

Отчёт по лабораторной работе №4

дисциплина: Архитектура компьютера

Маслова Анна Павловна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выполнение заданий для самостоятельной работы	9
4	Выводы	12
	Список литературы	13

Список иллюстраций

2.1	Создание каталога	6
2.2	Создание файла hello.asm	6
2.3	Ввод текста программы	7
2.4	Компиляция программы	7
2.5	Создание объектного файла	7
2.6	Компоновка файла	8
2.7	Создание исполняемого файла main	8
2.8	Запуск программы	8
3.1	Копирование файла	9
3.2	Редактирование файла lab4.asm	10
3.3	Передача программы транслятору	10
3.4	Компиляция кода	10
3.5	Компоновка объектного файла	11
3.6	Создание исполняемого файла	11
3.7	Запуск программы lab4	11

Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

Для начала создаём специальный каталог для работы с программами на языке ассемблера NASM и сразу перейдём в этот каталог (рис.2.1).

```
apmaslova@dk5n59 ~ $ mkdir -p ~/work/arch-pc/lab04
apmaslova@dk5n59 ~ $ cd ~/work/arch-pc/lab04
apmaslova@dk5n59 ~/work/arch-pc/lab04 $
```

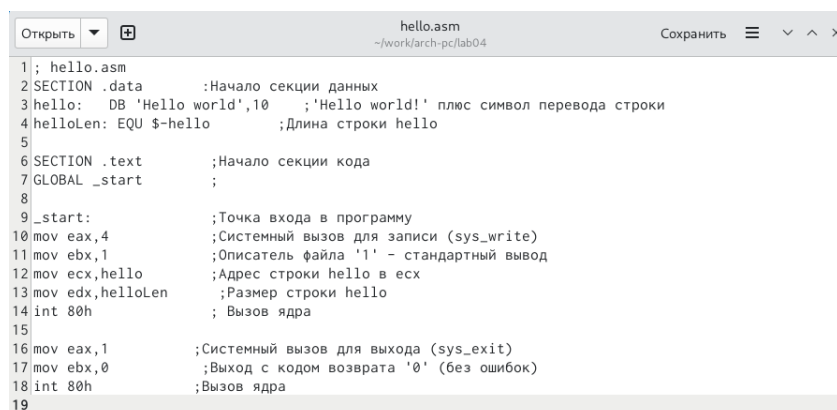
Рис. 2.1: Создание каталога

Далее создадим в этом каталоге текстовый файл с именем `hello.asm` и откроем его в редакторе `gedit` (рис.2.2).

```
apmaslova@dk5n59 ~/work/arch-pc/lab04 $ touch hello.asm
apmaslova@dk5n59 ~/work/arch-pc/lab04 $ gedit hello.asm
```

Рис. 2.2: Создание файла `hello.asm`

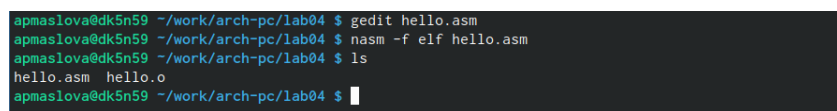
В `gedit` в этот файл введём текст программы для вывода на экран ‘Hello world!’ (рис.2.3).



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

Рис. 2.3: Ввод текста программы

Затем необходимо скомпилировать написанный код. Для этого воспользуемся следующей командой в консоли: (рис.2.4)

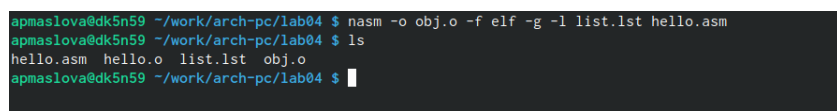


```
apmaslova@dk5n59 ~/work/arch-pc/lab04 $ gedit hello.asm
apmaslova@dk5n59 ~/work/arch-pc/lab04 $ nasm -f elf hello.asm
apmaslova@dk5n59 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o
apmaslova@dk5n59 ~/work/arch-pc/lab04 $
```

Рис. 2.4: Компиляция программы

После проверки убедились, что в каталоге есть файл с текстом программы с расширением .asm и файл с объектным кодом с расширением .o (убедились, что они имеют одинаковые названия - hello).

