

Лабораторная работа №5 Создание и процесс обработки программ на языке ассемблера NASM

Дисциплина: Архитектура ЭВМ

Алексей Назаров

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Программа Hello world!	6
2.1.1	Создание файла	6
2.1.2	Написание программы	6
2.1.3	Использование Транслятора	7
2.1.4	Использование компоновщика LD	8
2.1.5	Запустим исполняемый файл	8
2.2	Задания для самостоятельной работы	8
3	Выводы	11
	Список литературы	12

Список иллюстраций

2.1	Создание lab05	6
2.2	переход в lab05	6
2.3	Создание hello.asm	6
2.4	Открытие через gedit	7
2.5	Использование транслятора	7
2.6	Использование расширенного синтаксиса транслятора	7
2.7	Использование компоновщика	8
2.8	Использование компоновщика с измененными параметрами	8
2.9	Запуск исполняемого файла	8
2.10	Копирование файлов	9
2.11	Редактирование файла	9
2.12	Трансляция lab5.asm	9
2.13	Компановка lab5.o	9
2.14	Копирование папки	10
2.15	Использование git	10

Список таблиц

1 Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM

2 Выполнение лабораторной работы

2.1 Программа Hello world!

2.1.1 Создадие файла

Создадим каталог lab05

```
[amnazarov@localhost ~]$ mkdir ~/work/study/2022-2023/Архитектура\
компьютера/arch-pc/lab05
```

Рис. 2.1: Создание lab05

Перейдем в него

```
[amnazarov@localhost ~]$ cd ~/work/study/2022-2023/Архитектура\ ко
мпьютера/arch-pc/lab05/
[amnazarov@localhost lab05]$ █
```

Рис. 2.2: переход в lab05

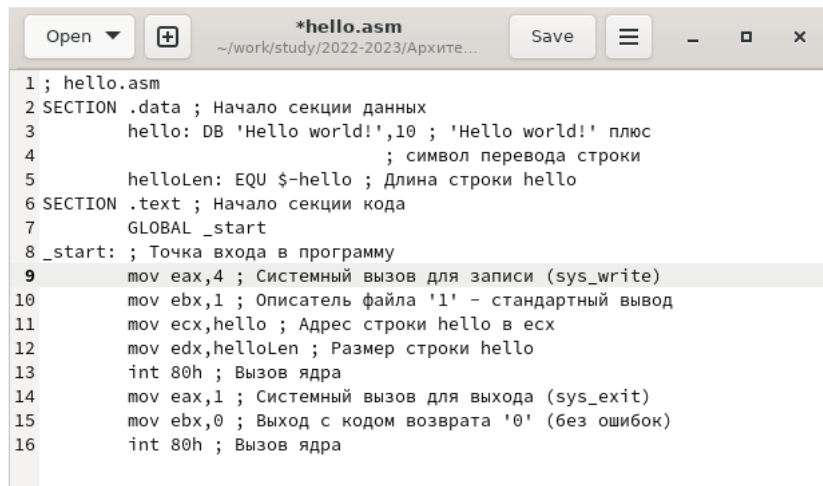
Создадим текстовый файл с именем hello.asm

```
[amnazarov@localhost lab05]$ touch hello.asm
[amnazarov@localhost lab05]$ ls
hello.asm
[amnazarov@localhost lab05]$ █
```

Рис. 2.3: Создание hello.asm

2.1.2 Написание программы

Откроем файл через gedit



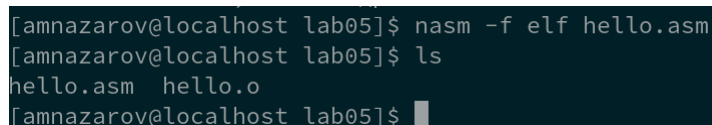
```
1 ; hello.asm
2 SECTION .data ; Начало секции данных
3     hello: DB 'Hello world!',10 ; 'Hello world!' плюс
4             ; символ перевода строки
5     helloLen: EQU $-hello ; Длина строки hello
6 SECTION .text ; Начало секции кода
7     GLOBAL _start
8 _start: ; Точка входа в программу
9     mov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

Рис. 2.4: Открытие через gedit

Введем в файл текст и сохраним

2.1.3 Использование Транслятора

Скомпилируем hello.asm. Введем команду в терминале. Проверим с помощью команды ls, что создался hello.o

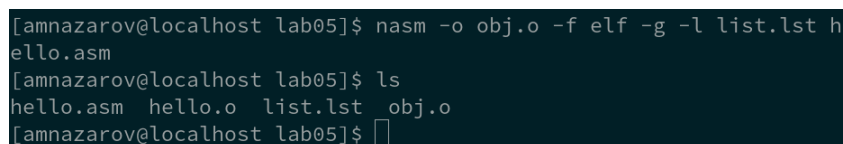


