

Отчёт по лабораторной работе №5

Процесс создания и обработки программ на языке ассемблера NASM

Мулин Иван Владимирович

Содержание

1	Цель работы	4
2	Задание	5
3	Ход работы	6
3.1	Выполнение лабораторной работы	6
3.2	Выполнение заданий для самостоятельной работы	8
4	Листинги написанных программ	9
5	Заключение	11

Список иллюстраций

3.1	Проверка трансляции файлов	6
3.2	Трансляция файлов с дополнительными опциями	7
3.3	Сборка программы	7
3.4	Исполнение полученной программы	7
3.5	Сборка программы с добавочными параметрами	7
3.6	Создание программы lab05	8
3.7	Выполнение программы	8

1 Цель работы

В результате выполнения лабораторной работы № 5 необходимо освоить процесс компиляции и сборки программ на языке ассемблера NASM. Данный отчёт, файлы написанных программ и изображения, фиксирующие ход выполнения лабораторной работы, можно найти по адресу https://github.com/ivmulin/study_2022-2023_arch-pc.

2 Задание

Написать и скомпилировать программы на языке ассемблера NASM.

3 Ход работы

3.1 Выполнение лабораторной работы

В рабочем каталоге `vendor` в файле `hell.asm` нужно написать программу. (Листинги программ приведены ниже.) Командой

```
nasm -f elf hell.asm
```

создаётся объектный файл `hell.o` в формате `elf`.

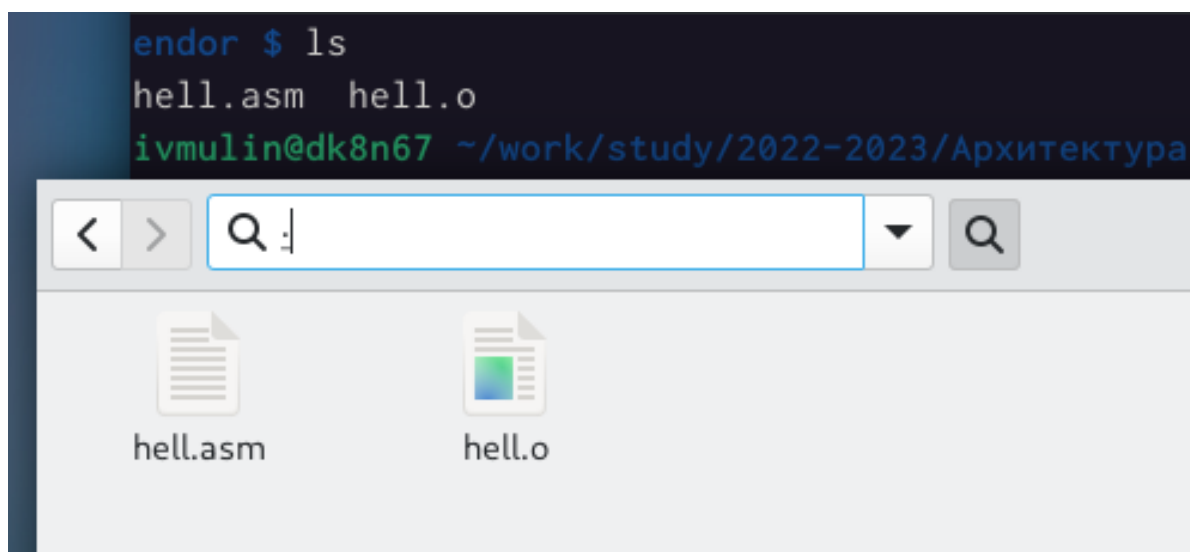


Рис. 3.1: Проверка трансляции файлов

Компиляцию программы можно выполнять и с другими ключами. К примеру, команда

```
nasm -f elf hell.asm -l list.lst -g -o obj.o
```

создаст файл листинга `list.lst` (за это отвечает опция `-l list.lst`) и объектный файл с включёнными в него символами для отладки (`-g -o obj.o`).

```
/vendor $ ls
hell.asm hell.o list.lst obj.o
```

Рис. 3.2: Трансляция файлов с дополнительными опциями

Далее стоит собрать программу и проверить её работоспособность:

```
/vendor $ ld -m elf_i386 hell.o -o hell
ivmulin@dk8n67 ~/work/study/2022-2023/Архитектура компь
/vendor $ ls
hell hell.asm hell.o list.lst obj.o
```

Рис. 3.3: Сборка программы

```
/vendor $ ./hell
Hellp me please!
```

Рис. 3.4: Исполнение полученной программы

Следующая команда:

```
ld -m elf_i386 hell.o -o hell
```

по выполнении из объектного файла `obj.o` собирает программу `main`:

```
/vendor $ ld -m elf_i386 obj.o -o main
ivmulin@dk8n67 ~/work/study/2022-2023/Архитектура компь
/vendor $ ls
hell hell.asm hell.o list.lst main obj.o
```

Рис. 3.5: Сборка программы с добавочными параметрами

3.2 Выполнение заданий для самостоятельной работы

В копии `lab05.asm` написанной программы `hell.asm` необходимо изменить строку, которую будет выводить система, так, чтобы отображались фамилия и имя автора отчёта. Далее нужно собрать программу:

```
/vendor $ cp hell.asm lab05.asm
ivmulin@dk8n67 ~/work/study/2022-2023/Архитектура компь
/vendor $ nasm -f elf lab05.asm -o lab05.o
ivmulin@dk8n67 ~/work/study/2022-2023/Архитектура компь
/vendor $ ld -m elf_i386 lab05.o -o lab05
```

Рис. 3.6: Создание программы `lab05`

Запустим полученный файл:

```
/vendor $ ./lab05
Мулина Ивана работа
```

Рис. 3.7: Выполнение программы

4 Листинги написанных программ

1. hell.asm

```
section .data ; Секция данных
```

```
    line: db 'Hellp me please!',10 ; строка плюс символ перевода строки  
    lineLength: EQU $-line ; Длина строки line
```

```
section .text ; Начало секции кода
```

```
    global _start
```

```
_start: ; Точка входа в программу
```

```
    mov eax, 4 ; Системный вызов для записи (sys_write)  
    mov ebx, 1 ; Описатель файла '1' - стандартный вывод  
    mov ecx, line ; Адрес строки line в ecx  
    mov edx, lineLength ; Размер строки hello  
    int 80h ; Вызов ядра  
    mov eax, 1 ; Системный вызов для выхода (sys_exit)  
    mov ebx, 0 ; Выход с кодом возврата '0' (без ошибок)  
    int 80h ; Вызов ядра
```

2. lab05.asm

```
section .data ; Секция данных
```

```
    line: db 'Мулина Ивана работа',10 ; строка плюс символ перевода строки  
    lineLength: EQU $-line ; Длина строки line
```

```

section .text ; Начало секции кода

    global _start

_start: ; Точка входа в программу

    mov eax, 4 ; Системный вызов для записи (sys_write)
    mov ebx, 1 ; Описатель файла '1' - стандартный вывод
    mov ecx, line ; Адрес строки line в ecx
    mov edx, lineLength ; Размер строки hello
    int 80h ; Вызов ядра

    mov eax, 1 ; Системный вызов для выхода (sys_exit)
    mov ebx, 0 ; Выход с кодом возврата '0' (без ошибок)
    int 80h ; Вызов ядра

```

5 Заключение

За время выполнения лабораторной работы № 5 был изучен процесс компиляции и сборки программ, написанных на языке ассемблера NASM. Задания для самостоятельной работы были преимущественно нацелены на закрепление полученных навыков.

Таким образом, цель данной работы была полностью достигнута.