

Отчёт по лабораторной работе №4

Дисциплина: Архитектура компьютера

Чистов Даниил Максимович

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
3.1	Программа Hello world!	6
3.2	Транслятор NASM	7
3.3	Расширенный синтаксис командной строки NASM	8
3.4	Компоновщик LD	8
3.5	Запуск исполняемого файла	9
4	Задание для самостоятельной работы	10
5	Выводы	14
	Список литературы	15

Список иллюстраций

3.1	Создание директории lab04	6
3.2	Создание файла hello.asm	6
3.3	Редактирование файла hello.asm	7
3.4	Создание файла hello.o	7
3.5	Компиляция командой nasm	8
3.6	Успешное выполнение команды	8
3.7	Успешное выполнение команды	9
3.8	Успешное выполнение команды	9
4.1	Копирование файлов	10
4.2	Редактирование файла	11
4.3	Преобразование файла	11
4.4	Преобразование файла	12
4.5	Успешное выполнение программы	12
4.6	Копирование файлов	12
4.7	Отчёт об изменениях для системы Github	13
4.8	Загрузка изменений на сервер	13

1 Цель работы

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

2 Задание

1. Выполнение лабораторной работы
2. Задание для самостоятельной работы

3 Выполнение лабораторной работы

3.1 Программа Hello world!

Перехожу в папку репозитория и создаю директорию под названием “lab04” (рис. 3.1).

```
[dmchistov@fedora ~]$ cd work/study/2023-2024/Архитектура\ компьютера/study_2023-2024_arh-pc/
[dmchistov@fedora study_2023-2024_arh-pc]$ ls
CHANGELOG.md  COURSE  LICENSE  prepare  README.en.md  README.md
config        labs    Makefile  presentation  README.git-flow.md  template
[dmchistov@fedora study_2023-2024_arh-pc]$ mkdir -p lab04
[dmchistov@fedora study_2023-2024_arh-pc]$ ls
CHANGELOG.md  lab04    Makefile  README.en.md  template
config        labs     prepare  README.git-flow.md
COURSE        LICENSE  presentation  README.md
[dmchistov@fedora study_2023-2024_arh-pc]$
```

Рис. 3.1: Создание директории lab04

Перехожу в созданную мной директорию и создаю файл “hello.asm” (рис. 3.2).

```
[dmchistov@fedora lab04]$ cd ..
[dmchistov@fedora study_2023-2024_arh-pc]$ cd lab04
[dmchistov@fedora lab04]$ touch hello.asm
[dmchistov@fedora lab04]$ ls
hello.asm
[dmchistov@fedora lab04]$
```

Рис. 3.2: Создание файла hello.asm

Открываю файл “hello.asm” редактором “mcedit” и вставляю в него код из документа по лабораторной работы (рис. 3.3).

```
dmchistov@fedora:~/work/study/2023-2024/Архитектура компьюте...
hello.asm [----] 21 L: [ 1+ 7 8/ 20] *(302 / 861b) 0010 0x00A [*] [X]
; hello.asm
SECTION .data ; Начало секции данных
<----->hello: DB 'Hello world!',10 ; 'Hello world!' плюс
<-----><-----><-----><-----> ; символ перевода строки
<----->helloLen: EQU $-hello ; Длина строки hello

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

Рис. 3.3: Редактирование файла hello.asm

3.2 Транслятор NASM

Компилирую файл “hello.asm”, тем самым создаю отдельный файл под названием “hello.o”. Компиляция прошла без ошибок, в этом можно также убедиться, написав команду ls и увидеть новый файл (рис. 3.4).

```
[dmchistov@fedora lab04]$ nasm -f elf hello.asm
[dmchistov@fedora lab04]$ ls
hello.asm hello.o
[dmchistov@fedora lab04]$
```

Рис. 3.4: Создание файла hello.o

3.3 Расширенный синтаксис командной строки NASM

Выполняю команду “`nasm -o obj.o -f elf -g -l list.lst hello.asm`”, которая скомпилирует несколько требуемых файлов, после чего пишу команду `ls`, чтобы проверить успешность выполнения команды (рис. 3.5).

