Шаблон отчёта по лабораторной работе

Простейший вариант

Дион Гонсан Седрик!

Содержание

# 1 Цель работы

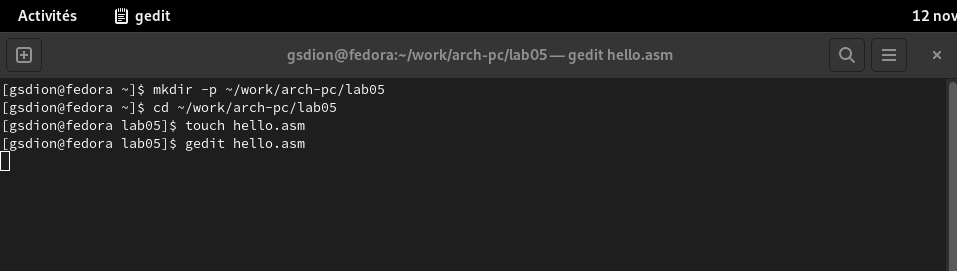
Освоение процедуры компиляции и сборки программ, написанных на ассем- блере NASM. # Задание

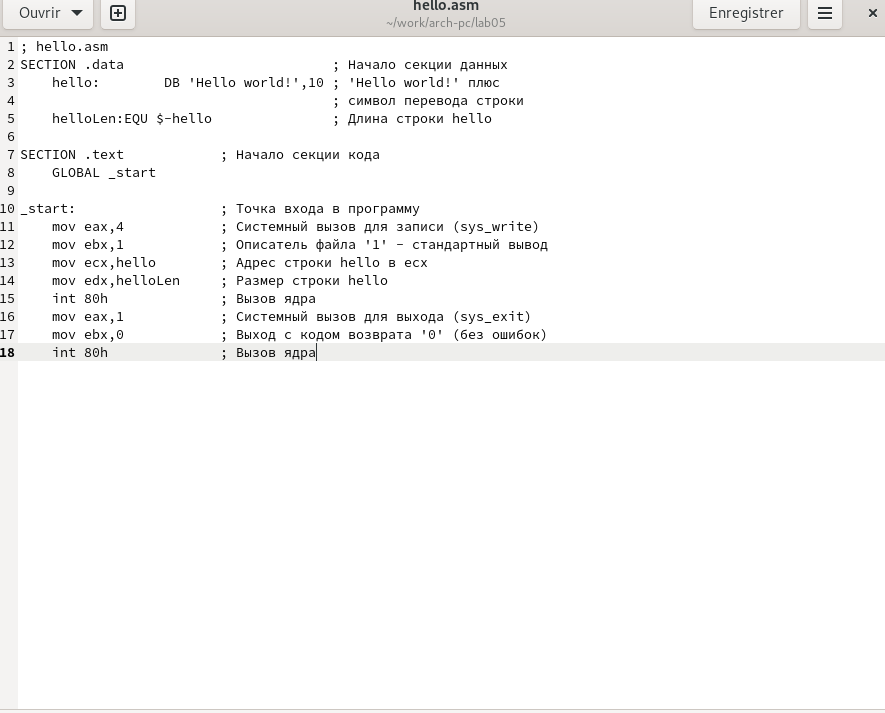
1. В каталоге ~/work/arch-pc/lab05 с помощью команды cp создайте копию файла hello.asm с именем lab5.asm
2. С помощью любого текстового редактора внесите изменения в текст про- граммы в файле lab5.asm так, чтобы вместо Hello world! на экран выво- дилась строка с вашими фамилией и именем.
3. Оттранслируйте полученный текст программы lab5.asm в объектный файл. Выполните компоновку объектного файла и запустите получивший- ся исполняемый файл.
4. Скопируйте файлы hello.asm и lab5.asm в Ваш локальный репозиторий в каталог ~/work/study/2022-2023/“Архитектура компьютера”/arch- pc/labs/lab05/. Загрузите файлы на Github.