Далее создадим объектный файл obj.o (рис.2.5)



```
apmaslova@dk5n59 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
apmaslova@dk5n59 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o  list.lst  obj.o
apmaslova@dk5n59 ~/work/arch-pc/lab04 $
```

Рис. 2.5: Создание объектного файла

После проверки убедились, что файл создан.

Этот файл необходимо компоновать. Передадим его компоновщику с помощью следующей команды и проверим наличие соответствующего файла: (рис.2.6)

```
hello.asm hello.o list.lst obj.o
apmaslova@dk5n59 ~/work/arch-pc/lab04 $ ld -m elf_i386 hello.o -o hello
apmaslova@dk5n59 ~/work/arch-pc/lab04 $ ls
hello hello.asm hello.o list.lst obj.o
apmaslova@dk5n59 ~/work/arch-pc/lab04 $
```

Рис. 2.6: Компоновка файла

Видим, что файл hello создан.

И после этого создаём исполняемый файл из объектного файла obj.o и задаём ему имя main (рис.2.7).

```
hello hello.asm hello.o list.lst obj.o
apmaslova@dk5n59 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
apmaslova@dk5n59 ~/work/arch-pc/lab04 $ ls
hello hello.asm hello.o list.lst main obj.o
apmaslova@dk5n59 ~/work/arch-pc/lab04 $
```

Рис. 2.7: Создание исполняемого файла main

Проверили, увидели, что исполняемый файл main создан.

Запускаем исполняемый файл: (рис.2.8).

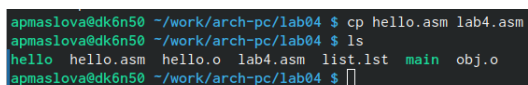
```
apmaslova@dk5n59 ~/work/arch-pc/lab04 $ ./hello
Hello world
apmaslova@dk5n59 ~/work/arch-pc/lab04 $
```

Рис. 2.8: Запуск программы

Увидели на экране фразу “Hello world!”.

3 Выполнение заданий для самостоятельной работы

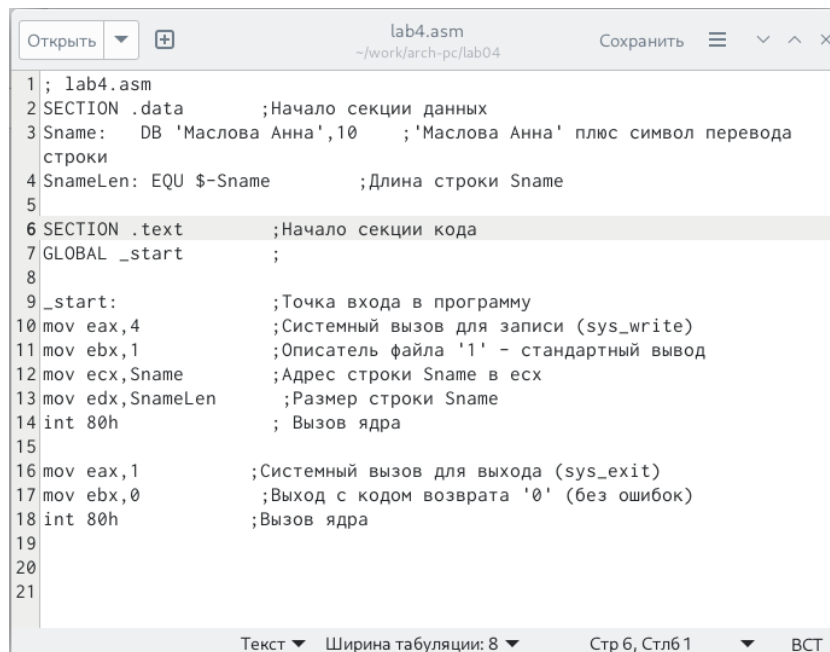
Сначала в каталоге `~/work/arch-pc/lab04` создаём копию файла `hello.asm` с именем `lab4.asm` (рис.3.1).



```
apmaslova@dk6n50 ~/work/arch-pc/lab04 $ cp hello.asm lab4.asm
apmaslova@dk6n50 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  lab4.asm  list.lst  main  obj.o
apmaslova@dk6n50 ~/work/arch-pc/lab04 $
```

Рис. 3.1: Копирование файла

Далее в редакторе изменяем текст программы, чтобы на экран выводилась строка “Маслова Анна” (рис.3.2).