```
[dmchistov@10 lab04]$ ls
hello.asm  hello.o
[dmchistov@10 lab04]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
[dmchistov@10 lab04]$ ls
hello.asm  hello.o  list.lst  obj.o
[dmchistov@10 lab04]$
```

Рис. 3.5: Компиляция командой `nasm`

3.4 Компоновщик LD

Передаю объектный файл на обработку компоновщику командой `ld`, после чего проверяю, создался ли файл командой `ls`. (рис. 3.6).

```
hello.asm  hello.o  list.lst  obj.o
[dmchistov@10 lab04]$ ld -m elf_i386 hello.o -o hello
[dmchistov@10 lab04]$ ls
hello  hello.asm  hello.o  list.lst  obj.o
[dmchistov@10 lab04]$
```

Рис. 3.6: Успешное выполнение команды

Затем выполняю следующую команду “`ld -m elf_i386 obj.o -o main`”. У созданного файла будет имя “`main`”, это можно подтвердить, прописав команду `ls` (рис. 3.7).


```
hello hello.asm hello.o list.lst obj.o
[dmchistov@10 lab04]$ ld -m elf_i386 obj.o -o main
[dmchistov@10 lab04]$ ls
hello hello.asm hello.o list.lst main obj.o
[dmchistov@10 lab04]$
```

Рис. 3.7: Успешное выполнение команды

3.5 Запуск исполняемого файла

Остаётся лишь запустить сгенерированный файл `hello`. Достаточно написать команду `./hello` (рис. 3.8).

```
hello hello.asm hello.o list.lst main obj.o
[dmchistov@10 lab04]$ ./hello
Hello world!
[dmchistov@10 lab04]$
```

Рис. 3.8: Успешное выполнение команды

Код успешно был выполнен!

4 Задание для самостоятельной работы

Копирую файл `hello.asm` и называю его `lab04.asm` командой `cp` (рис. 4.1).

```
Hello world!  
[dmchistov@10 lab04]$ cp hello.asm lab04.asm  
[dmchistov@10 lab04]$ ls  
hello  hello.asm  hello.o  lab04.asm  list.lst  main  obj.o  
[dmchistov@10 lab04]$
```

Рис. 4.1: Копирование файлов

Вношу изменения в файл с помощью утилиты `msedit`, чтобы при выполнении выводилось моё имя и фамилия (рис. 4.2).

```

lab04.asm      [-M--] 32 L:[ 1+ 2 3/ 20] *(104 / 874b) 0039 0x027 [*][X]
; hello.asm
SECTION .data ; Начало секции данных
<----->hello: DB 'Чистов Даниил',10 ; 'Hello world!' плюс
<-----><-----><-----><-----> ; символ перевода строки
<----->helloLen: EQU $-hello ; Длина строки hello

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

1Помощь 2Сох~ть 3Блок 4Замена 5Копия 6Пер~ть 7Поиск 8Уда~ть 9МенюМС10Выход

```

Рис. 4.2: Редактирование файла

Преобразовываю сгенерированный файл в объект утилитой nasm, проверяю успешное выполнение команды утилитой ls (рис. 4.3).

```

hello hello.asm hello.o lab04.asm list.lst main obj.o
[dmchistov@10 lab04]$ nasm -f elf lab04.asm
[dmchistov@10 lab04]$ ls
hello hello.asm hello.o lab04.asm lab04.o list.lst main obj.o
[dmchistov@10 lab04]$

```

Рис. 4.3: Преобразование файла

Отправка файл преобразованный файл объект компоновщику на преобразование в готовую программу (рис. 4.4).

```

hello hello.asm hello.o lab04.asm lab04.o list.lst main obj.o
[dmchistov@10 lab04]$ ld -m elf_i386 lab04.o -o lab04
[dmchistov@10 lab04]$ ls
hello hello.asm hello.o lab04 lab04.asm lab04.o list.lst main obj.o
[dmchistov@10 lab04]$

```

Рис. 4.4: Преобразование файла

Запускаю готовый файл (рис. 4.5).

