Лабораторная работа №4 Создание и процесс обработки программ на языке ассемблера NASM

Дисциплина: Архитектура ЭВМ

Шевырев Иван

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

2.1 Программа “Hello world!”

2.1.1 Создадим файл `hello.asm`

2.1.1.1 Создадим каталок lab05

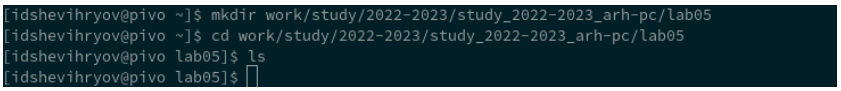

Рис. 2.1: Создание директори

2.1.1.2 Создадим текстовый файл lab05.asm

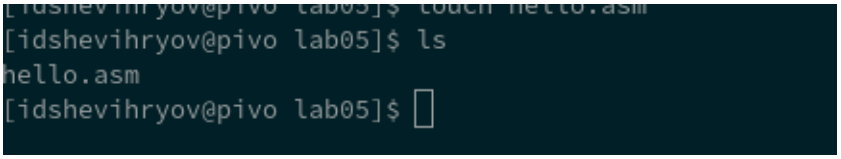

Рис. 2.2: Создание lab05.asm и комманда ls

2.1.1.3 Откроем файл через Gedit

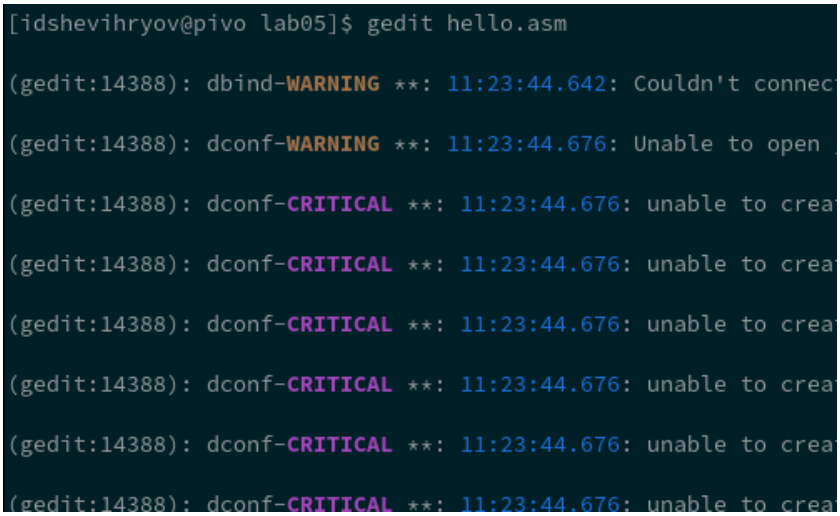

Рис. 2.3: Открытие файла через тектовый редактор gedit

2.1.1.4 Введем код в файл

```
1 ; hello.asm
2 SECTION .data ; Начало секции данных
3         hello: DB 'Hello world!',10 ; 'Hello world!' плюс
4                                     ; символ перевода строки
5
6         helloLen: EQU $-hello       ; Длина строки hello
7
8 SECTION .text                       ; Начало секции кода
9         GLOBAL _start
10
11 _start:       ; Точка входа в программу
12        mov eax,4                   ; Системный вызов для записи (sys_write)
13        mov ebx,1                   ; Описатель файла '1' - стандартный вывод
14        mov ecx,hello               ; Адрес строки hello в ecx
15        mov edx,helloLen            ; Размер строки hello
16        int 80h                     ; Вызов ядра
17
18        mov eax,1                   ; Системный вызов для выхода (sys_exit)
19        mov ebx,0                   ; Выход с кодом возврата '0' (без ошибок)
20        int 80h                     ; Вызов ядра
21
```

