

Отчет по лабораторной работе №4

Дисциплина: архитектура компьютера

Ардеев Никита Евгеньевич НММбд-01-23

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	10

Список иллюстраций

2.1	Создание файла hello.asm	6
2.2	Ввод текста программы	6
2.3	Запуск транслятора NASM	7
2.4	Компиляция исходного файла	7
2.5	Обработка компоновщиком объектного файла	7
2.6	Обработка компоновщиком объектного файла	7
2.7	Запуск исполняемого файла	8
2.8	Создание копии файла hello.asm с именем lab4.asm	8
2.9	Внесение изменений в текст	8
2.10	Трансляция текста программы, компоновка и запуск	9
2.11	Копирование в локальный репозиторий	9
2.12	Загрузка файлов на гитхаб	9

Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

Рассмотрю пример простой программы на языке ассемблера NASM. Традиционно первая программа выводит приветственное сообщение Hello world! на экран.

Создал каталог для работы с программами на языке ассемблера NASM, перешел в него, в нем создал текстовый файл с именем hello.asm (рис. 2.1).

```
neardeev@dk5n56 ~ $ mkdir -p ~/work/arch-pc/lab04
neardeev@dk5n56 ~ $ cd ~/work/arch-pc/lab04
neardeev@dk5n56 ~/work/arch-pc/lab04 $ touch hello.asm
neardeev@dk5n56 ~/work/arch-pc/lab04 $
```

Рис. 2.1: Создание файла hello.asm

Открыл файл помощью текстового редактора gedit и ввел в него текст, представленный в лабораторной работе (рис. 2.2).

```
neardeev@dk5n56 ~/work/arch-pc/lab04 $ gedit hello.asm
*hello.asm
~/work/arch-pc/lab04
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. 2.2: Ввод текста программы

С помощью NASM превратил текст программы в объектный код и с помощью команды `ls` проверил, что объектный файл был создан: он имеет имя `hello.o` (рис. 2.3).

```
neardeev@dk5n56 ~/work/arch-pc/lab04 $ nasm -f elf hello.asm
neardeev@dk5n56 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o
neardeev@dk5n56 ~/work/arch-pc/lab04 $
```

Рис. 2.3: Запуск транслятора NASM

Выполнил следующую команду: `nasm -o obj.o -f elf -g -l list.lst hello.asm` и с `ls` проверил, что файлы были созданы (рис. 2.4).

```
neardeev@dk5n56 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
neardeev@dk5n56 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o  list.lst  obj.o
neardeev@dk5n56 ~/work/arch-pc/lab04 $
```

Рис. 2.4: Компиляция исходного файла

Чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику: `ld -m elf_i386 hello.o -o hello`, с помощью команды `ls` проверил, что исполняемый файл `hello` был создан (рис. 2.5).

```
neardeev@dk5n56 ~/work/arch-pc/lab04 $ ld -m elf_i386 hello.o -o hello
neardeev@dk5n56 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  list.lst  obj.o
```

Рис. 2.5: Обработка компоновщиком объектного файла

Выполнил следующую команду: `ld -m elf_i386 obj.o -o main` и с помощью `ls` выяснил, что исполняемый файл будет иметь имя `main`, а объектный файл - `obj.o` (рис. 2.6).

```
hello  hello.asm  hello.o  list.lst  obj.o
neardeev@dk5n56 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
neardeev@dk5n56 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
neardeev@dk5n56 ~/work/arch-pc/lab04 $
```

Рис. 2.6: Обработка компоновщиком объектного файла

Запустил на выполнение исполняемый файл с помощью команды `./hello` (рис. 2.7).

```
neardeev@dk5n56 ~/work/arch-pc/lab04 $ ./hello
Hello world!
neardeev@dk5n56 ~/work/arch-pc/lab04 $
```

Рис. 2.7: Запуск исполняемого файла

#Задание для самостоятельной работы

В каталоге `~/work/arch-pc/lab04` с помощью команды `cp` создал копию файла `hello.asm` с именем `lab4.asm` (рис. 2.8).

