

# **Отчет по лабораторной работе № 4**

**Архитектура компьютера**

Иванов Александр Олегович

# Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	8
4	Самостоятельная работа	10
5	Выводы	12

## Список иллюстраций

3.1	Использование команды 'mkdir' для создания каталога . . . . .	8
3.2	Использование команды 'cd' для пеерхпда в каталог . . . . .	8
3.3	Использование команды 'touch' для создания файла . . . . .	8
3.4	Использование команды 'nasm -f elf' для создания объектного файла	8
3.5	Использование команды 'ld -m elf_i386 hello.o -o hello' . . . . .	9
3.6	Использование команды './hello' и выводимый результат . . . . .	9
4.1	Использование команды './hello' и выводимый результат . . . . .	10
4.2	самостоятельная работа с командами NASM и вывод результатов	10
4.3	использование команды 'add.' . . . . .	11

## Список таблиц

# 1 Цель работы

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

## 2 Теоретическое введение

Язык ассемблера (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 операционной системы. Именно на этом уровне и работают программы, написанные на ассемблере. Но в отличие от языков высокого уровня ассемблерная программа содержит только тот код, который ввёл программист. Таким образом язык ассемблера — это язык, с помощью которого понятным для человека образом пишутся команды для процессора. Следует отметить, что процессор понимает не команды ассемблера, а последовательности из нулей и единиц — машинные коды. До появления языков ассемблера программистам приходилось писать программы, используя только лишь машинные коды, которые были крайне сложны для запоминания, так как представляли собой числа, записанные в двоичной или шестнадцатеричной системе счисления. Преобразование или трансляция команд с Архитектура ЭВМ языка ассемблера в исполняемый машинный код осуществляется специальной программой транслятором — Ассемблер. Программы, написанные на языке ассемблера, не уступают в качестве и скорости программам, написанным на машинном языке, так как транслятор просто переводит мнемонические обозначения команд в

последовательности бит (нулей и единиц).

### 3 Выполнение лабораторной работы

Создание каталога для работы с программами на языке ассемблера NASM (рис. 3.1).

```
[z@fedora ~]$ mkdir -p ~/work/arch-pc/lab04
```

Рис. 3.1: Использование команды 'mkdir' для создания каталога

Переход в созданный каталог (рис. 3.2)

```
[z@fedora ~]$ cd ~/work/arch-pc/lab04
```

Рис. 3.2: Использование команды 'cd' для перехода в каталог

Создание текстового файла с именем hello.asm (рис. 3.4)

```
[z@fedora lab04]$ touch hello.asm
```

Рис. 3.3: Использование команды 'touch' для создания файла

Превращение текста программы в объектный код (рис. ??)

```
[z@fedora lab04]$ nasm -f elf hello.asm
```

Рис. 3.4: Использование команды 'nasm -f elf' для создания объектного файла

Компилирование исходного файла hello.asm в obj.o (опция -o позволяет задать имя объектного файла, в данном случае obj.o), при этом формат выходного файла



будет elf, и в него будут включены символы для отладки (опция -g), кроме того, будет создан файл листинга list.lst (опция -l) (рис. ??).

Использование команды 'nasm -o obj.o -f elf -g -l list.lst hello.asm' для создания объектного файла

Отправка объектного файла на обработку компоновщику(рис. 3.5)



```
[z@fedora lab04]$ ld -m elf_i386 hello.o -o hello
```

Рис. 3.5: Использование команды 'ld -m elf\_i386 hello.o -o hello'

Запуск на выполнение созданный файл(рис. 3.6)



```
[z@fedora lab04]$ ./hello
Hello world!
```

Рис. 3.6: Использование команды './hello' и выводимый результат

## 4 Самостоятельная работа

Создание копии файла hello.asm с именем lab4.asm(рис. 4.1)

```
[z@fedora lab04]$ cp hello.asm lab4.asm
```

Рис. 4.1: Использование команды './hello' и выводимый результат

Транслирование полученного текста программы lab4.asm в объектный файл. Выполнение компоновки объектного файла и запуск получившегося исполняемого файла (рис. 4.3)

```
[z@fedora lab04]$ ld -m elf_i386 obj.o -o mai
[z@fedora lab04]$ ./lab4
Иванов Александр
[z@fedora lab04]$
```

Рис. 4.2: самостоятельная работа с командами NASM и вывод результатов

Загрузка файлов на Github (рис. 4.3)

```
[z@fedora report]$ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc
[z@fedora arch-pc]$ git add .
[z@fedora arch-pc]$ git commit -am 'feat(main): add files lab-4'
[master aa7295b] feat(main): add files lab-4
17 files changed, 97 insertions(+), 34 deletions(-)
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/lab4.asm
create mode 100644 labs/lab04/report/5.png
create mode 100644 labs/lab04/report/image/1.png
create mode 100644 labs/lab04/report/image/2.png
create mode 100644 labs/lab04/report/image/3.png
create mode 100644 labs/lab04/report/image/4.png
create mode 100644 labs/lab04/report/image/5.png
create mode 100644 labs/lab04/report/image/6.png
create mode 100644 labs/lab04/report/image/7.png
create mode 100644 labs/lab04/report/image/8.png
create mode 100644 labs/lab04/report/image/9.png
create mode 100644 labs/lab04/report/report.docx
create mode 100644 labs/lab04/report/report.pdf
[z@fedora arch-pc]$ git push
Перечисление объектов: 34, готово.
```

Рис. 4.3: использование команды 'add.'

## **5 Выводы**

В ходе лабораторной работы я освоил язык программирования низкого уровня асемблер, а имеено научился компилировать и собирать программы