

Российский университет дружбы народов

**Факультет физико-математических и естественных наук
Кафедра компьютерных и информационных наук**

**ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4**

Дисциплина: Архитектура компьютеров

Студент: *Алексеев Тимофей*

Группа: НК Абд-07-25

Москва
2025 г.

Содержание

1.	Цель работы	5
2.	Задание	6
3.	Теоретическое введение	7
4.	Выполнение лабораторной работы.....	10
	4.1 Программа Hello World!	10
	4.2 Транслятор NASM	11
	4.3 Расширенный синтаксис командной строки NASM..	12
	4.4 компоновщик LD	13
	4.5 Запуск исполняемого файла	15
	4.6 Задания для самостоятельной работы	16
5.	Выводы	20
6.	Список литературы	21

Список иллюстраций

4.1 Создание рабочей директории	10
4.2 Создание .asm файлы	11
4.3 Редактирование файла	11
4.4 Компиляция программы	12
4.5 Возможности синтаксиса NASM	12
4.6 Отправка файла компоновщику	13
4.7 Создание исполняемого файла	13
4.8 Запуск программы	14
4.9 Создание копии	14
4.10 Редактирование копии	15
4.11 Проверка работоспособности скомпонованной программы	15
4.12 Отправка файлов в локальный репозиторий	16
4.13 Загрузка изменений	16

Список таблиц

1. Цель работы

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

2. Задание

1. Создание программы Hello world!
2. Работа с транслятором NASM
3. Работа с расширенным синтаксисом командной строки NASM
4. Работа с компоновщиком LD
5. Запуск исполняемого файла
6. Выполнение заданий для самостоятельной работы.

3. Теоретические введение

Основными функциональными элементами любой ЭВМ являются центральный процессор, память и периферийные устройства. Взаимодействие этих устройств осуществляется через общую шину, к которой они подключены. Физически шина представляет собой большое количество проводников, соединяющих устройства друг с другом. В современных компьютерах проводники выполнены в виде электропроводящих дорожек на материнской плате. Основной задачей процессора является обработка информации, а также организация координации всех узлов компьютера. В состав центрального процессора входят следующие устройства: - арифметико-логическое устройство (АЛУ) — выполняет логические и арифметические действия, необходимые для обработки информации, хранящейся в памяти; - устройство управления (УУ) — обеспечивает управление и контроль всех устройств компьютера; - регистры — сверхбыстрая оперативная память небольшого объёма, входящая в состав процессора, для временного хранения промежуточных результатов выполнения инструкций; регистры процессора делятся на два типа: регистры общего назначения и специальные регистры. Для того, чтобы писать программы на ассемблере, необходимо знать, какие регистры процессора существуют и как их можно использовать. Большинство команд в программах написанных на ассемблере используют регистры в качестве операндов. Практически все команды представляют собой преобразование данных хранящихся в регистрах процессора, это например пересылка данных между регистрами или между регистрами и памятью, преобразование (арифметические или логические

операции) данных хранящихся в регистрах. Доступ к регистрам осуществляется не по адресам, как к основной памяти, а по именам. Каждый регистр процессора архитектуры x86 имеет свое название, состоящее из 2 или 3 букв латинского алфавита. В качестве примера приведем названия основных регистров общего назначения (именно эти регистры чаще всего используются при написании программ): - RAX, RCX, RDX, RBX, RSI, RDI — 64-битные - EAX, ECX, EDX, EBX, ESI, EDI — 32-битные - AX, CX, DX, BX, SI, DI — 16-битные - AH, AL, CH, CL, DH, DL, BH, BL — 8-битные.

Другим важным узлом ЭВМ является оперативное запоминающее устройство (ОЗУ). ОЗУ — это быстродействующее энергозависимое запоминающее устройство, которое напрямую взаимодействует с узлами процессора, предназначенное для хранения программ и данных, с которыми процессор непосредственно работает в текущий момент. ОЗУ состоит из одинаковых пронумерованных ячеек памяти. Номер ячейки памяти — это адрес хранящихся в ней данных. Периферийные устройства в составе ЭВМ: - устройства внешней памяти, которые предназначены для долговременного хранения больших объёмов данных. - устройства ввода-вывода, которые обеспечивают взаимодействие ЦП с внешней средой.

В основе вычислительного процесса ЭВМ лежит принцип программного управления. Это означает, что компьютер решает поставленную задачу как последовательность действий, записанных в виде программы.

Коды команд представляют собой многоразрядные двоичные комбинации из 0 и 1. В коде машинной команды можно выделить две части: операционную и адресную. В операционной части хранится код команды, которую необходимо выполнить. В адресной части хранятся данные или адреса данных, которые участвуют в выполнении данной операции. При выполнении каждой команды процессор выполняет определённую последовательность стандартных действий,

