

Шаблон отчёта по лабораторной работе №4

Дисциплина: Архитектура компьютера

Мутале Чали

Содержание

| | | |
|---|---|---|
| 1 | Цель работы | 1 |
| 2 | Задание | 1 |
| 3 | Выполнение лабораторной работы | 1 |
| 4 | Выполнение заданий для самостоятельной работы | 4 |
| 5 | Выводы | 6 |
| 6 | Список литературы | 6 |

1 Цель работы

Цель данной лабораторной работы - освоить процедуры компиляции и сборки программ, написанных на ассемблере NASM.

2 Задание

1. Создание программы Hello world!
2. Работа с транслятором NASM
3. Работа с расширенным синтаксисом командной строки NASM
4. Работа с компоновщиком LD
5. Запуск исполняемого файла

3 Выполнение лабораторной работы

1. Создание программы Hello world!

Создаю каталог для работы с программами на языке ассемблера NASM и перехожу в созданный каталог:

```
(cmutale@cmutale)-[~/../arch-pc/labs/lab01/report]
$ cd ~/work/study/2024-2025/"архитектура компьютера"/arch-pc/labs/lab04
```

Рис 1.0

Создаю текстовый файл с именем hello.asm и проверяю с помощью ls:

```
(cmutale@cmutale)-[~/../архитектура компьютера/arch-pc/labs/lab04]
$ touch hello.asm

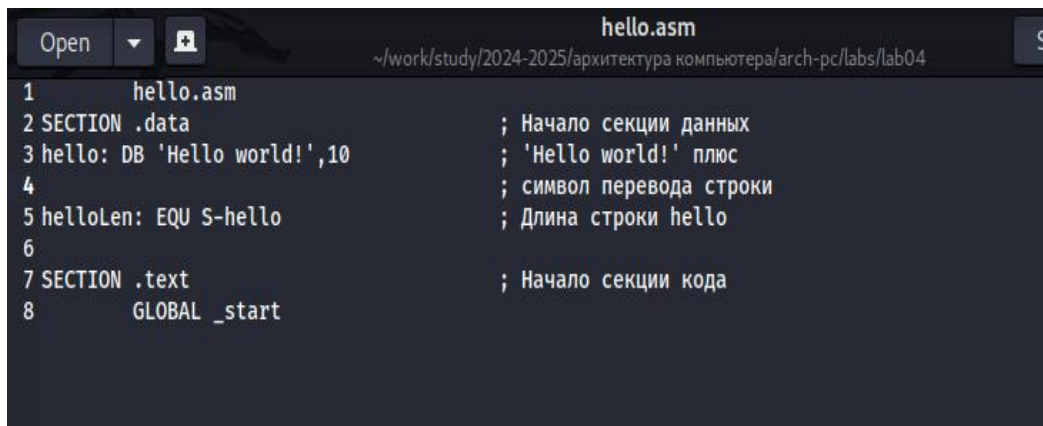
(cmutale@cmutale)-[~/../архитектура компьютера/arch-pc/labs/lab04]
$ ls
hello.asm presentation report
```

Рис 1.1

Открываю созданный файл в текстовом редакторе gedit и вставляю в него программу для вывода "Hello world!":

```
(cmutale@cmutale)-[~/../архитектура компьютера/arch-pc/labs/lab04]
$ gedit hello.asm
```

Рис 1.2



```
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
```

Рис 1.3

2. Работа с транслятором NASM

Превращаю текст программы для вывода "Hello world!" в объектный код с помощью транслятора NASM, используя команду `nasm -f elf hello.asm`. Потом проверяю с помощью ls:

```
(cmutale@cmutale)-[~/../архитектура компьютера/arch-pc/labs/lab04]
$ nasm -f elf hello.asm
```

Рис 2.0

```
(cmutale@cmutale)-[~/../архитектура компьютера/arch-pc/labs/lab04]
$ ls
hello.asm hello.o presentation report
```

Рис 2.1

3. Работа с расширенным синтаксисом командной строки NASM

Ввожу команду, которая скомпилирует файл hello.asm в файл obj.o и проверяю с помощью ls:

```
(cmutale@cmutale)-[~/../архитектура компьютера/arch-pc/labs/lab04]
$ nasm -o obj.o -f elf hello.asm
```

Рис 3.0

4. Работа с компоновщиком LD

Передаю объектный файл hello.o на обработку компоновщику LD, чтобы получить исполняемый файл hello и проверяю с помощью ls:

```
(cmutale@cmutale)-[~/../архитектура компьютера/arch-pc/labs/lab04]
$ ld -m elf_i386 hello.o -o hello

(cmutale@cmutale)-[~/../архитектура компьютера/arch-pc/labs/lab04]
$ ls
hello hello.asm hello.o list.list list.lst obj.o presentation report
```

Рис 4.0

Выполняю следующую команду (рис. 4.1). Исполняемый файл будет иметь имя main, т.к. после ключа -o было задано значение main. Объектный файл, из которого собран этот исполняемый файл, имеет имя obj.o:

```
(cmutale@cmutale)-[~/../архитектура компьютера/arch-pc/labs/lab04]
$ ld -m elf_i386 obj.o -o main

(cmutale@cmutale)-[~/../архитектура компьютера/arch-pc/labs/lab04]
$ ls
hello hello.o list.lst obj.o report
hello.asm list.list main presentation
```

