

Лабораторная работа №5

Зарипов Евгений Сергеевич

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	9

Список иллюстраций

2.1	Создаем необходимый файл	6
2.2	Перепишем представленный в лабораторной работе код на языке “насм” в открывшийся файл	6
2.3	Протранслируем файл <code>hello.nasm</code>	6
2.4	Тут мы получаем объектный файл и файл “листинга”, компилируем исполняемый файл	7
2.5	Создаем исполняемый файл из объектного	7
2.6	Запускаем исполняемый файл	7
2.7	Редактируем открытый файл <code>lab05.asm</code>	7
2.8	Создаем объектный файл и “листинг”. Компилируем его под именем <code>lab05</code> , после чего запускаем. После запуска получаем необходимый результат.	8

Список таблиц

1 Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM

2 Выполнение лабораторной работы

```
[ezaripov@fedora ~]$ cd ~/work/study/2022-2023/Архитектура\ компьютера/study_2022-2023_arh-pc/
[ezaripov@fedora study_2022-2023_arh-pc]$ ls
CHANGELOG.md  config  COURSE  labs  LICENSE  Makefile  prepare  README.en.md  README.git-flow.md  README.md  template
[ezaripov@fedora study_2022-2023_arh-pc]$ cd labs/lab05
[ezaripov@fedora lab05]$ ls
presentation  report
[ezaripov@fedora lab05]$ touch hello.asm
[ezaripov@fedora lab05]$ ls
hello.asm  presentation  report
[ezaripov@fedora lab05]$ gedit hello.asm
```

Рис. 2.1: Создаем необходимый файл

```
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. 2.2: Перепишем представленный в лабораторной работе код на языке “насм” в открывшийся файл

```
[ezaripov@fedora lab05]$ nasm -f elf hello.asm
[ezaripov@fedora lab05]$ ls
hello.asm  hello.o  presentation  report
[ezaripov@fedora lab05]$
```

Рис. 2.3: Протранслируем файл hello.nasm

```

[ezaripov@fedora lab05]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
[ezaripov@fedora lab05]$ ls
hello.asm hello.o list.lst obj.o presentation report
[ezaripov@fedora lab05]$ d -m elf_i386 hello.o -o hello
bash: d: команда не найдена...
[ezaripov@fedora lab05]$ ls
hello.asm hello.o list.lst obj.o presentation report
[ezaripov@fedora lab05]$ ld -m elf_i386 hello.o -o hello
bash: ld: команда не найдена...
Аналогичная команда: 'ld'
[ezaripov@fedora lab05]$ ls
hello.asm hello.o list.lst obj.o presentation report
[ezaripov@fedora lab05]$ ld -m elf_i386 hello.o -o hello
[ezaripov@fedora lab05]$ ls
hello hello.asm hello.o list.lst obj.o presentation report
[ezaripov@fedora lab05]$

```

Рис. 2.4: Тут мы получаем объектный файл и файл “листинга”, компилируем исполняемый файл

```

[ezaripov@fedora lab05]$ ld -m elf_i386 obj.o -o main
[ezaripov@fedora lab05]$ ls
hello hello.asm hello.o list.lst main obj.o presentation report
[ezaripov@fedora lab05]$

```

Рис. 2.5: Создаем исполняемый файл из объектного

```

[ezaripov@fedora lab05]$ ./hello
Hello world!
[ezaripov@fedora lab05]$

```

Рис. 2.6: Запускаем исполняемый файл

#Самостоятельная работа

Копируем файл hello.asm, как lab05.asm и открываем его

```

1 ; hello.asm
2 SECTION .data ; Начало секции данных
3 hello: DB 'Zaripov Evgenii',10 ; 'name and surname' 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. 2.7: Редактируем открытый файл lab05.asm

```

[ezaripov@fedora lab05]$ nasm -o lab05.o -f elf -g -l list_lab.lst lab05.asm
[ezaripov@fedora lab05]$ ls
hello hello.asm hello.o lab05.asm lab05.o lab.o list_lab.lst list.lst main obj_lab.o obj.o presentation
[ezaripov@fedora lab05]$ ld -m elf_i386 lab05.o -o lab05
[ezaripov@fedora lab05]$ ls
hello hello.asm hello.o lab05 lab05.asm lab05.o lab.o list_lab.lst list.lst main obj_lab.o obj.o presen
[ezaripov@fedora lab05]$ ./lab05
Zaripov Evgenii
[ezaripov@fedora lab05]$

```

Рис. 2.8: Создаем объектный файл и “листинг”. Компилируем его под именем lab05, после чего запускаем. После запуска получаем необходимый результат.

3 Выводы

Я получил навыки компиляции и сборки программ, написанных на ассемблере
NASM