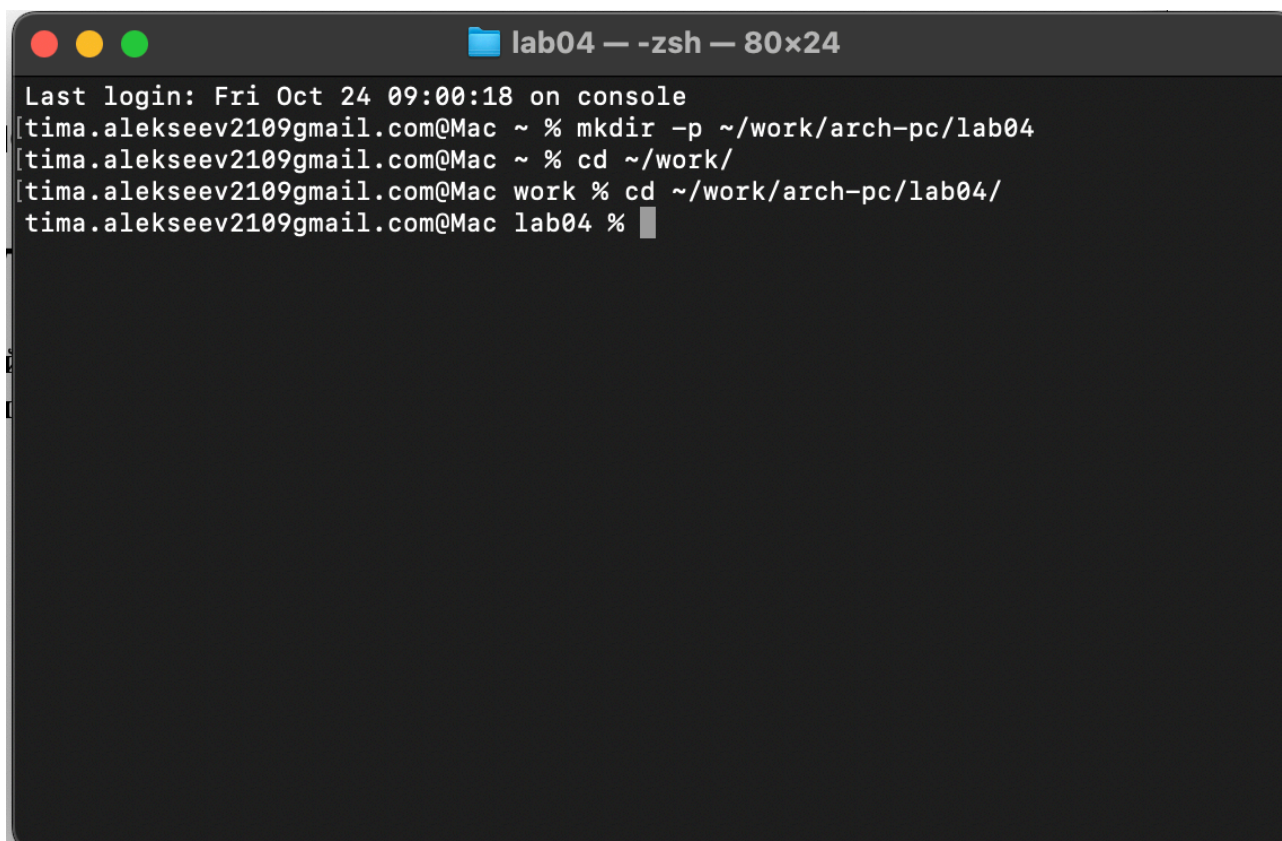
которая называется командным циклом процессора. Он заключается в следующем: 1. формирование адреса в памяти очередной команды; 2. считывание кода команды из памяти и её дешифрация; 3. выполнение команды; 4. переход к следующей команде.

Язык ассемблера (assembly language, сокращённо asm) — машинноориентированный язык низкого уровня. 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.

4. Выполнение лабораторной работы

4.1 Программа *Hello World!*

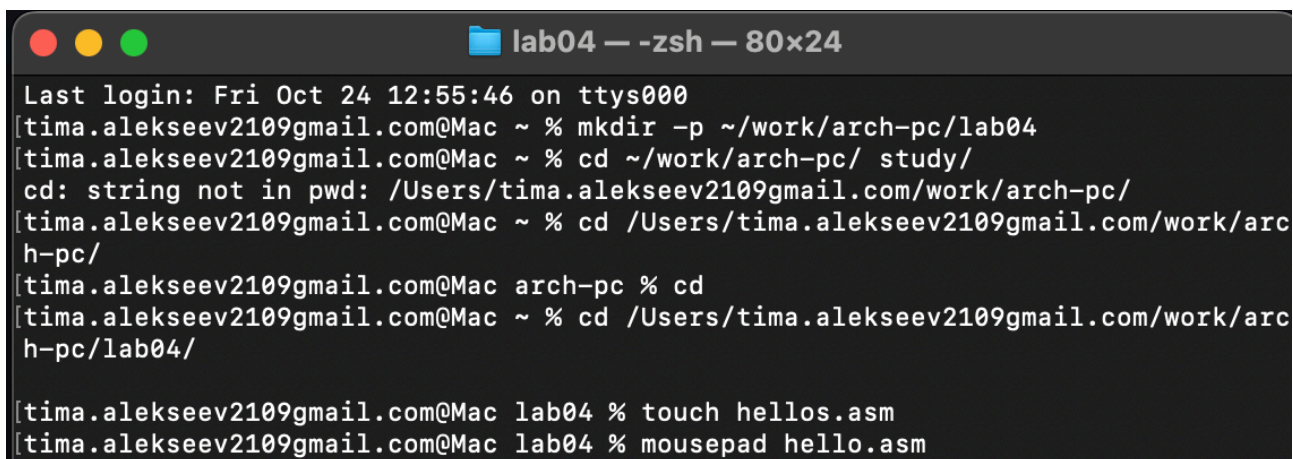
В домашней директории создаю каталог, в котором буду хранить файлы для текущей лабораторной работы. (рис. 4.1)

A screenshot of a terminal window titled "lab04 — -zsh — 80x24". The window shows a series of commands and their outputs. The first line is "Last login: Fri Oct 24 09:00:18 on console". The second line is "[tima.alekseev2109gmail.com@Mac ~ % mkdir -p ~/work/arch-pc/lab04". The third line is "[tima.alekseev2109gmail.com@Mac ~ % cd ~/work/". The fourth line is "[tima.alekseev2109gmail.com@Mac work % cd ~/work/arch-pc/lab04/". The fifth line is "[tima.alekseev2109gmail.com@Mac lab04 %]" with a cursor at the end. The terminal has a dark background and white text. The window title bar is dark gray with three colored buttons (red, yellow, green) on the left and the title text on the right.

```
Last login: Fri Oct 24 09:00:18 on console
[tima.alekseev2109gmail.com@Mac ~ % mkdir -p ~/work/arch-pc/lab04
[tima.alekseev2109gmail.com@Mac ~ % cd ~/work/
[tima.alekseev2109gmail.com@Mac work % cd ~/work/arch-pc/lab04/
[tima.alekseev2109gmail.com@Mac lab04 % ]
```

Рис. 4.1: Создание рабочей директории

Создаю в нем файл `hello.asm`, в котором буду писать программу на языке ассемблера. (рис. 4.2)