```

[dmchistov@10 lab04]$ ls
hello hello.asm hello.o lab04 lab04.asm lab04.o list.lst main obj.o
[dmchistov@10 lab04]$ ./lab04
Чистов Даниил
[dmchistov@10 lab04]$

```

Рис. 4.5: Успешное выполнение программы

Копирую файлы hello.asm и lab04.asm в директорию labs/lab04 (рис. 4.6).

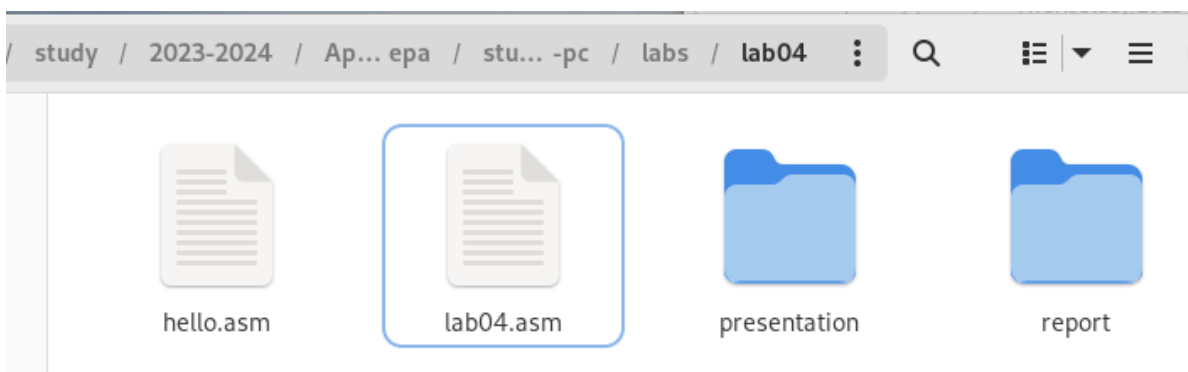


Рис. 4.6: Копирование файлов

Вношу изменения в репозиторий Github, сначала заявляю о них командой `git add`, после чего заявляю об отправке изменений командой `git commit`, называя их “submitting .asm files to lab04” (рис. 4.7).

```

[dmchistov@10 lab04]$ git add .
[dmchistov@10 lab04]$ git commit -m "Submitting .asm files to lab04"
[master fad531e] Submitting .asm files to lab04
 9 files changed, 59 insertions(+)
 create mode 100755 lab04/hello
 create mode 100644 lab04/hello.asm
 create mode 100644 lab04/hello.o
 create mode 100755 lab04/lab04
 create mode 100644 lab04/lab04.asm
 create mode 100644 lab04/lab04.o
 create mode 100644 lab04/list.lst
 create mode 100755 lab04/main
 create mode 100644 lab04/obj.o
[dmchistov@10 lab04]$

```

Рис. 4.7: Отчёт об изменениях для системы Github

Отправляю изменения на сервер командой git push (рис. 4.8).

```

[dmchistov@10 lab04]$ git push
Перечисление объектов: 13, готово.
Подсчет объектов: 100% (13/13), готово.
Сжатие объектов: 100% (12/12), готово.
Запись объектов: 100% (12/12), 3.20 КиБ | 656.00 КиБ/с, готово.
Всего 12 (изменений 6), повторно использовано 0 (изменений 0), повторно испо
вано пакетов 0
remote: Resolving deltas: 100% (6/6), completed with 1 local object.
To github.com:danchist/study_2023-2024_arh-pc
   ba383ae..fad531e  master -> master
[dmchistov@10 lab04]$

```

Рис. 4.8: Загрузка изменений на сервер

5 Выводы

При выполнении работы, мной были получены навыки работы с программами, написанные на языке NASM, освоение их компиляции и сборки.

Список литературы

Лабораторная работы №4