```
neardeev@dk5n56 ~/work/arch-pc/lab04 $ cp hello.asm lab4.asm
neardeev@dk5n56 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  lab4.asm  list.lst  main  obj.o
neardeev@dk5n56 ~/work/arch-pc/lab04 $
```

Рис. 2.8: Создание копии файла `hello.asm` с именем `lab4.asm`

С помощью текстового редактора `gedit` внес изменения в текст программы в файле `lab4.asm` так, чтобы вместо `Hello world!` на экран выводилась строка с моими фамилией и именем. (рис. 2.9).

```
hello  hello.asm  hello.o  lab4.asm  list.lst  main  obj.o
neardeev@dk5n56 ~/work/arch-pc/lab04 $ gedit lab4.asm\
>
Открыть  *lab4.asm  Сохранить
~/work/arch-pc/lab04
1; hello.asm
2SECTION .data ; Начало секции данных
3hello: DB 'Никита Ардеев!',10 ; 'Hello world!' плюс
4; символ перевода строки
5hello: equ FILL $-hello ; Длина строки hello
```

Рис. 2.9: Внесение изменений в текст

Оттранслировал полученный текст программы `lab4.asm` в объектный файл. Выполнял компоновку объектного файла и запустил получившийся исполняемый файл (рис. 2.10).


```
hello hello.asm hello.o lab4.asm list.lst main obj.o
neardeev@dk5n56 ~/work/arch-pc/lab04 $ gedit lab4.asm\
>
neardeev@dk5n56 ~/work/arch-pc/lab04 $ nasm -f elf lab4.asm
neardeev@dk5n56 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst lab4.asm
neardeev@dk5n56 ~/work/arch-pc/lab04 $ d -m elf_i386 lab4.o -o lab4
bash: d: команда не найдена
neardeev@dk5n56 ~/work/arch-pc/lab04 $ ld -m elf_i386 hello.o -o hello
neardeev@dk5n56 ~/work/arch-pc/lab04 $ ./lab4
bash: ./lab4: Нет такого файла или каталога
neardeev@dk5n56 ~/work/arch-pc/lab04 $ ld -m elf_i386 lab4.o -o lab4
neardeev@dk5n56 ~/work/arch-pc/lab04 $ ./lab4
Никита Ардеев!
neardeev@dk5n56 ~/work/arch-pc/lab04 $
```

Рис. 2.10: Трансляция текста программы, компоновка и запуск

Скопировал файлы hello.asm и lab4.asm в мой локальный репозиторий в каталог ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/ (рис. 2.11).

```
neardeev@dk5n56 ~/work/arch-pc/lab04 $ cp lab4.asm ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/
neardeev@dk5n56 ~/work/arch-pc/lab04 $ cp hello.asm ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/
neardeev@dk5n56 ~/work/arch-pc/lab04 $ ls ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/
hello.asm lab4.asm presentation report
neardeev@dk5n56 ~/work/arch-pc/lab04 $
```

Рис. 2.11: Копирование в локальный репозиторий

Загрузил файлы на Github. (рис. 2.12).

```
и репозиторий существует.
neardeev@dk5n56 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ git push
ssh: Could not resolve hostname github.com: Device or resource busy
fatal: Не удалось прочитать из внешнего репозитория.

Удостоверьтесь, что у вас есть необходимые права доступа
и репозиторий существует.
neardeev@dk5n56 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ git push
Перечисление объектов: 28, готово.
Подсчет объектов: 100% (28/28), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (23/23), готово.
Запись объектов: 100% (23/23), 584.87 КиБ | 14.26 МиБ/с, готово.
Всего 23 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 3 local objects.
To github.com:neardeev/study_2023-2024_arh-pc.git
   b626263..1e6bae9 master -> master
neardeev@dk5n56 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $
```

Рис. 2.12: Загрузка файлов на гитхаб

3 Выводы

Получил базовые навыки в работе с ассемблером.