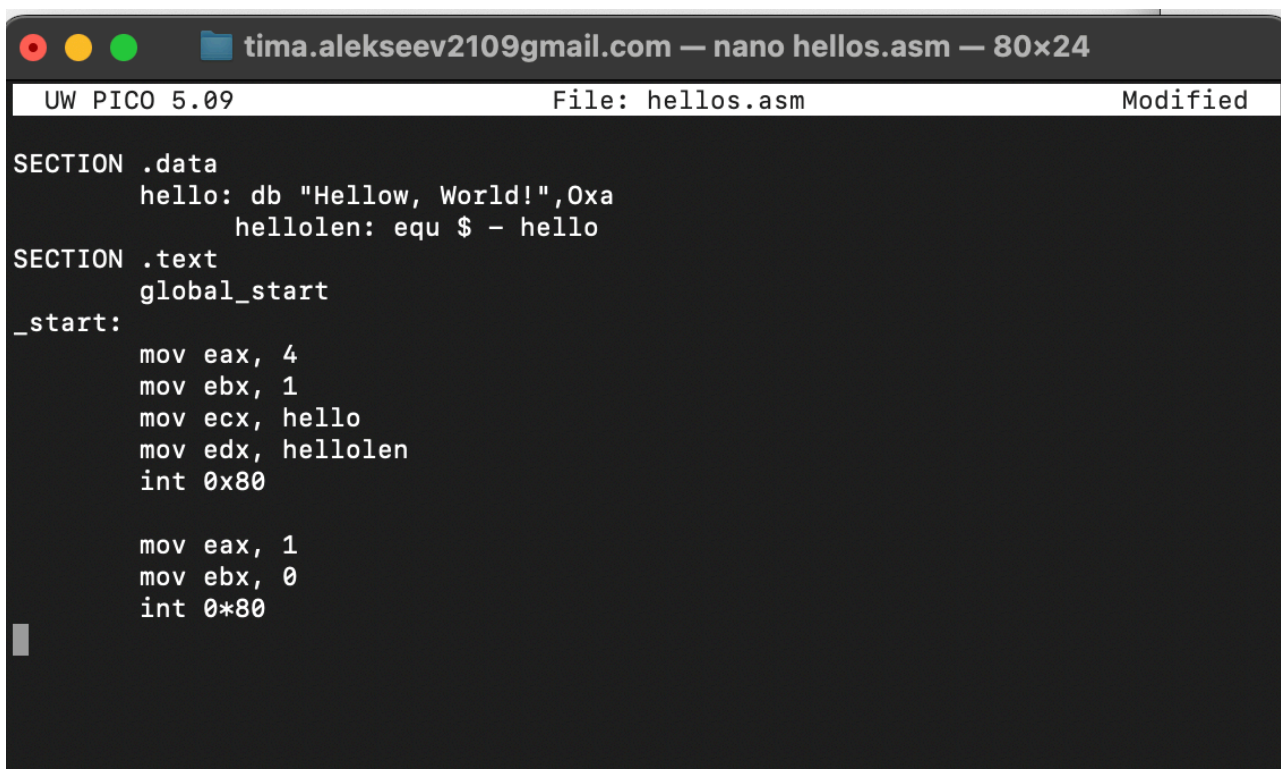


```
lab04 — -zsh — 80x24
Last login: Fri Oct 24 12:55:46 on ttys000
[tima.alekseev2109gmail.com@Mac ~ % mkdir -p ~/work/arch-pc/lab04
[tima.alekseev2109gmail.com@Mac ~ % cd ~/work/arch-pc/ study/
cd: string not in pwd: /Users/tima.alekseev2109gmail.com/work/arch-pc/
[tima.alekseev2109gmail.com@Mac ~ % cd /Users/tima.alekseev2109gmail.com/work/arch-pc/
[tima.alekseev2109gmail.com@Mac arch-pc % cd
[tima.alekseev2109gmail.com@Mac ~ % cd /Users/tima.alekseev2109gmail.com/work/arch-pc/lab04/
[tima.alekseev2109gmail.com@Mac lab04 % touch hellos.asm
[tima.alekseev2109gmail.com@Mac lab04 % mousepad hello.asm
```

(Mousepad не работает почему-то на macOS, открыл через nano)

Рис. 4.2: Создание .asm файла

С помощью редактора пишу программу в созданном файле. (рис. 4.3)