Рис 4.1

5. Запуск исполняемого файла

Запускаю на выполнение созданный файл hello:

```
(cmutale@cmutale)-[~/../архитектура компьютера/arch-pc/labs/lab04]
$ ./hello
Hello world!
```

Рис 5.0

4 Выполнение заданий для самостоятельной работы

С помощью `cp` создаю копию файла `hello.asm` с именем `lab4.asm`:

```
(cmutale@cmutale)-[~/../архитектура компьютера/arch-pc/labs/lab04]
$ cp hello.asm lab4.asm

(cmutale@cmutale)-[~/../архитектура компьютера/arch-pc/labs/lab04]
$ ls
hello    hello.o  list.list  main  presentation
hello.asm lab4.asm list.lst  obj.o  report
```

Рис 6.0

С помощью `gedit` открываю файл `lab5.asm` и вношу изменения в программу так, чтобы она выводила имя и фамилию:

```
(cmutale@cmutale)-[~/../архитектура компьютера/arch-pc/labs/lab04]
$ gedit lab4.asm
```

Рис 6.1

```
Open  lab4.asm
~/work/study/2024-2025/архитектура компьютера/
1 ; hello.asm
2 SECTION .data                ; Начало секции данных
3 hello: DB 'Мутале Чали',10    ; 'фамилия имя' плюс
4                               ; символ перевода строки
5 helloLen: EQU $-hello        ; Длина строки hello
6
7 SECTION .text                ; Начало секции кода
8 GLOBAL _start
9
```

Рис 6.2

Компилирую текст программы в объектный файл и проверяю с помощью `ls`, что файл `lab4.o` создан:

```
(cmutale@cmutale)-[~/../архитектура компьютера/arch-pc/labs/lab04]
$ nasm -f elf lab4.asm

(cmutale@cmutale)-[~/../архитектура компьютера/arch-pc/labs/lab04]
$ ls
hello    hello.o  lab4.o  list.lst  obj.o  report
hello.asm lab4.asm list.list  main  presentation
```

Рис 6.3

Передаю объектный файл lab4.o на обработку компоновщику LD, чтобы получить файл lab4:

```
(cmutale@cmutale)~/../архитектура компьютера/arch-pc/labs/lab04]
$ ld -m elf_i386 lab4.o -o lab4

(cmutale@cmutale)~/../архитектура компьютера/arch-pc/labs/lab04]
$ ls
hello    hello.o  lab4.asm  list.list  main  presentation
hello.asm lab4     lab4.o    list.lst   obj.o  report
```

Рис 6.4

Запускаю исполняемый файл lab4:

```
(cmutale@cmutale)~/../архитектура компьютера/arch-pc/labs/lab04]
$ ls
hello    hello.o  lab4.asm  list.list  main  presentation
hello.asm lab4     lab4.o    list.lst   obj.o  report
```

Рис 6.5

Копирую файлы hello.asm и lab4.asm в локальный репозиторий с помощью cp:

```
(cmutale@cmutale)~/work/arch-pc/lab04]
$ cp hello.asm ~/work/study/2024-2025/"архитектура компьютера"/arch-pc/labs/
lab04

(cmutale@cmutale)~/work/arch-pc/lab04]
$ cp lab4.asm ~/work/study/2024-2025/"архитектура компьютера"/arch-pc/labs/
lab04

(cmutale@cmutale)~/../архитектура компьютера/arch-pc/labs/lab03]
$ cd ~/work/study/2024-2025/"архитектура компьютера"/arch-pc/labs/lab04

(cmutale@cmutale)~/../архитектура компьютера/arch-pc/labs/lab04]
$ ls
hello    hello.o  lab4.asm  list.list  main  presentation
hello.asm lab4     lab4.o    list.lst   obj.o  report

(cmutale@cmutale)~/../архитектура компьютера/arch-pc/labs/lab04]
$
```

Рис 6.6

С помощью команд git add . и git commit добавляю файлы на GitHub:

```

(cmutale@cmutale)-[~/../архитектура компьютера/arch-pc/labs/lab04]
$ git add .

(cmutale@cmutale)-[~/../архитектура компьютера/arch-pc/labs/lab04]
$ git commit -m "Add existing file"
[master a3005dc] Add existing file
10 files changed, 82 insertions(+)
create mode 100755 labs/lab04/hello
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/hello.o
create mode 100755 labs/lab04/lab4
create mode 100644 labs/lab04/lab4.asm
create mode 100644 labs/lab04/lab4.o
create mode 100644 labs/lab04/list.list
create mode 100644 labs/lab04/list.lst
create mode 100755 labs/lab04/main
create mode 100644 labs/lab04/obj.o

```

Рис 6.7

Отправляю файлы на сервер с помощью команды git push:

```

(cmutale@cmutale)-[~/../архитектура компьютера/arch-pc/labs/lab04]
$ git push -f origin master
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 4 threads
Compressing objects: 100% (13/13), done.
Writing objects: 100% (13/13), 3.29 KiB | 420.00 KiB/s, done.
Total 13 (delta 7), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (7/7), completed with 2 local objects.
To github.com:cmutale-skept/study_2024-2025_arh-pc.git
bb34099..a3005dc master -> master

```

Рис 6.8

5 Выводы

При выполнении данной лабораторной работы я освоила процедуры компиляции и сборки программ, написанных на ассемблере NASM.

6 Список литературы

Архитектура ЭВМ