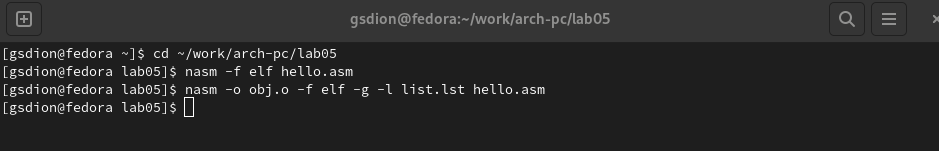
# 2 Теоретическое введение

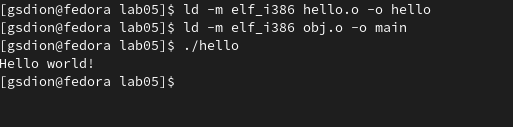
Основными функциональными элементами любой электронно-вычислительной машины (ЭВМ) являются центральный процессор, память и периферийные устройства (рис. 5.1). Взаимодействие этих устройств осуществляется через общую шину, к которой они подключены. Физически шина представляет собой большое количество про- водников, соединяющих устройства друг с другом. В современных компьютерах проводники выполнены в виде электропроводящих дорожек на материнской (системной) плате. 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 каче- стве операндов. Практически все команды представляют собой преобразование данных хранящихся в регистрах процессора, это например пересылка данных между регистрами или между регистрами и памятью, преобразование (арифме- тические 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 назначе- ния (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 (половинки 16-битных реги- стров). Например, AH (high AX) — старшие 8 бит регистра AX, AL (low AX) — младшие 8 бит регистра AX.

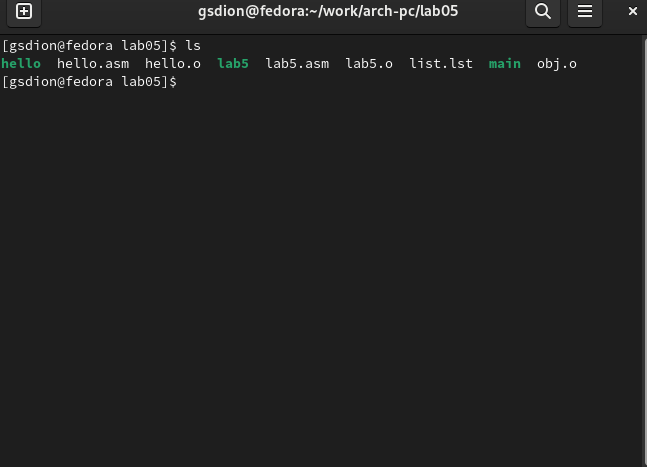
# 3 Выполнение лабораторной работы

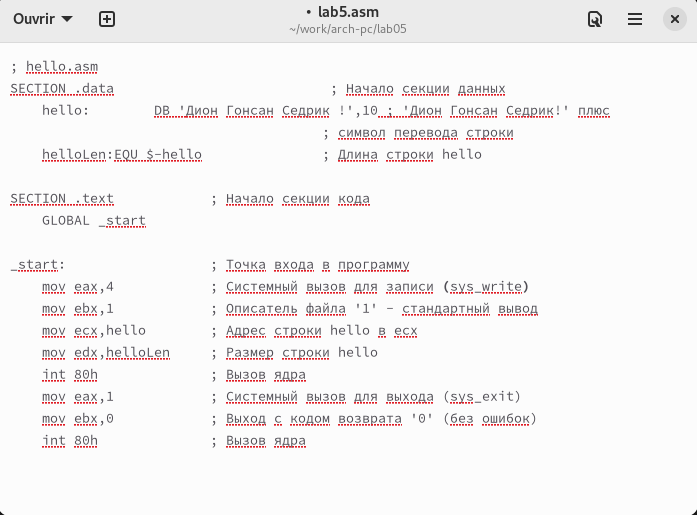
Создайте каталог для работы с программами на языке ассемблера NASM (рис1.**¿fig:001?**) 

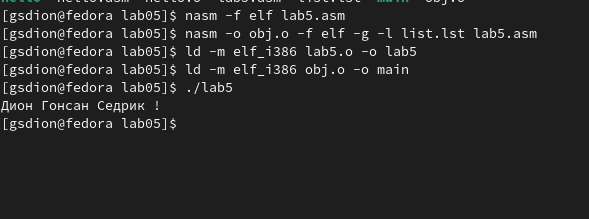
файл с помощью любого текстового редактора, например, gedit с помощью команды gedit hello.asm (рис2.**¿fig:002?**) 