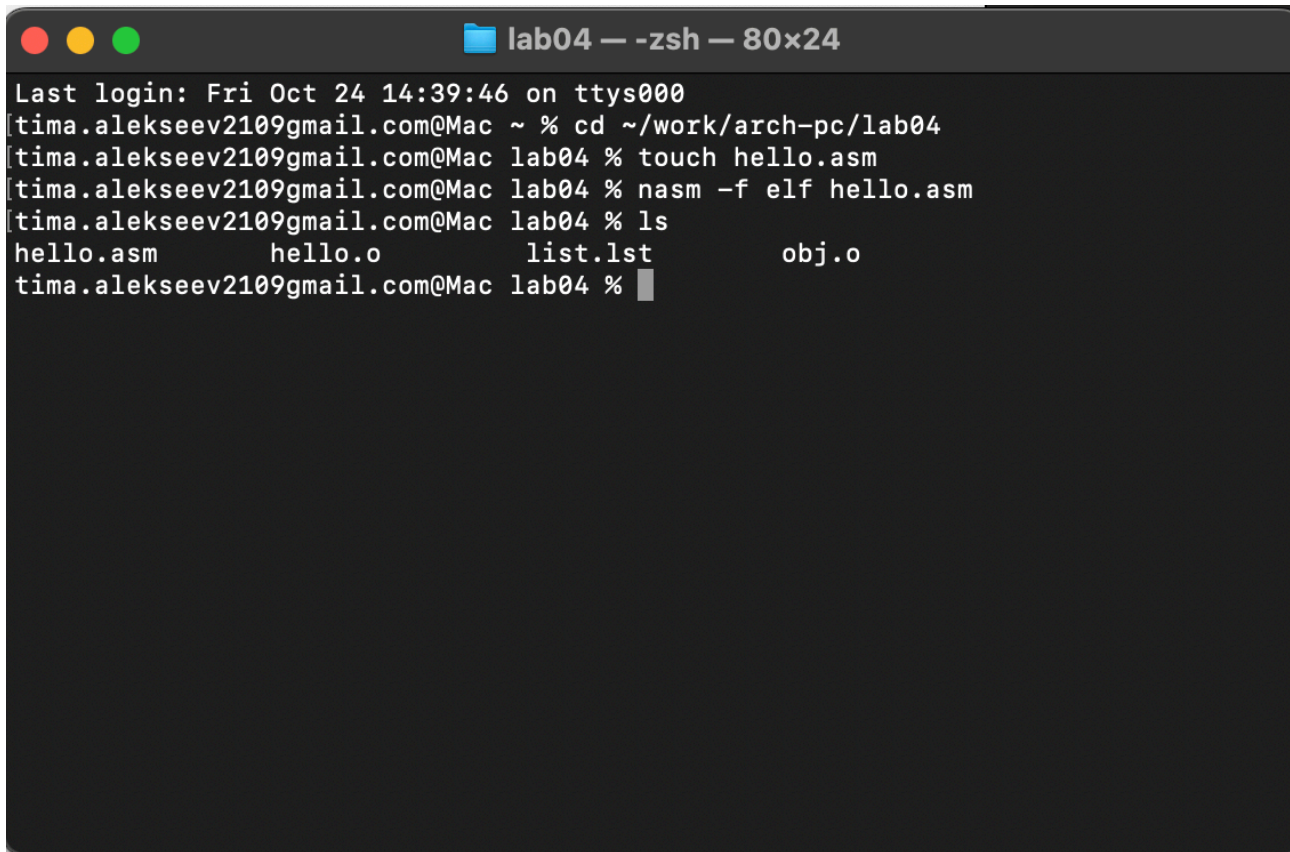
```
tima.alekseev2109gmail.com — nano hellos.asm — 80x24
UW PICO 5.09 File: hellos.asm Modified
SECTION .data
    hello: db "Hellow, World!",0xa
           hellolen: equ $ - hello
SECTION .text
    global _start
_start:
    mov eax, 4
    mov ebx, 1
    mov ecx, hello
    mov edx, hellolen
    int 0x80

    mov eax, 1
    mov ebx, 0
    int 0x80
```

Рис. 4.3: Редактирование файла

4.2 Транслятор NASM

Компилирую с помощью NASM свою программу. (рис. 4.4)

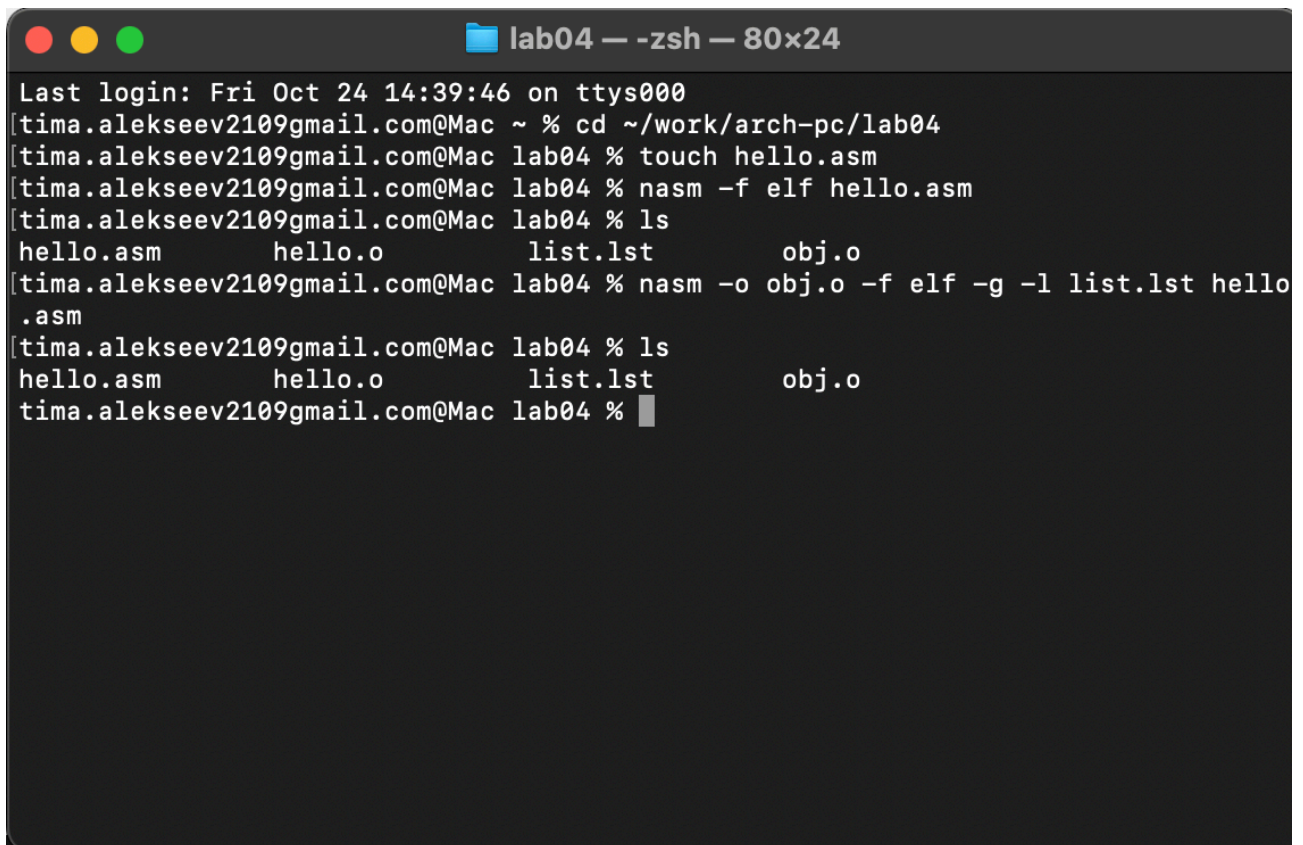
A terminal window titled 'lab04 — -zsh — 80x24' with three colored window control buttons (red, yellow, green) in the top-left corner. The terminal displays the following text:

```
Last login: Fri Oct 24 14:39:46 on ttys000
[tima.alekseev2109gmail.com@Mac ~ % cd ~/work/arch-pc/lab04
[tima.alekseev2109gmail.com@Mac lab04 % touch hello.asm
[tima.alekseev2109gmail.com@Mac lab04 % nasm -f elf hello.asm
[tima.alekseev2109gmail.com@Mac lab04 % ls
hello.asm      hello.o      list.lst      obj.o
[tima.alekseev2109gmail.com@Mac lab04 % █
```

