

Отчет по выполнению лабораторной работы №4

Дисциплина: Архитектура компьютера

Ефремова Полина Александровна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Задание для самостоятельной работы	14
6	Выводы	17
	Список литературы	18

Список иллюстраций

4.1	Новый каталог	9
4.2	Новый файл	10
4.3	Открытие файла	10
4.4	Открытие файла	11
4.5	Компиляция текста программы	11
4.6	Компиляция в obj.o	12
4.7	Передача файла	12
4.8	Выполнение команды	13
4.9	Выполнение команды	13
5.1	Переход	14
5.2	Копирование	15
5.3	Вывод собственной команды	15
5.4	Копирование файлов в локальный репозиторий	16
5.5	Загрузка на GitHub	16

Список таблиц

1 Цель работы

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

2 Задание

1. Знакомство с теоретической информацией (работа с языком “Ассемблер”)
2. Написание команды ‘Hello world!’
3. Научиться работать с шаблоном, чтобы создавать собственные команды.

3 Теоретическое введение

Ассемблер (англ. «Assembler») — это низкоуровневый язык программирования, который представляет собой промежуточное звено между машинным кодом и высокоуровневыми языками программирования. Он используется для написания программ, которые управляют компьютером или другими устройствами на более низком уровне, непосредственно взаимодействуя с аппаратным обеспечением. Код, написанный на этом языке, обычно сохраняется с помощью расширения ASM.

Директивы

В языке ассемблера директивы — это специальные инструкции. Они используются для предоставления дополнительной информации ассемблеру или компоновщику, а не выполняются как часть программы. Директивы обычно обозначают специальным символом, например точкой или решеткой.

SECTION: эта директива нужна для определения разделов программы, которые используют для группировки связанного кода и данных вместе.

ORG: чтобы установить исходный или начальный адрес программы или раздела.

EQU: чтобы определить константы или символы, которые используют во всей программе.

DB, DW, DD: для определения значений данных байтов, слов или двойных слов в памяти.

ALIGN: для выравнивания ячейки памяти следующей инструкции или значения данных с указанной границей.

EXTERN, GLOBAL: чтобы указать, определяется ли символ внешне или глобально. Эту информацию использует компоновщик для разрешения ссылок на символы в разных объектных файлах.

INCLUDE: для включения файла кода на языке ассемблера в текущую программу.

Директивы помогают управлять структурой и организацией программы на языке ассемблера, указывать дополнительную информацию для создания конечной исполняемой программы.

Команды

Команды языка ассемблера — основные строительные блоки программ. Эти инструкции используют, чтобы сообщить процессору, какие операции следует выполнять. В одних архитектурах сотни или тысячи различных инструкций, в других может быть всего несколько десятков.

Основные:

Команды перемещения данных: Перемещают данные между регистрами или ячейками памяти: MOV, PUSH и POP.

Арифметические команды: Выполняют арифметические операции с данными в регистрах или ячейках памяти: ADD, SUB и MUL.

Логические команды: Выполняют логические операции с данными в регистрах или ячейках памяти: AND, OR и XOR.

Команды ветвления: Управляют путем перехода к другому разделу кода: JMP, JZ и JE.

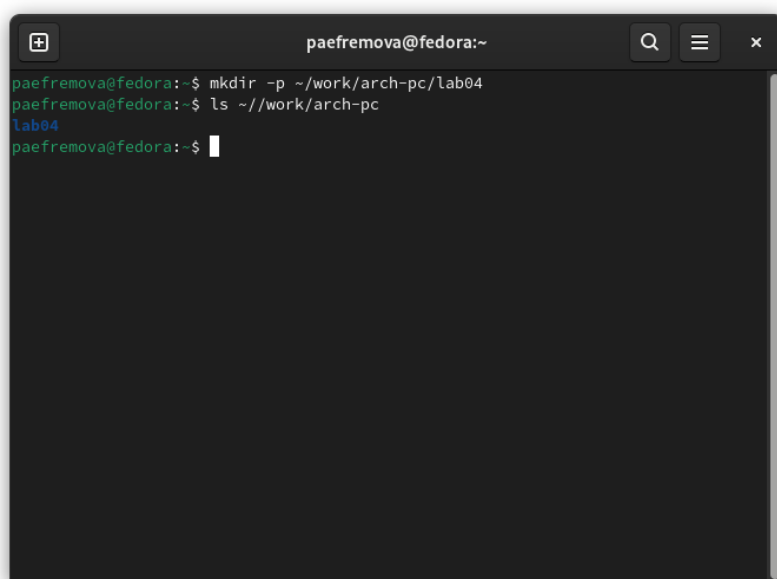
Команды стека: Управляют стеком — областью памяти для хранения данных — и управляющей информацией во время вызовов функций и возвратов: PUSH и POP.

