Отчет по лабораторной работе № 4

Дисциплина: архитектура компьютеров

Казазаев Даниил Михайлович

Содержание

# 1 Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM

# 2 Задания Лабораторной работы

1. Создать текстовый фалй с именем hello.asm
2. Ввести текст программы
3. Перевести текст программы в объектный код с помощью NASM
4. Скомпилировать файл hello.asm в obj.o
5. Передать объектный файл на обработку компоновщику
6. Запустить файл

# 3 Задания Самостоятельной работы

1. В каталоге ~/work/arch-pc/lab04 с помощью команды cp создайте копию файла hello.asm с именем lab4.asm
2. С помощью любого текстового редактора внесите изменения в текст программы в файле lab4.asm так, чтобы вместо Hello world! на экран выводилась строка с вашими фамилией и именем.
3. Оттранслируйте полученный текст программы lab4.asm в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл.
4. Скопируйте файлы hello.asm и lab4.asm в Ваш локальный репозиторий в каталог ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/. Загрузите файлы на Github.

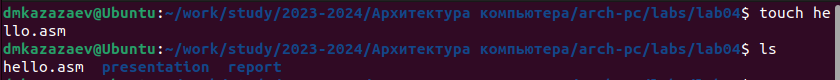
# 4 Выполнение лабораторной работы

Перехожу в нужную директорию для начала выполнения лабораторной работы. (рис. [??])

Переход в нужную директорию

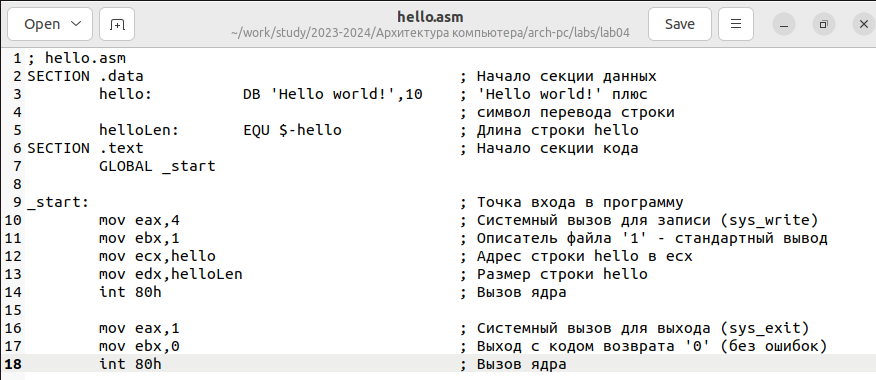
Переход в нужную директорию

Создаю файл hello.asm. (рис. [??])



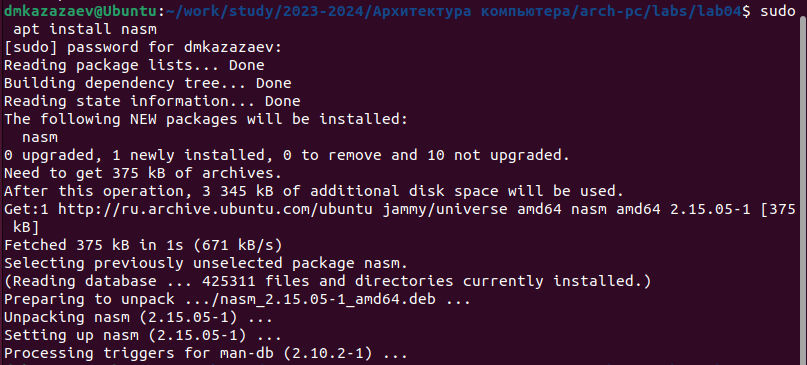
Создание файла hello.asm и проверка того, что он создан

Открываю файл и ввожу в него текст программы. (рис. [??])



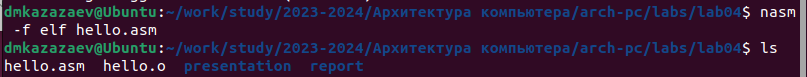
Введенный текст

Устанавливаю транслятор NASM. (рис. [??])



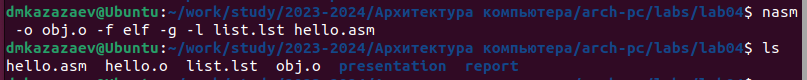
Установка NASM

Переношу файл hello.asm в объектный файл hello.o. (рис. [??])



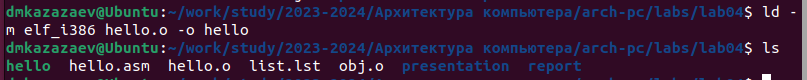
Перенос в объектный файл

Компилирую файл и создаю файл листинга list.lst. (рис. [??])



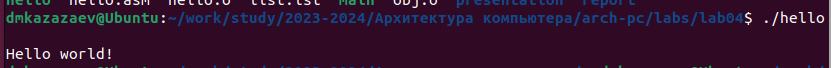
Компиляция файла, создание листинг файла и проверка

Передаю файл hello.o на обработку компоновщику. (рис. [??])



Передача компоновщику и проверка созданного файлаисполнителя

Запускаю исполняемый файл. (рис. [??])