Рис. 4.4: Компиляция программы

4.3 Расширенный синтаксис командной строки NASM

Выполняя команду, указанную на (рис. 4.5), она скомпилировала исходный файл `hello.asm` в `obj.o`, расширение `.o` говорит о том, что файл - объектный, помимо него флаги `-g -l` подготовят файл отладки и листинга соответственно

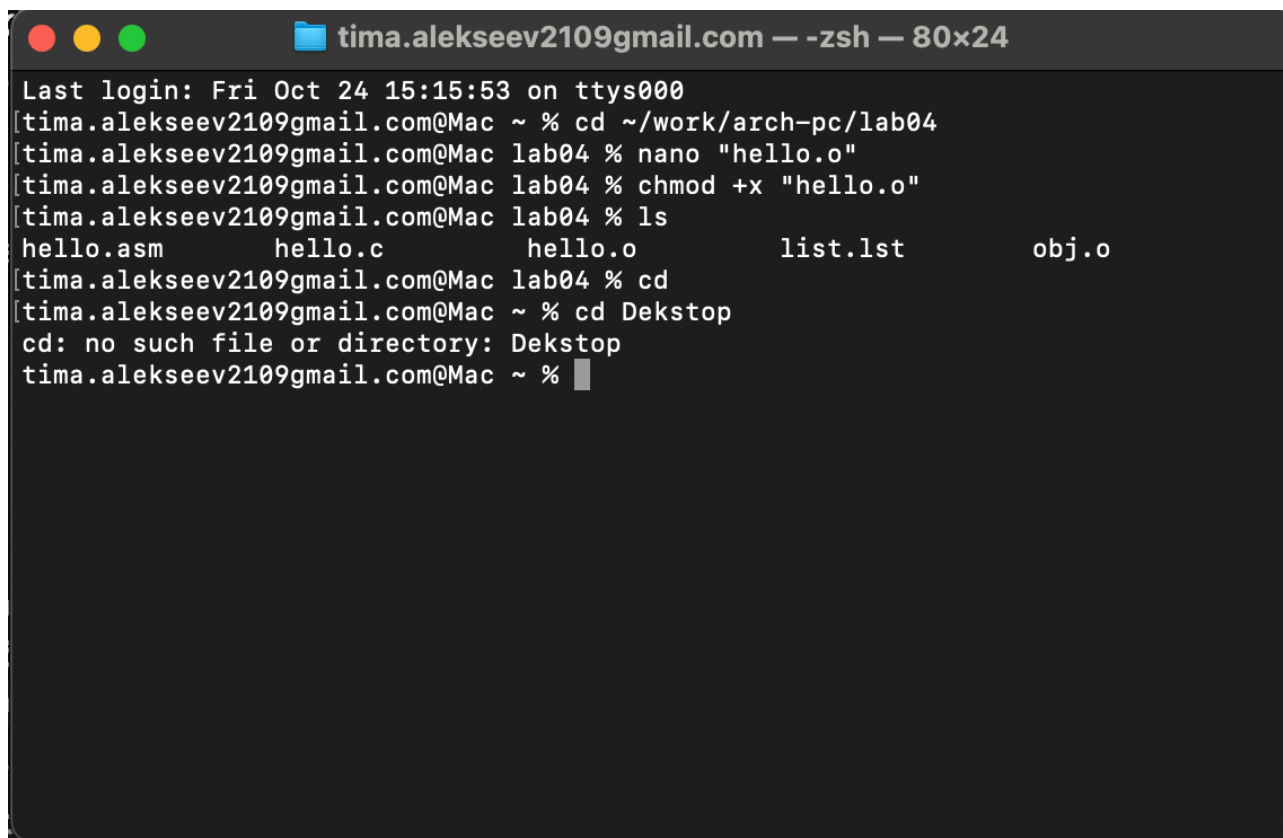
A terminal window titled 'lab04 — -zsh — 80x24' with standard macOS window controls (red, yellow, green buttons). The terminal shows a series of commands and their outputs. The user navigates to a directory, creates a file, assembles it with NASM, and lists the files. The output shows the creation of 'hello.o' and 'list.lst' files, and the assembly of 'hello.asm' into 'obj.o'.

```
Last login: Fri Oct 24 14:39:46 on ttys000
[tima.alekseev2109gmail.com@Mac ~ % cd ~/work/arch-pc/lab04
[tima.alekseev2109gmail.com@Mac lab04 % touch hello.asm
[tima.alekseev2109gmail.com@Mac lab04 % nasm -f elf hello.asm
[tima.alekseev2109gmail.com@Mac lab04 % ls
hello.asm      hello.o      list.lst     obj.o
[tima.alekseev2109gmail.com@Mac lab04 % nasm -o obj.o -f elf -g -l list.lst hello
.asm
[tima.alekseev2109gmail.com@Mac lab04 % ls
hello.asm      hello.o      list.lst     obj.o
[tima.alekseev2109gmail.com@Mac lab04 % █
```