Системные вызовы: Позволяют программам на ассемблере взаимодействовать с операционной системой или другими системными функциями, такими как INT, которые запускают программное прерывание.

4 Выполнение лабораторной работы

Устанавливаю необходимое ПО

Создаю каталог для работы с программами на языке ассемблера(рис. 4.1)

A terminal window titled 'paefremova@fedora:~' with search, menu, and close icons. It shows the execution of two commands: 'mkdir -p ~/work/arch-pc/lab04' and 'ls ~/work/arch-pc', with the output 'lab04' displayed on the second line.

```
paefremova@fedora:~$ mkdir -p ~/work/arch-pc/lab04
paefremova@fedora:~$ ls ~/work/arch-pc
lab04
paefremova@fedora:~$
```

Рис. 4.1: Новый каталог

Перехожу в созданный каталог и создаю текстовый файл с именем hello.asm (рис. 4.2)

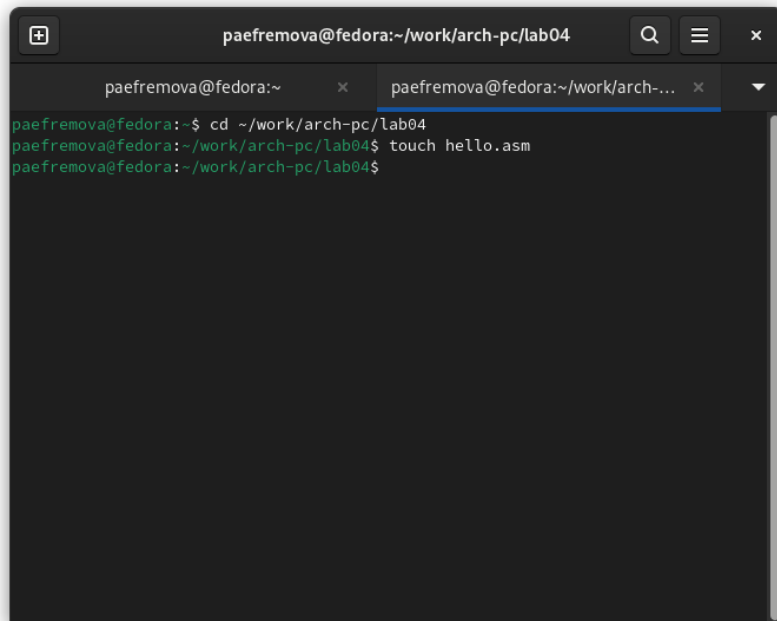


Рис. 4.2: Новый файл

Открываю созданный файл (рис. 4.3)

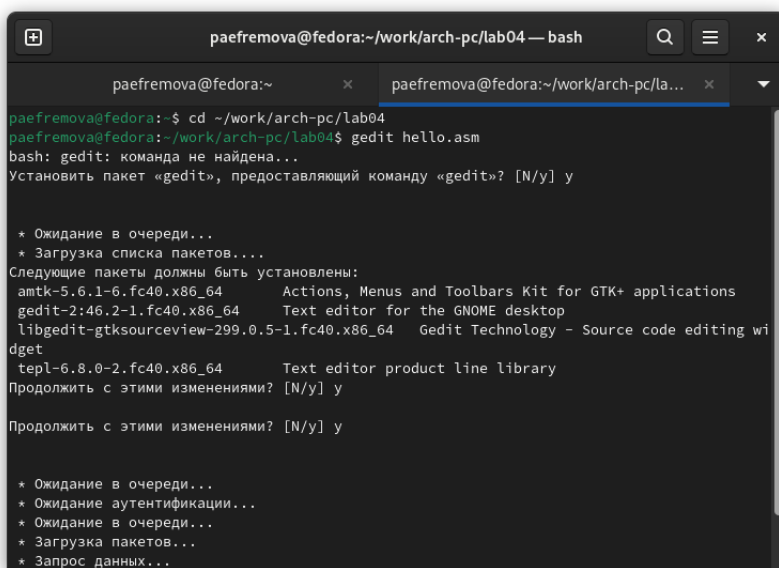


Рис. 4.3: Открытие файла

(рис. 4.4)

