

Отчёт по лабораторной работе №4

Кулакова Надежда Сергеевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выполнение заданий для самостоятельной работы	9
4	Выводы	11

Список иллюстраций

2.1	Создадим каталог для работы с программами на языке ассемблера NASM перейдем в него и создадим текстовый файл с именем hello.asm	6
2.2	Откроем этот файл с помощью текстового редактора	6
2.3	Введем в него текст	7
2.4	Скомпилируем данный текст и проверим, что нужный файл был создан с помощью команды ls.	7
2.5	Скомпилируем исходный файл hello.asm в obj.o и проверим что файлы были созданы.	7
2.6	Передадим объектный файл на обработку компоновщику и проверим что исполняемый файл был создан.	8
2.7	Запустим на выполнение созданный исполняемый файл	8
3.1	С помощью команды cp создадим копию файла hello.asm с именем lab4.asm и проверим, что нужный файл был создан с помощью команды ls	9
3.2	Внесем изменения в текст программы в файле lab4.asm так, чтобы вместо Hello world! на экран выводилась строка с фамилией и именем	9
3.3	Оттранслируем полученный текст программы lab4.asm в объектный файл, выполним компоновку объектного файла и запустим получившийся исполняемый файл	9
3.4	Скопируем файлы hello.asm и lab4.asm в локальный репозиторий в папку lab04	10
3.5	Загрузим файлы на Github.	10

Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

```
nskulakova@dk8n81 ~ $ mkdir -p ~/work/arch-pc/lab04  
nskulakova@dk8n81 ~ $ cd ~/work/arch-pc/lab04  
nskulakova@dk8n81 ~/work/arch-pc/lab04 $ touch hello.asm
```

Рис. 2.1: Создадим каталог для работы с программами на языке ассемблера NASM
перейдем в него и создадим текстовый файл с именем hello.asm

```
nskulakova@dk8n81 ~/work/arch-pc/lab04 $ gedit hello.asm
```

Рис. 2.2: Откроем этот файл с помощью текстового редактора

```

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 ;

```

Рис. 2.3: Введем в него текст

```

nskulakova@dk8n81 ~/work/arch-pc/lab04 $ nasm -f elf hello.asm
nskulakova@dk8n81 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o

```

Рис. 2.4: Скомпилируем данный текст и проверим, что нужный файл был создан с помощью команды ls.

```

nskulakova@dk8n81 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst
hello.asm
nskulakova@dk8n81 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o  list.lst  obj.o

```

Рис. 2.5: Скомпилируем исходный файл hello.asm в obj.o и проверим что файлы были созданы.

```
nskulakova@dk8n81 ~/work/arch-pc/lab04 $ ld -m elf_i386 hello.o -o hello
nskulakova@dk8n81 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
```

Рис. 2.6: Передадим объектный файл на обработку компоновщику и проверим что исполняемый файл был создан.

```
nskulakova@dk8n81 ~/work/arch-pc/lab04 $ ./hello
Hello world!
```

Рис. 2.7: Запустим на выполнение созданный исполняемый файл

3 Выполнение заданий для самостоятельной работы

```
nskulakova@dk8n81 ~/work/arch-pc/lab04 $ cp hello.asm lab4.asm
nskulakova@dk8n81 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  lab4.asm  list.lst  main  obj.o
```

Рис. 3.1: С помощью команды `cp` создадим копию файла `hello.asm` с именем `lab4.asm` и проверим, что нужный файл был создан с помощью команды `ls`

```
nskulakova@dk8n81 ~/work/arch-pc/lab04 $ gedit lab4.asm
```

Рис. 3.2: Внесем изменения в текст программы в файле `lab4.asm` так, чтобы вместо `Hello world!` на экран выводилась строка с фамилией и именем

```
nskulakova@dk8n81 ~/work/arch-pc/lab04 $ nasm -f elf lab4.asm
nskulakova@dk8n81 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  lab4.asm  lab4.o  list.lst  main  obj.o
nskulakova@dk8n81 ~/work/arch-pc/lab04 $ ld -m elf_i386 lab4.o -o lab4
nskulakova@dk8n81 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  lab4  lab4.asm  lab4.o  list.lst  main  obj.o
nskulakova@dk8n81 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst lab4.asm
nskulakova@dk8n81 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  lab4  lab4.asm  lab4.o  list.lst  main  obj.o
nskulakova@dk8n81 ~/work/arch-pc/lab04 $ ./lab4
Кулакова Надежда
```

Рис. 3.3: Оттранслируем полученный текст программы `lab4.asm` в объектный файл, выполним компоновку объектного файла и запустим получившийся исполняемый файл

```

nskulakova@dk8n81 ~/work/arch-pc/lab04 $ cp lab4.asm ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab04/
nskulakova@dk8n81 ~/work/arch-pc/lab04 $ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab04/
nskulakova@dk8n81 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04 $ ls
hello.asm lab4.asm presentation report
nskulakova@dk8n81 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04 $ cd
nskulakova@dk8n81 ~ $ cd ~/work/arch-pc/lab04
nskulakova@dk8n81 ~/work/arch-pc/lab04 $ cp hello.asm ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab04/
nskulakova@dk8n81 ~/work/arch-pc/lab04 $ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab04/
nskulakova@dk8n81 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04 $ ls
hello.asm lab4.asm presentation report

```

Рис. 3.4: Скопируем файлы hello.asm и lab4.asm в локальный репозиторий в папку lab04

```

nskulakova@dk8n81 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ git add .
nskulakova@dk8n81 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ git commit -am 'feat(main): add files lab-4'
[master 3f15c5f] feat(main): add files lab-4
2 files changed, 32 insertions(+)
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/lab4.asm
nskulakova@dk8n81 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ git push
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 979 байтов | 979.00 КиБ/с, готово.
Всего 6 (изменений 3), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.

```

Рис. 3.5: Загрузим файлы на Github.

4 Выводы

Мы освоили процедуры компиляции и сборки программ, написанных на ассемблере NASM.