Рис. 4.5: Возможности синтаксиса NASM

4.4 Компоновщик LD

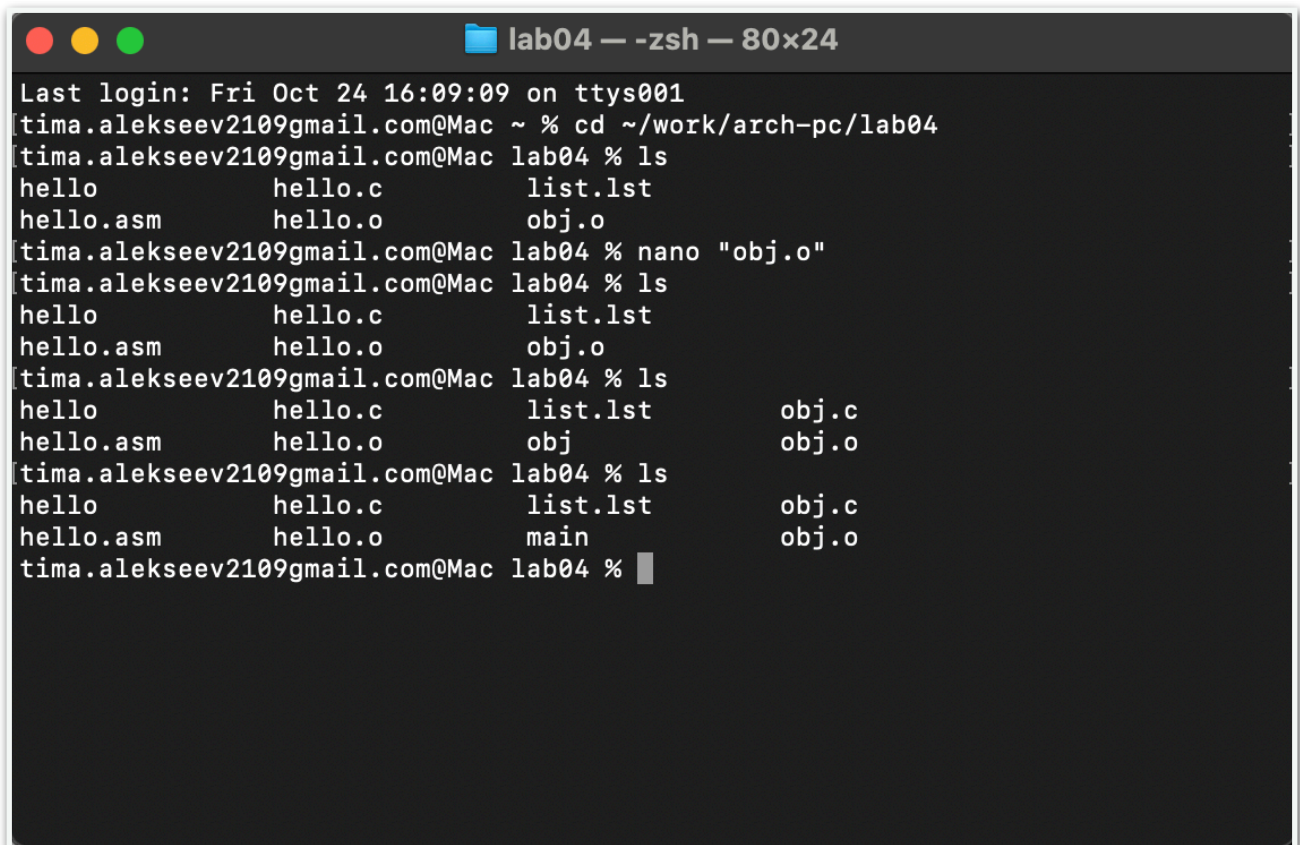
Затем мне необходимо передать объектный файл компоновщику, делаю это с помощью команды `ld`. (рис. 4.6)

A terminal window with a dark background and light gray text. The title bar at the top shows three colored circles (red, yellow, green) on the left, followed by a folder icon and the text "tima.alekseev2109gmail.com — -zsh — 80x24". The terminal content shows a series of commands and their outputs. The first line is "Last login: Fri Oct 24 15:15:53 on ttys000". Subsequent lines show the user navigating to a directory, creating a file with nano, setting permissions with chmod, listing files with ls, and attempting to change directories with cd. The ls command output lists several files: hello.asm, hello.c, hello.o, list.lst, and obj.o. The cd command fails with an error message because the directory does not exist.

```
tima.alekseev2109gmail.com — -zsh — 80x24
Last login: Fri Oct 24 15:15:53 on ttys000
tima.alekseev2109gmail.com@Mac ~ % cd ~/work/arch-pc/lab04
tima.alekseev2109gmail.com@Mac lab04 % nano "hello.o"
tima.alekseev2109gmail.com@Mac lab04 % chmod +x "hello.o"
tima.alekseev2109gmail.com@Mac lab04 % ls
hello.asm      hello.c      hello.o      list.lst     obj.o
tima.alekseev2109gmail.com@Mac lab04 % cd
tima.alekseev2109gmail.com@Mac ~ % cd Dekstop
cd: no such file or directory: Dekstop
tima.alekseev2109gmail.com@Mac ~ %
```

Рис. 4.6: Отправка файла компоновщику(я сделал через chmod, потому что не понимаю почему у меня не работает ld)

Выполняю следующую команду ..., результатом исполнения команды будет созданный файл main, скомпонованный из объектного файла obj.o. (рис. 4.7)

A terminal window titled "lab04 — -zsh — 80x24" with standard macOS window controls (red, yellow, green buttons). The terminal shows a sequence of commands and their outputs. The user logs in, changes to the directory ~/work/arch-pc/lab04, and lists files. They then use nano to create a file named "obj.o". After listing files again, they compile "obj.o" into "obj.c" and "obj.o". Finally, they compile "obj.c" into "main" and "obj.o".

```
Last login: Fri Oct 24 16:09:09 on ttys001
tima.alekseev2109gmail.com@Mac ~ % cd ~/work/arch-pc/lab04
tima.alekseev2109gmail.com@Mac lab04 % ls
hello          hello.c        list.lst
hello.asm      hello.o        obj.o
tima.alekseev2109gmail.com@Mac lab04 % nano "obj.o"
tima.alekseev2109gmail.com@Mac lab04 % ls
hello          hello.c        list.lst
hello.asm      hello.o        obj.o
tima.alekseev2109gmail.com@Mac lab04 % ls
hello          hello.c        list.lst      obj.c
hello.asm      hello.o        obj           obj.o
tima.alekseev2109gmail.com@Mac lab04 % ls
hello          hello.c        list.lst      obj.c
hello.asm      hello.o        main          obj.o
tima.alekseev2109gmail.com@Mac lab04 %
```

