

Отчет по лабораторной работе № 5

Дисциплина: Архитектура компьютера

Никулина Ксения Ильинична

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	6
4	Выполнение лабораторной работы	7
5	Выполнение самостоятельной работы	10
6	Выводы	12

Список иллюстраций

4.1	Создание каталога	7
4.2	Переход в каталог	7
4.3	Создание текстового файла	7
4.4	Текст	8
4.5	Команда	8
4.6	Проверка созданных файлов	8
4.7	Передача файла на обработку комповщику	8
4.8	Команда	8
4.9	Запуск исполняемого файла	9
5.1	Создание копии	10
5.2	Внесение изменений в текстовом редакторе	10
5.3	Вывод имени и фамилии	11
5.4	Копирование файлов в каталог	11
5.5	Загрузка файлов на Github	11

1 Цель работы

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

2 Задание

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

3 Теоретическое введение

Язык ассемблера88 (assembly language, сокращённо asm) — машинно-ориентированный язык низкого уровня. Можно считать, что он больше любых других языков приближен к архитектуре ЭВМ и её аппаратным возможностям, что позволяет получить к ним более полный доступ, нежели в языках высокого уровня, таких как C/C++, Perl, Python и пр. Заметим, что получить полный доступ к ресурсам компьютера в современных архитектурах нельзя, самым низким уровнем работы прикладной программы является обращение напрямую к ядру. NASM** — это открытый проект ассемблера, версии которого доступны под различные операционные системы и который позволяет получать объектные файлы для этих систем. В NASM используется Intel-синтаксис и поддерживаются инструкции x86-64 Программа на языке ассемблера также может содержать директивы — инструкции, не переводящиеся непосредственно в машинные команды, а управляющие работой транслятора. Например, директивы используются для определения данных (констант и переменных) и обычно пишутся большими буквами

4 Выполнение лабораторной работы

1. Создала каталог для работы с программами на языке ассемблера NASM (рис. 4.1)

```
kinikulina@dk8n72 ~ $ mkdir ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/lab05
kinikulina@dk8n72 ~ $
```

Рис. 4.1: Создание каталога

2. Перешла в созданный каталог (рис. 4.2)

```
kinikulina@dk8n72 ~ $ mkdir ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/lab05
kinikulina@dk8n72 ~ $ cd ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/lab05
kinikulina@dk8n72 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $
```

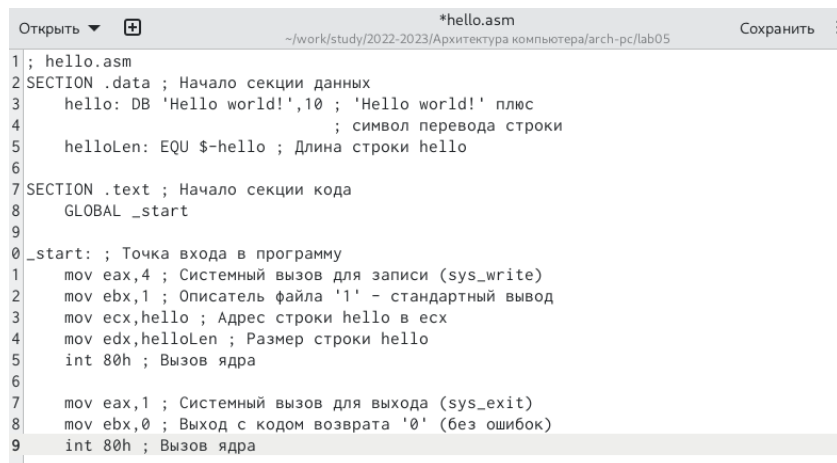
Рис. 4.2: Переход в каталог

3. Создала текстовый файл с именем hello.asm (рис. 4.3)

```
kinikulina@dk8n72 ~ $ mkdir ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/lab05
kinikulina@dk8n72 ~ $ cd ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/lab05
kinikulina@dk8n72 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ touch hello.asm
kinikulina@dk8n72 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $
```

Рис. 4.3: Создание текстового файла

4. Открыла этот файл с помощью текстового редактора gedit и ввела следующий текст (рис. 4.4)



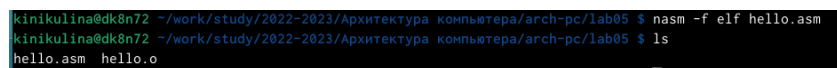
```

Открыть + *hello.asm Сохранить
~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05
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
9
10 _start: ; Точка входа в программу
11     mov eax,4 ; Системный вызов для записи (sys_write)
12     mov ebx,1 ; Описатель файла '1' - стандартный вывод
13     mov ecx,hello ; Адрес строки hello в есх
14     mov edx,helloLen ; Размер строки hello
15     int 80h ; Вызов ядра
16
17     mov eax,1 ; Системный вызов для выхода (sys_exit)
18     mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
19     int 80h ; Вызов ядра

```

Рис. 4.4: Текст

5. Для компиляции текста программы «Hello World» написала (рис. 4.5)



```

kinikulina@dk8n72 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ nasm -f elf hello.asm
kinikulina@dk8n72 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ls
hello.asm hello.o