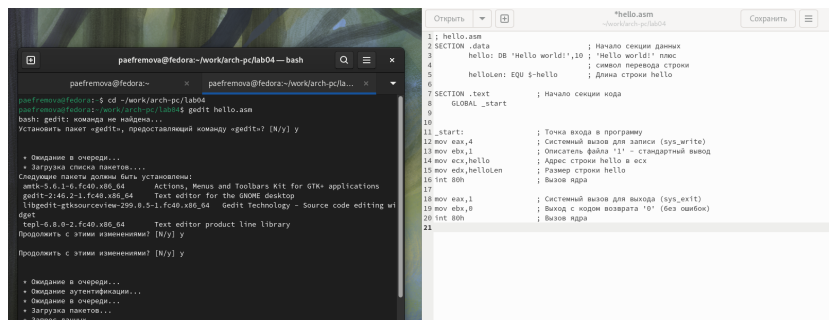


Рис. 4.4: Открытие файла

Для компиляции приведённого выше текста программы «Hello World» используя команду (рис. 4.5)

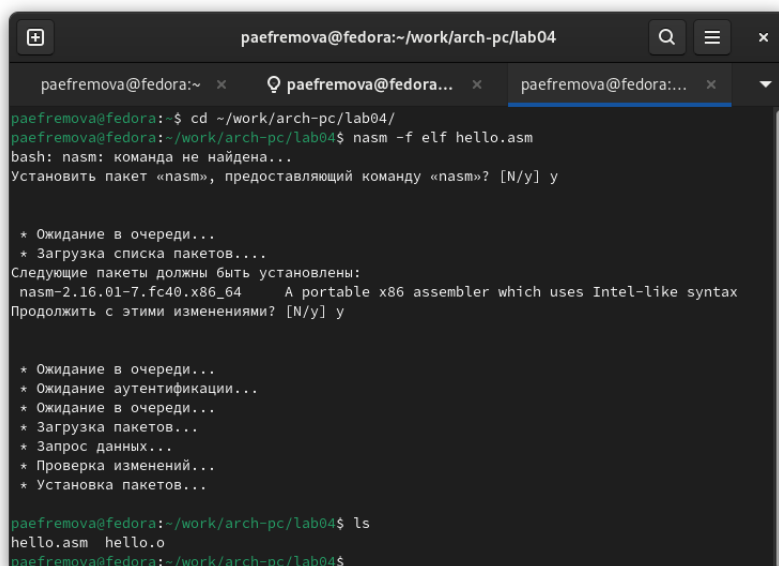
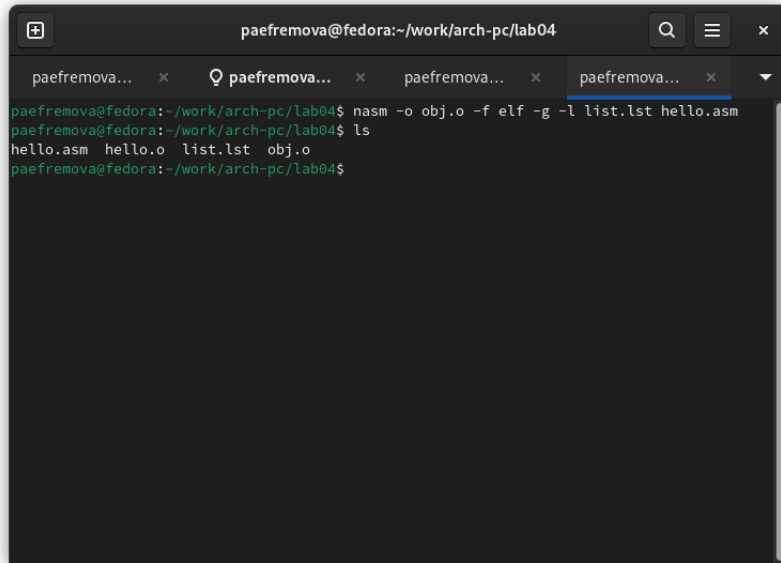