Рис. 4.7: Создание и
исполняемого файла

4.5 Запуск исполняемого файла

Запускаю исполняемый файл из текущего каталога. (рис. 4.8)

```
lab04 — -zsh — 80x24
Last login: Fri Oct 24 16:09:09 on ttys001
tima.alekseev2109gmail.com@Mac ~ % cd ~/work/arch-pc/lab04
tima.alekseev2109gmail.com@Mac lab04 % ls
hello          hello.c      list.lst
hello.asm      hello.o      obj.o
tima.alekseev2109gmail.com@Mac lab04 % nano "obj.o"
tima.alekseev2109gmail.com@Mac lab04 % ls
hello          hello.c      list.lst
hello.asm      hello.o      obj.o
tima.alekseev2109gmail.com@Mac lab04 % ls
hello          hello.c      list.lst      obj.c
hello.asm      hello.o      obj            obj.o
tima.alekseev2109gmail.com@Mac lab04 % ls
hello          hello.c      list.lst      obj.c
hello.asm      hello.o      main          obj.o
tima.alekseev2109gmail.com@Mac lab04 % ./hello
Hello, World!
tima.alekseev2109gmail.com@Mac lab04 %
```

Рис. 4.8: Запуск программы

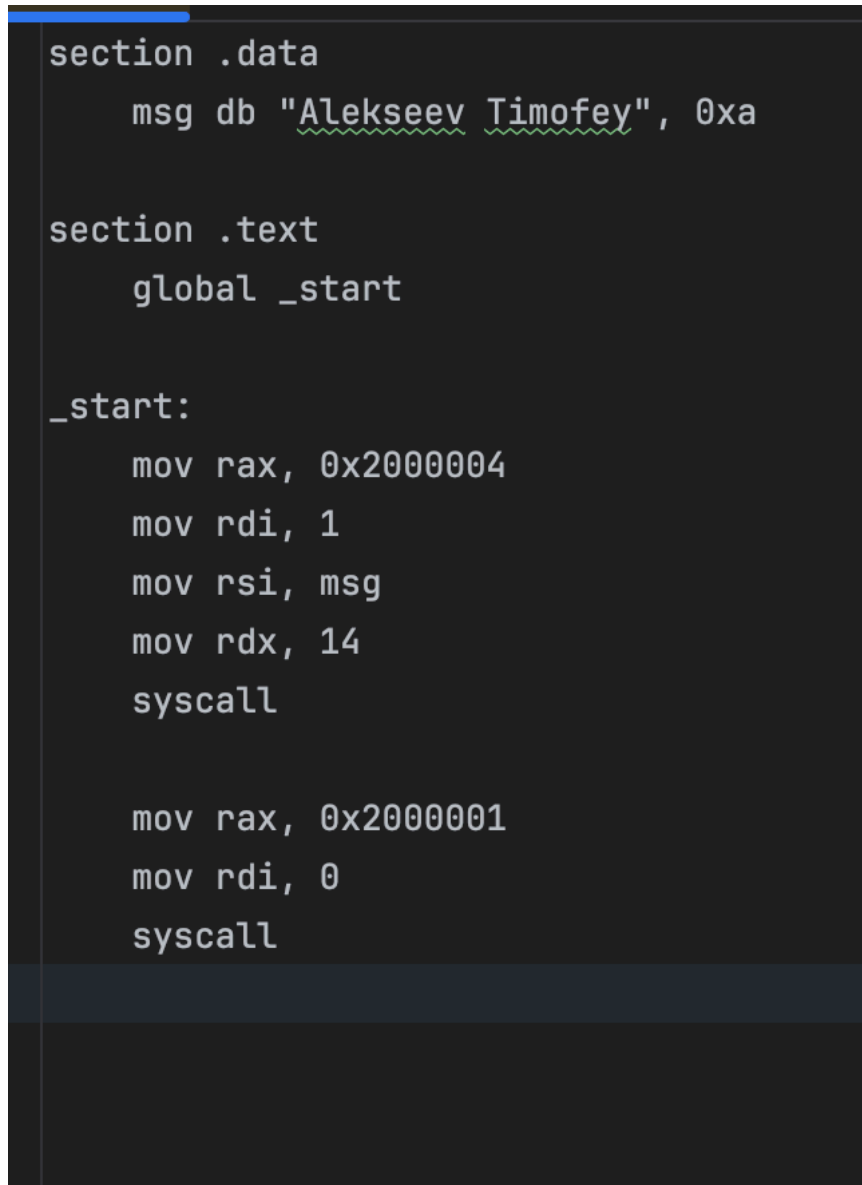
4.6 Задания для самостоятельной работы

Создаю копию файла для последующей работы с ней. (рис. 4.9)

```
lab04 — -zsh — 80x24
Last login: Fri Oct 24 16:09:09 on ttys001
[tima.alekseev2109gmail.com@Mac ~ % cd ~/work/arch-pc/lab04
[tima.alekseev2109gmail.com@Mac lab04 % ls
hello          hello.c      list.lst
hello.asm      hello.o      obj.o
[tima.alekseev2109gmail.com@Mac lab04 % nano "obj.o"
[tima.alekseev2109gmail.com@Mac lab04 % ls
hello          hello.c      list.lst
hello.asm      hello.o      obj.o
[tima.alekseev2109gmail.com@Mac lab04 % ls
hello          hello.c      list.lst      obj.c
hello.asm      hello.o      obj            obj.o
[tima.alekseev2109gmail.com@Mac lab04 % ls
hello          hello.c      list.lst      obj.c
hello.asm      hello.o      main          obj.o
[tima.alekseev2109gmail.com@Mac lab04 % ./hello
Hello, World!
[tima.alekseev2109gmail.com@Mac lab04 % cp hello.asm lab4.asm
[tima.alekseev2109gmail.com@Mac lab04 % ls
hello          hello.c      lab4.asm      main          obj.o
hello.asm      hello.o      list.lst      obj.c
tima.alekseev2109gmail.com@Mac lab04 %
```