Рис. 2.4: Код введеный в файл

2.1.2 Воспользуемся транслятором NASM

Скомпилируем вышенаписанную программу с помощью комманды `nasm -f elf hello.asm`

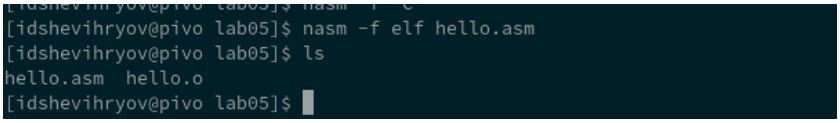

Рис. 2.5: Трансляция кода

Создадим файл с другим именем ипользуя `-o` и создадим листинг, с помощью `-l`

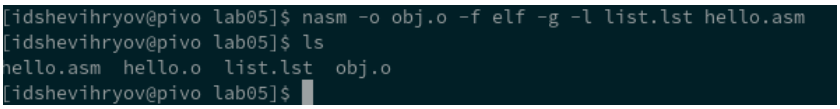

Рис. 2.6: Создание объектного файла с другим именем

2.1.3 Воспользуемся компоновщиком LD

Выполним комманду `ld -m elf_i386 hello.o -o hello`

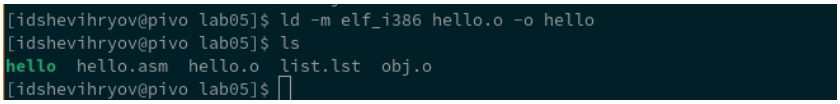

Рис. 2.7: компоновщик `ld`

С помощью комманды ls, увидим что файл hello создался

2.1.4 Скомпонуем в файл с именем `main` с помощью ключа `-o`

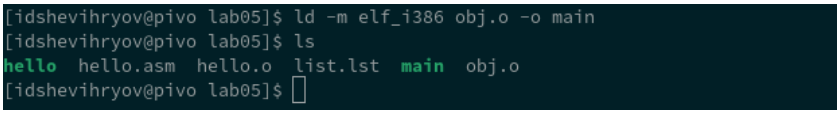

Рис. 2.8: Исполняемый файл с именем main

2.1.5 Запустим исполняемый файл

Рис. 2.9: Запуск ./hello

На экран вывелось “Hello world!”

3 Задания для самостоятельной работы

3.0.1 Создадим копию файла

Скопируем содержимое из файла `hello.asm` в `lab5.asm`

```
[idshevihryov@pivo study_2022-2023_arh-pc]$ cp lab05/hello.asm labs/lab05/lab5.asm
[idshevihryov@pivo study_2022-2023_arh-pc]$ ls labs/lab05/
lab5.asm  presentation  report
[idshevihryov@pivo study_2022-2023_arh-pc]$
```

Рис. 3.1: Копирование файлов

3.0.2 Изменим содержимое файла

С помощью редактора Gedit заменим строку с “Hello world” на свое имя и фамилию

```
1 ; hello.asm
2 SECTION .data ; Начало секции данных
3         hello: DB 'Шевырев Иван',10 ; 'Шевырев Иван' плюс
4                                     ; символ перевода строки
5
6         helloLen: EQU $-hello       ; Длина строки hello
7
```

Рис. 3.2: Редактирование через текстовый редактор gedit

3.0.3 Возспользуемся транслятором Nasm

```
[idshevihryov@pivo lab05]$ nasm -o lab5.o -f elf -g -l list.lst lab5.asm
[idshevihryov@pivo lab05]$ ls
lab5.asm  lab5.o  list.lst  presentation  report
[idshevihryov@pivo lab05]$
```

Рис. 3.3: Трансляция lab05

3.0.4 Воспользуемся компоновщиком

С помощью комманды ld создадим исполняемый файл `lab5`

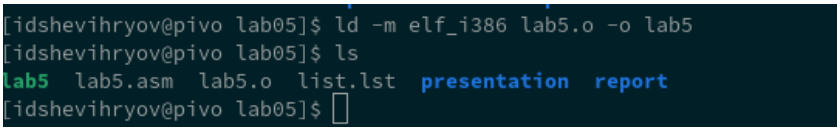

Рис. 3.4: Использование комманды ld

3.0.5 Запустим `./lab05`

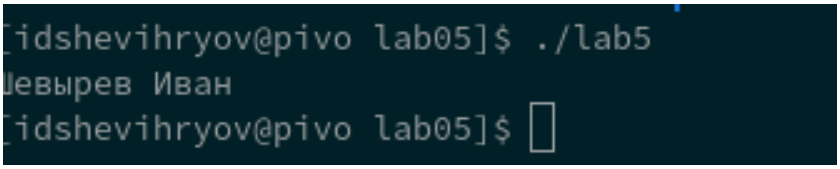

Рис. 3.5: Исполнение lab05

Видим как на экран выводятся наша имя и фамилия

3.1 Загрузим файлы на GitHub

3.1.1 Скопируем файлы в локальный репозиторий

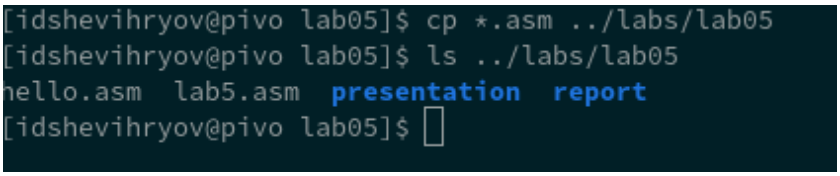

Рис. 3.6: Копирование всех asm файлов в локальный репозиторий

3.1.2 Создадим git commit

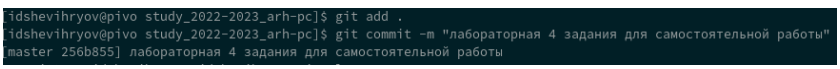

Рис. 3.7: git add и git commit

3.1.3 Выгрузим файлы на гитхаб

используем комманду git push

Рис. 3.8: git push

4 Выводы

За эту лабораторной работы мы научились переводить программы на языке ассемблера NASM в исполняемый файл с помощью трансляции и компановки через ld.

Мы написали программу на NASM, которая ввыводит в терминал нашу фамилию и имя.