Рис. 4.5: Компиляция текста программы

Компилирую исходный файл hello.asm в obj.o (рис. 4.6)

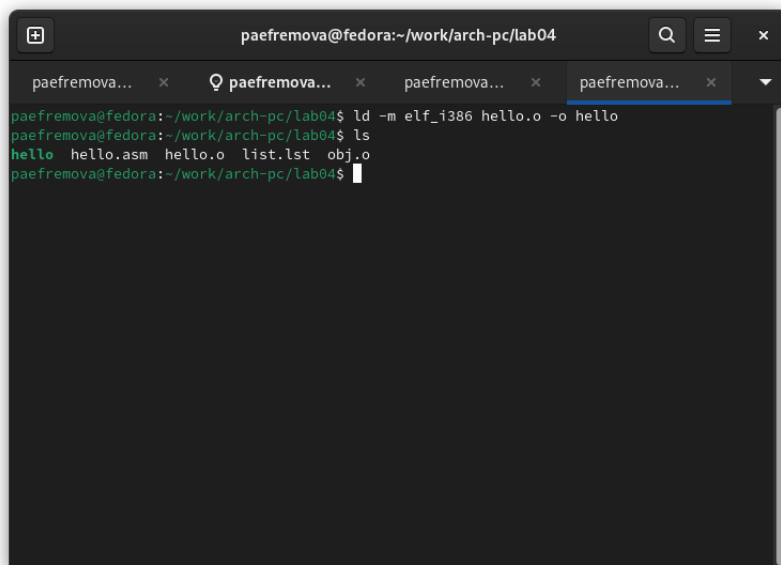


A terminal window titled 'paefremova@fedora:~/work/arch-pc/lab04'. It shows the following commands and output:

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

Рис. 4.6: Компиляция в obj.o

Передача объектного файла на работу компановщику (рис. 4.7)

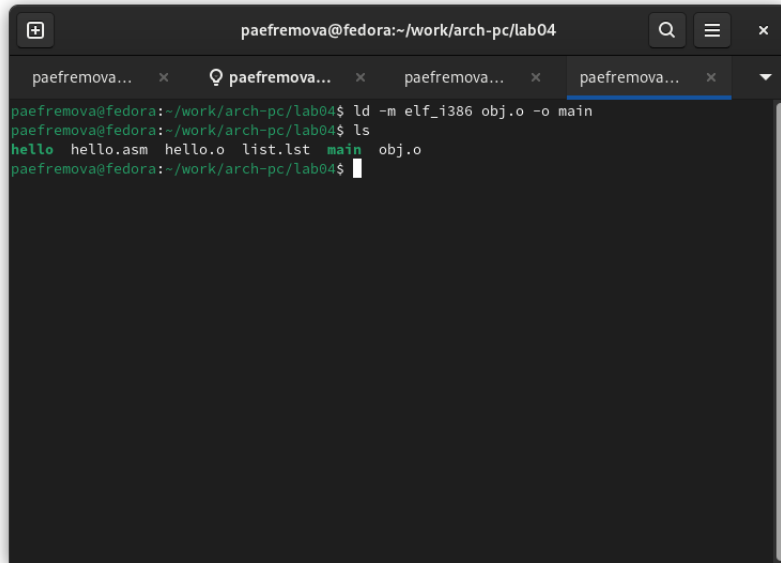


A terminal window titled 'paefremova@fedora:~/work/arch-pc/lab04'. It shows the following commands and output:

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

Рис. 4.7: Передача файла

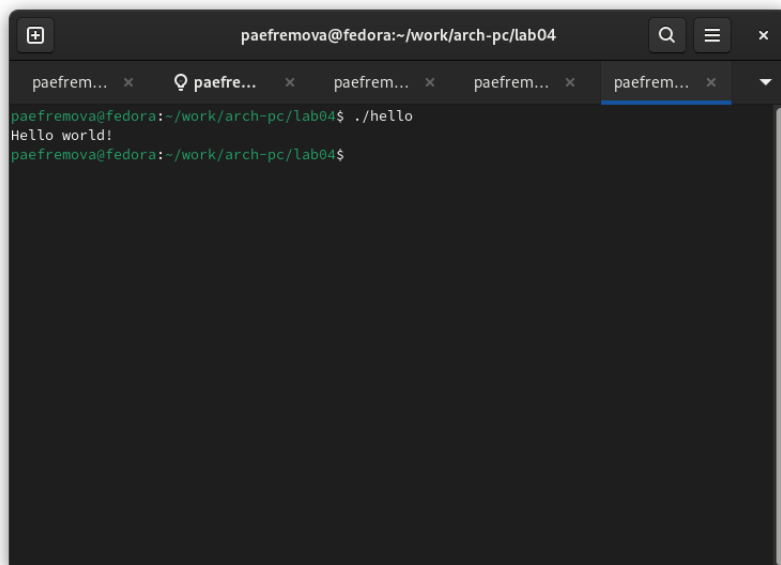
Выполняю еще одну команду (рис. 4.8)



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

Рис. 4.8: Выполнение команды

Запускаю на выполнение созданный файл (рис. 4.9)

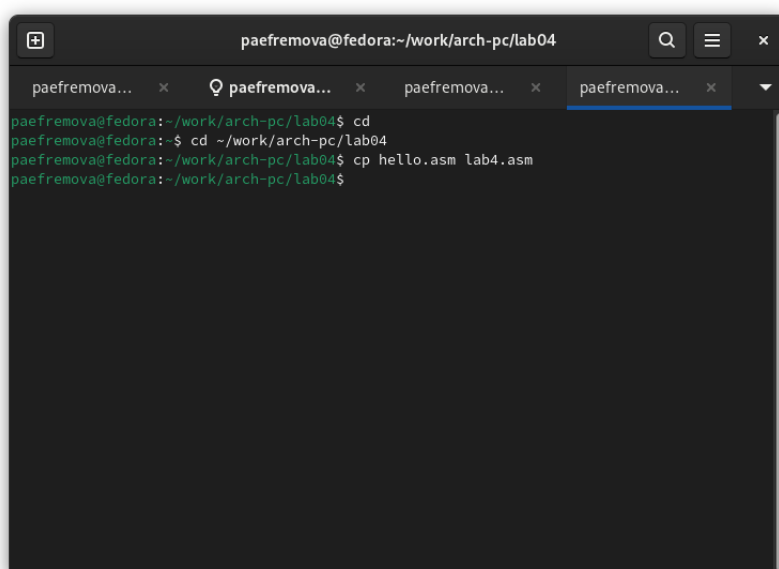


```
paefremova@fedora:~/work/arch-pc/lab04$ ./hello
Hello world!
paefremova@fedora:~/work/arch-pc/lab04$
```

Рис. 4.9: Выполнение команды

5 Задание для самостоятельной работы

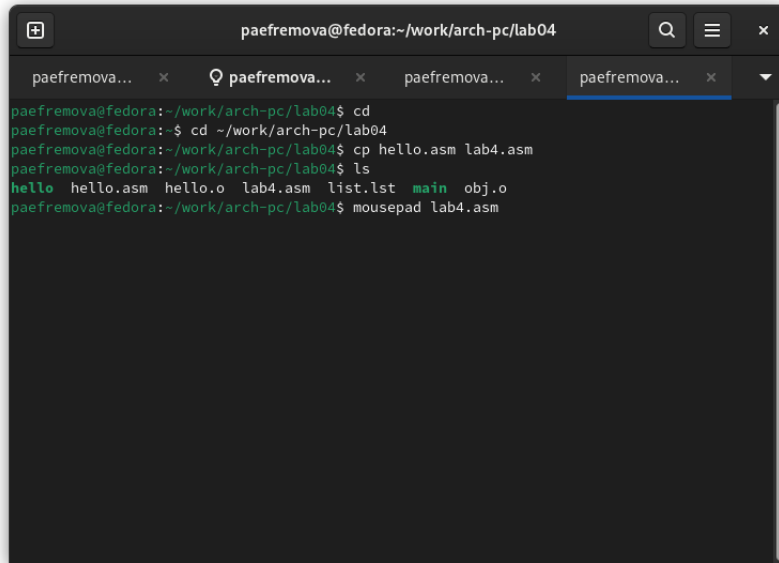
Перехожу в нужный каталог (рис. 5.1)



```
paefremova@fedora:~/work/arch-pc/lab04
paefremova@fedora:~/work/arch-pc/lab04$ cd
paefremova@fedora:~$ cd ~/work/arch-pc/lab04
paefremova@fedora:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
paefremova@fedora:~/work/arch-pc/lab04$
```