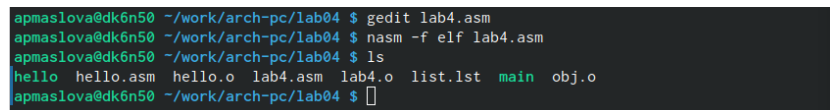


```
1 ; lab4.asm
2 SECTION .data          ;Начало секции данных
3 Sname: DB 'Маслова Анна',10 ;'Маслова Анна' плюс символ перевода
   строки
4 SnameLen: EQU $-Sname   ;Длина строки Sname
5
6 SECTION .text          ;Начало секции кода
7 GLOBAL _start          ;
8
9 _start:                 ;Точка входа в программу
10 mov eax,4              ;Системный вызов для записи (sys_write)
11 mov ebx,1              ;Описатель файла '1' - стандартный вывод
12 mov ecx,Sname          ;Адрес строки Sname в ecx
13 mov edx,SnameLen       ;Размер строки Sname
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
19
20
21
```

Рис. 3.2: Редактирование файла lab4.asm

Сохранили изменения.

Затем преобразуем текст написанной программы в объектный код (рис.3.3).

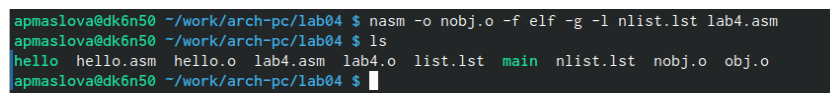


```
apmaslova@dk6n50 ~/work/arch-pc/lab04 $ gedit lab4.asm
apmaslova@dk6n50 ~/work/arch-pc/lab04 $ nasm -f elf lab4.asm
apmaslova@dk6n50 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  lab4.asm  lab4.o  list.lst  main  obj.o
apmaslova@dk6n50 ~/work/arch-pc/lab04 $
```

Рис. 3.3: Передача программы транслятору

Проверили наличие соответствующего файла.

Создадим объектный файл с именем nobj.o (рис.2.4).



```
apmaslova@dk6n50 ~/work/arch-pc/lab04 $ nasm -o nobj.o -f elf -g -l nlist.lst lab4.asm
apmaslova@dk6n50 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  lab4.asm  lab4.o  list.lst  main  nlist.lst  nobj.o  obj.o
apmaslova@dk6n50 ~/work/arch-pc/lab04 $
```

Рис. 3.4: Компиляция кода

Как мы видим, объектный файл создан.

Скомпонуем объектный файл (рис.3.5).

```

apmaslova@dk6n50 ~/work/arch-pc/lab04 $ ld -m elf_i386 lab4.o -o lab4
apmaslova@dk6n50 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  lab4  lab4.asm  lab4.o  list.lst  main  nlist.lst  nobj.o  obj.o
apmaslova@dk6n50 ~/work/arch-pc/lab04 $

```

Рис. 3.5: Компоновка объектного файла

И создаём исполняемый файл с именем nmain (рис.3.6).

```

apmaslova@dk6n50 ~/work/arch-pc/lab04 $ ld -m elf_i386 nobj.o -o nmain
apmaslova@dk6n50 ~/work/arch-pc/lab04 $ ls
hello  hello.o  lab4.asm  list.lst  nlist.lst  nobj.o
hello.asm  lab4  lab4.o  main  nmain  obj.o
apmaslova@dk6n50 ~/work/arch-pc/lab04 $

```

Рис. 3.6: Создание исполняемого файла

Как мы видим, файл создан.

Запускаем на выполнение созданный исполняемый файл (рис.3.7):

```

apmaslova@dk6n50 ~/work/arch-pc/lab04 $ ./lab4
Маслова Анна
apmaslova@dk6n50 ~/work/arch-pc/lab04 $

```

Рис. 3.7: Запуск программы lab4

На экране видим искомую строку с фамилией и именем.

4 Выводы

Мы освоили процедуры сборки программ, написанных на языке ассемблера NASM. Научились их компилировать.

Список литературы

1. GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>.
2. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>.
3. Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>.
4. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: <http://www.amazon.com/Learning-bash-Shell-Programming-Nutshell/dp/0596009658>.
6. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс,
- 11.