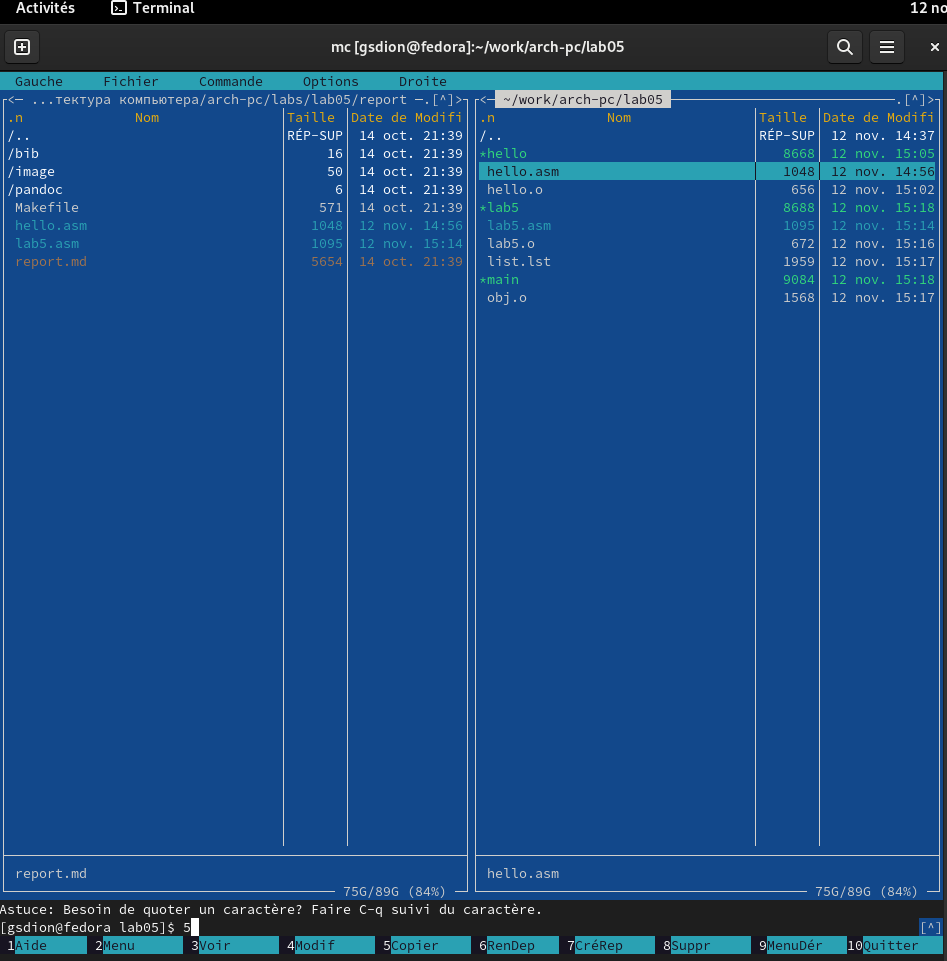
Транслятор NASMКомпоновщик LD (рис3. **¿fig:003?**) 

Компоновщик LD (рис4. **¿fig:004?**) 

В каталоге ~/work/arch-pc/lab05 с помощью команды cp создайте копию файла hello.asm с именем lab5.asm (рис8. **¿fig:008?**) 

С помощью любого текстового редактора внесите изменения в текст про- граммы в файле lab5.asm так, чтобы вместо Hello world! на экран выво- дилась строка с вашими фамилией и именем (рис5.**¿fig:005?**) 

Оттранслируйте полученный текст программы lab5.asm в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл. (рис6.**¿fig:006?**) 

Скопируйте файлы hello.asm и lab5.asm в Ваш локальный репозиторий в каталог ~/work/study/2022-2023/“Архитектура компьютера”/arch-pc/labs/lab05/. Загрузите файлы на Github.(рис7.**¿fig:007?**) 

# 4 Выводы

В ходе этой лабораторной работы я приобрел практический навык в освоении процедур компиляции и ассемблера программ, написанных на ассемблере NASM. # Список литературы{.unnumbered}