файла.

```
dashmavonyan@dk2n25 ~/work1/arch-pc1/lab04 $ ld -m elf_i386 obj.o -o main
dashmavonyan@dk2n25 ~/work1/arch-pc1/lab04 $
```

Рис. 3.10: Зададим имя создаваемого исполняемого файла

11. Запустим на выполнение созданный исполняемый файл, находящийся в текущем каталоге.

```
dashmavonyan@dk2n25 ~/work1/arch-pc1/lab04 $ ./hello
Hello world!
dashmavonyan@dk2n25 ~/work1/arch-pc1/lab04 $
```

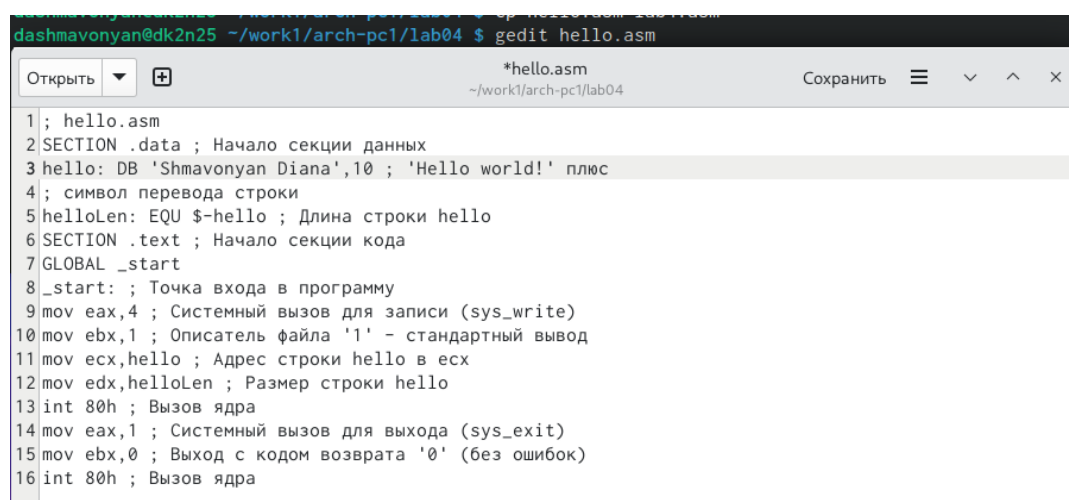
Рис. 3.11: Запуск на выполнение созданный исполняемый файл

12. Создадим копию файла hello.asm с именем lab4.asm

```
dashmavonyan@dk2n25 ~/work1/arch-pc1/lab04 $ cp hello.asm lab4.asm
dashmavonyan@dk2n25 ~/work1/arch-pc1/lab04 $
```

Рис. 3.12: Создание копии файла с именем lab4.asm

13. Внесём изменения в текст программы в файле lab5.asm



```
dashmavonyan@dk2n25 ~/work1/arch-pc1/lab04 $ gedit hello.asm
Открыть [иконка] *hello.asm
~/work1/arch-pc1/lab04 Сохранить [иконка] [иконка] [иконка] [иконка]
1; hello.asm
2 SECTION .data ; Начало секции данных
3 hello: DB 'Shmavonyan Diana',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. 3.13: Внесение изменения в текст программы

14. Оттранслируем полученный текст программы lab5.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл.

```

dashmavonyan@dk2n25 ~/work1/arch-pc1/lab04 $ cp hello.asm lab4.asm
dashmavonyan@dk2n25 ~/work1/arch-pc1/lab04 $ nasm -f elf lab4.asm
dashmavonyan@dk2n25 ~/work1/arch-pc1/lab04 $ nasm -o obj.o -f elf -g -l list.lst lab4.asm
dashmavonyan@dk2n25 ~/work1/arch-pc1/lab04 $ nasm -o Shmavonyan.o -f elf -g -l list.lst lab4.asm
dashmavonyan@dk2n25 ~/work1/arch-pc1/lab04 $ ld -m elf_i386 Shmavonyan.o -o Shmavonyan
dashmavonyan@dk2n25 ~/work1/arch-pc1/lab04 $ ./Shmavonyan
Shmavonyan Diana
dashmavonyan@dk2n25 ~/work1/arch-pc1/lab04 $ █

```

Рис. 3.14: Оттранслирование, компоновка, запуск

15. Скопируем файлы hello.asm и lab04.asm в локальный репозиторий в каталог

~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/ с по-

```

dashmavonyan@dk2n25 ~/work1/arch-pc1/lab04 $ cp hello.asm ~/work1/study1/2023-2024/"Архитектура компьютера"/arch-pc1/labs/lab04/report/
dashmavonyan@dk2n25 ~/work1/arch-pc1/lab04 $ cp lab4.asm ~/work1/study1/2023-2024/"Архитектура компьютера"/arch-pc1/labs/lab04/report/
dashmavonyan@dk2n25 ~/work1/arch-pc1/lab04 $ cd ~/work1/study1/2023-2024/"Архитектура компьютера"/arch-pc1/labs/lab04/report/
dashmavonyan@dk2n25 ~/work1/study1/2023-2024/"Архитектура компьютера"/arch-pc1/labs/lab04/report $ ls
bib hello.asm image lab4.asm Makefile pandoc report.md
dashmavonyan@dk2n25 ~/work1/study1/2023-2024/"Архитектура компьютера"/arch-pc1/labs/lab04/report $ █

```

мощью утилиты `cp` и проверим.

16. Загрузим файлы на Github.

4 Выводы

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