Рис. 5.1: Переход

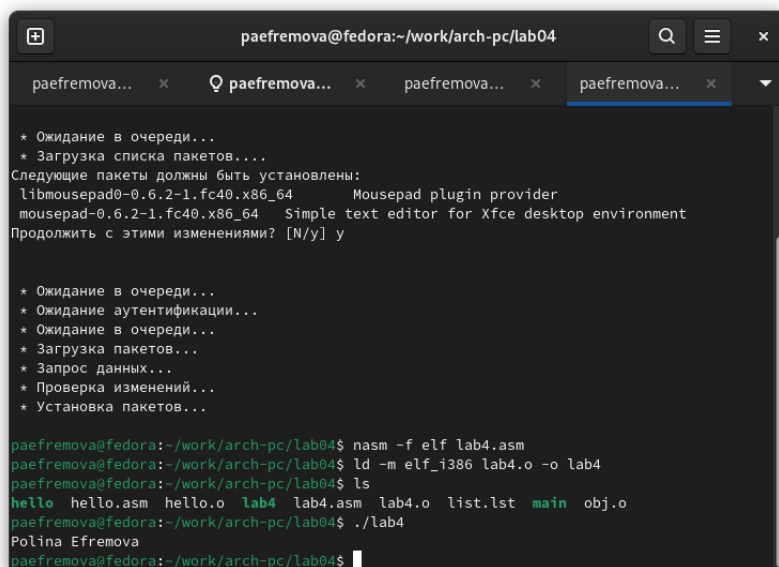
Создаю копию файла hello.asm с именем lab4.asm и начинаю его редактировать (рис. 5.2)



```
paefremova@fedora:~/work/arch-pc/lab04
paefremova@fedora:~/work/arch-pc/lab04$ cd
paefremova@fedora:~$ cd ~/work/arch-pc/lab04
paefremova@fedora:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
paefremova@fedora:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  lab4.asm  list.lst  main  obj.o
paefremova@fedora:~/work/arch-pc/lab04$ mousepad lab4.asm
```

Рис. 5.2: Копирование

С помощью любого текстового редактора вношу изменения в текст программы в файле lab4 и вывожу полученный результат (рис. 5.3)



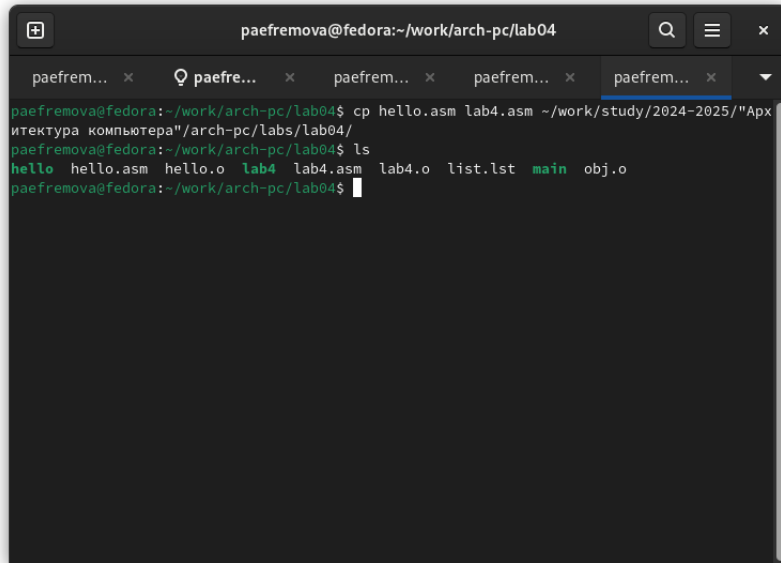
```
paefremova@fedora:~/work/arch-pc/lab04
* Ожидание в очереди...
* Загрузка списка пакетов...
Следующие пакеты должны быть установлены:
libmousepad0-0.6.2-1.fc40.x86_64      Mousepad plugin provider
mousepad-0.6.2-1.fc40.x86_64         Simple text editor for Xfce desktop environment
Продолжить с этими изменениями? [N/y] y

* Ожидание в очереди...
* Ожидание аутентификации...
* Ожидание в очереди...
* Загрузка пакетов...
* Запрос данных...
* Проверка изменений...
* Установка пакетов...

paefremova@fedora:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
paefremova@fedora:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
paefremova@fedora:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  lab4  lab4.asm  lab4.o  list.lst  main  obj.o
paefremova@fedora:~/work/arch-pc/lab04$ ./lab4
Polina Efremova
paefremova@fedora:~/work/arch-pc/lab04$
```