Запущенный исполнительный файл

Листинг программы hello:

1 ; hello.asm  
 2 SECTION .data ; Начало секции данных  
 3 00000000 48656C6C6F20776F72- hello: DB 'Hello world!',10 ; 'Hello world!' плюс  
 3 00000009 6C64210A   
 4 ; символ перевода строки   
 5 helloLen: EQU $-hello ; Длина строки hello  
 6 SECTION .text ; Начало секции кода  
 7 GLOBAL \_start  
 8   
 9 \_start: ; Точка входа в программу  
 10 00000000 B804000000 mov eax,4 ; Системный вызов для записи (sys\_write)  
 11 00000005 BB01000000 mov ebx,1 ; Описатель файла '1' - стандартный вывод  
 12 0000000A B9[00000000] mov ecx,hello ; Адрес строки hello в ecx  
 13 0000000F BA0D000000 mov edx,helloLen ; Размер строки hello  
 14 00000014 CD80 int 80h ; Вызов ядра  
 15   
 16 00000016 B801000000 mov eax,1 ; Системный вызов для выхода (sys\_exit)  
 17 0000001B BB00000000 mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)  
 18 00000020 CD80 int 80h ; Вызов ядра

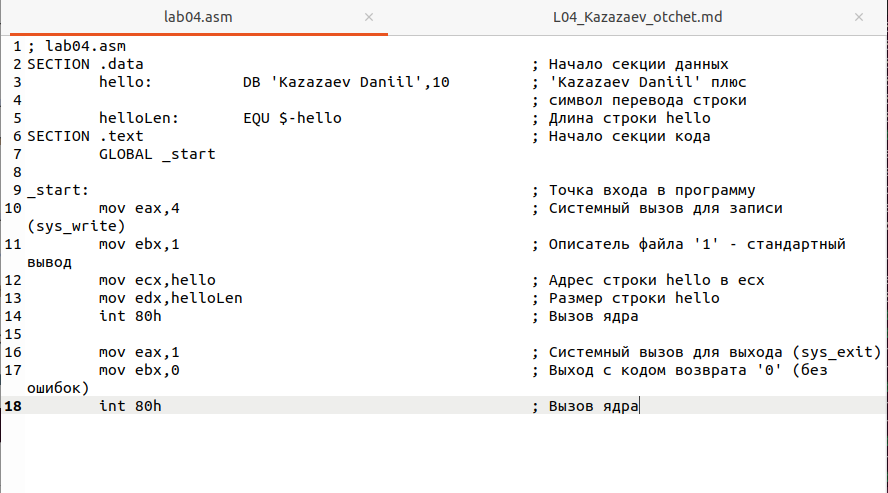
## 4.1 Выполнение заданий самостоятельной работы

Копирую файл hello.asm с названием lab04. (рис. [??])

Скопированный файл lab04.asm

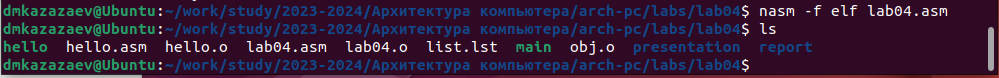
Скопированный файл lab04.asm

Редактирую файл. (рис. [??])



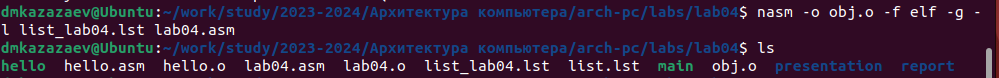
Отредактированный файл

Переношу файл lab04.asm в объектный файл lab04.o. (рис. [??])



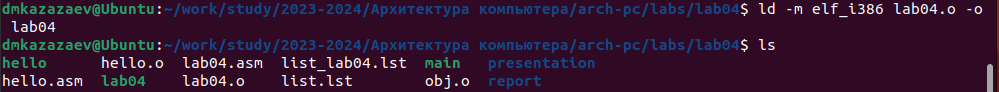
Перенос в объектный файл

Компилирую файл и создаю файл листинга list\_lab04.lst. (рис. [??])



Компиляция файла, создание листинг файла и проверка

Передаю файл lab04.o на обработку компоновщику. (рис. [??])



Передача компоновщику и проверка созданного файлаисполнителя

Запускаю исполняемый файл. (рис. [??])

Запущенный исполнительный файл

Запущенный исполнительный файл

Листинг программы lab04:

1 ; lab04.asm  
 2 SECTION .data ; Начало секции данных  
 3 00000000 4B617A617A61657620- hello: DB 'Kazazaev Daniil',10 ; 'Kazazaev Daniil' плюс  
 3 00000009 44616E69696C0A   
 4 ; символ перевода строки   
 5 helloLen: EQU $-hello ; Длина строки hello  
 6 SECTION .text ; Начало секции кода  
 7 GLOBAL \_start  
 8   
 9 \_start: ; Точка входа в программу  
 10 00000000 B804000000 mov eax,4 ; Системный вызов для записи (sys\_write)  
 11 00000005 BB01000000 mov ebx,1 ; Описатель файла '1' - стандартный вывод  
 12 0000000A B9[00000000] mov ecx,hello ; Адрес строки hello в ecx  
 13 0000000F BA10000000 mov edx,helloLen ; Размер строки hello  
 14 00000014 CD80 int 80h ; Вызов ядра  
 15   
 16 00000016 B801000000 mov eax,1 ; Системный вызов для выхода (sys\_exit)  
 17 0000001B BB00000000 mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)  
 18 00000020 CD80 int 80h ; Вызов ядра

# 5 Вывод

Итогом данной работы стала первая прогрограмма, которую я написал на языке ассемблера NASM