Рис. 4.9: Создание копии

Редактирую копию файла, заменив текст на свое имя и фамилию. (рис. 4.10)



```
section .data
    msg db "Alekseev Timofey", 0xa

section .text
    global _start

_start:
    mov rax, 0x20000004
    mov rdi, 1
    mov rsi, msg
    mov rdx, 14
    syscall

    mov rax, 0x20000001
    mov rdi, 0
    syscall
```

Рис. 4.10: Редактирование копии

Транслирую копию файла в объектный файл, компоную и запускаю. (рис. 4.11)

```
lab04 — -zsh — 80x24
Last login: Fri Oct 24 17:07:46 on ttys001
[tima.alekseev2109gmail.com@Mac ~ % cd ~/work/arch-pc/lab04/
[tima.alekseev2109gmail.com@Mac lab04 % ls
hello          hello.c          lab4.asm          list.lst          obj.c
hello.asm       hello.o          lab4.asm.c        main              obj.o
[tima.alekseev2109gmail.com@Mac lab04 % ./hello.o
./hello.o: line 3: syntax error near unexpected token `('
./hello.o: line 3: `int main() {'
[tima.alekseev2109gmail.com@Mac lab04 % ./hello
Hello, World!
[tima.alekseev2109gmail.com@Mac lab04 % ./lab4
zsh: no such file or directory: ./lab4
[tima.alekseev2109gmail.com@Mac lab04 % ./lab4
Alekseev Timofey
tima.alekseev2109gmail.com@Mac lab04 %
```

4.11: Проверка работоспособности скомпонованной программы

Убедившись в корректности работы программы, копирую рабочие файлы в свой локальный репозиторий. (рис. 4.12)

```
lab04 — -zsh — 80x24
Last login: Fri Oct 24 17:14:36 on ttys000
[tima.alekseev2109gmail.com@Mac ~ % cd ~/work/study/2025-2026/arch-pc/lab04
cd: no such file or directory: /Users/tima.alekseev2109gmail.com/work/study/2025-2026/arch-pc/lab04
[tima.alekseev2109gmail.com@Mac ~ % cd ~/work/arch-pc/lab04
[tima.alekseev2109gmail.com@Mac lab04 % cp hello.asm lab4.asm /study/2025-2026/arch-pc/labs/lab04/
cp: /study/2025-2026/arch-pc/labs/lab04 is not a directory
[tima.alekseev2109gmail.com@Mac lab04 % cp hello.asm lab4.asm /study/2025-2026/arch-pc/labs/lab04
cp: /study/2025-2026/arch-pc/labs/lab04 is not a directory
[tima.alekseev2109gmail.com@Mac lab04 % cp hello.asm lab4.asm /work/study/2025-2026/arch-pc/labs/lab04/
cp: /work/study/2025-2026/arch-pc/labs/lab04 is not a directory
[tima.alekseev2109gmail.com@Mac lab04 % /work/study/2025-2026/arch-pc/labs/
zsh: no such file or directory: /work/study/2025-2026/arch-pc/labs/
[tima.alekseev2109gmail.com@Mac lab04 % ls
hello          lab4.asm          obj.c
hello.asm       lab4.asm.c        obj.o
hello.c         lab4.c            presentation report.lst
hello.o         list.lst
lab4            main
tima.alekseev2109gmail.com@Mac lab04 %
```

Рис. 4.12: Отправка файлов в локальный репозиторий

Загрузка изменений на свой удаленный репозиторий на GitHub. (рис. 4.13)

```
lab04 — -zsh — 139x58
tima.alekseev2109gmail.com@Mac lab04 % git add .
tima.alekseev2109gmail.com@Mac lab04 % git status
warning: could not open directory '.Trash/': Operation not permitted
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   .idea/workspace.xml

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:    ../..study/2025-2026/Архитектура компьютеров/.DS_Store
    deleted:    ../..study/2025-2026/Архитектура компьютеров/Course

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    ../../../../CFUserTextEncoding
    ../../../../.DS_Store
    ../../../../.gitconfig
    ../../../../.idlerc/
    ../../../../.nuget/
    ../../../../.ssh/
    ../../../../.yandex/
    ../../../../.zprofile
    ../../../../.zsh_history
    ../../../../.zsh_sessions/
    ../../../../.zshrc
    ../../../../Applications/
    ../../../../CLionProjects/
    ../../../../Desktop/
    ../../../../Documents alias
    ../../../../Documents/
    ../../../../Downloads/
    ../../../../Library/
    ../../../../Movies/
    ../../../../Music/
    ../../../../Pictures/
    ../../../../Public/
    ../../../../PyCharmMiscProject/
    ../../../../hellos.asm.save
    ../../../../parentdir2/
    ../../../../parentdir3/
    ../../../../.DS_Store
    ../../study/.DS_Store
    ../../study/2025-2026/.DS_Store
    ../../study/2025-2026/arch-pc/

tima.alekseev2109gmail.com@Mac lab04 % git commit -m "feat(main):upload 4 lab work"
[master b26ce0a] feat(main):upload 4 lab work
1 file changed, 98 insertions(+)
create mode 100644 work/arch-pc/lab04/.idea/workspace.xml
tima.alekseev2109gmail.com@Mac lab04 % git push
fatal: No configured push destination.
Either specify the URL from the command-line or configure a remote repository using

    git remote add <name> <url>

and then push using the remote name
```

```
git push <name>

tima.alekseev2109gmail.com@Mac lab04 %
```

Рис. 4.13: Загрузка изменений

5. Выводы

При выполнении данной лабораторной работы я освоила процедуры компиляции и сборки программ, написанных на ассемблере NASM.

6. Список литературы

1. Пример выполнения лабораторной работы
2. Курс на ТУИС
3. Лабораторная работа №4
4. Программирование на языке ассемблера NASM Столяров А. В.