Рис. 5.3: Вывод собственной команды

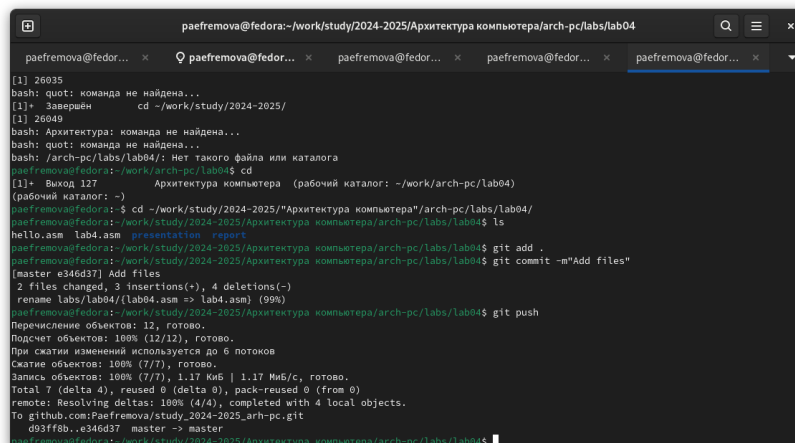
Копирую файлы в свой локальный репозиторий (рис. 5.4)



```
paefremova@fedora:~/work/arch-pc/lab04
paefremova@fedora:~/work/arch-pc/lab04$ cp hello.asm lab4.asm ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab04/
paefremova@fedora:~/work/arch-pc/lab04$ ls
hello  hello.o  lab4  lab4.asm  lab4.o  list.lst  main  obj.o
paefremova@fedora:~/work/arch-pc/lab04$
```

Рис. 5.4: Копирование файлов в локальный репозиторий

Провожу загрузку на GitHub (рис. 5.5)



```
paefremova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04
paefremova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ cd ~/work/study/2024-2025/
paefremova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ ls
hello.asm  lab4.asm  presentation  report
paefremova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ git add .
[master e346d37] Add files
2 files changed, 3 insertions(+), 4 deletions(-)
rename lab4/lab04/(lab04.asm => lab4.asm) (99%)
paefremova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ git commit -m"Add files"
paefremova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ git push
Перечисление объектов: 12, готово.
Подсчет объектов: 100% (12/12), готово.
При скатии изменений используется до 6 потоков
Сжатие объектов: 100% (7/7), готово.
Запись объектов: 100% (7/7), 1.17 КБ | 1.17 МБ/с, готово.
Total 7 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To github.com:Paefremova/study_2024-2025_arh-pc.git
d93ff0b..e346d37 master -> master
paefremova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$
```

Рис. 5.5: Загрузка на GitHub

6 Выводы

Данная лабораторная работа позволила мне познакомиться с таким языком как Ассемблер. Теперь я знаю как писать простейшие команды на данном языке через консоль. В заключение хочется отметить, что пцсть приведенный язык и не считается популярным, я все же убеждаюсь в его важности.

Список литературы

1. Ассемблер: что это за язык программирования, для чего нужен, пример кода
2. Архитектура ЭВМ
3. Что такое язык ассемблера и кому его нужно изучать/Skillbox Media