```

Рис. 4.5: Команда

6. Скомпилировала исходный файл hello.asm в obj.o и с помощью команды ls проверила, что файлы были созданы (рис. 4.6)



```

kinikulina@dk8n72 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
kinikulina@dk8n72 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ls
hello.asm hello.o list.lst obj.o

```

Рис. 4.6: Проверка созданных файлов

7. Объектный файл передала на обработку компоновщику (рис. 4.7)



```

kinikulina@dk8n72 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ld -m elf_i386 hello.o -o hello
kinikulina@dk8n72 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ls
hello hello.asm hello.o list.lst obj.o

```

Рис. 4.7: Передача файла на обработку компоновщику

8. Выполнила следующую команду (рис. 4.8)



```

kinikulina@dk8n72 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ./hello
kinikulina@dk8n72 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ls
hello hello.asm hello.o list.lst main obj.o

```

Рис. 4.8: Команда

9. Запустила на выполнение созданный исполняемый файл, находящийся в текущем каталоге, набрав в командной строке: (рис. 4.9)

A screenshot of a terminal window with a black background. The prompt is 'kinikulina@dk8n72 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 \$'. The command entered is './hello'. The output is 'Hello world!' on the next line.

```
kinikulina@dk8n72 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ./hello
Hello world!
```

Рис. 4.9: Запуск исполняемого файла

5 Выполнение самостоятельной работы

1. В каталоге ~/work/arch-pc/lab05 с помощью команды `cp` создала копию файла `hello.asm` с именем `lab5.asm` (рис. 5.1)

```
kinikulina@dk8n72 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ cp hello.asm lab05.asm
kinikulina@dk8n72 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ls
hello  hello.asm  hello.o  lab05.asm  list.lst  main  obj.o
```

Рис. 5.1: Создание копии

2. С помощью текстового редактора внесла изменения в текст программы в файле `lab5.asm` так, чтобы вместо `Hello world!` на экран выводилась строка с моими фамилией и именем (рис. 5.2)

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

Рис. 5.2: Внесение изменений в текстовом редакторе

3. Оттранслировала полученный текст программы `lab5.asm` в объектный файл. Выполнила компоновку объектного файла и запустила получившийся исполняемый файл. (рис. 5.3)

```

kinikulina@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ nasm -f elf lab05.asm
kinikulina@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ nasm -o obj.o -f elf -g -l list.lst lab05.asm
kinikulina@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ld -m elf_i386 lab05.o -o lab05
kinikulina@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ld -m elf_i386 obj.o -o main
kinikulina@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ./lab05
Nikulina Kseniia

```

Рис. 5.3: Вывод имени и фамилии

4. Скопировала файлы hello.asm и lab5.asm в мой локальный репозиторий в каталог ~/work/study/2022-2023/“Архитектура компьютера”/arch-pc/labs/lab05/. (рис. 5.4)

```

kinikulina@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ cp hello.asm /afs/dk.sci.gpu.edu.ru/home/k/kinikulina/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05/
kinikulina@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ cp lab05.asm /afs/dk.sci.gpu.edu.ru/home/k/kinikulina/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05/
kinikulina@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ cd ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05/
hello.asm lab05.asm presentation_report
kinikulina@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ ls

```

Рис. 5.4: Копирование файлов в каталог

5. Загрузила файлы на Github(рис. 5.5)

```

kinikulina@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ git add .
kinikulina@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ git commit -am 'lab05.asm hello.asm'
[master ba79d53] lab05.asm hello.asm
20 files changed, 38 insertions(+)
create mode 100644 labs/lab05/hello.asm
create mode 100644 labs/lab05/lab05.asm
create mode 100644 labs/lab05/report/image/1.png
create mode 100644 labs/lab05/report/image/10.png
create mode 100644 labs/lab05/report/image/11.png
create mode 100644 labs/lab05/report/image/12.png
create mode 100644 labs/lab05/report/image/13.png
create mode 100644 labs/lab05/report/image/14.png
create mode 100644 labs/lab05/report/image/15.png
create mode 100644 labs/lab05/report/image/16.png
create mode 100644 labs/lab05/report/image/17.png
create mode 100644 labs/lab05/report/image/2.png
create mode 100644 labs/lab05/report/image/3.png
create mode 100644 labs/lab05/report/image/4.png
create mode 100644 labs/lab05/report/image/5.png
create mode 100644 labs/lab05/report/image/6.png
create mode 100644 labs/lab05/report/image/7.png
create mode 100644 labs/lab05/report/image/8.png
create mode 100644 labs/lab05/report/image/9.png
delete mode 100644 labs/lab05/report/image/placeimg_800_600_tech.jpg
kinikulina@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ git push
Перечисление объектов: 30, готово.
Подсчет объектов: 100% (30/30), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (25/25), готово.
Запись объектов: 100% (25/25), 559,27 КиБ | 4,86 МБ/с, готово.
Всего 25 (изменений 5), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (5/5), completed with 3 local objects.
To github.com:kinikulina/study.2022-2023_arh-pc.git
30d2ac6..ba79d53 master -> master

```

Рис. 5.5: Загрузка файлов на Github

6 Выводы

В ходе проделанной работы я освоила процедуры компиляции и сборки программ, написанных на ассемблере NASM.