```
[amnazarov@localhost lab05]$ nasm -f elf hello.asm
[amnazarov@localhost lab05]$ ls
hello.asm  hello.o
[amnazarov@localhost lab05]$
```

Рис. 2.5: Использование транслятора

2.1.3.1 Использование расширенного синтаксиса NASM

Выполним следующую команду, чтобы создать объектный файл с именем obj.o и создать листинг



```
[amnazarov@localhost lab05]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
[amnazarov@localhost lab05]$ ls
hello.asm  hello.o  list.lst  obj.o
[amnazarov@localhost lab05]$
```

Рис. 2.6: Использование расширенного синтаксиса транслятора

Проверим с помощью команды ls, что создались obj.o и list.lst

2.1.4 Использование компоновщика LD

Передадим файл на компоновку с помощью команды в консоли

```
[amnazarov@localhost lab05]$ ld -m elf_i386 hello.o -o hello
[amnazarov@localhost lab05]$ ls
hello hello.asm hello.o list.lst obj.o
[amnazarov@localhost lab05]$
```

Рис. 2.7: Использование компоновщика

С помощью команды ls, убедимся, что файл hello создан

2.1.4.1 Воспользуемся дополнительными расширениями компоновщика

выполним команду, с опцией -o main

```
[amnazarov@localhost lab05]$ ld -m elf_i386 obj.o -o main
[amnazarov@localhost lab05]$ ls
hello hello.asm hello.o list.lst main obj.o
[amnazarov@localhost lab05]$
```

Рис. 2.8: Использование компоновщика с измененными параметрами

Как видим, создан файл с main

2.1.5 Запустим исполняемый файл

Исполним команду

```
[amnazarov@localhost lab05]$ ./hello
Hello world!
[amnazarov@localhost lab05]$
```

Рис. 2.9: Запуск исполняемого файла

На экран вывелось Hello World

2.2 Задания для самостоятельной работы

Скопируем файл hello.asm в lab5.asm и откроем его в текстовом редакторе


```

[amnazarov@localhost lab05]$ cp hello.asm lab5.asm
[amnazarov@localhost lab05]$ gedit lab5.asm

```

Рис. 2.10: Копирование файлов

Заменим “Hello World” на “Назаров Алексей”

```

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

```

Рис. 2.11: Редактирование файла

Оттранслируем lab5.asm в объектный файл

```

[amnazarov@localhost lab05]$ nasm -f elf lab5.asm
[amnazarov@localhost lab05]$ ls
hello      hello.o    lab5.o     main
hello.asm  lab5.asm  list.lst  obj.o
[amnazarov@localhost lab05]$ 

```

Рис. 2.12: Трансляция lab5.asm

Выполним компоновку объектного файла

```

[amnazarov@localhost lab05]$ ld -m elf_i386 lab5.o -o lab5
[amnazarov@localhost lab05]$ ls
hello      hello.o    lab5.asm  list.lst  obj.o
hello.asm  lab5      lab5.o    main
[amnazarov@localhost lab05]$ ./lab5
Назаров Алексей
[amnazarov@localhost lab05]$ 

```

Рис. 2.13: Компоновка lab5.o

Перекопируем папку в локальный репозиторий

```
[amnazarov@localhost arch-pc]$ cp -r lab05/ labs/
[amnazarov@localhost arch-pc]$ ls labs/lab05
hello      hello.o  lab5.asm  list.lst  obj.o      report
hello.asm  lab5     lab5.o    main      presentation
```

Рис. 2.14: Копирование папки

Закоммитим изменения

```
[amnazarov@localhost lab05]$ git add .
[amnazarov@localhost lab05]$ git commit -m "Скопировал файлы для 5
лабораторной в репозиторий"
[master 7decce0] Скопировал файлы для 5 лабораторной в репозиторий
9 files changed, 49 insertions(+)
create mode 100755 labs/lab05/hello
create mode 100644 labs/lab05/hello.asm
create mode 100644 labs/lab05/hello.o
create mode 100755 labs/lab05/lab5
create mode 100644 labs/lab05/lab5.asm
create mode 100644 labs/lab05/lab5.o
create mode 100644 labs/lab05/list.lst
create mode 100755 labs/lab05/main
create mode 100644 labs/lab05/obj.o
[amnazarov@localhost lab05]$
```

Рис. 2.15: Использование git

Загрузим файлы на GitHub командой git push

3 Выводы

Я освоил процедуры компиляции и сборки программ, написанных на ассемблере NASM, а также улучшил свои навыки в работе с текстовым редактором Gedit

Список литературы